CS-MUVI: Video Compressive Sensing for Spatial-Multiplexing Cameras

Aswin C. Sankaranarayanan, Christoph Studer, and Richard G. Baraniuk
Rice University

Abstract

Compressive sensing (CS)-based spatial-multiplexing cameras (SMCs) sample a scene through a series of coded projections using a spatial light modulator and a few optical sensor elements. SMC architectures are particularly useful when imaging at wavelengths for which full-frame sensors are too cumbersome or expensive. While existing recovery algorithms for SMCs perform well for static images, they typically fail for time-varying scenes (videos). In this paper, we propose a novel CS multi-scale video (CS-MUVI) sensing and recovery framework for SMCs. Our framework features a co-designed video CS sensing matrix and recovery algorithm that provide an efficiently computable low-resolution video preview. We estimate the scene's optical flow from the video preview and feed it into a convex-optimization algorithm to recover the high-resolution video. We demonstrate the performance and capabilities of the CS-MUVI framework for different scenes.

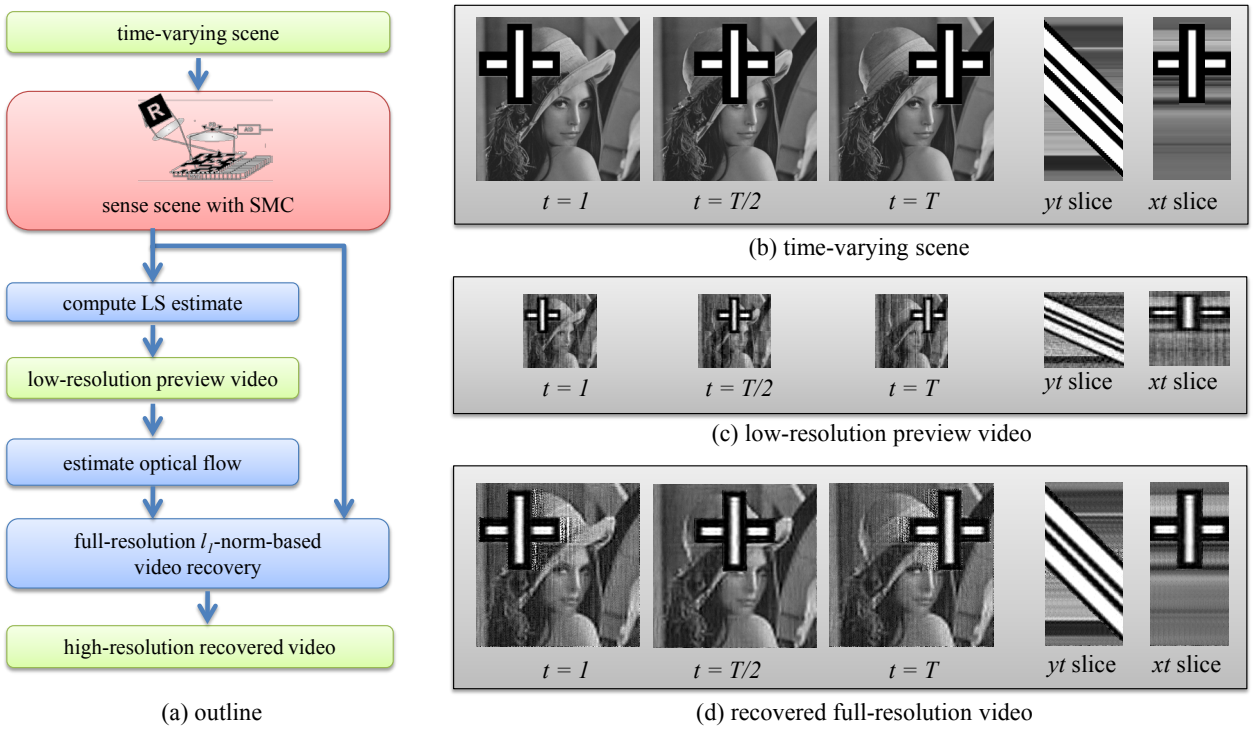
1. Introduction

Compressive sensing (CS) enables one to sample well-below the Nyquist rate, while still enabling the recovery of signals that admit a sparse representation in some basis [1, 2]. Since many natural and artificial signals exhibit sparsity, CS has the potential to reduce the sampling rates and costs of corresponding sampling devices in numerous applications.

Spatial-multiplexing cameras: The single-pixel camera (SPC) [3], the flexible voxels camera [4], and the P2C2 camera [5] are practical imaging architectures that rely on the theory of CS. In this paper, we focus on such spatial-multiplexing cameras (SMCs) that acquire random (or coded) projections of a (typically static) scene using a digital micro-mirror device (DMD) or liquid crystal on silicon (LCOS) in combination with a few optical sensing elements, such as photodetectors or bolometers. The use of a small number of optical sensors—in contrast to a full-frame sensor—turns out to be extremely useful when acquiring scenes at non-visible wavelengths. In particular, sensing beyond the visual spectrum often requires sensors built from exotic materials, which renders corresponding full-frame sensor devices cumbersome or too expensive.

Obviously, sampling with only a few sensors is, in general, not sufficient for acquiring complex scenes. Hence, SMCs acquire scenes by taking multiple consecutive measurements over time. For still images and for a single-pixel SMC architecture, this sensing strategy has been shown to deliver good results [3], but it fails for time-variant scenes (videos). The key challenge of video CS for SMCs is the fact that the scene to be captured is ephemeral, i.e., *each* compressive measurement senses a (slightly) *different* scene; the situation is further aggravated when we deal with SMCs having a small number of sensors (e.g., only one for the SPC). Virtually all proposed methods for CS-based video recovery (e.g., [6–10]) seem to overlook this important aspect. The approach described in [11] is a notable exception, but is designed specifically for time-varying *periodic* scenes; however, all other approaches that are suitable for more general scenes, e.g., [6–10], treat scenes as a sequence of *static* frames (i.e., videos) as opposed to a continuously changing scene. This disconnection between the real-world operation of SMCs and the assumptions commonly made for video CS motivates the continuing search for effective new algorithms.

The "chicken-and-egg" problem of video CS: Successful video CS recovery methods for camera architectures relying on temporal multiplexing (in contrast to spatial multiplexing as for SMCs) are generally inspired by video compression (i.e., exploit motion estimation) [5, 12, 13]. The use of such techniques for SMC architectures, however, results in a fundamental problem: On the one hand, obtaining motion estimates (e.g., optical flow) requires knowledge of the individual video frames. On the other hand, recovering the video frames in absence of motion estimates is difficult, especially when using low sampling rates and a small number of sensor elements. Attempts to address this "chicken-and-egg" problem either perform multi-scale sensing [6] or sense separate patches of the individual frames [10]. Both approaches ignore the time-varying nature of real-world scenes and rely on a piecewise static model.



(a) outline

(b) time-varying scene

(c) low-resolution preview video

(d) recovered full-resolution video

Figure 1. CS-MUVI example for a synthetic scene: (a) Flowchart of CS-MUVI, (b) ground-truth scene consisting of a background (Lena) and a single moving object (cross), (c) low-resolution preview video, and (d) high-resolution recovered video.

