Multitrack Compressed Sensing for Faster Hyperspectral Imaging

Sharvaj Kubal, Elizabeth Lee, Chor Yong Tay, and Derrick Yong

Abstract—Hyperspectral imaging (HSI) provides much more information than regular optical microscopy and spectroscopy, making it valuable in areas such as biomedicine, materials inspection and food safety. However HSI is challenging because of the large amount of data that has to be acquired, and large measurement times. Compressed sensing (CS) approaches to hyperspectral imaging have been developed to address this, albeit subject to tradeoffs between image reconstruction accuracy, time and generalizability to different types objects or scenes.

Here, we develop improved compressed sensing approaches for HSI, based on parallelized multitrack acquisition of multiple spectra per shot. This is an augmentation of single-pixel-camera-style acquisition for HSI, where a single spectrum is measured per shot. The multitrack architecture can be paired up with either of the two compatible CS algorithms developed here: (1) a sparse recovery algorithm based on block compressed sensing, and (2) an adaptive CS algorithm based on sampling in the wavelet domain. As a result, the measurement speed can be drastically increased, while maintaining reconstruction speed as well as accuracy.

The methods are validated computationally, via noiseless as well as noisy simulated measurements. Multitrack adaptive CS has a ∼10 times smaller measurement plus reconstruction time as compared to full sampling HSI, without compromise to reconstruction accuracies across the different sample images tested. Multitrack non-adaptive CS (sparse recovery) suffers a large reconstruction time, but is the most robust to Poisson noise.

Index Terms—Hyperspectral imaging; compressed sensing; wavelets; adaptive imaging.

1 INTRODUCTION

Hyperspectral images comprise of light intensity information from an object or scene, resolved into two spatial dimensions and a spectral (wavelength) dimension, forming a datacube. Such images can be interpreted as a collection of (i) electromagnetic spectra associated with each spatial pixel, or (ii) grayscale images corresponding to each of the spectral channels.

Hyperspectral imaging (HSI) – the capture of hyperspectral images – proves beneficial in areas as diverse as biomedicine, materials defect detection, agriculture monitoring and food safety, beyond the more traditional application: remote sensing of terrain. Variants of HSI such as Raman chemical imaging, where each spatial pixel records a Raman spectrum, are useful as label-free compositional analysis methods for spatially heterogeneous samples. Such tools can prove valuable in biopharmaceutical and cell manufacturing.

The biopharmaceutical industry has been trending towards integrated continuous manufacturing, where real-time measurements and process control are critical in ensuring and optimizing the quality of the products. This is particularly important due to the stringent regulatory landscape. Here, optical techniques such as HSI can provide information-rich yet non-contact methods for monitoring and inspection.

1.1 General challenges in HSI

Hyperspectral imaging presents a challenge due to the large amount of data that has to be acquired (and subsequently stored), and large amounts of time required to acquire this data. Established methods that fully sample the hyperspectral scene include point-scan, line-scan (pushbroom), wavelength-scan and snapshot imaging. Scanning methods can be time-consuming, especially for low-light imaging where detector integration times need to be large. Also, these methods often involve moving parts. Furthermore, if the scene to be imaged is dynamic, scanning can produce artifacts that are difficult to correct during postprocessing. On the other hand, snapshot imaging methods, which record the whole image in one shot, involve very large detector arrays and complex optical arrangements such as lenslet arrays and filter arrays. Overall, there is a necessity of subsampling (or undersampling) approaches to HSI, such that measurements can be made faster and more efficient, without losing critical information in the process.

1.2 Compressed sensing for HSI

A standard approach to subsampling of signals is compressed sensing (CS), also known as compressive sensing.

1For clarity in description, we distinguish between the terms hyperspectral scene and hyperspectral image. The former will refer to the object or scene that is to be imaged hyperspectrally; the latter will refer to the datacube captured by an imager.
or compressive sampling\textsuperscript{[9], [10]}. The theory of compressed sensing prescribes a subsampling procedure, together with a reconstruction algorithm, which recovers the signal (the datacube in our case) by making use of some prior assumptions about the structure of the signal. Recent HSI approaches have been leveraging this theory in a variety of ways\textsuperscript{[11]}. Coded aperture snapshot spectral imaging (CASSI)\textsuperscript{[12]– [14]} provides a prominent class of imaging architectures for compressed sensing HSI (CS-HSI), that can achieve snapshot measurement – capturing all the necessary data in one shot instead of using a sequence of measurements. Another important HSI architecture is the single pixel camera (SPC)-style hyperspectral imager\textsuperscript{[15]}, which is a direct extension of the SPC by replacing the single pixel detector with a spectrometer.

Classes of CS algorithms commonly employed are sparse recovery\textsuperscript{[16], [17]} (which is frequently augmented by dictionary learning\textsuperscript{[18]}), neural network-based reconstruction\textsuperscript{[19]– [22]}, and adaptive basis scan (or adaptive direct sampling)\textsuperscript{[23]– [26]}. Sparse recovery algorithms tend to have good performance guarantees, but generally are computationally intensive, making them slow for large signals like hyperspectral scenes. Neural network-based reconstruction, on the other hand, is much faster. Neural networks use training signals to learn an inverse transformation from the subsampled measurements back to the original signal, hence avoiding slow optimization steps post training. However, a lack of sufficient training data, or mismatches between the distributions of the training set and the set of target signals, can limit the generalizability and the scope of this method. Lastly, adaptive basis scan is based on progressive prediction and sampling of the most significant coefficients of a signals in some basis. These prediction steps are computationally lighter than the iterative steps involved in sparse recovery, and as a result, adaptive basis scan also achieves high reconstruction speed. Moreover, it is not limited by training data, unlike neural-network-based reconstruction.

1.3 Problem statement

In contemporary manufacturing, there is an increasing need for high-volume production with extensive personalization\textsuperscript{[27]}. This requires responsive yet autonomous operations together with end-to-end integration of the various processes. The monitoring and inspection systems involved in such manufacturing would require real-time processing and feedback while maintaining reliability, robustness and consistency\textsuperscript{[28]}.

CASSI is not very suitable for these purposes, since it cannot be used with adaptive basis scan (to our knowledge); it fails the ‘fast reconstruction’ criterion when using sparse recovery, and the ‘generalizability’ criterion when using neural-networks for reconstruction. The alternative here, the SPC-style hyperspectral imager, is also limited in its capability as it only measures one spectrum in a shot. Hence, a large number of shots are needed to obtain sufficient information for a good reconstruction of the hyperspectral datacube. As a consequence, the measurement times are high despite the fact that reconstruction times can be reduced by computationally light algorithms. Overall, there is a need for HSI methods that can achieve fast measurement as well as reconstruction, while maintaining imaging accuracy and generalizability across various types of scenes.

1.4 Contributions and outline

Here, we formulate improved compressed sensing approaches for HSI, which overcome the aforementioned limitations of measurement and reconstruction speeds. This is achieved by multitrack compressed sensing, based on a multitrack acquisition architecture which enables several spectra to be measured per shot. Besides being usable with standard sparse recovery techniques, multitrack compressed sensing also supports adaptive basis scan, so that multiple basis coefficient can be sampled in parallel. As a result, the measurement speed can be drastically increased, while maintaining reconstruction speed as well as accuracy.

The paper is organized as follows. Section 2 reviews the prior work on compressed sensing for HSI. Section 3 describes the proposed multitrack acquisition architecture, and develops the two compatible CS algorithms for it: a non-adaptive sparse-recovery-based algorithm, and an adaptive basis scan algorithm. Section 4 describes the numerical experiments conducted, and Section 5 presents the reconstruction results for noiseless as well as noisy measurements. Finally, Section 6 concludes.

2 PRIOR WORK ON CS FOR HSI

2.1 CS formulation and notation

Here, we describe the general framework for compressed sensing\textsuperscript{[9], [10]}.

Let \( x \in \mathbb{R}^n \) (represented by a column vector) be a discrete signal that is to be measured. Measurement (alternatively sampling or acquisition) of \( x \) is described by the equation:

\[
y = \Phi x
\]

where \( y \in \mathbb{R}^m \) is a column vector whose components are all the measurement values, and \( \Phi \in \mathbb{R}^{m \times n} \) is the measurement matrix. Here, an individual measurement \( y_j \) can be written as a dot product (or projection) between the signal and the \( j \)-th row \( \varphi_j^T \) of \( \Phi \):

\[
y_j = \varphi_j^T x, \quad \varphi_j \in \mathbb{R}^n, \quad 1 \leq j \leq m
\]

Subsampling occurs when the number of measurements \( m \) is less than the length of the signal \( n \). Hence we can define the sampling ratio \( r_{\text{sampling}} = m/n \), which is less than 1 for subsampling.

