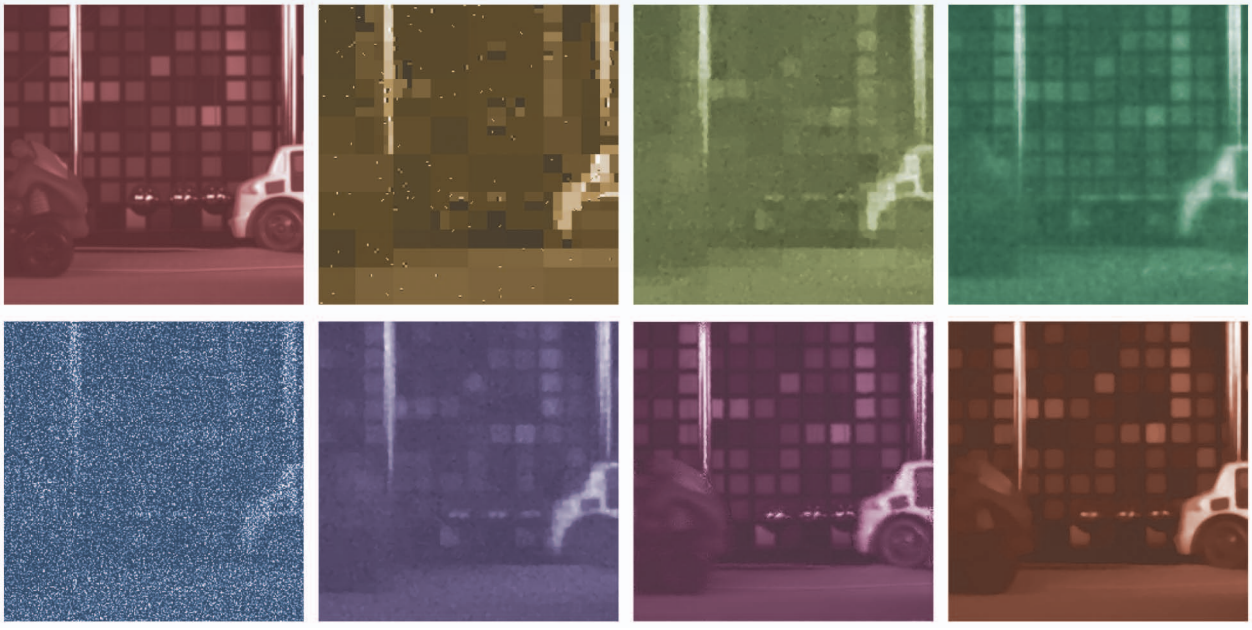
Richard G. Baraniuk, Tom Goldstein,
Aswin C. Sankaranarayanan, Christoph Studer,
Ashok Veeraraghavan, and Michael B. Wakin

Compressive Video Sensing

Algorithms, architectures, and applications



The design of conventional sensors is based primarily on the Shannon–Nyquist sampling theorem, which states that a signal of bandwidth W Hz is fully determined by its discrete time samples provided the sampling rate exceeds $2W$ samples per second. For discrete time signals, the Shannon–Nyquist theorem has a very simple interpretation: the number of data samples must be at least as large as the dimensionality of the signal being sampled and recovered. This important result enables signal processing in the discrete time domain without any loss of information. However, in an increasing number of applications, the Shannon–Nyquist sampling theorem dictates an unnecessary and often prohibitively high sampling rate (see "What Is the Nyquist Rate of a Video Signal?"). As a motivating example, the high resolution of the image sensor hardware in modern cameras reflects the large amount of data sensed to capture an image. A 10-megapixel camera, in effect, takes 10 million measurements of the scene. Yet, almost immediately after acquisition, redundancies in the image are exploited to compress the acquired data significantly, often at compression ratios of 100:1 for visualization and even higher for detection and classification tasks. This example suggests immense wastage in the overall design of conventional cameras.

Compressive sensing (CS) (see "CS 101" and [6], [14], [16], and [24]) is a powerful sensing paradigm that seeks to alleviate the daunting sampling rate requirements imposed by the Shannon–Nyquist principle. CS exploits the inherent structure (or redundancy) within the acquired signal to enable sampling and reconstruction at sub-Nyquist rates. The signal structure most commonly associated with CS is that of sparsity in a transform basis. This is the same structure exploited by image compression algorithms, which transform images into a basis [e.g., using a wavelet or discrete cosine transform (DCT)] where they are (approximately) sparse. In a typical scenario, a CS still-image camera takes a small number of coded, linear measurements of the scene—far fewer measurements than the

Digital Object Identifier 10.1109/MSP.2016.2602099
Date of publication: 11 January 2017

number of pixels being reconstructed. Given these measurements, an image is recovered by searching for the image that is sparsest in some transform basis (wavelets, DCT, or other) while being consistent with the measurements.

In essence, CS provides a framework to sense signals with far fewer measurements than their ambient dimensionality (i.e., Nyquist rate), which translates to practical benefits including decreased sensor cost, bandwidth, and time of acquisition. These benefits are most compelling for imaging modalities where sensing is expensive; examples include imaging in the nonvisible spectrum (where sensors are costly), imaging at high spatial and temporal resolutions (where the high bandwidth of sensed data requires costly electronics), and medical imaging (where the time of acquisition translates to costs or where existing equipment is too slow to acquire certain dynamic events). In this context, architectures like the single-pixel camera (SPC) [27] provide a promising proof of concept that still images can be acquired using a small number of coded measurements with inexpensive sensors.

There are numerous applications where it is desirable to extend the CS imaging framework beyond still images to incorporate video. After all, motion is ubiquitous in the real world, and capturing the dynamics of a scene requires us to go beyond static images. A hidden benefit of video is that it offers tremendous opportunities for more dramatic undersampling (the ratio of signal dimensionality to measurement dimensionality). That is, we can exploit the rich temporal redundancies in a video to reconstruct frames from far fewer measurements than is possible with still images. Yet the demands of video CS in terms of the complexity of imaging architectures, signal models, and reconstruction algorithms are significantly greater than those of compressive still-frame imaging.

There are three major reasons that the design and implementation of CS video systems are significantly more difficult than those of CS still-imaging systems. The first challenge is the gap between compression and CS. State-of-the-art video models rely on two powerful ideas: first, motion fields enable the accurate prediction of image frames by propagating intensities across frames; second, motion fields are inherently more compressible than the video itself. This observation has led to today's state-of-the-art video compression algorithms (not to be confused with CS of videos) that exploit motion information in one of many ways, including block-based motion estimation (MPEG-1), per-pixel optical flow (H.265), and wavelet lifting (LIMAT). Motion fields enable models that can be tuned to the specific video that is being sensed/processed. This is a powerful premise that typically provides an order of magnitude improvement in video compression over image compression.

The use of motion fields for video CS raises an important challenge. Unlike the standard video compression problem,

What Is the Nyquist Rate of a Video Signal?

Conventional videos, sampled at 24–60 frames/second (fps), may, in fact, be highly undersampled in time—objects in the scene can move multiple pixels between adjacent frames. Some compressive sensing (CS) architectures, however, measure a video at a much higher temporal rate. For example, the single-pixel camera (SPC) may take tens of thousands of serial measurements per second. In such cases, the scene may change very little between adjacent measurements. This raises some interesting questions: what is the Nyquist rate of a video signal, and how does it compare to CS measurement rates?

One can gain insight into these questions by considering the three-dimensional analog video signal that arrives at a camera lens; both conventional and CS imaging systems can be viewed as blurring this signal spatially (due to the optics and the pixelated sensors) and sampling or measuring it digitally. If a video consists of moving objects with sharp edges, then the analog video will actually have infinite bandwidth in both the spatial and temporal dimensions. However, it can be argued that the support of the video's spectrum will tend to be localized into a certain bowtie shape, as shown in blue in Figure S1. The salient feature of this shape is that high temporal frequencies coincide only with high spatial frequencies. Thus, because of the limited spatial resolution of both the camera optics and the pixel sensors, when the spatial bandwidth of the video is limited, so too is its temporal bandwidth, as illustrated by the black rectangle in the figure. This suggests that the video sensed by architectures such as the SPC may in fact have a finite temporal bandwidth, and this fact can be used to reduce the computational complexity of sensing and reconstructing the video. In particular, it is not necessary to reconstruct at a rate of thousands of fps. Additional details are provided in [62].

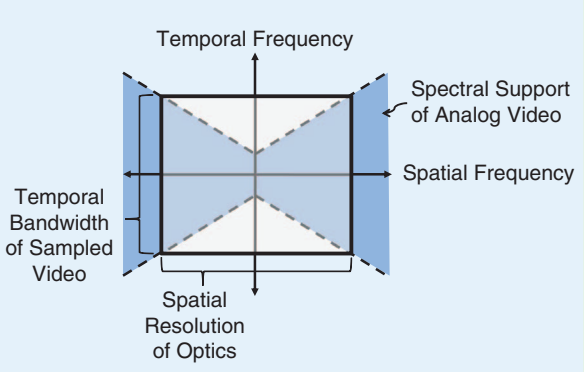


FIGURE S1. The limited spatial resolution of an imaging system may also limit its temporal bandwidth.

CS 101

Compressive sensing (CS) exploits the fact that a small and carefully selected set of nonadaptive linear measurements of a compressible signal, image, or video carries enough information for reconstruction and processing [16], [24]; for a tutorial treatment see [6], [14].

The traditional digital data acquisition approach uniformly samples the three-dimensional analog signal corresponding to the time variations of a scene; the resulting samples $V[x, y, t]$ in space (x, y) and time (t) are sufficient to perfectly recover a bandlimited approximation to the scene at the Nyquist rate. Let the abstract vector s represent the Nyquist-rate samples of the scene $V[x, y, t]$; see "What Is the Nyquist Rate of a Video Signal?" for a discussion of the Nyquist rate of a time-varying scene. Because the number of samples required for real-world scenes, N, is often very large, for example, in the billions for today's consumer digital video cameras, the raw image data is typically reduced via data compression methods that typically rely on transform coding.

As an alternative, CS bypasses the Nyquist sampling process and directly acquires a compressed signal representation using $M < N$ linear measurements between s and a collection of linear codes $\{\phi[m]\}_{m=1}^{M}$ as in $y[m] = \langle s, \phi[m] \rangle$. Stacking the measurements $y[m]$ into the M-dimensional vector y and the transpose of the codes $\phi[m]^T$ as rows into an $M \times N$ sensing matrix Φ, we can write $y = \Phi s$.