The CS-MUVI framework: In this paper, we propose a novel sensing and recovery method for videos acquired by SMC architectures such as the SPC [3]. We start (in Sec. 3) by studying the recovery performance of time-varying scenes and demonstrate that the performance degradation caused by violating the static-scene assumption is severe, even at moderate levels of motion. We then detail a novel CS strategy for SMC architectures that overcomes the static-scene assumption. Our approach, illustrated in Fig. 1, is a co-design of acquisition and recovery. We propose a novel class of CS that enables us to obtain a low-resolution "preview" of the scene with very low computational complexity. This preview video is then used to extract robust motion estimates (i.e., the optical flow) of the scene at full-resolution (see Sec. 4). We exploit these motion estimates to recover the full-resolution video by using off-the-shelf convex-optimization algorithms typically used for CS (detailed in Sec. 5). We demonstrate the performance and capabilities of our SMC video-recovery algorithm for a different scenes in Sec. 6 and discuss our findings in Sec. 7. Given the multiscale nature of our framework, we refer to it as CS multiscale video recovery (CS-MUVI).

2. Background

We next summarize the basics of compressive sensing and review existing CS-based camera architectures.

2.1. Compressive sensing

CS deals with the recovery of a signal vector $\mathbf{x} \in \mathbb{R}^N$ from $M < N$ non-adaptive linear measurements [1, 2]

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{z}, \tag{1}$$

where $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ is the sensing matrix and $\mathbf{z}$ represents measurement noise. Estimating the signal $\mathbf{x}$ from the compressive measurements $\mathbf{y}$ is ill-posed, in general, since the (noiseless) system of equations $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$ is underdetermined. Nevertheless, a fundamental result from CS theory states that the signal vector $\mathbf{x}$ can be recovered stably from

$$M \sim K \log(N/K) \tag{2}$$

measurements if: i) the signal $\mathbf{x}$ admits a K-sparse representation $\mathbf{s} = \mathbf{\Psi}^T\mathbf{x}$ in an orthonormal basis $\mathbf{\Psi}$, and ii) the matrix $\mathbf{\Phi\Psi}$ satisfies the restricted isometry property (RIP). For example, if the entries of the matrix $\mathbf{\Phi}$ are i.i.d. zero-mean (sub-)Gaussian distributed, then $\mathbf{\Phi\Psi}$ is known to satisfy the RIP with overwhelming probability. Furthermore, any K-sparse signal $\mathbf{x}$ satisfying (2) can be recovered stably from the noisy measurement $\mathbf{y}$ by solving a convex-optimization problem such as [1]

$$\text{(P1)} \quad \text{minimize } \|\mathbf{\Psi}^T\mathbf{x}\|_1 \quad \text{subject to } \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2 \leq \epsilon$$

where $(\cdot)^T$ denotes matrix transposition and ϵ controls the accuracy of the estimate.

2.2. Spatial-multiplexing camera architectures

Spatial-multiplexing cameras (SMCs) are practical imaging architectures that build on the ideas of CS. Such cameras employ a spatial light modulator, e.g., a digital micro-mirror device (DMD) or liquid crystal on silicon (LCOS), to optically calculate a series linear projections of a scene $\mathbf{x}$ by implementing the sensing process (1) using pseudo-random patterns that ultimately determine the sensing matrix $\mathbf{\Phi}$. A prominent example of an SMC architecture is the single-pixel camera (SPC) [3]; its main feature is the ability of acquiring images using only a *single* sensor element (i.e., a single pixel) and by taking significantly fewer measurements than the number of pixels of the scene to be recovered (cf. (2)). Since SMCs rely on only a few sensor elements, they can operate at wavelengths where corresponding full-frame sensors are too expensive. In the recovery stage, the image $\mathbf{x}$ is recovered from the compressive measurements collected in $\mathbf{y}$. In practice, recovery is performed either by using (P1) or a greedy algorithm.

2.3. Related work on video CS

Multi-scale video CS: One approach to video CS for SMC architectures relies on the observation that the perception of motion is heavily dependent on the spatial resolution of the video. Specifically, for a given scene, reducing its spatial resolution lowers the error caused by the static-scene assumption [14]. Simultaneously, decreasing the spatial resolution reduces the dimensionality of the individual video frames. Both observations build the foundation of the multi-scale recovery approach proposed in [6], where several compressive measurements are acquired at multiple scales for each video frame. 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). The key drawback of this approach is the fact that it relies on the assumption that each frame of the video remains static during the acquisition of the CS measurements at various scales. For scenes violating this assumption—as is the case in virtually all real-world situations—this approach results in a poor recovery quality.

Optical-flow-based video CS: Another recovery method was developed in [5] for the P2C2 camera, which differs considerably from SMC architectures. The P2C2 camera performs temporal multiplexing (instead of spatial multiplexing) with the aid of a full-frame sensor and a per-pixel shutter. The recovery of videos from the P2C2 camera is achieved by using the optical flow between pairs of consecutive frames of the scene. The implementation of the recovery procedure described in [5] is tightly coupled to the imaging architecture and inhibits its use for SMC architectures. Nevertheless, the use of optical-flow estimates for video CS recovery inspired the recovery stage of CS-MUVI as detailed in Sec. 5.

3. Spatio-temporal trade-off

We now study the recovery error that results from the static-scene assumption while sensing a time-varying scene (video) with an SMC. We also identify a fundamental trade-off underlying a multi-scale recovery procedure, which is used in Sec. 4 to identify novel sensing matrices that minimize the spatio-temporal recovery errors. Since the SPC is the most challenging SMC architecture (i.e., it only provides a single pixel sensor), we solely focus on the SPC in the following. Generalizing our results to other SMC architectures with more than one sensor is straightforward.

3.1. SMC acquisition model

The compressive measurements $y_t \in \mathbb{R}$ taken by a single-sensor SMC at the sample instants $t = 1, \dots, T$ can be written as $y_t = \langle \phi_t, \mathbf{x}_t \rangle + z_t$, where T is the total number of acquired samples, $\phi_t \in \mathbb{R}^{N\times 1}$ is the sensing vector, $z_t \in \mathbb{R}$ is the measurement noise, and $\mathbf{x}_t \in \mathbb{R}^{N\times 1}$ is the scene (or frame) at sample instant t; here, $\langle \cdot, \cdot \rangle$ denotes the inner product. In the remainder of the paper, we assume that the 2-dimensional scene consists of $n \times n$ spatial pixels, which, when vectorized, results in the vector $\mathbf{x}_t$ of dimension $N = n^2$. We also use the notation $\mathbf{y}_{1:W}$ to represent the vector consisting of a window of $W \leq T$ successive compressive measurements (samples), i.e,

$$\mathbf{y}_{1:W} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_W \end{bmatrix} = \begin{bmatrix} \langle \phi_1, \mathbf{x}_1 \rangle + z_1 \\ \langle \phi_2, \mathbf{x}_2 \rangle + z_2 \\ \vdots \\ \langle \phi_W, \mathbf{x}_W \rangle + z_W \end{bmatrix}. \quad (3)$$

