REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From-to)*
14-12-2010	Final Technical Report	15-SEP-09 to 14 SEP-10

4. TITLE AND SUBTITLE
Compressive Video Acquisition, Fusion and Processing

5a. CONTRACT NUMBER

5b. GRANT NUMBER
N00014-09-1-1162

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Baraniuk, Richard G.; Chellappa, Rama; Wakin, Michael

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Rice University
6100 Main St. MS 16
Houston TX 77005

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Tristan Nguyen
Office of Naval Research, Code 311
875 N Randolph St.
Arlington, VA 22203-1995

10. SPONSOR/MONITOR'S ACRONYM(S)
ONR

11. SPONSORING/MONITORING AGENCY REPORT NUMBER

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Modern developments in sensor technology, signal processing, and wireless communications have enabled the conception and deployment of large-scale networked sensing systems spanning numerous collection platforms and varied modalities. These systems have the potential to make intelligent decisions by integrating information from massive amounts of sensor data. Before such benefits can be achieved, significant advances must be made in methods for communicating, fusing, and processing this ever-growing volume of diverse data.

In this one-year research project, we aimed to expose the fundamental issues and pave the way for further careful study of compressive approaches to video acquisition, fusion, and processing. In doing so, we developed a theoretical definition of video temporal bandwidth and applied the theory to compressive sampling and reconstruction. We created a new framework for compressive video sensing based on linear dynamical systems, lowering the compressive measurement rate. Finally, we applied our own joint manifold model to a variety of relevant image processing problems, demonstrating the model's effectiveness and ability to overcome noise and occlusion obstacles. We also showed how joint manifold models can discover an object's trajectory, an important step towards video fusion.

15. SUBJECT TERMS
compressive sensing, compressive sampling, compressive video, manifolds, linear dynamical systems, data fusion, classification

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	SAR	60	Richard G. Baraniuk
U	U	U			19b. TELEPONE NUMBER (*Include area code*) 713-348-5132

Compressive Video Acquisition, Fusion, and Processing
Final Report

Richard G. Baraniuk, Rice University
Rama Chellappa, University of Maryland
with *Michael B. Wakin*, Colorado School of Mines

This is the final report for ONR Grant N00014-09-1-1162 *Compressive Video Acquisition, Fusion, and Processing*. We begin by restating the motivation for our work, reviewing the project objectives, and summarizing our accomplishments. We present detailed results and deliverables, and conclude with a list of publications supported by the grant, followed by a list of project personnel.

1 Review of motivation

Recent developments in sensor technology, signal processing, and wireless communications have enabled the conception and deployment of large-scale networked sensing systems comprising coordinated stationary and mobile platforms carrying sensors of diverse modalities. The promise of such systems is that they might intelligently make decisions by integrating information from massive amounts of sensor data. However, meeting this promise will require overcoming several significant obstacles including:

- *Growing volumes of sensor data*: Multiple mobile and stationary sensors operating continuously over increasingly large and complicated environments produce prodigious volumes of data that must be communicated, fused, organized, and processed in (near) real-time without collapsing the communications fabric.
- *Increasingly diverse data*: The apparent lack of correlation among images and other data taken from different viewpoints and with different sensor modalities and resolutions thwarts naive approaches to data fusion and processing.
- *Diverse and changing operating conditions*: The targets of interest may be cluttered, camouflaged, occluded, and sensed under different illuminations with miscalibrated or noisy sensors. Novel targets can appear, greatly complicating online operation.
- Increasing mobility: Multiple mobile sensing platforms must be remotely or autonomously maneuvered and coordinated to optimize decision making performance.

A promising concept for surmounting these obstacles has emerged recently under the moniker of compressive sensing (CS).

1.1 Compressive sensing in theory

CS combines sampling and compression into a single nonadaptive linear measurement process [1–3]. Rather than measuring pixel samples of the scene under view, we measure inner products between the scene and a set of test functions. Interestingly, random test functions play a key role, making each measurement a random sum of pixel values taken across the entire image. When the scene under view is compressible in some basis expression (e.g., wavelets), the CS theory enables us to stably reconstruct an image of the scene from fewer measurements than the number of reconstructed pixels. In this manner we achieve sub-Nyquist image acquisition.

1.2 Compressive sensing in practice

In preliminary work, we have constructed a "single-pixel" CS camera architecture that comprises an optical computer (comprising a digital micromirror device, two lenses, a single photon detector, and an analog-to-digital (A/D) converter) that computes random linear measurements of the scene under view [4]. The image is then recovered or

processed from the measurements by a digital computer. The camera design reduces the required size, complexity, and cost of the photon detector array down to a single unit, which enables small and light cameras for UAV applications and enables the use of exotic detectors that would be impossible in a conventional digital camera. The random CS measurements also enable a tradeoff between space and time during image acquisition. Finally, since the camera compresses as it images, it has the capability to efficiently and scalably handle high-dimensional data sets from applications like video and hyperspectral imaging.

2 Project objectives and accomplishments

In this one-year basic research project, we aimed to expose the fundamental issues and pave the way for further careful study of video **acquisition**, **processing**, and **fusion** using dimensionality reduction and CS techniques.

2.1 Reconstruction from compressive video measurements

Background
We can acquire video sequences using the single-pixel camera. Recall that a traditional video camera opens a shutter periodically to capture a sequence of images (called video frames) that are then compressed by an algorithm like MPEG that jointly exploits their spatio-temporal redundancy. In contrast, the single-pixel video camera needs no shutter; we merely continuously sequence through randomized test functions ϕ_m and then reconstruct a video sequence using an optimization that exploits the videos spatio-temporal redundancy [5].

If we view a video sequence as a 3D space/time cube, then the test functions ϕ_m lie concentrated along a periodic sequence of 2D image slices through the cube. A naive way to reconstruct the video sequence would group the corresponding measurements $y[m]$ into groups where the video is quasi-stationary and then perform a 2D frame-by-frame reconstruction on each group. This exploits the compressibility of the 3D video cube in the space but not time direction.

Objective
A more powerful alternative that we can explore in detail exploits the fact that even though each ϕ_m is testing a different 2D image slice, the image slices are often related temporally through smooth object motions in the video. Exploiting this 3D compressibility in both the space and time directions and inspired by modern 3D video coding techniques [6], we set out to reconstruct the video space/time cube from compressive measurements.

Accomplishments
Since the CS camera compressively measures 2D time-constant slices of the 3D space-time video cube, it is of foundational importance to understand the temporal bandwidth of the video. A significant thrust of our research work was studying the temporal bandwidth of video in relation to compressive measurements of the space-time cube. We related temporal bandwidth to the spacial resolution of the camera and the speed of objects in the scene. We applied our findings to develop sampling and interpolation theory, and then to video reconstruction from compressive measurements. For more details on these accomplishments, see Section 3.

2.2 Data processing on compressive video measurements

Objective
Before exploitation by automated means or a human observer, raw video data is usually processed to bring out its salient information. Routine processing tasks include video stabilization, background subtraction, detection and tracking of moving objects, and verification and recognition of moving objects. Our goal was to determine which of these operations can be performed directly on the compressive measurements without requiring a potentially expensive video reconstruction.

Accomplishments
In our work exploring data processing from compressive measurements, we discovered that a new framework for compressive video could lead to advances in such processing. We developed 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, from which the image frames are then reconstructed. By exploiting the low-dimensional dynamic parameters (the

state sequence) and high-dimensional static parameters (the observation matrix), we lowered the compressive measurement rate considerably. Our original objective in mind, we applied our approach to classification experiments to demonstrate the effectiveness of our new framework. For more details on these accomplishments, see Section 4.

2.3 Data fusion using manifold methods

Background

Many image and video processing algorithms can be seen in a new light as processing on a low-dimensional manifold. For example, sensor localization/calibration, 3-D target pose and location estimation, image registration, and super-resolution all involve estimating elements of given a set of points on the manifold [7]. Estimating the target pose given noisy image data involves projecting the noisy image point onto the manifold. Sensor fusion of an object imaged with multiple modalities involves docking a family of related manifolds. Finding similar images to a target involves searching for a nearest neighbor point under the manifold-specific geodesic distance. Classifying objects involves comparing the distance between a test image and a number of candidate model manifolds (one for each object). In fact, the distance between image manifolds is a fundamental quantity that characterizes the discriminability of the corresponding objects and hence ultimate ATR performance [8].

A key enabler of our approach is the very recent result developed by our team that the same kind of information preservation that enables CS for sparse data also applies to smooth manifolds [12]. That is, an $M \times N$ random projection matrix stably embeds an L-dimensional smooth manifold from $\mathbb{R}^N$ into $\mathbb{R}^M$ if $M = O(L \log N)$. Like the fundamental bound in CS for sparse data, our requisite M is linear in the information level (now the dimension of the manifold L, not the sparsity level K) and only logarithmic in the ambient dimension N; additionally we have identified a logarithmic dependence on the volume and curvature of the manifold [12]. Intriguingly, the signals or images on the manifold need not be sparse in any basis!

This result guarantees that many of the key image/video manifold properties are preserved under a random projection to lower-dimensional space, including its dimension, topology, geodesic distances, and curvature. This opens up a wide array of sensing sub-tasks for random projection methods, including manifold based processing and navigation, and Steifel manifold-based classification. Moreover, our result indicates that manifold diffusion algorithms for point clouds drawn from image manifolds will also work in lower dimensions Our multiscale geometric data representations provide new means for fusing signal and image ensembles generated from different target views, illumination conditions, and sensor modalities into robust statistical models for subsequent learning and classification. For instance, in many operational scenarios, we will acquire data on a target of interest from J different modalities and platforms (assume each sensor is of resolution N). The total amount of raw data acquired is thus $O(JN)$ total bits; as J and N grow, it quickly becomes infeasible to communicate or process this amount of information. Even using random encoding (CS) at each sensor, we still must sense and communicate $O(JK \log(N/K))$ bits, which grows rapidly with the number of sensors J. This is a potential deal-breaker for the entire concept of networked video fusion and decision making.

Objective

Previously, we had developed a method for drastically reducing the dimensionality of multi-view, multi-modal data such as video data from a number of maneuvering UAVs via a joint manifold model (JMM) [9]. The key realization is that when J sensors are observing the same target, a single L-dimensional articulation parameterization spans the J articulation manifolds. Thus, if at each time instant data is available we simply stack it (no matter what modality) into a large JN-dimensional vector, then the resulting data still lies on an L- dimensional manifold, but now in $\mathbb{R}^{JN}$. Since this manifold lies in an expanded space, it has improved curvature and volume properties, which translate directly into improved performance for manifold-based algorithms such as smoothing, fusion, and learning [9]. We sought to explore in detail the JMM approach to video fusion using both simulated and real compressive video data.

Accomplishments

We applied our JMM theory to a classification problem with three cameras. We found that joint manifold fusion proves to be more effective than high level fusion algorithms like majority voting. We then applied our work in joint manifold learning in experiments with real and simulated data. We examined cases of noise in realistic images and of images with occlusions. Finally, as first step to video fusion, we used compressive measurements from four cameras to discover an object's trajectory. For more details on these accomplishments, see Section 5.

3 Bandlimited video for compressive sensing

The task of recovering a video from compressive measurements can be very challenging due to the sheer dimensionality of the data to be recovered. In streaming measurement systems such as the Rice "single-pixel camera," this challenge can be particularly daunting, because we may record as little as one measurement per time instant. Presented with this information, it would be possible to follow standard Compressive Sensing (CS) formulations and set up a system of linear equations that relate the measurements to the underlying high-rate video voxels. However, since the video can change from one instant to the next, the number of unknown voxels in such a formulation would unfortunately scale with the number of measurements collected. Moreover, if the inter-measurement time interval were to go to zero, the size of the reconstruction problem would continue to grow.

There are reasons to believe that we should be able control this explosion of dimensionality. Intuitively, for example, we know that as the inter-measurement time interval goes to zero, the video should change very little between the times when adjacent measurements are collected. In this document, we argue that underlying high-rate video voxels—despite growing in number—actually have a bounded complexity, and that one way of characterizing their relatively few degrees of freedom is to study the temporal bandlimitedness of the video. We argue analytically that, at the imaging sensor, many videos should have limited temporal bandwidth due to the spatial lowpass filtering that is inherent in typical imaging systems. Under modest assumptions about the motion of objects in the scene, this spatial filtering prevents the temporal complexity of the video from being arbitrarily high. Consequently, we conclude that under various interpolation kernels (sinc, linear, etc.), the unknown high-rate voxels can be represented as a linear combination of a low-rate (e.g., Nyquist-rate) "anchor" set of sampled video frames.

Our bandwidth analysis uses fairly standard arguments from imaging processing and communications theory but goes deeper than the "constant velocity" model commonly assumed for object motion. Typical analysis under the constant velocity reflects that the video's temporal bandwidth will be proportional to the object's (constant) speed. For more representative motion models, we show that that proportionality still basically holds true, but with a $2\times$ higher constant of proportionality, and with an additional additive factor due to the bandwidth of the signal tracing the object's path. Although there is room for debate in the assumptions we make, our analysis reveals an interesting tradeoff between the spatial resolution of the camera, the speed of any moving objects, and the temporal bandwidth of the video. We use a study of synthetic and real videos to quantify how well this bandlimited/interpolation model fits.

After arguing that many videos have limited temporal bandwidth, we revisit the CS reconstruction problem. We explain how the problem can be reformulated by setting up a system of linear equations that relate the number of measurements to the underlying degrees of freedom of the video (specifically, the anchor frames). Notably, the number of anchor frames does not depend on the temporal measurement rate, and so the reconstruction problem can be much more tractable than in the original formulation described above. We demonstrate experimentally that the CS reconstruction performance depends on the type of factors that impact the video's temporal bandwidth (e.g., spatial resolution of the camera, the speed of any moving objects, etc.), and we explore possible tradeoffs in the design of a reconstruction algorithm. Using even a linear interpolation kernel, we also show that the reconstruction performance is is significantly better than what can be achieved by raw "aggregation" of measurements, which corresponds to using a rectangular, nearest neighbor interpolation kernel.

3.1 Videos with one spatial dimension

3.1.1 Problem setup

Signal model

We start our analysis by considering "videos" that have just one spatial dimension. We will use the variable $t \in \mathbb{R}$ to index time (which we measure in seconds), and we will use the variable $x \in \mathbb{R}$ to index spatial position (which for convenience we measure in an arbitrary unit we call "pix"). Though we will begin by considering continuous-space, continuous-time videos, the "pix" unit is intended to symbolize what might be the typical pixel size in a subsequent discretization of this video. One could easily replace pix with meters or any other arbitrary unit of distance.

Now, let $g(x)$ denote a 1D function of space (think of this as a continuous-space "still image"), and consider a continuous-space, continuous-time video $f(x,t)$ in which each "frame" of the video merely consists of a shifted version of this prototype frame. More formally, suppose that

$$f(x,t) = g(x - h(t))$$

where $h(t)$ is some function that controls how much (in pix) the prototype frame is shifted at each time step.

Bandwidth considerations
Because we have an interest in video imaging systems with high temporal sampling rates, our purpose in this document is to describe some conditions under which the model video $f(x,t)$ will have finite temporal bandwidth. This limited bandwidth, in turn, suggests that such videos can be characterized in terms of a temporal Nyquist-rate sampling of frames, that one may characterize the performance of various temporal interpolation procedures, etc.

We suggest that the limited temporal bandwidth of $f(x,t)$ could arise in plausible scenarios under which the prototype frame $g(x)$ and translation signal $h(t)$ have limited complexity. For example, in a physical imaging system, we may envision $f(x,t)$ as the video that exists at the imaging sensor prior to sampling. It may be reasonable to expect that, due to optical blurring and due to the implicit filtering that occurs from the spatial extent of each light integrator, the prototype frame $g(x)$ may have limited spatial bandwidth. Similarly, if the camera motion is constrained or due to the physics governing the movement of objects in the scene, one might expect that the translation signal $h(t)$ may have limited slope and/or limited temporal bandwidth. In the sections that follow, we will explain how such scenarios can allow us to bound the approximate temporal bandwidth of $f(x,t)$.

Fourier setup
Let $F(\omega_x, \omega_t)$ denote the 2D Fourier transform of $f(x,t)$, and let $G(\omega_x)$ denote the 1D Fourier transform of $g(x)$. Keeping track of units, ω_x is measured in terms of rad/pix, and ω_t is measured in terms of rad/s. The following relationships will be of use to us.

First, let $\mathcal{F}_x\{\cdot\}$ and $\mathcal{F}_t\{\cdot\}$ denote operators on $L_2(\mathbb{R}^2)$ that perform 1D Fourier transforms in the spatial and temporal directions, respectively. Due to the separability of the 2D Fourier transform, we know that

$$F(\omega_x, \omega_t) = \mathcal{F}_x\{\mathcal{F}_t\{f\}\}(\omega_x, \omega_t) = \mathcal{F}_t\{\mathcal{F}_x\{f\}\}(\omega_x, \omega_t). \tag{1}$$

Now, due to the shift property of the 1D Fourier transform, we have

$$\mathcal{F}_x\{f\}(\omega_x, t) = \mathcal{F}_x\{g(x - h(t))\}(\omega_x, t) = G(\omega_x)e^{-j\omega_x h(t)}.$$

We note that this function is complex-valued and has constant magnitude in the temporal direction; therefore it has only phase changes in the temporal direction. Following (1), F will be the result of taking the 1D Fourier transform of this function $\mathcal{F}_x\{f\}$ in the temporal direction. We have

$$F(\omega_x, \omega_t) = \mathcal{F}_t\{G(\omega_x)e^{-j\omega_x h(t)}\}(\omega_x, \omega_t) = G(\omega_x) \cdot L(\omega_x, \omega_t),$$

where

$$L(\omega_x, \omega_t) := \mathcal{F}_t\{e^{-j\omega_x h(t)}\}(\omega_x, \omega_t). \tag{2}$$

For fixed ω_x, $L(\omega_x, \omega_t)$ equals the 1D Fourier transform of $e^{-j\omega_x h(t)}$ with respect to time, evaluated at the frequency ω_t.

3.1.2 Temporal bandwidth analysis

The appearance of the $h(t)$ term within an exponent in (2) can complicate the task of characterizing the bandwidth of $f(x,t)$ in terms of properties of $h(t)$. However, by imposing certain assumptions on $h(t)$, this analysis can become tractable. In Section 3.1.2 below, we briefly discuss a "constant velocity" model for $h(t)$ that is commonly seen in textbook discussions of video bandwidth (see, e.g., [10]). In Section 3.1.2, we then go more deeply into the analysis of a "bounded velocity" model; while we are not aware of any such analysis in the existing image and video processing literature, the concepts do borrow heavily from standard results in communications and modulation theory (see, e.g., [11]). Except where noted, all subsequent discussion in this document will build on this bounded velocity analysis, rather than the constant velocity analysis.