The transformation from s to y is a dimensionality reduction and does not, in general, preserve information. In particular, because $M < N$, there are infinitely many vectors s' that satisfy $y = \Phi s'$. The magic of CS is that Φ can be designed such that sparse or compressible signals s can be recovered exactly or approximately from the measurements y. By *sparse*, we mean that only $K \ll N$ of the entries in s are nonzero, or that there exists a sparsifying transform Ψ such that most of the coefficients of $\alpha := \Psi s$ are zero. By *compressible*, we mean that s or α is approximately sparse. Let $\Psi^{-1} := [\psi_1, \psi_2, \ldots, \psi_N]$ represent the inverse of the $N \times N$ basis matrix; then, $s = \Psi^{-1}\alpha$ and $y = \Phi s = \Phi\Psi^{-1}\alpha$.

Typically in CS, the sparse signal s or its sparse coefficients α is recovered by solving an optimization problem of the form (1), where f measures the fidelity of the recovery (e.g., using the squared error $\|y - \Phi\Psi^{-1}\alpha\|_2^2$) and g is a regularization penalty (e.g., the ℓ_1-norm $\|\alpha\|_1$, which promotes sparsity of α). In these cases, the resulting problem is convex, which guarantees a single global minimizer that can be found using a range of algorithms.

While the design of the sensing matrix Φ is beyond the scope of this review, typical CS approaches employ a random matrix. For example, we can draw the entries of Φ as independent and identically distributed ± 1 random variables from a uniform Bernoulli distribution [8]. Then, the measurements y are merely M different sign-permuted linear combinations of the elements of s. Other choices for Φ exist in the literature, such as randomly subsampled Fourier or Hadamard bases. In this case, multiplication by Φ can be accomplished using fast transform algorithms, which enables faster reconstruction than is possible with random matrices.

It is important to emphasize that CS is not a panacea for all the world's sampling problems [7]. In particular, to apply the concept profitably, it is critical that the signal s possess a lower inherent dimensionality than its ambient dimensionality (e.g., sparse structure) and that the degree of undersampling N/M be balanced with respect to the signal's signal-to-noise ratio [22].

where the frames of the video are explicitly available to perform motion estimation, in CS, we have access only to coded and undersampled measurements of the video. We are thus faced with a chicken-or-egg problem. Given high-quality video frames, we could precisely estimate the motion fields; but we need precise motion estimates in the first place to obtain high-quality video frames. The second challenge is the laws of causality and imaging architectures. Time waits for no one. A distinguishing property of the video sensing problem over still imaging is the fundamental difference between space and time. The ephemeral nature of time poses significant limitations on the measurement process—clearly, we cannot obtain additional measurements of an event after it has occurred. As a consequence, it is entirely possible that a compressive camera does not capture a sufficient number of measurements to recover the frames of the video. Overcoming this challenge requires both an understanding of the spatial-temporal resolution tradeoffs associated with video CS and development of novel compressive imaging architectures that can deliver very high measurement rates or reconstruct at different resolutions depending on the available data. The third challenge is computational complexity. Even moderate resolution videos result in high bandwidth streaming measurements. Typical CS video recovery algorithms are highly nonlinear and often involve expensive iterative optimization routines. Fast (or even real-time) reconstruction of CS video is challenging because it requires a data measurement system, fast iterative algorithms, and high-performance hardware jointly designed to enable sufficiently high throughout.

The goal of this article is to overview the current approaches to video CS and demonstrate that significant gains can be obtained using carefully designed CS video architectures and algorithms. However, these gains can only be realized when there is cohesive progress across three distinct fields: video models, compressive video sensing architectures, and video reconstruction algorithms. This article reviews progress that has been made in advancing and bringing these fields together.

We discuss some of the landmark results in video CS and highlight their key properties and the rich interplay among models, architectures, and algorithms that enable them. We also lay out a research agenda to attack the key open research problems and practical challenges to be resolved in video CS.

Video sensing systems

In this section, we discuss the current compressive imaging architectures that have been proposed for CS video. The architectures can be broken down into three categories (see Table 1).

- *Spatial multiplexing cameras* (*SMCs*): SMCs optically superresolve a low-resolution sensor to boost spatial resolution. SMCs are invaluable in regimes where high-resolution sensors are unavailable, as in terahertz/millimeter-wave and magnetic resonance imaging (MRI), or extremely costly, as in short or medium wavelength infrared (SWIR and MWIR) sensing.
- *Temporal multiplexing cameras* (*TMCs*): TMCs optically superresolve a low-frame-rate camera to boost temporal resolution. TMCs are mainly used to overcome the limitations imposed on the measurement rate by the analog-to-digital converter (ADC) and are optimized to produce a high-frame-rate video at high spatial resolution with low-frame-rate sensors.
- *Spectral and angular multiplexing cameras* (*SAMCs*): SAMCs boost resolution in the spectral domain, which can be useful for hyperspectral and light-field video sensing. As with TMCs, the bottleneck of these architectures is also the measurement rate constraint imposed by the ADC.

Each of these flavors of a CS system aims to break the Nyquist barrier to obtain either higher spatial, temporal, or spectral resolution. In the following sections we discuss the key design considerations and existing implementations of these three camera types.

SMCs

SMCs apply CS multiplexing in space to boost the spatial resolution of images and videos obtained from sensor arrays with low spatial resolution. The use of a low-resolution sensor enables SMCs to operate at wavelengths where corresponding full-frame sensors are too expensive, such as at SWIR, MWIR, terahertz, and millimeter wavelengths. SMCs employ a spatial light modulator, such as a digital micromirror device (DMD) or liquid crystal on silicon (LCoS), to optically compute a series of coded inner products with the rasterized scene s; these linear inner products determine the rows of the sensing matrix Φ (recall the notation from "CS 101"). It is worth mentioning that the SMC approach

Table 1. The key architectures for CS video and their properties.

Type	Name	Application	Modulator	Best-known capabilities	Limitations
SMC	SPC	Infrared imaging	DMD	Spatial resolution 128 × 128 Time resolution 64 fps Result [27]	Operational speed of DMD
	LiSens/FPA-CS	Infrared imaging	DMD	Spatial resolution 1,024 × 768 Time resolution 10 fps Result [19], [78]	Need for precise optical alignment/calibration
TMCs	Coded strobing	High-speed imaging	Mechanical/ ferroelectric shutter	Spatial resolution (sensor) Time resolution 2,000 fps Result [75]	Periodic scenes
	Flutter shutter	High-speed imaging	Mechanical/ ferroelectric shutter	Spatial resolution (sensor) Time resolution 4 × sensor fps Result [64]	Locally linear motion
	P2C2	High-speed imaging	LCoS	Spatial resolution (sensor) Time resolution 16 × sensor fps Result [65]	Dynamic range of sensor
	Per-pixel shutter	High-speed imaging	LCoS/electronic shutter	Spatial resolution (sensor) Time resolution 16 × sensor fps Result [39]	Light loss
	CACTI	High-speed imaging	Translating mask	Spatial resolution (sensor) Time resolution 100 × sensor fps Result [51]	Mechanical motion
Light-field video		Dynamic refocusing	LCoS, used as programmable coded aperture	Time resolution sensor fps Result [71]	Loss of spatial resolution can be severe for high spectral/ angular resolutions
Hyperspectral video	CASSI	Spectroscopy	Static mask	Time resolution sensor fps Result [76]	

fps: frames/second; FPA: focal plane array; P2C2: programmable pixel compressive camera; CACTI: coded aperture compressive temporal imaging; CASSI: coded aperture snapshot spectral imaging.

is equally applicable to modalities outside the scope of this article, such as MRI [52], where the physics of image formation produces a measurement system that can be interpreted as subsampling the Fourier transform of the sensed image.

Fast (or even real-time) reconstruction of CS video is challenging because it requires a data measurement system, fast iterative algorithms, and high-performance hardware jointly designed to enable sufficiently high throughout.

SPC

The SPC [27] acquires images using only a single sensor element (i.e., a single pixel) and taking significantly fewer multiplexed measurements than the number of scene pixels. In the SPC, light from the scene is focused onto a programmable DMD, which directs light from a subset of activated micromirrors onto the single photodetector. The programmable nature of the DMD's micromirror orientation enables one to direct light either toward or away from the photodetector. As a consequence, the voltage measured at the photodetector corresponds to an inner product of the image focused on the DMD and the micromirrors directed toward the sensor (see Figure 1). Specifically, at time t, if the DMD pattern is represented by $\phi[t]$ and the time-varying scene by $V[x,y,t]$ (where x and y are the two spatial dimensions and t is the temporal dimension), then the photodetector measures a scalar value $y[t] = \langle \phi[t], V[\cdot,\cdot,t] \rangle + e[t]$, where $\langle \cdot,\cdot \rangle$ denotes the inner product between the vectors and $e[t]$ accounts for the measurement noise. If the scene is static, that is, $V[x,y,t] = V_0[x,y]$, then the measurement vectors can be stacked as columns into a measurement matrix, with $\Phi = [\phi_1, \phi_2, \ldots, \phi_M]^T$. The SPC leverages the relatively high pattern rate of the DMD, which is defined as the number of unique micromirror configurations that can be obtained in unit time. This pattern rate, typically 10–20 kHz for commercially available devices, defines the measurement bandwidth (i.e., the number of measurements per second) and is one of the key factors that defines the achievable spatial and temporal resolutions. Because SPCs rely on the DMD to modulate images onto a single sensor, the spatial resolution is limited by the density of mirrors on the DMD.

Since the proposal of the original SPC in [27], numerous authors have developed alternative SPC architectures that do not require a DMD for spatial light modulation. In [41], a liquid-crystal display panel is used for spatial light modulation; the use of a transmissive light modulator enables a lensless architecture. Sen and Darabi [70] use a camera-projector system to construct an SPC exploiting a concept referred to as *dual photography* [69]; the hallmark of this system is its use of active and coded illumination that can be beneficial in certain applications, particularly microscopy.