3.2. Static-scene and down-sampling errors

Suppose that we rewrite our (time-varying) scene $\mathbf{x}_t$ for a window of W consecutive sample instants as follows:

$$\mathbf{x}_t = \mathbf{b} + \Delta\mathbf{x}_t, \quad t = 1, \dots, W.$$

Here, $\mathbf{b}$ is a static component (assumed to be invariant for W samples), and $\Delta\mathbf{x}_t = \mathbf{x}_t - \mathbf{b}$ is the error at sample instant t caused by assuming a static scene. By defining $e_t = \langle \phi_t, \Delta\mathbf{x}_t \rangle$, we can rewrite (3) as

$$\mathbf{y}_{1:W} = \mathbf{\Phi}\mathbf{b} + \mathbf{e}_{1:W} + \mathbf{z}_{1:W}, \quad (4)$$

where $\mathbf{\Phi} \in \mathbb{R}^{W\times N}$ is a sensing matrix whose tth row corresponds to the transposed vector ϕ_t.

We now consider the error caused by spatial downsampling of the static component $\mathbf{b}$ in (4). To this end, let $\mathbf{b}_L \in \mathbb{R}^{N_L}$ be the down-sampled static component, and as-

sume $N_L = n_L \times n_L$ with $N_L < N$. By defining a linear up-sampling and down-sampling operator as $\mathbf{U} \in \mathbb{R}^{N \times N_L}$ and $\mathbf{D} \in \mathbb{R}^{N_L \times N}$, respectively, we can rewrite (4) as

$$\begin{aligned}\mathbf{y}_{1:W} &= \mathbf{\Phi}(\mathbf{U}\mathbf{b}_L + \mathbf{b} - \mathbf{U}\mathbf{b}_L) + \mathbf{e}_{1:W} + \mathbf{z}_{1:W} \\ &= \mathbf{\Phi}\mathbf{U}\mathbf{b}_L + \mathbf{\Phi}(\mathbf{b} - \mathbf{U}\mathbf{b}_L) + \mathbf{e}_{1:W} + \mathbf{z}_{1:W} \\ &= \mathbf{\Phi}\mathbf{U}\mathbf{b}_L + \mathbf{\Phi}(\mathbf{I} - \mathbf{U}\mathbf{D})\mathbf{b} + \mathbf{e}_{1:W} + \mathbf{z}_{1:W} \qquad (5)\end{aligned}$$

since $\mathbf{b}_L = \mathbf{D}\mathbf{b}$. Inspection of (5) reveals three sources of error in the CS measurements of the low-resolution static scene $\mathbf{\Phi}\mathbf{U}\mathbf{b}_L$: i) The *spatial-approximation error* $\mathbf{\Phi}(\mathbf{I} - \mathbf{U}\mathbf{D})\mathbf{b}$ caused by down-sampling, ii) the *temporal-approximation error* $\mathbf{e}_{1:W}$ caused by assuming the scene remains static for W samples, and iii) the *measurement error* $\mathbf{z}_{1:W}$.

3.3. Estimating a low-resolution image

In order to analyze the trade-off that arises from the static-scene assumption and the down-sampling procedure, consider the scenario where the effective matrix $\mathbf{\Phi}\mathbf{U}$ is of dimension $W \times N_L$ with $W \geq N_L$; that is, we aggregate at least as many compressive samples as the down-sampled spatial resolution. If $\mathbf{\Phi}\mathbf{U}$ has full (column) rank, then we can obtain a least-squares (LS) estimate $\widehat{\mathbf{b}}_L$ of the low-resolution static scene $\mathbf{b}_L$ from (5) as

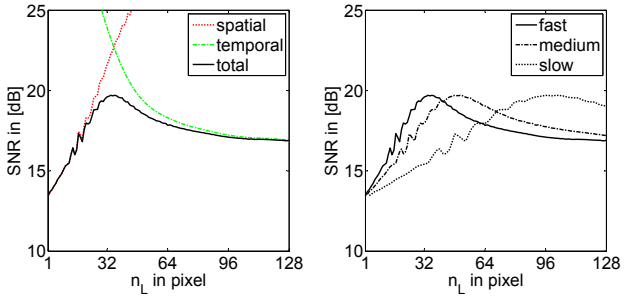
$$\begin{aligned}\widehat{\mathbf{b}}_L &= (\mathbf{\Phi}\mathbf{U})^{\dagger}\mathbf{y}_{1:W} \\ &= \mathbf{b}_L + (\mathbf{\Phi}\mathbf{U})^{\dagger}\big(\mathbf{\Phi}(\mathbf{I} - \mathbf{U}\mathbf{D})\mathbf{b} + \mathbf{e}_{1:W} + \mathbf{z}_{1:W}\big) \qquad (6)\end{aligned}$$

where $(\cdot)^{\dagger}$ denotes the (pseudo) inverse. From (6) we can observe the following facts: i) The window length W controls a trade-off between the spatial-approximation error $\mathbf{\Phi}(\mathbf{I} - \mathbf{U}\mathbf{D})\mathbf{b}$ and the error $\mathbf{e}_{1:W}$ induced by assuming a static scene $\mathbf{b}$, and ii) the least squares (LS) estimator matrix $(\mathbf{\Phi}\mathbf{U})^{\dagger}$ (potentially) amplifies all three error sources.

3.4. Characterizing the trade-off

As developed in Sec. 3.3, the spatial-approximation error and the temporal-approximation error are both a function of the window length W. We now show that carefully selecting W minimizes the combined spatial and temporal error in the low-resolution estimate $\widehat{\mathbf{b}}_L$. Inspection of (6) shows that for $W = 1$, the temporal-approximation error is zero, since the static component $\mathbf{b}$ is able to perfectly represent the scene at each sample instant t. As W increases, the temporal-approximation error increases for time-varying scenes; simultaneously, increasing W reduces the error caused by down-sampling $\mathbf{\Phi}(\mathbf{I} - \mathbf{U}\mathbf{D})\mathbf{b}$ (see Fig. 2(a)). For $W \geq N$ there is no spatial approximation error (if $\mathbf{\Phi}\mathbf{U}$ is invertible). Note that characterizing both errors analytically is difficult, in general, as they depend on the scene under consideration.

Figure 2 illustrates the trade-off controlled by W and



(a) separate error sources (b) impact of temporal changes

Figure 2. Trade-off between spatial and temporal approximation errors (measured in terms of the CS recovery SNR) for the scene in Fig. 1. (a) The SNRs caused by spatial and temporal approximation errors for different window lengths W. (b) The dependence of the total approximation error on the speed of the cross.

the individual spatial and temporal approximation errors, characterized in terms of the recovery signal-to-noise-ratio (SNR), for the synthetic scene shown in Fig. 1. The figure highlights the important fact that there is an optimal window length W for which the total recovery SNR is maximized. In particular, we see from Fig. 2(b) that the optimum window length increases (i.e., towards higher spatial resolution) when the scene changes slowly; in contrary, when the scene changes rapidly, the window length (and consequently, the spatial resolution) should be low. Since $N_L \leq W$, the optimal window length W dictates the resolution for which accurate low-resolution motion estimates can be obtained.