Post the subsampling process, the original signal \( x \) needs to be reconstructed from the measurements \( y \). This is, in general, not possible via a direct inversion of the measurement process (1), because the problem is underdetermined when \( m < n \). Hence, prior knowledge about the nature of the signal is required.

A standard assumption on the structure of the signal is sparsity or compressibility in some basis. Natural signals such as images often have a sparse representation:

\[
x = \Psi s
\]

where \( \Psi \in \mathbb{R}^{n \times n} \) is a matrix whose columns form a basis for \( \mathbb{R}^n \), and where only a few components of the representation
s ∈ ℜⁿ are significantly different from zero. To reconstruct x, the problem is first re-written by combining (3) and (1):

\[ y = \Phi \Psi s = \Theta s \]  

(4)

where Θ = ΦΨ ∈ ℜᵐ×ⁿ is sometimes called the sensing matrix. Provided that Θ satisfies certain conditions (e.g. the restricted isometry property \([29]\)), s can be reconstructed to a high degree of accuracy (via methods such as sparse recovery). Random matrices, in particular, generated by various kinds of distributions satisfy the restricted isometry property with high probability.

Sparse recovery aims to recover s by searching for sparse solutions to (4). A classical sparse recovery technique is Basis Pursuit, which involves solving the following convex problem:

\[ \hat{s} = \arg \min_s \|s\|_1 \text{ subject to } y = \Theta s \]  

(5)

where \(\|\cdot\|_1\) denotes the \(\ell_1\)-norm. When the signal is an image (regular or hyperspectral), total variation (TV) minimization is a particularly effective sparse recovery technique. In TV methods, the \(\ell_1\)-norm in (5) is replaced by certain TV-norms, which are norms of the discrete gradient of a signal.

The particular type of signals addressed in this paper is hyperspectral scenes (and images), which can be naturally organized into datacubes. Mathematically, this is a 3D array of shape \(n_h × n_w × n_c\), where \(n_h\) is the spatial height of the scene or image in pixels, \(n_w\) is the spatial width in pixels, and \(n_c\) is the number of wavelength channels. \(n_p \equiv n_h × n_w\) denotes the total number of spatial pixels. It is common, and convenient for use in later equations however, to reshape the elements of the datacube into a matrix: \(X ∈ ℜ^{(n_hn_w) × n_c}\), whose row indices correspond to spatial pixel locations and column indices correspond to wavelength channels. This is done by taking spatial slices of the datacube at each wavelength channel, and unrolling these slices into tall column vectors which end up forming the columns of \(X\).

Note that the symbol \(X\) here plays the same role as the symbol \(x\) in Section 2.1 – the only difference is that the elements of \(X\) are arranged into a matrix instead of a column vector. This notation will be used in the following sections of the paper.

At a glance, compressed sensing HSI involves (1) making valid assumptions about the structure of \(X\) (e.g. sparsity), (2) subsampling \(X\) via a measurement operator, and (3) using reconstruction algorithms (e.g. sparse recovery) to obtain the reconstructed image \(\hat{X}\).

2.2 Acquisition architectures

Besides the CS algorithms, the imaging architecture used also plays a major role in determining the effectiveness of a compressed sensing HSI (CS-HSI) system. Several such architectures that have been employed \([11]\) make use of devices that can modulate and code light spatially. Spatial light modulators include masks \([30]\), which block out light in some regions and not in others, and digital micromirror devices (DMD) \([31]\) which are arrays of micrometer-scale mirrors that can dynamically form patterns that selectively redirect light from different regions.

Coded aperture snapshot spectral imaging (CASSI) \([12]\–[14]\) provides a prominent class of imaging architectures for CS-HSI, that can achieve snapshot HSI. Light from the scene is spectrally dispersed, and spatially coded by a mask or DMD. The transformed light is then collected onto a 2D detector array as a snapshot, forming the set of measurements to be fed into the reconstruction algorithm.

The snapshot nature of CASSI even enables the capture of hyperspectral video at high frame rates \([32]\). But such powerful architectures can come with increased hardware costs – to image an \(n_h(\text{height})\times n_w(\text{width})\times n_c(\text{wavelength channels})\) voxel scene using single dispersive CASSI for example, the detector array needs to have dimensions of at least \((n_h) × (n_w + n_c - 1)\) pixels. The number \(n_w + n_c - 1\) can generally be very large, and this additionally presents a trade-off between the amount of spatial and spectral information that can be captured at once. Dual dispersive CASSI can function with a detector array having fewer pixels, but there is still an issue that significantly low amounts of spectral information can be captured in one shot. Multiframe CASSI \([33]\) acquires more information through multiple shots, each using a distinct mask or DMD pattern, albeit the snapshot nature of the method is inevitably lost. Hence, there exists a different trade-off: one between spectral information and acquisition time. Furthermore, CASSI systems use prisms as the dispersive element and since the dispersion of a prism is nonlinear, the spectral resolution of CASSI varies with wavelength \([34]\).

High-accuracy and high-resolution spectral data warrant the use of a spectrometer. The single pixel camera (SPC)-style hyperspectral imager \([15]\) is an acquisition architecture that leverages this – it consists of an SPC with the single pixel detector replaced by a spectrometer. Light captured from the scene is modulated by a pattern formed on a DMD, after which it is focused and coupled into a spectrometer, with a spectrum collected. This sequence is repeated multiple times, with different DMD patterns giving rise to different collected spectra, which are then used to reconstruct the full hyperspectral image using CS methods.

As mentioned before, this architecture benefits from the high spectral accuracy and resolution that the spectrometer provides. Also, the design is modular – a single pixel camera module followed by a spectrometer module – so that it is easy to replace or tweak one module without affecting the other.

Being an extension of the single pixel camera, this architecture allows us to borrow many of the CS methods developed originally for single pixel imaging, including adaptive basis scan. The capability of the architecture to sample dot products of patterns on the DMD with the spatial slices of the datacube allows for direct sampling of the wavelet coefficients of the scene.

Note, however, that unlike snapshot CASSI, the SPC-style imager requires sequential measurements of spectra. This is a serious limitation of the architecture, especially because the information contained in a single spectrum is generally very low, and a large number of shots need to be taken in order to collect enough spectra to achieve a decent reconstruction accuracy. Hence, even though the faster adaptive reconstruction techniques are enabled, the slow measurement process nullifies this advantage. Addressing
2.3 Adaptive basis scan algorithms

Adaptive basis scan was briefly described in Section 1.2. It is based on progressive prediction and sampling of the most significant coefficients of a signal in some basis. The most popular bases for this method are wavelets [35]. Wavelet transforms decompose signals into coefficients that capture the signals' features at multiple scales and locations. Such multiresolution analyses provide the transform coefficients a hierarchical structure, so that some coefficients have predictive power over others. In particular, low resolution information that is contained in the wavelet coefficients at coarser scales can be used to predict which of the wavelet coefficients at finer scales are significant.

Deutsch et al. [23] predicted the significance of the fine-scale coefficients by estimating local directional Lipschitz exponents from the already-sampled coarse-scale coefficients. Averbuch et al. [24], on the other hand, relied on statistical modeling of the wavelet coefficients and their parent-child relationships. Dai et al. [25] made use of both parent-children relationships and sibling relationships in extended wavelet trees. In [36], Hahn et al. adapted Deutsch's adaptive algorithm to hyperspectral imaging. Finally, Rouset et al. [26] developed an adaptive basis scan algorithm whose wavelet coefficient prediction step relies on bicubic interpolation in image space. This was further extended to time-resolved HSI in [37]. In [36], [37] however, the measurement step is not multitrack – the sensor elements measure only a single spectrum per shot.

The multitrack adaptive CS algorithm we present in Section 3.3 is an extension of the Adaptive Basis Scan with Wavelet Prediction (ABS-WP) algorithm from [26], [37]. We make this choice because ABS-WP produces high accuracy reconstruction, even in the presence of noise. Also, ABS-WP has already been demonstrated to work with HSI via minor changes to the original algorithm. However, multitrack CS is potentially compatible with the other adaptive basis scan algorithms as well.

3 Proposed Multitrack Compressed Sensing

The single-track acquisition of the SPC-style hyperspectral imager measures only one spectrum per shot, and hence, a large number of shots are needed to obtain sufficient information for a good reconstruction of the hyperspectral scene. This is especially a problem when integration times are large, as is the case for low-light imaging. To develop more powerful acquisition architectures, one can draw inspiration from pushbroom hyperspectral imagers in which multiple spectra are measured at once.

