doi:10.1017/S1431927619009152
Microsc. Microanal. 25 (Suppl 2), 2019

Making Compressive Sensing Accessible in Scientific Imaging

Andrew Stevens[1], Houari Amari[2] Christopher Buurma[1], Benjamin Bammes[3] Dineth Wije[1], Daniel Nicholls[2], Nathan Johnson[2], Robert Bilhorn[3] and Nigel D. Browning[1,2]

1. Sivananthan Laboratories, Sensor Analytics, Bolingbrook, IL USA
2. University of Liverpool, Mechanical, Materials & Aerospace Eng, Liverpool UK
3. Direct Electron, San Diego, CA USA

Compressive sensing (CS) [1] is opening up new possibilities in scientific imaging. It has been applied in (scanning) transmission electron microscopy (S\TEM) to reduce dose, acquisition time, and data volume [2, 3, 4]. For example, near atomic resolution images of a metal organic framework were acquired using CS in [5], CS of TEM videos was proposed in [6], and CS-STEM ptychography was proposed in [7].

CS is a mathematical framework for signal acquisition that provides a guarantee for recovering a signal from certain kinds of measurements. The term compression arises because acquiring a signal or image using fewer measurements than pixels requires the data to either be multiplexed (*i.e.*, weighted sums of pixels) or to simply not acquire some of the data. The guarantees provided by CS concern the type of weighting used and the minimum number of measurements.

There are two major hurdles to adoption of CS in scientific imaging. The first is that, since the compression occurs in the microscope hardware, CS changes the design of the microscope or deviates from the standard. This means that most of the advances are only available to people building/modifying there own microscopes. In order to address this problem, we have designed new acquisition hardware for both STEM and TEM that can be retrofitted. Specifically, a subsampling STEM scan controller that skips a random subset of pixels, and a direct electron detector that acquires random subsets of pixels.

The second hurdle is that the acquired data has to be processed by a CS recovery algorithm. There are many complications here because now the hardware and software depend on each other. Moreover, using CS means that there is an extra step before the user can see an image—standard CS recovery methods are much slower than STEM acquisition. To address the software issue, we will be releasing open-source CS software specifically developed for microscopy that operates on CPU and GPU architectures (including for example tensor dictionary learning [8] and hyperspectral dictionary learning [9]). An example of our basis pursuit CS recovery of a STEM image is shown in Figure 1 and a comparison of CPU and GPU processing times is shown in Figure 2. For this example we used the NESTA algorithm with a Fourier basis [10].

References:

[1] E. J. Candès and M. B. Wakin. IEEE signal processing magazine **25(2)**, (2008), pp. 21.
[2] A. Stevens, H. Yang, L. Carin *et al.* Microscopy **63(1)**, (2013), pp. 41.
[3] L. Kovarik, A. Stevens, A. Liyu *et al.* Applied Physics Letters **109(16)**, (2016), p. 164102.
[4] A. Stevens. Compressive sensing in transmission electron microscopy. Ph.D. thesis, Duke Univ. (2018).
[5] A. Stevens, L. Luzi, H. Yang *et al.* Applied Physics Letters **112(4)**, (2018), p. 043104.
[6] A. Stevens, L. Kovarik, P. Abellan *et al.* Adv Structural and Chemical Imaging **1(1)**, (2015), p. 10.
[7] A. Stevens, H. Yang, W. Hao *et al.* Applied Physics Letters **113(3)**, (2018), p. 033104.
[8] A. Stevens, Y. Pu, Y. Sun *et al.* Artificial Intelligence and Statistics **54**, (2017), pp. 121.
[9] Z. Xing, M. Zhou, A. Castrodad *et al.* SIAM Journal on Imaging Sciences **5(1)**, (2012), pp. 33.
[10] S. Becker, J. Bobin and E. J. Candès. SIAM Journal on Imaging Sciences **4(1)**, (2011), pp. 1.

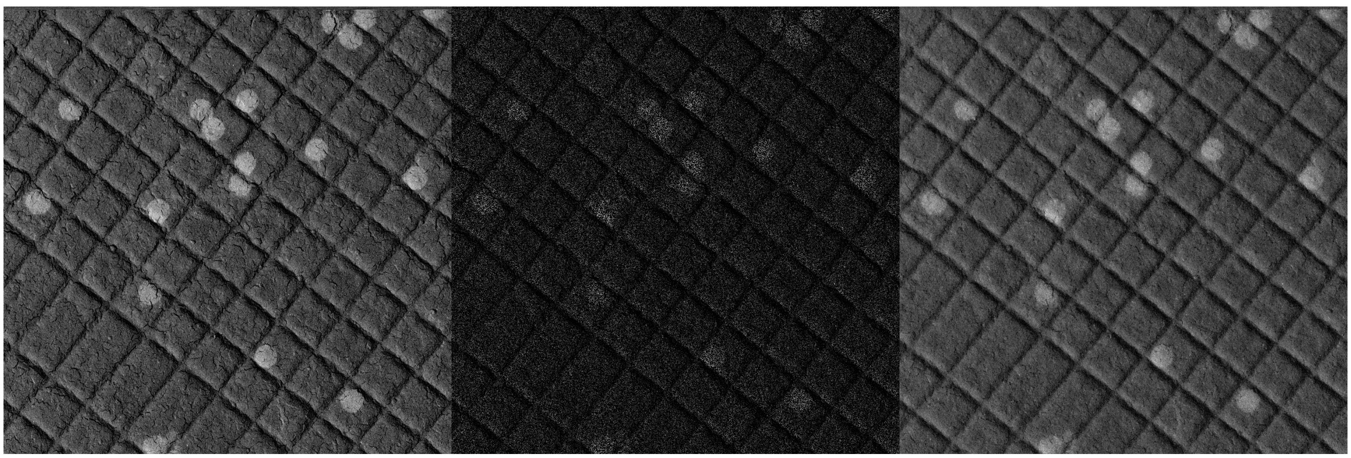

Figure 1: A CS recovery example using Fourier basis pursuit. (Left) original STEM acquisition, (Center) synthetic random sampling (33%), (Right) Recovered image.

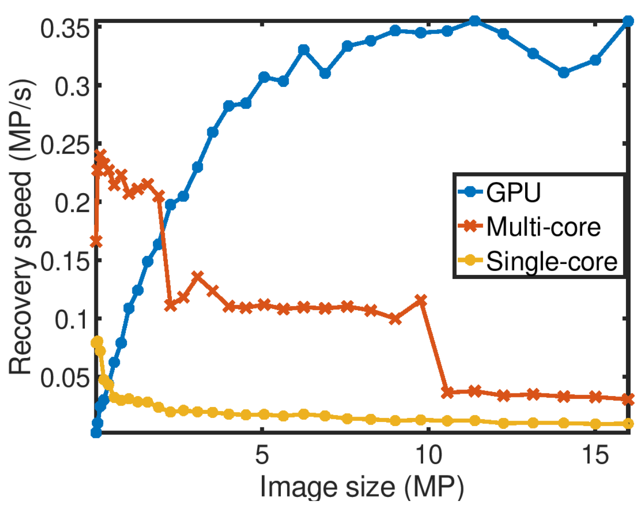

Figure 2: A comparison of CS recovery speed using basis pursuit in the Fourier basis. The image shown in Figure 1 was resized to produce images with sizes between 128×128 and 4096×4096 pixels. The test was performed using a random 25% of the pixels on a laptop with a Nvidia Quadro P2000 Mobile GPU and a 12-core Intel i9-8950HK CPU.

https://doi.org/10.1017/S1431927619009152 Published online by Cambridge University Press