4. Design of sensing matrix

In order to bootstrap CS-MUVI, a low-resolution estimate of the scene is required. We next now that carefully designing the CS sensing matrix $\mathbf{\Phi}$ enables us to compute high-quality low-resolution scene estimates at low complexity, which improves the performance of video recovery.

4.1. Dual-scale sensing matrices

The choice of the sensing matrix $\mathbf{\Phi}$ and the upsampling operator $\mathbf{U}$ are critical to arrive at a high-quality estimate of the low-resolution image $\mathbf{b}_L$. Indeed, if the compound matrix $\mathbf{\Phi}\mathbf{U}$ is ill-conditioned, then application of $(\mathbf{\Phi}\mathbf{U})^{\dagger}$ amplifies all three sources of errors in (6), resulting in a poor estimate. For a large class of conventional CS matrices $\mathbf{\Phi}$, such as i.i.d. (sub-)Gaussian matrices, as well as sub-sampled Fourier or Hadamard matrices, right multiplying them with an upsampling operator $\mathbf{U}$ typically results in an ill-conditioned matrix. Hence, using well-established CS matrices for obtaining a low-resolution preview turns out to be a poor choice. Figures 3(a) and 3(b) show recovery results for naïve recovery using (P1) and LS, respectively, using a subsampled noiselet CS matrix. One can immediately

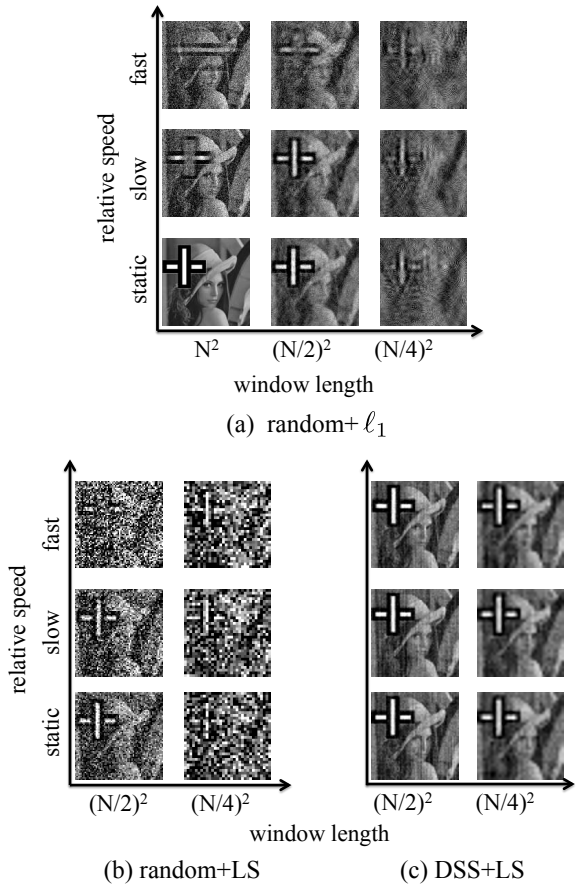


Figure 3. Comparison between (a) ℓ_1-norm recovery, (b) LS recovery using a subsampled noiselet matrix, and (c) LS recovery using a dual-scale sensing (DSS) matrix of the scene in Fig. 1 for various relative speeds (of the cross) and window lengths W.

see that both recovery methods fail spectacularly for large values of W or for a small amount of motion.

In order to achieve good CS recovery performance *and* have minimum noise enhancement when computing low-resolution estimates $\widehat{\mathbf{b}}_L$ according to (6), we propose a novel class of sensing matrices, referred to as *dual-scale sensing* (DSS) matrices. In particular, we wish to use matrices that i) satisfy the RIP and ii) remain well-conditioned when right-multiplied by a given up-sampling operator $\mathbf{U}$. The second condition requires mutual orthogonality among the columns of $\mathbf{\Phi U}$ to minimize the noise enhancement in (6). Random matrices are known to satisfy the RIP with overwhelming probability.However, they typically fail to meet the second constraint, because they have decaying singular values. The power of DSS matrices is demonstrated in Fig. 3(c), even for small window lengths W or fast motion.

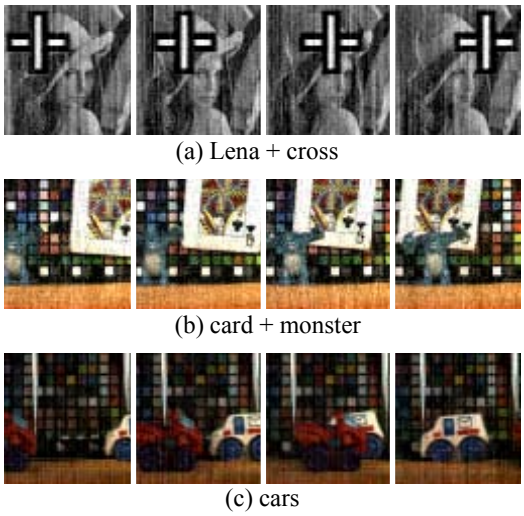


Figure 4. Preview frames for three different scenes. All previews consist of 64×64 pixels. Preview frames are obtained simply using an inverse Hadamard transform, which opens up a variety of new (real-time) applications for video CS.

4.2. Preview mode

If we additionally impose the constraint that a downsampled DSS matrix $\mathbf{\Phi U}$ has a fast inverse transform, then it will significantly speed up the recovery of the low-resolution scene. Such a "fast" DSS matrix has the key capability of generating a high-quality *preview* of the scene (see Fig. 4) with very low computational complexity; this is beneficial for video CS as it allows us to easily extract an estimate of the scene motion. The motion estimate can then be used to recover the video at its full resolution (see Section Sec. 5). In addition to this, the use of fast DSS matrices can be beneficial in various other ways, including (but not limited to):

Real-time preview: Conventional SMC architectures do not enable the observation of the scene until CS recovery is performed. Due to the high computational complexity of most existing CS recovery algorithms, there is typically a large latency between the acquisition of a scene and its observation. Fast DSS matrices offer an *instantaneous* visualization of the scene, i.e., they can provide us with a real-time digital viewfinder. This capability substantially simplifies the setup of an SMC in practice.

Adaptive sensing: The immediate knowledge of the scene—even at a low resolution—can potentially be used to design adaptive sensing strategies. For example, one may seek to extract the changes that occur in a scene from one frame to the next or track moving objects, while avoiding the latency caused by ℓ_1-norm recovery algorithms.

4.3. Sensing matrix design

There are many ways to construct fast DSS matrices. In this section, we detail one design that is particularly suited for SMC architectures.