Constant velocity model for $h(t)$
Our analysis simplifies dramatically if we assume that $h(t) = \Gamma t$ for some constant Γ (having units of pix/s). In this case we have $L(\omega_x, \omega_t) = \delta(\omega_t + \omega_x\Gamma)$, and so

$$F(\omega_x, \omega_t) = G(\omega_x) \cdot \delta(\omega_t + \omega_x\Gamma),$$

which corresponds to a diagonal line in the 2D Fourier plane with slope (say, $\Delta\omega_t$ over $\Delta\omega_x$) that depends linearly on Γ.

To see the implications of this in terms of bandwidth, let us suppose that $G(\omega_x)$ is bandlimited (or essentially bandlimited) to the range of frequencies $\omega_x \in [-\Omega_x, \Omega_x]$ rad/pix. (This may occur if the profile $g(x)$ has been blurred in the x direction, for example.) In this case, it follows that $F(\omega_x, \omega_t)$ must be bandlimited (or essentially bandlimited) to the range of frequencies $(\omega_x, \omega_t) \in [-\Omega_x, \Omega_x] \times [-\Gamma\Omega_x, \Gamma\Omega_x]$. In other words, the temporal bandwidth of the video is no greater than $\Gamma\Omega_x$ rad/s.

Bounded velocity model for $h(t)$

In slightly more generality, we could assume that the position function $h(t)$ has bounded slope, i.e., that for some $\Gamma > 0$,

$$\left|\frac{dh(t)}{dt}\right| \leq \Gamma \text{ pix/s}$$

for all t. This corresponds to a bound on the speed at which the object can move in the video, without requiring that this speed be constant. We also assume that the translation signal $h(t)$ is bandlimited, with bandwidth given by Ω_h rad/s.

For any fixed ω_x, we can recognize $e^{-j\omega_x h(t)}$ as a phase-modulated (PM) sinusoid having carrier frequency 0 and phase deviation (in radians)

$$\phi(t) = -\omega_x h(t),$$

or equivalently, as a frequency-modulated (FM) sinusoid having carrier frequency 0 and instantaneous frequency (in rad/s)

$$\omega_i(t) = \frac{d\phi(t)}{dt} = -\omega_x \frac{dh(t)}{dt}.$$

For am FM signal, we have

$$\frac{d\phi(t)}{dt} = \omega_d m(t),$$

where $m(t)$ is known as the modulating signal and ω_d is the frequency deviation constant. Thus, we have

$$m(t) = \frac{1}{\omega_d}\frac{d\phi(t)}{dt} = \frac{-\omega_x}{\omega_d}\frac{dh(t)}{dt}.$$

Let us also define the deviation term

$$D := \frac{\omega_d}{\Omega_h} \max|m(t)| = \frac{|\omega_x|}{\Omega_h} \max\left|\frac{dh(t)}{dt}\right| \leq \frac{|\omega_x|\Gamma}{\Omega_h}.$$

From Carson's bandwidth rule for frequency modulation, we have that for fixed ω_x, at least 98% of the total power of $e^{-j\omega_x h(t)}$ must be concentrated in the frequency range $\omega_t \in [-2(D+1)\Omega_h, 2(D+1)\Omega_h]$ rad/s. Since $D \leq \frac{|\omega_x|\Gamma}{\Omega_h}$, we conclude that at least 98% of the total power of $e^{-j\omega_x h(t)}$ must be concentrated in the frequency range $\omega_t \in [-(2|\omega_x|\Gamma + 2\Omega_h), 2|\omega_x|\Gamma + 2\Omega_h)]$ rad/s. We note that the dependence of this bandwidth on ω_x is essentially linear.

We conclude that $L(\omega_x, \omega_t)$ will have a characteristic "butterfly shape" with most of its total power concentrated between two diagonal lines that intercept the ω_t-axis at $\pm 2\Omega_h$ and have slope approximately $\pm 2\Gamma$. This shape is illustrated in Figure 1(a). Though not shown, the corresponding figure for the constant velocity model discussed in Section 3.1.2 would involve a single diagonal line intersecting the origin and having slope Γ (which is half as large as the slope that appears in our more general bounded velocity analysis).

To see the implications of this in terms of bandwidth, let us again suppose that $G(\omega_x)$ is bandlimited (or essentially bandlimited) to the range of frequencies $\omega_x \in [-\Omega_x, \Omega_x]$. We must then have that $F(\omega_x, \omega_t) = G(\omega_x) \cdot L(\omega_x, \omega_t)$ is also essentially bandlimited in the spatial direction to the range of frequencies $\omega_x \in [-\Omega_x, \Omega_x]$. Because of the butterfly structure in $L(\omega_x, \omega_t)$, however, this will also cause $F(\omega_x, \omega_t)$ to be essentially bandlimited in the temporal direction to the range of frequencies

$$\omega_t \in [-(2\Omega_x\Gamma + 2\Omega_h), 2\Omega_x\Gamma + 2\Omega_h] \text{ rad/s}. \tag{3}$$

This fact, which is illustrated in Figure 1(b), exemplifies a central theme of our work: *filtering a video in the spatial direction can cause it to be essentially bandlimited both in space and in time.*

3.1.3 Sampling implications

Based on the temporal bandwidth predicted in (3), where we assume that $G(\omega_x)$ is bandlimited (or essentially bandlimited) to the range of frequencies $\omega_x \in [-\Omega_x, \Omega_x]$, the Nyquist theorem suggests that in order to avoid aliasing, the video should be sampled a minimum rate of

$$\frac{2\Omega_x\Gamma + 2\Omega_h}{\pi} \text{ samples/s.}$$

As a point of reference, during a sampling interval of $\left(\frac{2\Omega_x\Gamma+2\Omega_h}{\pi}\right)^{-1}$ seconds, an object moving with maximum speed Γ pix/s can traverse a maximum of

$$\frac{\pi\Gamma}{2\Omega_x\Gamma + 2\Omega_h} \text{ pix.} \tag{4}$$

Let us plug in some plausible numbers to illustrate the implications of these bounds. First, consider the spatial bandwidth Ω_x of the prototype frame. In a reasonable imaging system, we might expect the pixel size to be balanced with the spatial bandwidth of the frame so that spatial aliasing is avoided. (This should occur naturally if we assume each pixel integrates spatially over a window of size approximately 1 pix.) Thus, one might anticipate that Ω_x will be on the order of π rad/pix. (This corresponds to a spatial bandwidth of $\frac{\pi}{2\pi} = \frac{1}{2}$ cycles/pix, which suggests a spatial Nyquist sample rate of one sample per pix.)

Under the assumption that $\Omega_x = \pi$, (3) suggests the video will have temporal bandwidth limited to approximately $2\pi\Gamma + 2\Omega_h$. We note that Ω_h, the temporal bandwidth of $h(t)$, does not depend on the amplitude or slope of $h(t)$, but only on its shape and smoothness. The term Γ, in contrast, increases with the amplitude or slope of $h(t)$, which in turn could increase for objects closer to the camera. We conjecture that in practice, the $2\pi\Gamma$ term will typically dominate the $2\Omega_h$ term.[1] If this is indeed the case, (4) suggests that, in typical scenarios, to avoid temporal aliasing we should not allow a moving object to traverse more than $\approx \frac{1}{2}$ pix between adjacent sampling times. While this of course makes strong intuitive sense, we have arrived at this conclusion through a principled analysis.

Let us make this even more concrete by further specifying some example parameter values. For an imaging system with, say, 1000 pixels per row, a distant object might move a maximum of $\Gamma \approx 10$ pix/s, while a close object could move at up to $\Gamma \approx 1000$ pix/s. These suggest minimum temporal bandwidths of 20π rad/s and 2000π rad/s, respectively, which correspond to minimum temporal Nyquist sampling rates of 20 samples/s and 2000 samples/s, respectively. As predicted above, these sampling rates would allow the object to move no more than about $\frac{1}{2}$ pix per temporal sample.

Aside from exceptionally non-smooth motions $h(t)$, we do strongly suspect that the influence of the temporal bandwidth Ω_h will be minor in comparison to the influence of the $2\pi\Gamma$ term, and so *in general a temporal Nyquist sampling rate of* 2Γ *samples/s will likely serve as a reasonable rule of thumb.* Certainly, this rule of thumb illustrates the direct relationship between the speed of object motion in the video and the video's overall temporal bandwidth.

3.1.4 Experiments within our model assumptions

In this section, we analytically define continuous-space, continuous-time videos that allow us to test these behaviors and observe the tradeoffs between object speed, spatial filtering, and temporal bandwidth. In all experiments in this section, we let the prototype function

$$g(x) = \text{sinc}\left(\frac{\Omega_x x}{\pi}\right) \tag{5}$$

for some value of Ω_x that we set as desired. This definition ensures that $g(x)$ is bandlimited and that its bandwidth equals precisely Ω_x rad/pix.

We oversample the videos compared to the predicted spatial and temporal bandwidths and show the approximate spectrum using the FFT. (All plots in fact show the magnitude of the FFT on a $\log_{10}$ scale.) In some cases we apply a smooth Blackman-Harris window to the samples before computing the FFT; this helps remove artifacts from the borders of the sampling region.

Constant velocity model for $h(t)$

For our first experiment, we let the translation signal

$$h(t) = \Gamma t$$

[1] A refined "Carson-type" analysis would permit the consideration of functions $h(t)$ that are not strictly bandlimited.

Table 1: *Video parameters used for experiments in several figures.*

Test Video	Spatial BW Ω_x	Motion BW Ω_h	Max. Speed Γ	Max. Predicted Temporal BW $2\Omega_x\Gamma + 2\Omega_h$
Row 1	π rad/pix	15 rad/s	25 pix/s	187 rad/s
Row 2	3π rad/pix	15 rad/s	25 pix/s	501 rad/s
Row 3	π rad/pix	45 rad/s	25 pix/s	247 rad/s
Row 4	π rad/pix	15 rad/s	200 pix/s	1287 rad/s

for some value of Γ that we set as desired. As in (5), we let $g(x) = \text{sinc}\left(\frac{\Omega_x x}{\pi}\right)$ for some value of Ω_x that we set as desired.

Based on our discussion in Section 3.1.2, we anticipate that for this video, $F(\omega_x, \omega_t)$ will be supported along a diagonal line in the 2D Fourier plane with slope (say, $\Delta\omega_t$ over $\Delta\omega_x$) equal to Γ. Figure 2 illustrates our estimated FFT spectrum for one experiment with $\Omega_x = \pi$ rad/pix and $\Gamma = 25$ pix/s; the solid blue box outlines the region $(\omega_x, \omega_t) \in [-\Omega_x, \Omega_x] \times [-\Gamma\Omega_x, \Gamma\Omega_x]$, and the diagonal, dashed blue line indicates the predicted support of the spectrum. We do indeed see that the estimated spectrum follows this diagonal line, and we see that this line ends at the spatial bandwidth of approximately Ω_x rad/s, giving rise to a temporal bandwidth of $\Gamma\Omega_x = 25\pi$ rad/s. Tests with other values of Ω_x and Γ also behave as predicted.

Bounded velocity model for $h(t)$

In order to test our predictions in the case of bounded velocity, we set

$$h(t) = \sum_{i=1}^{5} a_i \text{sinc}\left(\frac{\Omega_h(t - d_i)}{\pi}\right), \tag{6}$$

where Ω_h controls the total bandwidth (and can be set as we desire), the delays d_i are chosen randomly, and the amplitudes a_i are chosen somewhat arbitrarily but ensure that the maximum value attained by $|h(t)|$ equals some parameter Γ, which we can set as we desire. Note that we can thus independently articulate the bandwidth and the maximum slope of this signal. This "sum of sinc functions" model for $h(t)$ was chosen to give this function an interesting shape while still allowing us to ensure that it is perfectly bandlimited. As in (5), we let $g(x) = \text{sinc}\left(\frac{\Omega_x x}{\pi}\right)$ for some value of Ω_x that we set as desired.

For several choices of our parameters $(\Omega_x, \Omega_h, \Gamma)$, Figure 3 shows the video $f(x, t)$ along with its estimated spectrum. The blue lines in each case indicate the predicted "butterfly shape" which should bound the nonzero support of $F(\omega_x, \omega_t)$. Table 1 lists the values of our parameters $(\Omega_x, \Omega_h, \Gamma)$ used in each row of the figure. In these experiments, we see that, as we vary the bandwidth and velocity parameters, the approximate support of the estimated spectrum stays within the "butterfly shape" predicted by our theory in Section 3.1.2.

As another way to support this theory, we have computed for each video the empirical temporal bandwidth based on our estimated spectrum. To do this, we determine the value of Ω_t for which 99.99% of the energy in the FFT (or windowed FFT) falls within the range $|\omega_t| \leq \Omega_t$. For each of the four videos shown in Figure 3, we see that this empirical bandwidth Ω_t equals roughly 48 to 52% of the bandwidth $2\Omega_x\Gamma + 2\Omega_h$ predicted by our theory. (There are occasional exceptions where the unwindowed FFT gives a higher estimate, but this is likely due to sampling artifacts.) We note that this behavior is *consistent* with our theory, since our bandwidth prediction is merely an upper bound. Indeed, it is encouraging that such a high proportion of the empirical energy falls within our predicted region, since Carson's bandwidth rule considers only up to about 98% of the signal power.

Sinusoidal model for $h(t)$

As one last interesting case that obeys all of our model assumptions, we let

$$h(t) = a\cos(\Omega_h t),$$

where Ω_h controls the bandwidth (and can be set as we desire), and the amplitude a is chosen to ensure that the maximum value attained by $|h(t)|$ equals some parameter Γ, which we can set as we desire. As in (5), we let $g(x) = \text{sinc}\left(\frac{\Omega_x x}{\pi}\right)$ for some value of Ω_x that we set as desired.

Figure 4 shows the corresponding videos and estimated spectra, where the four rows correspond to the same four parameter sets for $(\Omega_x, \Omega_h, \Gamma)$ specified in Table 1. We see a distinct "striped" pattern in the estimated spectra;

this is to be expected given the temporal periodicity of the video. (Sinusoidal functions $h(t)$ are often chosen as a standard example in communications textbooks since the spectrum of $e^{jh(t)}$ can be expressed analytically in terms of an impulse train with weights determined by Bessel functions [11].) Still, this signal obeys all of our model assumptions and its estimated spectrum generally does indeed stay within the "butterfly shape" predicted by our theory. Moreover, for each of the four videos shown in Figure 4, we see that the empirical bandwidth Ω_t equals roughly 52 to 57% of the bandwidth $2\Omega_x\Gamma + 2\Omega_h$ predicted by our theory.

3.1.5 Experiments beyond our model assumptions

Our formal analysis and the experiments in Section 3.1.4 pertain specifically to translational videos in which $g(x)$ is bandlimited, $h(t)$ has either constant velocity or bounded velocity and bandwidth, and the entire contents of the frame translate *en masse*. However, real world videos may contain objects whose appearance (neglecting translation) changes over time, objects that move in front of a stationary background, multiple moving objects, and so on. We suspect that as a general rule of thumb, the temporal bandwidth of real world videos will be dictated by the same tradeoffs of spatial resolution and object motion that our theory suggests. In particular, the prediction of $2\Omega_x\Gamma + 2\Omega_h$ given by our theory may be approximately correct, if we let Ω_x be the essential spatial bandwidth of the imaging system, Γ be the maximum speed of any object moving in the video, and Ω_h be the essential bandwidth of any object motion. This last parameter is perhaps the most difficult to predict for a given video, but we suspect that in many cases its value will be small and thus its role minor in determining the overall temporal bandwidth.

To support these conjectures, we have identified the characteristic "butterfly" shape in the following experiments. However, it remains an open problem to back this up with theoretical analysis.

Removing the bandlimitedness assumption from $g(x)$

We now consider a non-bandlimited profile function $g(x)$ constructed by convolving a Gaussian bump having standard deviation σ (which we may set as desired) with the unit-step function $u(x)$. While $g(x)$ is not strictly bandlimited, we may crudely approximate its essential bandwidth according to the formula $\Omega_x \approx \frac{6}{\sigma}$. For the moment, we will continue to use a bandlimited model for $h(t)$, choosing the "sum of sinc functions" as in (6), where we can specify the bandwidth as Ω_h and the maximum value of $|h(t)|$ as Γ.

Figure 5 shows the corresponding videos and estimated spectra, where the four rows correspond to the same four parameter sets for $(\Omega_x, \Omega_h, \Gamma)$ specified in Table 1. Although the video is not strictly bandlimited spatially, we do see a reasonable decay of the spectrum in the vicinity of our predicted bandwidth Ω_x. Aside from this distinction, the behavior follows our theory very closely, and the estimated spectrum generally follows the predicted butterfly shape. For each of the four videos shown in Figure 5, we see that the empirical bandwidth Ω_t equals roughly 5 to 20% of the bandwidth $2\Omega_x\Gamma + 2\Omega_h$ predicted by our theory. These lower ratios compared to our experiment in Figure 3 likely result from the decay of the Gaussian spectrum (compared to the flat spectrum of a sinc function) and due to our conservative formula relating Ω_x to σ.

As a second experiment, we consider a non-bandlimited profile function[2]

$$g(x) = \begin{cases} 0, & x \leq -\frac{\sigma}{2} \\ \frac{1}{2} + \frac{x}{\sigma}, & -\frac{\sigma}{2} < x < \frac{\sigma}{2} \\ 1, & x \geq \frac{\sigma}{2} \end{cases}$$

where in this case σ denotes the width of the linear transition region from 0 to 1 and can be set as we desire. This function has slower decay in the frequency domain, but we may crudely approximate its essential bandwidth according to the formula $\Omega_x \approx \frac{12}{\sigma}$. We again use the "sum of sinc functions" model for $h(t)$. Figure 6 shows the corresponding videos and estimated spectra, where the four rows correspond to the same four parameter sets for $(\Omega_x, \Omega_h, \Gamma)$ specified in Table 1. We notice that while the basic behavior is the same as above, the slower decay of the spectrum in the spatial direction now gives rise to "longer wings" of the butterfly shape. Despite this, the estimated essential temporal bandwidths remain quite modest, typically with Ω_t equal to roughly 9 to 26% of the bandwidth $2\Omega_x\Gamma + 2\Omega_h$ predicted by our theory. However, one should take caution in interpreting this comparison given the crudeness of our relation between Ω_x and σ.