3.1 Multitrack acquisition architecture

Consider the hyperspectral scene \( X \in \mathbb{R}^{n_p \times n_r} \) whose components have units of photons per second. Different spatial regions – rectangular tracks (Fig.1(a)) – of the hyperspectral scene are to be treated separately, yet in parallel. Let \( n_r \) be the number of tracks. Then each track of the scene will be \( n_h \)-by-\((n_w/n_r)\)-by-\(n_c \) in terms of voxels. Spatially, each track has \( n_h \cdot (n_w/n_r) = n_p/n_r \) pixels. In our matrix notation of \( X \), this division into tracks can be described by breaking down \( X \) into submatrices:

\[
X = \begin{bmatrix}
X^{(1)} \\
X^{(2)} \\
\vdots \\
X^{(n_r)}
\end{bmatrix}
\]  

where \( X^{(1)}, \ldots, X^{(n_r)} \in \mathbb{R}^{(n_p/n_r) \times n_c} \) represent the tracks. Next, the different tracks of \( X \) need to be modulated separately, but in parallel. This is achieved by dividing the DMD into tracks of size \( n_h \)-by-\((n_w/n_r) \) as well, to correspond spatially to the tracks of the hyperspectral scene (Fig.1(b)). The pixels on the \( i \)-th track of the DMD form the \( i \)-th subpattern \( \zeta^{(i)} \in \mathbb{R}^{(n_p/n_r)} \). This modulates the \( i \)-th track of \( X \) i.e. \( X^{(i)} \).

All tracks of \( X \) simultaneously undergo such modulation, after which the entire modulated scene is focused into a line by a cylindrical lens (Fig.1(c)). Different sections of the line comprise of information from different tracks. This can be modeled by a trackwise list of dot products between
subpatterns and the corresponding tracks of \(X\):  
\[
\begin{bmatrix}
    (\zeta^{(1)})^T X^{(1)} \\
    (\zeta^{(2)})^T X^{(2)} \\
    \vdots \\
    (\zeta^{(n_r)})^T X^{(n_r)}
\end{bmatrix} \in \mathbb{R}^{n_r \times n_c}
\]  
(7)

where each \((\zeta^{(i)})^T X^{(i)} \in \mathbb{R}^{1 \times n_c}\). Finally, the transformed scene enters a spectrometer where the focused line is dispersed. The 2D detector array captures one spectrum from each track i.e. \(n_r\) spectra in total. With integration time \(\Delta t\), the recorded spectra in a given shot (in the units of photon counts) are described by the rows of:  
\[
\begin{bmatrix}
    (\zeta^{(1)})^T X^{(1)} \Delta t \\
    (\zeta^{(2)})^T X^{(2)} \Delta t \\
    \vdots \\
    (\zeta^{(n_r)})^T X^{(n_r)} \Delta t
\end{bmatrix} \in \mathbb{R}^{n_r \times n_c}
\]  
(8)

assuming no measurement noise. Overall, one shot of the multitrack acquisition process comprises of the simultaneous (or parallel) action of subpatterns \(\zeta^{(1)}, \ldots, \zeta^{(n_r)}\) to linearly map \(X\) onto the recorded spectra \((\zeta^{(i)})^T X^{(i)} \Delta t\).

### 3.2 Multitrack non-adaptive compressed sensing

This section describes the first of our proposed methods. It is based upon block compressed sensing, and uses non-adaptive sparse recovery for reconstruction.

Block compressed sensing \cite{Lustig2007, Candés2006} involves dividing a scene into blocks and applying the measurement matrix to those blocks separately. The reconstruction can then be carried out either separately for each block, or globally by processing all the blocks together. The aim of block CS is to reduce the reconstruction complexity and the memory burden of the imaging process.

The track-wise acquisition naturally allows for the use of block compressed sensing methods, where the tracks can be treated as blocks that have their own measurement matrices. The reconstruction algorithm then takes into account this measurement matrix structure, as will be described later.

Note that a non-adaptive compressed sensing HSI architecture with a similar multitrack acquisition principle was first developed, to our knowledge, in \cite{Romano2019}. Per shot, a spectrum is captured from each pixel-wide column of the scene; this is the same as having one-pixel-wide tracks in our formulation from Section 3.1. The use of block compressed sensing provides a generalization to this, allowing the tracks to have a width greater than one pixel. This is important when the number of spectra that can be measured per shot is limited by the detector pixel count. Several spectrometers have 2D detector arrays of 256-by-1024 pixels, for example, where a 1-by-1024 strip is meant for a single spectrum. In that case, the maximum number of spectra that can be measured in a shot is 265. Moreover, due to optical aberrations near the edges of the detector array, the maximum number of spectra might be even lower in practice. This is a limiting factor when the desired datacube is more than 256 pixels wide. Our use of block CS, however, enables the use of wider tracks. As a result, fewer tracks can be used to cover the whole datacube, and fewer spectra need to be measured per shot.

#### 3.2.1 Measurement process

As separate tracks are measured separately, we can rewrite Eq. 1 for each track:

\[
Y^{(i)} = \Phi^{(i)} X^{(i)} \text{ for each track } i
\]  
(9)

where \(\Phi^{(i)} \in \mathbb{R}^{m^{(i)} \times (n_p/n_r)}\) is track \(i\)'s measurement matrix, and \(m^{(i)}\) is the number of samples taken from track \(i\), which is related to the sampling ratio by \(m^{(i)} = \tau_{\text{sampling}} \times n_p/n_r\). \(Y^{(i)} \in \mathbb{R}^{m^{(i)} \times n_c}\) are the measured spectra from the track (up to some scaling factors).

The measurement matrices \(\Phi^{(i)}\) we use are composed of entries drawn from a transformed Bernoulli distribution: \(\pm \sqrt{m^{(i)}}\) with equal probabilities. Here \(\sqrt{m^{(i)}}\) works as a normalization.

Now, we describe how the measurement matrices can be implemented using a DMD. Consider track \(i\). The \(k\)-th projection from this track can be written as the dot product

\[
Y^{(i)}_{k,:} = (\varphi_k^{(i)})^T X^{(i)}
\]  
(10)

where \(Y^{(i)}_{k,:}\) denotes the \(k\)-th row of \(Y^{(i)}\) (equivalent MATLAB-like notation would be \(Y^{(i)}(k,:)\), and \((\varphi_k^{(i)})^T\) is the \(k\)-th row of \(\Phi^{(i)}\). To implement this projection, we employ the following scaling and shifting operations on \(\varphi_k^{(i)}\), and obtain physically implementable \(b\)-bit (grayscale) subpatterns:

\[
\zeta^{(i)}_k = \frac{2^b - 1}{2^b - 1} \left( \frac{\varphi_k^{(i)}}{\|\varphi_k^{(i)}\|_\infty} + 1^{(i)}_{n_p} \right) \text{ for all tracks } i
\]  
(11)

where \(1^{(i)}_{n_p}\) is a subpattern \([1, \ldots, 1]^T \in \mathbb{R}^{(n_p/n_r)}\) of the all-ones pattern \(1_{n_p} \in \mathbb{R}^{n_p}\):

\[
1_{n_p} = \begin{bmatrix} 1^{(1)}_{n_p} \\ \vdots \\ 1^{(n_r)}_{n_p} \end{bmatrix}
\]

An additional offset subpattern is also required to compensate for the scaling and shifting mentioned above:

\[
\zeta^{(i)}_{\text{offset}} = \frac{2^b - 1}{2^b - 1} \cdot 1^{(i)}_{n_p} \text{ for all tracks } i
\]  
(12)

These operations are needed because \(\varphi_k^{(i)}\) cannot be physically implemented in a direct manner. As the DMD pixels values are physically restricted to the interval \([0, 1]\), the \(\pm 1/\sqrt{m^{(i)}}\) values of \(\varphi_k^{(i)}\) cannot be realized on the DMD. Furthermore, \(b\)-bit DMD pixel values are subject to the discretization \(\left\{ \frac{0}{2^b-1}, \frac{1}{2^b-1}, \ldots, \frac{2^b-1}{2^b-1} \right\}\) which needs to be accounted for while constructing the subpatterns.
pressed sensing is to reformulate Eq. (9) as
\[
G(k) = \begin{bmatrix}
(\zeta_k^{(1)})^T X^{(1)} \Delta t \\
\vdots \\
(\zeta_k^{(n_r)})^T X^{(n_r)} \Delta t 
\end{bmatrix}
\]
(13)

with integration time \( \Delta t \). From the offset subpatterns, we obtain
\[
G_{\text{offset}} = \begin{bmatrix}
(\zeta_{\text{offset}}^{(1)})^T X^{(1)} \Delta t \\
\vdots \\
(\zeta_{\text{offset}}^{(n_r)})^T X^{(n_r)} \Delta t 
\end{bmatrix}
\]
(14)

The raw measurement can be processed as follows, based on Eq. (10)–Eq. (14) to obtain the elements of \( Y \):
\[
Y_{\alpha,\ell}^{(i)} = \frac{2^{b_i} - 1}{2^{b_i} - 1} \| \phi_k^{(i)} \|_\infty \| (G(k))_{i,\ell} - [G_{\text{offset}}]_{i,\ell} \|, \quad \text{for all } \alpha, \ell
\]
(15)

where \( (G(k))_{i,\ell} \) and \( [G_{\text{offset}}]_{i,\ell} \) denote \((i, \ell)\)-th entries of \( G(k) \) and \( G_{\text{offset}} \) by respectively. Note that \( G_{\text{offset}} \) does not depend on the projection index \( k \), and thus it only needs to be measured once. The measured value can subsequently be reused for other values of \( k \). However, in the case of noisy measurements, it would be preferable to measure \( G_{\text{offset}} \) a few more times, with an average taken in the end.