In SMC architectures, we are constrained in the choice of the sensing matrix $\mathbf{\Phi}$. Practically, the DMD limits us to matrices having entries of constant modulus (e.g., ± 1). Since we are interested in a fast DSS matrix, we propose the matrix $\mathbf{\Phi}$ to satisfy $\mathbf{H} = \mathbf{\Phi U}$, where $\mathbf{H}$ is a $W \times W$ Hadamard matrix[1] and $\mathbf{U}$ is a predefined up-sampling operator. For SMC architectures, Hadamard matrices have the following advantages: i) They have orthogonal columns, ii) they exhibit optimal SNR properties over matrices restricted to $\{-1, +1\}$ entries [15, 16], and iii) applying the (inverse) Hadamard transform requires very low computational complexity (i.e., as a fast Fourier transform).

We now show the construction of a suitable fast DSS matrix $\mathbf{\Phi}$ (see Fig. 5(a)). A simple way is to start with a $W \times W$ Hadamard matrix $\mathbf{H}$ and to write the CS matrix as

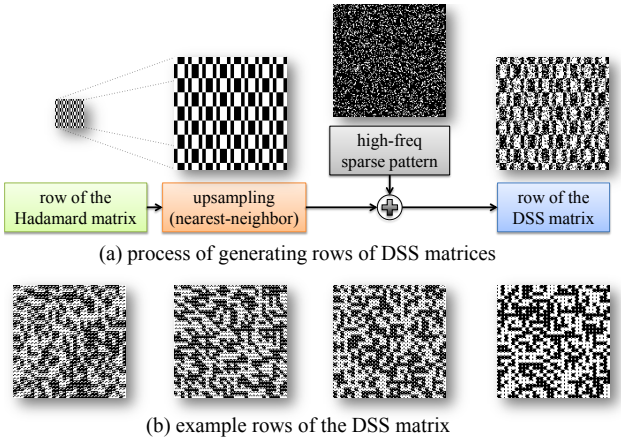
$$\mathbf{\Phi} = \mathbf{HD} + \mathbf{F}, \tag{7}$$

where $\mathbf{D}$ is a down-sampling matrix satisfying $\mathbf{DU} = \mathbf{I}$, and $\mathbf{F} \in \mathbb{R}^{W \times N}$ is an auxiliary matrix that obeys the following constraints: i) The entries of $\mathbf{\Phi}$ are ± 1,[2] ii) the matrix $\mathbf{\Phi}$ has good CS recovery properties (e.g., satisfies the RIP), and iii) $\mathbf{F}$ should be chosen such that $\mathbf{FU} = \mathbf{0}$. Note that an easy way to ensure that $\mathbf{\Phi}$ be ± 1 is to interpret $\mathbf{F}$ as sign flips of the Hadamard matrix $\mathbf{H}$. Note that one could chose $\mathbf{F}$ to be an all-zeros matrix; this choice, however, results in a sensing matrix $\mathbf{\Phi}$ having poor CS recovery properties. In particular, such a matrix would inhibit the recovery of high spatial frequencies. Choosing random entries in $\mathbf{F}$ such that $\mathbf{FU} = \mathbf{0}$ (i.e., by using random patterns of high spatial frequency) provides excellent performance.

To arrive at an efficient implementation of CS-MUVI, we additionally want to avoid the storage of an entire $W \times N$ matrix. To this end, we generate each row $\mathbf{f}_i \in \mathbb{R}^N$ of $\mathbf{F}$ as follows: Associate each row vector $\mathbf{f}_i$ to an $n \times n$ image of the scene, partition the scene into blocks of size $(n/n_L) \times (n/n_L)$, and associate an $(n/n_L)^2$-dimensional vector $\hat{\mathbf{f}}_i$ with each block. We can now use the same vector $\hat{\mathbf{f}}_i$ for each block and choose $\hat{\mathbf{f}}_i$ such that the full matrix satisfies $\mathbf{FU} = \mathbf{0}$. We also permute the columns of the Hadamard matrix $\mathbf{H}$ to achieve better incoherence with the sparsifying bases used in Sec. 5 (see Fig. 5(b) for the details).

[1] In the ensuing discussion, we assume that W is chosen such that a Hadamard matrix of size $W \times W$ exists.

[2] Practical implementation of ± 1 matrices is done by an appropriate shift and scaling to convert each element to $\{0, 1\}$.



(a) process of generating rows of DSS matrices

(b) example rows of the DSS matrix

Figure 5. Generating DSS patterns. (a) Outline of the process in (7). (b) In practice, we permute the low-resolution Hadamard for better incoherence with the sparsifying wavelet basis. Fast generation of the DSS matrix requires us to impose additional structure on the high-frequency patterns. In particular, each sub-block of the high-frequency pattern is forced to be the same, which enables fast computation via convolutions.

5. Optical-flow-based video recovery

We next detail the second part of CS-MUVI. Fig. 6 illustrates the algorithm used to recover the full-resolution video frames (see also the flowchart in Fig. 1).

5.1. Optical-flow estimation

Thanks to the preview mode, we can estimate the optical flow between any two (low-resolution) frames $\widehat{\mathbf{b}}_L^i$ and $\widehat{\mathbf{b}}_L^j$. For CS-MUVI, we compute optical-flow estimates at full spatial resolution between pairs of upsampled preview frames; this approach turns out to result in more accurate optical-flow estimates compared to an approach that first estimates the optical flow at low resolution followed by upsampling of the optical flow. Hence, we start by upsampling the preview frames according to $\widehat{\mathbf{b}}^i = \mathbf{U}\widehat{\mathbf{b}}_L^i$, and then extract the optical flow at full resolution. The optical flow at full resolution can be written as

$$\widehat{\mathbf{b}}^i(x, y) = \widehat{\mathbf{b}}^j(x + u_{x,y}, y + v_{x,y}),$$

where $\widehat{\mathbf{b}}^i(x, y)$ denotes the pixel (x, y) in the $n \times n$ plane of $\widehat{\mathbf{b}}^i$, and $u_{x,y}$ and $v_{x,y}$ correspond to the translation of the pixel (x, y) between frame i and j (see [17, 18]).

