SIAM J. Imaging Sciences
Vol. 6, No. 4, pp. 2109–2133

Compressive Acquisition of Linear Dynamical Systems*

Aswin C. Sankaranarayanan[†], Pavan K. Turaga[‡], Rama Chellappa[§], and Richard G. Baraniuk[¶]

Abstract. Compressive sensing (CS) enables the acquisition and recovery of sparse signals and images at sampling rates significantly below the classical Nyquist rate. Despite significant progress in the theory and methods of CS, little headway has been made in compressive video acquisition and recovery. Video CS is complicated by the ephemeral nature of dynamic events, which makes direct extensions of standard CS imaging architectures and signal models difficult. In this paper, we develop a new framework for video CS for dynamic textured scenes that models the evolution of the scene as a linear dynamical system (LDS). This reduces the video recovery problem to first estimating the model parameters of the LDS from compressive measurements and then reconstructing the image frames. We exploit the low-dimensional dynamic parameters (the state sequence) and high-dimensional static parameters (the observation matrix) of the LDS to devise a novel compressive measurement strategy that measures only the time-varying parameters at each instant and accumulates measurements over time to estimate the time-invariant parameters. This enables us to lower the compressive measurement rate considerably. We validate our approach and demonstrate its effectiveness with a range of experiments involving video recovery and scene classification.

Key words. compressive sensing, linear dynamical system, video compressive sensing

AMS subject classifications. 68U10, 68T45, 93B30

DOI. 10.1137/120863307

1. Introduction.

The Shannon–Nyquist theorem dictates that to sense features at a particular frequency we must sample uniformly at twice that rate. For generic imaging applications, this sampling rate might be too high; in modern digital cameras, invariably, the sensed image is compressed immediately without much loss in quality. For other applications, such as high speed imaging and sensing in the nonvisual spectrum, camera/sensor designs based on the Shannon–Nyquist theorem lead to impractical and costly designs. Part of the reason for this is that the Shannon–Nyquist sampling theory does not exploit any structure in the sensed signal beyond that of band-limitedness. Signals with redundant structures can potentially be

[*] Received by the editors January 23, 2012; accepted for publication (in revised form) July 29, 2013; published electronically October 22, 2013. The work of the first and fourth authors was partially supported by the grants NSF CCF-0431150, CCF-0728867, CCF-0926127, CCF-1117939, ARO MURI W911NF-09-1-0383, W911NF-07-1-0185, DARPA N66001-11-1-4090, N66001-11-C-4092, N66001-08-1-2065, ONR N00014-12-1-0124, and AFOSR FA9550-09-1-0432.

http://www.siam.org/journals/siims/6-4/86330.html

[†] Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 (saswin@ece.cmu.edu).

[‡] Arts Media and Engineering Department, Arizona State University, Tempe, AZ 85287 (pturaga@asu.edu).

[§] Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20740 (rama@cfar.umd.edu). This author's work was partially supported by the Office of Naval Research under grant N00014-12-1-0124.

[¶] Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005 (richb@rice.edu, http://dsp.rice.edu).

sensed more parsimoniously. This is the key idea underlying the new field of *compressive sensing* (CS) [7]. When the signal of interest exhibits a sparse representation, CS enables sensing at measurement rates below the Nyquist rate. Indeed, signal recovery is possible from a number of measurements that is proportional to the sparsity level of the signal, as opposed to its bandwidth.

In this paper, we consider the problem of sensing *videos* compressively. We are interested in this problem motivated by the success of video compression algorithms, which indicates that videos are highly redundant. Bridging the gap between compression and sensing can lead to compelling camera designs that significantly reduce the amount of data sensed and enable designs for application domains where sensing is inherently costly.

Video CS is challenging for two main reasons:

- *Ephemeral nature of videos:* The scene changes during the measurement process; moreover, we cannot obtain additional measurements of an event after it has occurred.
- *High-dimensional signals:* Videos are significantly higher-dimensional than images. This makes the recovery process computationally intensive.

One way to address these challenges is to narrow our scope to certain parametric models that are suitable for a broad class of videos; this morphs the video recovery problem to one of parameter estimation and provides a scaffold for addressing the challenges listed above.

In this paper, we develop a CS framework for videos modeled as linear dynamical systems (LDSs), which is motivated, in part, by the extensive use of such models in characterizing dynamic textures [10, 15, 33], activity modeling, and video clustering [37]. Parametric models, like LDSs, offer lower-dimensional representations for otherwise high-dimensional videos. This significantly reduces the number of free parameters that need to be estimated and, as a consequence, reduces the amount of data that needs to be sensed. In the context of video sensing, LDSs offer interesting trade-offs by characterizing the video signal using a mix of dynamic/time-varying parameters and static/time-invariant parameters. Further, the generative nature of LDSs provides a prior for the evolution of the video in both forward and reverse time. To a large extent, this property helps us circumvent the challenges presented by the ephemeral nature of videos.

This paper makes the following contributions. We propose a framework called *CS-LDS* for video acquisition using an LDS model coupled with sparse priors for the parameters of the LDS model. The core of the framework is a *two-step measurement strategy* that enables the recovery of the LDS parameters from compressive measurements by solving a sequence of linear and convex problems. We demonstrate that CS-LDS is capable of sensing videos with far fewer measurements than the Nyquist rate. Finally, the LDS parameters form an important class of features for activity recognition and scene analysis, thereby making our camera designs purposive [25] as well.

2. Background.

2.1. Compressive sensing.

CS deals with the recovery of a signal $\mathbf{y} \in \mathbb{R}^N$ from undersampled linear measurements of the form $\mathbf{z} = \Phi\mathbf{y} + \mathbf{e}$, where $\Phi \in \mathbb{R}^{M \times N}$ is the measurement matrix, $M < N$, and $\mathbf{e}$ is the measurement noise [7, 14]. Estimating $\mathbf{y}$ from the measurements $\mathbf{z}$ is ill-conditioned, since the linear system formed by $\mathbf{z} = \Phi\mathbf{y}$ is underdetermined. CS works under the assumption that the signal $\mathbf{y}$ is sparse in a basis Ψ; that is, the signal $\mathbf{s}$, defined as $\mathbf{y} = \Psi\mathbf{s}$, has at most K nonzero components. Exploiting the sparsity of $\mathbf{s}$, the signal $\mathbf{y}$ can be recovered exactly from $M = O(K\log(N/K))$ measurements, provided that the matrix $\Phi\Psi$ satisfies the so-called *restricted isometry property* (RIP) [4]. In particular, when Ψ is an orthonormal basis and the entries of the matrix Φ are i.i.d. (independent and identically distributed) samples from a sub-Gaussian distribution, the product $\Phi\Psi$ satisfies the RIP. Further, the signal $\mathbf{y}$ can be recovered from $\mathbf{z}$ by solving a convex problem of the form

$$(1) \qquad \min \|\mathbf{s}\|_1 \quad \text{subject to } \|\mathbf{z} - \Phi\Psi\mathbf{s}\|_2 \leqslant \epsilon,$$

where ϵ is an upper bound on the measurement noise $\mathbf{e}$. It can be shown that the solution to (1) is with high probability the K-sparse solution that we seek. The theoretical guarantees of CS have been extended to *compressible* signals, where the sorted coefficients of $\mathbf{s}$ decay rapidly according to a power-law [22].

There exist a wide range of algorithms to solve (1) under various approximations or reformulations [7, 38]. Greedy techniques such as orthogonal matching pursuit [28] and CoSAMP [26] solve the sparse approximation problem efficiently with strong convergence properties and low computational complexity. It is also simple to impose structural constraints such as block sparsity into CoSAMP, giving variants such as model-based CoSAMP [3].

2.2. Video CS. In this paper, we model a video as a sequence of time-indexed images. Specifically, if $\mathbf{y}_t$ is the image of a scene at time t, then $\mathbf{y}_{1:T} = \{\mathbf{y}_1, \dots, \mathbf{y}_T\}$ is the video of the scene from time 1 to T. Further, we also refer to $\mathbf{y}_t$ as the "video frame" at time t.

In video CS, the goal is to sense a time-varying scene using compressive measurements of the form $\mathbf{z}_t = \Phi_t\mathbf{y}_t$, where $\mathbf{z}_t, \Phi_t$, and $\mathbf{y}_t$ are the compressive measurements, the measurement matrix, and the video frame at time t, respectively. Given the sequence of compressive measurements $\mathbf{z}_{1:T} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_T\}$, our goal is to recover the video $\mathbf{y}_{1:T} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T\}$. There are currently two fundamentally different imaging architectures for video CS: the single pixel camera (SPC) and the programmable pixel camera. The SPC [16] uses a single or a small number of sensing elements. Typically, a photo-detector is used to obtain a single measurement at each time instant of the form $\mathbf{z}_t = \phi_t^T\mathbf{y}_t$, where ϕ_t is a pseudorandom vector of 0's and 1's. Typically, under an assumption of a slowly varying scene, consecutive measurements from the SPC are grouped as measurements of the same video frame. This assumption works only when the scene motion is small or when the number of measurements associated with a frame is small. The SPC provides complete freedom in the spatial multiplexing of pixels; however, there is no temporal multiplexing. In contrast, programmable pixel cameras [23, 31, 43] use a full frame sensor array; during each exposure of the sensor array, the shutter at each pixel is temporally modulated. This enables extensive temporal multiplexing but a limited amount of spatial multiplexing. A key advantage of SPC-based designs is that they can operate efficiently at wavelengths (such as the far infrared) that require exotic detectors; in such cases, building a full frame sensor can be prohibitively expensive.

To date, recovery algorithms for the SPC have used various signal models to reconstruct the sensed scene. Wakin et al. [45] use three-dimensional (3D) wavelets as the sparsifying basis for recovering videos from compressive measurements. Park and Wakin [27] use a coarse-to-fine estimation framework wherein the video, reconstructed at a coarse scale, is used to estimate motion vectors that are subsequently used to design dictionaries for reconstruction at a finer scale. Vaswani [40] and Vaswani and Lu [41] use a sequential framework that exploits the similarity of support of the signal between adjacent frames of a video. Under this model, a frame of video is reconstructed using a linear inversion over the support at the previous time instant and a small-scale CS recovery over the residue to detect components beyond the known support. Cevher et al. [9] provide a CS framework for directly sensing innovations over a static scene thereby enabling background subtraction from compressive measurements.

2.3. LDS model for video sequences. LDSs represent an important class of parametric models for time-series data. A wide variety of spatio-temporal signals have often been modeled as realizations of LDSs. These include dynamic textures [15], traffic scenes [10], video inpainting [13], multicamera tracking [2], and human activities [37]. The interested reader is referred to [36] for a survey of the use of LDSs as a concise representation for a wide range of computer vision problems.

Intuitively, an LDS for a video comprises two models. First is an *observation model* that suggests that frames of the video lie close to a d-dimensional subspace; the frame of the video at time t can be represented as $\mathbf{y}_t \approx C\mathbf{x}_t$, where C is a basis for the subspace and $\mathbf{x}_t$ are the subspace coefficients or the state vector at time t. Second, the trajectory that the video charts out in this d-dimensional subspace varies smoothly, is predictable, and is modeled by a linear evolution of the form $\mathbf{x}_{t+1} \approx A\mathbf{x}_t$. Figure 1 provides an example of an LDS.