### 3.2.2 Reconstruction algorithm based on block compressed sensing

A direct way to approach reconstruction would be to use sparse recovery on each track separately. But this leads to blocking artifacts. An alternative solution for block compressed sensing is to reformulate Eq. (9) as
\[
\begin{bmatrix}
Y^{(1)} \\
\vdots \\
Y^{(n_r)}
\end{bmatrix} = \begin{bmatrix}
\Phi^{(1)} \\
\vdots \\
\Phi^{(n_r)}
\end{bmatrix} \begin{bmatrix}
X^{(1)} \\
\vdots \\
X^{(n_r)}
\end{bmatrix}
\]
(16)

Sparse recovery can now be used to solve for all of \( X \) from the measurements \( Y \). Though \( X \) can now be reconstructed globally, it is often simpler and more efficient to reconstruct spatial slices \( X_{\alpha,\ell} \) for the different wavelength channels \( \ell \in \{1, \ldots, n_r\} \) separately. Note that \( X_{\cdot,\ell} \) denotes the \( \ell \)th column of \( X \) (equivalent MATLAB-like notation would be \( X(:, \ell) \)). This yields
\[
Y_{\alpha,\ell} = \Phi X_{\alpha,\ell} \quad \text{for each channel } \ell
\]
(17)

where \( Y_{\alpha,\ell} \), the \( \ell \)th column of \( Y \), are the measurements at the \( \ell \)th wavelength channel. The reconstruction algorithm we use is 2D TV-minimization (used on each slice separately):
\[
\tilde{X}_{\alpha,\ell} = \arg \min_{X_{\alpha,\ell}} \text{TV}_2(X_{\alpha,\ell}) \quad \text{subject to } Y_{\alpha,\ell} = \Phi X_{\alpha,\ell}
\]
(18)

### 3.3 Multitrack adaptive compressed sensing

This section describes the second of our proposed methods. It involves implementing an adaptive basis scan algorithm where multiple basis coefficients are sampled in parallel. Similar to most adaptive basis scans, we use a 2D wavelet basis – specifically, 2D Haar wavelets \([35]\). Besides their simplicity, Haar wavelets have a property that is critical to our parallelized sampling: when viewed as a \( n_h \times n_w \) pixel 2D pattern, the support (i.e. region of non-zero pixels) of a Haar wavelet is localized, and can be completely contained inside a track or subpattern. This allows for sampling a wavelet coefficient using only a subpattern instead of a full \( n_h \times n_w \) pattern. The other subpatterns are free to parallelly sample other wavelet coefficients. Details of the process are described below.

#### 3.3.1 Measurement of wavelet coefficients

This problem consists of designing compound patterns made up of the subpatterns, such that each of the recorded spectra provide information on the wavelet coefficients.

Consider a 2D Haar wavelet \( \psi_q \in \mathbb{R}^{n_p} \) labeled by an index \( q \). This, when viewed as a a \( n_h \times n_w \) 2D wavelet pattern, can also be divided into \( n_r \) tracks of size \( n_h \times \lfloor n_w/n_r \rfloor \), where the division into tracks can again be described by breaking down \( \psi_q \) into submatrices:
\[
\psi_q = \begin{bmatrix}
\psi^{(1)}_q \\
\psi^{(2)}_q \\
\vdots \\
\psi^{(n_r)}_q
\end{bmatrix}
\]
(19)

Here, \( \psi^{(i)}_q \in \mathbb{R}^{n_p/n_r} \) is the \( i \)th track of the wavelet. Now, if \( \psi_q \) is a Haar wavelet pattern whose support lies entirely in the \( i \)th track, then its pixel values must be zero in all other tracks. This would mean that for any track \( i' \) other that \( i \), we have \( \psi^{(i')}_q = 0 \), and \( \psi_q \) reduces to:
\[
\psi_q = \begin{bmatrix}
0 \\
\vdots \\
\psi^{(i)}_q \\
\vdots \\
0
\end{bmatrix}
\]
(20)

Hence, the sub-matrix (or subpattern) \( \psi^{(i)}_q \) can play the role of the full wavelet \( \psi_q \), despite being smaller in size. Importantly, it can still sample dot products:
\[
\psi_q^T X = \begin{bmatrix}
X^{(1)} \\
\vdots \\
X^{(i)} \\
\vdots \\
X^{(n_r)}
\end{bmatrix} \begin{bmatrix}
\psi^{(1)}_q \\
\vdots \\
\psi^{(i)}_q \\
\vdots \\
\psi^{(n_r)}_q
\end{bmatrix} = \begin{bmatrix}
\psi^{(i)}_q X^{(i)} \\
\vdots \\
\psi^{(i)}_q X^{(i)}
\end{bmatrix}
\]
(21)

Note that the left-hand side is just the wavelet coefficients of the slices of \( X \): \( \psi_q^T X = [\psi^{(1)}_q X_{\cdot,1}, \ldots, \psi^{(n_r)}_q X_{\cdot,n_r}] = \)
Fig. 2. Flow chart of the multitrack adaptive CS scheme for hyperspectral imaging. Steps 1-6 are described briefly here, and explained further in the main text.

Thus, wavelet subpatterns can be used instead of full wavelets to obtain the wavelet coefficients. This, together with the aforementioned facts that (1) several DMD subpatterns $\zeta^{(i)}$ can be formed parallelly on the DMD, and (2) the corresponding spectra can be measured, enables the extraction of multiple wavelet coefficients per slice in a shot.

The precise steps involved in this process are as follows. Let $W_{q_1,\ell}, W_{q_2,\ell}, \ldots, W_{q_n,\ell}$, for all channels $\ell \in \{1, \ldots, n_c\}$, be wavelet coefficients that are to be extracted simultaneously in a shot. To enable this, we require that the supports of the corresponding patterns $\psi_{q_1}, \psi_{q_2}, \ldots, \psi_{q_n}$ lie in separate tracks. Without loss of generality, suppose the support of $\psi_{q_i}$ lies in the $i$-th track, for each $i \in \{1, \ldots, n_T\}$. Then, from (21)

$$[W_{q_1,1}, \ldots, W_{q, n_c}] = \psi_{q_i}^T X = \begin{pmatrix} \psi_{q_1}^{(i)} \\ \psi_{q_2}^{(i)} \\ \vdots \\ \psi_{q_n}^{(i)} \end{pmatrix}^T X^{(i)}$$

Using this equation, we can obtain a formula similar to Eq. (15) which extracts the wavelet coefficients from the measured raw spectra. Analogous to Section 3.2, we employ the following scaling and shifting operations on the wavelet subpatterns to account for the non-negative and discretized range of the DMD pixel values:

$$\zeta^{(i)} = \frac{2^{b-1} - 1}{2^b - 1} \left( \frac{\psi_{q_i}}{\| \psi_{q_i} \|_\infty} + 1^{(i)}_{n_p} \right)$$

for all tracks $i$ (23)

where $b$ is the number of bits again, and $1^{(i)}_{n_p}$ is a subpattern $[1, \ldots, 1]^T \in \mathbb{R}^{(n_p)/n_T}$ of the all-ones pattern $1_{n_p}$. Again, an additional set of offset subpatterns is required to account for the scaling and shifting mentioned above:

$$\zeta_{\text{offset}}^{(i)} = \frac{2^{b-1} - 1}{2^b - 1} \cdot 1^{(i)}_{n_p}$$

for all tracks $i$ (24)
Now, the final subpatterns $c_{q_j}^{(i)}$, for $i \in \{1, \ldots, n_r\}$, can be formed parallelly as a compound pattern on the DMD (Fig. 1b). The pixels on the $i$-th track of the DMD form the $i$-th subpattern. With integration time $\Delta t$, the recorded spectra (see Eq. (8)) are:

$$G(q_1, \ldots, q_{n_r}) = \begin{bmatrix} \left(c_{q_1}^{(1)}\right)^T X^{(1)} \Delta t \\ \vdots \\ \left(c_{q_{n_r}}^{(n_r)}\right)^T X^{(n_r)} \Delta t \end{bmatrix}$$

