SIAM J. Imaging Sciences
Vol. 12, No. 1, pp. 392–419

SPARCOM: Sparsity Based Super-resolution Correlation Microscopy*

Oren Solomon†, Yonina C. Eldar†, Maor Mutzafi‡, and Mordechai Segev‡

Abstract. In traditional optical imaging systems, the spatial resolution is limited by the physics of diffraction, which acts as a low-pass filter. The information on subwavelength features is carried by evanescent waves, never reaching the camera, thereby posing a hard limit on resolution: the so-called diffraction limit. Modern microscopic methods enable super-resolution, by employing fluorescence techniques. State-of-the-art localization based fluorescence subwavelength imaging techniques such as PALM and STORM achieve subdiffraction spatial resolution of several tens of nanometers. However, they require tens of thousands of exposures, which limits their temporal resolution. We have recently proposed *sparsity based super-resolution correlation microscopy* (SPARCOM), which exploits the sparse nature of the fluorophore distribution, alongside a statistical prior of uncorrelated emissions, and showed that SPARCOM achieves spatial resolution comparable to PALM/STORM, while capturing the data hundreds of times faster. Here, we provide a detailed mathematical formulation of SPARCOM, which in turn leads to an efficient numerical implementation, suitable for large-scale problems. We further extend our method to a general framework for sparsity based super-resolution imaging, in which sparsity can be assumed in other domains such as wavelet or discrete-cosine, leading to improved reconstructions in a variety of physical settings.

Key words. fluorescence, high-resolution imaging, compressed sensing, correlation

AMS subject classifications. 65R32, 78A46, 62H20

DOI. 10.1137/18M1174921

1. Introduction.

Spatial resolution in diffractive optical imaging is limited by one half of the optical wavelength, known as Abbe's diffraction limit [5, 15]. Modern microscopic methods enable super-resolution, even though information on subwavelength features is absent in the measurements. One of the leading subwavelength imaging modalities is based on fluorescence (PALM [4] and STORM [30]). Its basic principle consists of attaching fluorescent molecules (point emitters) to the features within the sample, exciting the fluorescence with short-wavelength illumination, and then imaging the fluorescent light. PALM and STORM rely on acquiring a sequence of diffraction-limited images, such that in each frame only a sparse set of emitters (fluorophores) is active. The position of each fluorophore is found through a superlocalization procedure [31]. Subsequent accumulation of single-molecule localizations results in a grainy high-resolution image, which is then smoothed to form the final super-

*Received by the editors March 12, 2018; accepted for publication (in revised form) November 26, 2018; published electronically February 19, 2019. A portion of this work is based on preliminary results presented in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP), 2017, pp. 6215–6219.

http://www.siam.org/journals/siims/12-1/M117492.html

Funding: This work was supported by the European Union's Horizon 2020 research and innovation program under grant agreement 646804-ERC-COG-BNYQ and by the Ollendorf Foundation.

†Department of Electrical Engineering, Technion Israel Institute of Technology, Haifa 32000, Israel (orensol@campus.technion.ac.il, yonina@ee.technion.ac.il).

‡Department of Physics and Solid State Institute, Technion Israel Institute of Technology, Haifa 32000, Israel (maormutzafi@gmail.com, msegev@tx.technion.ac.il).

resolved image. The final image has a spatial resolution of tens of nanometers.

A major disadvantage of these fluorescence techniques is that they require tens of thousands of exposures. This is because in every frame, the diffraction-limited image of each emitter must be well separated from its neighbors, to enable the identification of its exact position. This inevitably leads to a long acquisition cycle, typically on the order of several minutes [30]. Consequently, fast dynamics cannot be captured by PALM/STORM.

To reduce acquisition time, an alternative technique named super-resolution optical fluctuation imaging (SOFI) was proposed [10], which uses high fluorophore density, to reduce integration time. In SOFI, the emitters usually overlap in each frame, so that superlocalization cannot be performed. However, since the emitted light from each emitter, which are uncorrelated between different emitters, are captured over a period of several frames by the camera, consecutive frames contain information in the pixelwise temporal correlation between them. The measurements are therefore processed such that correlative information is used, enabling the recovery of features that are smaller than the diffraction limit by a factor of $\sqrt{2}$. By calculating higher order statistics (HOS) in the form of cumulants [20] of the time-trace of each pixel, a theoretical resolution increase equal to the square root of the order of the statistics can in principle be achieved. Using the cross-correlation between pixels over time, it is possible to increase the resolution gain further, to an overall factor that scales linearly with the order of the statistical calculation [11].

SOFI enables the processing of images with high fluorophore density, thus reducing the number of required frames for image recovery and achieving increased temporal resolution over localization based techniques. However, at least thus far, the spatial resolution offered by SOFI does not reach the level of super-resolution obtained through STORM and PALM, even when using HOS. The use of HOS can in principle increase the spatial resolution, but higher (than the order of two) statistical calculations require an increasingly large number of frames for their estimation, degrading temporal resolution. Moreover, SOFI suffers from a phenomenon known as *dynamic range expansion*, in which weak emitters are masked in the presence of strong ones. The effect is worsened as the statistical order increases, which in practice limits the applicability of SOFI to second order statistics and a moderate improvement in spatial resolution.

Recently, we proposed a method for super-resolution imaging with short integration time called *sparsity based super-resolution correlation microscopy* (SPARCOM) [33]. In [33], we have shown that our method achieves spatial resolution similar to PALM/STORM, from only tens/hundreds of frames, by performing sparse recovery [12] on correlation information, leading to an improvement of the temporal resolution by two orders of magnitude. Mathematically, SPARCOM recovers the support of the emitters, by recovering their variance values. Sparse recovery from correlation information was previously proposed to improve sparse recovery from a small number of measurements [26, 12, 8]. When the nonzero entries of the sparse signal are uncorrelated, support size recovery can be theoretically increased up to $O(M^2)$, where M is the length of a single measurement vector. In SPARCOM, we use similar concepts to enhance resolution and improve the signal to noise ratio (SNR) in optical imaging. By performing sparse recovery on correlation information, SPARCOM enjoys the same features of SOFI (processing of high fluorophore density frames over short movie ensembles and the use of correlative information), while offering the possibility of achieving single-molecule resolution comparable to that of PALM/STORM. Moreover, by relying on correlation information only, SPARCOM overcomes the dynamic range problem of SOFI when HOS are used and results in improved image reconstruction.

In this paper, we focus on three major contributions with respect to our recent work. The first is to provide a thorough and detailed formulation of SPARCOM, elaborating on its mathematical aspects. Second, we extend SPARCOM to the case when super-resolution is considered in additional domains such as the wavelet or discrete cosine transform domains. Third, we show how SPARCOM exploits structural information to achieve a computationally efficient implementation. This goal is achieved by considering the SPARCOM reconstruction model in the sampled Fourier space, which leads to fast image reconstruction, suitable for large-scale problems, without the need to store large matrices in memory.

The rest of the paper is organized as follows. Section 2 explains the problem and the key idea of SOFI. In section 3, we formulate our proposed solution. A detailed explanation of our algorithm, implementation, and additional extensions to super-resolution in arbitrary bases are provided in sections 4 and 5. Simulation results are presented in section 6.

Throughout the paper, x represents a scalar, $\mathbf{x}$ represents a vector, $\mathbf{X}$ represents a matrix, and $\mathbf{I}_{N\times N}$ represents the $N \times N$ identity matrix. The notation $||\cdot||_p$ represents the standard p-norm, and $||\cdot||_F$ is the Frobenius norm. Subscript x_l denotes the lth element of $\mathbf{x}$, and $\mathbf{x}_l$ is the lth column of $\mathbf{X}$. Superscript $\mathbf{x}^{(p)}$ represents $\mathbf{x}$ at iteration p, $\mathbf{T}^*$ denotes the adjoint of $\mathbf{T}$, and $\bar{\mathbf{A}}$ is the complex conjugate of $\mathbf{A}$.

2. Problem formulation and SOFI.

Following [10, 11], the acquired fluorescence signal in the object plane is modeled as a set of L independently fluctuating point sources, with resulting spatial fluorescence source distribution

$$J(\mathbf{r},t) = \sum_{k=0}^{L-1} \delta(\mathbf{r}-\mathbf{r}_k)s_k(t).$$

Each source (or emitter) has its own time-dependent brightness function $s_k(t)$ and is located at position $\mathbf{r}_k \in \mathbb{R}^2,\ k = 0,\dots,L-1$. The acquired signal in the image plane is the result of the convolution between $J(\mathbf{r},t)$ and the impulse response of the microscope $u(\mathbf{r})$ (also known as the *point spread function* (PSF)),

$$f(\mathbf{r},t) = \sum_{k=0}^{L-1} u(\mathbf{r}-\mathbf{r}_k)s_k(t). \tag{2.1}$$

We assume that the measurements are acquired over a period of $t \in [0,T]$. Ideally, our goal is to recover the locations of the emitters, $\mathbf{r}_k$, and their variances with high spatial resolution and short integration time. The final high-resolution image is constructed from the recovered variance value for each emitter.

To proceed, we assume the following:

A1. The locations $\mathbf{r}_k,\ k = 0,\dots,L-1$, do not depend on time.

A2. The brightness is uncorrelated in space; namely, $E\{\tilde{s}_i(t_1)\tilde{s}_j(t_2)\} = 0$ for all $i \neq j$ and for all t_1, t_2, where $\tilde{s}_k(t) = s_k(t) - E_k$, with $E_k = E\{s_k(t)\}$.

A3. The brightness functions $s_k(t)$, $k = 0, \dots, L-1$, are wide sense stationary so that $E\{\tilde{s}_k(t)\tilde{s}_k(t+\tau)\} = g_k(\tau)$ for some function $g_k(\tau)$.

Using assumptions **A2** and **A3**, the autocorrelation function at each point $\mathbf{r}$ can be computed as

$$G_f(\mathbf{r},\tau) = E\{\tilde{f}(\mathbf{r},t)\tilde{f}(\mathbf{r},t+\tau)\} = \sum_{k=0}^{L-1} u^2(\mathbf{r}-\mathbf{r}_k)g_k(\tau), \tag{2.2}$$

where $\tilde{f}(\mathbf{r},t) = f(\mathbf{r},t) - E\{f(\mathbf{r},t)\} = \sum_{k=0}^{L-1} u(\mathbf{r}-\mathbf{r}_k)\tilde{s}_k(t)$. Assumption **A1** indicates that $\mathbf{r}_k$ are time-independent during the acquisition period. The final SOFI image is the value of $G_f(\mathbf{r},0)$ at each point $\mathbf{r}$, where $g_k(0)$ represents the variance of emitter s_k. We see from (2.2) that the autocorrelation function depends on the PSF squared. If the PSF is assumed to be Gaussian, then this calculation reduces its width by a factor of $\sqrt{2}$. However, the final SOFI image retains the same low-resolution grid as the captured movie. Similar statistical calculations can be performed for adjacent pixels in the movie, leading to a simple interpolation grid with an increased number of pixels in the high-resolution image, but at the cost of increased statistical order using cumulants [20]. HOS reduce the PSF size further but at the expense of degraded SNR and dynamic range for a given number of frames [11].

In the next section, we provide a rigorous and detailed description of our sparsity based method, first presented in [33], for estimating $\mathbf{r}_k$ and $g_k(0)$ on a high-resolution grid. We rely on correlation only, without resorting to HOS, thus maintaining a short acquisition time, similar to correlation based SOFI. In contrast to SOFI, we exploit the sparse nature of the emitters' distribution and recover a high-resolution image on a much denser grid than the camera's grid. This leads to spatial super-resolution without the need to perform interpolation using HOS [11].

3. SPARCOM.

3.1. High-resolution representation.

To increase resolution by exploiting sparsity, we start by introducing a Cartesian sampling grid with spacing Δ_L, which we refer to as the *low-resolution grid*. The low-resolution signal (2.1) can be expressed over this grid as

$$f[m\Delta_L, n\Delta_L, t] = \sum_{k=0}^{L-1} u[m\Delta_L - m_k, n\Delta_L - n_k]s_k(t), \quad m,n = [0,\dots,M-1], \tag{3.1}$$

where $\mathbf{r}_k = [m_k, n_k]^T \in \mathbb{R}^2$. We discretize the possible locations of the emitters $\mathbf{r}_k$, over a discrete Cartesian grid $i,l = 0,\dots,N-1$, $L \ll N$, with resolution Δ_H, such that $[m_k, n_k] = [i_k, l_k]\Delta_H$ for some integers $i_k, l_k \in [0,\dots,N-1]$. We refer to this grid as the *high-resolution grid*. For simplicity, we assume that $\Delta_L = P\Delta_H$ for some integer $P \geq 1$, and consequently $N = PM$. As each pixel $[m_k, n_k]$ is now divided into P times smaller pixels, the high-resolution grid allows us to detect emitters with a spatial error which is P times smaller than on the camera grid. Typical values of camera pixels sizes can be around 100nm, which is typically half the diffraction limit. Thus, recovering the emitters on a finer grid leads to a better depiction of subdiffraction features.