Removing the bandlimitedness assumption from $h(t)$

We can also construct with videos for which the motion function $h(t)$ is not bandlimited. For example, for each of the four parameter sets for $(\Omega_x, \Omega_h, \Gamma)$ specified in Table 1, we construct videos of the form $f(x,t) = g(x - h(t))$, where $g(x)$ is the non-bandlimited Gaussian-filtered step function with approximate bandwidth Ω_x (as described in

[2]It is debatable whether such a profile would be realistic for a physical imaging system.

Section 3.1.5), and $h(t)$ is a periodic triangle wave having fundamental frequency Ω_h and maximum slope Γ. We emphasize that this triangle wave $h(t)$ is *not* bandlimited, but for the sake of comparison with our other experiments, we set its fundamental frequency to be equal to the desired Ω_h.

Figure 7 shows the corresponding videos and estimated spectra. We see a distinct "striped" pattern in the estimated spectra due to the temporal periodicity of the video (recall Section 3.1.4). Although the video is not strictly bandlimited spatially or temporally, we do see a reasonable decay of the spectrum in the vicinity of our predicted spatial and temporal bandwidths Ω_x and $2\Omega_x\Gamma + 2\Omega_h$, respectively. (The decay is less strong in the temporal direction due to the stronger high-frequency content of $h(t)$ compared to $g(x)$.) Overall, the behavior still generally follows our theory, and the estimated spectrum is somewhat concentrated within the predicted butterfly shape, though this is less true than in some of our other experiments due to the high-frequency content of $h(t)$. For each of the four videos shown in Figure 7, we see that the empirical bandwidth Ω_t equals roughly 10 to 55% of the bandwidth $2\Omega_x\Gamma + 2\Omega_h$ predicted by our theory.

Videos with multiple moving edges

We now extend our experiments to account for videos with multiple moving edges. We let $g(x)$ be a Gaussian-filtered step function as described in Section 3.1.5, and we consider two moving edges with translation signals $h(t)$ obeying the "sum of sinc functions" model. Thus, our setup is exactly as described in Section 3.1.5, except that we have two edges moving simultaneously, instead of one.

Figure 8 shows the corresponding videos and estimated spectra, where the four rows correspond to the same four parameter sets for $(\Omega_x, \Omega_h, \Gamma)$ specified in Table 1. Despite the presence of multiple moving edges, the behavior of the estimated spectrum is virtually the same as that of the single-edge experiment in Figure 5, with much of the spectrum falling within the predicted butterfly-shaped region. Moreover, for each of the four videos shown in Figure 8, we see that the empirical bandwidth Ω_t again equals roughly 5 to 20% of the prediction $2\Omega_x\Gamma + 2\Omega_h$.

Videos with occlusions

To examine the effects of occlusions, we now consider videos constructed as follows. We construct $h(t)$ using the "sum of sinc functions" as in (6), where we can specify the bandwidth as Ω_h and the maximum value of $|h(t)|$ as Γ. We then consider a translated unit-step function $u(x - h(t))$ which moves in front of a stationary background consisting of a square wave taking values 0 and $\frac{1}{2}$ and with period 8 pix. Wherever $u(x - h(t)) = 1$, the translated step occludes the background; wherever $u(x-h(t)) = 0$, the background is unoccluded. This "occluded background" signal is then filtered spatially with a Gaussian kernel having standard deviation σ. (We apply the filtering after occlusion, rather than vice versa, because this seems more plausible in a real imaging system.) For the purposes of our experiments, we estimate the spatial bandwidth of this filtered signal to be $\Omega_x \approx \frac{6}{\sigma}$.

Figure 9 shows the corresponding videos and estimated spectra, where the four rows correspond to the same four parameter sets for $(\Omega_x, \Omega_h, \Gamma)$ specified in Table 1. Compared to Figure 5, we do see somewhat slower decay in the estimated spectrum along the temporal direction. Despite this, there does remain a distinct concentration of the spectrum in the predicted butterfly region, and for each of the four videos shown in Figure 9, the empirical bandwidth Ω_t equals roughly 5 to 15% of the prediction $2\Omega_x\Gamma + 2\Omega_h$. It is interesting to note that the "temporal spreading" of the spectrum appears to be less pronounced than in the experiment of Section 3.1.5, in which $h(t)$ was taken to be a triangle wave; one would expect both cases to suffer from the nondifferentiability of the video.

Real world videos

We conclude this examination with a series of experiments on real-world videos. These videos (courtesy of MERL) were collected in a laboratory setting using a high-speed video camera. For each video, we select a 2D "slice" of the 3D video cube, extracting one spatial dimension and one temporal dimension.

We begin with the *Candle* video which features two candle flames in front of a dark background; the video was acquired at a rate of 1000 frames per second. We extract 512 pixels from row 300 of each video frame. Figure 10 shows the videos and estimated spectra, where from top to bottom, we keep the first 512, 2048, 4096, and 8192 time samples of the video. In the estimated spectra, we recognize the approximate butterfly shape and an approximate limitation to the video bandwidth, both spatially and temporally. More specifically, we typically see a collection of lines with various slopes, with the lines passing roughly through the origin. However, there is also a bit of "thickness" near the origin due to a possible Ω_h term. The slopes of these lines match what might be expected based on an empirical examination of the video itself. For example, during the first 512 frames of the video, the candle flame moves relatively slowly. From the windowed FFT, we see that the slope of the butterfly wings equals approximately 150 pix/s, whereas from the video itself, we see that the maximum translational speed of the candle flame appears to equal roughly 100 pix/s. The ratio of 1.5 between these estimates is within the maximum ratio of

2.0 predicted by our theory. As we consider more frames of the candle video (2048 or more), we begin to see faster motion of the candle flames. This increases the slope of the butterfly wings as expected. For example, at roughly 2.6 seconds into the video, the candle flame appears to translate to the right with a speed of roughly 1500 pix/s, and consequently we see a portion of the estimated spectrum oriented along a line with slope of approximately 4000 pix/s. Overall, for the four videos shown in Figure 10, the empirical temporal bandwidth (the value of Ω_t for which 99% of the energy in the windowed FFT falls within the range $|\omega_t| \leq \Omega_t$) equals 50 rad/s, 117 rad/s, 348 rad/s, and 468 rad/s, respectively. This suggests that the video's temporal sampling rate (1000 frames/sec) may have been higher than necessary.

Next, we consider the *Pendulum + Cars* video, featuring two translating cars and an occlusion as one car passes in front of the other; the video was acquired at a rate of 250 frames per second. We extract 640 pixels from row 390 of each video frame. Figure 11 shows the videos and estimated spectra, where from top to bottom, we keep the first 512, 2048, 4096, and 8192 time samples of the video. Once again, in the estimated spectra, we recognize the approximate butterfly shape and an approximate limitation to the video bandwidth, both spatially and temporally, and once again, we typically see a collection of lines with various slopes, but now with a bit more "thickness" near the origin due to a possible Ω_h term. The slopes of these lines match what might be expected based on an empirical examination of the video itself. For example, the maximum slope appears to be on the order of 140 pix/s, while the maximum translational speed of the cars appears to be on the order of 70 pix/s; however, a closer inspection of the video would be warranted to make this number more precise. The overall empirical temporal bandwidth is typically on the order of 25 to 35 rad/s, and consequently, this video's temporal sampling rate (250 frames/sec) may also have been higher than necessary.

Finally, we consider the *Card + Monster* video, featuring a playing card translating in front of a dark background; the video was acquired at a rate of 250 frames per second. We extract 640 pixels from row 40 of each video frame. Figure 12 shows the videos and estimated spectra, where from top to bottom, we keep the first 512, 2048, 4096, and 8192 time samples of the video. Our conclusions are similar to the experiments above, though we do see a more significant "thickness" in the butterfly shape for this video.

3.2 Videos with two spatial dimensions

3.2.1 Problem setup

Signal model

Our analysis is easily generalized to the more conventional case of videos having two spatial dimensions. We will again use the variable $t \in \mathbb{R}$ to index time in seconds, and we will use the variables $x, y \in \mathbb{R}$ to index spatial position in pix.

Let $g(x, y)$ denote a 2D function of space (think of this as a continuous-space "still image"), and consider a continuous-space, continuous-time video $f(x, y, t)$ in which each "frame" of the video consists of a shifted version of this prototype frame. More formally, suppose that

$$f(x, y, t) = g(x - h_x(t), y - h_y(t))$$

where $h(t) = (h_x(t), h_y(t))$ is a function that controls how much (in pix) the prototype frame is shifted in the x- and y-directions at each time step.

Fourier setup

Let $F(\omega_x, \omega_y, \omega_t)$ denote the 3D Fourier transform of $f(x, y, t)$, and let $G(\omega_x, \omega_y)$ denote the 2D Fourier transform of $g(x, y)$. Keeping track of units, ω_x and ω_y are measured in terms of rad/pix, and ω_t is measured in terms of rad/s. The following relationships will be of use to us.

First, let $\mathcal{F}_{x,y}\{\cdot\}$ denote an operator on $L_2(\mathbb{R}^3)$ that performs the 2D Fourier transform in the spatial directions, and let $\mathcal{F}_t\{\cdot\}$ denote an operator on $L_2(\mathbb{R}^3)$ that performs the 1D Fourier transform in the temporal direction. Due to the separability of the 3D Fourier transform, we know that

$$F(\omega_x, \omega_y, \omega_t) = \mathcal{F}_{x,y}\{\mathcal{F}_t\{f\}\}(\omega_x, \omega_y, \omega_t) = \mathcal{F}_t\{\mathcal{F}_{x,y}\{f\}\}(\omega_x, \omega_y, \omega_t). \tag{7}$$

Now, due to the shift property of the 1D Fourier transform, we have

$$\mathcal{F}_{x,y}\{f\}(\omega_x, \omega_y, t) = \mathcal{F}_{x,y}\{g(x - h_x(t), y - h_y(t))\}(\omega_x, \omega_y, t) = G(\omega_x, \omega_y)e^{-j\omega_x h_x(t) - j\omega_y h_y(t)}.$$

Again, this function is complex-valued and has constant magnitude in the temporal direction; therefore it has only phase changes in the temporal direction.

Following (7), F will be the result of taking the 1D Fourier transform of this function $\mathcal{F}_{x,y}\{f\}$ in the temporal direction. We have

$$F(\omega_x, \omega_y, \omega_t) = \mathcal{F}_t\{G(\omega_x, \omega_y)e^{-j\omega_x h_x(t) - j\omega_y h_y(t)}\}(\omega_x, \omega_y, \omega_t) = G(\omega_x, \omega_y) \cdot L(\omega_x, \omega_y, \omega_t),$$

where

$$L(\omega_x, \omega_y, \omega_t) := \mathcal{F}_t\{e^{-j\omega_x h_x(t) - j\omega_y h_y(t)}\}(\omega_x, \omega_y, \omega_t).$$

For fixed ω_x, ω_y, $L(\omega_x, \omega_y, \omega_t)$ equals the 1D Fourier transform of $e^{-j\omega_x h_x(t) - j\omega_y h_y(t)}$ with respect to time, evaluated at the frequency ω_t.

3.2.2 Temporal bandwidth analysis

Constant velocity model for $h_x(t)$ and $h_y(t)$
Suppose the translation has a constant velocity, i.e., that $h_x(t) = \Gamma_x t$ and that $h_y(t) = \Gamma_y t$ for some constants Γ_x, Γ_y (having units of pix/s). In this case we have $L(\omega_x, \omega_y, \omega_t) = \delta(\omega_t - \omega_x\Gamma_x - \omega_y\Gamma_y)$, and so

$$F(\omega_x, \omega_y, \omega_t) = G(\omega_x, \omega_y) \cdot \delta(\omega_t - \omega_x\Gamma_x - \omega_y\Gamma_y),$$

which corresponds to a diagonal line in the 3D Fourier plane.

To see the implications of this in terms of bandwidth, let us suppose that $G(\omega_x, \omega_y)$ is bandlimited (or essentially bandlimited) to the range of frequencies $(\omega_x, \omega_y) \in [-\Omega_x, \Omega_x] \times [-\Omega_y, \Omega_y]$ rad/pix. In this case, it follows that $F(\omega_x, \omega_y, \omega_t)$ must be bandlimited (or essentially bandlimited) to the range of frequencies $(\omega_x, \omega_y, \omega_t) \in [-\Omega_x, \Omega_x] \times [-\Omega_y, \Omega_y] \times [-(\Gamma_x\Omega_x + \Gamma_y\Omega_y), \Gamma_x\Omega_x + \Gamma_y\Omega_y]$. In other words, the temporal bandwidth of the video is no greater than $\Gamma_x\Omega_x + \Gamma_y\Omega_y$ rad/s.

Bounded velocity model for $h_x(t)$ and $h_y(t)$
In slightly more generality, we could assume that the position functions $h_x(t)$ and $h_y(t)$ have bounded slope, i.e., that for some $\Gamma_x, \Gamma_y > 0$,

$$\left|\frac{dh_x(t)}{dt}\right| \le \Gamma_x \text{ pix/s} \quad \text{and} \quad \left|\frac{dh_y(t)}{dt}\right| \le \Gamma_y \text{ pix/s}$$

for all t. This corresponds to a bound on the "speed" at which the object can move in the video. We also assume that both translation signals $h_x(t)$ and $h_y(t)$ have bandwidths bounded by Ω_h rad/s. (This guarantees that for any fixed ω_x and ω_y, the bandwidth of $\phi(t)$ below is also bounded by Ω_h rad/s.)

For any fixed ω_x and ω_y, we can recognize $e^{-j\omega_x h_x(t) - j\omega_y h_y(t)}$ as a phase-modulated (PM) sinusoid having carrier frequency 0 and phase deviation (in radians)

$$\phi(t) = -\omega_x h_x(t) - \omega_y h_y(t),$$

or equivalently, as a frequency-modulated (FM) sinusoid having carrier frequency 0 and instantaneous frequency (in rad/s)

$$\omega_i(t) = \frac{d\phi(t)}{dt} = -\omega_x \frac{dh_x(t)}{dt} - \omega_y \frac{dh_y(t)}{dt}.$$

For am FM signal, we have

$$\frac{d\phi(t)}{dt} = \omega_d m(t),$$

where $m(t)$ is known as the modulating signal and ω_d is the frequency deviation constant. Thus, we have

$$m(t) = \frac{1}{\omega_d}\frac{d\phi(t)}{dt} = -\frac{\omega_x}{\omega_d}\frac{dh_x(t)}{dt} - \frac{\omega_y}{\omega_d}\frac{dh_y(t)}{dt}.$$

Let us also define the deviation term

$$D := \frac{\omega_d}{\Omega_h}\max|m(t)| \le \frac{|\omega_x|}{\Omega_h}\max\left|\frac{dh_x(t)}{dt}\right| + \frac{|\omega_y|}{\Omega_h}\max\left|\frac{dh_y(t)}{dt}\right| \le \frac{|\omega_x|\Gamma_x + |\omega_y|\Gamma_y}{\Omega_h}.$$

From Carson's bandwidth rule for frequency modulation, we have that for fixed ω_x and ω_y, at least 98% of the total power of $e^{-j\omega_x h_x(t) - j\omega_y h_y(t)}$ must be concentrated in the frequency range $\omega_t \in [-2(D+1)\Omega_h, 2(D+1)\Omega_h]$ rad/s. Since $D \le \frac{|\omega_x|\Gamma_x + |\omega_y|\Gamma_y}{\Omega_h}$, we conclude that at least 98% of the total power of $e^{-j\omega_x h_x(t) - j\omega_y h_y(t)}$ must be

concentrated in the frequency range $\omega_t \in [-(2|\omega_x|\Gamma_x + 2|\omega_y|\Gamma_y + 2\Omega_h), 2|\omega_x|\Gamma_x + 2|\omega_y|\Gamma_y + 2\Omega_h]$ rad/s. We note that the dependence of this bandwidth on ω_x is essentially linear.

We conclude that $L(\omega_x, \omega_y, \omega_t)$ will have a characteristic "polytope hourglass shape". Considering the first octant of the 3D frequency space (in which ω_x, ω_y, $\omega_t \geq 0$), most of the total power of $L(\omega_x, \omega_y, \omega_t)$ will fall below (in the temporal direction) a plane passing through the points $(0, 0, 2\Omega_h)$, $(1, 0, 2\Gamma_x + 2\Omega_h)$, and $(0, 1, 2\Gamma_y + 2\Omega_h)$. The other seven octants follow symmetrically.

To see the implications of this in terms of bandwidth, let us again suppose that $G(\omega_x, \omega_y)$ is bandlimited (or essentially bandlimited) to the range of frequencies $(\omega_x, \omega_y) \in [-\Omega_x, \Omega_x] \times [-\Omega_y, \Omega_y]$. We must then have that $F(\omega_x, \omega_y, \omega_t) = G(\omega_x, \omega_y) \cdot L(\omega_x, \omega_y, \omega_t)$ is also essentially bandlimited in the spatial direction to the range of frequencies $(\omega_x, \omega_y) \in [-\Omega_x, \Omega_x] \times [-\Omega_y, \Omega_y]$. Because of the hourglass structure in $L(\omega_x, \omega_y, \omega_t)$, however, this will also cause $F(\omega_x, \omega_y, \omega_t)$ to be essentially bandlimited in the temporal direction to the range of frequencies

$$\omega_t \in [-(2\Omega_x\Gamma_x + 2\Omega_y\Gamma_y + 2\Omega_h), 2\Omega_x\Gamma_x + 2\Omega_y\Gamma_y + 2\Omega_h]. \tag{8}$$

Therefore, we see that filtering such a video in the spatial directions can cause it to be essentially bandlimited both in space and in time.