(25)

A set of measurements from the offset subpatterns is also needed:

$$G_{\text{offset}} = \begin{bmatrix} \left(c_{\text{offset}}^{(1)}\right)^T X^{(1)} \Delta t \\ \vdots \\ \left(c_{\text{offset}}^{(n_r)}\right)^T X^{(n_r)} \Delta t \end{bmatrix}$$

(26)

Denoting the $(i, \ell)$-th entries of $G(q_1, \ldots, q_{n_r})$ and $G_{\text{offset}}$ by $[G(q_1, \ldots, q_{n_r})]_{i,\ell}$ and $[G_{\text{offset}}]_{i,\ell}$ respectively, we finally obtain as a result of Eq. (22):

$$W_{q_i,\ell} = \frac{2^b - 1}{2^{b-1} - 1} \frac{\|\psi_{q_i}\|_{\infty}}{\Delta t} \left([G(q_1, \ldots, q_{n_r})]_{i,\ell} - [G_{\text{offset}}]_{i,\ell}\right),$$

for all $i, \ell$ (27)

3.3.2 Extension of the adaptive basis scan algorithm to incorporate multitrack measurements

Steps 1-4 in Fig. 2 are closely based on Rousset et al. [26], [37] and will be described only briefly here. Steps 5 and 6 are novel, and designed to incorporate the multitrack requirement into the adaptive algorithm.

The setup is similar to [26], [37]. The decomposition level of the wavelet transform is set to some $J \in \mathbb{N}$, which determines the coarsest scale at which information from the scene is captured. However, the need for multitrack sampling imposes a restriction on $J$ due to the requirement that the support of each wavelet pattern must be situated completely within the boundaries of a track. Since the support a Haar wavelet under decomposition level $J$ can be up to $2^J$-by-$2^J$ pixels in size, we need $2^J$ to be less than the width of a track in pixels: $2^J \leq n_w/n_r$.

Next, the sampling percentages are set for each scale $p_j$, $\ldots$, $p_1$, with $j = J$ being the coarsest, and $j = 1$ being the finest.

The algorithm is divided into $J + 1$ acquisition cycles where wavelet coefficients are measured, and $J$ prediction cycles where the significance of yet-to-be-measured wavelet coefficients is predicted using the already-measured coefficients. The resulting predicted wavelet coefficients then guide the next acquisition cycle and so on.

The algorithm starts off with an acquisition cycle (Step 1 in Fig. 2) where all the approximation wavelet coefficients of all slices $X_{s,t}$ of the scene are measured. Unlike [26], [37], these are now measured parallelly with the multitrack system, using the corresponding approximation compound patterns made up of subpatterns that extract approximation coefficients. The first prediction cycle (described in the next paragraph) predicts the detail wavelet coefficients at scale $j = J$, and this is followed by an acquisition cycle that measures detail coefficients at scale $j = J$. The scale is then lowered by one, $j \rightarrow j - 1$, and the cycles repeat.

In a prediction cycle for detail coefficients at scale $j \in \{1, \ldots, J\}$, the steps are as follows. First a low resolution intermediate reconstruction is obtained by inverse transforming, for every slice, all the wavelet coefficients that were measured or set to zero in all the previous acquisition cycles (Step 2 in Fig. 2). Binning of all slices at the various wavelength channels gives a grayscale image, that is then oversampled by a factor of 4 in number of pixels via a bicubic interpolation. (Step 3 in Fig. 2). The oversampled image is then one-level 2D wavelet transformed to give predicted detail wavelet coefficients at scale $j$ (Step 4 in Fig. 2). Until now, all the prediction steps have been similar to the ABS-WP algorithm from [26], [37]. The next two steps contain the differences that are needed to make ABS-WP compatible with multitrack measurement (see also Steps 5 and 6 in Fig. 2).

Step 5: Predicted wavelet coefficients are queued according to their magnitudes. The higher the magnitude of a predicted wavelet coefficient, the earlier in the queue it appears as it is expected to be more significant. Each predicted wavelet coefficient will correspond to a unique wavelet, creating an ordered queue of wavelets. Viewing the wavelets as 2D patterns, the support of each wavelet pattern is completely situated within the boundaries of a track, and the corresponding track that the support sits within can be referred to as the pattern’s supporting track. The queued wavelet patterns are further sorted into sub-queues based on their supporting tracks, while maintaining their ordering within each sub-queue. Based on $p_j$, a corresponding ordering of the more significant wavelet patterns are retained from each of the sub-queues, while the less significant wavelet patterns (at the back of the queues) are discarded.

This sub-queueing process can result in some imaging accuracy losses however. In the single-track ABS-WP, the patterns to be used correspond to the $p_j$ most significant predicted coefficients overall. This is different in general, however, from the $p_j$ most significant coefficients from each sub-queue, because some sub-queues might have more of the significant coefficients than other sub-queues.

Step 6: One wavelet pattern from each sub-queue is combined to form a compound pattern: if the wavelet patterns $\psi_{q_1}, \psi_{q_2}, \ldots, \psi_{q_{n_r}}$ appear first in sub-queues $1, 2, \ldots, n_r$ respectively (so that the supporting track of $\psi_{q_i}$ is the $i$-th track), then the first compound pattern is formed by corresponding subpatterns $c_{q_i}^{(i)}$, for $i \in \{1, \ldots, n_r\}$, obtained from Eq. (23). This is continued sequentially, with the second wavelet pattern from each sub-queue combined into the second compound pattern and so on.

This prediction cycle at scale $j$ is followed by an acquisition cycle, as mentioned before, for the measurement of detail coefficients at scale $j$ for all slices $X_{s,t}$ of the scene. The compound patterns from Step 6 are used to parallelly measure the detail coefficients that were predicted to be significant in the preceding prediction cycle. The rest of the coefficients at scale $j$ are set to zero. Finally, $j$ is lowered by one and the cycles repeat. The algorithm terminates when
The final reconstruction is obtained by an inverse wavelet transform, for each slice, of all the wavelet coefficients that were measured (or set to zero) from that slice of the scene in all the acquisition cycles. This is identical to the reconstruction step in \cite{27}.

\section{Numerical experiments}

The performance and applicability of multitrack CS for HSI is assessed computationally on hyperspectral images. The measurements are simulated, and fed into the adaptive sampling or reconstruction algorithms being evaluated.

First, noise-less measurement simulations and reconstruction are performed on the three different hyperspectral scenes: \textit{Colour Checker}, \textit{Green Peppers}, and \textit{U2OS cell}, obtained from \cite{41} and \cite{42}.

Both multitrack non-adaptive CS (Section 3.2) and multitrack adaptive CS (Section 3.3) are compared against benchmarks:

\textbf{Multitrack full sampling:} This involves sampling the whole datacube using a scheme similar to pushbroom. But for a fair comparison with the multitrack CS methods, it is equipped with the same acquisition architecture as the multitrack methods – the same number of detector tracks, in particular. All spectra are measured from each track of the scene, in a ‘multitrack point-scan’ fashion. Hence, no reconstruction step is needed.

\textbf{Single-track adaptive CS:} This involves the sampling of a single spectrum per shot by a SPC-style hyperspectral imager \cite{15}, using the adaptive ABS-WP algorithm of \cite{26}. Note that we use an adaptive algorithm for benchmarking here because it generally outperforms single-track non-adaptive CS in terms of speed as well as accuracy.

Details of the compared methods are summarized in Table \ref{tab:methods}.

For multitrack non-adaptive CS, the sparse recovery algorithm used is total variation (TV) minimization with equality constraint, implemented using the TVAL3 package \cite{43}. The settings used are ‘anisotropic TV’ with ‘positivity’, since these tend to give the best results. For multitrack adaptive CS as well as single-track adaptive CS, the decomposition level used is \( J = 4 \) for \textit{Colour Checker} and \textit{Green Peppers}, and \( J = 3 \) for \textit{U2OS cell}. Higher values of \( J \) enable the use of smaller sampling ratios, and are generally preferable. But since the \textit{U2OS} cell scene is of a smaller pixel count, our use of multitrack adaptive CS requires \( J \) to be less than 4 to ensure that the supports of the wavelet patterns lie completely inside individual tracks.