Beyond SPCs—Multipixel detectors

As mentioned previously, the measurement rate of an SPC is limited by the pattern rate of its DMD, which is typically in the tens of kilohertz. This measurement rate can be insufficient for scenes with very high spatial and temporal resolutions. This issue can be combatted using an SMC with F sensor pixels (photodetectors), each capturing light from a nonoverlapping region of the DMD. The measurement rate of the SMC increases linearly with the number of photodetectors. Taking into account that the maximum measurement rate is capped by the sampling rate of the ADC, we can write the measurement rate for an SMC with F photodetectors as

$$\min\{F \times R_{\text{DMD}}, R_{\text{ADC}}\},$$

where R_{DMD} is the pattern rate of the DMD and R_{ADC} is the sampling rate of the ADC. Hence, the smallest number of photodetectors for which the measurement rate is maximized is

$$\text{(minimum number of sensor pixels)} \quad F_{\min} = R_{\text{ADC}}/R_{\text{DMD}}.$$

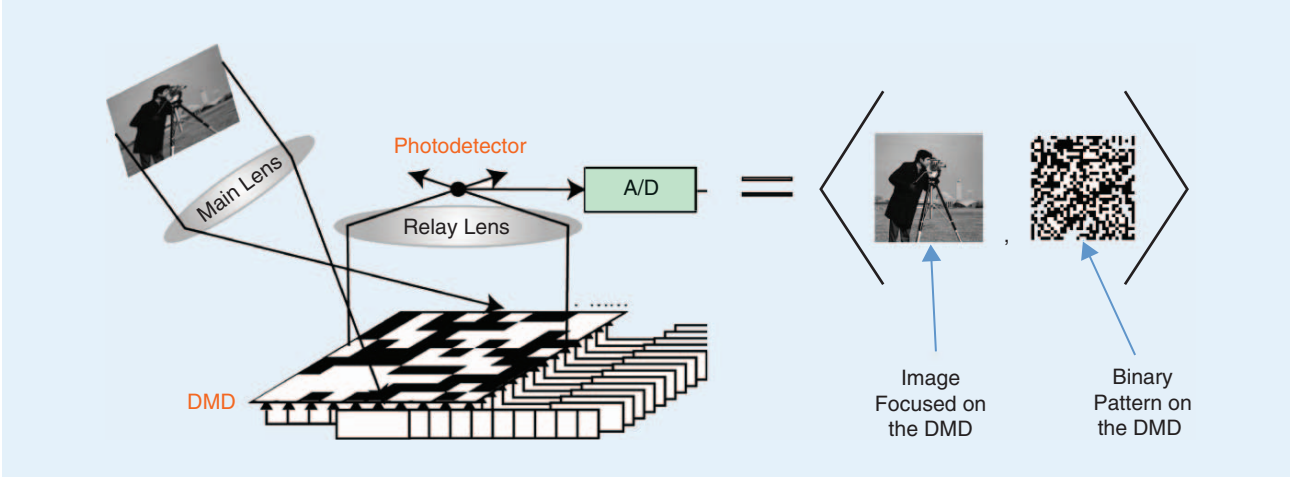


FIGURE 1. The operation principle of the SPC. Each measurement corresponds to an inner product between the binary mirror–mirror orientation pattern on the DMD and the scene to be acquired. (Figure courtesy of [67].)

In essence, at $F = F_{\min}$ we can obtain the measurement rate of a full-frame sensor but using a device with potentially a fraction of the number of photodetectors. This can be invaluable for sensing in many wavebands, for example, SWIR.

As a case study, consider an SMC with a DMD pattern rate $R_{\text{DMD}} = 10$ kHz and an ADC with a sampling rate $R_{\text{ADC}} = 10$ MHz. Then, for a sensor with $F_{\min} = 1{,}000$ pixels, we can acquire 10 million measurements per second. An SPC, in comparison, would acquire only 10,000 measurements per second. Consequently, multipixel SMCs can acquire videos at significantly higher spatial and temporal resolutions than an SPC.

There have been many multipixel extensions to the SPC concept. The simplest approach [46] maps the DMD to a low-resolution sensor array, as opposed to a single photodetector, such that each pixel on the sensor observes a nonoverlapping patch or a block of micromirrors on the DMD. SMCs based on this design have been proposed for sensing in the visible [78], SWIR [19], and MWIR [54]. Figure 2 shows an example of the increased measurement rates offered by the LiSens camera [78], which uses a linear array of 1,024 photodetectors. More recently, there have also emerged multipixel multiplexing-based cameras that completely get rid of the lens and replace the lens with a mask and computational reconstruction algorithms [2].

TMCs

TMCs apply CS multiplexing in time to boost the temporal resolution of videos obtained from sensor arrays with low temporal resolution. Again, let $V[x,y,t]$ be a three-dimensional (3-D) signal representing a time-varying scene. Due to the assumed low frame rate of the sensor, we obtain a scene measurement once every T seconds, where T is too large. If the SLM has an operational speed of one pattern every T_{SLM} seconds, then each measurement of a TMC takes the form of a coded image:

$$y[x,y,t_0] = \sum_{j=0}^{C-1} \phi[x,y,j]\, V[x,y,t_0 + jT_{\text{SLM}}],$$

where $\phi[x,y,j]$ is the attenuation pattern on the SLM at spatial location (x, y) and time jT_{SLM}. Here, each coded image measured by the TMC multiplexes C frames of the high-speed

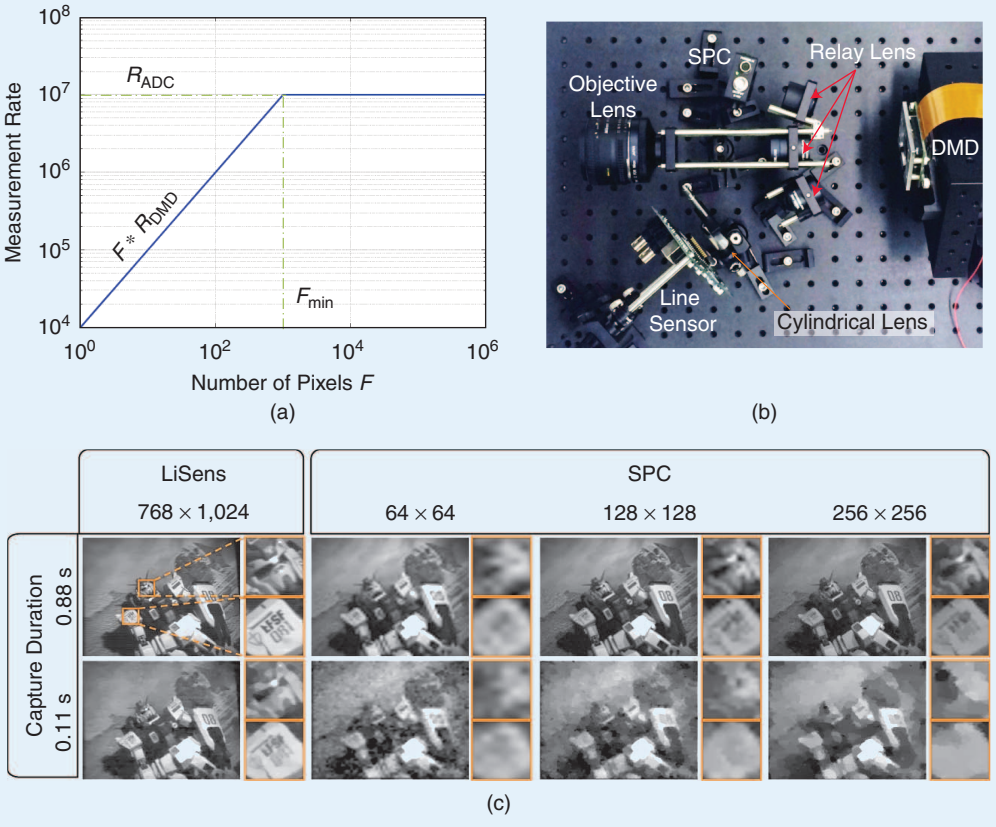


FIGURE 2. The multipixel SMCs support significantly higher sensing rates than an SPC. (a) The measurement rate as a function of the number of sensor pixels. An optimized SMC with $F_{\min}$ pixels delivers the highest possible measurement rate. (b) Lab prototypes of the SPC and LiSens cameras, each placed on the one arm of a single DMD. The measurement rate of the LiSens camera is nearly 1 MHz, while that of the SPC is 20 kHz. (c) Comparisons between LiSens, which uses 1,024 sensor pixels, and an SPC for a static scene. Each row corresponds to a different capture duration, defined as the total amount of time that the cameras have for acquiring compressive measurements. The larger measurement rate of the LiSens camera enables it to sense scenes with very high spatial resolution even for small capture durations. (Photos courtesy of [78].)

video, and hence, we obtain one coded image every CT_{SLM} seconds. Our goal is to recover the frames of the high-speed video $V[x,y,kT_{\mathrm{SLM}}]$ from a single or a sequence of coded images/measurements.

SMCs apply CS multiplexing in space to boost the spatial resolution of images and videos obtained from sensor arrays with low spatial resolution.

Global shutters

The simplest instance of a TMC uses a global shutter together with a conventional camera. In a global shutter, the SLM code $\Phi[x,y,j] = \Phi[j]$ is spatially invariant, which can be implemented by using a programmable shutter or by using the image sensor's built-in electronic shutter. Veeraraghavan et al. [75] showed that periodic scenes can be imaged at very high temporal resolutions using a global shutter [64]. This idea has been extended to nonperiodic scenes in [40], where a union-of-subspace model was used to temporally superresolve the captured scene. However, global shutters are fundamentally limited to providing only spatially invariant coding of the video; this can be insufficient to provide a rich-enough encoding of a high-speed video. Hence, in spite of their simplicity, global shutters fail for scenes with complex motion patterns.

Per-pixel shutters

Reddy et al. [65] proposed the programmable pixel compressive camera (P2C2), which extends the global shutter idea with per-pixel shuttering. Here, each pixel has its own unique code that is typically binary valued and pseudorandom. The P2C2 architecture uses an LCoS SLM placed optically at the sensor plane and carefully aligned to a high-resolution two-dimensional (2-D) sensor. The P2C2 prototype achieves 16 × temporal superresolution, even for complex motion patterns. Hitomi et al. [39] extended the P2C2 camera using a per-pixel coding that is more amenable to implementation in modern image sensors with per-pixel electronic shutters. Here, $\Phi[x,y,j] = \delta[j - j_0(x,y)]$; that is, each pixel observes the intensity at one of the subframes of the high-speed video, and the selection of this subframe varies spatially. Llull et al. [51] and Koller et al. [47] proposed a TMC that achieves temporal multiplexing via a translating mask in the sensor plane. This approach avoids the hardware complexity involved with DMD and LCoS SLMs and enjoys other benefits, including low operational power consumption at the cost of having a mechanical component (the translating mask).