3.2.3 Sampling implications

Based on the temporal bandwidth predicted in (8), where we assume that $G(\omega_x, \omega_y)$ is bandlimited (or essentially bandlimited) to the range of frequencies $(\omega_x, \omega_y) \in [-\Omega_x, \Omega_x] \times [-\Omega_y, \Omega_y]$, the Nyquist theorem suggests that in order to avoid aliasing, the video should be sampled a minimum rate of

$$\frac{2\Omega_x\Gamma_x + 2\Omega_y\Gamma_y + 2\Omega_h}{\pi} \text{ samples/s.}$$

As a point of reference, during a sampling interval of $\left(\frac{2\Omega_x\Gamma_x+2\Omega_y\Gamma_y+2\Omega_h}{\pi}\right)^{-1}$ seconds, an object moving with maximum speed of Γ_x pix/s in the x-direction can traverse a maximum of

$$\frac{\pi\Gamma_x}{2\Omega_x\Gamma_x + 2\Omega_y\Gamma_y + 2\Omega_h} \text{ pix}$$

in the x-direction. Similarly, an object moving with maximum speed of Γ_y pix/s in the y-direction can traverse a maximum of

$$\frac{\pi\Gamma_y}{2\Omega_x\Gamma_x + 2\Omega_y\Gamma_y + 2\Omega_h} \text{ pix.}$$

Using the triangle inequality for the sake of simplicity, we conclude that the object can move no more than

$$\frac{\pi(\Gamma_x + \Gamma_y)}{2\Omega_x\Gamma_x + 2\Omega_y\Gamma_y + 2\Omega_h} \text{ pix} \tag{9}$$

in any direction.

Let us plug in some plausible numbers to illustrate the implications of these bounds. If we expect that both Ω_x and Ω_y will be on the order of π rad/pix, and if we assume that the $2\Omega_x\Gamma_x + 2\Omega_y\Gamma_y$ term will typically dominate the $2\Omega_h$ term, then (9) suggests that, in typical scenarios, to avoid temporal aliasing we should not allow a moving object to traverse more than $\approx \frac{1}{2}$ pix in any direction between adjacent sampling times.

Aside from exceptionally non-smooth motions $h(t)$, we do strongly suspect that the influence of the temporal bandwidth Ω_h will be minor in comparison to the influence of the $2\Omega_x\Gamma_x + 2\Omega_y\Gamma_y$ term, and so *in general a temporal Nyquist sampling rate of* $2(\Gamma_x + \Gamma_y)$ *samples/s will likely serve as a reasonable rule of thumb.*[3] Again, this rule of thumb illustrates the direct relationship between the speed of object motion in the video and the video's overall temporal bandwidth.

3.2.4 Experiments

It would not be difficult to conduct experiments analogous to those presented in Sections 3.1.4 and 3.1.5. Due to time limitations, however, we have not included such experiments.

[3]It is possible that this could be reduced to $2\sqrt{\Gamma_x^2 + \Gamma_y^2}$ through the natural modifications to our analysis.

3.3 Sampling and interpolation principles

The insight we have developed in the past two sections suggests that many videos of interest may indeed be exactly or approximately bandlimited in the temporal direction. For problems involving CS of such videos, this implies that there may be a limit to the "complexity" of the information collected by compressive measurement devices with high temporal sampling rates. One way of exploiting this limited complexity is in the context of classical interpolation identities for bandlimited signals. We briefly review these identities in this section, before exploring their applications for CS reconstruction in Section 3.4.

3.3.1 Interpolation theory

Before considering 2D or 3D video signals, let us first review the basic principles involved in sampling, interpolation, and reconstruction of a bandlimited 1D signal. Suppose that $f(t)$ is a signal with temporal bandwidth bounded by Ω_t rad/s. The Nyquist theorem states that this signal can be reconstructed from a discrete set of samples $\{f(nT_s)\}_{n\in\mathbb{Z}}$, where the sampling interval $T_s \leq \frac{\pi}{\Omega_t}$ seconds. In particular, it holds that

$$f(t) = \sum_{n\in\mathbb{Z}} f(nT_s)\text{sinc}\left(\frac{t-nT_s}{T_s}\right), \tag{10}$$

where $\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$. Instead of actually *reconstructing* the continuous-time signal $f(t)$, a more important consequence of (10) for us will be the fact that, for any $t_0 \in \mathbb{R}$, $f(t_0)$ can be *represented* as a linear combination of the discrete samples $\{f(nT_s)\}_{n\in\mathbb{Z}}$.

With varying degrees of approximation, it is possible to replace the sinc interpolation kernel in (10) with other, more localized kernels. We will write

$$\widetilde{f}(t) = \sum_{n\in\mathbb{Z}} f(nT_s)\gamma\left(\frac{t-nT_s}{T_s}\right), \tag{11}$$

where $\gamma(t)$ is a prototype interpolation kernel. In addition to the sinc kernel, for which

$$\gamma(t) = \text{sinc}\,(t)\,,$$

other possible choices include the zero-order hold (rectangular) kernel, for which

$$\gamma(t) = \text{rect}\,(t) = \begin{cases} 1, & |t| \leq \frac{1}{2} \\ 0, & \text{otherwise,} \end{cases}$$

the first-order hold (triangular, or "linear interpolation") kernel, for which

$$\gamma(t) = \text{tri}\,(t) = \begin{cases} 1-|t|, & |t| \leq 1 \\ 0, & \text{otherwise,} \end{cases}$$

and a variety of cubic interpolation kernels [12].

In general, a smoother choice for for $\gamma(t)$ will better approximate the ideal sinc kernel.[4] However, smoother kernels tend to have wider temporal supports, and it can be desirable in some applications (such as the CS recovery problem discussed below) to limit the temporal support of the kernel. One way to improve the performance of the lower-order, more narrow interpolation kernels is to decrease the sampling interval T_s.[5] However, for our CS recovery problem discussed below, this too will increase the complexity of the recovery algorithm by increasing the number of "anchor frames".

For a 2D or 3D video with limited temporal bandwidth, the separability of the Fourier transform implies that the interpolation formulas presented above should hold for each spatial location (i.e., for each pixel). For several of the approximately bandlimited test videos considered in Sections 3.1.4 and 3.1.5, we explore below the quality of these interpolation formulas as a function of the temporal sampling rate and the interpolation kernel type.

[4] The implications of this can be appreciated in the frequency domain. Since one can view the reconstruction process as convolution of an impulse train with the scaled interpolation kernel $\gamma\left(\frac{t}{T_s}\right)$, in the frequency domain we are windowing the periodic spectrum of $f(t)$ with the frequency response of the interpolation kernel. Smoother interpolation kernels better approximate the ideal lowpass filter.

[5] Doing so provides more space in the frequency domain between the replicated copies of the spectrum of $f(t)$, permitting nonideal lowpass filters a better opportunity to select only the baseband copy and to do so with minimal distortion.

3.3.2 Experiments within our model assumptions

First, we recall Section 3.1.4 and consider videos $f(x,t) = g(x - h(t))$ with bandlimited sinc profile $g(x)$ and bandlimited "sum of sinc functions" model for the translation signal $h(t)$. In the right side Figure 13(a), we show a video with parameters $(\Omega_x, \Omega_h, \Gamma)$ drawn from the fourth row of Table 1; the time axis runs vertically. We set T_s equal to 0.5 times the Nyquist sampling interval corresponding to the predicted temporal bandwidth of $2\Omega_x\Gamma + 2\Omega_h$ rad/s. In the left side of Figure 13(b), using a linear interpolation kernel $\gamma(t)$, we plot as a function of time t the maximum interpolation error

$$\max_x |f(x,t) - \widetilde{f}(x,t)|$$

over all x in our sampling grid; the times t are selected from a fine grid, and again the time axis runs vertically and is presented for comparison with the video on the right. We see that interpolation errors are greatest for times in the video at which $h(t)$ has large slope.

For each of the four parameter sets $(\Omega_x, \Omega_h, \Gamma)$ listed in Table 1, we also compute the maximum interpolation error

$$\max_{x,t} |f(x,t) - \widetilde{f}(x,t)|$$

as a function of the temporal sampling interval T_s. Using a linear interpolation kernel for $\gamma(t)$, Figure 13(b) plots the maximum interpolation error for each type of video, where the colored lines represent each of the four video types (blue = row 1 in Table 1, green = row 2, red = row 3, and cyan = row 4), and the horizontal axis represents the ratio of T_s to the Nyquist sampling interval corresponding to the predicted temporal bandwidth of $2\Omega_x\Gamma + 2\Omega_h$ for each video. Figure 13(c) repeats this experiment using a cubic interpolation kernel $\gamma(t)$. In each case, we see that the sampling near the predicted Nyquist rate allows interpolation of the missing video frames with reasonable accuracy. Also, the similar nature of the four curves suggests that the actual values of $(\Omega_x, \Omega_h, \Gamma)$ are not critical in determining interpolation accuracy; what seems to be important is how these values combine into a predicted Nyquist sample rate, and how fast the video is sampled compared to this rate.

3.3.3 Experiments beyond our model assumptions

Removing the bandlimitedness assumption from $g(x)$

In Figure 14 we repeat all of the above experiments for the type of videos examined in Section 3.1.5, with bandlimited sinc profile $g(x)$ and bandlimited "sum of sinc functions" model for the translation signal $h(t)$. From panel (a), we see that interpolation errors are again typically highest at times for which $h(t)$ has large slope. From panels (b) and (c), however, we see that compared to Figure 13, interpolation errors for these videos are relatively lower for a given sampling rate (as a fraction of the predicted Nyquist rate). As discussed in Section 3.1.5, this is likely due to the decay of the Gaussian spectrum (compared to the flat spectrum of a sinc function) and due to our conservative formula relating Ω_x to σ.

Removing the bandlimitedness assumption from $h(t)$

In Figure 15 we repeat all of these experiments for the type of videos examined in Section 3.1.5, with non-bandlimited Gaussian-filtered step function $g(x)$ and non-bandlimited triangle wave model for $h(t)$. From panel (a), we see that interpolation errors are typically highest at times where $h(t)$ has an abrupt change of slope (at the peaks of the triangles). Despite the non-bandlimited nature of $h(t)$ we do see from panels (b) and (c) that reasonable interpolation quality is achievable as long as the video is sampled near the predicted (approximate) Nyquist rate.

Videos with occlusions

In Figure 16 we repeat all of these experiments for the type of videos examined in Section 3.1.5, with $f(x,t)$ containing a moving edge occluding a stationary background pattern. Similar conclusions hold.

Real world videos

Finally, Figure 17, Figure 18, and Figure 19 present similar experiments conducted on the *Candle*, *Pendulum + Cars*, and *Card + Monster* videos, respectively. The experiments are conducted by subsampling the original high-rate videos and attempting to interpolate the values of the omitted frames. Panel (a) of each figure illustrates, for each t, the maximum interpolation error over all x. We see that moments of high speed motion present the most difficulty for accurate interpolation. Panel (b) of each figure plots the maximum interpolation error over all x and t for both linear interpolation (solid blue line) and cubic interpolation (dashed red line) as a function of the absolute number of samples retained per second. It is clear that these videos contain some features beyond our

model assumptions, which prevents the maximum interpolation error from achieving a small value. However, we do see characteristic improvements as we increase our sample rate.

3.3.4 A manifold-based interpretation

As a brief note, we mention that one could reinterpret these tradeoffs of bandwidth vs. interpolation quality in the context of manifolds. In past work [7], we have shown that typical manifolds of images containing sharp edges are non-differentiable. However, these manifolds contain a multiscale structure that can be accessed by regularizing the images. The more one smoothes the images, the smoother the manifold will become, and local tangent approximations will remain accurate over longer distances across the manifold.

In the context of this document, the spatial profile function $g(x)$ is often assumed to be bandlimited due to an inherent resolution limit in the imaging system. Thus, our translational video model $f(x,t) = g(x-h(t))$ corresponds to a "walk" along a manifold consisting of blurred images of the same object, but translated to different positions. From the manifold perspective, a decrease in the bandwidth of $g(x)$ corresponds to an increase in the amount of blurring, which smoothes the manifold. From the perspective of this document, a decrease in the bandwidth of $g(x)$ decreases the anticipated temporal bandwidth of the video, which improves the quality of linear (and other) interpolation schemes. The reason for the improved interpolation accuracy can be appreciated by considering the increased smoothness of the manifold; in particular, linear interpolation improves as the tangent spaces twist more slowly.

These intuitive connections could be explored more formally in future work. For example, the performance of higher order interpolation methods (such as cubic interpolation) may be tied to the rate at which higher order derivatives change along the manifold.

3.4 Reconstruction of temporally bandlimited videos from streaming measurements

In this section, we will consider how the Compressive Sensing (CS) reconstruction problem can be formulated in streaming scenarios, where we acquire one measurement of a video per time instant.

3.4.1 Measurement process

Consider a continuous-space, continuous-time video $f(x,t)$ that has temporal bandwidth bounded by Ω_t rad/s. Let $f_d : \{1,2,\ldots,N_x\} \times \mathbb{R} \to \mathbb{R}$ denote a sampled discrete-space, continuous-time version of this video, where for $p = 1,2,\ldots,N_x$,

$$f_d(p,t) = f(p\Delta_x, t). \tag{12}$$

In the expression above, Δ_x represents the spatial sampling resolution (in pix). We note that the temporal bandwidth of f_d will still be bounded by Ω_t rad/s.

Now, let T_1 denote a measurement interval (in seconds); typically T_1 will be much smaller than the Nyquist sampling interval of $\frac{\pi}{\Omega_t}$ suggested by the video's bandwidth. Suppose that one linear measurement is collected from f_d every T_1 seconds. Letting $y(m)$ denote the measurement collected at time mT_1, we can write

$$y(m) = \sum_{p=1}^{N_x} \phi_m(p) f_d(p, mT_1) = \langle \phi_m, f_d(:, mT_1) \rangle, \tag{13}$$

where $\phi_m \in \mathbb{R}^{N_x}$ is a vector of random numbers, and we use "Matlab notation" to refer to a vector $f_d(:,t) \in \mathbb{R}^{N_x}$.

Stacking all of the measurements, we have

$$y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(M) \end{bmatrix} = \begin{bmatrix} \langle \phi_1, f_d(:,T_1) \rangle \\ \langle \phi_2, f_d(:,2T_1) \rangle \\ \vdots \\ \langle \phi_M, f_d(:,MT_1) \rangle \end{bmatrix} = \underbrace{\begin{bmatrix} \phi_1^T & & & \\ & \phi_2^T & & \\ & & \ddots & \\ & & & \phi_M^T \end{bmatrix}}_{M \times MN_x} \underbrace{\begin{bmatrix} f_d(:,T_1) \\ f_d(:,2T_1) \\ \vdots \\ f_d(:,MT_1) \end{bmatrix}}_{MN_x \times 1} \tag{14}$$

Unfortunately, there are two difficulties with attempting to use this formulation for a CS recovery: first, the recovery problem is very highly underdetermined, with the number of measurements representing only $\frac{1}{N_x}$ times the number of unknowns; and second, the size of the recovery problem, with MN_x unknowns, can be immense. (Both of these difficulties will be even more significant for videos with two spatial dimensions.)

3.4.2 Simplifying the linear equations

Fortunately, the bandlimitedness of the video allows us to simplify this recovery process somewhat. Let T_s denote a sampling interval no greater than the Nyquist limit ($\frac{\pi}{\Omega_t}$ seconds) suggested by the video's bandwidth, and assume that $T_s = VT_1$ for some integer $V \geq 1$. Then for any integer j, we can write

$$\begin{aligned}
f_d(:, jT_1) &= \sum_{n\in\mathbb{Z}} f_d(:, nT_s)\gamma\left(\frac{jT_1 - nT_s}{T_s}\right) \\
&= \sum_{n\in\mathbb{Z}} f_d(:, nT_s)\gamma\left(\frac{j\frac{T_s}{V} - nT_s}{T_s}\right) \\
&= \sum_{n\in\mathbb{Z}} f_d(:, nT_s)\gamma\left(\frac{j}{V} - n\right) \\
&= \begin{bmatrix} \cdots & \gamma\left(\frac{j}{V}-1\right)I_{N_x} & \gamma\left(\frac{j}{V}-2\right)I_{N_x} & \gamma\left(\frac{j}{V}-3\right)I_{N_x} & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ f_d(:, T_s) \\ f_d(:, 2T_s) \\ f_d(:, 3T_s) \\ \vdots \end{bmatrix},
\end{aligned}$$

where I_{N_x} denotes the $N_x \times N_x$ identity matrix. Therefore,

$$\begin{bmatrix} f_d(:, T_1) \\ f_d(:, 2T_1) \\ \vdots \\ f_d(:, MT_1) \end{bmatrix} = \begin{bmatrix} \cdots & \gamma\left(\frac{1}{V}-1\right)I_{N_x} & \gamma\left(\frac{1}{V}-2\right)I_{N_x} & \gamma\left(\frac{1}{V}-3\right)I_{N_x} & \cdots \\ \cdots & \gamma\left(\frac{2}{V}-1\right)I_{N_x} & \gamma\left(\frac{2}{V}-2\right)I_{N_x} & \gamma\left(\frac{2}{V}-3\right)I_{N_x} & \cdots \\ & \vdots & \vdots & \vdots & \\ \cdots & \gamma\left(\frac{M}{V}-1\right)I_{N_x} & \gamma\left(\frac{M}{V}-2\right)I_{N_x} & \gamma\left(\frac{M}{V}-3\right)I_{N_x} & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ f_d(:, T_s) \\ f_d(:, 2T_s) \\ f_d(:, 3T_s) \\ \vdots \end{bmatrix} \tag{15}$$

Assuming $\gamma(t)$ has temporal support within some reasonable bound, the matrix above will have size $MN_x \times (\frac{MN_x}{V} + O(1))$ and so this allows a dimensionality reduction by a factor of V.