We now formally define the LDS for a video. The model equations are given by

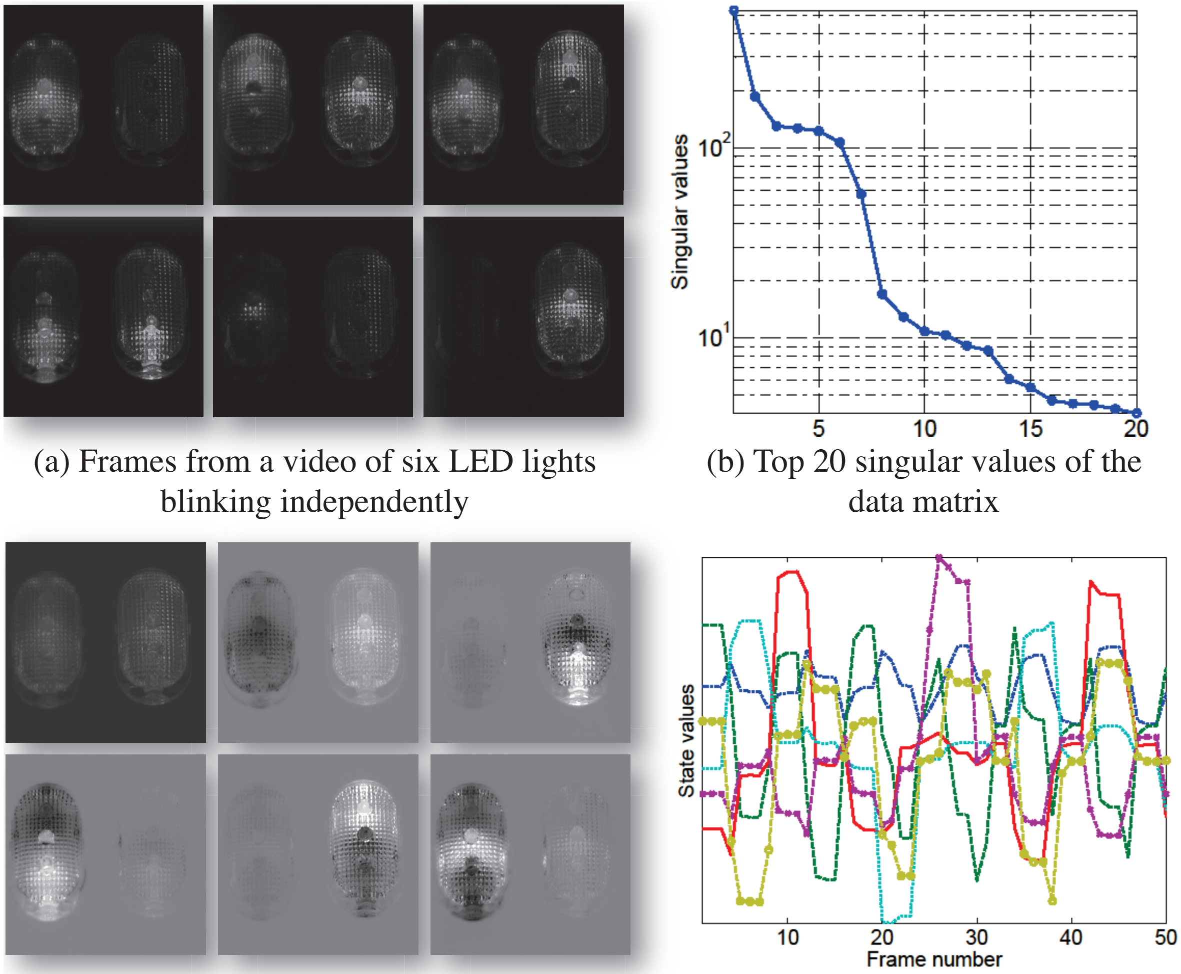
$$\mathbf{y}_t = C\mathbf{x}_t + \mathbf{w}_t, \qquad \mathbf{w}_t \sim N(0, R), \tag{2}$$

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + \mathbf{v}_t, \qquad \mathbf{v}_t \sim N(0, Q), \tag{3}$$

where $\mathbf{x}_t \in \mathbb{R}^d$ is the state vector at time t, d is the dimension of the state space, $A \in \mathbb{R}^{d\times d}$ is the state transition matrix, $C \in \mathbb{R}^{N\times d}$ is the observation matrix, and $\mathbf{y}_t \in \mathbb{R}^N$ represents the observed measurements, where for the videos of interest in this paper, $d \ll N$. $\mathbf{w}_t$ and $\mathbf{v}_t$ are noise components modeled as Gaussian with 0 mean vector and covariance matrices given by $R \in \mathbb{R}^{N\times N}$ and $Q \in \mathbb{R}^{d\times d}$, respectively. The Gaussian assumption for the process noise is not necessarily an optimal one but is made for the sake of simplifying the model estimation algorithm. It is known to work well for representing a large class of dynamic textures [15].

An LDS is parameterized by the matrix pair (C, A). Note that the choice of C and the state sequence $\mathbf{x}_{1:T}$ is unique only up to a $d \times d$ linear transformation, given the inherent ambiguities in the notion of a state space. In particular, given *any* invertible $d \times d$ matrix L, the LDS defined by (C, A) with the state sequence $\mathbf{x}_{1:T}$ is equivalent to the LDS defined by $(CL, L^{-1}AL)$ with the state sequence $L^{-1}\mathbf{x}_{1:T} = \{L^{-1}\mathbf{x}_1, L^{-1}\mathbf{x}_2, \ldots, L^{-1}\mathbf{x}_T\}$. This lack of uniqueness has implications that we will touch upon later in section 5.

Given a video sequence, the most common approach to fitting an LDS model is to first estimate a lower-dimensional embedding of the observations via principal component analysis (PCA) and then learn the temporal dynamics captured in $\mathbf{x}_t$, and equivalently A. The most popular model estimation algorithms are N4SID [39], PCA-ID [35], and expectation-maximization (EM) [10]. N4SID is a subspace identification algorithm that provides an asymptotically optimal solution for the model parameters. However, for large problems the computational requirements make this method prohibitive. PCA-ID [35] is a suboptimal solution to



(a) Frames from a video of six LED lights blinking independently

(b) Top 20 singular values of the data matrix

(c) Basis vectors for a six-dimensional approximation of the data

(d) State sequence associated with the six-dimensional basis

Figure 1. *An example of an LDS and the models that define it.* (a) *A few frames of a video of six LEDs flashing independently.* (b) *Top* 20 *singular values of the data matrix* $[\mathbf{y}_{1:T}]$*—formed by stacking frames of the video as its columns. Note how the singular values, outside the top six, decay rapidly. This is a consequence of the linear nature of light, suggesting that the frames of the video lie on a* 6*D subspace. In practice, deviations from linearity due to saturation lead to small deviations from the* 6*D subspace as noted from the decaying singular values.* (c) *Basis vectors associated with a* 6*D approximation of the data. Blacker pixels denote nonnegative entries, while whiter pixels denote positive entries. Together, they define a* 6*D subspace that defines the observation model of the LDS.* (d) *State sequence associated with the* 6*D approximation. The smooth variation of the state values indicate predictability over small time durations—one of the key hallmarks of an LDS. These smooth transitions are captured by the state transition model.*

the learning problem. It makes the assumption that estimation of the observation matrix C and the state transition matrix A can be separable, which makes it possible to estimate the parameters of the model very efficiently via PCA. Under this assumption, one first estimates the observation matrix C (space-filter), and then uses the result to estimate the state transition matrix A (time-filter) [15]. This learning problem can also be posed as a maximum likelihood estimation of the model parameters that maximize the likelihood of the observations, which can be solved by the EM algorithm [10].

3. CS-LDS architecture.

We provide a high level overview of our proposed framework for video CS; the goal here is to build a CS framework, implementable on the SPC, for videos that are modeled as LDSs. We flesh out the details in sections 4 and 5. This amounts to estimating the LDS parameters from compressive measurements; i.e, we seek to recover the model parameters C and $\mathbf{x}_{1:T}$ given compressive measurements of the form $\mathbf{z}_t = \Phi_t \mathbf{y}_t = \Phi_t C \mathbf{x}_t$. We recall that C is the time-invariant observation matrix of the LDS, and $\mathbf{y}_t$ and $\mathbf{x}_t$ are the video frame and the state at time t, respectively. The compressive measurements $\mathbf{z}_{1:T}$ are hence expressed as bilinear terms in the unknown parameters C and $\mathbf{x}_{1:t}$. Handling bilinear unknowns typically requires nonconvex optimization techniques, thereby invalidating conventional CS recovery algorithms. To avoid this, we propose a two-step sensing method that is specifically designed to address the bilinearity; we refer to this sensing method and its associated recovery algorithm as the *CS-LDS* framework [34].

Measurement model: We summarize the CS-LDS measurement model as follows. At time t, we take two sets of measurements:

$$\mathbf{z}_t = \begin{pmatrix} \check{\mathbf{z}}_t \\ \tilde{\mathbf{z}}_t \end{pmatrix} = \begin{bmatrix} \check{\Phi} \\ \tilde{\Phi}_t \end{bmatrix} \mathbf{y}_t = \Phi_t \mathbf{y}_t, \tag{4}$$

where $\check{\mathbf{z}}_t \in \mathbb{R}^{\widecheck{M}}$ and $\tilde{\mathbf{z}}_t \in \mathbb{R}^{\widetilde{M}}$ such that the total number of measurements at each frame is $M = \widecheck{M} + \widetilde{M}$.[1] The measurement matrix in (4) is composed of two distinct components: the *time-invariant* part $\check{\Phi}$ and the *time-varying* part $\tilde{\Phi}_t$. We denote by $\check{\mathbf{z}}_t$ the *common* measurements, and by $\tilde{\mathbf{z}}_t$ the *innovation* measurements.

We solve for the LDS parameters in two steps. First, we obtain an estimate of the state sequence using only the common measurements $\check{\mathbf{z}}_{1:T}$. Second, we use this state sequence estimate to recover the observation matrix C using the innovation measurements.

State sequence estimation: We recover the state sequence $\mathbf{x}_{1:T}$ using only the common measurements $\check{\mathbf{z}}_{1:T}$. The key idea is that when $\mathbf{y}_{1:T}$ form the observations of an LDS with system matrices (C, A), the measurements $\check{\mathbf{z}}_{1:T}$ form the observations of an LDS with system matrices $(\check{\Phi}C, A)$. Estimation of the state sequence now can be mapped to a simple exercise in system identification. In particular, an estimate of the state sequence can be obtained by the singular value decomposition (SVD) of the block-Hankel matrix

$$\mathrm{Hank}(\check{\mathbf{z}}_{1:T}, d) = \begin{bmatrix} \check{\mathbf{z}}_1 & \check{\mathbf{z}}_2 & \cdots & \cdots & \check{\mathbf{z}}_{T-d+1} \\ \check{\mathbf{z}}_2 & \iddots & \iddots & & \check{\mathbf{z}}_{T-d+2} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ \vdots & & \iddots & \iddots & \vdots \\ \check{\mathbf{z}}_d & \cdots & \cdots & \check{\mathbf{z}}_{T-1} & \check{\mathbf{z}}_T \end{bmatrix}. \tag{5}$$

[1] The SPC obtains only one measurement at each time instant. Multiple measurements for a video frame are obtained by grouping consecutive measurements from the SPC. When M is small compared to the sampling rate of the SPC, this is an acceptable approximation especially for slowly varying scenes.

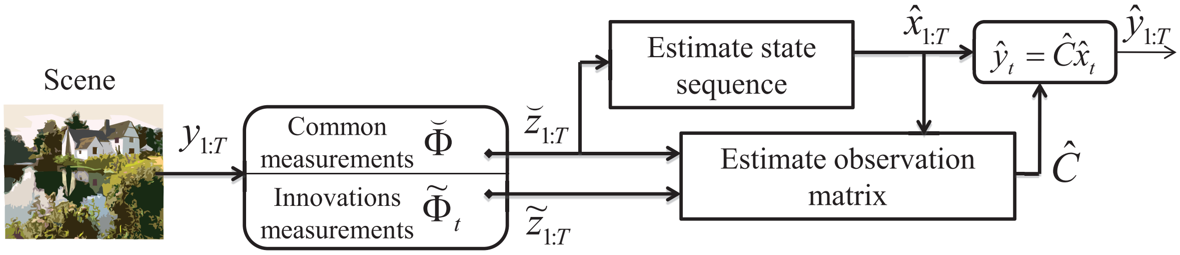

Scene
$y_{1:T}$
Common measurements $\breve{\Phi}$
Innovations measurements $\widetilde{\Phi}_t$
$\breve{z}_{1:T}$
$\widetilde{z}_{1:T}$
Estimate state sequence
$\hat{x}_{1:T}$
$\hat{y}_t = \hat{C}\hat{x}_t$
$\hat{y}_{1:T}$
Estimate observation matrix
$\hat{C}$

Figure 2. *Block diagram of the CS-LDS framework.*

Given the SVD(Hank($\breve{\mathbf{z}}_{1:T}, d$)) $= U_H S_H V_H^T$, the state sequence estimate is given by

$$[\widehat{\mathbf{x}}_{1:T}] = S_H V_H^T.$$

In section 4, we leverage results from system identification to analyze the properties of this particular estimate as well as characterize the number of measurements $\breve{M}$ required.