The latter discretization implies that (3.1) is sampled (spatially) over a grid of size $M \times M$, while the emitters reside on a grid of size $N \times N$, with the ilth pixel having a fluctuation function $s_{il}(t)$ (only L such pixels actually contain fluctuating emitters, according to (3.1)). If there is no emitter in the ilth pixel, then $s_{il}(t) = 0$ for all t. We further assume that the PSF u is known.

Rewriting (3.1) in Cartesian form with respect to the grid of emitters yields

$$f[m\Delta_L, n\Delta_L, t] = \sum_{i=0}^{N-1} \sum_{l=0}^{N-1} u[m\Delta_L - i\Delta_H, n\Delta_L - l\Delta_H] s_{il}(t), \tag{3.2}$$

and additionally it holds that

$$m\Delta_L - i\Delta_H = (mP - i)\Delta_H.$$

Omitting the spacing Δ_H, we can rewrite (3.2) as

$$f[mP, nP, t] = \sum_{i,l=0}^{N-1} u[mP - i, nP - l] s_{il}(t). \tag{3.3}$$

3.2. Fourier analysis.

We next present (3.3) in the Fourier domain, which will lead to an efficient implementation of our method.

Since $y[m,n,t] = f[mP, nP, t]$ is an $M \times M$ sequence, denote by $Y[k_m, k_n, t]$ its $M \times M$ two-dimensional (2D) discrete Fourier transform (DFT). Performing an $M \times M$ 2D-DFT on $y[m,n,t]$ yields

$$\begin{aligned} Y[k_m, k_n, t] &= \sum_{m,n=0}^{M-1} f[mP, nP, t] e^{-j\frac{2\pi}{M}k_m m} e^{-j\frac{2\pi}{M}k_n n} \\ &= \sum_{i,l=0}^{N-1} s_{il}(t) \sum_{\hat{m},\hat{n}=0,P,\dots}^{MP-P} u[\hat{m} - i, \hat{n} - l] e^{-j\frac{2\pi}{MP}k_m \hat{m}} e^{-j\frac{2\pi}{MP}k_n \hat{n}}, \end{aligned}$$

where we defined $\hat{m} = mP$ and $\hat{n} = nP$ and $k_m, k_n = 0, \dots, M-1$. Next, consider $\hat{m}, \hat{n} = 0, \dots, N-1$ and define the $N \times N$ sequence,

$$\tilde{u}[\hat{m}, \hat{n}] = \begin{cases} u\left[\hat{m}, \hat{n}\right], & \hat{m}, \hat{n} = 0, P, \dots, N-P, \\ 0 & \text{else}, \end{cases} \tag{3.4}$$

where u is the discretized PSF sampled over $M \times M$ points of the low-resolution grid. We can then equivalently write

$$Y[k_m, k_n, t] = \sum_{i,l=0}^{N-1} s_{il}(t) \sum_{\hat{m},\hat{n}=0}^{N-1} \tilde{u}[\hat{m} - i, \hat{n} - l] e^{-j\frac{2\pi}{N}k_m \hat{m}} e^{-j\frac{2\pi}{N}k_n \hat{n}}. \tag{3.5}$$

By defining $p = \hat{m} - i$ and $q = \hat{n} - l$, (3.5) becomes

$$Y[k_m, k_n, t] = \tilde{U}[k_m, k_n] \sum_{i,l=0}^{N-1} s_{il}(t) e^{-j\frac{2\pi}{N}k_m i} e^{-j\frac{2\pi}{N}k_n l}, \tag{3.6}$$

with

$$\tilde{U}[k_m, k_n] = \sum_{p,q=0}^{N-1} \tilde{u}[p,q] e^{-j\frac{2\pi}{N}k_m p} e^{-j\frac{2\pi}{N}k_n q}. \tag{3.7}$$

Note that $\tilde{U}[k_m, k_n]$ is the $N \times N$ 2D-DFT of the $N \times N$ sequence $\tilde{u}$, evaluated at discrete frequencies $k_m, k_n = 0, \ldots, M-1$. From (3.4) and (3.7), it holds that $\tilde{U}[e^{-j\frac{2\pi}{N}k_m}, e^{-j\frac{2\pi}{N}k_n}] = U[e^{-j\frac{2\pi}{M}k_m}, e^{-j\frac{2\pi}{M}k_n}]$ for $k_m, k_n = 0, \ldots, M-1$ ($N = PM$), where U is the $M \times M$ 2D-DFT of u sampled on the low-resolution grid.

Denote the columnwise stacking of each frame $Y[k_m, k_n, t]$ as an M^2 long vector $\mathbf{y}(t)$. In a similar manner, $\mathbf{s}(t)$ is a length-N^2 vector stacking of $s_{il}(t)$ for all il. We further define the $M^2 \times M^2$ diagonal matrix $\mathbf{H} = \text{diag}\{U[0,0], \ldots, U[M-1, M-1]\}$. Vectorizing (3.6) yields

$$\mathbf{y}(t) = \mathbf{H}(\mathbf{F}_M \otimes \mathbf{F}_M)\mathbf{s}(t) = \mathbf{A}\mathbf{s}(t), \quad \mathbf{A} \in \mathbb{C}^{M^2 \times N^2}, \tag{3.8}$$

where $\mathbf{s}(t)$ is an L-sparse vector and $\mathbf{F}_M$ denotes a partial $M \times N$ DFT matrix whose M rows are the corresponding M low frequency rows from a full $N \times N$ discrete Fourier matrix.

Define the autocorrelation matrix of $\mathbf{y}(t)$ as

$$\mathbf{R}_y(\tau) = E\left\{(\mathbf{y}(t) - E\{\mathbf{y}(t)\})(\mathbf{y}(t+\tau) - E\{\mathbf{y}(t+\tau)\})^H\right\}. \tag{3.9}$$

From (3.8),

$$\mathbf{R}_y(\tau) = \mathbf{A}\mathbf{R}_s(\tau)\mathbf{A}^H. \tag{3.10}$$

Under assumption **A2**, $\mathbf{R}_s(\tau)$, the autocorrelation matrix of $\mathbf{s}(t)$, is a diagonal matrix. Therefore, (3.10) may be written as

$$\mathbf{R}_y(\tau) = \sum_{l=1}^{N^2} \mathbf{a}_l \mathbf{a}_l^H r_{s_l}(\tau), \tag{3.11}$$

with $\mathbf{a}_l$ being the lth column of $\mathbf{A}$, $\mathbf{r}_s(\tau) = \text{diag}\{\mathbf{R}_s(\tau)\}$, and $r_{s_l}(\tau)$ being the lth entry of $\mathbf{r}_s(\tau)$. By taking $\tau = 0$, we estimate the variance of $s_{ij}(t)$, $i, j = 0, \ldots, N-1$ (as written in assumption **A3**). It is also possible to take into account the fact that the autocorrelation matrix $\mathbf{R}_y(\tau)$ may be nonzero for $\tau \neq 0$; for simplicity, we use $\tau = 0$. The support of $\mathbf{r}_s(\tau)$ is equivalent to the support of $\mathbf{s}(t)$, which in turn indicates the locations of the emitters on a grid with spacing Δ_H. Thus, our high-resolution problem reduces to recovering the L nonzero values of $r_{s_l}(0)$ in (3.11).

3.3. Sparse recovery. SPARCOM is based on (3.11), taking into account that $\mathbf{x} = \mathbf{r}_s(0)$ is a sparse vector. We therefore find $\mathbf{x}$ by using a sparse recovery methodology. In our implementation of SPARCOM, we use the LASSO formulation [35] to construct the following convex optimization problem:

$$\text{(F-LASSO)} \qquad \min_{\mathbf{x}\geq\mathbf{0}} \lambda||\mathbf{x}||_1 + \frac{1}{2}\left|\left|\mathbf{R}_y(0) - \sum_{l=1}^{N^2}\mathbf{a}_l\mathbf{a}_l^H x_l\right|\right|_F^2,$$

with a regularization parameter $\lambda \geq 0$ and x_l denoting the lth entry in $\mathbf{x}$. We note that it is possible to write a similar formulation to (F-LASSO) accounting for $\tau > 0$ (without the nonnegativity constraint). Other approaches to sparse recovery may similarly be used.

We solve (F-LASSO) iteratively using the FISTA algorithm [27, 1, 36], which at each iteration performs a gradient step and then a thresholding step. By performing the calculations in the DFT domain, we can calculate the gradient of the smooth part of (F-LASSO), that is, the squared Frobenius norm, very efficiently. We discuss this efficient implementation in detail in section 5.

To achieve even sparser solutions, we implement a reweighted version of (F-LASSO) [6],

$$\text{(3.12)} \qquad \mathbf{x}^{(p+1)} = \underset{\mathbf{x}^{(p)}\geq\mathbf{0}}{\operatorname{argmin}} \lambda||\mathbf{W}^{(p)}\mathbf{x}^{(p)}||_1 + \frac{1}{2}\left|\left|\mathbf{R}_y(0) - \sum_{l=1}^{N^2}\mathbf{a}_l\mathbf{a}_l^H x_l^{(p)}\right|\right|_F^2,$$

where $\mathbf{W}$ is a diagonal weighting matrix and p denotes the number of the current reweighting iteration. Starting from $p = 1$ and $\mathbf{W} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix of appropriate size, the weights are updated after a predefined number of FISTA iterations according to the output of $\mathbf{x}$ as

$$W_i^{(p+1)} = \frac{1}{|x_i^{(p)}| + \epsilon}, \quad i = 1, \dots, N^2,$$

where ϵ is a small nonnegative regularization parameter. After updating the weights, the FISTA algorithm is performed again.

In practice, for a discrete time-lag τ and total number of frames T, $\mathbf{R}_y(\tau)$ is estimated from the movie frames using the empirical correlation

$$\mathbf{R}_y(\tau) = \frac{1}{T-\tau}\sum_{t=1}^{T-\tau}(\mathbf{y}(t) - \bar{\mathbf{y}})(\mathbf{y}(t+\tau) - \bar{\mathbf{y}})^H,$$

with

$$\text{(3.13)} \qquad \bar{\mathbf{y}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{y}(t).$$

In the following sections, we elaborate on our proposed algorithms for solving F-LASSO and the reweighted scheme (3.12). In particular, we explain how they can be implemented efficiently and extended to a more general framework of super-resolution under assumptions of sparsity. Table 1 provides a summary of the different symbols and their roles for convenience.

Table 1
List of symbols.

Symbol	Description
$\otimes$	Kronecker product
$\odot$	Hadamard (elementwise) product
M	Number of pixels in one dimension of the low-resolution grid
N	Number of pixels in one dimension of the high-resolution grid
P	Ratio between N and M
Δ_L	Low-resolution grid sampling interval
Δ_H	High-resolution grid sampling interval
T	Number of acquired frames
L	Number of emitters in the captured sequence
m_k, n_k	Possible positions of emitters on the high-resolution Cartesian grid
L_f	Upper bound on the Lipschitz constant
$\mathcal{T}_\alpha(\cdot)$	Soft thresholding operator with parameter α defined in (4.2)
λ	Regularization parameter
μ	Smoothing parameter for Algorithm 4.4
$u(\cdot)$	$M \times M$ discretized PSF
$\mathbf{y}(t)$	Vectorized $M \times M$ input frame at time t, after FFT
$\mathbf{s}(t)$	Vectorized $N \times N$ emitters intensity frame at time t
$\mathbf{F}_M$	Partial $M \times N$ DFT matrix of the M lowest frequencies
$\mathbf{H}$	Diagonal $M^2 \times M^2$ matrix containing the (vectorized) DFT of the PSF
$\mathbf{A}$	$\mathbf{A} = \mathbf{H}(\mathbf{F}_M \otimes \mathbf{F}_M)$, known as the $M^2 \times N^2$ sensing matrix, as defined in (3.8)
$\mathbf{a}_i$	ith column of $\mathbf{A}$
$\bar{\mathbf{y}}$	Empirical average of the acquired low-resolution frames defined in (3.13)
$\mathbf{R}_y(\tau)$	Autocovariance matrix of input movie's pixels for time-lag τ
$\mathbf{R}_s(\tau)$	Autocovariance matrix of the emitters for time-lag τ
$\mathbf{r_s}/\mathbf{x}$	Diagonal of $\mathbf{R}_s(\tau)$
$\mathbf{M}$	$\mathbf{M} = \lvert\mathbf{A}^H\mathbf{A}\rvert^2$
$\mathbf{v}$	$\mathbf{v} = [\mathbf{a}_1^H\mathbf{R}_y(0)\mathbf{a}_1, \dots, \mathbf{a}_{N^2}^H\mathbf{R}_y(0)\mathbf{a}_{N^2}]^T$
$\nabla f(\cdot)$	Gradient of f given by (4.4)
$K_{\max}$	Maximum number of iterations
$\mathcal{M}(\cdot)$	Vector to matrix transformation, defined in (5.1)
$\mathcal{V}(\cdot)$	Matrix to vector transformation, defined in (5.2)

4. Proximal gradient descent algorithms.

4.1. Variance recovery.

Problem (F-LASSO) can be viewed as a minimization of a decomposition model

$$\min_{\mathbf{x}\geq\mathbf{0}} \lambda g(\mathbf{x}) + f(\mathbf{x}),$$

where f is a smooth, convex function with a Lipschitz continuous gradient and g is a possibly nonsmooth but proper, closed, and convex function. Following [1, 36], we adapt a *fast-proximal algorithm*, similar to FISTA, to minimize the objective of (F-LASSO), as summarized in Algorithm 4.1. Solving (F-LASSO) iteratively involves finding *Moreau's proximal* (prox) mapping [22, 34] of αg for some $\alpha \geq 0$, defined as

$$\text{prox}_{\alpha g}(\mathbf{x}) = \underset{\mathbf{u}\in\mathbb{R}^n}{\text{argmin}} \left\{ \alpha g(\mathbf{u}) + \frac{1}{2}||\mathbf{u} - \mathbf{x}||_2^2 \right\}. \tag{4.1}$$

For $g(x) = ||\mathbf{x}||_1$, $\text{prox}_{\alpha g}(\mathbf{x})$ is given by the well-known *soft-thresholding* operator,

$$\text{prox}_{\alpha||\cdot||_1}(\mathbf{x}) = \mathcal{T}_\alpha(\mathbf{x}) = \max\{|\mathbf{x}| - \alpha, 0\} \cdot \text{sign}(\mathbf{x}), \tag{4.2}$$

where the multiplication, max, and sign operators are performed elementwise. In its simplest form, the proximal gradient method calculates the prox operator on the gradient step of f at each iteration.