Putting all of this together, we have

$$\begin{aligned}
y &= \underbrace{\begin{bmatrix} \phi_1^T & & & \\ & \phi_2^T & & \\ & & \ddots & \\ & & & \phi_M^T \end{bmatrix}}_{M\times MN_x} \underbrace{\begin{bmatrix} f_d(:, T_1) \\ f_d(:, 2T_1) \\ \vdots \\ f_d(:, MT_1) \end{bmatrix}}_{MN_x\times 1} \\
&= \underbrace{\begin{bmatrix} \phi_1^T & & & \\ & \phi_2^T & & \\ & & \ddots & \\ & & & \phi_M^T \end{bmatrix}}_{M\times MN_x} \underbrace{\begin{bmatrix} \cdots & \gamma\left(\frac{1}{V}-1\right)I_{N_x} & \gamma\left(\frac{1}{V}-2\right)I_{N_x} & \gamma\left(\frac{1}{V}-3\right)I_{N_x} & \cdots \\ \cdots & \gamma\left(\frac{2}{V}-1\right)I_{N_x} & \gamma\left(\frac{2}{V}-2\right)I_{N_x} & \gamma\left(\frac{2}{V}-3\right)I_{N_x} & \cdots \\ & \vdots & \vdots & \vdots & \\ \cdots & \gamma\left(\frac{M}{V}-1\right)I_{N_x} & \gamma\left(\frac{M}{V}-2\right)I_{N_x} & \gamma\left(\frac{M}{V}-3\right)I_{N_x} & \cdots \end{bmatrix}}_{MN_x\times(\frac{MN_x}{V}+O(1))} \underbrace{\begin{bmatrix} \vdots \\ f_d(:, T_s) \\ f_d(:, 2T_s) \\ f_d(:, 3T_s) \\ \vdots \end{bmatrix}}_{(\frac{MN_x}{V}+O(1))\times 1} \\
&= \underbrace{\begin{bmatrix} \cdots & \gamma\left(\frac{1}{V}-1\right)\phi_1^T & \gamma\left(\frac{1}{V}-2\right)\phi_1^T & \gamma\left(\frac{1}{V}-3\right)\phi_1^T & \cdots \\ \cdots & \gamma\left(\frac{2}{V}-1\right)\phi_2^T & \gamma\left(\frac{2}{V}-2\right)\phi_2^T & \gamma\left(\frac{2}{V}-3\right)\phi_2^T & \cdots \\ & \vdots & \vdots & \vdots & \\ \cdots & \gamma\left(\frac{M}{V}-1\right)\phi_M^T & \gamma\left(\frac{M}{V}-2\right)\phi_M^T & \gamma\left(\frac{M}{V}-3\right)\phi_M^T & \cdots \end{bmatrix}}_{M\times(\frac{MN_x}{V}+O(1))} \underbrace{\begin{bmatrix} \vdots \\ f_d(:, T_s) \\ f_d(:, 2T_s) \\ f_d(:, 3T_s) \\ \vdots \end{bmatrix}}_{(\frac{MN_x}{V}+O(1))\times 1}
\end{aligned} \tag{16}$$

Now we see that we have reduced the total number of unknowns, from MN_x to $\frac{MN_x}{V} + O(1)$. Indeed, the largest dimension of any matrix or vector in this formulation is now limited to $\frac{MN_x}{V} + O(1)$ instead of MN_x. Moreover,

due to decay in γ, the matrix above will be banded, with zeros in all positions sufficiently far from the diagonal. This facilitates storage of the matrix and possible streaming reconstruction.

From the formulation above, we see that it is possible to focus the reconstruction process on recovery of a relatively low-rate stream of "anchor frames", rather than the high rate stream of frames measured by the imaging system. If the anchor frames are defined at the video's temporal Nyquist rate, and if there are no additional assumptions made about the video, then one should not expect any temporal correlations to remain among the anchor frames. In many real world settings, however, there will be objects moving within the scene, and the smoothness of the object motion can lead to temporal correlations, e.g., that can be captured via motion compensated transforms. Thus, in order to impose the strongest possible model on the vector of anchor frames, it may be necessary to look for sparsity in a motion compensated wavelet transform, to invoke a dynamical system model for the changing anchor frames, or to find some other model for exploiting temporal correlations. In many applications, this may indeed be the only way to reduce the number of unknowns to some quantity smaller than M and thus to truly be able to solve the CS inverse problem.

3.4.3 Experiments

Single moving pulse

Let us illustrate the basic idea through the following experiment. We consider a video $f(x,t) = g(x - h(t))$ having one spatial dimension. The profile $g(x)$ is taken to be a Gaussian pulse having standard deviation $\sigma = 0.5$. Unlike earlier experiments in this document, we do *not* convolve the Gaussian with a unit step function for this experiment. We may crudely approximate the essential bandwidth of $g(x)$ according to the formula $\Omega_x \approx \frac{6}{\sigma} = 12$ rad/pix. For the translational signal $h(t)$ we use the "sum of sinc functions" model (6), with $\Omega_h = 8$ rad/s and $\Gamma = 50$ pix/s. This video has a predicted temporal bandwidth of $2\Omega_x\Gamma + 2\Omega_h = 1216$ rad/s, and so Nyquist suggests a minimum temporal sampling rate of approximately 387 samples/sec. We discretize this video in space, taking $N_x = 50$ pixels/frame at a spatial sampling resolution of $\Delta_x = 1$ pix/pixel. Following (12), we let $f_d(p,t)$ denote the discretized video; the video is shown in Figure 20(a).

We then set $T_1 = \frac{1}{2 \cdot 387} = \frac{1}{774}$ seconds, and letting m range from 1 to 774 (i.e., t ranges from 0 to 1 sec), we collect a total of 774 random projections of this video, one at a time in equispaced intervals, as dictated by (13). To be clear, for each m, the measurement we collect at time mT_1 represents the inner product of $f_d(:, mT_1)$ against a length-50 random vector. In Figure 20(b), we follow (14) and employ a standard CS reconstruction algorithm[6] to reconstruct f_d at all of the time instants $\{mT_1\}_{m=1}^{774}$. In total, we are solving for $50 \cdot 774 = 38700$ unknowns using just 774 random measurements, and since the sparsity level of this video in the canonical basis is clearly at least 774 (at least one pixel is nonzero at every time instant), there is no hope for a decent reconstruction.

For various values of V, then, we consider the formulation (16), using a linear interpolation kernel $\gamma(t)$, and reconstruct only the anchor frames. Since the anchor frames are spaced approximately $\frac{V}{774}$ seconds apart, the number of unknowns we are solving for reduces to approximately $\frac{38700}{V}$; we are attempting solve for these unknowns using the same 774 measurements. After solving for the anchor frames, we employ (15) to recover an estimate for f_d at all of the time instants $\{mT_1\}_{m=1}^{774}$. Figures 20(c) through 20(g) show the resulting reconstructions for $V = 8$, $V = 12$, $V = 16$, $V = 20$, and $V = 24$, respectively. Figure 20(h) plots the MSE of the discretized frames $f_d(:,t)$ as a function of time, for each of the various values of V.

Across a wide range of V, we see in these experiments that the reconstruction error is significantly improved compared to the full-scale reconstruction suggested by (14). However, we do see that by choosing V too low, there is a degradation in performance, which is presumably due to the fact that we must solve for too many unknowns given the available measurements. We also see that by choosing V too large, there is a degradation in performance, which is presumably due to the fact that the interpolation equation (15) on which we base our technique, breaks down. For a fixed V, we also see that the reconstruction error increases as the slope of $h(t)$ increases, which is to be expected and is again due to the fact that (15) is breaking down.

We can also examine the role played by the choice of interpolation kernel. In Figure 21 we repeat the above experiment, using a rectangular nearest neighbor interpolation kernel $\gamma(t)$ instead of a linear interpolation kernel. With this choice of kernel, the accuracy of (15) is diminished, and the reconstruction performance suffers as one would expect. By comparing Figure 21 to Figure 20 we see the true potential value in our kernel-based approach, as Figure 21 could be interpreted as a simple "clustering" of the measurements in time, which is a natural approach that comes to mind when one considers streaming measurements from an architecture such as the Rice single pixel camera.

[6] This video was chosen due to the fact that it is sparse in the canonical basis; in practice other sparse bases could be employed for reconstruction.

Two moving pulses
To test the performance of these techniques when the complexity of the video is increased, we repeat the above experiment (using a linear interpolation kernel) for a video containing two moving pulses. The results are shown in Figure 22. It is clear that the sparsity level of this video in the canonical basis is roughly twice that of the video from our initial experiment above. Consequently, the reconstruction performance is diminished in all cases. All of our relative statements about the role of V and the implications of the slope of $h(t)$ carry over, however.

Real world video
We have also experimented with the *Candle* video discussed previously; see Figure 23(a). In this case, we select $N_x = 225$ pixels from row 300 of the video. We consider 400 adjacent frames of the video, which were collected at a sample rate of 1000 frames/sec. To construct streaming random measurements of this video, we actually compute 5 random measurements of each recorded frame;[7] thus the total number of random measurements collected is $5 \cdot 400 = 2000$.

In Figure 23(b), we adapt (14) to account for the fact that 5 (rather than 1) measurements are collected at each time instant and employ a standard CS reconstruction algorithm[8] to reconstruct the 400 frames. In total, we are solving for $225 \cdot 400 = 90000$ unknowns using just 2000 random measurements. Since the sparsity level of this video exceeds 10000, there is no hope for a decent reconstruction.

For various values of V, then, we adapt (16) as appropriate and, using a linear interpolation kernel $\gamma(t)$, reconstruct only the anchor frames. In each case, the number of unknowns we are solving for reduces to approximately $\frac{90000}{V}$; we are attempting solve for these unknowns using the same 2000 measurements. After solving for the anchor frames, we employ (15) to recover an estimate for all 400 original frames. Figures 23(c) through 23(g) show the resulting reconstructions for $V = 8$, $V = 12$, $V = 20$, $V = 30$, and $V = 40$, respectively. Figure 23(h) plots the MSE of the reconstructed frames as a function of time, for each of the various values of V. Overall, these experiments illustrate the same issues concerning the role of V and speed of the motion in the video.

3.4.4 Toward a general purpose reconstruction algorithm

The preceding experiments provide a promising proof of concept for interpolation-based simplifications of the CS recovery problem. There are many directions in which this basic idea may be extended in developing a more general purpose CS reconstruction algorithm. Several such directions are briefly surveyed below.

Streaming reconstruction
In practice, we may be presented with a long stream of measurements and wish to reconstruct a video over a long time span. While it would be possible in theory to exploit the relationship (16) over the whole video at once, it would also be possible to window the measurements into smaller segments and reconstruct the segments individually using (16). If the segments are chosen to overlap slightly, then the final anchor frames in one segment could be used as a prior to inform the reconstruction of the initial anchor frames in the next segment. Or, when a start/end anchor frame is not available for properly initializing a reconstruction based on (16), certain interference cancellation schemes could be exploited to account for the fact that those anchor frames may not be properly recoverable.

Optimizing anchor spacing
In streaming or non-streaming reconstructions, one would have the option of changing the anchor spacing (parameterized by V) adaptively throughout the video, perhaps based on real-time estimates of the speed of object motion in the video. Our plots in Figures 20(h), 21(h), 22(h), and 23(h), however, suggest that the favorable choices of V do not change significantly over time. A deeper study of real world videos would be required to characterize the potential gains of changing V on the fly.

Optimizing the spatial resolution of the measurements
One could also imagine a compressive imaging architecture that, in times of fast object motion,[9] adaptively reduces the spatial resolution of the measurement vectors in order to artificially reduce the spatial (and thus also the temporal) bandwidth of the video. The goal of throttling back the measurement resolution in such a manner

[7] Roughly speaking, this would correspond to the operation of a "five pixel" camera. Though such an architecture is perhaps not realistic in practice, we needed some way to construct more than 400 measurements for the 400 frames of interest.

[8] Again, for simplicity this video was chosen due to its sparsity in the canonical basis. However, reconstruction of this and other videos could likely be improved through the use of sparse transforms that exploit spatial correlations.

[9] The video could perhaps be probed periodically with repeated pairs of measurement functions to detect object speeds.

would be to better ensure that the identity (15) would approximately hold. To test the viability of such an approach, we repeat in Figure 24 the same experiment presented in Figure 20, but where each measurement vector has been lowpass filtered and downsampled by a factor of 2. Similarly, Figure 25 shows the results for a video with two moving pulses; it should be compared with Figure 22.

In general, we see that reducing the measurement resolution reduces the spatial resolution of the reconstructed frames. Although we rarely see an overall gain in terms of mean square error, there can occasionally be clear visual improvements in the lower resolution reconstructions. A deeper study of real world videos would be required to better characterize the potential gains of changing the measurement resolution on the fly. It would also be useful to consider whether it is possible to adapt just the reconstruction resolution without changing the actual measurement resolution.

Future work: exploiting residual correlations among anchor frames

In the hypothetical case where no residual correlations remain among the anchor frames, there would be little room for improvement upon this reconstruction approach. In real-world videos, however, we do expect that significant correlations may remain. The question of how to exploit these correlations has been addressed in the more conventional CS recovery literature. For example, one could employ a linear dynamical system model to capture the restricted degrees of freedom of the anchor frames [13]. Alternatively, one could attempt to estimate the motion of the video, and then formulate the recovery of the anchor frames in a motion-compensated sparse basis [14]. As better solutions continue to be developed for exploiting such correlations, we expect that they could be combined with the methods discussed to permit better reconstruction from streaming or "single-pixel" measurements. It may also be possible to combine our insights with techniques for "manifold lifting" [15], building on the connections discussed briefly in Section 3.3.4.

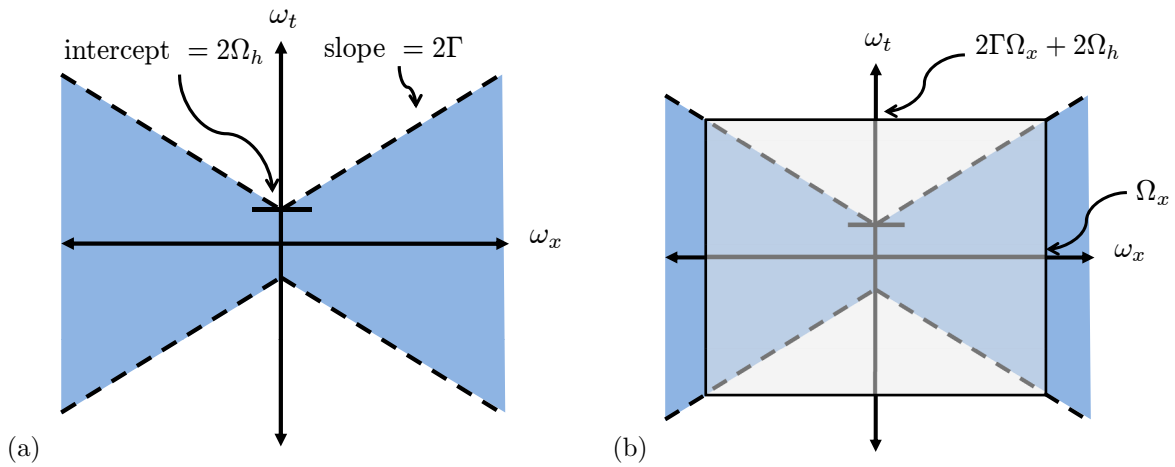


Figure 1: *(a) Butterfly structure in the two-dimensional Fourier transform of a simple translational 1D video. The slope of the lines is proportional to the maximum speed of the translation. (b) Bandlimiting the video in space (e.g., by spatial lowpass filtering) will also bandlimit the video in time. The resulting temporal bandwidth will be proportional to the spatial bandwidth times the maximum translational speed.*

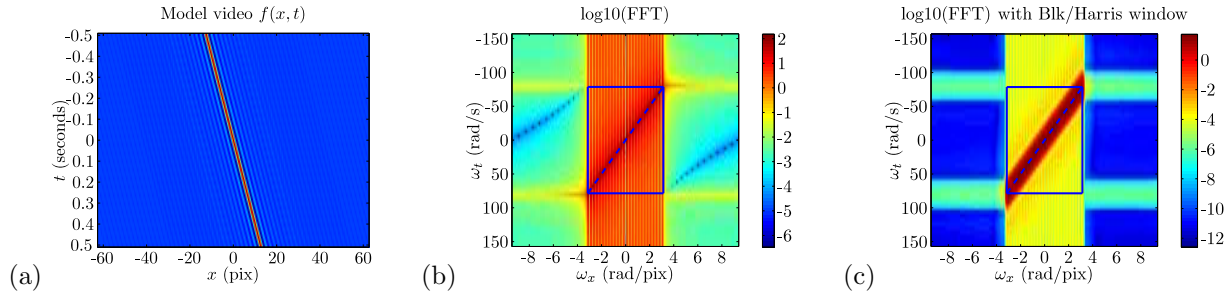


Figure 2: *(a) Video $f(x,t) = g(x - h(t))$ with bandlimited sinc profile $g(x)$ and constant velocity translation signal $h(t)$; see Section 3.1.4. (b) Estimated spectrum. (c) Estimated spectrum with windowing to alleviate border artifacts. For code, see* `mbwBLtests11.m`.

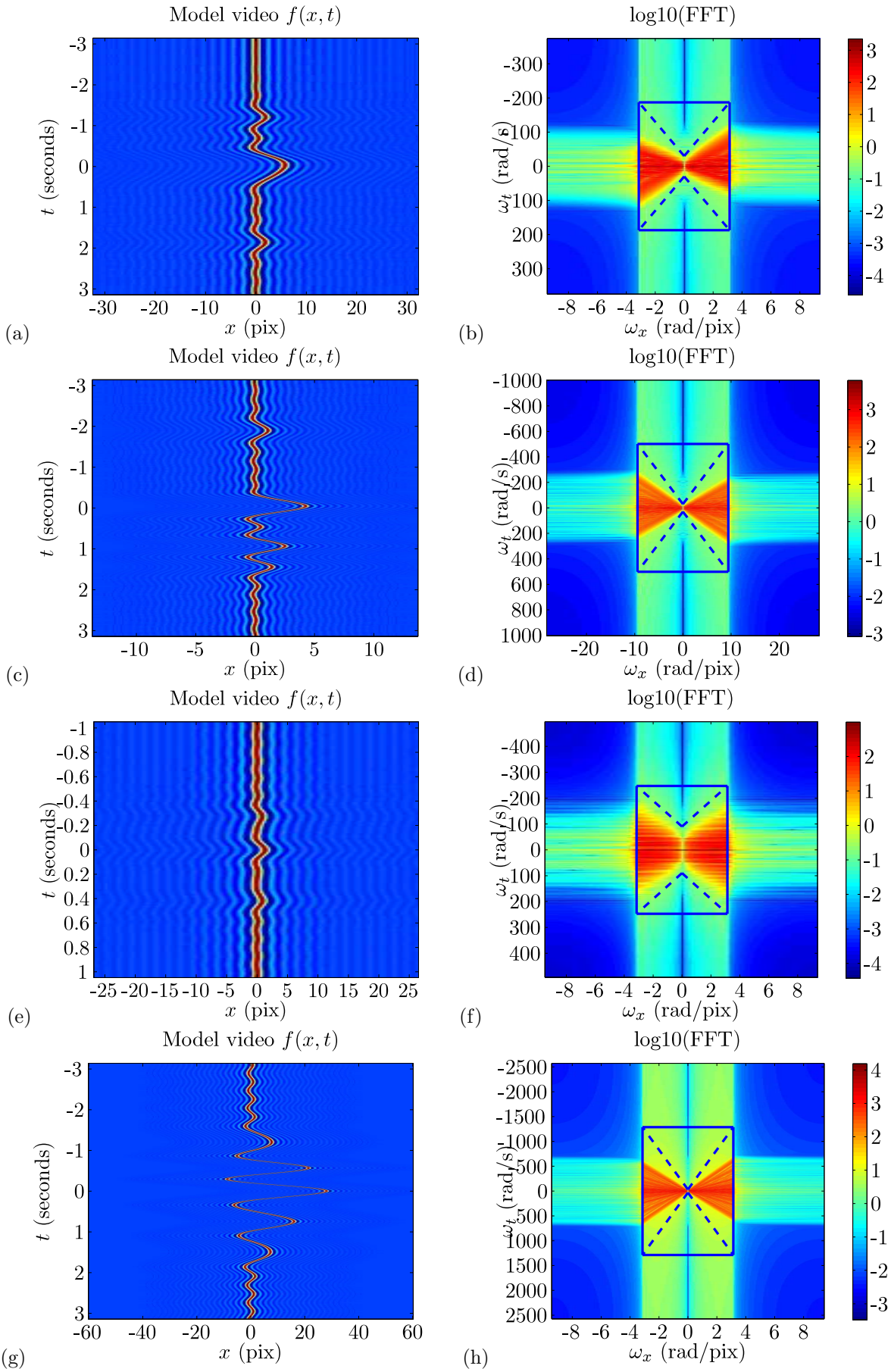


Figure 3: *Left column: Videos $f(x,t) = g(x - h(t))$ with bandlimited sinc profile $g(x)$ and bandlimited "sum of sinc functions" model for the translation signal $h(t)$; see Section 3.1.4. Each row corresponds to one set of parameter values $(\Omega_x, \Omega_h, \Gamma)$ as specified in Table 1. Right column: Estimated spectra. For code, see* `mbwBLtests12.m`.