Observation matrix estimation: Given an estimate of the state sequence, $\widehat{\mathbf{x}}_{1:T}$, the relationship between the observation matrix C and the innovation measurements is linear; i.e., $\widetilde{\mathbf{z}}_t = \widetilde{\Phi}_t C \widehat{\mathbf{x}}_t$. In addition, C is *time-invariant*. Hence, we can accumulate innovation measurements over a duration of time to stably reconstruct C. This significantly reduces the number of innovation measurements $\widetilde{M}$ required at *each* frame. This is especially important in the context of sensing videos, since the scene changes as we acquire measurements. Hence, requiring fewer measurements for each reconstructed frame of the video implies less error due to motion blur.

Using the estimates of the state sequence $\widehat{\mathbf{x}}_{1:T}$, we can recover C by solving the following convex problem:

$$\min \sum_{i=1}^{d} \|\Psi^T \mathbf{c}_i\|_1 \quad \text{s.t.} \quad \forall t,\ \|\mathbf{z}_t - \Phi_t C \widehat{\mathbf{x}}_t\|_2 \leqslant \epsilon, \tag{6}$$

where $\mathbf{c}_i$ denotes the ith column of C and Ψ is a sparsifying basis for the columns of C. Note that, in (6), we use *all* of the compressive measurements $\mathbf{z}_t$ obtained for each frame of the video—that is, we use both the common and innovation measurements, since the common measurements, much like the innovation measurements, are linear measurements of the frames. Further, as we show later in section 5.2, ambiguities in the estimation of the state sequence induce a structured sparsity pattern in the support of C. The convex program (6) can be modified to incorporate such constraints. In addition to this, in section 5, we also propose a greedy alternative for solving a variant of the convex program.

To summarize, the two-step measurement process described in (4) enables a two-step recovery (see Figure 2). First, we obtain an estimate of the state sequence using SVD on just the common measurements. Second, we use the state sequence estimate for recovering the observation matrix using a convex program. The details of these two steps are discussed in the next two sections.

4. Estimating the state sequence. In this section, we discuss methods to estimate the state sequence $\mathbf{x}_{1:T}$ from the compressive measurements $\breve{\mathbf{z}}_{1:T}$. In particular, we seek to establish sufficient conditions under which the state sequence can be estimated reliably.

4.1. Observability of the state sequence. Consider the compressive measurements given by

$$\breve{\mathbf{z}}_t = \breve{\Phi} y_t + \omega_t, \tag{7}$$

where $\breve{\mathbf{z}}_t \in \mathbb{R}^{\breve{M}}$ are the compressive measurements at time t, $\breve{\Phi} \in \mathbb{R}^{\breve{M} \times N}$ is the corresponding measurement matrix, and $\omega_t \in \mathbb{R}^{\breve{M}}$ is the measurement noise. Note that $\breve{\Phi}$ is time-invariant; hence, (7) is a part of the measurement model described in (4) relating to the common measurements. A key observation is that, when $\mathbf{y}_{1:T}$ form the observations of an LDS defined by (C, A), the compressive measurement sequence $\breve{\mathbf{z}}_{1:T}$ forms an LDS as well; that is,

$$\begin{aligned} \breve{\mathbf{z}}_t &= \breve{\Phi} y_t + \omega = \breve{\Phi} C \mathbf{x}_t + \omega'_t, \\ \mathbf{x}_t &= A\mathbf{x}_{t-1} + w_t. \end{aligned}$$

The LDS associated with $\breve{\mathbf{z}}_{1:T}$ is parameterized by the system matrices $(\breve{\Phi}C, A)$. Estimating the state sequence from the observations of an LDS is possible only when the LDS is *observable* [5]. Thus, it is important to consider the question of observability of the LDS parameterized by $(\breve{\Phi}C, A)$.[2]

Definition 4.1 (observability of an LDS [5]). *An LDS is* observable *if, for any possible state sequence, the current state can be estimated from a finite number of observations.*

Lemma 4.2 (test for observability of an LDS [5]). *An LDS defined by the system matrices (C, A) and of state space dimension d is observable if and only if the observability matrix*

$$(O(C, A))^T = \left[C^T \, (CA)^T \, (CA^2)^T \, \cdots \, (CA^{d-1})^T \right]^T \tag{8}$$

is full rank.

A necessary condition for the observability of the LDS defined by $(\breve{\Phi}C, A)$ is that the LDS defined by (C, A) is observable. However, for the LDSs we consider in this paper, $N \gg d$; for such systems, the LDS defined by (C, A) is observable. Given this assumption, we next consider the observability of the LDS parameterized by $(\breve{\Phi}C, A)$.

Lemma 4.3. *For $N > d$, the LDS defined by $(\breve{\Phi}C, A)$ is observable, with high probability, if $\breve{M} \geqslant d$ and the entries of the matrix $\breve{\Phi}$ are sampled i.i.d. from a sub-Gaussian distribution.*

Proof. This is established by proving that $\text{rank}(\breve{\Phi}C) = d$ when $\breve{M} \geqslant d$. Assume that $\text{rank}(\breve{\Phi}C) < d$; i.e., there exists $\alpha \in \mathbb{R}^d$ such that $\breve{\Phi}C\alpha = 0, \alpha \neq 0$. Let ϕ^T be a row of $\breve{\Phi}$. The event that $\phi^T C\alpha = 0$ is one of negligible probability when the elements of ϕ are assumed to be i.i.d. according to a sub-Gaussian distribution such as Gaussian or Bernoulli. Hence, with high probability, $\text{rank}(\breve{\Phi}C) = d$ when $\breve{M} \geqslant d$. ■

[2] Observability of LDSs in the context of CS has been studied earlier by Wakin, Sanandaji, and Vincent [46], who consider the scenario when the observation matrix C is randomly generated and the state vector $\mathbf{x}_0$ at $t = 0$ is sparse. In contrast, the analysis we present is for a nonsparse state vector.

Observability is the key criterion for recovering the state sequence from the common measurements. When the LDS associated with the common measurements is observable, we can estimate the state sequence—up to a linear transformation—by factorizing the block-Hankel matrix $\mathrm{Hank}(\breve{\mathbf{z}}_{1:T}, d)$ in (5). $\mathrm{Hank}(\breve{\mathbf{z}}_{1:T}, d)$ can be written as

$$\mathrm{Hank}(\breve{\mathbf{z}}_{1:T}, d) = O(\breve{\Phi}C, A)[\mathbf{x}_1 \; \mathbf{x}_2 \; \cdots \; \mathbf{x}_{T-d+1}].$$

Hence, when the observability matrix $O(\breve{\Phi}C, A)$ is full rank, we can recover the state sequence by factoring the Hankel matrix using the SVD. Suppose that the SVD of the Hankel matrix is $\mathrm{Hank}(\breve{\mathbf{z}}_{1:T}, d) = USV^T$. Then, the estimate of the state sequence is obtained by

$$(9) \qquad [\widehat{\mathbf{x}}_{1:T-d+1}] = S_d V_d^T,$$

where S_d is the diagonal matrix containing the d largest singular values in S, and V_d is the matrix composed of the right singular vectors corresponding to these singular values. The estimate of the state sequence obtained from SVD differs from its true value by a linear transformation. This is a fundamental ambiguity that stems from the lack of uniqueness in the definition of the state space (see section 2.3). The state sequence estimate in (9) can be improved, especially for high levels of measurement noise, by using system identification techniques mentioned in section 2.3. However, the simplicity of this estimate makes it amenable to further analysis.

When $\breve{M} > d$, we can choose to factorize a smaller-sized Hankel matrix $\mathrm{Hank}(\breve{\mathbf{z}}_{1:T}, q)$, provided $q > d/\breve{M}$. Note that when $q = 1$, we do not enforce the constraints provided by the state transition model, thereby simply reducing the LDS to a linear system. For $q > 1$, we enforce the state transition model over q successive time instants; i.e., we enforce

$$\mathbf{x}_t = A\mathbf{x}_{t-1} = A^2\mathbf{x}_{t-2} = \cdots = A^{q-1}\mathbf{x}_{t-q+1}, \qquad q \leqslant t \leqslant T.$$

Larger values of q lead to smoother state sequences, since the estimates conform to the state transition model for longer durations.

We next study the observability properties of specific classes of interesting LDSs and the conditions on $\breve{\Phi}$ under which the observability of $(\breve{\Phi}C, A)$ holds.

4.2. Case: $\breve{M} = 1$.

A particularly interesting scenario is when we obtain exactly one common measurement for each video frame. For such a scenario, $\breve{M} = 1$, and hence the measurement matrix can be written as a row-vector: $\breve{\Phi} = \phi^T \in \mathbb{R}^{1\times N}$. We now establish conditions when the observability matrix $O(\phi^T C, A)$ is full rank for this particular scenario. Let $\breve{\mathbf{c}} = (\phi^T C)^T = C^T\phi$ and $B = A^T$. We seek a condition when the observability matrix, or equivalently its transpose,

$$(10) \qquad \left(O(\breve{\mathbf{c}}^T, B^T)\right)^T = \left[\breve{\mathbf{c}} \;\; B\breve{\mathbf{c}} \;\; B^2\breve{\mathbf{c}} \;\; \cdots \;\; B^{d-1}\breve{\mathbf{c}}\right],$$

is full rank.[3] We concentrate on the specific scenario where the matrix B (and hence, A) is diagonalizable; i.e., $B = Q\Lambda Q^{-1}$, where $Q \in \mathbb{R}^{d\times d}$ is an invertible matrix (hence, full rank)

[3] There is an interesting connection to Krylov-subspace methods here. In Krylov-subspace methods, a low-rank approximation to a matrix K is obtained by forming the matrix $[\mathbf{c} \;\; K\mathbf{c} \;\; K^2\mathbf{c} \;\; \cdots]$ with $\mathbf{c}$ randomly chosen. Convergence proofs for this method are closely related to Theorem 4.4. To the best of our knowledge, diagonalizability of K plays an important role in most of these proofs. The interested reader is referred to [32] for more details.

and Λ is a diagonal matrix with diagonal elements $\{\lambda_i,\ 1 \leqslant i \leqslant d\}$. For such matrices, the transpose of the observability matrix can be written as

$$\begin{aligned}\left(O(\breve{\mathbf{c}}^T, B^T)\right)^T &= \left[\breve{\mathbf{c}}\ \ B\breve{\mathbf{c}}\ \ B^2\breve{\mathbf{c}}\ \ \cdots\ \ B^{d-1}\breve{\mathbf{c}}\right]\\ &= \left[QQ^{-1}\breve{\mathbf{c}}\ \ Q\Lambda Q^{-1}\breve{\mathbf{c}}\ \ \cdots\ \ Q\Lambda^{d-1}Q^{-1}\breve{\mathbf{c}}\right]\\ &= Q\left[\mathbf{e}\ \ \Lambda\mathbf{e}\ \ \Lambda^2\mathbf{e}\ \ \cdots\ \ \Lambda^{d-1}\mathbf{e}\right],\end{aligned}$$

where $\mathbf{e} = Q^{-1}\breve{\mathbf{c}}$. This can be expanded as

$$Q\begin{bmatrix} e_1 & \lambda_1 e_1 & \lambda_1^2 e_1 & \cdots & \lambda_1^{d-1} e_1\\ e_2 & \lambda_2 e_2 & \lambda_2^2 e_2 & \cdots & \lambda_2^{d-1} e_2\\ \vdots & \vdots & \vdots & \cdots & \vdots\\ e_d & \lambda_d e_d & \lambda_d^2 e_d & \cdots & \lambda_d^{d-1} e_d \end{bmatrix}$$

and further into

$$Q\begin{bmatrix} e_1 & & & 0\\ & e_2 & & \\ & & \ddots & \\ 0 & & & e_d \end{bmatrix}\begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{d-1}\\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{d-1}\\ \vdots & \vdots & \vdots & \cdots & \vdots\\ 1 & \lambda_d & \lambda_d^2 & \cdots & \lambda_d^{d-1} \end{bmatrix}.$$

We can establish a sufficient condition for when the observability matrix is full rank.