Additional TMC designs

Gu et al. [36] used the rolling shutter of a complementary metal–oxide–semiconductor (CMOS) sensor to enable higher temporal resolution. The key idea is to stagger the exposures of each row randomly and use image/video models to recover a high-frame-rate video. Harmany et al. [37] extended coded aperture systems by incorporating a global shutter; the resulting TMC provides immense flexibility in the choice of the measurement matrix Φ.

SAMCs

SAMCs apply CS multiplexing to sense variations of light in a scene beyond the spatial and temporal dimensions. Two specific examples include hyperspectral CS video cameras that sense spatial, spectral, and temporal variations of light in a scene and light-field video cameras that sense spatial, angular, and temporal variations. In both cases, imaging at high resolution across all modalities simultaneously requires that we handle both high measurement rates (this is typically limited by the ADC sampling rate) and low light levels (due to scene light being resolved into various modalities). CS techniques, more specifically, signal models, can address both bottlenecks. Examples of compressive cameras include the coded aperture snapshot spectral imaging architecture [76] and compressive hyperspectral imaging using spectrometers [50] for spectral multiplexing and the work of Marwah et al. [58] and Tambe et al. [71] for angular multiplexing.

Models for video structure

Recovering a video from compressive linear measurements requires one to extract the video signal s from the measurements $y = \Phi s$ (recall "CS 101"). Here, s might represent a certain block of pixels, an entire video frame, or an ensemble of frames, depending on the sensing architecture and the specific recovery algorithm employed. All of these are functions of the underlying time-varying scene $V[x,y,t]$. Because the number of measurements M is less than the video signal's ambient dimensionality N, infinitely many vectors s' may satisfy $y = \Phi s'$. Hence, to recover s from y, a model that captures the scene structure (or a priori information) of s with a small number of degrees of freedom is required; the model can then be included in the recovery algorithm. This section surveys several popular models for characterizing low-dimensional structure in videos.

Single-frame structure

The structure of a single video frame can be characterized using standard models for conventional 2-D images. Natural images have been shown to exhibit sparse representations in the 2-D DCT, 2-D wavelet, and curvelet domains [15], [56]. Images have also been shown to have sparse gradients. The total variation (TV) seminorm promotes such gradient sparsity simply by minimizing the ℓ_1 norm of an image's 2-D gradient [52]. To fully exploit the structure in a 3-D video, one needs to characterize the spatial and temporal dimensions simultaneously, rather than reconstructing each frame independently and only accounting for spatial structure. Hence, the spatial 2-D regularizers described previously often appear as building blocks of more sophisticated 3-D video models.

Sparse innovation models

One of the simplest possible models accounting for multiframe structure assumes that a video can be reduced into a static and a dynamic component. This model—while restrictive—is applicable, for example, in surveillance applications,

where a scene is observed from a distant static camera. We can decompose each frame of such a video into a static background frame and a number of small (sparse) foreground objects that may change location from frame to frame. A natural way of modeling such structure is to assume that the differences between consecutive frames have a sparse representation in some transform basis. That is, for two consecutive video frames $V[x,y,t_1]$ and $V[x,y,t_2]$, one may assume that the difference frame $V[x,y,t_2]-V[x,y,t_1]$ has a sparse representation in a basis such as a 2-D wavelet basis. Such models have been explored in detail in the context of CS [17], [57], [74] and can be viewed as special cases of the more advanced motion-compensation techniques described below.

One of the simplest possible models accounting for multiframe structure assumes that a video can be reduced into a static and a dynamic component.

Low-rank matrix models

An alternative approach to scene modeling involves reorganizing a 3-D video signal into a 2-D matrix, where each column of the matrix contains a rasterized ordering of the pixels of one video frame. A variety of popular concise models for matrix structure can then be interpreted as models for video structure. One of the most prominent models asserts that the matrix is low rank; this is equivalent to assuming that the columns of the data matrix live in a common, low-dimensional subspace. In the context of video modeling, a seminal result by Basri and Jacobs [9] showed that collections of images of a Lambertian object under varying lighting often cluster close to a nine-dimensional subspace. This property can be useful for modeling videos of stationary scenes where the illumination conditions change over time.

To account for both variations in background illumination and for sparse foreground objects that move with time, one can extend the low-rank matrix model to a low-rank-plus-sparse model [79], [80]. A sparse matrix, added to the original low-rank matrix, accounts for sparse foreground innovations, such as small moving objects. Again, such models are particularly suitable for surveillance applications.

TV minimization and sparse dictionaries

Sparsifying transforms such as wavelets, curvelets, and the DCT have natural extensions to 3-D [56], [77], [82] and can be employed for jointly reconstructing an ensemble of video frames. TV minimization can also be extended to 3-D [35], [49]; minimizing the 3-D TV seminorm of a video promotes frames with sparse gradients across spatial and temporal dimensions.

It is also possible to learn specialized (possibly overcomplete) bases that enable sparse representations of patches, frames, and videos from training data. A variety of so-called dictionary learning algorithms have been proposed that learn sparsifying frames Ψ (see, e.g., [1] and "CS 101"). Dictionary learning algorithms can be used not only to generate dictionaries that sparsify images but also to sparsify videos in both the spatial and temporal dimensions. This approach has been successfully employed for CS video reconstruction in [39].

Linear dynamical systems

Linear dynamical systems (LDSs) model the dynamics in a video using linear subspace models. Such models have been used extensively in the context of activity analysis and dynamic textures. Video CS using LDS reduces to the estimation of the LDS parameters, including the observation matrix and the state transition matrix, from compressive measurements. Approaches for parameter estimation have included recursive [73] as well as batch methods [66]. Furthermore, [66] demonstrates the use of the recovered LDS parameters for activity classification.

Motion compensation

While regularizers such as 3-D wavelets and 3-D TV minimization can be used for CS video reconstruction, it is worth noting that conventional video compression algorithms (such as H.264) do not employ such simple techniques. Rather, because objects in a video may move (or translate) several pixels between adjacent frames, it is typical to employ block-based motion compensation and prediction, where each video frame is partitioned into blocks, the location of each block is predicted in the next frame, and only the residual of this prediction is encoded.

Some CS video architectures may require reconstructions of video sequences with high temporal frame rates. In these cases, there may be relatively little object motion between consecutive frames. Consequently, motion compensation may not be required, and techniques such as 3-D TV may result in high-quality scene recovery.

In other cases, however, it may be necessary to predict and compensate for the motion of objects between consecutive frames. This presents an interesting chicken-or-egg problem: motion compensation can help in reconstructing a video, but the motion predictions themselves cannot be made until (at least part of) the video is reconstructed. One iterative, multiscale technique has been proposed [62] that alternates between motion estimation and video reconstruction: the recovered video at coarse scales (low spatial resolution) is used to estimate motion, which is then used to boost the recovery at finer scales (high spatial resolution). Given the estimated motion vectors, a motion-compensated 3-D wavelet transform can be defined using the LIMAT technique [68]. Another approach initially reconstructs frames individually, estimates the motion between the frames, and then attempts to reconstruct any residual not accounted for by the motion prediction [30]; see also [45] for a related technique. The logistics of block-based video sensing and reconstruction are discussed in detail in [30].

Optical flow

A more general approach to motion compensation involves the optical flow field. Given two frames of a video, $V[x,y,t_1]$ and

$V[x,y,t_2]$, optical flow refers to the flow field $\{u(x,y),v(x,y)\}$ such that $V[x+u(x,y),y+v(x,y),t_1]=V[x,y,t_2]$. Optical flow enables one to represent the frames of a video using a small collection of key frames plus optical flow fields that synthesize (extrapolate) the video from the key frames. Optical flow fields are often significantly more compressible than images. Such an approach is closely related to the block-based motion compensation models described earlier but is distinguished by its explicit attempt to model motion on a per-pixel basis.

A key challenge in the use of optical flow models for video CS is—once again—that, in the context of sensing, we do not have access to the flow fields nor do we have access to high-quality images from which to estimate the flow fields. Reddy et al. [65] resolve this chicken-or-egg problem by first recovering a video with simple image-based priors, estimating the optical flow field on the initial reconstruction, and subsequently recovering the video again while simultaneously enforcing the brightness constancy constraints derived using optical flow. They show that a 30-frames/second (fps) sensor can be superresolved to a 240–480-fps sensor by temporal modulation using an LCoS device. In the context of SMCs, Sankaranarayanan et al. [67] use a specialized dual-scale sensing (DSS) matrix that provides robust and computationally inexpensive initial scene estimates at a lower spatial resolution. This enables this approach to robustly estimate optical flow fields on a low-resolution video. Optical flow-based video CS has also been applied for the dynamic MRI problem, where carefully selected Fourier measurements provide robust initial scene estimates [3]. The concept of DSS sensing matrices has been improved recently by the sum-to-one (STOne) transform [35], which enables the fast recovery of low-resolution scene estimates at multiple resolutions.

Video recovery techniques

While the mathematical formulations of video CS recovery problems resemble other canonical sparse recovery problems, three important factors set video recovery apart from other types of sparse coding. First, video recovery problems are extremely large and have high memory requirements. Methods for high-resolution video recovery must scale to hundreds of millions of unknowns. Second, sparse representations of videos with complex structures may contain tens of thousands (or more) of nonzero entries. Consequently, algorithm implementations that require large dense matrix systems are intractable, and methods must exploit fast transforms. Third, high-quality video recovery often involves noninvertible sparsity transforms, and so reconstruction methods that handle cosparsity models are desirable. Some recovery problems require more sophisticated (or unstructured) models, such as optical flow constraints, that cannot be handled efficiently by simple algorithms. All of these factors impact algorithm performance on different reconstruction applications.