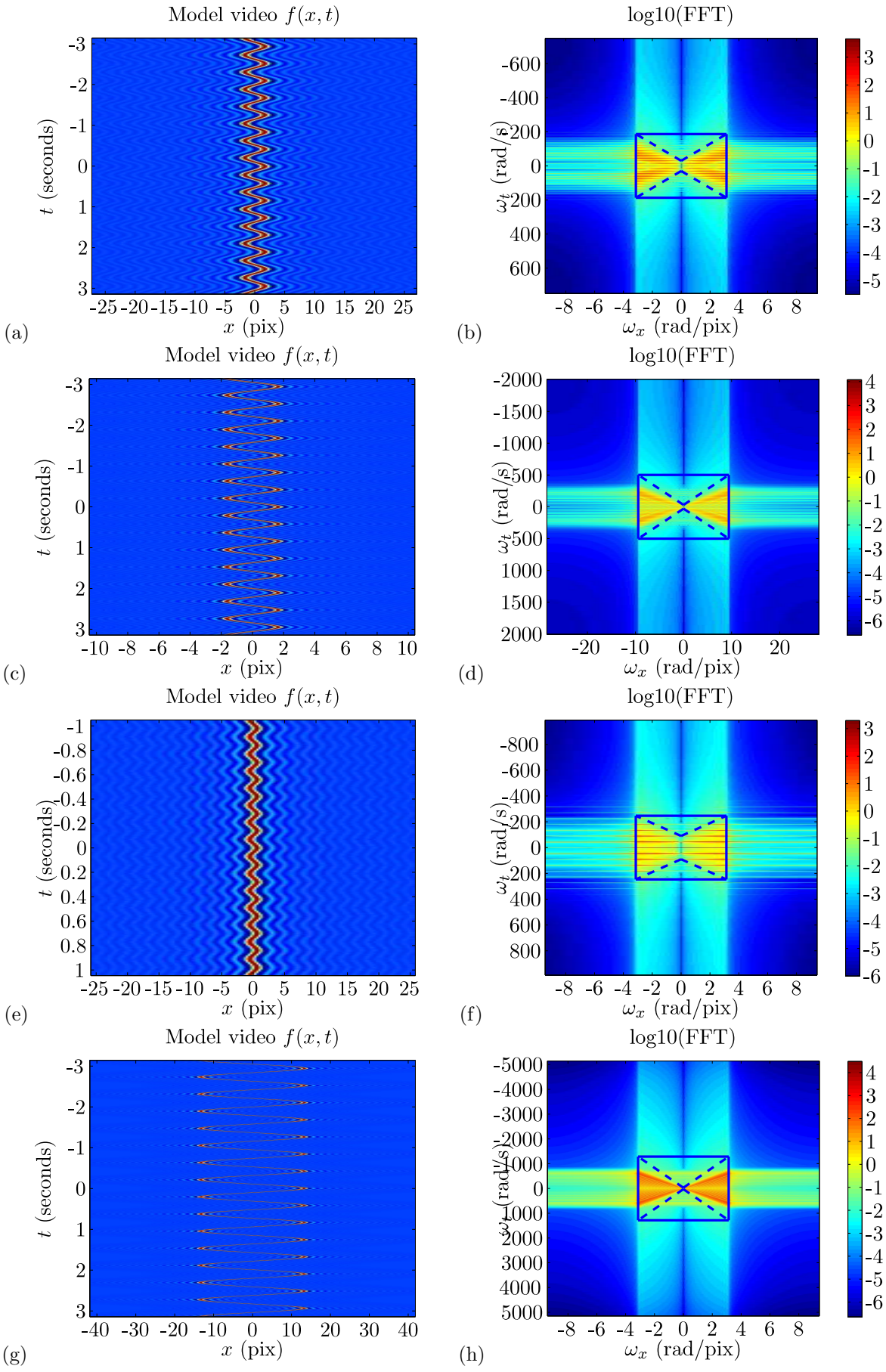


Figure 4: *Left column: Videos $f(x,t) = g(x - h(t))$ with bandlimited profile $g(x)$ and bandlimited sinusoidal model for $h(t)$; see Section 3.1.4. Each row corresponds to one set of parameter values $(\Omega_x, \Omega_h, \Gamma)$ as specified in Table 1. Right column: Estimated spectra. For code, see* `mbwBLtests17.m`.

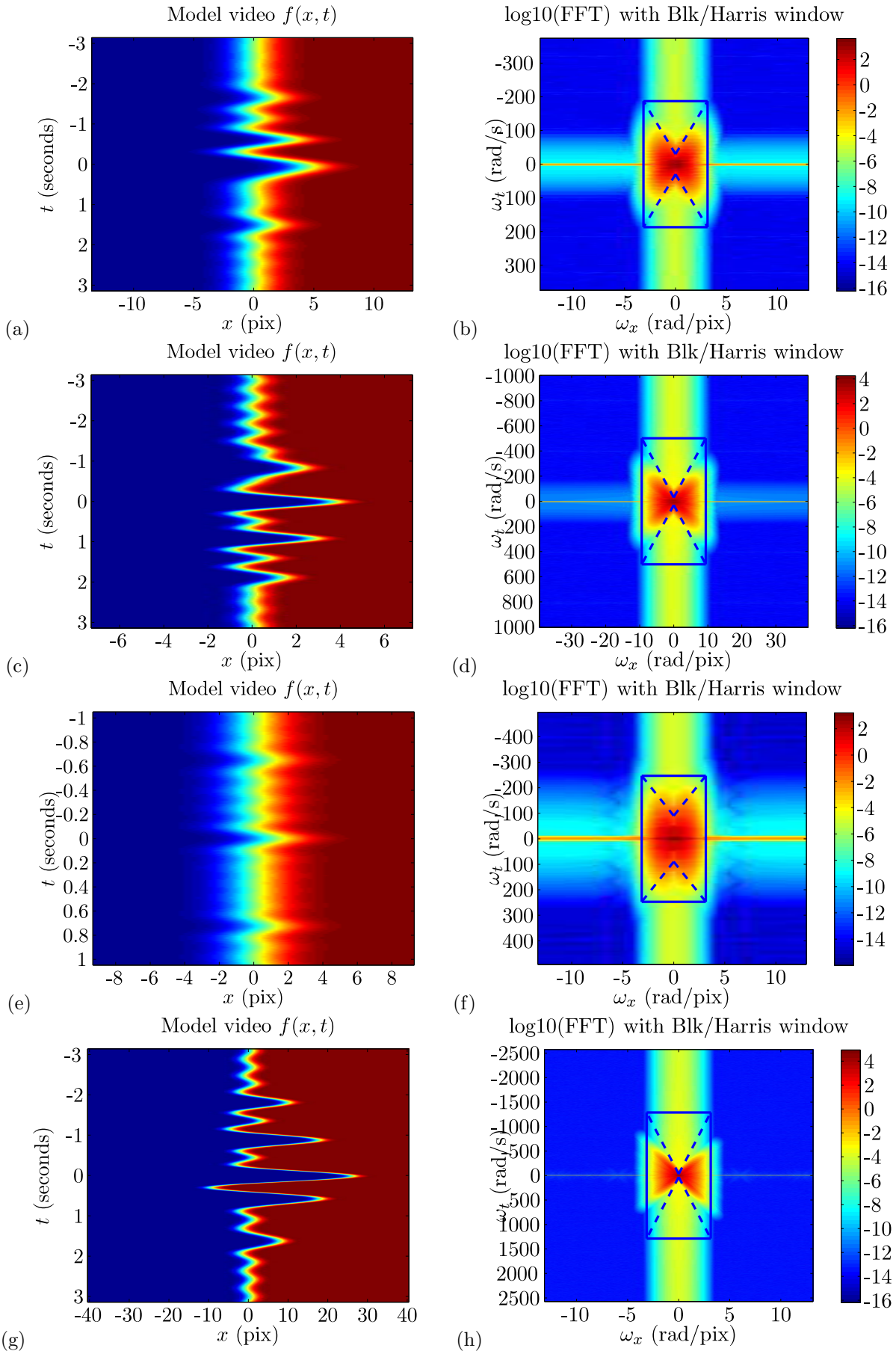


Figure 5: *Left column: Videos $f(x,t) = g(x - h(t))$ with non-bandlimited Gaussian-filtered step function $g(x)$ and bandlimited "sum of sinc functions" model for $h(t)$; see Section 3.1.5. Each row corresponds to one set of parameter values $(\Omega_x, \Omega_h, \Gamma)$ as specified in Table 1. Right column: Estimated spectra with windowing to alleviate border artifacts. For code, see* `mbwBLtests13.m`.

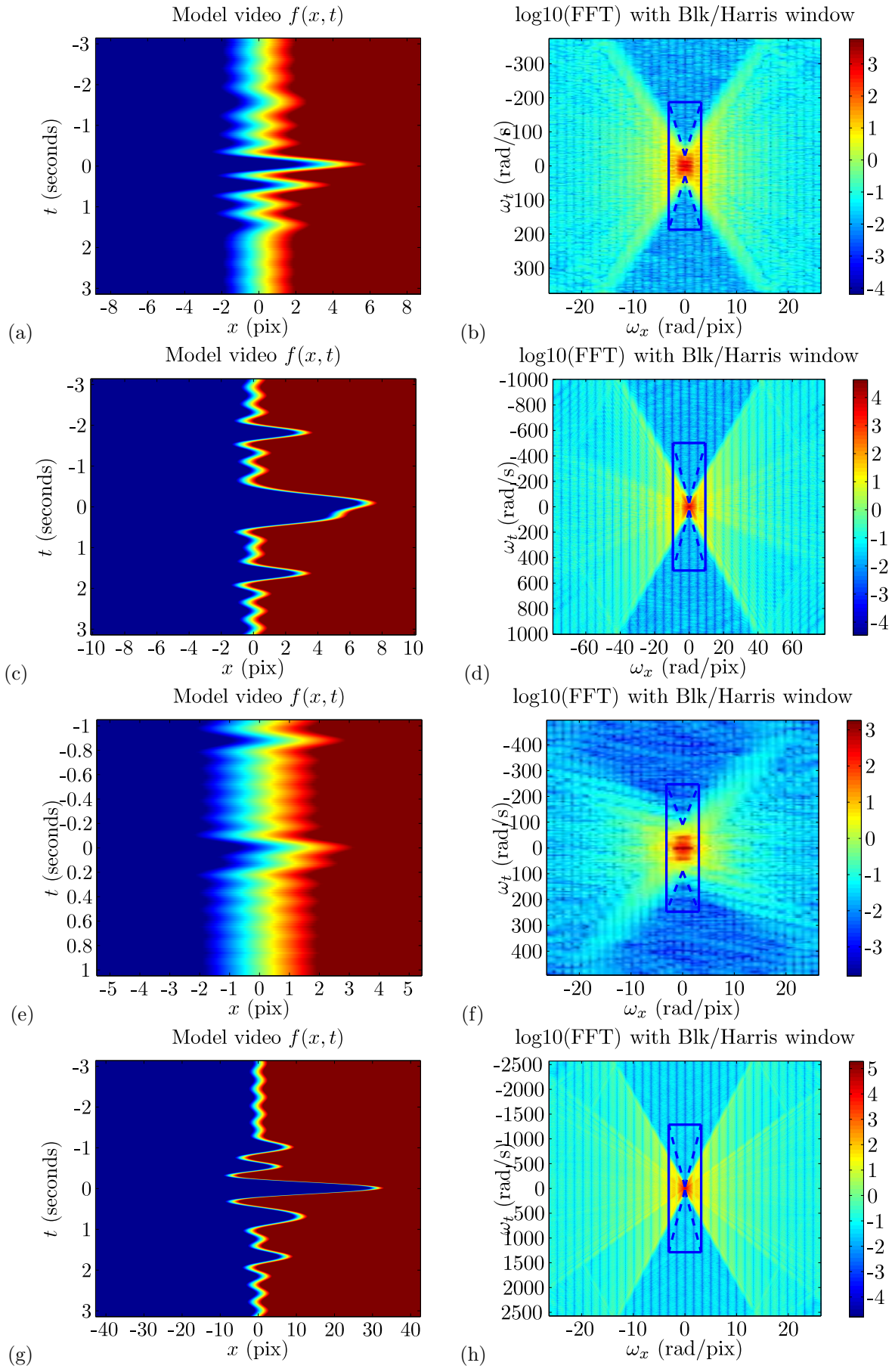


Figure 6: *Left column: Videos $f(x,t) = g(x - h(t))$ with non-bandlimited "triangular step" function $g(x)$ and bandlimited "sum of sinc functions" model for $h(t)$; see Section 3.1.5. Each row corresponds to one set of parameter values $(\Omega_x, \Omega_h, \Gamma)$ as specified in Table 1. Right column: Estimated spectra with windowing to alleviate border artifacts. For code, see* `mbwBLtests18.m`.

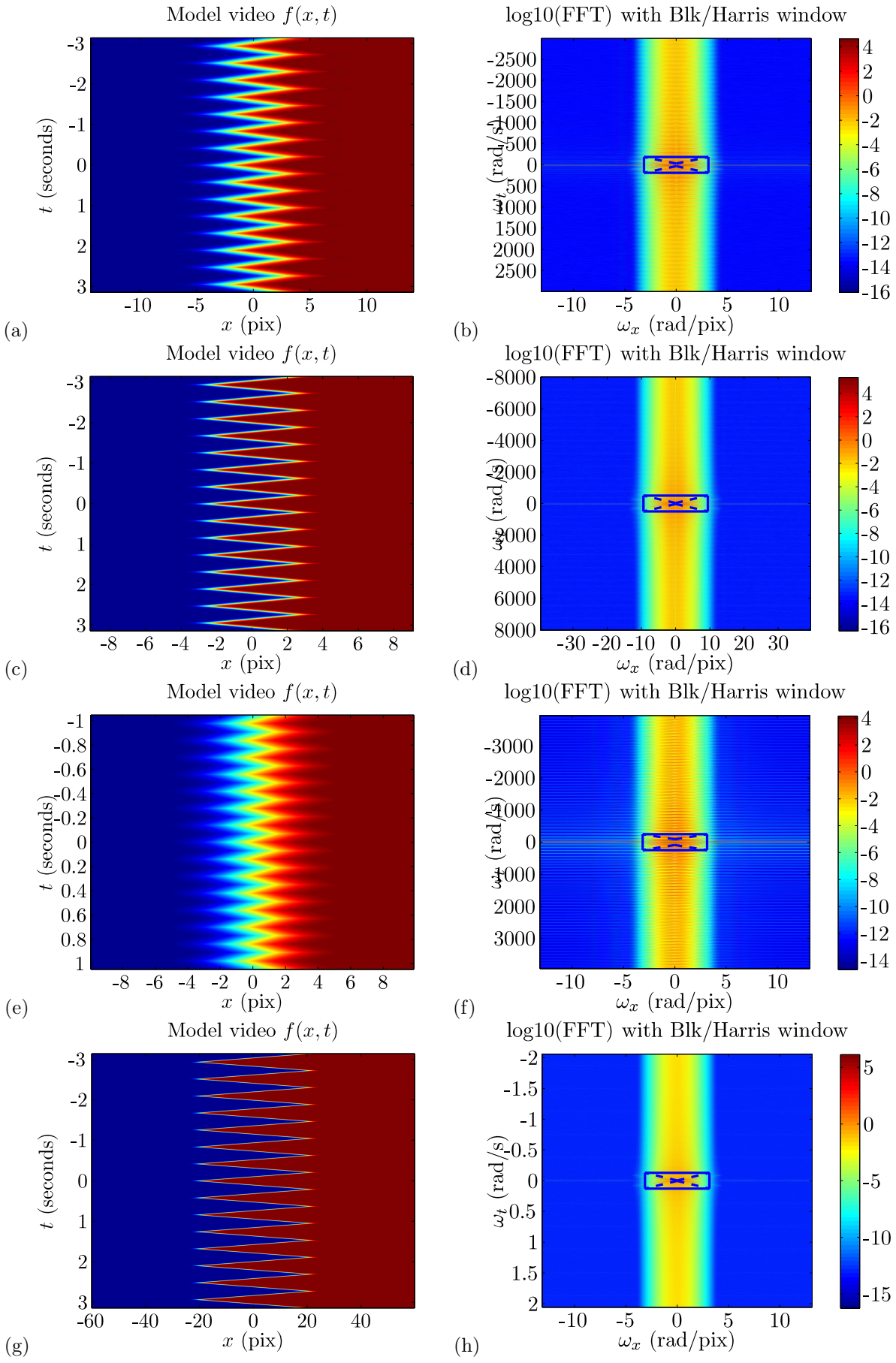


Figure 7: *Left column: Videos $f(x,t) = g(x - h(t))$ with non-bandlimited Gaussian-filtered step function $g(x)$ and non-bandlimited triangle wave model for $h(t)$; see Section 3.1.5. Each row corresponds to one set of parameter values $(\Omega_x, \Omega_h, \Gamma)$ as specified in Table 1. Right column: Estimated spectra with windowing to alleviate border artifacts. For code, see* `mbwBLtests14.m`.