Theorem 4.4. *Let $\breve{M} = 1$, and let the elements of $\check{\Phi} = \phi^T$ be i.i.d. from a sub-Gaussian distribution. Then, with high probability, the observability matrix is full rank when the state transition matrix is diagonalizable and its eigenvectors and eigenvalues are unique.*

Proof. From the discussion above, the observability matrix can be written as a product of three square matrices: Q, the matrix of eigenvectors of A^T; a diagonal matrix with entries defined by the vector $\mathbf{e} = Q^{-1}C^T\phi$; and a Vandermonde matrix defined by the vector of eigenvalues of A $\{\lambda_i,\ 1 \leqslant i \leqslant d\}$. When the eigenvectors and eigenvalues are distinct, the first and last matrices are full rank. Given that the elements of ϕ are i.i.d., the probability that $e_i = 0$ is negligible, and hence the diagonal matrix is full rank with high probability. Since the product of full rank square matrices is full rank as well, this implies that the observability matrix is full rank with high probability. ■

Remark. Theorem 4.4 requires that the state-transition matrix be full rank (nonzero eigenvalues) and be diagonalizable with unique eigenvalues. Most matrices are diagonalizable (once we allow complex eigenvalues), and hence the requirement that state transition matrix be diagonalizable is not restrictive. A more restrictive condition is requiring that the eigenvalues of the matrix be unique. Unfortunately, this eliminates some commonly observed state transition matrices such as the identity matrix—which is coupled with Brownian processes. Nonetheless, Theorem 4.4 is intriguing, since it guarantees recovery of the state sequence even when we obtain only *one* common measurement per time instant. This is immensely useful in reducing the number of measurements required to sense a video sequence.

Interestingly, we can reduce $\breve{M}$ even further. This is achieved by not obtaining common measurements at some time instants.

4.3. Missing measurements: Case $\breve{M} < 1$. If we do not obtain common measurements at some time instants, then is it still possible to obtain an estimate of the state sequence? One way to view this problem is that we have incomplete knowledge of the Hankel matrix defined in (5) and we seek to *complete* this matrix. Matrix completion, especially for low-rank matrices, has received significant attention recently [6, 8, 30].

Given that the Hankel matrix $\text{Hank}(\breve{\mathbf{z}}_{1:T}, q)$ in (5) is low rank for videos modeled as LDSs, we formulate the missing measurement recovery problem as one of matrix completion. Suppose that we have the common measurements only at time instants given by the index set $\mathscr{I} \subset \{1, \dots, T\}$; i.e., we have knowledge of $\{\breve{\mathbf{z}}_i,\ i \in \mathscr{I}\}$. We can recover the missing measurements by exploiting the low-rank property of $\text{Hank}(\breve{\mathbf{z}}_{1:T}, q)$. Specifically, we solve the following problem to obtain the missing measurements:

$$\min \text{rank}(\text{Hank}(\mathbf{g}_{1:T}, q)) \quad \text{s.t.} \ \ \mathbf{g}_i = \breve{\mathbf{z}}_i,\ i \in \mathscr{I}.$$

However, rank$(\cdot)$ is a nonconvex function which renders the above problem NP-complete. In practice, we can solve a convex relaxation of this problem,[4]

$$\min \|\text{Hank}(\mathbf{g}_{1:T}, q)\|_* \quad \text{s.t.} \ \ \mathbf{g}_i = \breve{\mathbf{z}}_i,\ i \in \mathscr{I}, \tag{11}$$

where $\|H\|_*$ is the nuclear norm of the matrix H, which equals the sum of its singular values. Once we fill in the missing measurements, we use (9) to recover an estimate of the state sequence.

An important quantity to characterize is the proportion of time instants in which we can choose to not obtain common measurements. This amounts to developing a sampling theorem for the completion of low-rank Hankel matrices; to the best of our knowledge, there has been little theoretical work on this problem. Instead, we address it empirically in section 6.

5. Estimating the observation matrix.

In this section, we discuss estimation of the observation matrix C given the estimates of the state space sequence $\widehat{\mathbf{x}}_{1:T}$.

5.1. Need for innovation measurements. Given estimates of the state sequence $\widehat{\mathbf{x}}_{1:T}$, the matrix C is linear in the compressive measurements, which enables a host of conventional ℓ_2-based methods as well as ℓ_1-based recovery algorithms for estimating C. However, recall that C is an $N \times d$ matrix, and hence the common measurements by themselves are not enough to recover C unless $\breve{M}$ is large.

The common measurements $\breve{\mathbf{z}}_{1:T}$ used in the estimation of the state sequence are measured using a time-invariant measurement matrix $\breve{\Phi}$. A time-invariant measurement matrix, by itself, is not sufficient for estimating C unless $\breve{M}$ is very large. To alleviate this problem, we take additional compressive measurements of each frame using a time-varying measurement matrix. Let $\widetilde{\mathbf{z}}_t = \widetilde{\Phi}_t \mathbf{y}_t + \omega_t = \widetilde{\Phi}_t C \mathbf{x}_t + \omega_t$, where $\widetilde{\mathbf{z}}_t \in \mathbb{R}^{\widetilde{M}}$ and $\widetilde{\Phi}_t \in \mathbb{R}^{\widetilde{M} \times N}$ are the compressive measurements and the corresponding measurement matrix at time t. As mentioned earlier in section 3, we refer to these as innovation measurements. Noting that C is a time-invariant

[4] Historically, the use of nuclear norm-based optimization for system identification goes back to Fazel, Hindi, and Boyd [19, 20]. Since then, there has been much work towards establishing the equivalence of these two problems [6, 30]. Further, the convex program in (11) was used for video inpainting in [13].

parameter, we can collect innovation measurements over a period of time before reconstructing C. This enables a significant reduction in the number of measurements taken at each time instant.

5.2. Structured sparsity for C. Individual frames of a video, being images, exhibit sparsity/compressibility in certain transform bases such as wavelets and DCT (discrete cosine transform). If the supports of the frames are highly overlapping—this is to be expected, given the redundancies in a video—then columns of C are compressible in the same transform bases, a consequence of C being a basis for the frames of the video. Further, note that the columns of C are also the top principal components and, hence, capture the dominant motion patterns in the scene; when motion in the scene is spatially correlated, the columns of C are compressible in the wavelet/DCT basis. For these reasons, we assume that the columns of C are *compressible* in a wavelet/DCT basis and employ sparse priors in the recovery of the observation matrix C. We can potentially obtain an estimate of C by solving the following convex program:

$$(12) \qquad (P_{\ell_1}) \quad \min \sum_{i=1}^{d} \|\Psi^T \mathbf{c}_i\|_1 \;\text{ s.t }\; \forall t,\; \|\mathbf{z}_t - \Phi_t C \widehat{\mathbf{x}}_t\|_2 \leqslant \epsilon.$$

Here, we denote the columns of the matrix C as $\mathbf{c}_i, i = 1, \dots, d$. Ψ is a sparsifying basis for the columns of C; we have the freedom to choose different sparsifying bases for different columns of C.

The assumption of compressibility in a transform basis was sufficient for all the videos we tested (see section 6). However, it is entirely possible that a video is not compressible in a transform basis. There are two possible ways to address such a scenario. First, given training data, we can use dictionary learning algorithms [24] to learn an appropriate basis in which the columns of C are sparse/compressible. Second, in the absence of training data, we revert to ℓ_2-based methods to recover C; in such cases, we would typically need more measurements to recover C.

However, the convex program (P_{ℓ_1}) is not sufficient as-is to recover C. The reason for this stems from ambiguities in the definition of the LDS (see section 2.3). The use of SVD for recovering the state sequence introduces an ambiguity in the estimates of the state sequence in the form of $[\widehat{\mathbf{x}}_{1:T}] \approx L^{-1}[\mathbf{x}_{1:T}]$, where L is an invertible $d \times d$ matrix. As a consequence, this will lead to an estimate $\widehat{C} = CL$ satisfying $\mathbf{z} = \Phi\widehat{C}\widehat{\mathbf{x}}_t = \Phi(CL)(L^{-1}\mathbf{x}_t) = \Phi C \mathbf{x}_t$. Suppose the columns of C are K-sparse (equivalently, compressible for a certain value of K) each in Ψ with support $\mathcal{S}_k$ for the kth column. Then, the columns of CL are potentially dK-sparse with identical supports $\mathcal{S} = \bigcup_k \mathcal{S}_k$. The support is exactly dK-sparse when the $\mathcal{S}_k$ are disjoint and L is dense. At first glance, this seems to be a significant drawback, since the overall sparsity of $\widehat{C}$ has increased to d^2K (the sparsity of C is dK). However, this apparent increase in sparsity is alleviated by the columns having identical supports, which can be exploited in the recovery process [17].

Given the estimates $\widehat{\mathbf{x}}_{1:T}$, we estimate the matrix C by solving the following convex program:

$$(13) \qquad (P_{\ell_2-\ell_1}) \; \min \sum_{i=1}^{N} \|\mathbf{s}_i\|_2 \quad \text{s.t } C = \Psi S \text{ and } \forall t,\; \|\mathbf{z}_t - \Phi_t C \widehat{\mathbf{x}}_t\|_2 \leqslant \epsilon,$$

where $\mathbf{s}_i$ is the ith row of the matrix $S = \Psi^T C$ and Ψ is a sparsifying basis for the columns of C. The above problem is an instance of an $\ell_2 - \ell_1$ mixed-norm optimization that promotes group sparsity; in this instance, we use it to promote group column sparsity in the matrix S; i.e., all columns have the same sparsity pattern.

There are multiple efficient ways to solve $(P_{\ell_2 - \ell_1})$, including solvers such as $SPG - L1$ [38] and model-based CoSAMP [3]. Algorithm 1 summarizes a model-based CoSAMP algorithm used for recovering the observation matrix C. The specific model used here is a union-of-subspaces model that groups each row of $S = \Psi^T C$ into a single subspace/model.

Algorithm 1: $\widehat{C}$ = Model-based CoSAMP $(\Psi, K, \mathbf{z}_t, \widehat{\mathbf{x}}_t, \Phi_t, t = 1, \ldots, T)$

```
Notation:
    supp($vec; K$) returns the support of the $K$ largest elements of $vec$.
    $A_{|\Omega,\cdot}$ represents the submatrix of $A$ with rows indexed by $\Omega$ and all columns.
    $A_{|\cdot,\Omega}$ represents the submatrix of $A$ with columns indexed by $\Omega$ and all rows.
Initialization
    $\forall t, \Theta_t \leftarrow \Phi_t \Psi$.
    $\forall t, \mathbf{v}_t \leftarrow \mathbf{0} \in \mathbb{R}^M$.
    $\Omega_{\text{old}} \leftarrow \phi$.
while (stopping conditions are not met) do
    Compute signal proxy:
        $R = \sum_t \Theta_t^T \mathbf{v}_t \widehat{\mathbf{x}}_t^T$.
    Compute energy in each row:
        $k \in [1, \ldots, N],\ \mathbf{r}(k) = \sum_{i=1}^{d} R^2(k, i)$.
    Support identification and merger:
        $\Omega \leftarrow \Omega_{\text{old}} \bigcup \text{supp}(\mathbf{r}; 2K)$.
    Least squares estimation:
        Find $A \in \mathbb{R}^{|\Omega| \times d}$ that minimizes $\sum_t \|\mathbf{z}_t - (\Theta_t)_{|\cdot,\Omega} A \widehat{\mathbf{x}}_t\|_2$.
        $B_{|\Omega,\cdot} \leftarrow A,\ B_{|\Omega^c,\cdot} \leftarrow 0$.
    Pruning support:
        $k \in [1, \ldots, N],\ \mathbf{b}(k) = \sum_{i=1}^{d} B^2(k, i)$.
        $\Omega \leftarrow \text{supp}(\mathbf{b}; K),\ S_{|\Omega,\cdot} \leftarrow B_{|\Omega,\cdot},\ S_{|\Omega^c,\cdot} \leftarrow 0$.
    Form new estimate of $C$:
        $\widehat{C} \leftarrow \Psi S$.
    Update residue:
        $\forall t, \mathbf{v}_t \leftarrow \mathbf{z}_t - \Theta_t S \widehat{\mathbf{x}}_t$.
        $\Omega_{\text{old}} \leftarrow \Omega$.
end
```

5.3. Value of $\widetilde{M}$. For stable recovery of the observation matrix C, we need in total $O(dK \log(N/K))$ measurements; for a large class of practical solvers, a rule of thumb is $4dK \log(N/K)$. Given that we measure $\widetilde{M}$ time-varying compressive measurements at each time instant, over a period of T time instants, we have $\widetilde{M}T$ compressive measurements for estimating C. Hence, for stable recovery of C, we need approximately

$$\widetilde{M}T = 4dK\log(N/K) \implies \widetilde{M} = 4\frac{dK}{T}\log(N/K). \tag{14}$$

This indicates extremely favorable operating scenarios for the CS-LDS framework, especially when T is large (as in high frame rate capture). Let $T = \tau f_s$, where τ is the time duration of the video in seconds and f_s is the sampling rate of the measurement device. The number of compressive measurements required in this case is $\widetilde{M} = 4\frac{dK}{\tau f_s}\log\left(\frac{N}{K}\right)$. Given that the complexity of the LDS typically (however, not always) depends on τ, for a fixed τ the number of measurements required to estimate C decreases as $1/f_s$ as the sampling rate f_s is increased. Indeed, as the sampling rate f_s increases, $\widetilde{M}$ can be decreased while keeping Mf_s constant. This will ensure that (14) is satisfied, enabling stable recovery of C.