In practice, the estimated optical flow may contain subpixel translations, i.e., $u_{x,y}$ and $v_{x,y}$ are not necessarily integers. In this case, we approximate $\widehat{\mathbf{b}}^j(x + u_{x,y}, y + v_{x,y})$

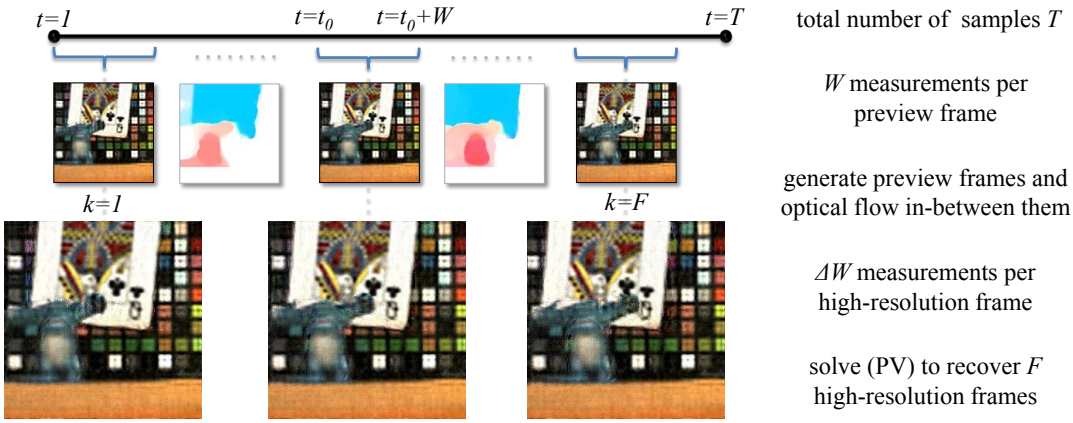


Figure 6. Outline of CS-MUVI recovery (see also Fig. 1(a)): Given a total number of T measurements, we group them into windows of size ΔW resulting in a total of $F = T/\Delta W$ frames. For each frame, we first compute a preview (see Sec. 4.2) using a window of $W \geq \Delta W$ neighboring measurements. Then, we compute the optical flow between successive preview frames (the optical flow is color-coded as in [17]). Finally, we use the preview frames together with the optical-flow estimates in (PV) to obtain F high-resolution video frames.

as a linear combination of its four closest neighboring pixels

$$\widehat{\mathbf{b}}^j(x+u_{x,y}, y+v_{x,y}) \approx \sum_{k,\ell\in\{0,1\}} w_{k,\ell}\widehat{\mathbf{b}}^j(\lfloor x+u_{x,y}\rfloor + k, \lfloor y+v_{x,y}\rfloor + \ell),$$

where $\lfloor\cdot\rfloor$ denotes rounding towards $-\infty$ and the weights $w_{k,\ell}$ are chosen according to the location within the four neighboring pixels.[3] In order to obtain robustness against occlusions, we enforce consistency between the forward and backward optical flows; specifically, we discard optical flow constraints at pixels where the sum of the forward and backward flow causes a displacement greater than one pixel.

5.2. Choosing the recovery frame rate

Before we detail the individual steps of the CS-MUVI video-recovery procedure, it is important to specify the rate of the frames to be recovered. When sensing scenes with SMC architectures, there is no obvious notion of frame rate. Our sole criterion is that we want each "frame" to contain only a small amount of motion. In other words, we wish to find the largest window size $\Delta W \leq W$ such that there is virtually no motion at full resolution ($n \times n$). In practice, an estimate of ΔW can be obtained by analyzing the preview frames. Hence, given a total number of T compressive measurements, we ultimately recover $F = T/\Delta W$ full-resolution frames (see Fig. 6). Note that a smaller value of ΔW would decrease the amount of motion associated with each recovered frame; this would, however, increase the computational complexity (and memory requirements) substantially as the number of full-resolution frames to be recovered increases.

[3] More sophisticated interpolation schemes could be used but result in higher computational complexity.

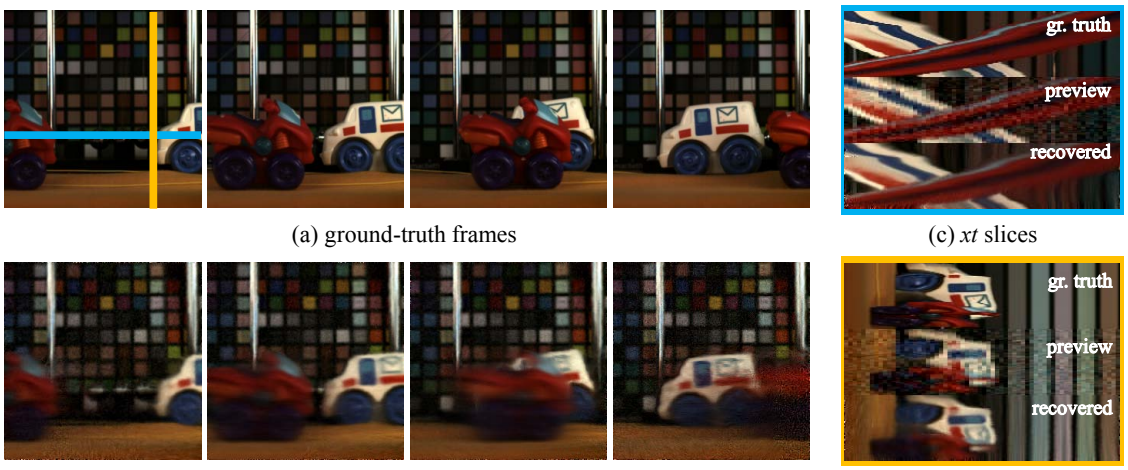
5.3. Recovery of full-resolution frames

We are now ready to detail the final steps of CS-MUVI. Assume that ΔW is chosen such that there is little to no motion associated with each preview frame. Next, associate a preview frame with a high-resolution frame $\hat{\mathbf{x}}_k$, $k \in \{1, \ldots, T\}$ by grouping $W = N_L$ compressive measurements in the immediate vicinity of the frame (since $\Delta W \leq W$). Then, compute the optical-flow between successive (up-scaled) preview frames.

We can now recover the individual high-resolution video frames as follows. Each frame $\hat{\mathbf{x}}_t$ is assumed to have a sparse representation in a 2-dimensional orthogonal wavelet basis $\mathbf{\Psi}$; hence, our objective is to minimize the overall ℓ_1-norm $\sum_{k=1}^{F} \|\mathbf{\Psi}^T\hat{\mathbf{x}}_k\|_1$. We furthermore consider the following two constraints: i) Consistency with the acquired CS measurements, i.e, $\mathbf{y}_t = \langle \phi_t, \hat{\mathbf{x}}_{I(t)} \rangle$, where $I(t)$ maps the sample index t to the associated frame index k, and ii) estimated optical-flow constraints between consecutive frames. Together, we arrive at the following convex optimization problem:

$$\text{(PV)} \quad \begin{cases} \text{min.} & \sum_{k=1}^{F} \|\mathbf{\Psi}^T\hat{\mathbf{x}}_k\|_1 \\ \text{s. t.} & \|\langle \phi_t, \hat{\mathbf{x}}_{I(t)} \rangle - y_t\|_2 \leq \epsilon_1, \forall t \\ & \|\hat{\mathbf{x}}_i(x,y) - \hat{\mathbf{x}}_j(x+u_x, y+v_y)\|_2 \leq \epsilon_2, \forall i,j, \end{cases}$$