The results are reported in Section 5.1. The performance of the compared methods is quantified by the following metrics:

\textbf{Reconstruction accuracy:} This is quantified by relative error and peak signal to noise ratio (PSNR).

\textbf{Acquisition time:} This is composed of (i) the time required to adaptively determine the DMD patterns (0 for non-adaptive), (ii) the time required to load the DMD patterns into a computer’s memory, (iii) the time required to switch the DMD patterns, as per the pattern switch rate, and (iv) the total spectrometer integration time, which is the total number of shots times \( \Delta t \).

\textbf{Reconstruction time:} This is the time required to compute the final reconstruction.

\textbf{Total time:} Acquisition time + Reconstruction time.

In Section 5.2, we report the simulated effects of illumination level and noise on imaging accuracy, and assess the robustness of the methods developed here. We model measurement noise as Poissonian shot noise together with dark noise. Given a subpattern \( \zeta^{(i)} \) for track \( i \), and integration time \( \Delta t \), the recorded noisy spectrum from the track is modeled as

\begin{equation}
\text{Pois} \left( \left( \frac{\zeta^{(i)}}{T} \right)^T X^{(i)} \Delta t + \alpha \Delta t \right)
\end{equation}

where \( \alpha \in \mathbb{R}_+ \) is the dark current in the detector pixels in units of photons pixel\(^{-1}\) s\(^{-1}\), and \text{Pois}(\mu) stands for a Poisson random variable with mean \( \mu \).

In low-light imaging, where the scene intensity is low, both shot noise and dark noise become more prominent.

\textbf{Shot noise:} The standard deviation of a \text{Pois}(\mu) random variable is \( \sqrt{\mu} \); as a result, the ratio (coefficient of variation) \( \frac{\Delta t}{\mu} \) grows larger as \( \mu \) shrinks. The implication is that when the intensity of the scene \( X \) is small, the standard deviation of the measured signal is large with respect to its mean.

\textbf{Dark noise:} The \( \alpha \Delta t \) term can overpower the \( \left( \frac{\zeta^{(i)}}{T} X^{(i)} \Delta t \right) \) term when the scene intensity is small.

Finally, note that the simulations are conducted using MATLAB, on a HP Z240 Tower Workstation with 4 cores of Intel® Core™ i5-6500 CPU @ 3.20GHz, and 16 GB of RAM.
### Results and Discussion

#### 5.1 Simulation results on various hyperspectral images

The simulation results for the Colour Checker and U2OS cell images, under noiseless measurement, are presented in Fig. 3. We evaluate the two proposed multitrack methods against multitrack full sampling and single-track adaptive CS. First, we observe that single-track acquisition is slow: its acquisition time (as well as total time) is about twice that of multitrack full sampling. Since only one spectrum is measured per shot, a large number of shots are required even with a low sampling ratio of about 6.2%. The other three methods, on the other hand, need fewer shots due to their multitrack acquisition architecture.

Multitrack non-adaptive CS, has the smallest acquisition time – about 6-7% of the acquisition time of full sampling. This is because of the employed sampling ratios of about 6.2%. For the 512-by-512-by-31 voxel Colour Checker scene, full sampling needs 8192 shots, whereas sampling at about 6.2% amounts to 516 shots. Note that the 516 shots here include 10 extra shots corresponding to the offset patterns mentioned in Section 3. A similar comparison applies to the U2OS cell scene which is 256-by-256-by-231 voxels. A low sampling ratio, together with multitrack acquisition, reduces acquisition time significantly. Also, no adaptive prediction steps must be carried out during the acquisition process, unlike for multitrack adaptive CS. Hence, the pattern sequence does not need to be computed on the fly, and can be pre-programmed into the DMD’s field-programmable gate array (FPGA), for increased speed. However, due to large reconstruction times of 947 s for the Colour Checker scene and 478 s for the U2OS cell scene, the total time for multitrack non-adaptive CS is even larger than that for full sampling. Thus, despite a quick acquisition step, the measurements need to be post-processed in an offline computation.

In terms of total time, multitrack adaptive CS performs the best. It is faster than multitrack non-adaptive CS due to a much smaller reconstruction time, even though the acquisition time is up to 8 s higher due to the adaptive steps. Multitrack adaptive CS is also faster than full sampling, again because of the small sampling ratio which requires fewer shots.

For the multitrack methods, an increase in the number of tracks can also increase the imaging speed due to a higher extent of parallelization in sampling. However, this is valid only up to a limit in the case of multitrack adaptive CS – the tracks still have to be wide enough to contain the supports of the wavelet patterns, as was discussed earlier.

Regarding reconstruction accuracy, the simulation re-
Multitrack full sampling (ground truth) | Single-track adaptive CS | Multitrack non-adaptive CS | Multitrack adaptive CS
---|---|---|---

| Relative error: 0% | Relative error: 5.45% | Relative error: 7.32% | Relative error: 5.88% |
| PSNR: $\infty$ | PSNR: 42.6 dB | PSNR: 40.0 dB | PSNR: 41.9 dB |
| Acquisition time: 834 s | Acquisition time: 1695 s | Acquisition time: 54.6 s | Acquisition time: 62.6 s |
| Reconstruction time: 0 s | Reconstruction time: 1.52 s | Reconstruction time: 929 s | Reconstruction time: 1.55 s |
| Total time: 834 s | Total time: 1697 s | Total time: 984 s | Total time: 64.2 s |

![Graphs](image)

**Fig. 4.** Simulations are carried out on the hyperspectral scene Green Peppers [41]. The reconstruction results are presented for each of the methods tested. Pseudo-RGB versions of the reconstructed images are displayed in the first row. Reconstructed spectra are plotted in the second row. The dotted lines (---) are from a point on the green leaves background; the dashed lines (---) are from a point on the pepper near the top left corner; and the solid lines (----) are from a point on a chilli at the bottom right corner. Finally, the third row displays the pixel-wise ratio of the spectral bands at 560 nm and 680 nm.

Results show that multitrack adaptive CS outperforms its non-adaptive counterpart for both the Colour Checker and the U2OS cell scene; the PSNR for multitrack adaptive CS is 2-3 dB higher. Hence, there is evidence that the wavelet coefficients measured capture the important information from the scene effectively. This discrepancy in accuracy could also be due to the scenes being more easily representable in the wavelet basis, or due to the scenes not having sufficiently low total variation as would be necessary for TV-based sparse recovery. Full sampling achieves the best imaging accuracy; but the (relative) errors are smaller than 5% for the multitrack CS methods. The slight loss of accuracy for a huge gain in speed presents a good tradeoff in favor of multitrack adaptive CS in particular.

Besides the above comparisons, one needs to make sure that the accuracy of multitrack adaptive CS is not much behind that of single-track adaptive CS. The simulation results indeed confirm that despite the losses induced by the sub-queue formation in multitrack adaptive CS, the PSNR discrepancy of the two methods does not exceed 1 dB. Hence, this shows that the ABS-WP algorithm has been successfully extended to the multitrack measurement scenario.

Next, **Fig. 4** shows a similar comparison between the four methods for the Green Peppers image. The first row of **Fig. 4** yields similar conclusions as **Fig. 3** multitrack non-
adaptive CS has the shortest acquisition time; multitrack adaptive CS has the shortest total time; and the relative errors of the multitrack CS methods, though higher than in Fig. 3, are still only about 6–7%.

In the second and third rows of Fig. 4, we demonstrate the potential applicability of multitrack CS HSI to agricultural monitoring. Reconstructed spectra from each of the compared methods are plotted in the second row. The dotted lines are from a point on the green leaves background; the dashed lines are from a point on the pepper near the top left corner; and the solid lines are from a point on a chili at the bottom right corner. The multitrack CS methods can distinguish the three, just like full sampling and single-track. Finally, the third row displays the pixel-wise ratio of the spectral bands at 560 nm and 680 nm. The ratio images clearly distinguish the peppers from the background for each of the compared methods, possibly making it easier to perform semantic segmentation. Note that the red regions at the bottom left corners correspond to chilies.

5.2 The effect of illumination level and noise

We modeled measurement noise as Poissonian shot noise with dark noise, as described in Section 4. The factors determining the severity of noise were intensity of the scene $X$, integration time $\Delta t$, and dark current $\alpha$. The effects of these on the CS reconstructions are examined here via simulations, for the multitrack methods. We do not analyse single track CS further, since we established in Section that it is much slower than the multitrack methods.