5.4. Mean + LDS. In many instances, a dynamical scene is modeled better as an LDS over a static background, that is, $\mathbf{y}_t = C\mathbf{x}_t + \mu$. This can be handled with two small modifications to Algorithm 1. First, the state sequence $[\hat{\mathbf{x}}_{1:T}]$ is obtained by performing an SVD on the matrix $\text{Hank}(\check{z}_{1:T}, d_{\text{guess}})$ modified such that each row sums to zero. This works under the assumption that the sample mean of $\check{z}_{1:T}$ is equal to $\check{\Phi}\mu$, the compressive measurement of μ. Second, given that the support of μ need not be similar to that of C, the resulting optimization problem can be reformulated as

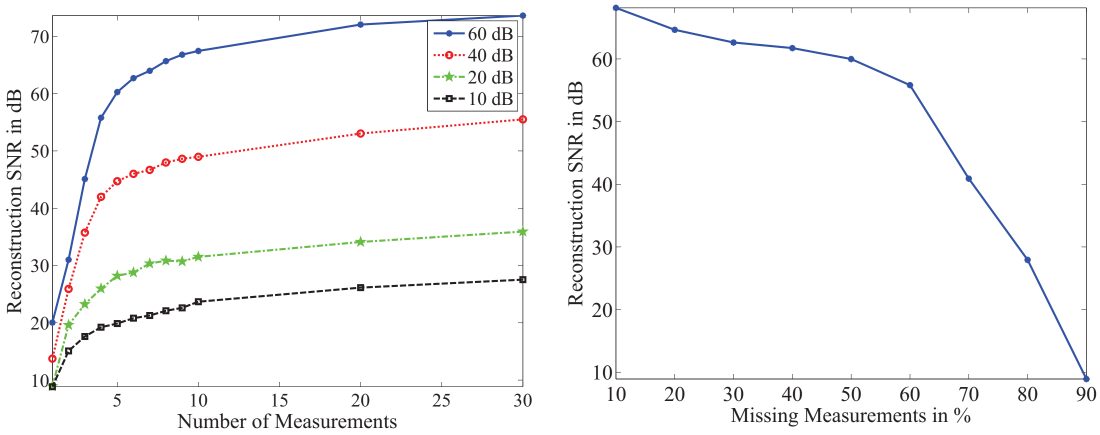
$$(P_{\mu,\ell_2-\ell_1})\ \min \|\Psi^T\mu\|_1 + \sum_{i=1}^{N}\|\mathbf{s}_i\|_2 \quad \text{s.t } C = \Psi S \text{ and } \forall t,\ \|\widetilde{\mathbf{z}}_t - \widetilde{\Phi}_t(\mu + C\widehat{\mathbf{x}}_t)\|_2 \leqslant \epsilon. \tag{15}$$

As with the convex formulation, the model-based CoSAMP algorithm described in Algorithm 1 can be modified to incorporate the mean term μ; an additional modification here is the requirement to specify a priori the sparsity of the mean $K_\mu = \|\Psi^T\mu\|_0$.

6. Experiments. We present a range of experiments validating various aspects of the CS-LDS framework. We use permuted noiselets [12] for the measurement matrices, since they have a fast scalable implementation. We use the term *compression ratio* N/M to denote the reduction in the number of measurements as compared to the Nyquist rate. Finally, we use the reconstruction signal-to-noise ratio (SNR) to evaluate the recovered videos. Given the ground truth video $\mathbf{y}_{1:T}$ and a reconstruction $\widehat{\mathbf{y}}_{1:T}$, the reconstruction SNR in dB is defined by

$$10\log_{10}\left(\frac{\sum_{t=1}^{T}\|\mathbf{y}_t\|_2^2}{\sum_{t=1}^{T}\|\mathbf{y}_t - \widehat{\mathbf{y}}_t\|_2^2}\right). \tag{16}$$

We compare CS-LDS against *frame-by-frame* CS, where each frame of the video is recovered separately using conventional CS techniques. We use the term *oracle LDS* when the parameters and video reconstruction are obtained by operating on the original data itself. Oracle LDS estimates the parameters using a rank-d approximation of the ground truth data.



(a) $\breve{M} \geqslant 1$ (b) $\breve{M} < 1$

Figure 3. *Accuracy of state sequence estimation from common measurements. Shown are aggregate results over* 100 *Monte Carlo runs for an LDS with* $d = 10$ *and* $T = 500$*. For each Monte Carlo run, the system matrices and the state sequence were generated randomly.* (a) *Reconstruction SNR as a function of the number of common measurements* $\breve{M}$ *per frame. Each curve is for a different level of measurement noise as measured using input SNR. For low noise levels, we obtain a good reconstruction SNR* ($> 20dB$) *even at* $\breve{M} = 1$*; this hints at very high compression ratios.* (b) *Reconstruction SNR of the Hankel matrix for the scenario with missing common measurements. We can estimate the Hankel matrix very accurately even at* 80% *missing measurements. This suggests immense flexibility in the implementation of the CS-LDS system.*

The reconstruction SNR of the oracle LDS gives an upper bound on the achievable SNR. Finally, the ambiguity in the observation matrix (due to nonuniqueness of the SVD-based factorization) as estimated by oracle LDS and CS-LDS is resolved by finding the best $d \times d$ linear transformation that registers the two estimates.

6.1. State sequence estimation. We first provide empirical verification of the results derived in sections 4.1 and 4.2. It is worth noting that, in the absence of noise, Theorem 4.4 suggests exact recovery of the state sequence. In practice, it is important to check the robustness of the estimate to measurement noise. Figure 3(a) analyzes the performance of the state space estimation for different values of the number of common measurements $\breve{M}$ and different SNRs of the measurement noise. We define input SNR in dB as $10\log_{10}((\sum \|\mathbf{y}_t\|_2^2)/(T\sigma^2))$, where σ is the standard deviation of the noise. Here, we consider the scenario when $\breve{M} \geqslant 1$. The underlying state space dimension is $d = 10$ with $T = 500$ frames. As expected, for low SNRs, the reconstruction SNR is very high even for small values of $\breve{M}$. In addition to this, the accuracy at $\breve{M} = 1$ is acceptable, especially at low SNRs.

Next, we validate the implications of section 4.3, where we discuss the scenario of $\breve{M} < 1$ by simulating various proportions of missing common measurements. Figure 3(b) shows reconstruction SNR for the Hankel matrix in (5) for varying amounts of missing measurements. We recover the Hankel matrix by solving (11) using CVX [21]. Figure 3(b) demonstrates a very high reconstruction SNR even at a very high rate of missing measurements. As mentioned earlier, not having to sense common measurements at all frames is very useful, since we can stagger our acquisition of common and innovation measurements. In theory, this enables a measurement strategy where we need to sense only one measurement per frame of the

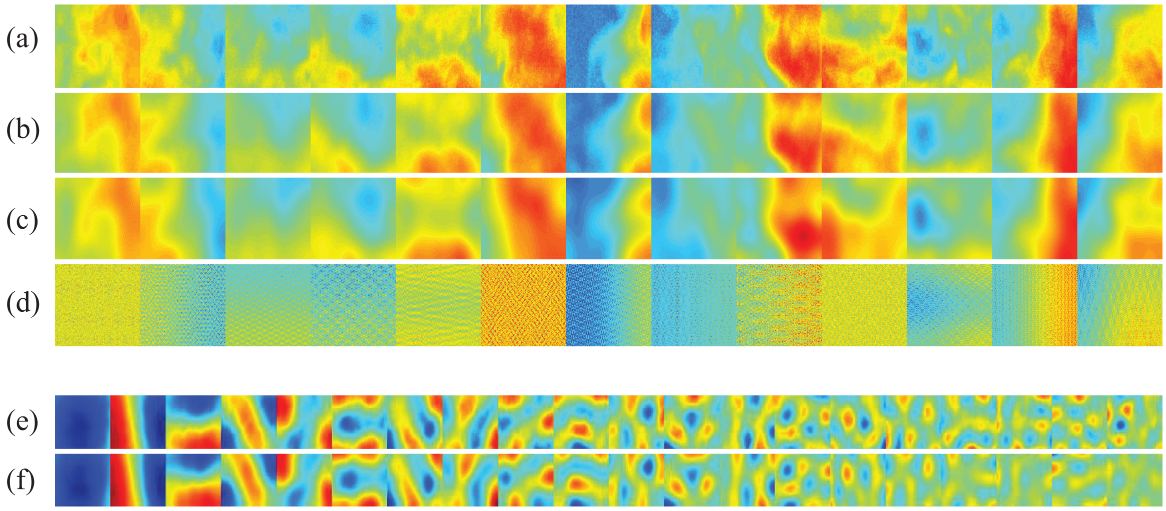


Figure 4. *Reconstruction of a fire texture of length* 250 *frames and resolution of* $N = 128 \times 128$ *pixels.* (a)–(d) *Sampling of frames of the* (a) *ground truth video,* (b) *oracle LDS reconstruction,* (c) *CS-LDS reconstruction, and* (d) *naive frame-to-frame CS reconstruction. The CS-LDS reconstruction closely resembles the oracle LDS result. For the CS-LDS results, compressive measurements were obtained at* $\breve{M} = 30$ *and* $\widetilde{M} = 40$ *measurements per frame, thereby giving a compression ratio of* 234×. *Reconstruction was performed with* $d = 20$ *and* $K = 30$. (e) *Ground truth observation matrix* C. (f) *CS-LDS estimate of the observation matrix* $\widehat{C}$. *In* (e) *and* (f), *the column of the observation matrix is visualized as an image. Both the frames of the videos and the observation matrices are shown in false color for better contrast.*

video without having to group consecutive measurements of the SPC. Hence, we can aim to reconstruct videos at the sampling rate of the SPC. To the best of our knowledge, this is the first video CS acquisition design capable of doing this.

6.2. Dynamic textures.

Our test dataset comprises videos from the DynTex dataset [29]. We used the mean+LDS model from section 5.4 for all the video CS experiments, with the 2D DCT as the sparsifying basis for the columns of C and 2D wavelets as the sparsifying basis for the mean. We used the model-based CoSAMP solver in Algorithm 1 for these results, since it provides explicit control of the sparsity of the mean and the columns of C. We used (14) as a guide to select these values.

Figure 4 shows video reconstruction of a dynamic texture from the DynTex dataset [29]. Reconstruction results are under a compression $N/M = 234$; this is an operating point where frame-to-frame CS recovery is completely infeasible. However, the dynamic component of the scene is relatively small ($d = 20$), which allows us to recover the video from relatively few measurements. The reconstruction SNRs of the recovered videos shown are as follows: oracle LDS = 24.97 dB, frame-to-frame CS = 11.75 dB, and CS-LDS = 22.08 dB.

Figure 5 shows the reconstruction of a video, of six blinking LED lights, from the DynTex dataset. We show reconstruction results at different compression ratios as well as different image resolutions. It is noteworthy that, even at a 100× compression, the reconstruction at a resolution of 256 × 256 pixels preserves fine details.