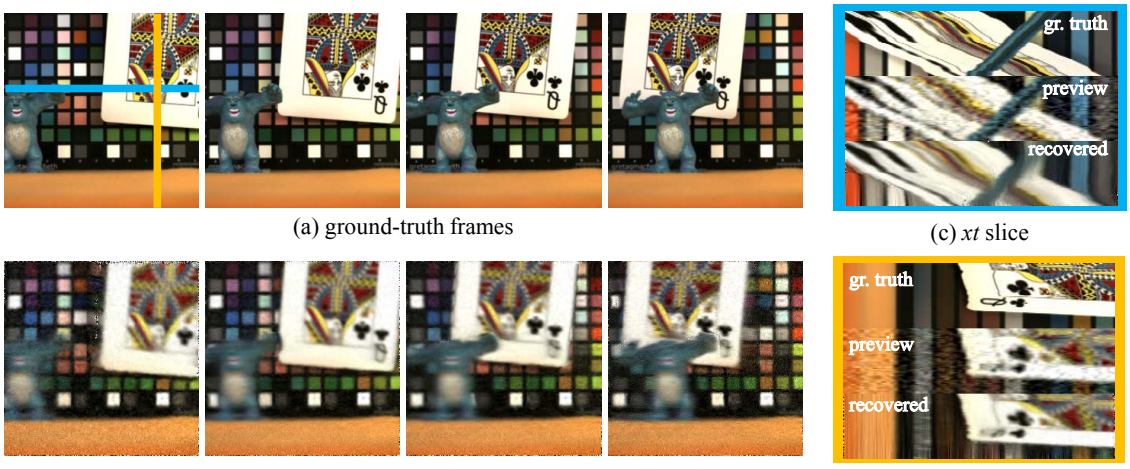
which can be solved using off-the-shelf algorithms tuned to solve ℓ_1-recovery problems [19]. The parameters $\epsilon_1 \geq 0$ and $\epsilon_2 \geq 0$ can be used to "tweak" the recovery performance.



(a) ground-truth frames (c) *xt* slices

(b) frames from the recovered video (d) *yt* slice

Figure 7. CS-MUVI recovery results of a video obtained from a high-speed camera. Shown are frames of (a) the ground truth and (b) the recovered video (PSNR = 25.0 dB). The xt and yt slices shown in (c) and (d) correspond to the color-coded lines of the first frame in (a). Preview frames for this video are shown in Fig. 4. (The xt and yt slices are rotated clockwise by 90 degrees.)



(a) ground-truth frames (c) *xt* slice

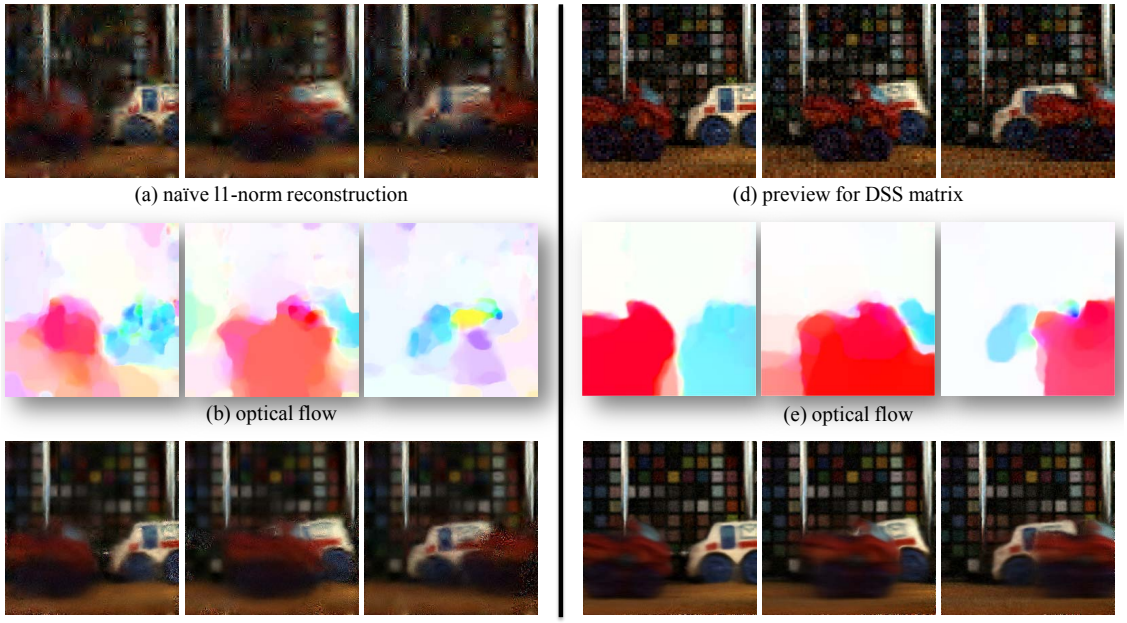
(b) frames from the recovered video (d) *yt* slice

Figure 8. CS-MUVI recovery results of a video obtained from a high-speed camera. Shown are frames of (a) the ground truth and (b) the recovered video (PSNR = 20.4 dB). The xt and yt slices shown in (c) and (d) correspond to the color-coded lines of the first frame in (a). Preview frames for this video are shown in Fig. 4. (The xt and yt slices are rotated clockwise by 90 degrees.)

6. Experiments

We validate the performance and capabilities of the CS-MUVI framework for several scenes. All simulation results were generated from video sequences having a spatial resolution of $n \times n = 256 \times 256$ pixels. The preview videos have a spatial resolution of 64×64 pixels with (i.e., $W = 4096$). We assume an SPC architecture as described in [3]. Noise was added to the compressive measurements using an i.i.d. Gaussian noise model such that the resulting SNR was 60 dB. Optical-flow estimates were extracted using [17] and (PV) is solved using SPGL1 [19]. The computation time of CS-MUVI is dominated by solving (PV), which requires 2–3 hours using an off-the-shelf quad-core CPU. The low-resolution preview is, of course, extremely fast.

Synthetic scene with sub-pixel motion: In Fig. 1 we simulate a fast-moving object that traverses the entire field-of-view of the camera within the considered number of samples T. The goal of this synthetic experiment is to emulate a scene that changes for every compressive measurement. To this end, we simulated sub-pixel movement of the foreground object, i.e., there is a small movement of the cross for *every* compressive measurement. We acquired a total of $T = 256^2$ compressive measurements and gener-



(a) naïve l1-norm reconstruction (d) preview for DSS matrix

(b) optical flow (e) optical flow

(c) P2C2 recovered video (f) CS-MUVI recovered video

Figure 9. Comparison of the algorithm used for the P2C2 camera applied to SMC architectures with CS-MUVI recovery. Shown are frames of (a) naïve ℓ_1-norm reconstruction, (b) the resulting optical-flow estimates, and (c) the P2C2 recovered video. The frames in (d) correspond to preview frames when using DSS matrices, (e) are the optical-flow estimates, and (f) is the scene recovered by CS-MUVI.

ated $F = 31$ preview frames ($\Delta W = 2048$) from which we estimated the respective optical flows. Figure 1 shows both the efficacy of the proposed DSS measurement matrix for providing robust LS estimates for the preview video (see also Fig. 4), as well as the quality of the recovered scene.