![Fig. 5. The effect of illumination level on reconstructed image PSNR for multitrack full sampling, multitrack non-adaptive CS, and multitrack adaptive CS. The illumination level (horizontal axis) decreases from left to right, corresponding to an increasing prominence of shot noise. The gray shading is also meant to indicate the illumination level. The dark current $\alpha$ is fixed at 0.](image)

Fig. 5. The effect of illumination level on reconstructed image PSNR for multitrack full sampling, multitrack non-adaptive CS, and multitrack adaptive CS. The illumination level (horizontal axis) decreases from left to right, corresponding to an increasing prominence of shot noise. The gray shading is also meant to indicate the illumination level. The dark current $\alpha$ is fixed at 0.

We modeled measurement noise as Poissonian shot noise with dark noise, as described in Section 4. The factors determining the severity of noise were intensity of the scene $X$, integration time $\Delta t$, and dark current $\alpha$. The effects of these on the CS reconstructions are examined here via simulations, for the multitrack methods. We do not analyse single track CS further, since we established in Section that it is much slower than the multitrack methods.

![Fig. 6. The effect of dark current $\alpha$ on reconstructed image PSNR for multitrack full sampling, multitrack non-adaptive CS, and multitrack adaptive CS. The max scene intensity is fixed at $5 \times 10^4$ photons $s^{-1} \text{voxel}^{-1}$, and the integration time is 0.1 s](image)

First, we set $\alpha = 0$ and note that $X$ and $\Delta t$ play a similar role in the resulting measurement:

$$\text{Pois} \left( \left( \zeta^{(i)} \right)^T X^{(i)} \Delta t \right)$$

Hence, it is reasonable to evaluate the effect of the product $X\Delta t$ on the reconstruction accuracy, instead of using $X$ and $\Delta t$ individually. Furthermore, $X\Delta t$ has units of photons, and therefore, is physically meaningful as the photon counts emanating from the scene during the integration time interval. As a result, we quantify the illumination level by $\max(X\Delta t)$ – the max photon count per voxel emanating from the scene during the time interval $\Delta t$.

Fig. 5 shows the simulation results – the effect of illumination level on reconstructed image accuracy (in PSNR) – for multitrack full sampling, multitrack non-adaptive CS, and multitrack adaptive CS. The illumination level plotted on the horizontal axis decreases from left to right, corresponding to an increasing prominence of shot noise.

Reconstruction accuracy is highest for full sampling, similar to Section 5.1 across all illumination levels. However, the accuracy of non-adaptive CS starts approaching that of full sampling at lower illumination levels of $\sim 5 \times 10^2$ photons $\text{voxel}^{-1}$. We also note that for illumination levels higher than $\sim 5 \times 10^4$ photons $\text{voxel}^{-1}$, the accuracy of adaptive CS overtakes that of non-adaptive CS.

A prominent trend is that the reconstruction accuracy of full sampling and adaptive CS are more sensitive to illumination level as compared to non-adaptive CS. This is because the DMD patterns used in the former two have
ports than the wavelet patterns used in adaptive CS and the uses delta function subpatterns, which have smaller sup-}

strengths at the detector, as mentioned earlier. Full sampling requirements of detectors.

can allow the use of cheaper detectors, or relax the cooling CS methods an advantage – their robustness to dark noise compared to adaptive and non-adaptive CS. This gives the full sampling is much more sensitive to dark current as the simulation results – reconstruction accuracies across a mination level of \( \alpha = 5 \times 10^4 \text{ photons s}^{-1} \text{ voxel}^{-1} \) are fixed.

small supports, such that only a small percentage of incident light is reflected towards the detector by the DMD. This leads to a weaker signal at the detector, which is more susceptible to shot noise. On the contrary, the DMD patterns used in non-adaptive CS reflect about 50% of the incident light towards the detector, which is much higher than the other two methods. The resultant stronger signal at the detector is thus less affected by shot noise.

Noting also that this sensitivity is a measure of robust-ness to shot noise, we can conclude from comparing the slopes of the PSNR-illumination plots that the robustness of our CS methods is comparable (or even better) to the robustness of full sampling.

Next, we evaluate the effect of dark current by varying \( \alpha \). We fix the max scene intensity at \( 5 \times 10^3 \text{ photons s}^{-1} \text{ voxel}^{-1} \), and the integration time 0.1 s, yielding an illumination level of \( 5 \times 10^3 \text{ photons voxel}^{-1} \). Fig. 6 shows the simulation results – reconstruction accuracies across a wide range of dark current values. The plots indicate that full sampling is much more sensitive to dark current as compared to adaptive and non-adaptive CS. This gives the CS methods an advantage – their robustness to dark noise can allow the use of cheaper detectors, or relax the cooling requirements of detectors.

These discrepancies could again be attributed to signal strengths at the detector, as mentioned earlier. Full sampling uses delta function subpatterns, which have smaller sup-ports than the wavelet patterns used in adaptive CS and the random patterns used in non-adaptive CS. Furthermore, the CS methods have a regularization method built into them – sparsity in the wavelet domain for adaptive CS, and low TV norms for non-adaptive CS – making them more noise-robust in general.

Lastly, we examine the effect of integration time. Both reconstruction accuracy and acquisition time are affected by this quantity, leading to tradeoffs. The effect of integration time on accuracy was effectively captured in Fig. 5 since the illumination level was determined by \( X \Delta t \), which is linear in \( \Delta t \). The \( \Delta t \)-PSNR relationship is plotted explicitly in the top panel of Fig. 7 for the three multitrack methods. Here, we have fixed \( \alpha = 0 \) and \( \max(X) = 5 \times 10^4 \text{ photons s}^{-1} \text{ voxel}^{-1} \).

Acquisition time and total time are related to integration time as per the following equation:

\[
\text{total time} = (\text{number of shots} \times \Delta t) + \Delta \text{DMD-pattern-related time} + \text{reconstruction time} \quad (29)
\]

The DMD-pattern-related time is independent of integration time, and comprises of (i) the time required to adaptively determine the DMD patterns (0 s for non-adaptive), (ii) the time required to load the DMD patterns into a computer’s memory, (iii) the time required to switch the DMD patterns, as per the pattern switch rate. Similarly, the reconstruction time is also independent of integration time. These components are offset terms in Eq. (29). The remaining term provides a linear relationship between total time and \( \Delta t \), the slope of which is governed by the sampling ratio. The bottom panel of Fig. 7 plots these relationships for the three multitrack methods.

The total time curve for full sampling is the steepest, due to a larger number of shots. Non-adaptive CS, on the other hand, has the highest reconstruction time, leading to a large offset when \( \Delta t \to 0 \). The lowest total time is achieved by adaptive CS, as was also shown in Section 5.1.

Both panels in Fig. 7 can be used together for choosing the optimal multitrack method for given accuracy and time requirements. Suppose a minimum PSNR must be achieved. Then the top panel would yield the integration times required by each of the methods to achieve this PSNR threshold. Based on the integration times, the bottom panel would then predict the total time required by the three methods, allowing us to choose the one with the smallest time. Conversely, if we have a maximum total time requirement, a similar procedure based on Fig. 7 could then be used to choose the method which achieves the highest PSNR under this time constraint.

6 Conclusion

In this paper, we have developed two methods for multitrack compressed-sensing-based hyperspectral imaging – a non-adaptive approach based on sparse recovery, and an adaptive approach based on sampling significant wavelet coefficients of the hyperspectral scene.

We have verified via simulations that both the methods can achieve short measurement times, while maintaining high imaging accuracy. But they have different advantages:
the adaptive method enjoys a much smaller reconstruction time and computational burden, whereas the non-adaptive method is more robust to noise.

Next, we have also shown that the two methods are generalizable to different types of scenes – reflectance of macroscopic objects as well as Raman signals of cells. This is likely because the methods do not rely upon information from fixed data sets.

Overall, the CS-HSI methods achieve reconstruction accuracies close to full sampling, and noise-robustness even better than full sampling. This, along with the high acquisition speed as well as high reconstruction speed in the case of multitrack adaptive CS, demonstrates these methods to be effective.

ACKNOWLEDGMENT

The authors would like to acknowledge the support for this work from the Agency of Science Technology and Research (A*STAR), Singapore and Nanyang Technological University, Singapore.