Performance with measurement noise. We validate the performance of our recovery algorithm under various amounts of measurement noise. Note that the columns of C

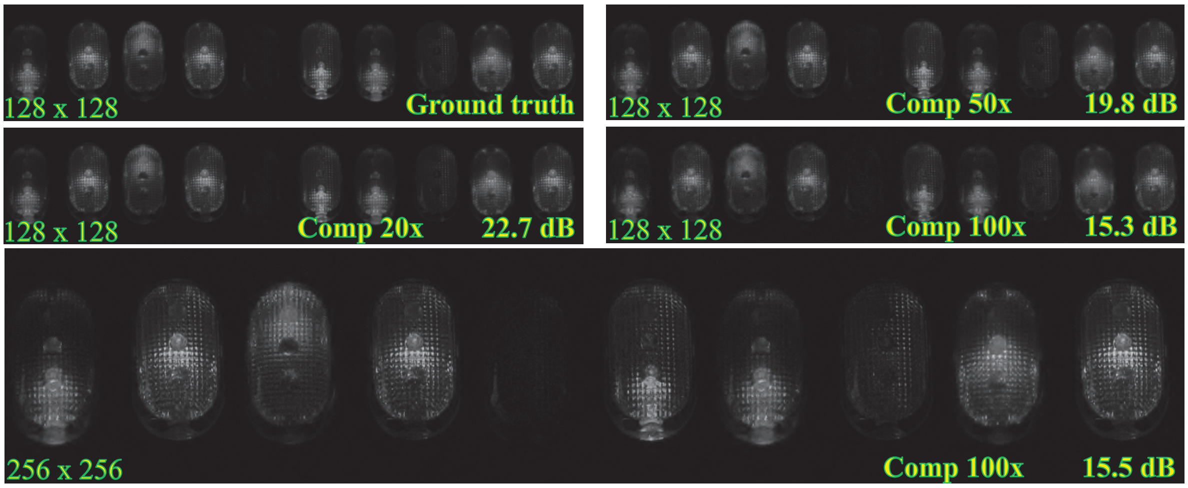
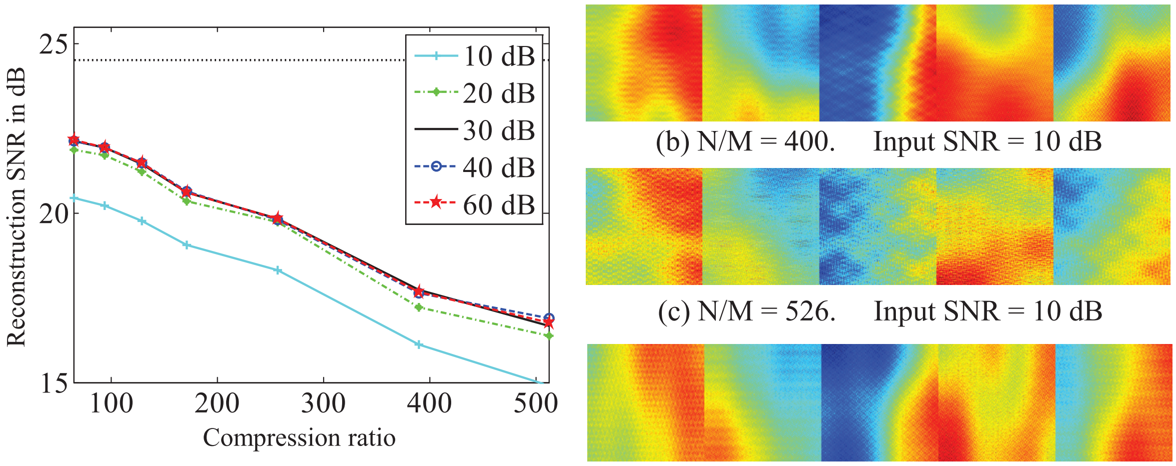


Figure 5. *Reconstruction of a video composed of six blinking LED lights. We used* $d = 7$, $\breve{M} = 3d$, *and* $\widetilde{M}$ *chosen based on the overall compression ratio* $N/(\breve{M} + \widetilde{M})$. *Each row shows a sampling of frames of the video reconstructed at a different compression ratios. Inset in each row is the resolution of the video used as well as the compression at sensing and the reconstruction SNR. While performance degrades with increasing compression, it also gains significantly for higher-dimensional data; the reconstruction at* 256×256 *pixels preserves finer details.*

with larger singular values are, inherently, better conditioned to deal with this measurement error. The columns corresponding to the smaller singular values are invariably estimated with higher error. Figure 6 shows the performance of the recovery algorithm for various levels of measurement noise. The effect of the measurement noise on the reconstructions is perceived only at low input SNRs. In part, this robustness to measurement noise is due to the LDS model mismatch dominating the reconstruction error at high input SNRs. As the input SNR drops significantly below the model mismatch term, predictably, it starts influencing the reconstructions more. This provides a certain amount of flexibility in the design of potential CS-LDS cameras.

Computation time and spatial resolution. Figure 7 shows the recovery algorithm applied to a video of length 560 frames at different spatial resolutions. Shown in Figure 7 are the amount of time taken for each recovery, which scales gracefully for increasing spatial resolution, and reconstruction SNR, which approaches the performance of an oracle LDS. The improvement in reconstruction comes due to the increase in the number of compressive measurements at high resolutions, since the compression ratio is held fixed. However this comes at the cost of requiring a faster compressive camera to acquire the data, since a larger number of measurements is required.

Gallery of results. Finally, in Figure 8, we demonstrate performance of the CS-LDS methodology for sensing and reconstructing a wide range of videos. The reader is directed to the supplemental material (86330_01.avi [local/web 2.32MB], 86330_02.avi [local/web 1.64MB], 86330_03.avi [local/web 1.97MB], 86330_04.avi [local/web 3.13MB], 86330_05.avi [local/web 327KB], 86330_06.avi [local/web 329KB], 86330_07.avi [local/web 3.38MB], 86330_08.avi [local/web 802KB], and 86330_09.avi [local/web 1.94MB]) as well as the project Web page [1] for animated videos of these results.



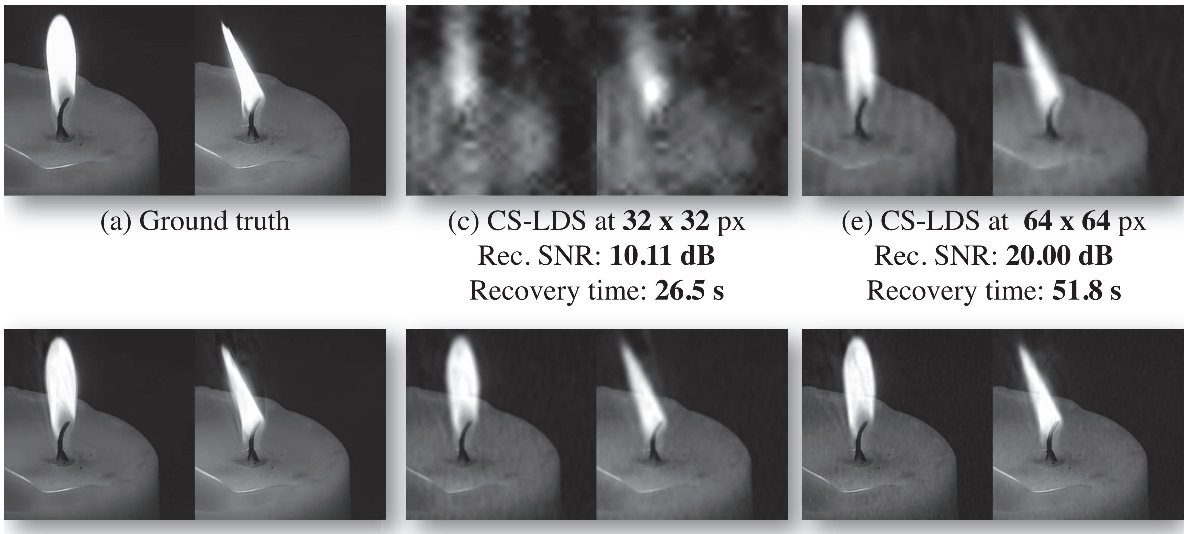
(a) Performance

(b) N/M = 400. Input SNR = 10 dB

(c) N/M = 526. Input SNR = 10 dB

(d) N/M = 526. Input SNR = 30 dB

Figure 6. *Resilience of the CS-LDS framework to measurement noise.* (a) *Performance plot charting the reconstruction SNR as a function of compression ratio* N/M. *Each curve is for a different level of measurement noise as measured using input SNR. Reconstruction SNRs were computed using* 32 *Monte Carlo simulations. The black-dotted line shows the reconstruction SNR for a* $d = 20$ *oracle LDS.* (b)–(d) *Snapshots of video frames at various operating points. The dynamic texture of Figure* 4 *was used for this result.*



(a) Ground truth

(c) CS-LDS at **32 x 32** px
Rec. SNR: **10.11 dB**
Recovery time: **26.5 s**

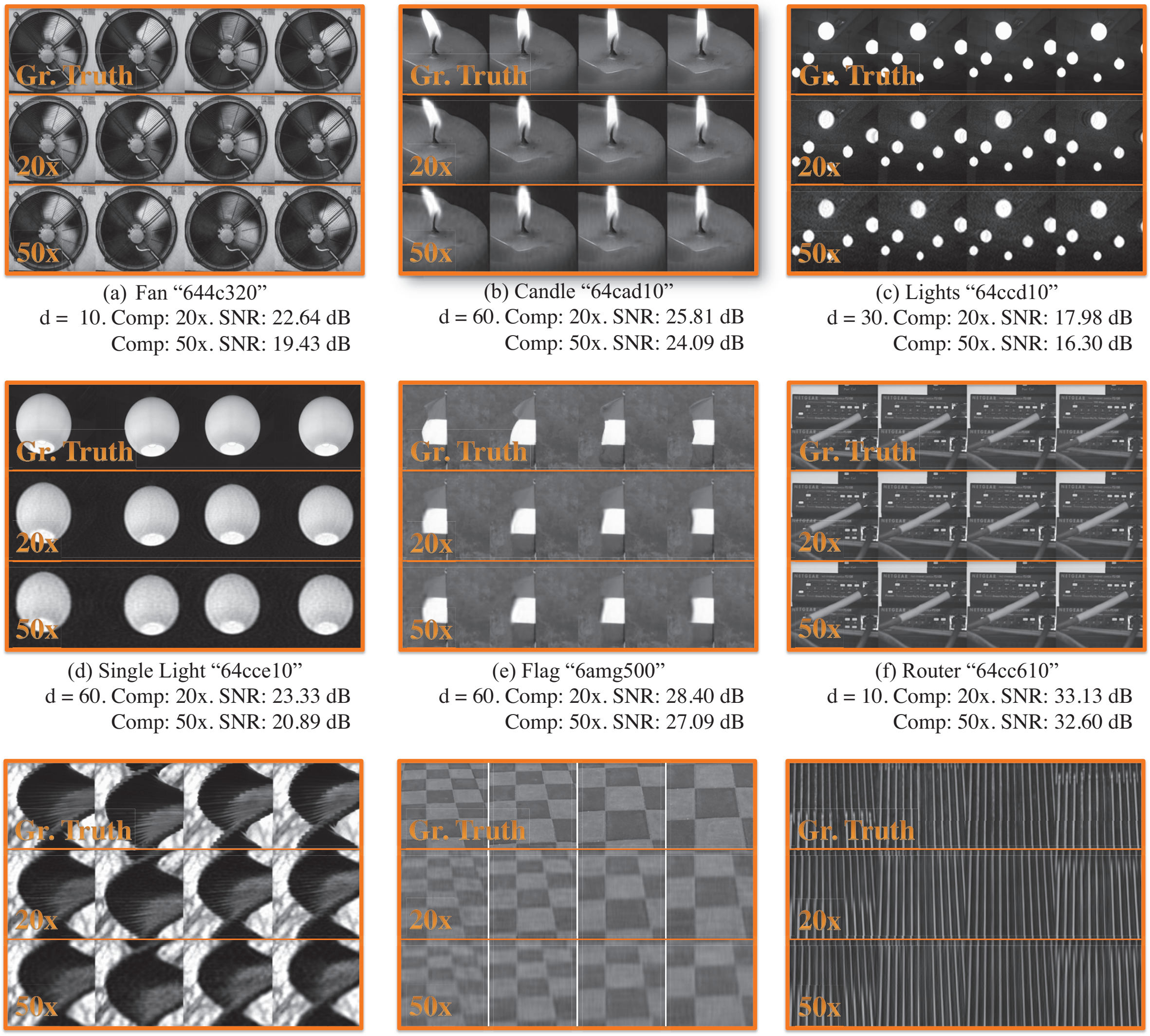
(e) CS-LDS at **64 x 64** px
Rec. SNR: **20.00 dB**
Recovery time: **51.8 s**