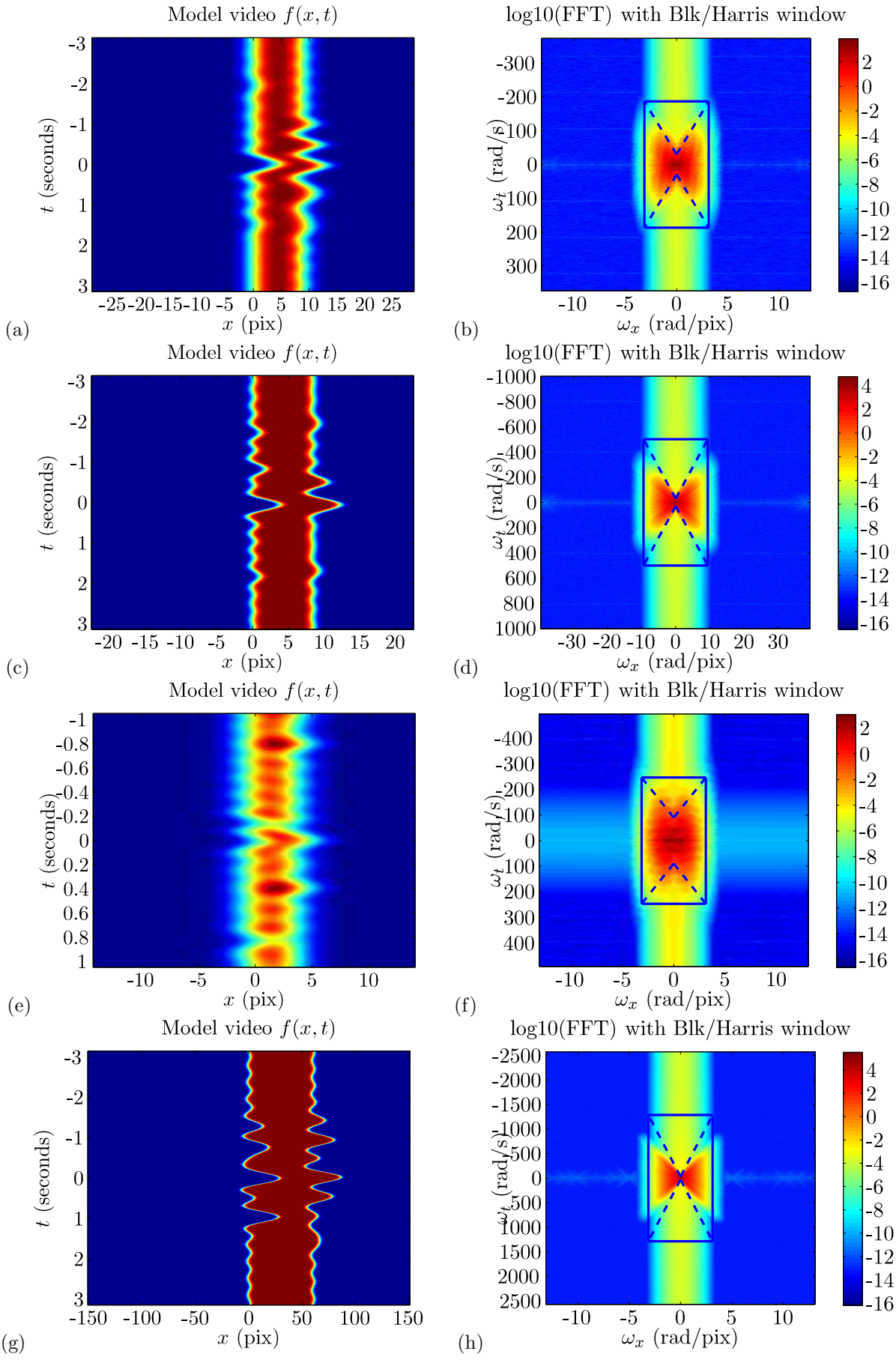


Figure 8: *Left column: Videos $f(x,t)$ containing multiple moving edges with non-bandlimited Gaussian-filtered step functions $g(x)$ and bandlimited "sum of sinc functions" models for $h(t)$; see Section 3.1.5. Each row corresponds to one set of parameter values $(\Omega_x, \Omega_h, \Gamma)$ as specified in Table 1. Right column: Estimated spectra with windowing to alleviate border artifacts. For code, see* `mbwBLtests15.m`.

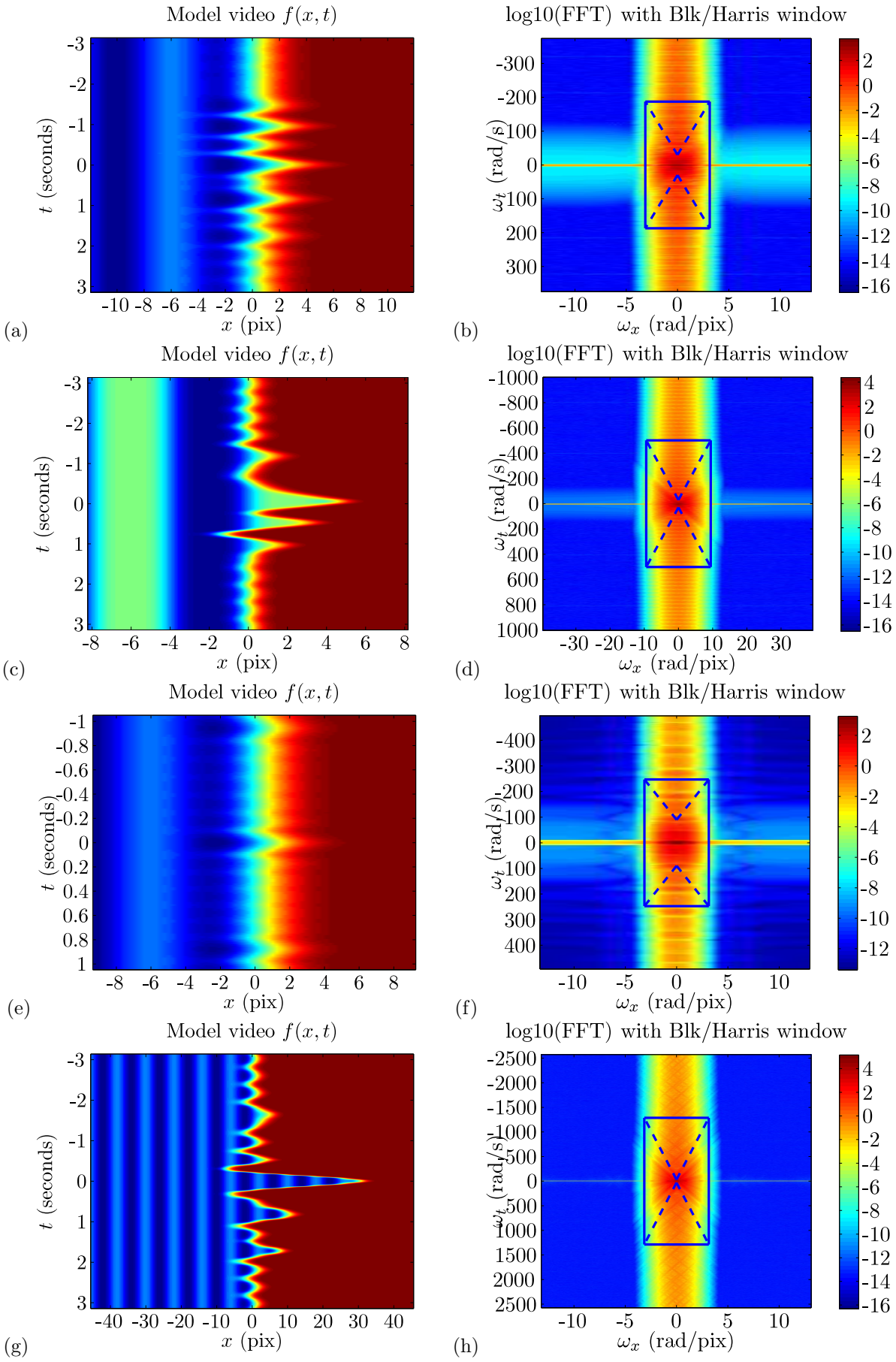


Figure 9: *Left column: Videos $f(x,t)$ with a moving edge occluding a stationary background pattern; see Section 3.1.5. Each row corresponds to one set of parameter values $(\Omega_x, \Omega_t, \Gamma)$ as specified in Table 1. Right column: Estimated spectra with windowing to alleviate border artifacts. For code, see* `mbwBLtests19.m`.

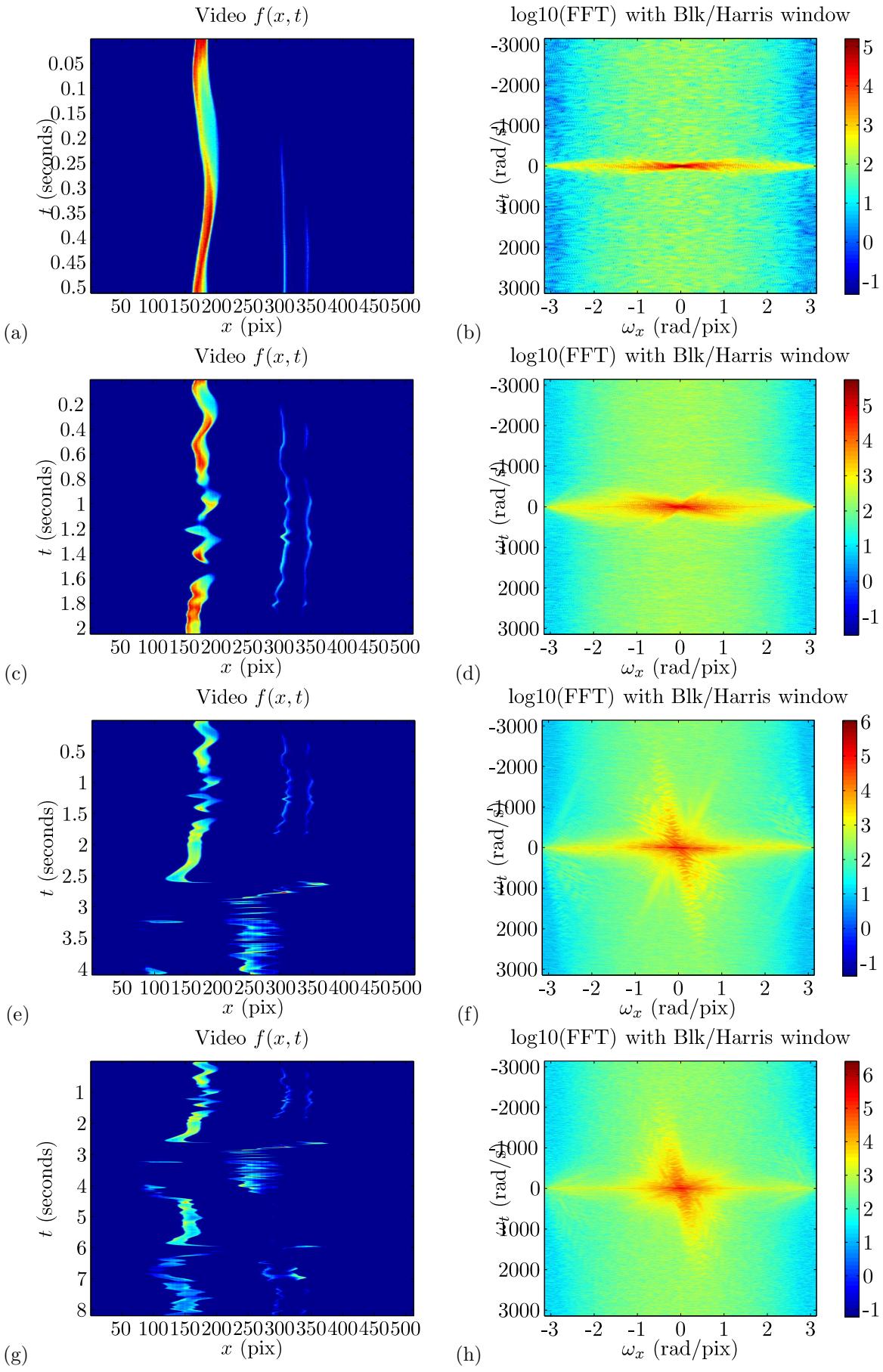


Figure 10: *Left column: Sampled 1D rows from* Candle *video; see Section 3.1.5. From top to bottom, we keep the first 512, 2048, 4096, and 8192 time samples of the video. Right column: Estimated spectra with windowing to alleviate border artifacts. For code, see* `mbwRealVideoTests02.m`.

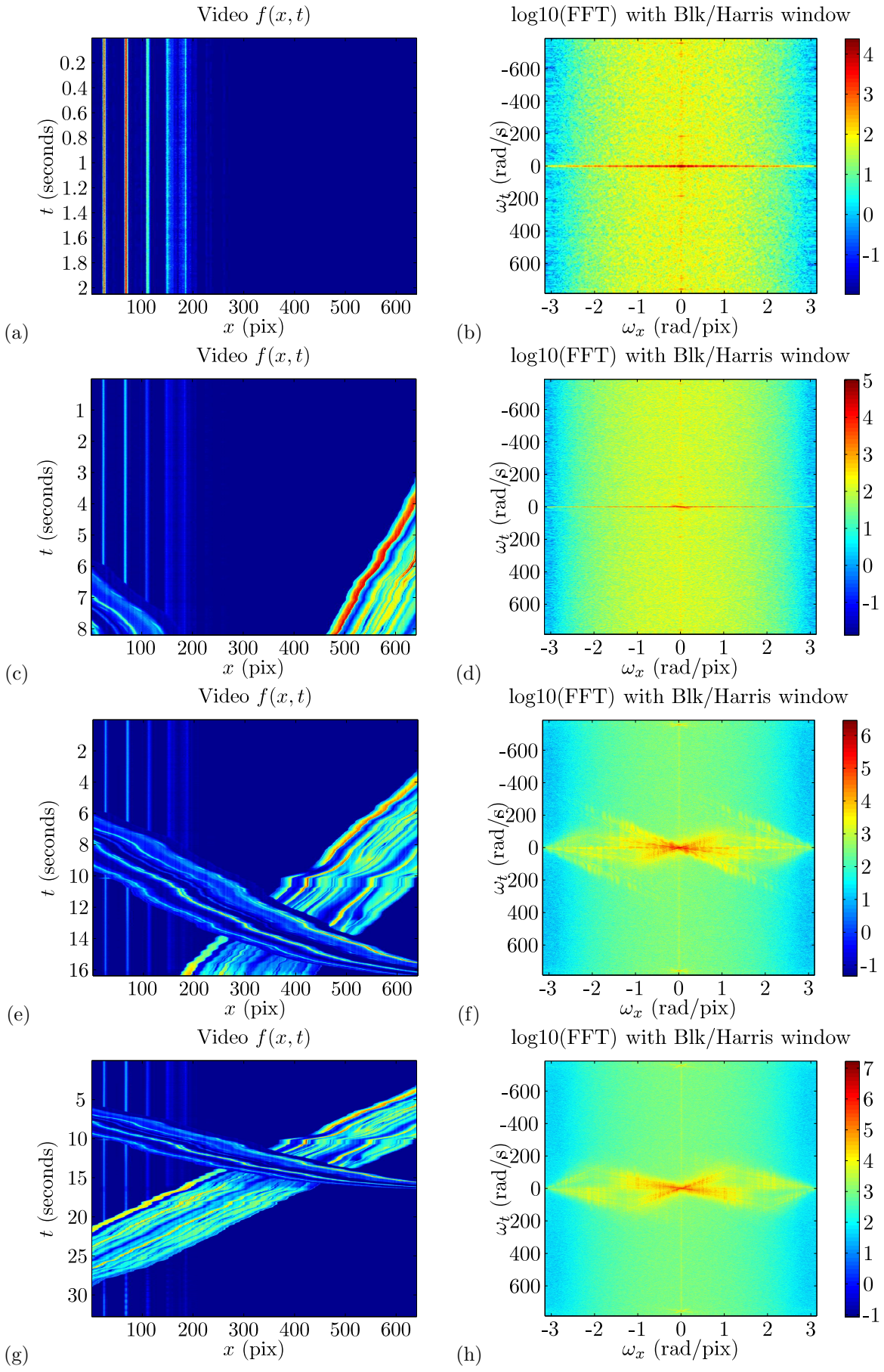


Figure 11: *Left column: Sampled 1D rows from* Pendulum + Cars *video; see Section 3.1.5. From top to bottom, we keep the first 512, 2048, 4096, and 8192 time samples of the video. Right column: Estimated spectra with windowing to alleviate border artifacts. For code, see* `mbwRealVideoTests03.m`.