REFERENCES

[1] G. Lu and B. Fei, “Medical hyperspectral imaging: a review,” J. Biomed. Opt., vol. 19, no. 1, p. 019001, 2014.
[2] Q. Li, W. Wang, C. Ma, and Z. Zhu, “Detection of physical defects in solar cells by hyperspectral imaging technology,” Opt. Laser Technol., vol. 42, no. 6, pp. 1010–1013, 2010.
[3] L. M. Dale et al., “Hyperspectral imaging applications in agriculture and agro-food product quality and safety control: A review,” Appl. Spectrosc. Rev., vol. 48, no. 2, pp. 142–159, 2013.
[4] A. F. Goetz, “Three decades of hyperspectral remote sensing of the earth: A personal view,” Remote Sens. Environ., vol. 113, pp. 55–516, 2009, Imaging Spectroscopy Special Issue.
[5] J. Qin, K. Chao, and M. Kim, “Chapter 6 - Introduction to Raman chemical imaging technology,” in Computer Vision Technology for Food Quality Evaluation (Second Edition), 2nd ed., D.-W. Sun, Ed. Academic Press, 2016, ch. 6, pp. 141–171.
[6] A. C. Fisher, M.-H. Kamga, C. Agarabi, K. Bronson, S. L. Lee, and S. Yoon, “The current scientific and regulatory landscape in advancing integrated continuous biopharmaceutical manufacturing,” Trends Biotechnol., vol. 37, no. 3, pp. 253–267, 2019.
[7] C. Únědky, S. Ertung, T. Mistretta, and B. Loosse, “Applied advanced process analytics in biopharmaceutical manufacturing: Challenges and prospects in real-time monitoring and control,” J. Process Control, vol. 20, no. 9, pp. 1009–1018, 2010, ADCHEM 2009 Special Issue.
[8] N. A. Hagen and M. W. Kudennov, “Review of snapshot spectral imaging technologies,” Opt. Eng., vol. 52, no. 9, pp. 1–23, 2013.
[9] E. J. Candés, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” IEEE Trans. Inf. Theory, vol. 52, no. 2, pp. 489–509, Feb. 2006.
[10] D. L. Donoho, “Compressed sensing,” IEEE Trans. Inf. Theory, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[11] I. Y. August and A. Stern, “Spectral and hyperspectral compressive imaging,” in Optical compressive imaging, A. Stern, Ed. CRC Press, 2017, ch. 9.
[12] A. Wagadarikar, R. John, R. Willett, and D. Brady, “Single disperser design for coded aperture snapshot spectral imaging,” Opt. Eng., vol. 47, no. 10, pp. B4–B51, 2008.
[13] G. R. Arce, D. J. Brady, L. Carin, H. Arguello, and D. S. Kittle, “Compressive coded aperture spectral imaging: An introduction,” IEEE Signal Process. Mag., vol. 31, no. 1, pp. 105–115, Jan. 2014.
[14] X. Cao et al., “Computational snapshot multispectral cameras: Toward dynamic capture of the spectral world,” IEEE Signal Process. Mag., vol. 33, no. 5, pp. 95–108, Sep. 2016.
[15] T. Sun and K. Kelly, “Compressive sensing hyperspectral imager,” in Computational Optical Sensing and Imaging, Optical Society of America, 2009, paper CThA5.
[16] M. Rani, S. B. Dhook, and R. B. Deshmukh, “A systematic review of compressive sensing: Concepts, implementations and applications,” IEEE Access, vol. 6, pp. 4875–4894, Jan. 2018.
[17] E. C. Marques, N. Maciel, L. Naviner, H. Cai, and J. Yang, “A review of sparse recovery algorithms,” IEEE Access, vol. 7, pp. 1300–1322, Dec. 2019.
[18] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” IEEE Trans. Signal Process., vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
[19] A. Mousavi, A. B. Patel, and R. G. Baraniuk, “A deep learning approach to structured signal recovery,” in 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 1336–1343.
[20] R. Li, Y. Zheng, D. Wen, and Z. Song, “A deep learning approach to real-time recovery for compressive hyper spectral imaging,” in 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC), pp. 1030–1034.
[21] Z. Xiong, Z. Shi, H. Li, L. Wang, D. Liu, and F. Wu, “HSCNN: CNN-based hyperspectral image recovery from spectrally undersampled projections,” in 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 518–525.
[22] L. Wang, T. Zhang, Y. Fu, and H. Huang, “HyperReconNet: Joint coded aperture optimization and image reconstruction for compressive hyperspectral imaging,” IEEE Trans. Image Process., vol. 28, no. 5, pp. 2257–2270, May 2019.
[23] S. Deutsch, A. Averbush, and S. Dekel, “Adaptive compressed image sensing based on wavelet modeling and direct sampling,” in SAMPTA09, L. Fesquet and B. Torrésani, Eds., Marseille, France, May 2009, p. General session.
[24] A. Averbush, S. Dekel, and S. Deutsch, “Adaptive compressed image sensing using dictionaries,” SIAM J. Imag. Sci., vol. 5, no. 1, pp. 57–89, 2012.
[25] H. Dai et al., “Adaptive compressed sampling based on extended wavelet trees,” Appl. Opt., vol. 53, no. 29, pp. 6619–6628, 2014.
[26] F. Rousset, N. Ducros, A. Farina, G. Valentinì, C. D’Andrea, and F. Pevrin, “Adaptive basis scan by wavelet prediction for single-pixel imaging,” IEEE Trans. Comput. Imag., vol. 3, no. 1, pp. 36–46, 2017.
[27] Y. Lu, X. Xu, and L. Wang, “Smart manufacturing process and system automation – a critical review of the standards and envisioned scenarios,” J. Manuf. Syst., vol. 56, pp. 312–325, 2020.
[28] S.-H. Huang and Y.-C. Pan, “Automated visual inspection in the semiconductor industry: A survey,” Comput. Ind., vol. 66, pp. 10–15, 2015.
[29] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, “A simple proof of the restricted isometry property for random matrices,” Constr. Approx., vol. 28, no. 3, pp. 253–263, 2008.
[30] T. Gold, E. Gehm, R. John, D. J. Brady, R. M. Winkler, and T. J. Schulz, “Single-shot compressive spectral imaging with a dual-disperser architecture,” Opt. Express, vol. 15, no. 21, pp. 14013–14027, 2007.
[31] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, “Single-pixel imaging via compressive sampling,” IEEE Signal Process. Mag., vol. 25, no. 2, pp. 83–91, Mar. 2008.
[32] A. A. Wagadarikar, N. P. Pitsianis, X. Sun, and D. J. Brady, “Video rate spectral imaging using a coded aperture snapshot spectral imager,” Opt. Express, vol. 17, no. 8, pp. 6368–6388, 2009.
[33] D. Kittle, K. Choi, A. Wagadarikar, and D. J. Brady, “Multiframe image estimation for coded aperture snapshot spectral imagers,” Appl. Opt., vol. 49, no. 36, pp. 6824–6833, Dec. 2010.
[34] F. Magalhaes, F. M. Araújo, M. Correia, M. Abolbashari, and F. Farahi, “High-resolution hyperspectral single-pixel imaging system based on compressive sensing,” Opt. Eng., vol. 51, no. 7, pp. 1–7, 2012.
[35] S. Mallat, A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way, 3rd ed. USA: Academic Press, Inc., 2008.
[36] J. Hahn, C. Debos, M. Leisniger, and A. M. Zoubir, “Compressive sensing and adaptive direct sampling in hyperspectral imaging,” Digit. Signal Process., vol. 26, pp. 113–126, 2014.
[37] F. Rousset, N. Ducros, F. Pevrin, G. Valentinì, C. D’Andrea, and A. Farina, “Time-resolved multispectral imaging based on an adaptive single-pixel camera,” Opt. Express, vol. 26, no. 8, pp. 10550–10558, 2018.
[38] G. Lu, “Block compressed sensing of natural images,” in 2007 15th International Conference on Digital Signal Processing, pp. 403–406.
[39] J. E. Fowler, S. Mun, and E. W. Tramel, “Block-based compressed sensing of images and video,” Found. Trends Signal Process., vol. 4, no. 4, pp. 297–416, 2012.

[40] Y. Pfeffer and M. Zibulevsky, “A micro-mirror array based system for compressive sensing of hyperspectral data,” Computer Science Department, Technion, Tech. Rep., 2010.

[41] B. Arad and O. Ben-Shahar, “Sparse recovery of hyperspectral signal from natural RGB images,” in Computer Vision – ECCV 2016. Springer International Publishing, pp. 19–34.

[42] F. Masia, A. Karuna, P. Borri, and W. Langbein, “Hyperspectral image analysis for CARS, SRS, and Raman data,” J. Raman Spectrosc., vol. 46, no. 8, pp. 727–734, 2015.

[43] C. Li, “An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing,” Master’s thesis, Rice University, 2010.