Denoting

$$f(\mathbf{x}) = \frac{1}{2}\left|\left|\mathbf{R}_y(0) - \sum_{l=1}^{N^2} \mathbf{a}_l\mathbf{a}_l^H x_l\right|\right|_F^2 \tag{4.3}$$

and differentiating it with respect to $\mathbf{x}$ yields

$$\nabla f(\mathbf{x}) = \mathbf{M}\mathbf{x} - \mathbf{v}, \tag{4.4}$$

where $\mathbf{v} = [\mathbf{a}_1^H\mathbf{R}_y(0)\mathbf{a}_1, \ldots, \mathbf{a}_{N^2}^H\mathbf{R}_y(0)\mathbf{a}_{N^2}]^T$, $\mathbf{M} = |\mathbf{A}^H\mathbf{A}|^2$, and we have used the fact that $\mathbf{x}$ is real since it represents the variance of light intensities. The operation $|\cdot|^2$ is performed elementwise. The (upper bound on the) Lipschitz constant L_f of $f(\mathbf{x})$ is readily given by $L_f = ||\mathbf{M}||_2$, corresponding to the largest eigenvalue of $\mathbf{M}$, since by (4.4)

$$||\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})||_2 \leq ||\mathbf{M}||_2||\mathbf{x} - \mathbf{y}||_2.$$

Calculation of (4.4) is the most computationally expensive part of Algorithm 4.1.[1] Since $\mathbf{M}$ is of dimensions $N^2 \times N^2$, it is usually impossible to store it in memory and apply it straightforwardly in multiplication operations. In section 5 we present an efficient implementation that overcomes this issue, by exploiting the structure of $\mathbf{M}$. We also develop a closed form expression for L_f.

Algorithm 4.1. Fast proximal gradient descent for SPARCOM.

```
Input: $L_f$, $\mathbf{R}_y(0)$, $\lambda > 0$, $K_{\max}$
  Initialize $\mathbf{z}_1 = \mathbf{x}_0 = \mathbf{0}$, $t_1 = 1$, and $k = 1$
  while $k \leq K_{\max}$ or stopping criteria not fulfilled do
    $\nabla f(\mathbf{z}_k) = \mathbf{M}\mathbf{z}_k - \mathbf{v}$
    $\mathbf{x}_k = \mathcal{T}_{\frac{\lambda}{L_f}}(\mathbf{z}_k - \frac{1}{L_f}\nabla f(\mathbf{z}_k))$
    Project to the nonnegative orthant $\mathbf{x}_k(\mathbf{x}_k < \mathbf{0}) = \mathbf{0}$
    $t_{k+1} = 0.5(1 + \sqrt{1 + 4t_k^2})$
    $\mathbf{z}_{k+1} = \mathbf{x}_k + \frac{t_k - 1}{t_{k+1}}(\mathbf{x}_k - \mathbf{x}_{k-1})$
    $k \leftarrow k + 1$
  end while
  return $\mathbf{x}_{K_{\max}}$
```

[1] SPARCOM code is available online from http://webee.technion.ac.il/people/YoninaEldar/software.php.

Implementing the reweighted l_1 minimization of (3.12) involves calculation of the following elementwise soft-thresholding operator:

$$\mathcal{T}_{\frac{\lambda}{L_f}W_i}(x_i) = \max\left\{|x_i| - \frac{\lambda}{L_f}W_i, 0\right\} \cdot \text{sign}(x_i), \tag{4.5}$$

with W_i being the current value of the ith entry of the diagonal of the weighting matrix $\mathbf{W}$. The reweighting procedure is summarized in Algorithm 4.2.

Algorithm 4.2. Iterative reweighted fast proximal gradient for (F-LASSO).

```
Input: $L_f$, $\mathbf{R}_y(0)$, $\lambda > 0$, $\epsilon > 0$, $P_{\max}$
  Initialize Set iteration counter $l = 1$ and $\mathbf{W}^1 = \mathbf{I}$
  while $p \leq P_{max}$ or stopping criteria not fulfilled do
    1: Solve (F-LASSO) using Algorithm 4.1 with (4.5)
    2: Update weights for $i = 1, \dots, N^2$
         $W_i^{(p+1)} = \text{diag}\left\{\frac{1}{\left|\mathbf{x}_1^{(p)}\right|+\epsilon}, \dots, \frac{1}{\left|x_{N^2}^{(p)}\right|+\epsilon}\right\}$
    3: $p \leftarrow p + 1$
  end while
  return $\mathbf{x}_{P_{\max}}$
```

4.2. Regularized super-resolution.

Recall that to achieve super-resolution we assumed that the recovered signal is sparse. Such an assumption arises in the context of fluorescence microscopy, in which the imaged object is labeled with fluorescing molecules such that the molecular distribution or the desired features themselves are spatially sparse. In many cases, the sought after signal has additional structure which can be exploited alongside sparsity, especially since attaching fluorescing molecules to subcellular organelles serves as a means to image these structures, which are of true interest. Thus, when considering sparsity based super-resolution reconstruction, we can consider a more general context of sparsity within the desired signal.

4.2.1. Total variation super-resolution imaging.

We first modify (F-LASSO) to incorporate a total variation (TV) regularization term on $\mathbf{x}$ [29, 7]; that is, we assume that the reconstructed super-resolved correlation image is piecewise constant:

$$\min_{\mathbf{x} \geq \mathbf{0}} \lambda \text{TV}(\mathbf{x}) + \frac{1}{2}\left|\left|\mathbf{R}_y(0) - \sum_{l=1}^{N^2} \mathbf{a}_l \mathbf{a}_l^H x_l\right|\right|_F^2. \tag{F-TV}$$

We follow the definition of the discrete $\text{TV}(\mathbf{x})$ regularization term as described in [2] for both the isotropic and the anisotropic cases. The proximity mapping $\text{prox}_{\alpha\text{TV}}(\mathbf{x})$ does not have a closed form solution in this case. Instead, the authors of [2] proposed to solve $\text{prox}_{\alpha\text{TV}}(\mathbf{x})$ iteratively. The minimizer of (4.1) is the solution to a *denoising* problem with the regularizer

$\alpha g(\cdot)$ on the recovered signal. In particular, $\text{prox}_{\alpha\text{TV}}(\mathbf{x})$ is the denoising solution with TV regularization. Many TV denoising algorithms exist (see, e.g., [29, 7, 24, 13]), and thus any one of them can be used to calculate the proximity mapping iteratively. In particular, we chose to follow the fast TV denoising method suggested in [2] and denoted as Algorithm GP. The algorithm accepts an observed image, a regularization parameter λ which balances between the level of sparsity and compatibility to the observations and a maximal number of iterations N_{max}. The output is a TV denoised image. Thus, as summarized in Algorithm 4.3, each iterative step is composed of a gradient step of f and a subsequent application of Algorithm GP.

Algorithm GP already incorporates a projection onto box constraints, which also includes as a special case the nonnegativity constraints of (F-TV). Hence we have omitted the projection step in Algorithm 4.3.

Algorithm 4.3. Fast proximal gradient descent for (F-TV).

```
Input: $L_f$, $\mathbf{R}_y(0)$, $\lambda > 0$, $K_{\max}$, $N_{\max}$
  Initialize $\mathbf{z}_1 = \mathbf{x}_0 = \mathbf{0}$, $t_1 = 1$, and $k = 1$
  while $k \leq K_{\max}$ or stopping criteria not fulfilled do
    1: $\nabla f(\mathbf{z}_k) = \mathbf{M}\mathbf{z}_k - \mathbf{v}$
    2: $\mathbf{x}_k = \text{GP}(\mathbf{z}_k - \frac{1}{L_f}\nabla f(\mathbf{z}_k), \lambda, N_{\max})$
    3: $t_{k+1} = 0.5(1 + \sqrt{1 + 4t_k^2})$
    4: $\mathbf{z}_{k+1} = \mathbf{x}_k + \frac{t_k - 1}{t_{k+1}}(\mathbf{x}_k - \mathbf{x}_{k-1})$
    5: $k \leftarrow k + 1$
  end while
  return $\mathbf{x}_{K_{\max}}$
```

4.2.2. Analysis type super-resolution imaging. In many scenarios, additional priors can be exploited alongside sparsity, to achieve subwavelength resolution. Examples include wavelet transforms and the discrete cosine transforms (DCTs). In general, the problem we wish to solve is

$$\min_{\mathbf{x}} \lambda||\mathbf{T}^*\mathbf{x}||_1 + \frac{1}{2}\left|\left|\mathbf{R}_y(0) - \sum_{l=1}^{N^2} \mathbf{a}_l\mathbf{a}_l^H x_l\right|\right|_F^2,$$

where $\mathbf{T} \in \mathbb{C}^{M \times N}$ is some known transformation. The prox mapping of the regularization term $||\mathbf{T}^*\mathbf{x}||_1$ does not admit a closed form solution. The authors of [34] suggested to approximate the generally nondifferentiable function $f(\mathbf{x})+g(\mathbf{T}^*\mathbf{x})$ with a surrogate differentiable function, thus alleviating the need to calculate the prox mapping of the nondifferentiable term $g(\mathbf{T}^*\mathbf{x})$. The smooth surrogate function used is the *Moreau envelope of* g [22], given by

$$g_\mu(\mathbf{x}) = \min_{\mathbf{u}}\left\{g(\mathbf{u}) + \frac{1}{2\mu}||\mathbf{u} - \mathbf{x}||_2^2\right\}.$$

We therefore propose a smooth counterpart to (F-LASSO),

$$\text{(F-SM)} \qquad \min_{\mathbf{x}\geq\mathbf{0}} f(\mathbf{x}) + g_\mu(\mathbf{T}^*\mathbf{x}),$$

with $f(\mathbf{x})$ given by (4.3) and $g_\mu(\mathbf{x})$ given by

$$g_\mu(\mathbf{T}^*\mathbf{x}) = \min_{\mathbf{u}} \left\{ \lambda ||\mathbf{u}||_1 + \frac{1}{2\mu}||\mathbf{u} - \mathbf{T}^*\mathbf{x}||_2^2 \right\}.$$

The gradient of (F-SM) is now a combination of the gradients of $f(\mathbf{x})$ and $g_\mu(\mathbf{x})$, with

$$\nabla g_\mu(\mathbf{T}^*\mathbf{x}) = \frac{1}{\mu}\mathbf{T}(\mathbf{T}^*\mathbf{x} - \mathcal{T}_{\lambda\mu}(\mathbf{T}^*\mathbf{x})). \tag{4.6}$$

Using (4.6), we have modified the SFISTA algorithm in [34] to solve (F-SM), as summarized in Algorithm 4.4. Note that the Lipschitz constant of $f(\mathbf{x}) + g_\mu(\mathbf{T}^*\mathbf{x})$ is given by $L_f + \frac{||\mathbf{T}||_2^2}{\mu}$.