This section overviews the range of existing recovery techniques and investigates the tradeoffs between reconstruction quality and computational complexity. For simplicity, we focus on two categories of reconstruction methods, variational and greedy. Note that there are algorithms that do not fit well into these categories (such as iterative hard thresholding [12], which has features of both); a discussion of such methods is beyond the scope of this article.

Variational methods

Variational methods for CS video recovery perform scene reconstruction by solving optimization problems using iterative algorithms. Most variational methods suitable for high-dimensional problems can be classified into two categories, constrained and unconstrained, as detailed next.

Constrained problems

The first category solves constrained problems of the form

$$\hat{s}=\arg\min_{s,z} f(\Phi s\,|\,y)+g(z)\quad\text{subject to } z=\Psi s. \tag{1}$$

Here, the function f models the video acquisition process (optics, modulation, and sampling), and g is a regularizer that promotes sparsity under the transformation defined by Ψ. For example, basic frame-by-frame recovery with 2-D wavelet sparsity can be formulated as an unconstrained problem with $f(\Phi s\,|\,y)=\|\,y-\Phi s\,\|_2^2$ and $g(z)=\gamma\|\,z\,\|_1$, where s contains a vectorized image frame, Φ is the sensing matrix, Ψ is a 2-D wavelet transform, and $\gamma>0$ is a regularization parameter. Under a TV scene model, the matrix Ψ is a discrete gradient operator that computes differences between adjacent pixels. 3-D TV video recovery can be achieved by stacking multiple vectorized video frames into s and defining Ψ to be the 3-D discrete gradient across both spatial dimensions and time. Optical flow constraints can be included by forming a sparse matrix Ψ that differences pixels in one frame with pixels that lie along its flow trajectory in other frames.

It can be shown that the solution to (1) corresponds to a saddle point of the so-called augmented Lagrangian function

$$\mathcal{L}(s,z,\lambda)=f(\Phi s\,|\,y)+g(z)+\frac{\beta}{2}\|\,z-\Psi s-\lambda\,\|_2^2, \tag{2}$$

where λ is a vector of Lagrange multipliers. Constrained problems of the form (1) for CS video can be solved efficiently using the alternating direction method of multipliers (ADMM) [13], [28], [31] or the primal-dual hybrid gradient (PDHG) method [18], [29]. The ADMM and PDHG methods alternate between minimization steps for s and z and maximization steps for λ until convergence is reached. Such methods have the key advantage that they enable the inclusion of powerful, noninvertible video models such as 3-D TV or optical flow. This advantage, however, comes at the cost of higher memory requirements and somewhat more complicated iterations. To improve the convergence rates of solvers for constrained problems, accelerated algorithm variants have been developed [18], [32], [33].

Unconstrained problems

If the sparsity transform Ψ is invertible, then the constraint in (1) can be removed by replacing the vector s with $\Psi^{-1}z$. This

leads to the second category of recovery methods that solve unconstrained problems of the following simpler form:

$$\hat{z} = \arg\min_z f(\hat{\Phi}z \mid y) + g(z). \tag{3}$$

Here, the matrix $\hat{\Phi} = \Phi\Psi^{-1}$ and z contains the representation of a single frame or the entire video in the sparsity transform domain. For example, in the case of wavelet sparsity, solving (3) recovers the video's wavelet coefficients; the final video is obtained by applying the inverse wavelet transform to the solution.

Unconstrained problems of the form (3) can be solved efficiently using forward–backward splitting (FBS) [20], fast iterative shrinkage/thresholding (FISTA) [10], fast adaptive shrinkage/thresholding algorithm (FASTA) [34], sparse reconstruction by separable approximation (SpaRSA) [81], or approximate message passing (AMP) [25], [55]. FBS is the most basic variant for solving unconstrained problems and performs the following two steps for the iterations $k = 1, 2, \ldots$ until reaching convergence:

$$\hat{z}^{k+1} = z^k + \tau^k \hat{\Phi}^* \nabla f(\hat{\Phi}z^k \mid y) \text{ and} \tag{4}$$

$$z^{k+1} = \arg\min_z g(z) + \frac{1}{2}\| z - \hat{z}^{k+1} \|_2^2, \tag{5}$$

where $\{\tau^k\}$ is some step size sequence. FBS finds a global minimum of the objective function (3) by alternating between the explicit gradient-descent step (4) in the function f and the proximal (or implicit gradient) step (5) in the function g. The key operations of the gradient step (4) are matrix–vector multiplications with $\hat{\Phi}$ and $\hat{\Phi}^*$. These multiplications can be carried out efficiently when $\hat{\Phi}$ is a composition of fast transforms, such as subsampled Hadamard/Fourier matrices and wavelet or DCT operators. When g is a simple sparsity-promoting regularizer, such as the ℓ_1 norm, the proximal step (5) is easy to compute in closed form using wavelet shrinkage. The computational complexity of FBS can be reduced significantly using adaptive step-size rules for selecting $\{\tau^k\}$, acceleration schemes, restart rules, momentum (or memory) terms, and so forth, as is the case for FISTA, FASTA, SpaRSA, and AMP. See the review article [34] for more details.

Greedy pursuit algorithms

Greedy pursuit algorithms are generally used for unconstrained problems and iteratively construct a sparse set of nonzero transform coefficients. Each iteration begins by identifying a candidate sparsity pattern for the unknown vector z. Then, a least-squares problem is solved to minimize $\| \hat{\Phi}z - y \|_2^2$, where z is constrained to have the prescribed sparsity pattern.

Existing greedy pursuit algorithms can be classified into sequential greedy pursuit algorithms and parallel greedy pursuit algorithms. Sequential methods include orthogonal matching pursuit (OMP), regularized OMP (ROMP), and stagewise OMP (StOMP) [26], [61], [72]. These methods successively add more and more indices to the support set until a maximum sparsity K is reached. Parallel methods, such as compressive sampling matching pursuit (CoSaMP) and subspace pursuit [21], [60], constantly maintain a full support set of K nonzero entries but add strong and replace weak entries in an iterative fashion. Parallel greedy pursuit algorithms have the advantage that they can enforce structured models on the support set, such as a wavelet tree structure [5].

The main drawbacks of greedy algorithms, however, are that 1) they are typically unable to handle noninvertible sparsity transforms used for video reconstruction such as TV, optical flow, or overcomplete wavelet frames; 2) accurate solutions are guaranteed only when the measurement operator satisfies stringent conditions (such as the restricted isometry property or similar incoherence conditions [60], [72]); and 3) they require solving large linear systems on every iteration. For small numbers of unknowns (<10,000), the factorization of these systems can be explicitly represented and updated cheaply using rank-one updates. For the large video CS problems considered here, iterative (conjugate gradient) methods are recommended. These methods require only matrix multiplications (which can exploit fast transforms) and have lower memory requirements because they do not require the storage of large and dense matrices.

Reconstruction quality versus computational complexity

There are many choices to make when building a compressive video pipeline, including measurement operators, video models, and reconstruction algorithms. Most reconstruction algorithms are restricted as to what measurement operators and sparsity models they can support. To achieve the best performance, the reconstruction algorithms, video models, and data acquisition pipelines must be designed jointly; this implies that there are tradeoffs to be made among reconstruction speed, algorithm simplicity, and video quality.

The classical approach to CS video recovery is to search for the video that is compatible with the observed measurements while being as sparse as possible in the wavelet domain. When an invertible wavelet transform is used, the reconstruction problem can be transformed into an unconstrained problem of the form (3), which can be solved efficiently using variational methods such as FBS. If we further assume that the wavelet transform is orthogonal, then we can use off-the-shelf greedy pursuit algorithms, such as CoSaMP. Unfortunately, while unconstrained optimization is simple to implement and highly efficient, wavelet-based scene priors generally result in lower reconstruction quality than noninvertible/redundant sparsity models like TV. For this reason, we are often interested in constrained solvers that interface with TV-based video models and optical flow constraints.

To examine the associated performance/complexity tradeoffs, we compare a variety of reconstruction methods using the same measurement operator. A stream of 65,536 STOne measurements [35] was acquired from a 256 × 256 pixel video having 16 frames. Videos were reconstructed separately using various models and solvers that were implemented in MATLAB. We consider unconstrained recovery using CoSaMP and FBS, which are restricted to using invertible regularizers. In the wavelet case, we consider 1) 2-D frame-by-frame