(b) Oracle LDS at **d = 50**
Rec. SNR: **24.62 dB**

(d) CS-LDS at **128 x 128** px
Rec. SNR: **22.06 dB**
Recovery time: **290.2 s**

(f) CS-LDS at **256 x 256** px
Rec. SNR: **22.12 dB**
Recovery time: **1582.1 s**

Figure 7. *Reconstruction of a video at different spatial resolutions.* (a) *Two frames from the ground truth video of* 560 *frames.* (b) *Reconstructed frame using an oracle with* $d = 50$. (c)–(f) *CS-LDS reconstructions for varying spatial resolution, at a compression of* 20× *and with* $d = 50$. *Shown are reconstruction SNR as well as recovery times for each reconstruction. Note that as the spatial resolution increases, the reconstruction performance increases and approaches the performance of oracle LDS. However, for the same compression, recovering at a higher resolution also requires a compressive camera capable of sampling faster.*



(a) Fan “644c320”
d = 10. Comp: 20x. SNR: 22.64 dB
Comp: 50x. SNR: 19.43 dB

(b) Candle “64cad10”
d = 60. Comp: 20x. SNR: 25.81 dB
Comp: 50x. SNR: 24.09 dB

(c) Lights “64ccd10”
d = 30. Comp: 20x. SNR: 17.98 dB
Comp: 50x. SNR: 16.30 dB

(d) Single Light “64cce10”
d = 60. Comp: 20x. SNR: 23.33 dB
Comp: 50x. SNR: 20.89 dB

(e) Flag “6amg500”
d = 60. Comp: 20x. SNR: 28.40 dB
Comp: 50x. SNR: 27.09 dB

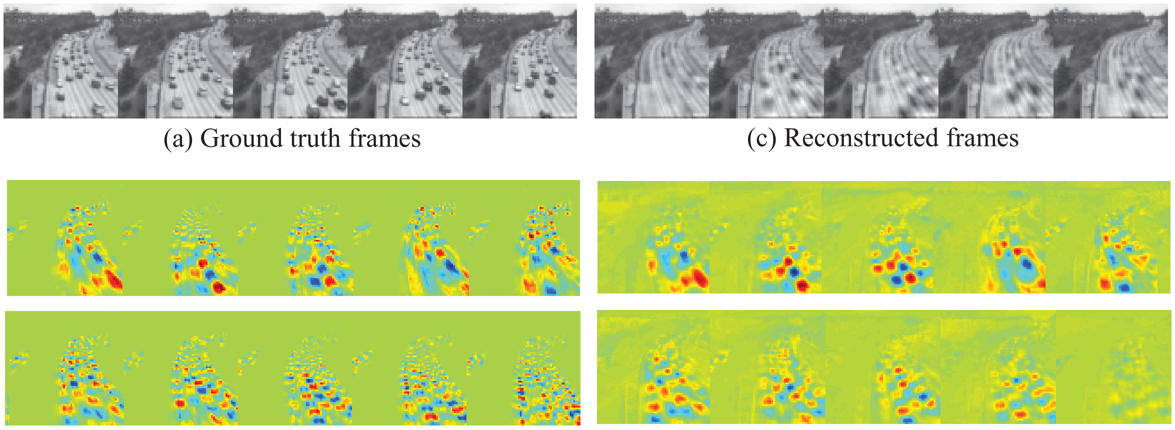
(f) Router “64cc610”
d = 10. Comp: 20x. SNR: 33.13 dB
Comp: 50x. SNR: 32.60 dB

(g) Spiral “645a910”
d = 60. Comp: 20x. SNR: 22.48 dB
Comp: 50x. SNR: 19.68 dB

(h) Tiles “649j210”
d = 60. Comp: 20x. SNR: 24.22 dB
Comp: 50x. SNR: 23.55 dB

(i) Stairs “54pe210”
d = 60. Comp: 20x. SNR: 20.16 dB
Comp: 50x. SNR: 17.32 dB

Figure 8. *A gallery of reconstruction results using the CS-LDS framework. Each subfigure* (a)–(i) *shows reconstruction results for a different video. The three rows of each sub-figure correspond to, from top-bottom, the ground truth video and CS-LDS reconstructions at compression ratios of* $20\times$ *and* $50\times$*. Each column is a frame of the video and its reconstruction. Also noted with each reconstruction is the value of* d *and the reconstruction SNR for that result. All videos are from the DynTex dataset* [29] *downsampled at a spatial resolution of* 256×256 *pixels. The “code” in quotes refers to the name of the sequence in the database. For all videos,* $\breve{M} = 3d$*. Results are best viewed under the “zoom” tool. The interested reader is directed to the project Web page* [1] *and the supplemental material* (86330_01.avi [local/web 2.32MB], 86330_02.avi [local/web 1.64MB], 86330_03.avi [local/web 1.97MB], 86330_04.avi [local/web 3.13MB], 86330_05.avi [local/web 327KB], 86330_06.avi [local/web 329KB], 86330_07.avi [local/web 3.38MB], 86330_08.avi [local/web 802KB], and 86330_09.avi [local/web 1.94MB]) *for videos of these results.*



(a) Ground truth frames

(c) Reconstructed frames

(b) Observation matrix

(d) Estimated observation matrix

Figure 9. *Reconstructions of a traffic scene of $N = 64 \times 64$ pixels at a compression ratio $N/M = 25$, with $d = 15$ and $K = 40$.* (a), (c) *Sampling of frames of the ground truth and reconstructed video.* (b), (d) *The first ten columns of the observation matrix C and the estimated matrix $\widehat{C}$; both are shown in false color for improved contrast. The quality of reconstruction and LDS parameters is sufficient for capturing the flow of traffic, as seen in the classification results in Table 1.*

6.3. Application in activity analysis. As mentioned in section 2.3, LDSs are often used in classification problems, especially in the context of scene/activity analysis. A key experiment in this context is to check whether the CS-LDS framework recovers videos that are sufficiently informative for such applications. To this end, we experiment with two different activity analysis datasets: the UCSD Traffic Dataset [10] and the UMD Human Activity Dataset [42].

Activity recognition methodology. In both the scenarios considered here (single human activity and traffic), we model the observed video using the linear dynamical model framework. For recognition, we used the Procrustes distance [11] between the column spaces of the observability matrices in conjunction with a nearest-neighbor classifier. Given the observability matrix $O(C, A)$ defined in (8), let Q be an orthonormal matrix such that $\text{span}(Q) = \text{span}(O(C, A))$. Given two LDSs, the squared Procrustes distance between them is given by

$$d^2(Q_1, Q_2) = \min_{R \in \mathbb{R}^{d \times d}} \text{tr}(Q_1 - Q_2 R)^T (Q_1 - Q_2 R),$$

where $\text{span}(Q_1) = \text{span}(O(C_1, A_1))$ and $\text{span}(Q_2) = \text{span}(O(C_2, A_2))$. We use this distance function in a nearest-neighbor classifier in both the activity classification experiments.

The UCSD Traffic Dataset [10] consists of 254 videos capturing car traffic of three types: light, moderate, and heavy. Each video is of length 50 frames at a resolution of 64×64 pixels. Figure 9 shows the reconstruction results on a traffic sequence from the dataset. We perform a classification experiment of the videos into these three categories. There are four different train-test scenarios provided with the dataset. For comparison, we also perform the same experiments with fitting the LDS model on the original frames (oracle LDS). We perform classification at two different values of the state space dimension d and at a fixed

Table 1
Classification results (in %) on the UCSD Traffic Dataset.

	Expt 1	Expt 2	Expt 3	Expt 4	Average
(d = 10)					
Oracle LDS	85.71	85.93	87.5	92.06	87.8%
CS-LDS	84.12	87.5	89.06	85.71	86.59%
(d = 5)					
Oracle LDS	77.77	82.81	92.18	80.95	83.42%
CS-LDS	85.71	73.43	78.1	76.1	78.34%
State KL-SVM (d = 10) [10]	n.a.	n.a.	n.a.	n.a.	93%
State KL-SVM (d = 5) [10]	n.a.	n.a.	n.a.	n.a.	87%

Table 2
Classification results (in %) on the UMD Human Activity Database.

Activity	100×	150×	200×	Shape dynamics [44]
Pickup object	100	100	100	100
Jog	100	100	90	100
Push	100	90	50	100
Squat	90	100	100	100
Wave	100	100	60	100
Kick	100	90	80	100
Bend	100	100	100	100
Throw	100	100	90	100
Turn around	100	100	100	100
Talk on cellphone	100	20	10	100
Average	94%	90%	78%	100%

compression ratio of 25×. Table 1 shows classification results. We also show comparative results obtained using a probabilistic kernel on dynamic texture models [10] in conjunction with support vector machines (SVMs) in the last two rows of the table. Results for each individual experiment were not reported; only an aggregate number was reported, which is shown in the table. It can be seen that, even without sophisticated nonlinear classifiers, we are able to obtain comparable performance using a simple nearest-neighbor classifier using the dynamic texture model parameters. This shows that the obtained parameters possess discriminatory properties and can be used in conjunction with other sophisticated classifiers that build on dynamic texture models as in [10].

The UMD Human Activity Dataset [42] consists of 100 videos, each of length 80 frames, depicting 10 different activities: *pickup object, jog, push, squat, wave, kick, bend, throw, turn around, and talk on cellphone.* Each activity was repeated 10 times, so there were a total of 100 sequences in the dataset. As with the traffic experiment, we use an LDS model on the image intensity values without any feature extraction. Images were cropped to contain the human and resized to 330 × 300. The state space dimension was fixed at $d = 5$, and the compression was varied from 50× to 200×. We performed a leave-one-execution-out test. The results are summarized in Table 2. As can be seen, the CS-LDS framework obtained a classification performance that is comparable to the oracle LDS. For this dataset, both oracle LDS and CS-LDS obtained a perfect classification score of 100% up to a compression ratio of 50×. Further, as shown in Table 2, we obtain performance comparable to a far more sophisticated method employing advance shape-based features for activity recognition. This suggests that the CS-LDS framework should be extremely useful in a wide range of applications beyond just video recovery, and can provide a basis to acquiring more sophisticated features for tackling challenging activity recognition problems.

7. Discussion.

In this paper, we have proposed a framework for the compressive acquisition of dynamic scenes modeled as LDSs. In particular, this paper emphasizes the power of predictive/generative video models. In this regard, we have shown that a strong model for the scene dynamics enables stable video reconstructions at very low measurement rates. In particular, it enables the estimation of the state sequence associated with a video even at fractional number of common measurements per video frame ($\breve{M} \leqslant 1$). The use of CS-LDS for dynamic scene modeling and classification also highlights the purposive nature of the framework.

Implementation issues: The results provided in the paper are mainly based on simulations. While a full-fledged implementation on hardware is beyond the scope of this paper, we discuss some of the key issues and challenges in obtaining such results. Focusing on the single pixel camera (SPC) as our imaging architecture, the achievable compression and resolution are limited by the amount of motion in the scene and the sampling rate of the camera. We discuss the roles these two parameters play in practice.