Algorithm 4.4. Fast proximal gradient descent for (F-SM).

```
Input: $L_f$, $\mu$, $\mathbf{R}_y(0)$, $\lambda > 0$, $K_{\max}$
  Initialize $\mathbf{z}_1 = \mathbf{x}_0 = \mathbf{0}$, $t_1 = 1$, and $k = 1$
  while $k \leq K_{\max}$ or stopping criteria not fulfilled do
    1: $\nabla f(\mathbf{z}_k) = \mathbf{M}\mathbf{z}_k - \mathbf{v}$
    2: $\nabla g_\mu(\mathbf{T}^*\mathbf{x}_{k-1}) = \frac{1}{\mu}\mathbf{T}(\mathbf{T}^*\mathbf{x}_{k-1} - \mathcal{T}_{\lambda\mu}(\mathbf{T}^*\mathbf{x}_{k-1}))$
    3: $\mathbf{y}_k = \mathbf{z}_k - \frac{1}{L_f}(\nabla f(\mathbf{z}_k) + \nabla g_\mu(\mathbf{T}^*\mathbf{x}_{k-1}))$
    4: $t_{k+1} = 0.5(1 + \sqrt{1 + 4t_k^2})$
    5: $\mathbf{z}_{k+1} = \mathbf{x}_k + \frac{t_k - 1}{t_{k+1}}(\mathbf{x}_k - \mathbf{x}_{k-1}) + \frac{t_k}{t_{k+1}}(\mathbf{y}_k - \mathbf{x}_k)$
    6: $k \leftarrow k + 1$
  end while
  return $\mathbf{x}_{K_{\max}}$
```

5. Efficient implementation.

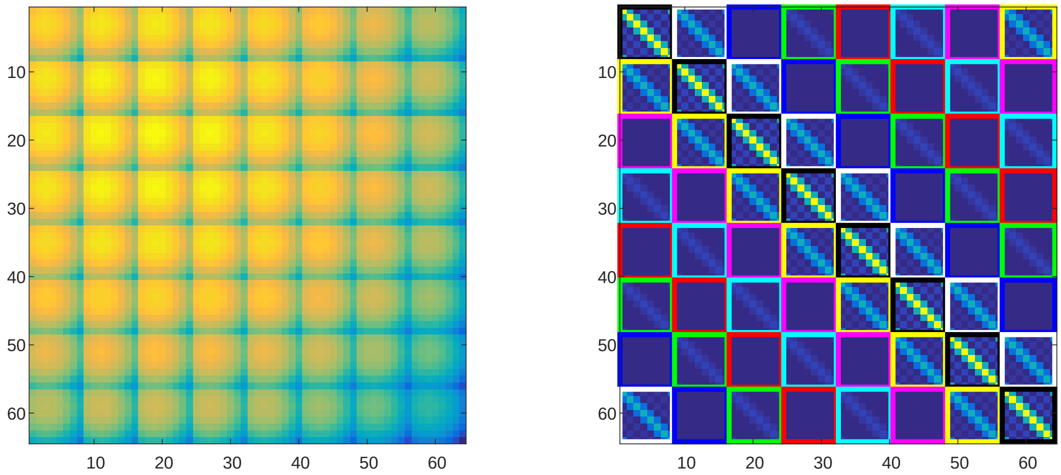
Solving (F-LASSO), (F-TV), and (F-SM) in practice can be very demanding in terms of numerical computations, due to the large dimensions of the reconstructed super-resolved image. Consider, for example, an input movie with frames of size 64×64 pixels and a reconstructed super-resolved image of size 512×512 pixels (an eight-fold increase in the density of the high-resolution grid compared to the low-resolution captured movie). Calculating $\mathbf{R}_y(0)$ yields a covariance data matrix of size $64^2 \times 64^2$, and $\mathbf{R}_s(0)$ is of size $512^2 \times 512^2$ pixels (though in practice it is a diagonal matrix with a diagonal of length 512^2 pixels). The exponential growth in the problem dimensions on the one hand and the diagonal structure of the covariance matrix of the super-resolved image on the other prompts the search for an efficient implementation for Algorithms 4.1–4.4. We now show that by considering the signal model in the spatial frequency domain as in (3.6), an efficient implementation based on FFT and IFFT operations is possible.

5.1. Frequency domain structure.

Recall that

$$\nabla f(\mathbf{x}) = \mathbf{M}\mathbf{x} - \mathbf{v},$$

with $\mathbf{M} = |\mathbf{A}^H\mathbf{A}|^2$, $\mathbf{A} = \mathbf{H}(\mathbf{F}_M \otimes \mathbf{F}_M)$, and $\mathbf{v} = [\mathbf{a}_1^H\mathbf{R}_y(0)\mathbf{a}_1, \ldots, \mathbf{a}_{N^2}^H\mathbf{R}_y(0)\mathbf{a}_{N^2}]^T$. Reconstruction of a super-resolved image of size $N \times N$ dictates that $\mathbf{M}$ will be of size $N^2 \times N^2$. In



(a) Image space domain

(b) Discrete Fourier domain

Figure 1. *Comparison of the structure of* $\mathbf{M}$ *for two possible formulations: image space domain* (a) *and spatial discrete frequency domain* (b).

most cases, it is impossible to store $\mathbf{M}$ in memory or perform matrix-vector multiplications. Instead, we exploit the special structure of $\mathbf{M}$ to achieve efficient matrix-vector operations without explicitly storing it.

Figure 1 illustrates the structure of $\mathbf{M}$ for two formulations: Figure 1(a) shows the structure of $\mathbf{M}$ if we do not consider performing an FFT on (3.3). In this case, the nth column of $\mathbf{A}$ contains the vectorized $M \times M$ PSF centered at the high-resolution pixel ij, $i, j \in [0, \dots, N-1]$. Figure 1(b) illustrates the structure of $\mathbf{M}$ in the spatial-frequency domain, derived from (3.6). In both cases, $\mathbf{M}$ is of size 64×64 pixels and is generated from a PSF of size 4×4 pixels. Both panels represent a reconstruction of a 64×64 super-resolved image.

Figure 1(b) implies that the spatial-frequency formulation of $\mathbf{M}$ has a cyclic structure. This special structure will play a crucial role in our algorithm, as it leads to efficient implementation of matrix-vector multiplications. More specifically, in Appendix A we show that $\mathbf{M}$ is *block circulant with circulant blocks* (BCCB) [17]. Figure 1(b) can be divided into 8×8 blocks (the different blocks are marked with rectangles of different colors to illustrate the block circular structure of the matrix), each block of size 8×8 pixels. As can be seen, $\mathbf{M}$ is circulant with respect to the blocks and each block is also circulant.

Similar to circulant matrices which are diagonalizable by the DFT matrix, BCCB matrices are diagonalizable by the Kronecker product of two DFT matrices of appropriate dimensions. Such structure allows the implementation of a fast matrix-vector multiplication using FFT and inverse FFT operations without the need to store $\mathbf{M}$ in memory. In the following sections, we describe the implementation of (4.4) in detail, by defining several operators which play a crucial role in its calculation.

5.2. Efficient implementation of $\mathbf{Mx}$.

We first define $\mathcal{M}(\mathbf{x})$, which takes $\mathbf{x} \in \mathbb{C}^{N^2}$ and transforms it into a matrix $\mathbf{X} \in \mathbb{C}^{N \times N}$, that is,

$$\mathbf{X} = \mathcal{M}(\mathbf{x}). \tag{5.1}$$

This operation is performed using a columnwise division from top to bottom of $\mathbf{x}$. Upon dividing $\mathbf{x}$ into N subvectors of length N each, the ith column of $\mathbf{X}$ corresponds to the ith subvector of $\mathbf{x}$. Similarly, we denote the vectorization of $\mathbf{X} \in \mathbb{C}^{N \times N}$ which stacks the columns of $\mathbf{X}$ by

$$\mathbf{x} = \mathrm{vec}(\mathbf{X}) = \mathcal{V}(\mathbf{X}). \tag{5.2}$$

Here, $\mathbf{x}$ is a vector of length N^2, whose ith subvector of length N corresponds to the ith column of $\mathbf{X}$.

In Appendix A, we show that $\mathbf{M}$ is an $N^2 \times N^2$ BCCB matrix with blocks of size $N \times N$. It is well known that such a matrix is diagonalizable by the Kronecker product of two discrete $N \times N$ Fourier matrices $\mathbf{F}_2 = \mathbf{F} \otimes \mathbf{F}$ [17], so that

$$\mathbf{M} = \mathbf{F}_2^H \mathbf{\Lambda} \mathbf{F}_2, \tag{5.3}$$

with $\mathbf{\Lambda}$ a diagonal matrix containing the eigenvalues of $\mathbf{M}$ on its diagonal. To compute $\mathbf{Mx}$, we therefore need to calculate the eigenvalues of $\mathbf{M}$ and apply $\mathbf{F}_2$ and $\mathbf{F}_2^H$ on a given vector.

Now,

$$\mathbf{F}_2\mathbf{x} = (\mathbf{F} \otimes \mathbf{F})\mathbf{x} = \mathcal{V}(\mathbf{F}\mathcal{M}(\mathbf{x})\mathbf{F}^T). \tag{5.4}$$

The matrix $\mathbf{F}\mathcal{M}(\mathbf{x})\mathbf{F}^T$ corresponds to applying the FFT on each column of $\mathcal{M}(\mathbf{x})$ and then again over the rows of the result. In MATLAB, $\mathbf{F}_2\mathbf{x}$ is easily performed by reshaping $\mathbf{x}$ to $\mathcal{M}(\mathbf{x})$, applying the `fft2` command on $\mathcal{M}(\mathbf{x})$, and vectorizing the result. Similarly, calculation of the 2D inverse FFT of an $N \times N$ matrix $\mathbf{X}_f$ is equivalent to $\frac{1}{N^2}\mathbf{F}^H\mathbf{X}_f\bar{\mathbf{F}}$ and is easily implemented in MATLAB with the `ifft2` command. To compute the eigenvalues of $\mathbf{M}$ efficiently, we first need to be able to compute $\mathbf{Ax}$ and $\mathbf{A}^H\mathbf{x}$ for some $\mathbf{x} \in \mathbb{C}^{N^2}$.

5.2.1. Calculation of $\mathbf{Ax}$.

Recall that

$$\mathbf{A} = \mathbf{H}(\mathbf{F}_M \otimes \mathbf{F}_M),$$

where $\mathbf{F}_M \in \mathbb{C}^{M \times N}$ denotes a partial Fourier matrix, corresponding to the low-pass values of a full $N \times N$ Fourier matrix. The operator $\mathbf{Ax}$ corresponds to taking $\mathbf{X} = \mathcal{M}(\mathbf{x})$, calculating $\mathbf{F}_M\mathbf{X}\mathbf{F}_M^T$, vectorizing the result, and multiplying by $\mathbf{H}$. Denote

$$\mathcal{F}_{M_2}(\mathbf{X}) = \mathbf{F}_M\mathbf{X}\mathbf{F}_M^T. \tag{5.5}$$

The application of $\mathbf{F}_M$ on an $N \times N$ matrix $\mathbf{X}$ can be implemented by computing an FFT on each column of $\mathbf{X}$ and taking only the first M rows of the result. Similarly, calculation of $\mathbf{X}\mathbf{F}_M^T = (\mathbf{F}_M\mathbf{X}^T)^T$ is achieved by performing an FFT on each row of $\mathbf{X}$, taking the first M rows of the result, and performing the transpose operation.

Equation (5.5) implements a partial 2D-FFT operation on $\mathbf{X}$, where the full 2D-FFT operation is written as $\mathbf{F}\mathbf{X}\mathbf{F}^T$ with an $N \times N$ discrete Fourier matrix $\mathbf{F}$. The multiplication $\mathbf{Ax}$ can then be summarized as follows:

$$\mathbf{Ax} = \mathbf{H} \cdot \mathcal{V}(\mathcal{F}_{M_2}(\mathcal{M}(\mathbf{x}))). \tag{5.6}$$

Since $\mathbf{H}$ is a diagonal matrix, the matrix-vector multiplication in (5.6) corresponds simply to multiplying the diagonal of $\mathbf{H}$ and the corresponding vector. If instead of a vector $\mathbf{x}$ we perform $\mathbf{AZ}$ on some matrix $\mathbf{Z} \in \mathbb{C}^{N^2 \times L}$, then the operation is performed on each column of $\mathbf{Z}$.

5.2.2. Calculation of $\mathbf{A}^H\mathbf{x}$.

For $\mathbf{x} \in \mathbb{C}^{M^2}$,

$$\mathbf{A}^H\mathbf{x} = (\mathbf{F}_M^H \otimes \mathbf{F}_M^H)\mathbf{H}^H\mathbf{x} = (\mathbf{F}_M^H \otimes \mathbf{F}_M^H)\mathbf{z}.$$

Upon reformulating $\mathbf{z}$ as an $M \times M$ matrix $\mathbf{Z} = \mathcal{M}(\mathbf{z})$, we have

$$\mathcal{M}(\mathbf{A}^H\mathbf{x}) = \mathbf{F}_M^H\mathbf{Z}\bar{\mathbf{F}}_M.$$

Since $\mathbf{F}^H = N\mathbf{F}^{-1}$, $\mathbf{F}_M^H\mathbf{Z}$ corresponds to performing an inverse FFT on the zero-padded columns of $\mathbf{Z}$ and multiplying by N. We denote the result as $\mathbf{Y}$. Next, notice that $\mathbf{Y}\bar{\mathbf{F}}_M = (\mathbf{F}_M^T\mathbf{Y}^H)^H$. Since the DFT matrix is a symmetric matrix, the second step involves computing an FFT on the zero-padded columns of $\mathbf{Y}^H$ and finally taking the Hermitian operation. By denoting $\mathcal{F}_{M_2}^H(\mathbf{X}) = \mathbf{F}_M^H\mathbf{X}\bar{\mathbf{F}}_M$, we can write

$$\mathbf{A}^H\mathbf{x} = \mathcal{V}(\mathcal{F}_{M_2}^H(\mathcal{M}(\mathbf{H}^H\mathbf{x}))). \tag{5.7}$$

If instead of a vector $\mathbf{x}$ we perform $\mathbf{A}^H(\mathbf{Q})$ on some matrix $\mathbf{Q} \in \mathbb{C}^{M^2 \times L}$, then the operation is performed on each column of $\mathbf{Q}$.