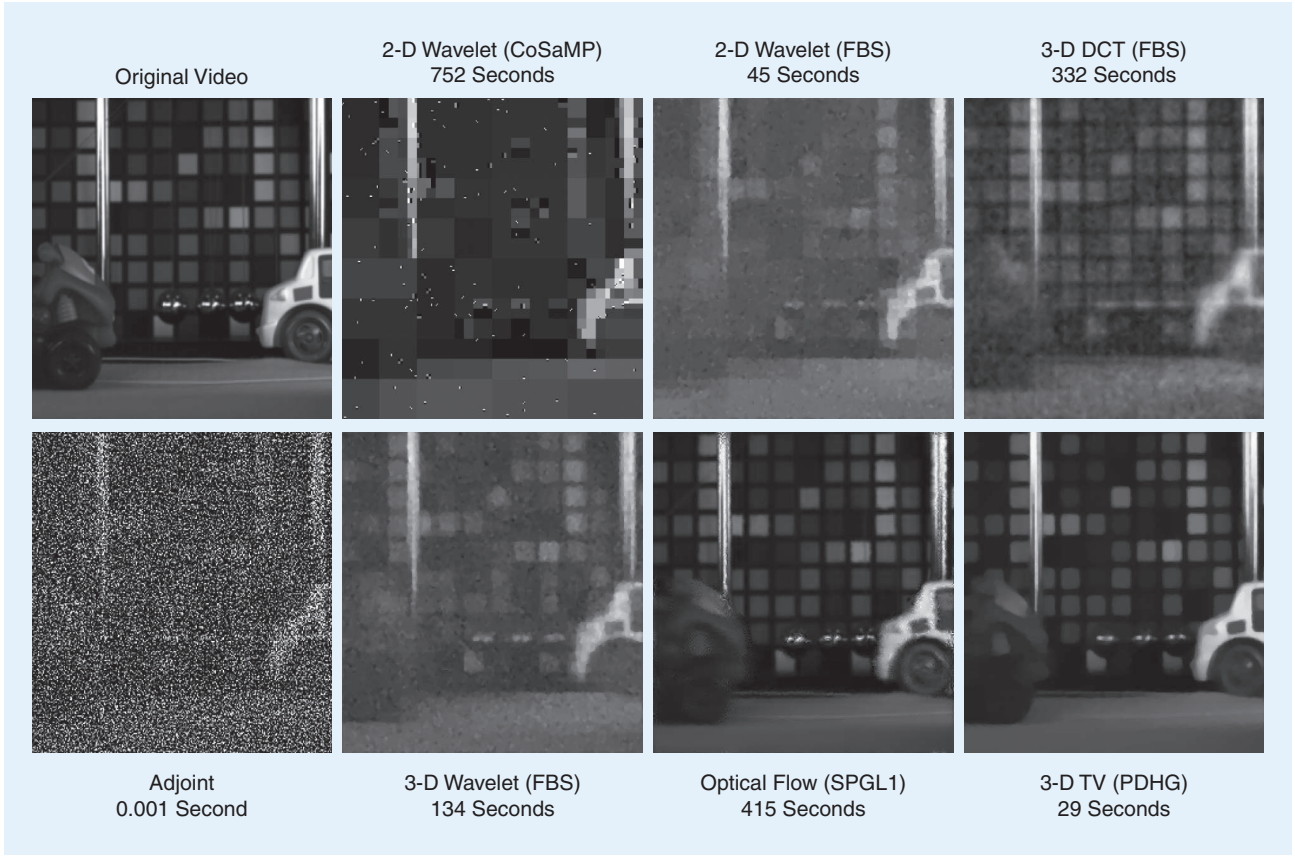


FIGURE 3. A CS video recovery comparison with different video models. For each model, we recover a 16-frame video with 256 $\times$ 256 pixel resolution from 2^{16} STOne transform measurements, corresponding to a 16:1 compression ratio. Sparsity models include 2-D (across space) and 3-D (across space and time) wavelet sparsity using the Haar wavelet, the 3-D DCT, optical flow constraints, and 3-D TV. For each experiment, we also provide the total runtime for recovering 16 frames.

recovery that does not exploit correlations across time, and 2) 3-D wavelet recovery that performs a 3-D wavelet transform across space and time. We also consider sparsity under the 3-D DCT, which is invertible and enjoys extensive use in image and video compression. We furthermore consider solvers for constrained problems that handle more sophisticated sparsity models. In particular, we compare 3-D TV models with PDHG and optical flow constraints with ADMM (as in CS-MUVI [67]). As a baseline, we perform CS video recovery without scene priors by simply computing $\Phi^T y$, the product of the adjoint of the measurement operator with the vector of measurements. Because the measurement operator is a subsampled orthogonal matrix, this corresponds to a least-squares recovery using the pseudoinverse. All experiments are carried out on an off-the-shelf laptop with 16 GB memory and a 2.6 GHz i5 central processing unit (CPU) with two physical cores (no parallelism was used for reconstruction).

Sample frames from our experiments together with the required runtime are shown in Figure 3. We observe that TV regularization and optical flow models dramatically outperform wavelet-based recovery in terms of video quality. Furthermore, 3-D models lead to significantly improved image quality with fewer artifacts than 2-D models, despite the fact that both reconstructions see the same amount of data. This demonstrates the efficacy of exploiting correlations across time. The key advantage of 2-D models is that they enable parallel frame-by-frame reconstruction, for example, by dispatching different recovery problems on separate CPU cores. Finally, we see that for these types of large-scale reconstruction problems, variational methods require substantially lower runtimes than greedy pursuit algorithms. The CoSaMP result in Figure 3 is for frame-by-frame reconstructions with a sparsity level of $K = 256$ nonzero wavelet coefficients per image. CoSaMP's runtime increases dramatically for larger K or when 3-D regularizers are used. This is because each iteration requires the solution to a large least-squares problem using multiple iterative (conjugate gradient) steps. Hence, such greedy pursuit algorithms turn out to be efficient only for highly sparse signals and not for general CS video problems.

Perspectives and open research questions

The video CS problem has spawned a growing body of research that spans signal representations and models, computational sensing architectures, and efficient optimization techniques. This has led to a vibrant ecosystem of methodologies that have transitioned the theoretical ideas of CS into concrete application-specific concepts. We conclude by highlighting some of the important open questions and future research directions.

Real-time CS video recovery with today's hardware

High-quality CS video recovery requires complex algorithms that include powerful video models. While offline video recovery is always feasible, reconstruction using more sophisticated scene models (e.g., using optical flow) can easily take several seconds to minutes even for only a few low-resolution frames. As a consequence, applications that necessitate real-time video recovery face extreme implementation challenges. From our experiments in Figure 3, we see that even the fastest algorithms with basic video models are more than 20× to 200× below real time when executed in MATLAB on off-the-shelf CPUs.

Quite surprisingly, when counting the number of floating-point operations (FLOPs) required for the main transforms of these methods, we observe that real-time CS video recovery with variational methods is within reach of existing hardware. In fact, variational-based scene recovery of a 256 × 256 pixel scene at 12 fps requires only about 20 GFLOPs, which is well below that of programmable processing hardware, such as CPUs, graphics-processing units (GPUs), and field-programmable gate arrays (FPGAs) that achieve peak throughputs of a few TFLOPs. Similarly, existing application specific integrated circuit (ASIC) designs that target CS recovery problems [11], [53] are able to solve variational problems with more than 200 GOPS (the computations are typically carried out with fixed-point arithmetic instead of floating point) using low silicon area and low power when implemented in modern CMOS technology nodes. In Figure 4, we compare the complexity versus the resolution of various CS video recovery methods. One can observe that even higher resolutions like 1080p HD are feasible in real time with computationally efficient algorithms. Nevertheless, no real-time CS video recovery implementation has been proposed in the open literature, which can mainly be attributed to the lack of highly optimized and massively parallel CS video recovery pipelines for programmable hardware (CPUs, GPUs, or FPGAs) as well as dedicated integrated circuits (ASICs). This is definitely a fruitful area for future work.

Compressive inference rather than recovery

The main results of CS are directed toward providing novel sampling theorems that determine the feasibility of signal reconstruction from an underdetermined set of linear measurements. However, reconstruction is often not the eventual goal in most applications, which range from detection and classification to tracking and parameter estimation. While these tasks can all be performed postreconstruction (on the output of a reconstruction procedure), there are important benefits to be gained by performing them directly on the compressive measurements. First, tasks like detection, classification, and tracking are inherently simpler than reconstruction—hence, there is hope that we can perform them with fewer measurements. Second, CS reconstruction is intrinsically tied to the signal models used for the unknown signal, and these signal models prioritize features that deal with visual perception, which often is not the most relevant for the subsequent processing tasks.

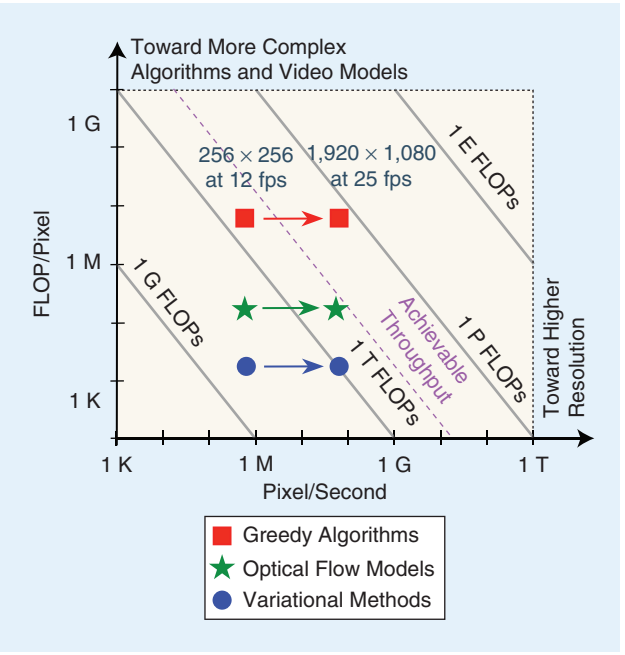


FIGURE 4. The complexity (in FLOPs per pixel) versus resolution (in pixels per second) for greedy algorithms, variational methods, and optical-flow models for the video scene in Figure 3. Variational methods (including 3-D TV and 3-D/2-D wavelets) require the lowest complexity and enable real-time CS video recovery with existing hardware (the diagonal dotted line shows the FLOPs limit of current reprogrammable hardware). Optical flow models exceed the capabilities of current hardware and require the development of more efficient computational methods and faster processing architectures.

Third, as previously discussed at length, CS reconstruction algorithms have high computational complexity, and hence avoiding a reconstruction step in the overall processing pipeline can be beneficial.

There has been some limited work on inference from linear compressive measurements. Davenport et al. [23] perform compressive classification and detection by using a matched filter in the compressive domain. Their key observation is that random projections preserve distances as well as inner products between sparse vectors; thus, inference tasks like hypothesis testing and certain filtering operations can be performed directly in the compressive domain. Hegde et al. [38] show that manifold learning (or nonlinear dimensionality reduction) can be performed just as well on the compressive measurements as on the original data, provided the data arises from a manifold with certain smoothness properties. Sankaranarayanan et al. [66] demonstrate that for time-varying systems well approximated as linear dynamical systems, the parameters of the dynamical system can be directly estimated given compressive measurements. Recently, Kulkarni and Turaga [44] proposed a novel method based on recurrence textures for action recognition from compressive cameras especially for self-similar feature sequences [43]. Apart from these early attempts, there is very little in the literature exploring high-level inference from compressive imagers.

A major hurdle to successful compressive inference in the video context is the mismatch between part-based models, used in computer vision, and global random embeddings, the cornerstone of the CS theory. Part-based models have had remarkable success over the past decade in object detection and classification problems. The key enabler of part-based inference is a local feature description that helps isolate objects from background clutter and provides robustness against object variations. However, the conventional CS measurements are dense random projections that are not conducive to local feature extraction without reconstructing the signal first. Hence, there is an urgent need for CS measurement operator designs that enable local feature extraction.