Video sequences from a high-speed camera: The results shown in Figs. 7 and 8 correspond to scenes acquired by a high-speed (HS) video camera operating at 250 frames per second. Both videos show complex (and fast) movement of large objects as well as severe occlusions. For both sequences, we emulate an SPC operating at 8192 compressive measurements per second. For each video, we used 2048 frames of the HS camera to obtain a total of $T = 32 \times 2048$ compressive measurements. The final recovered video sequences consist of $F = 61$ frames ($\Delta W = 1024$). Both recovered videos demonstrate the effectiveness of CS-MUVI.

Comparison with the P2C2 algorithm: Figure 9 compares CS-MUVI to the recovery algorithm for the P2C2 camera [5]. Note that the P2C2 camera algorithm was developed for temporal multiplexing cameras and *not* for SMC architectures. Nevertheless, we observe from Figs. 9 (a) and (d) that naïve ℓ_1-norm recovery delivers significantly worse initial estimates than the preview mode of CS-MUVI. The advantage of CS-MUVI for SMC architectures is also visible in the corresponding optical-flow estimates (see Figs. 9 (b) and (e). The P2C2 recovery algorithm has substantial artifacts, whereas CS-MUVI recovery is visually pleasing.

7. Discussion

Summary: In this paper, we have proposed CS-MUVI, a novel compressive-sensing (CS)-based multi-scale video recovery framework for scenes acquired by spatial-multiplexing cameras (SMCs). Our main contribution is the design of a novel class of sensing matrices and an optical-flow based video reconstruction algorithm. In particular, we have proposed dual-scale sensing (DSS) matrices that i) exhibit no noise enhancement when performing least-squares estimation at low spatial resolution and ii) preserve information about high spatial frequencies. We have developed a DSS matrix having a fast transform, which enables us to compute instantaneous *preview* images of the scene at low cost. The preview computation supports a large number of novel applications for SMC-based devices, such as providing a digital viewfinder, enabling human-camera interaction, or triggering adaptive sensing strategies. Finally, CS-MUVI is the first video CS algorithm for the SPC that works well for scenes with fast and complex motion.

Reconstruction artifacts: There are some artifacts visible in Figs. 1(d), 7, and 8. The major portion stems from inaccurate optical-flow estimates—a result of residual noise in the preview images. It is worth noting, however, that we are using an off-the-shelf optical-flow estimation algorithm;

such an approach ignores the continuity of motion across *multiple* frames. We envision significant performance improvements if we use multi-frame optical-flow estimation. A smaller portion of the recovery artifacts is caused by using dense measurement matrices, which spread local errors (such as those from the inaccurate optical-flow estimates) across the entire image. This problem is inherent to imaging with SMCs that involve a high degree of spatial multiplexing; imaging architectures that perform only local spatial multiplexing (such as the P2C2 camera) do not suffer from this problem.

Compression: The videos in Figs. 7 and 8 have $256 \times 256 \times 61$ pixels and were obtained from 256^2 compressive measurements; hence, a naïve estimate would suggest a compression of $61\times$. However, the blur in the recovered videos suggest that the finest spatial frequencies are not present. A formal study of the true compression ratio would require the use of resolution charts and a characterization of the finest spatial and temporal frequencies resolved; this is an important direction for future work.

Limitations: Since CS-MUVI relies on optical-flow estimates obtained from low-resolution images, it can fail to recover small objects with rapid motion. More specifically, moving objects that are of sub-pixel size in the preview mode are lost. Figure 7 shows an example of this limitation: The cars are moved using fine strings, which are visible in Fig. 7(a) but not in Fig. 7(b). Increasing the spatial resolution of the preview images eliminates this problem at the cost of more motion blur. To avoid these limitations altogether, one must increase the sampling rate of the SMC.

Future work: A drawback of our approach is the need to specify the resolution at which preview frames are recovered; this requires prior knowledge of object speed. An important direction for future work is to relax this requirement via the construction of multi-scale sensing matrices that go beyond the DSS matrices proposed here. In addition, reducing the complexity of solving (PV) is of paramount importance for practical implementations of CS-MUVI.

Acknowledgments

Thanks to K. Kelly, L. Xu, and A. Veeraraghavan for inspiring discussions. The work of C. Studer was supported by the Swiss National Science Foundation (SNSF) under Grant PA00P2-134155. The work of A. C. Sankaranarayanan and R. G. Baraniuk was supported by the Grants NSF CCF-1117939, CCF-0431150, CCF-0728867, CCF-0926127; DARPA N66001-11-1-4090, N66001-11-C-4092; ONR N00014-08-1-1112, N00014-10-1-0989; AFOSR FA9550-09-1-0432; ARO MURIs W911NF-07-1-0185 and W911NF-09-1-0383.

References

[1] 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, pp. 489–509, Feb. 2006.

[2] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289–1306, Apr. 2006.

[3] 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, pp. 83–91, Mar. 2008.

[4] M. Gupta, A. Agrawal, A. Veeraraghavan, and S. Narasimhan, "Flexible voxels for motion-aware videography," in *Euro. Conf. Comp. Vision*, (Crete, Greece), Sep. 2010.

[5] D. Reddy, A. Veeraraghavan, and R. Chellappa, "P2C2: Programmable pixel compressive camera for high speed imaging," in *IEEE Conf. Comp. Vision and Pattern Recog*, (Colorado Springs, CO, USA), June 2011.

[6] J. Y. Park and M. B. Wakin, "A multiscale framework for compressive sensing of video," in *Pict. Coding Symp.*, (Chicago, IL, USA), May 2009.

[7] A. C. Sankaranarayanan, P. Turaga, R. Baraniuk, and R. Chellappa, "Compressive acquisition of dynamic scenes," in *Euro. Conf. Comp. Vision*, (Crete, Greece), Sep. 2010.

[8] N. Vaswani, "Kalman filtered compressed sensing," in *IEEE Conf. Image Process.*, (San Diego, CA, USA), Oct. 2008.

[9] 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 *Pict. Coding Symp.*, (Beijing, China), Apr. 2006.

[10] S. Mun and J. E. Fowler, "Residual reconstruction for block-based compressed sensing of video," in *Data Comp. Conf.*, (Snowbird, UT, USA), Apr. 2011.

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

[12] 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 *IEEE Intl. Conf. Comp. Vision*, (Barcelona, Spain), Nov. 2011.

[13] D. Mahajan, F. C. Huang, W. Matusik, R. Ramamoorthi, and P. Belhumeur, "Moving gradients: A path-based method for plausible image interpolation," *ACM Trans. Graph.*, vol. 28, pp. 1–42, Aug. 2009.

[14] M. B. Wakin. Personal communication, 2010.

[15] M. Harwit and N. Sloane, *Hadamard transform optics*. New York: Academic Press, 1979.

[16] Y. Y. Schechner, S. K. Nayar, and P. N. Belhumeur, "Multiplexing for optimal lighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 1339–1354, Aug. 2007.

[17] C. Liu, *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, Mass. Inst. Tech., 2009.

[18] B. Horn and B. Schunck, "Determining optical flow," *Artif. Intel.*, vol. 17, pp. 185–203, Apr. 1981.

[19] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM J. Scientific Comp.*, vol. 31, pp. 890–912, Nov. 2008.