5.2.3. Calculation of the eigenvalues of $\mathbf{M}$.

To calculate the eigenvalues of $\mathbf{M}$, denoted by $\boldsymbol{\lambda}$, note that from (5.3), $\frac{1}{N}\mathbf{F}_2\mathbf{M} = \boldsymbol{\Lambda}\mathbf{F}_2$, which implies that $\frac{1}{N}\mathbf{F}_2\mathbf{m}_1 = \boldsymbol{\Lambda}\mathbf{f}_1$, with $\mathbf{m}_1$ and $\mathbf{f}_1$ being the first columns of $\mathbf{M}$ and $\mathbf{F}_2$, respectively. Since $\mathbf{f}_1$ is a vector of ones, we have

$$\frac{1}{N}\mathbf{F}_2\mathbf{m}_1 = \boldsymbol{\lambda},$$

with $\boldsymbol{\lambda} = \operatorname{diag}\{\boldsymbol{\Lambda}\}$. To compute $\mathbf{m}_1$, we note that since $\mathbf{M} = |\mathbf{A}^H\mathbf{A}|^2$, $\mathbf{m}_1 = |\mathbf{A}^H\mathbf{a}_1|^2$, where $\mathbf{a}_1$ is the first column of $\mathbf{A}$. From the definition of $\mathbf{A}$, $\mathbf{a}_1 = \mathbf{h}$, where $\mathbf{h} = \operatorname{diag}\{\mathbf{H}\}$, and therefore $\mathbf{m}_1 = \left|(\mathbf{F}_M^H \otimes \mathbf{F}_M^H)|\mathbf{h}|^2\right|^2$. In MATLAB, this can be implemented using `fft` / `ifft` operations, as noted by the first two steps of Algorithm 5.1. By denoting the $M \times M$ DFT of the PSF as $\mathbf{U}$, it follows straightforwardly that $\mathcal{M}(\mathbf{m}_1) = \mathbf{Z} = \left|\mathcal{F}_{M_2}^H(|\mathbf{U}|^2)\right|^2$, where the operation $|\cdot|^2$ is performed elementwise. After the calculation of $\mathbf{m}_1$, finding $\boldsymbol{\lambda}$ is straightforward since

$$\mathcal{M}(\boldsymbol{\lambda}) = \mathbf{B} = \mathbf{F}\mathbf{Z}\mathbf{F}^T,$$

which can be computed using the 2D-FFT.

Algorithm 5.1. Calculation of $\mathbf{Mx}$.

```
Input: The DFT of the PSF $\mathbf{U}$ and $\mathbf{x}$
  Eigenvalues calculation:
    1: Calculate $\mathbf{T} = N\texttt{ifft}\{|\mathbf{U}|^2\}$ of length $N$
    2: Calculate $\mathbf{E} = \texttt{fft}\{\mathbf{T}^H\}$ of length $N$
    3: Eigenvalues calculation $\mathbf{B} = \mathbf{F}|\mathbf{E}^H|^2\mathbf{F}^T$ using fft2
  Application of Mx:
    4: Calculate $\mathbf{Q} = \mathbf{B} \odot (\mathbf{F}\mathcal{M}(\mathbf{x})\mathbf{F}^T)$
    5: Calculate $\mathbf{Y} = \frac{1}{N^2}\mathbf{F}^H\mathbf{Q}\bar{\mathbf{F}}$ using ifft2
  return $\mathcal{V}(\mathbf{Y}) = \mathbf{Mx}$.
```

We can summarize the application of $\mathbf{M}$ on $\mathbf{x}$ in Algorithm 5.1, with $\mathbf{A} \odot \mathbf{B}$ representing the Hadamard elementwise product of two matrices $\mathbf{A}$ and $\mathbf{B}$ and with the `fft` / `ifft` operations performed columnwise.

Algorithms 4.1–4.4 require the Lipschitz constant L_f of $\mathbf{M}$. This constant is readily given by noting that

$$L_f = ||\mathbf{M}||_2 = \max_i \lambda_i, \quad i = 1, \ldots, N^2,$$

with λ_i being the ith entry of $\boldsymbol{\lambda}$. The value $\max_i \lambda_i$ is calculated as part of Algorithm 5.1 and is given by

$$L_f = \max_{i,j} b_{ij}, \quad i, j = 1, \ldots, N,$$

where b_{ij} is the ijth entry of $\mathbf{B}$ from line 2 in Algorithm 5.1.

5.3. Efficient calculation of v.

The vector $\mathbf{v}$ in (4.4) is the input data to Algorithms 4.1–4.4. Its ith element is given by

$$v_i = \mathbf{a}_i^H \mathbf{R}_y(0)\mathbf{a}_i, \quad i = 0, \ldots, N^2 - 1,$$

with $\mathbf{a}_i$ representing the ith column of $\mathbf{A}$. Since $\mathbf{v}$ is an N^2 long vector, calculating its entries strictly by applying $\mathbf{A}^H$ and $\mathbf{A}$ on $\mathbf{R}_y(0)$ and taking the resulting diagonal is impractical as N increases. Instead, it is possible to calculate its entries in two steps as follows.

The application of $\mathbf{a}_i$ on a matrix is very similar to the application of $\mathbf{A}$, only for a specific index i. We may write $\mathbf{a}_i$ more explicitly as

$$\mathbf{a}_i = \mathbf{H}(\mathbf{f}_{k_i} \otimes \mathbf{f}_{l_i}),$$

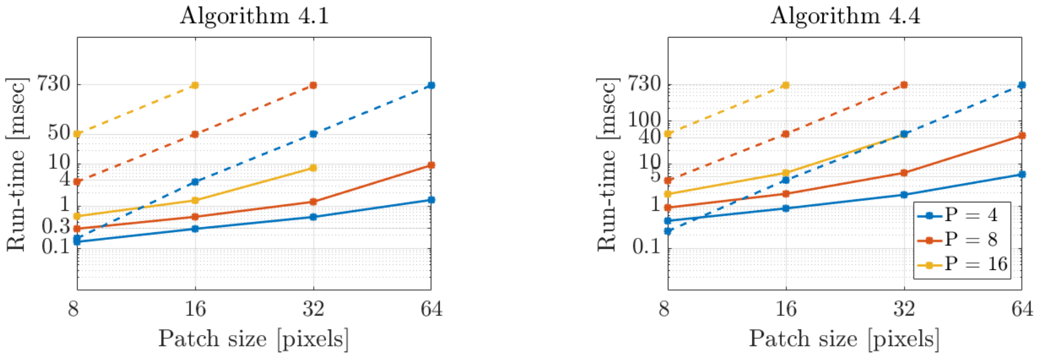
with $k_i = \lfloor \frac{i}{N} \rfloor$ and $l_i = i \bmod N$. By using the previously defined operations, $\mathbf{v}$ can be calculated as summarized in Algorithm 5.2. This calculation needs to be performed only once, at the beginning of Algorithms 4.1–4.4.

5.4. Algorithm run-time.

In this section, we compare the average run-time of our Fourier based formulation, i.e., using the structure depicted in Figure 1(b) against the spatial domain formulation (Figure 1(a)). Figure 2 shows the average run-time for a single iteration of Algorithm 4.1 (left panel) and Algorithm 4.4 (right panel), performed on a 64GB RAM, Intel

Algorithm 5.2. Calculation of $\mathbf{v}$.

```
Input: $\mathbf{H}$ and $\mathbf{R}_y(0)$
  Calculation of $\mathbf{Z}^H = \mathbf{A}^H\mathbf{R}_y(0)$:
    1: Calculate $\mathbf{Q} = \mathbf{H}^H\mathbf{R}_y(0)$
    For each column of $\mathbf{Q}$, $\mathbf{q}_i$, $i = 0, \dots, M^2 - 1$:
      2: Calculate $\mathbf{T}_i = N\texttt{ifft}\{\mathcal{M}(\mathbf{q}_i)\}$ of length $N$
      3: Calculate $\mathbf{E}_i = \texttt{fft}\{\mathbf{T}_i^H\}$ of length $N$
      4: Take the $i$th column of $\mathbf{Z}^H$ as $\mathcal{V}(\mathbf{E}_i^H)$
  Calculation of each element in $\mathbf{v}$:
    For each $i = 0, \dots, N^2 - 1$:
      5: $\mathbf{B} = \mathbf{F}^H\mathcal{M}(\mathbf{H}^H\mathbf{z}_i)$, with $\mathbf{z}_i$ the $i$th column of $\mathbf{Z}$
      6: Calculate $\mathbf{u} = \mathbf{F}\mathbf{b}_{l_i}$, with $\mathbf{b}_{l_i}$ the $l_i$th row of $\mathbf{B}$
      7: Take $v_i = u_{k_i}$, the $k_i$ entry of $\mathbf{u}$.
  return v.
```



(a) l_1 based recovery.

(b) Wavelet based recovery.

Figure 2. Algorithm run-time as a function of patch size. *Left panel shows run-time for a single iteration of Algorithm* 4.1*. Right panel shows run-time for a single iteration of Algorithm* 4.4 *(Daubechies wavelet filter with eight taps and decomposition level of* 2*). Solid lines correspond to the frequency domain formulation (exploiting the BCCB structure of* $\mathbf{A}^T\mathbf{A}$*), while the dashed curves correspond to the spatial domain formulation. All values were averaged over* 2000 *iterations. The vertical axis is in logarithmic scale, but the values are given in linear scale.*

i7-5960X@3GHz machine and implemented in MATLAB. Each value is the average over 2000 runs. The eigenvalues of $\mathbf{M}$ and $\mathbf{v}$ are calculated a priori.

As expected, run-time increases as patch size increases and as the value of P increases. All curves are roughly linear, indicating an exponential growth in complexity as patch size increases, as the vertical axis is displayed in logarithmic scale (numerical values are given in linear scale). For the frequency domain formulation (solid lines), Figure 2 shows that the execution time of each iteration is very fast for patches of sizes 8 × 8, 16 × 16, and 32 × 32. On the other hand, run-time curves for the spatial domain formulation (dashed curves) are between one to two orders of magnitude higher, especially for higher values of P, such as 8 and 16. The value of P needs to be increased, as smaller features are needed to be resolved.

These curves clearly motivate the use of our frequency domain formulation. Moreover, it is recommended to divide the entire field of view to patches of 8–32 pixels and process each patch independently. Since each patch is processed independently, the entire computational process can be parallelized for additional gain in efficiency.

6. Simulations.

In this section, we provide further examples and characterization of SPARCOM. We start by providing an additional simulation to the results given in [33], showing the ability of SPARCOM in recovering fine features absent in the diffraction limited movie, as well as providing additional comparisons to an improved SOFI formulation, termed balanced SOFI (bSOFI) [14], and high emitter density STORM, implemented with the freely available ThunderSTORM software [25]. This subdiffraction object and its corresponding SPARCOM recovery serve as a basis for an additional sensitivity analysis of SPARCOM to inexact knowledge of the PSF, presented in Figures 7 and 8. The next simulation presents the key advantages of SPARCOM in scenarios where assuming sparsity in domains other than the image domain leads to improved recovery results. We finish by providing experimental reconstruction results of SPARCOM, with our general super-resolution framework. These aspects complement the demonstration and analysis performed in [33], thus providing a more comprehensive understanding of SPARCOM and its applications.

6.1. Comparison of different super-resolution methods.

We numerically simulated a movie of subwavelength features over 1000 frames, contaminated by additive Gaussian noise with SNR = 14.95dB,

$$\mathrm{SNR} = 20 \cdot \log_{10} \frac{||\mathbf{Y}_{\mathrm{movie}}||_F}{||\mathbf{N}_{\mathrm{movie}}||_F},$$

where $\mathbf{Y}_{\mathrm{movie}}$ is an $M^2 \times T$ matrix representing the entire blurred movie (each movie frame is column stacked as a single column in $\mathbf{Y}_{\mathrm{movie}}$) and $\mathbf{N}_{\mathrm{movie}}$ is the added noise to all the frames (same dimensions as $\mathbf{Y}_{\mathrm{movie}}$). The movie also includes the simulation of out-of-focus filaments, which simulate unwanted fluorescence from objects outside the focal plane. Thus, they appear much wider than the in-focus simulated filaments. For both the in-focus and out-of-focus objects, we used the same Gaussian PSF, generated using the freely available PSF generator [18, 19], but with focal depths of $0nm$ and $1\mu m$, respectively.

In Figure 3(a), we show the simulated ground truth of the image with subwavelength features of size 512×512 pixels. The imaging wavelength is $800nm$ with a numerical aperture of 1.4. We simulated two movies. The first is composed of 1000 high emitter density frames, while the second is composed of 5000 low emitter density frames of the same features.

Figure 3(b) illustrates a single frame from the high density movie (each frame size is 64×64 pixels, and the pixel size corresponds to $160nm$), while Figure 3(c) shows the diffraction limited image (a sum of all 1000 frames).