From measurements to bits—Toward nonlinear sensing architectures

One of the important distinctions between video CS and video compression is the nature of representing the compressed data. Compression aims to reduce the number of bits used to represent the video. In contrast, CS measurements are typically represented in terms of real values with infinite (or arbitrarily large) precision; here, the number of actual measurements is the criterion to reduce/optimize. The focus on reducing the number of measurements is often misplaced in many sensing scenarios; for example, in high-speed video CS, the bottleneck is solely due to the operating speed of the ADC, whose performance is measured in the number of bits acquired per second. Hence, compressively sensing while respecting the bottlenecks imposed by the ADC sampling frequency requires us to consider measurements in terms of bits. While there has been some effort in the area of 1-bit CS [4], [42], [63] and the tradeoff between measurement bits and measurement rate [48], this aspect is still largely unexplored in literature. In particular, there is a need for new kinds of nonlinear sensing architectures that optimize system performance in the context of the practical realities of sensing (quantization, saturation, etc.). Some initial progress in this direction for CS has been made in [59], but the area remains wide open for research.

Acknowledgments

We thank David Robert Jones for his invaluable suggestions and Doug Jones for the JAM. Richard G. Baraniuk was supported by National Science Foundation (NSF) grants CCF-1527501 and CCF-1502875, Defense Advanced Research Projects Agency (DARPA) Revolutionary Enhancement of Visibility by Exploiting Active Light-fields grant HR0011-16-C-0028, and Office of Naval Research (ONR) grant N00014-15-1-2735. Tom Goldstein was supported by NSF grant CCF-1535902 and ONR grant N00014-15-1-2676. Aswin C. Sankaranarayanan was supported by NSF grant IIS-1618823 and Army Research Office grant W911NF-16-1-0441. Christoph Studer was supported in part by Xilinx Inc. and by NSF grants ECCS-1408006 and CCF-1535897. Ashok Veeraraghavan was supported by NSF grant CCF-1527501. Michael B. Wakin was supported by NSF CAREER grant CCF-1149225 and grant CCF-1409258.

Authors

Richard G. Baraniuk (richb@rice.edu) received his B.S. degree from the University of Manitoba, Canada, in 1987; his M.S. degree from the University of Wisconsin-Madison in 1988; and his Ph.D. degree from the University of Illinois at Urbana-Champaign in 1992. He is the Victor E. Cameron Professor of Electrical and Computer Engineering at Rice University, Houston, Texas, and the founder and director of OpenStax (openstax.org). His research interests include new theory, algorithms, and hardware for sensing, signal processing, and machine learning. He is a Fellow of the IEEE and the American Association for the Advancement of Science and has received national young investigator awards from the National Science Foundation and the Office of Naval Research; the Rosenbaum Fellowship from the Isaac Newton Institute of Cambridge University, United Kingdom; the Electrical and Computer Engineering Young Alumni Achievement Award from the University of Illinois at Urbana–Champaign; the IEEE Signal Processing Society Best Paper, Best Column, Education, and Technical Achievement Awards; and the IEEE James H. Mulligan, Jr. Medal.

Tom Goldstein (tomg@cs.umd.edu) received his B.A. degree in mathematics from Washington University, St. Louis, Missouri, in 2006 and his Ph.D. degree in mathematics from the University of California, Los Angeles, in 2010. He has been a visiting research scientist with Stanford University, California, and Rice University, Houston, Texas. He is currently an assistant professor of computer science at the University of Maryland, College Park. His research interests include numerical optimization, distributed computing, image processing, and machine learning.

Aswin C. Sankaranarayanan (saswin@andrew.cmu.edu) received his B.S. degree in electrical engineering from the Indian Institute of Technology, Madras, in 2003 and his Ph.D. degree from the University of Maryland, College Park, where he was awarded the distinguished dissertation fellowship by the Department of Electrical and Computer Engineering (ECE) in 2009. He was a postdoctoral researcher in the Digital Signal Processing group at Rice University, Houston, Texas. He is currently an assistant professor in the ECE Department at Carnegie Mellon University, Pittsburgh, Pennsylvania. His research encompasses problems in compressive sensing and computational imaging. He has received best paper awards at the Computer Vision and Pattern Recognition Workshops on Computational Cameras and Displays (2015) and Analysis and Modeling of Faces and Gestures (2010).

Christoph Studer (studer@cornell.edu) received his M.S. and Ph.D. degrees from ETH Zurich, Switzerland, in 2005 and 2009, respectively. He has been a postdoctoral student and research scientist at ETH Zurich and Rice University, Houston, Texas, and is currently an assistant professor in the School of Electrical and Computer Engineering, Cornell University, Ithaca, New York. His research interests are at the intersection of digital VLSI circuit and system design, signal and image processing, and wireless communication.

Ashok Veeraraghavan (vashok@rice.edu) received his B.S. degree in electrical engineering from the Indian Institute of Technology, Madras, in 2002 and his M.S. and Ph.D. degrees from the Department of Electrical and Computer Engineering, University of Maryland, College Park, in 2004 and 2008, respectively. He is an assistant professor of electrical and computer engineering at Rice University, Houston, Texas, where he directs the Computational Imaging and Vision Lab. His research interests are broadly in the areas of computational imaging, computer vision, and robotics. Before joining Rice University, he spent three years as a research scientist at Mitsubishi Electric Research Labs in Cambridge, Massachusetts. His work has received numerous awards, including the doctoral dissertation award from the Department of Electrical and Computer Engineering at the University of Maryland, the Hershel M. Rich Invention Award from Rice University, and the Best Poster Runner-Up Award from the 2014 International Conference on Computational Photography.

Michael B. Wakin (mwakin@mines.edu) received his B.S. degree in electrical engineering and his B.A. degree in mathematics in 2000, his M.S. degree in electrical engineering in 2002, and his Ph.D. degree in electrical engineering in 2007 from Rice University, Houston, Texas. He is the Ben L. Fryrear associate professor in the Department of Electrical Engineering and Computer Science at the Colorado School of Mines (CSM), Golden. He was a National Science Foundation (NSF) mathematical sciences postdoctoral research fellow at the California Institute of Technology, Pasadena, in 2006–2007 and an assistant professor at the University of Michigan, Ann Arbor, in 2007–2008. His research interests include sparse, geometric, and manifold-based models for signal processing and compressive sensing. He has received the NSF CAREER Award, the Defense Advanced Research Projects Agency Young Faculty Award, and the CSM Excellence in Research Award for his research as a junior faculty member.

References

[1] 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, 2006.

[2] M. S. Asif, A. Ayremlou, A. Sankaranarayanan, A. Veeraraghavan, and R. Baraniuk, "Flatcam: Thin, lensless cameras using coded aperture and computation," *IEEE Trans. Comput. Imag.*, to be published. DOI: 10.1109/TCI.2016.2593662

[3] M. S. Asif, L. Hamilton, M. Brummer, and J. Romberg, "Motion-adaptive spatio-temporal regularization for accelerated dynamic MRI," *Magn. Reson. Medicine,* vol. 70, no. 3, pp. 800–812, 2013.

[4] R. Baraniuk, S. Foucart, D. Needell, Y. Plan, and M. Wootters, "Exponential decay of reconstruction error from binary measurements of sparse signals," arXiv preprint arXiv:1407.8246, 2014.

[5] R. G. Baraniuk, "Optimal tree approximation with wavelets," *Proc. SPIE*, vol. 3813, pp. 196–207, 1999.

[6] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Process. Mag.,* vol. 24, no. 4, pp. 118–121, 2007.

[7] R. G. Baraniuk. (2015). Compressive nonsensing, Norbert Weiner Lecture, Univ. of Maryland. [Online]. Available: http://www.norbertwiener.umd.edu/FFT/2015/15-TAs/baraniuk.html

[8] R. G. 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, Dec. 2008.

[9] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 25, no. 2, pp. 218–233, 2003.

[10] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.,* vol. 18, no. 11, pp. 2419–2434, 2009.

[11] D. E. Bellasi, L. Bettini, C. Benkeser, T. Burger, Q. Huang, and C. Studer, "VLSI design of a monolithic compressive-sensing wideband analog-to-information converter," *IEEE Trans. Emerg. Sel. Topics Circuits Syst.,* vol. 3, no. 4, pp. 552–565, 2013.

[12] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Comput. Harmon. Anal.,* vol. 27, no. 3, pp. 265–274, 2009.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.,* vol. 3, no. 1, pp. 1–122, 2011.

[14] E. Candès and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[15] E. J. Candès and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise c2 singularities," *Commun. Pure Appl. Math.,* vol. 57, no. 2, pp. 219–266, 2004.

[16] E. J. Candès, 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.

[17] V. Cevher, A. C. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa, "Compressive sensing for background subtraction," in *Proc. European Conf. Computer Vision*, Marseille, France, 2008, pp. 155–168.

[18] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Mathematical Imaging Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[19] H. Chen, S. Asif, A. C. Sankaranarayanan, and A. Veeraraghavan, "FPA-CS: Focal plane array-based compressive imaging in short-wave infrared," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* Boston, MA, 2015, pp. 2358–2366.

[20] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering.* New York: Springer-Verlag, 2011, pp. 185–212.

[21] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing reconstruction," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[22] M. Davenport, J. Laska, J. R. Treichler, and R. G. Baraniuk, "The pros and cons of compressive sensing for wideband signal acquisition: Noise folding versus dynamic range," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4628–4642, 2012.

[23] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk, "Signal processing with compressive measurements," *IEEE J. Sel. Topics Signal Process.,* vol. 4, no. 2, pp. 445–460, 2010.

[24] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory,* vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[25] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 106, no. 45, pp. 18914–18919, 2009.

[26] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Trans. Inf. Theory,* vol. 58, no. 2, pp. 1094–1121, 2012.

[27] 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.

[28] J. Eckstein and D. P. Bertsekas, "On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming,* vol. 55, nos. 1–3, pp. 293–318, 1992.

[29] E. Esser, X. Zhang, and T. F. Chan, "A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science," *SIAM J. Imag. Sci.,* vol. 3, no. 4, pp. 1015–1046, 2010.

[30] J. E. Fowler, S. Mun, E. W. Tramel, M. R. Gupta, Y. Chen, T. Wiegand, and H. Schwarz, "Block-based compressed sensing of images and video," *Found. Trends Signal Processing,* vol. 4, no. 4, pp. 297–416, 2010.

[31] R. Glowinski and P. Le Tallec, Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics. Philadelphia, PA: SIAM, 1989.