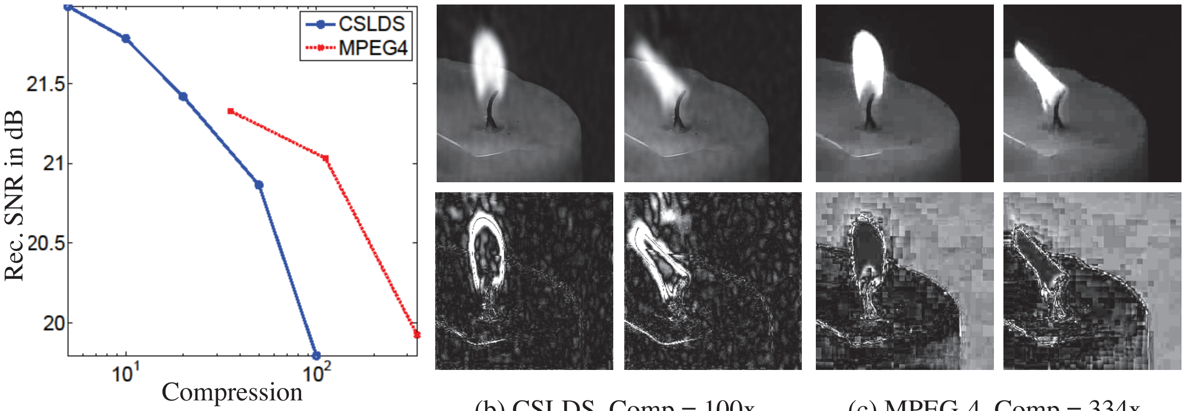
Amount of motion determines an inherent notion of frame-rate of the video; note that real-life scenes have no notion of "frame-rate." If the scene changes negligibly for a time duration τ, then $1/\tau$ (for the largest value of τ) becomes a good measure of frame-rate for a scene. For example, static scenes do not change over an infinite time duration ($\tau = \infty$) and, hence, can be sensed at $1/\tau = 0$ measurements per second. Given that we seek to sense this scene at a spatial resolution of N pixels, a Nyquist camera would need to operate at N/τ measurements per second.

Suppose that this scene over a duration of T seconds can be well approximated by a d-dimensional LDS; then the total number of free variables to estimate is approximately $d(T/\tau)$ for the state sequence and Kd for the observation matrix. An SPC operating at $1/f_s$ samples per second obtains a total of Tf_s compressive measurements. If CS-LDS were employed at a compression ratio of C, then

$$CTf_s \geqslant c_0 \frac{dT}{\tau} + c_1 Kd \log(N/K) \implies f_s \geqslant c_0 \frac{d}{C\tau} + c_1 \frac{Kd}{CT} \log(N/K).$$

The key dependencies here are on how d, K, and τ change as functions of N. In particular, even if d and K increase as $\sqrt{N}$, then f_s would need to scale linearly in N to maintain the same compression level.

Connection to affine-rank minimization: The pioneering work of Fazel [18] in developing convex optimization techniques to system identification problems has interesting parallels to the ideas proposed in this paper. One of the key ideas espoused in [18] is that, when the video sequence $\mathbf{y}_{1:T}$ is an LDS, the block-Hankel matrix $\mathrm{Hank}(\mathbf{y}_{1:\mathrm{T}}, \mathrm{q})$ is low rank. When we have linear measurements of the video frames, we can solve an affine-rank problem to recover the video. However, such methods optimize on the Hankel matrix directly and lead to computationally infeasible designs even for videos of very small dimensions. In contrast, CS-LDS has been shown to be fast and computationally feasible for very large videos involving



(a)

(b) CSLDS. Comp = 100x. Rec. SNR = 19.80 dB

(c) MPEG-4. Comp = 334x. Rec. SNR = 19.93 dB

Figure 10. *Performance of MPEG-4 video compression and CS-LDS on the "candle" video (see Figure 7). Shown are* (a) *reconstruction SNR at various compression ratios, and ((*b), (c) *- top) a few reconstructed frames and ((*b), (c) *- bottom) error in reconstruction magnified 10×. It is worth nothing that the MPEG-4 algorithm has complete access to the ground truth video, while CS-LDS works purely with undersampled linear measurements of the video. Nonetheless, even at the same reconstruction SNR, the quality of MPEG-4 recovery is significantly better. This can be attributed to the nonlinear and adapted coding that seeks to mitigate errors that are perceptually dominant.*

millions of variables. The key is our two-step solution that isolates the space of unknowns into two manageable sets and solves for each separately.

Universality: An attractive property of random matrix-based CS measurement is the universality of the measurement process. Universality implies that the sensing process is independent of the subsequent reconstruction algorithm. This makes the sensing design "future-proof"; for such systems, if we devise a more sophisticated and powerful recovery algorithm in the future, then we do not need to redesign the camera or the sensing framework. The CS-LDS framework violates this property. The two-step measurement process of section 3, which is key to breaking the bilinearity introduced by the LDS prior, implies that the CS-LDS design is not universal. An intriguing direction for future research is the design of a universal CS-LDS measurement process.

Online tracking: We have made the assumption of a static observation matrix C. However, as the length of the video increases, the assumption of a static C is satisfied only by increasing the state space dimension. An alternate approach is to allow for a time-varying observation matrix $C(t)$ and track it from the compressive measurements. This would give us the benefit of a low state space dimension and, yet, be accurate when we sense for long durations.

Beyond LDS: Figure 10 captures the relative performance of MPEG-4 compression algorithm and CS-LDS on a video. MPEG-4 has access to the ground truth video and, as a consequence, it achieves significantly better compressions for the same performance in recovery (see Figure 10(a)). Further, it is worth noting that the nonlinear encoding in MPEG-4 produces errors that are *imperceptible* and hence, even at the same level of reconstruction error, produces videos that are of higher visual quality (see Figure 10(b,c)). This points at the inherent drawbacks of a linear encoder. While the CS-LDS framework makes a compelling case study of LDSs for video CS, its applicability to arbitrary videos is limited. In particular, it does not extend to simple nonstationary scenes such as people walking or panning cameras (see the result associated with Figure 8(h)). This motivates the search for models more general than LDS. In this regard, a promising line of future research is to leverage models from the video compression literature for CS recovery.

REFERENCES

[1] *Compressive Sensing of Dynamical Scenes*, CS-LDS project Web page, http://www.ece.rice.edu/∼as48/research/cslds.

[2] M. Ayazoglu, B. Li, C. Dicle, M. Sznaier, and O. I. Camps, *Dynamic subspace-based coordinated multicamera tracking*, in Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 2462–2469.

[3] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, *Model-based compressive sensing*, IEEE Trans. Inform. Theory, 56 (2010), pp. 1982–2001.

[4] R. G. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, *A simple proof of the restricted isometry property for random matrices*, Constr. Approx., 28 (2008), pp. 253–263.

[5] R. W. Brockett, *Finite Dimensional Linear Systems*, Wiley, New York, 1970.

[6] E. J. Candès and B. Recht, *Exact matrix completion via convex optimization*, Found. Comput. Math., 9 (2009), pp. 717–772.

[7] E. J. Candès, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inform. Theory, 52 (2006), pp. 489–509.

[8] E. J. Candès and T. Tao, *The power of convex relaxation: Near-optimal matrix completion*, IEEE Trans. Inform. Theory, 56 (2010), pp. 2053–2080.

[9] V. Cevher, A. C. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa, *Compressive sensing for background subtraction*, in Proceedings of the European Conference on Computer Vision, 2008, pp. 155–168.

[10] A. B. Chan and N. Vasconcelos, *Probabilistic kernels for the classification of auto-regressive visual processes*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 846–851.

[11] Y. Chikuse, *Statistics on Special Manifolds*, Springer-Verlag, Berlin, 2003.

[12] R. Coifman, F. Geshwind, and Y. Meyer, *Noiselets*, Appl. Comput. Harmon. Anal., 10 (2001), pp. 27–44.

[13] T. Ding, M. Sznaier, and O. I. Camps, *A rank minimization approach to video inpainting*, in Proceedings of the IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[14] D. L. Donoho, *Compressed sensing*, IEEE Trans. Inform. Theory, 52 (2006), pp. 1289–1306.

[15] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, *Dynamic textures*, Int. J. Comput. Vision, 51 (2003), pp. 91–109.

[16] 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., 25 (2008), pp. 83–91.

[17] M. F. Duarte, M. B. Wakin, D. Baron, S. Sarvotham, and R. G. Baraniuk, *Measurement bounds for sparse signal ensembles via graphical models*, IEEE Trans. Inform. Theory, 59 (2013), pp. 4280–4289.

[18] M. Fazel, *Matrix Rank Minimization with Applications*, Ph.D. thesis, Stanford University, Stanford, CA, 2002.

[19] M. Fazel, H. Hindi, and S. P. Boyd, *A rank minimization heuristic with application to minimum order system approximation*, in Proceedings of the IEEE American Control Conference, 2001, pp. 4734–4739.

[20] M. Fazel, H. Hindi, and S. P. Boyd, *Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices*, in Proceedings of the IEEE American Control Conference, 2003.

[21] M. Grant and S. Boyd, *CVX: MATLAB Software for Disciplined Convex Programming, Version* 1.21, 2011, available at http://cvxr.com/cvx.

[22] J. Haupt and R. Nowak, *Signal reconstruction from noisy random projections*, IEEE Trans. Inform. Theory, 52 (2006), pp. 4036–4048.

[23] 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 Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 287–294.

[24] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T. W. Lee, and T. J. Sejnowski, *Dictionary learning algorithms for sparse representation*, Neural Comput., 15 (2003), pp. 349–396.

[25] S. K. Nayar, V. Branzoi, and T. E. Boult, *Programmable imaging: Towards a flexible camera*, Int. J. Comput. Vision, 70 (2006), pp. 7–22.

[26] D. Needell and J. A. Tropp, *CoSaMP: Iterative signal recovery from incomplete and inaccurate samples*, Appl. Comput. Harmon. Anal., 26 (2009), pp. 301–321.

[27] J. Y. Park and M. B. Wakin, *A multiscale framework for compressive sensing of video*, in Proceedings of the Picture Coding Symposium, 2009, pp. 197–200.

[28] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, *Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition*, in Proceedings of the Asilomar Conference on Signals Systems and Computers, 1993.

[29] R. Péteri, S. Fazekas, and M. J. Huiskes, *DynTex: A comprehensive database of dynamic textures*, Pattern Recog. Lett., 31 (2010), pp. 1627–1632.

[30] B. Recht, M. Fazel, and P. A. Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM Rev., 52 (2010), pp. 471–501.

[31] D. Reddy, A. Veeraraghavan, and R. Chellappa, *P2C2: Programmable pixel compressive camera for high speed imaging*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 329–336.

[32] Y. Saad, *Krylov subspace methods for solving large unsymmetric linear systems*, Math. Comput., 37 (1981), pp. 105–126.

[33] P. Saisan, G. Doretto, Y. Wu, and S. Soatto, *Dynamic texture recognition*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2001, pp. II-58–II-63.

[34] A. C. Sankaranarayanan, P. Turaga, R. Baraniuk, and R. Chellappa, *Compressive acquisition of dynamic scenes*, in Proceedings of the European Conference on Computer Vision, 2010, pp. 129–142.

[35] S. Soatto, G. Doretto, and Y. N. Wu, *Dynamic textures*, in Proceedings of the IEEE International Conference on Computer Vision, 2001, pp. 439–446.

[36] M. Sznaier, *Compressive information extraction: A dynamical systems approach*, in Proceedings of the 16th IFAC Symposium on System Identification, 2012, pp. 1559–1568.

[37] P. Turaga, A. Veeraraghavan, and R. Chellappa, *Unsupervised view and rate invariant clustering of video sequences*, Comp. Vision and Image Understd., 113 (2009), pp. 353–371.

[38] E. van den Berg and M. P. Friedlander, *Probing the pareto frontier for basis pursuit solutions*, SIAM J. Sci. Comput., 31 (2008), pp. 890–912.

[39] P. Van Overschee and B. De Moor, *N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems*, Automatica, 30 (1994), pp. 75–93.

[40] N. Vaswani, *Kalman filtered compressed sensing*, in Proceedings of the IEEE Conference on Image Processing, 2008, pp. 893–896.

[41] N. Vaswani and W. Lu, *Modified-CS: Modifying compressive sensing for problems with partially known support*, in Proceedings of the International Symposium on Information Theory, 2009, pp. 488–492.

[42] A. Veeraraghavan, R. Chellappa, and A. K. Roy-Chowdhury, *The function space of an activity*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 959–968.

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

[44] A. Veeraraghavan, A. K. Roy-Chowdhury, and R. Chellappa, *Matching shape sequences in video with applications in human movement analysis*, IEEE Trans. Pattern Anal. Mach. Intell., 27 (2005), pp. 1896–1909.

[45] M. B. Wakin, J. N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, *Compressive imaging for video representation and coding*, in Proceedings of the Picture Coding Symposium, 2006.

[46] M. B. Wakin, B. M. Sanandaji, and T. L. Vincent, *On the observability of linear systems from random, compressive measurements*, in Proceedings of the IEEE Conference on Decision and Control, 2010, pp. 4447–4454.