Figure 3(d) shows a smoothed ThunderSTORM [25] reconstruction (freely available code) from the low emitter density movie. This image serves as a reference for the best possible reconstruction, when there are no temporal considerations. On the other hand, Figure 3(e) depicts smoothed ThunderSTORM reconstruction, performed with the high density movie of 1000 frames. Since the ground truth is of size 512×512 pixels, the raw localizations image was resized to a 512×512 image and smoothed with a Gaussian kernel. Figures 3(f) and 3(g) show the second and fourth order SOFI images, respectively (absolute values, zero time-lag).

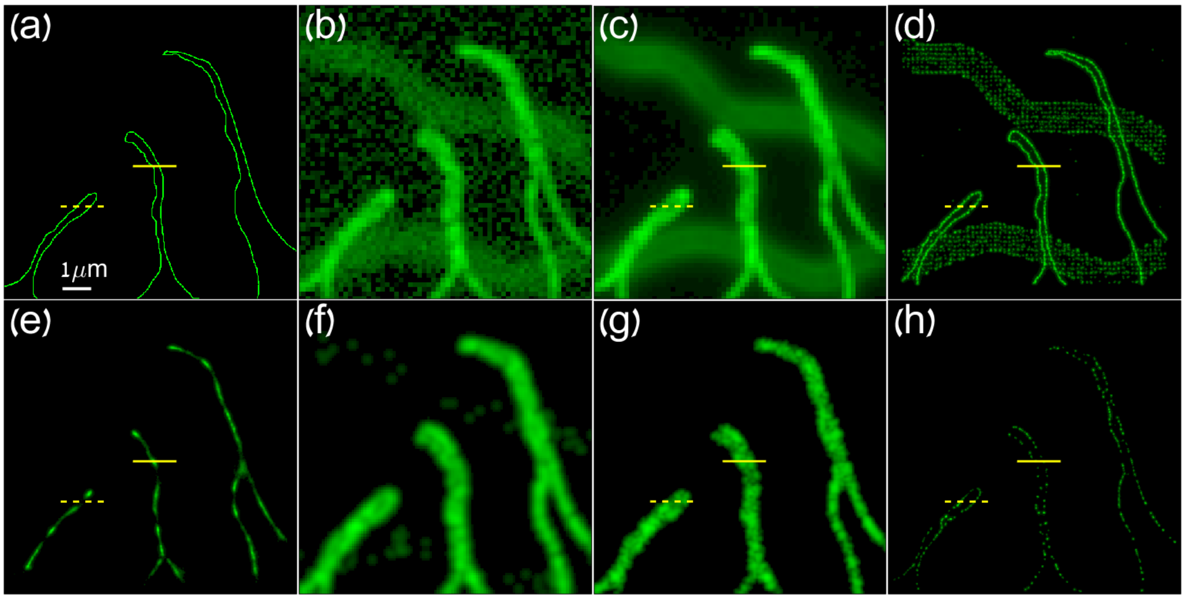


Figure 3. Reconstruction performance comparison of different methods. Upper row: (a) *Ground truth: high resolution image of simulated subwavelength features.* (b) *Single diffraction limited frame from the movie, created by convolving the movie of fluctuating point emitters according to the locations in* (a) *with the PSF and adding Gaussian noise.* (c) *Diffraction-limited image, taken by averaging all the frames in the movie.* (d) *ThunderSTORM recovery from* 5000 *low density frames.* Lower row: recovered images from a noisy sequence of 1000 frames. (e) *Smoothed ThunderSTORM.* (f) *Correlations SOFI (zero time-lag).* (g) *fourth order SOFI (in absolute value, zero time-lag).* (h) *SPARCOM recovery.*

SOFI reconstructions were performed using the freely available code of bSOFI [14], which also includes a Richardson–Lucy deconvolution step with the discretized PSF used in our method. Last, Figure 3(h) displays the SPARCOM reconstruction (512×512 pixels) after smoothing with the same kernel used in Figures 3(d) and 3(e). Reconstruction was performed over 2000 iterations and with $\lambda = 10^{-3}$.

Note that the SOFI reconstructions do not compare in resolution to the ThunderSTORM and SPARCOM recoveries. This additional comparison shows that, even when considering more advanced implementations of SOFI, such as bSOFI, the resolution increase does not match that of SPARCOM. Furthermore, it is evident that the SPARCOM recovery (Figure 3(h)) detects the "cavities" within the hollow features, similar to low density ThunderSTORM (Figure 3(d)). When high emitter density is used, Figure 3(e) illustrates that ThunderSTORM recovery fails and no clear depiction of these features is possible.

In order to further quantify the performance of SPARCOM, Figure 4 presents selected intensity cross-sections along two lines. In both profiles (solid and dashed yellow lines in the panels of Figure 3), several observations can be made. First, there is a good match between the locations and width of the SPARCOM (solid red) and low density ThunderSTORM (dash dot green) recoveries with the ground truth (dashed blue), indicating that SPARCOM achieves spatial resolution comparable to that of ThunderSTORM when there are no tempo-

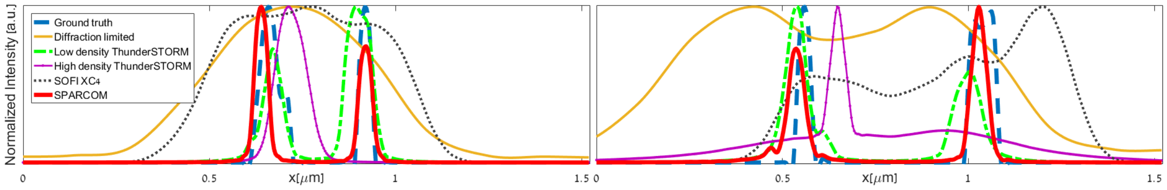


Figure 4. *Normalized cross-sections along the solid yellow line (left) and the dashed yellow line (right) of Figure* 3*, comparing the ground truth (dashed blue, Figure* 3(a)*), diffraction-limited image (solid yellow, Figure* 3(c)*), ThunderSTORM using* 5000 *low density frames (dash dot green, Figure* 3(d)*), ThunderSTORM using* 1000 *high density frames (solid thin purple, Figure* 3(e)*), fourth order SOFI (dot black, Figure* 3(g)*), and SPARCOM (solid red, Figure* 3(h)*).*

ral constraints. Second, if temporal resolution is critical, i.e., it is essential to capture only a small number of high emitter density frames, then high density ThunderSTORM fails (solid thin purple), detecting only a single, misplaced peak, compared with the two peaks of the ground truth. Finally, in this scenario, SOFI reconstruction (dot black) failed in achieving good recovery.

Figures 3 and 4 demonstrate that sparse recovery in the correlation domain achieves increased resolution with increased temporal resolution (five times in this example) and detects the cavities within the subwavelength features which are absent in the low-resolution movie, high density ThunderSTORM, and SOFI reconstructions. This simulation adds upon the simulations presented in [33], by comparing SPARCOM with bSOFI, which provides additional steps to the original SOFI scheme, such as a deconvolution step, as well as demonstrating the disadvantages of localization based methods in the high density scenario (which can lead to a reduction in the total acquisition time).

6.2. Super-resolution under different regularizers. Next, we tested our more general framework for super-resolution reconstruction. We simulated a movie of thick subdiffraction filaments over 1000 frames with Gaussian noise (SNR $= 17.72$dB). In Figure 5(a), we show the simulated ground truth of size 512×512 pixels. The imaging wavelength is $800nm$ with a numerical aperture of 1.4. Figure 5(b) shows the positions of the emitters for the first frame in the movie, while Figure 5(c) shows the diffraction limited image (a sum of all 1000 frames). Figure 5(d) shows a single frame from the simulated movie, where each frame size is 64×64 pixels and the pixel size corresponds to $160nm$. We used the same PSF as before.

Figure 5(e) shows reconstruction in the 2D wavelet domain, while Figure 5(f) considers reconstruction under the assumption of a sparse distribution of molecules (Algorithm 4.1, 2000 iterations, $\lambda = 10^{-4}$, and smoothed with the same kernel as before). For the wavelet reconstruction, we used Algorithm 4.4 with 2000 iterations, $\lambda = 8 \cdot 10^{-4}$, and $\mu = 10^{-5}$. The wavelet and inverse-wavelet transform were implemented using the *Rice Wavelet Toolbox*[2] V.3, with two decomposition levels and a Daubechies scaling filter of 32 taps produced by the function `daubcqf` [9]. Figure 5(g) considers reconstruction in the 2D-DCT domain, while Figure 5(h) shows reconstruction under an isotropic TV assumption. The DCT reconstruction used Algorithm 4.4 with 2000 iterations, $\lambda = 5 \cdot 10^{-4}$, and $\mu = 10^{-5}$, and the isotropic total

[2] https://github.com/ricedsp/rwt

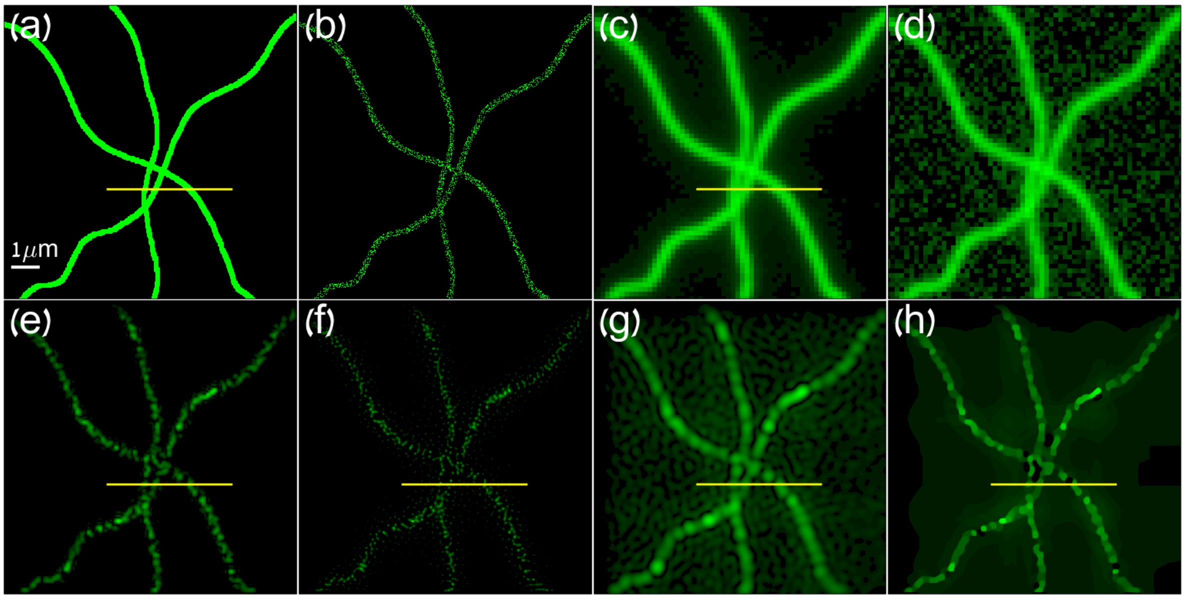


Figure 5. Regularized super-resolution. Upper row: unprocessed data. (a) *Ground truth: high-resolution simulated image.* (b) *Positions of emitters in the first frame.* (c) *Diffraction-limited image.* (d) *Single diffraction limited frame.* Lower row: recovered images from a noisy sequence of 1000 frames. (e) *2D wavelet reconstruction.* (f) l_1 *reconstruction.* (g) *2D-DCT reconstruction.* (h) *Isotropic TV reconstruction.*

variation recovery was performed using Algorithm 4.3 with 500 iterations and $\lambda = 10^{-4}$. Each denoising step (Algorithm GP from [2]) used 100 iterations.

In Figure 6, we show the normalized intensity profiles of the yellow lines in Figure 5, comparing the reconstruction performance of the various algorithms used previously. It is clear that the diffraction limited profile (dashed yellow) conceals two filaments (solid blue curve), which are distinguishable in all methods. However, the l_1 based reconstruction (i.e., sparsity assumption in the positions of the emitters) results in artifacts which give the reconstructed image a grainy appearance and do not capture the true width of the filaments. On the other hand, the wavelet and TV based images show the filaments' width more precisely, while the DCT recovers a blurrier image of them.

Though this example is artificial, it serves to demonstrate that in some cases assuming sparsity in domains other than the original sparsity assumption may help produce reconstructions which are more faithful to the desired object and have smoother textures.