[32] T. Goldstein, M. Li, and X. Yuan, "Adaptive primal-dual splitting methods for statistical learning and image processing," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett, and R. Garnett, Eds. Red Hook, NY: Curran Associates, Inc., 2015, pp. 2080–2088.

[33] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk, "Fast alternating direction optimization methods," *SIAM J. Imag. Sci.,* vol. 7, no. 3, pp. 1588–1623, 2014.

[34] T. Goldstein, C. Studer, and R. Baraniuk, "A field guide to forward–backward splitting with a fasta implementation," *arXiv preprint arXiv*:1411.3406, 2014.

[35] T. Goldstein, L. Xu, K. F. Kelly, and R. G. Baraniuk, "The STOne transform: Multi-resolution image enhancement and real-time compressive video," *arXiv preprint arXiv*:1311.3405, 2013.

[36] J. Gu, Y. Hitomi, T. Mitsunaga, and S. Nayar, "Coded rolling shutter photography: Flexible space-time sampling," in *Proc. 2010 IEEE Int. Conf. Computational Photography*, Cambridge, MA, pp. 1–8.

[37] Z. T. Harmany, R. F. Marcia, and R. M. Willett, "Compressive coded aperture keyed exposure imaging with optical flow reconstruction," *arXiv preprint arXiv*:1306.6281, 2013.

[38] C. Hegde, M. Wakin, and R. Baraniuk, "Random projections for manifold learning," in *Proc. 22nd Annu. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2008, pp. 641–648.

[39] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S. K. Nayar, "Video from a single coded exposure photograph using a learned over-complete dictionary," in *Proc. 2011 Int. Conf. Computer Vision*, Barcelona, Spain, pp. 287–294.

[40] J. Holloway, A. C. Sankaranarayanan, A. Veeraraghavan, and S. Tambe, "Flutter shutter video camera for compressive sensing of videos," in *Proc. 2012 IEEE Int. Conf. Computational Photography*, Seattle, WA, pp. 1–9.

[41] G. Huang, H. Jiang, K. Matthews, and P. Wilford, "Lenless imaging by compressive sensing," in *Proc. Int. Conf. Image Processing*, Melbourne, Australia, 2013, pp. 2101–2105.

[42] L. Jacques, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk, "Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors," *IEEE Trans. Inf. Theory,* vol. 59, no. 4, pp. 2082–2102, 2013.

[43] I. Junejo, E. Dexter, I. Laptev, and P. Perez, "View-independent action recognition from temporal self-similarities," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 33, no. 1, pp. 172 –185, 2011.

[44] K. Kulkarni and P. Turaga, "Recurrence textures for activity recognition using compressive cameras," in *Proc. 2012 19th IEEE Int. Conf. Image Processing*, Lake Buena Vista, FL, pp. 1417–1420.

[45] L.-W. Kang and C.-S. Lu, "Distributed compressive video sensing," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Taiwan, China, 2009, pp. 1169–1172.

[46] K. Kelly, R. Baraniuk, L. McMackin, R. Bridge, S. Chatterjee, and T. Weston, "Decreasing image acquisition time for compressive imaging devices," U.S. Patent 8,860,835, Oct. 14, 2014.

[47] R. Koller, L. Schmid, N. Matsuda, T. Niederberger, L. Spinoulas, O. Cossairt, G. Schuster, and A. K. Katsaggelos, "High spatio-temporal resolution video with compressed sensing," *Opt. Express,* vol. 23, no. 12, pp. 15992–16007, 2015.

[48] J. Laska and R. G. Baraniuk, "Regime change: Bit-depth versus measurement-rate in compressive sensing," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3496–3505, 2012.

[49] Y. Le Montagner, E. Angelini, and J.-C. Olivo-Marin, "Video reconstruction using compressed sensing measurements and 3d total variation regularization for bio-imaging applications," in *Proc. 2012 19th IEEE Int. Conf. Image Processing*, Lake Buena Vista, FL, pp. 917–920.

[50] C. Li, T. Sun, K. F. Kelly, and Y. Zhang, "A compressive sensing and unmixing scheme for hyperspectral data processing," *IEEE Trans. Image Process.,* vol. 21, no. 3, pp. 1200–1210, 2012.

[51] P. Llull, X. Liao, X. Yuan, J. Yang, D. Kittle, L. Carin, G. Sapiro, and D. J. Brady, "Coded aperture compressive temporal imaging," *Opt. exp., vol.* 21, no. 9, pp. 10526–10545, 2013.

[52] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, "Compressed sensing MRI," *IEEE Signal Process. Mag.,* vol. 25, no. 2, pp. 72–82, 2008.

[53] P. Maechler, C. Studer, D. E. Bellasi, A. Maleki, A. Burg, N. Felber, H. Kaeslin, and R. G. Baraniuk, "VLSI design of approximate message passing for signal restoration and compressive sensing," *IEEE J. Emerg. Sel. Topics Circuits Syst.,* vol. 2, no. 3, pp. 579–590, 2012.

[54] A. Mahalanobis, R. Shilling, R. Murphy, an d R. Muise, "Recent results of medium wave infrared compressive sensing," *Appl. Opt.,* vol. 53, no. 34, pp. 8060–8070, 2014.

[55] M. A. Maleki, "Approximate message passing algorithms for compressed sensing," Ph.D. dissertation, Stanford Univ., Stanford, CA, 2010.

[56] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way,* 3rd ed. New York: Academic Press, 2008.

[57] R. Marcia and R. M. Willett, "Compressive coded aperture video reconstruction," in *Proc. 2008 16th European Signal Processing Conf.*, Lausanne, Switzerland, pp. 1–5.

[58] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar, "Compressive light field photography using overcomplete dictionaries and optimized projections," *ACM Trans. Graph.,* vol. 32, no. 4, p. 46, 2013.

[59] A. Mousavi, A. Patel, and R. G. Baraniuk, "A deep learning approach to structured signal recovery," in *Proc. 53rd Annu. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, 2015.

[60] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples,". *Appl. Comput. Harmon. Anal.,* vol. 26, no. 3, pp. 301–321, Aug. 2009.

[61] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Found. Comp. Math.,* vol. 9, no. 3, pp. 317–334, 2009.

[62] J. Y. Park and M. B. Wakin, "Multiscale algorithm for reconstructing videos from streaming compressive measurements," *J. Electronic Imaging,* vol. 22, no. 2, p. 021001, 2013.

[63] R. Saab, R. Wang, and O. Yilmaz, "From compressed sensing to compressed bit-streams: Practical encoders, tractable decoders," *arXiv preprint arXiv*:1604.00700, 2016.

[64] R. Raskar, A. Agrawal, and J. Tumblin, "Coded exposure photography: Motion deblurring using fluttered shutter," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 795–804, 2006.

[65] D. Reddy, A. Veeraraghavan, and R. Chellappa, "P2C2: Programmable pixel compressive camera for high speed imaging," in *Proc. 2011 IEEE Conf. Computer Vision and Pattern Recognition*, Colorado Springs, CO, pp. 329–336.

[66] A. C. Sankaranarayanan, P. K. Turaga, R. Chellappa, and R. G. Baraniuk, "Compressive acquisition of linear dynamical systems," *SIAM J. Imag. Sci.,* vol 6, no. 4, pp. 2109–2133, 2013.

[67] A. C. Sankaranarayanan, L. Xu, C. Studer, Y. Li, K. F. Kelly, and R. G. Baraniuk, "Video compressive sensing for spatial multiplexing cameras using motion-flow models," *SIAM J. Imag. Sci.,* vol. 8, no. 3, pp. 1489–1518, 2015.

[68] A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression," *IEEE Trans. Image Process.,* vol. 12, no. 12, pp. 1530–1542, 2003.

[69] P. Sen, B. Chen, G. Garg, S. R. Marschner, M. Horowitz, M. Levoy, and H. Lensch, "Dual photography," *ACM Trans. Graph.,* vol. 24, no. 3, pp. 745–755, 2005.

[70] P. Sen and S. Darabi, "Compressive dual photography," *Comput. Graphics Forum,* vol. 28, no. 2, pp. 609–618, 2009.

[71] S. Tambe, A. Veeraraghavan, and A. Agrawal, "Towards motion aware light field video for dynamic scenes," in *Proc. 2013 Int. Conf. Computer Vision*, Sydney, Australia, pp. 1009–1016.

[72] J. Tropp et al., "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans.* Inf. Theory, vol. 50, no. 10, pp. 2231–2242, 2004.

[73] N. Vaswani, "Kalman filtered compressed sensing," in *Proc. 15th IEEE Int. Conf. Image Processing*, San Diego, CA, 2008, pp. 893–896.

[74] N. Vaswani and W. Lu, "Modified-cs: Modifying compressive sensing for problems with partially known support," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4595–4607, 2010.

[75] A. Veeraraghavan, D. Reddy, and R. Raskar, "Coded strobing photography: Compressive sensing of high speed periodic events," *IEEE Trans. Pattern Anal. Mach. Intell.,* 33(4):671–686, Apr. 2011.

[76] A. Wagadarikar, R. John, R. Willett, and D. Brady, "Single disperser design for coded aperture snapshot spectral imaging," *Appl. Opt.,* vol. 47, no. 10, pp. 44–51, 2008.

[77] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. G. Baraniuk, "Compressive imaging for video representation and coding," in *Proc. Picture Coding Symposium*, Beijing, China, 2006.

[78] J. Wang, M. Gupta, and A. C. Sankaranarayanan, "LiSens—a scalable architecture for video compressive sensing," in *Proc. 2015 IEEE Int. Conf. Computational Photography*, Houston, TX, pp. 1–9.

[79] A. E. Waters, A. C. Sankaranarayanan, and R. G. Baraniuk, "SpaRCS: Recovering low-rank and sparse matrices from compressive measurements," in *Proc. Neural Information Processing Systems*, Granada, Spain, 2011, pp. 1089–1097.

[80] J. Wright, A. Ganesh, K. Min, and Y. Ma, "Compressive principal component pursuit," *Information and Inference,* vol. 2, no. 1, pp. 32–68, 2013.

[81] S. J. Wright, R. D. Nowak, and M. A. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.,* vol. 57, no. 7, pp. 2479–2493, 2009.

[82] L. Ying, L. Demanet, and E. Candes, "3D discrete curvelet transform," *Proc. SPIE*, vol. 5914, p. 591413, 2005.

SP