6.3. Sensitivity of reconstruction to the PSF. Knowledge of the PSF is crucial for all the algorithms presented in this work. In practice, this knowledge is often imperfect and the PSF is usually estimated from the data [11] or from a specific experiment used to determine it [28]. When measuring the PSF of the microscope in an experiment, the position of the emitters or beads may not be exactly in the focal-plane but rather a few hundreds of nanometers above or below it. Hence, we tested the reconstruction performance of Algorithm 4.1 when used with different out-of-focus PSFs, to assess its sensitivity to inexact knowledge of the PSF. We used

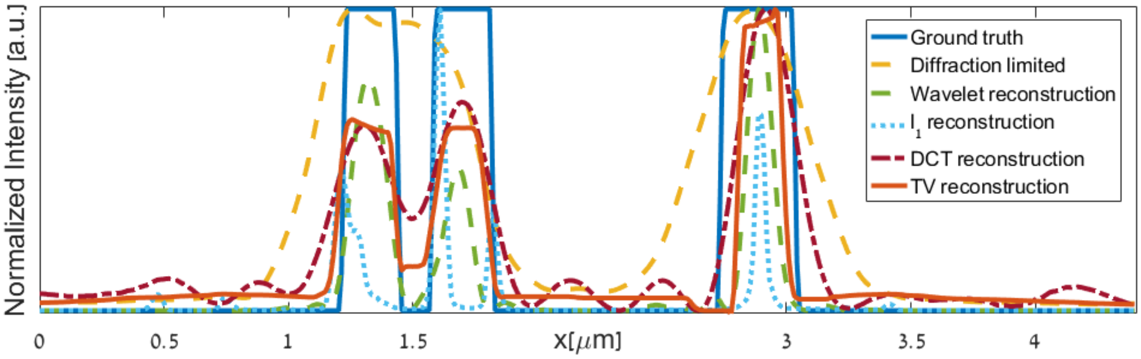


Figure 6. *Normalized cross-sections along the solid yellow line of Figure 5, comparing the ground truth (solid blue, Figure 5(a)), diffraction-limited image (dashed yellow, Figure 3(c)), 2D wavelet reconstruction (dashed green, Figure 3(e)), l_1 reconstruction (blue dot, Figure 3(f)), 2D-DCT reconstruction (dash-dot red, Figure 3(g)), and isotropic TV reconstruction (orange solid, Figure 3(h)).*

the same simulated data (and same SNR) as in Figure 3 (which was generated with the PSF in Figure 7(d)) and simulated several PSFs measured at different distances from the focal plane. All reconstructions were performed over 2000 iterations and with $\lambda = 10^{-3}$. Figures 7(a)–(d) illustrate the different (binned) PSFs with varying distances from the focal plane, $z = 750, 500, 250, 0$nm, respectively. Each PSF was generated using the PSF generator, and for $z = 500$nm, the PSF width is twice the width of the in-focus PSF ($z = 0$nm).

Figures 7(e)–(h) show the reconstruction results when used with the PSFs in Figures 7(a)–(d), respectively, while Figures 8(a)–(d) show a zoom-in on the area inside the yellow rectangles in Figures 7(e)–(h). It is clear from Figures 7 and 8 that as the PSF widens, reconstruction quality degrades, but similar reconstruction results in this example are given even for a PSF that is twice as wide ($z = 500$nm) as the in-focus PSF ($z = 0$nm). This observation suggests that SPARCOM is fairly robust to inexact knowledge of the PSF, and deviations in its width (which correspond to deviations of several hundreds of nanometers in the axial depth of the PSF) can still lead to good reconstructions.

7. Experimental results.

In this section, we further assess SPARCOM reconstructions on experimental datasets, under sparsity assumptions in different domains. In this example, applying Algorithm 4.1 did not yield meaningful reconstruction, due to the width of the subdiffraction features. Thus, we consider reconstruction under our generalized framework and show that indeed performing sparse recovery with our generalized framework can clearly resolve subdiffraction features from high density movies. The dataset is freely available [21] and consists of 160 high density frames of endoplasmic reticulum (ER) protein, fused to tdEos in a U2OS cell. The experimental setup consists of an imaging wavelength of 561nm, numerical aperture of 1.3, and pixel size of 100nm. Each frame is 64×64 pixels. The PSF was generated based on these acquisition parameters. We set $P = 8$ and apply SPARCOM to this dataset, using the reconstruction algorithms presented in section 4.

Panel (a) of Figure 9 shows the diffraction limited image, while panels (b)–(d) show SPARCOM recoveries under the wavelet domain ($\lambda = 2 \cdot 10^{-3}, \mu = 10^{-5}$), DCT ($\lambda =, \mu =$

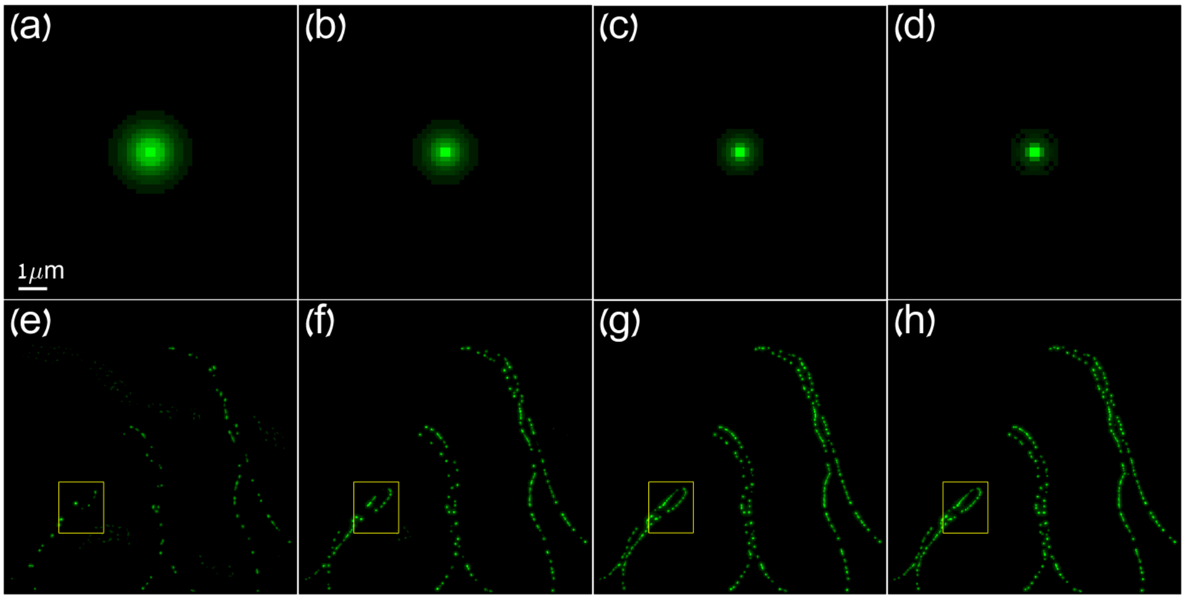


Figure 7. Reconstruction with out-of-focus PSFs. Upper row: (a)–(d) *show simulated PSFs at distances* $z = 750, 500, 250, 0nm$ *from the focal plane, respectively. The captured movie was generated with the PSF in* (d). Lower row: (e)–(h) *show the reconstructed images using Algorithm* 4.1 *for each of the PSFs* (a)–(d), *respectively. Yellow rectangles represent a magnified area shown in Figure* 8.

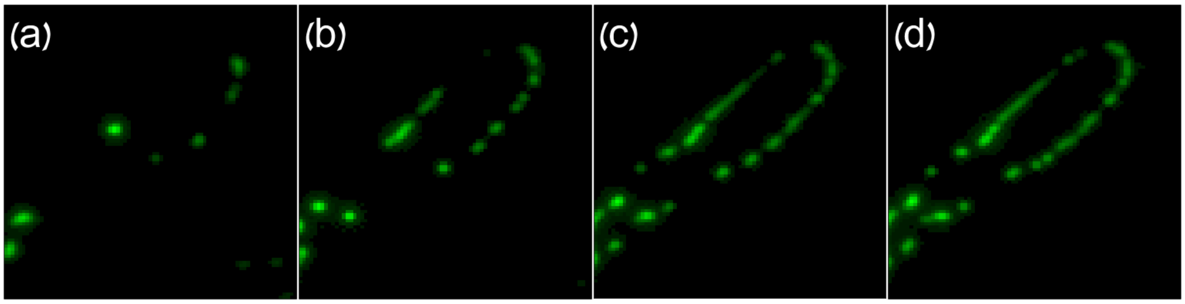


Figure 8. (a)–(d) *show a zoom-in on the yellow rectangular windows in Figures* 7(e)–(h).

$4 \cdot 10^{-4}, \mu = 10^{-5}$), and isotropic TV ($\lambda =, \mu = 10^{-4}$), respectively. All recoveries were performed with 2000 iterations, and wavelet recovery was performed using the *Rice Wavelet Toolbox* V.3 with two decomposition levels and a Daubechies scaling filter of 32 taps. Yellow insets indicate corresponding enlarged regions in the upper left corner of each panel.

Considering panels (b)–(d), the wavelet based reconstruction seems to be the sharpest, presenting subdiffraction features which are absent in panel (a), while also depicting smooth and continuous features. DCT reconstruction seems to also present a smooth, albeit blurrier recovery. Finally, the TV-based image seems to also produce a consistent recovery, but with poorer resolution compared with the wavelet based recovery.

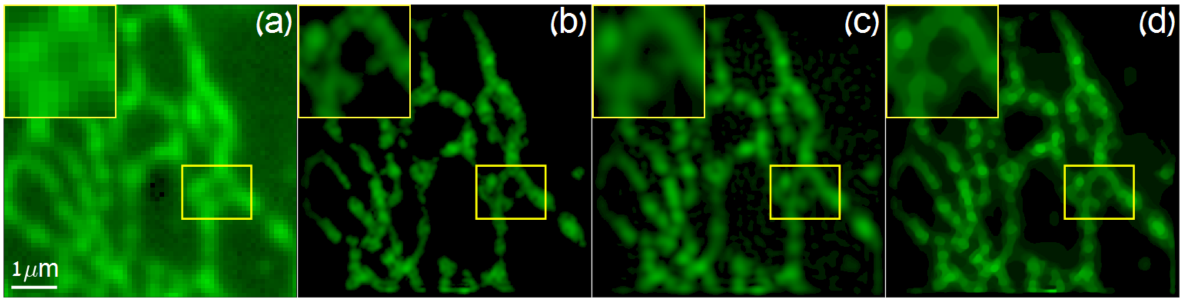


Figure 9. Experimental dataset results. (a) *Diffraction limited image composed of* 160 *frames.* (b) *SPARCOM reconstruction in the wavelet domain.* (c) *SPARCOM recovery in the DCT domain.* (d) *SPARCOM reconstruction in the TV domain. Yellow insets indicate corresponding enlarged regions in the upper left corner of each panel.*

The enlarged regions show that all regularizers, but especially the wavelet regularizer, can resolve subdiffraction features which are completely absent in the diffraction limited image (left portion of the insets), while preserving a smooth depiction of the objects. This demonstration shows the benefit of performing recovery in additional domains, such as the wavelet domain, especially if the recovered features have an intricate morphology, of varying width.

8. Discussion and conclusions.

In this paper, SPARCOM, a method for super-resolution fluorescence microscopy with short integration time and comparable spatial resolution to state-of-the-art methods, is further described and extended. By relying on sparse recovery and the uncorrelated emissions of fluorescent emitters, SPARCOM manages to reduce the total integration time by several orders of magnitudes compared to commonly practiced methods. We developed a thorough and detailed mathematical formulation of our method and showed that considering reconstruction in the sampled Fourier domain results in a special structure of the gradient, which leads to a numerically efficient implementation relying on FFT operations. Moreover, we explored additional extensions of SPARCOM to scenarios in which assuming sparsity in domains other than simply the locations of the emitters leads to better recovery results.

We conclude the paper by addressing the question of stable recovery of emitters in the noiseless and noisy cases from a theoretical point of view. The authors of [23] considered stable recovery of positive point sources from low-pass measurements. By solving a simple convex optimization problem in the noiseless case, they show that a sufficient condition for recovery is that $||\mathbf{x}||_0 < M/2$, where M is the number of low-pass measurements, without any regard to where the sources are on the high-resolution grid. That is, perfect recovery is possible, although the measurement matrix $\mathbf{A}$ is highly coherent. In the presence of noise, such a condition is not sufficient, and it is important to know how regular the positions of the emitters are, that is, how many spikes are clustered together within a resolution cell (see Definition 1 in [23] for a proper definition of regularity). The bounds given in [23] are with respect to specific theoretical PSFs. For example, the authors of [3] found that the length of the resolution cell in the case of a 1D Gaussian kernel is $1.1/\sigma$, where σ is the standard deviation of the Gaussian.

Since SPARCOM recovers the variance of each emitter, this scenario deals with the recovery of positive quantities, where now the desired signal is the variance of the emitters, and not their actual intensities. Thus, similar to the work of [26], in the noiseless case we theoretically expect it to be possible to recover up to $O(M^2)$ emitter locations instead of $M/2$ for the same number of measurements.

Appendix A. Proof of matrix $\mathbf{M}$ being BCCB. We begin by defining BCCB matrices [17, 16].

Definition 1. *A matrix $\mathbf{C} \in \mathbb{C}^{N\times N}$ is said to be circulant if*

$$C_{ij} = c_{(j-i) \bmod N} \quad \forall i, j = 1, \dots, N,$$

for some $c_{(\cdot)} \in \mathbb{C}$, where C_{ij} is the ijth entry of $\mathbf{C}$.

Definition 2. *A matrix is said to be BCCB if it can be divided into $N \times N$ square blocks, where each block is circulant and the matrix is circulant with respect to its blocks, e.g.,*

$$\text{(A.1)} \qquad \mathbf{B} = \begin{bmatrix} \mathbf{C}_0 & \mathbf{C}_{N-1} & \dots & \mathbf{C}_1 \\ \mathbf{C}_1 & \mathbf{C}_0 & \dots & \mathbf{C}_2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{N-1} & \mathbf{C}_{N-2} & \dots & \mathbf{C}_0 \end{bmatrix},$$

where each $\mathbf{C}_i$, $i = 0, \dots, N-1$, is an $N \times N$ circulant matrix.

A circulant matrix of size $N \times N$ is completely defined by its first column vector and hence has N degrees of freedom. Similarly, a BCCB matrix of size $N^2 \times N^2$ is completely defined by its first column and has N^2 degrees of freedom. Denote the first column of $\mathbf{B}$ as $\mathbf{b} \in \mathbb{C}^{N^2}$, such that its ith element is denoted by b_i. In the following proof, we will show that the general element of $\mathbf{B}$ (and $\mathbf{M}$) can be represented by two independent sets of indices, the first corresponding to block circularity between $N \times N$ blocks and the second corresponding to circularity of the entries within each block. These two sets of indices correspond to partitioning $\mathbf{b}$ into N nonoverlapping vectors, each of length N; the first set indicates which is the right partition and the second to the right element within that partition. For the general element of (A.1), this property can be written more explicitly as

$$\text{(A.2)} \qquad B_{ij} = b_{((k_j-k_i) \bmod N)\cdot N + (l_j-l_i) \bmod N},$$

with $b_{(\cdot)} \in \mathbb{C}$, $i = k_iN + l_i$, $i = 0, \dots, N^2-1$ (same for the index j), such that $l_i, l_j = 0, \dots, N-1$ correspond to the position of (A.2) inside an $N \times N$ circulant block, and $k_i, k_j = 0, \dots, N-1$ correspond to one of the $N \times N$ blocks of $\mathbf{B}$. Notice that by the above construction, the values of k_i and k_j are increased by one every N increments of l_i and l_j.

We now prove that $\mathbf{M}$ is a BCCB matrix.

Proof. Recall that $\mathbf{M} = |\mathbf{A}^H\mathbf{A}|^2$ and that $|\cdot|^2$ is performed elementwise. We start by considering the structure of $\mathbf{A}^H\mathbf{A}$:

$$\text{(A.3)} \qquad \mathbf{A}^H\mathbf{A} = (\mathbf{F}_M^H \otimes \mathbf{F}_M^H)\mathbf{H}^H\mathbf{H}(\mathbf{F}_M \otimes \mathbf{F}_M),$$

with $\mathbf{F}_M$ being a partial $M \times N$ discrete Fourier matrix (its M rows are the corresponding M low frequency rows from a full $N \times N$ discrete Fourier matrix) and $\mathbf{H}$ an $M^2 \times M^2$ diagonal matrix. Denoting the mth column of $\mathbf{F}_M^H \otimes \mathbf{F}_M^H$ by $\tilde{\mathbf{f}}_m$, $m = 1, \dots, M^2$, we may write (A.3) equivalently as

$$\text{(A.4)} \qquad \mathbf{A}^H\mathbf{A} = \sum_{m=1}^{M^2} h_m \tilde{\mathbf{f}}_m \tilde{\mathbf{f}}_m^H,$$

with h_m the mth entry diagonal element of $\mathbf{H}^H\mathbf{H}$.

The mth column of $\mathbf{F}_M^H \otimes \mathbf{F}_M^H$ is the Kronecker product of two columns from $\mathbf{F}_M^H$, say $\hat{\mathbf{f}}_{m_1}$ and $\hat{\mathbf{f}}_{m_2}$, $m_1, m_2 \in 0, \dots, M-1$, where

$$\hat{\mathbf{f}}_m = [1, e^{j\frac{2\pi}{N}m}, \dots, e^{j\frac{2\pi}{N}m(N-1)}]^T, \quad m = 0, \dots, M-1.$$

Replacing the summation over m with a double sum over m_1 and m_2, (A.4) can be written more explicitly as

$$\text{(A.5)} \qquad \mathbf{A}^H\mathbf{A} = \sum_{m_1=0}^{M-1}\sum_{m_2=0}^{M-1} (\hat{\mathbf{f}}_{m_1} \otimes \hat{\mathbf{f}}_{m_2})(\hat{\mathbf{f}}_{m_1}^H \otimes \hat{\mathbf{f}}_{m_2}^H) h_{(M\cdot m_1 + m_2)}.$$

The ijth element of $\mathbf{M}$ is derived directly from (A.5) and has the form

$$\text{(A.6)} \qquad M_{ij} = \left| \sum_{m_1=0}^{M-1}\sum_{m_2=0}^{M-1} e^{j\frac{2\pi}{N}m_1(k_j - k_i)} e^{j\frac{2\pi}{N}m_2(l_j - l_i)} h_{(M\cdot m_1 + m_2)} \right|^2, \quad i, j = 0, \dots, N^2 - 1,$$

where $k_i = \lfloor \frac{i}{N} \rfloor$ and $l_i = i \bmod N$ (also for the index j). Note that the value of k_i changes only between each $N \times N$ block, while the value of l_i changes between the entries of each block. This construction directly implies that $i = k_i N + l_i$ (also for j), as indicated in (A.2).

It can now be observed that M_{ij} is composed of two independent sets of indices. Since $\mathbf{M}$ is of size $N^2 \times N^2$, we can divide it into $N \times N$ nonoverlapping blocks of the same size (similar to Figure 1, right panel). The first exponential term $e^{j\frac{2\pi}{N}m_1(k_j - k_i)}$ corresponds to M being a block circulant matrix with $N \times N$ blocks. This can be seen by the construction of k_i and k_j since $k_j - k_i \in [-(N-1), \dots, N-1]$ and by the periodicity by N of the exponential term.

The second set of indices, $l_j - l_i$, corresponds to each $N \times N$ block being circulant. This can be seen by the term $e^{j\frac{2\pi}{N}m_2(l_j - l_i)}$ since $l_j - l_i \in [-(N-1), \dots, N-1]$ and due to the periodicity by N of the exponent. Thus, M_{ij} has a structure similar to (A.2), with two independent sets of indices; the first corresponds to block circularity and the second to circularity within each block. Consequently, $\mathbf{M}$ is a BCCB matrix. ■

Acknowledgments. We thank Prof. Shimon Weiss and Xiyu Yi for fruitful discussions on super-resolution fluorescence microscopy in general, and SOFI in particular.

REFERENCES

[1] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202, https://doi.org/10.1137/080716542.

[2] A. Beck and M. Teboulle, *Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems*, IEEE Trans. Image Process., 18 (2009), pp. 2419–2434.

[3] T. Bendory, S. Dekel, and A. Feuer, *Robust recovery of stream of pulses using convex optimization*, J. Math. Anal. Appl., 442 (2016), pp. 511–536.

[4] E. Betzig, G. H. Patterson, R. Sougrat, O. W. Lindwasser, S. Olenych, J. S. Bonifacino, M. W. Davidson, J. Lippincott-Schwartz, and H. F. Hess, *Imaging intracellular fluorescent proteins at nanometer resolution*, Science, 313 (2006), pp. 1642–1645, https://doi.org/10.1126/science.1127344.

[5] M. Born and E. Wolf, *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, Cambridge University Press Archive, Cambridge, UK, 2000.

[6] E. J. Candes and M. Wakin, *An introduction to compressive sampling*, IEEE Signal Process. Mag., 25 (2008), pp. 21–30, https://doi.org/10.1109/MSP.2007.914731.

[7] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vision, 20 (2004), pp. 89–97.

[8] D. Cohen and Y. C. Eldar, *Sub-Nyquist sampling for power spectrum sensing in cognitive radios: A unified approach*, IEEE Trans. Signal Process., 62 (2014), pp. 3897–3910.

[9] I. Daubechies, *Orthonormal bases of compactly supported wavelets*, Comm. Pure Appl. Math., 41 (1988), pp. 909–996.

[10] T. Dertinger, R. Colyer, G. Iyer, S. Weiss, and J. Enderlein, *Fast, background-free, 3D super-resolution optical fluctuation imaging (SOFI)*, Proc. Natl. Acad. Sci. USA, 106 (2009), pp. 22287–22292, https://doi.org/10.1073/pnas.0907866106.

[11] T. Dertinger, R. Colyer, R. Vogel, J. Enderlein, and S. Weiss, *Achieving increased resolution and more pixels with Superresolution Optical Fluctuation Imaging (SOFI)*, Opt. Express, 18 (2010), pp. 18875–18885, https://doi.org/10.1364/OE.18.018875.

[12] Y. C. Eldar, *Sampling Theory: Beyond Bandlimited Systems*, Cambridge University Press, Cambridge, UK, 2015.

[13] M. A. Figueiredo, J. B. Dias, J. P. Oliveira, and R. D. Nowak, *On total variation denoising: A new majorization-minimization algorithm and an experimental comparison with wavelet denoising*, in Proceedings of the IEEE International Conference on Image Processing, 2006, pp. 2633–2636.

[14] S. Geissbuehler, N. L. Bocchio, C. Dellagiacoma, C. Berclaz, M. Leutenegger, and T. Lasser, *Mapping molecular statistics with balanced super-resolution optical fluctuation imaging (bSOFI)*, Opt. Nanoscopy, 1 (2012), 4.

[15] J. Goodman, *Introduction to Fourier Optics*, 3rd ed., Roberts and Company, Englewood, CO, 2005.

[16] R. M. Gray, *Toeplitz and Circulant Matrices: A Review*, Now Publishers, Hanover, MA, 2006.

[17] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, Fundam. Algorithms 3, SIAM, Philadelphia, 2006, https://doi.org/10.1137/1.9780898718874.

[18] H. Kirshner, F. Aguet, D. Sage, and M. Unser, *3D PSF fitting for fluorescence microscopy: Implementation and localization application*, J. Microsc., 249 (2013), pp. 13–25.

[19] H. Kirshner, D. Sage, and M. Unser, *3D PSF models for fluorescence microscopy in ImageJ*, in Proceedings of the Twelfth International Conference on Methods and Applications of Fluorescence Spectroscopy, Imaging and Probes (MAF), 2011, p. 54.

[20] J. M. Mendel, *Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications*, Proc. IEEE, 79 (1991), pp. 278–305.

[21] J. Min, C. Vonesch, H. Kirshner, L. Carlini, N. Olivier, S. Holden, S. Manley, J. C. Ye, and M. Unser, *FALCON: Fast and unbiased reconstruction of high-density super-resolution microscopy data*, Sci. Rep., 4 (2014), 4577, https://doi.org/10.1038/srep04577.

[22] J. J. Moreau, *Proximité et dualité dans un espace hilbertien*, Bull. Soc. Math. France, 93 (1965), pp. 273–299.

[23] V. I. Morgenshtern and E. J. Candès, *Super-resolution of positive sources: The discrete setup*, SIAM J. Imaging Sci., 9 (2016), pp. 412–444, https://doi.org/10.1137/15M1016552.

[24] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An iterative regularization method for total variation-based image restoration*, Multiscale Model. Simul., 4 (2005), pp. 460–489, https://doi.org/10.1137/040605412.

[25] M. Ovesnỳ, P. Křížek, J. Borkovec, Z. Švindrych, and G. M. Hagen, *ThunderSTORM: A comprehensive ImageJ plug-in for PALM and STORM data analysis and super-resolution imaging*, Bioinformatics, 30 (2014), pp. 2389–2390.

[26] P. Pal and P. P. Vaidyanathan, *Pushing the limits of sparse support recovery using correlation information*, IEEE Trans. Signal Process., 63 (2015), pp. 711–726.

[27] D. P. Palomar and Y. C. Eldar, *Convex Optimization in Signal Processing and Communications*, Cambridge University Press, Cambridge, UK, 2010.

[28] J. Rietdorf, *Microscopy Techniques*, Adv. Biochem. Eng. Biotechnol. 95, Springer, Berlin, Heidelberg, 2005.

[29] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.

[30] M. J. Rust, M. Bates, and X. Zhuang, *Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (storm)*, Nature Methods, 3 (2006), pp. 793–796.

[31] A. Small and S. Stahlheber, *Fluorophore localization algorithms for super-resolution microscopy*, Nat. Methods, 11 (2014), pp. 267–279.

[32] O. Solomon, M. Mutzafi, M. Segev, and Y. C. Eldar, *Sparsity based super-resolution optical imaging using correlation information*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 6215–6219.

[33] O. Solomon, M. Mutzafi, M. Segev, and Y. C. Eldar, *Sparsity-based super-resolution microscopy from correlation information*, Opt. Express, 26 (2018), pp. 18238–18269.

[34] Z. Tan, Y. C. Eldar, A. Beck, and A. Nehorai, *Smoothing and decomposition for analysis sparse recovery*, IEEE Trans. Signal Process., 62 (2014), pp. 1762–1774.

[35] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. Roy. Statist. Soc. Ser. B, 58 (1996), pp. 267–288.

[36] T. Wimalajeewa, Y. C. Eldar, and P. K. Varshney, *Recovery of Sparse Matrices via Matrix Sketching*, preprint, https://arxiv.org/abs/1311.2448v1, 2013.