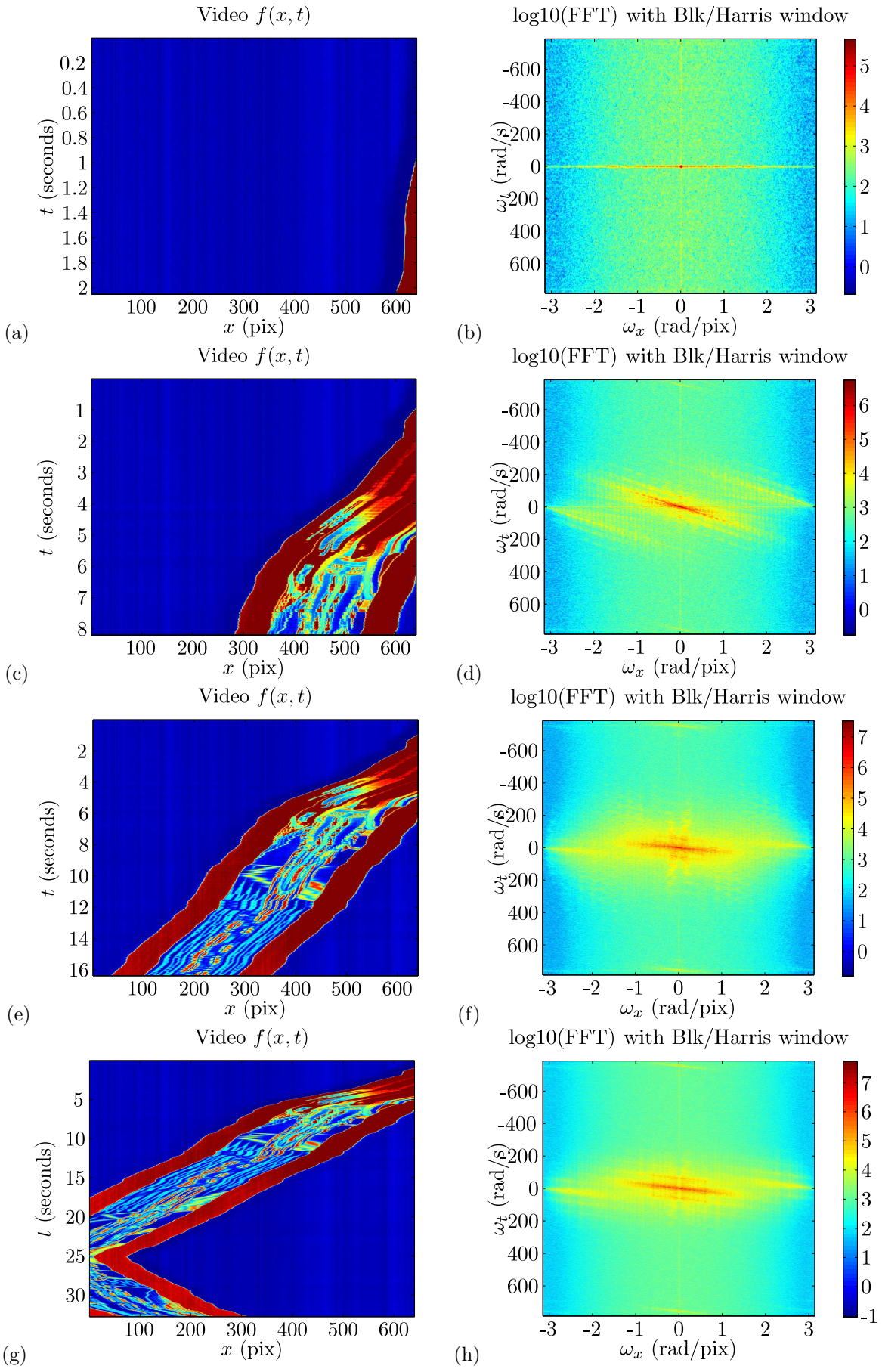


Figure 12: *Left column: Sampled 1D rows from* Card + Monster *video; see Section 3.1.5. From top to bottom, we keep the first 512, 2048, 4096, and 8192 time samples of the video. Right column: Estimated spectra with windowing to alleviate border artifacts. For code, see* `mbwRealVideoTests04.m`.

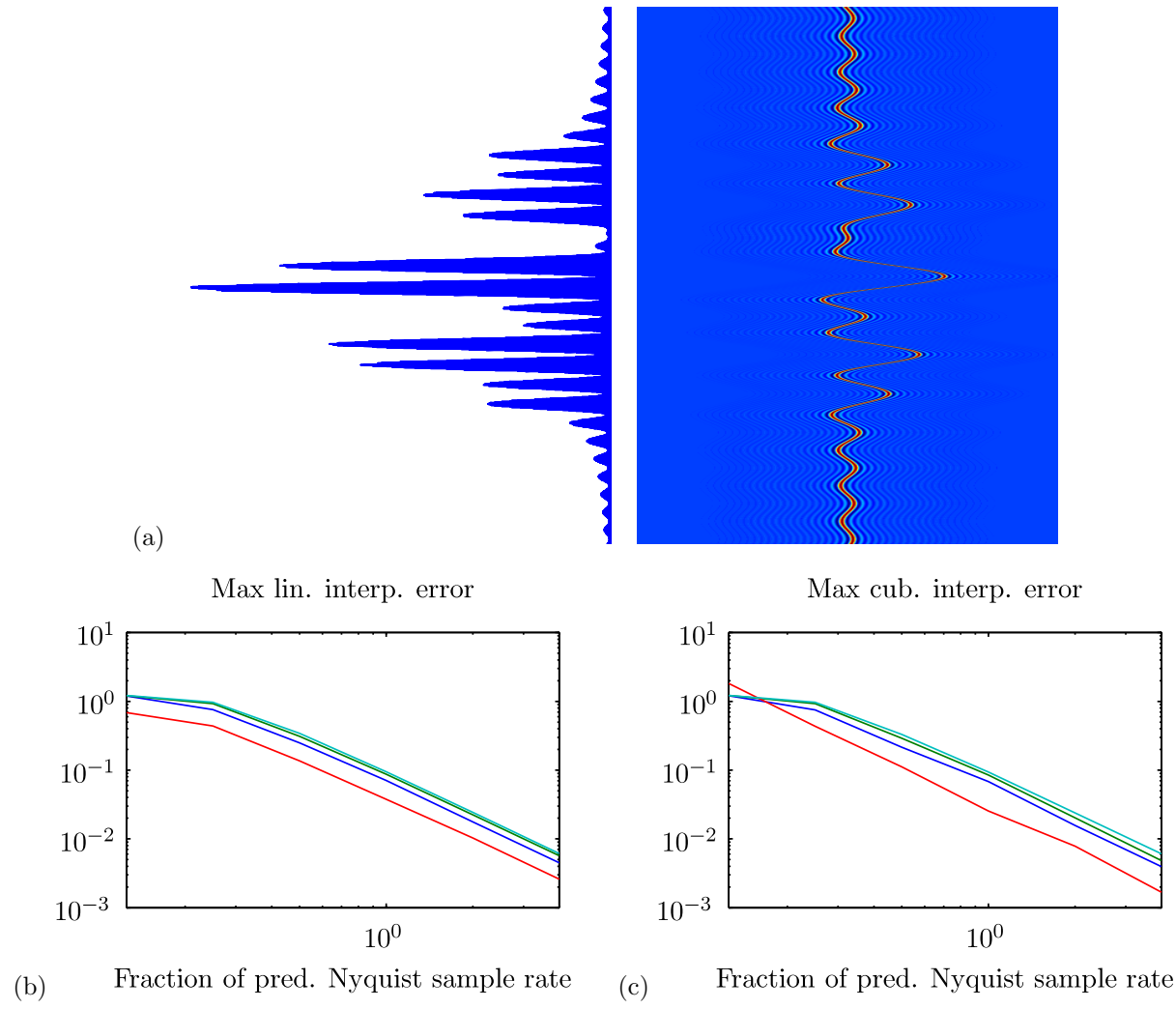


Figure 13: *Interpolation experiments for videos $f(x,t) = g(x - h(t))$ with bandlimited sinc profile $g(x)$ and bandlimited "sum of sinc functions" model for the translation signal $h(t)$; see Section 3.3.2 for details. For code, see* `mbwBLtests12int.m`.

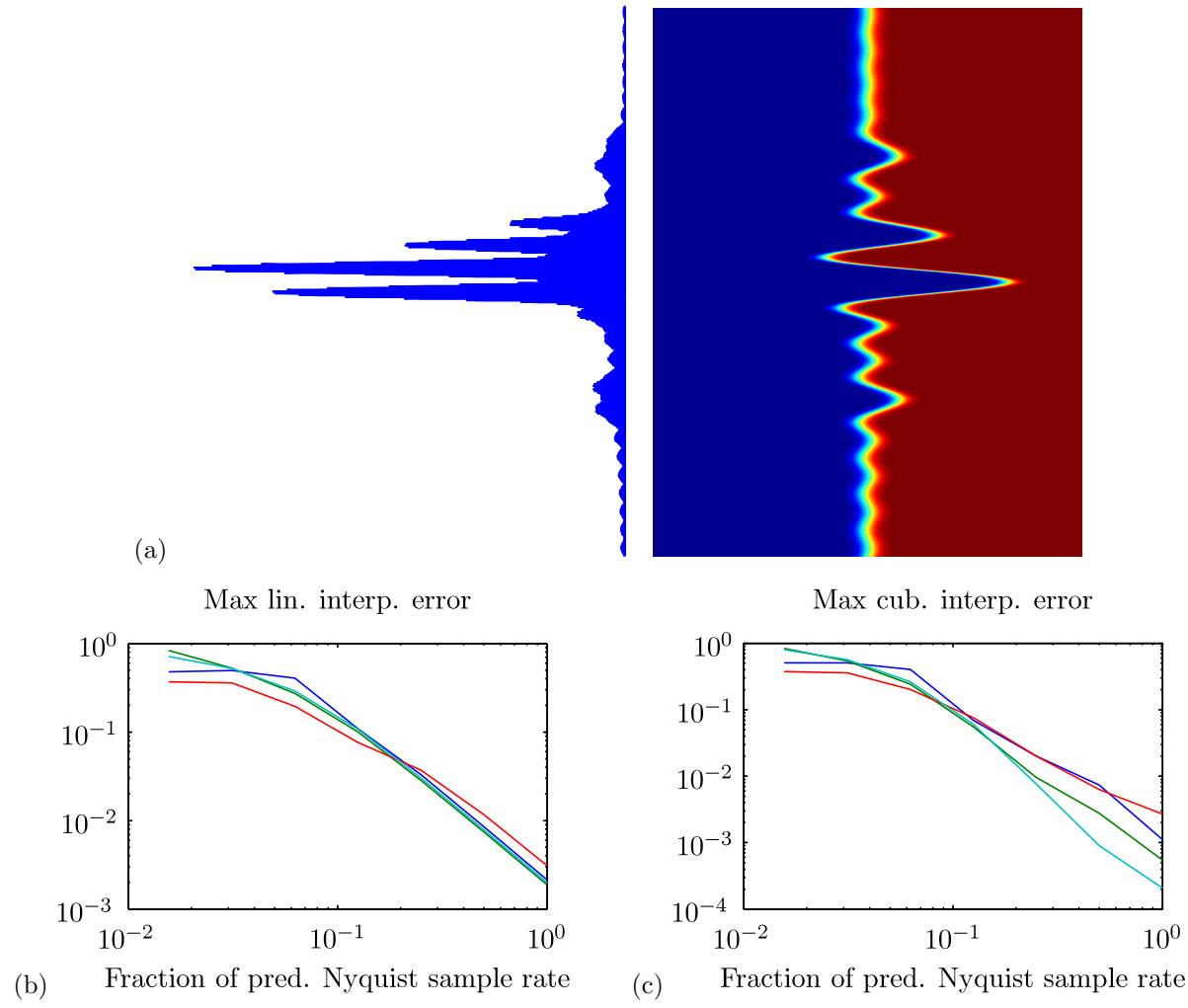


Figure 14: *Interpolation experiments for videos $f(x,t) = g(x - h(t))$ with non-bandlimited Gaussian-filtered step function $g(x)$ and bandlimited "sum of sinc functions" model for $h(t)$; see Section 3.3.3 for details. For code, see* `mbwBLtests13int.m`.

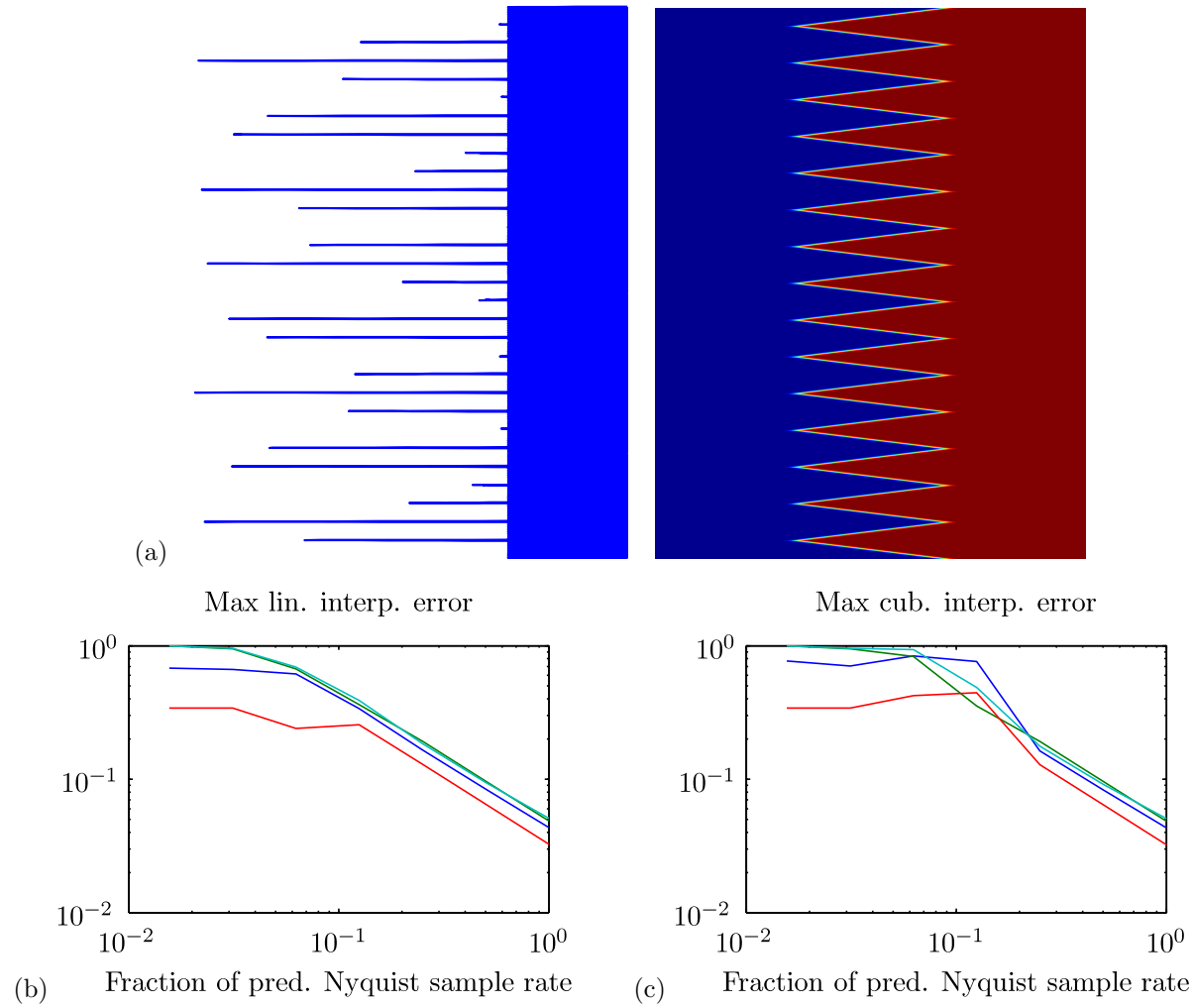


Figure 15: *Interpolation experiments for videos $f(x,t) = g(x - h(t))$ with non-bandlimited Gaussian-filtered step function $g(x)$ and non-bandlimited triangle wave model for $h(t)$; see Section 3.3.3 for details. For code, see* `mbwBLtests14int.m`.

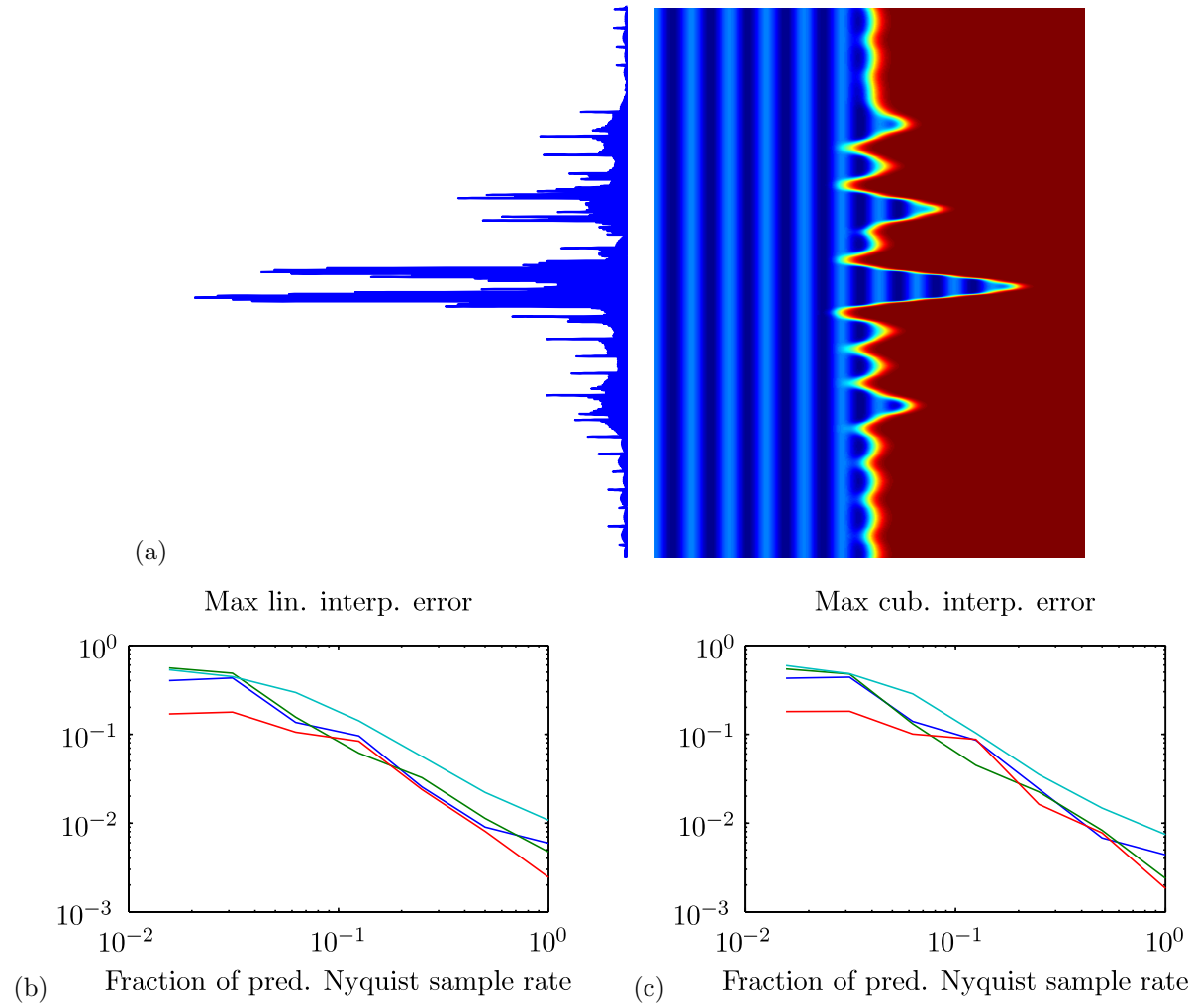
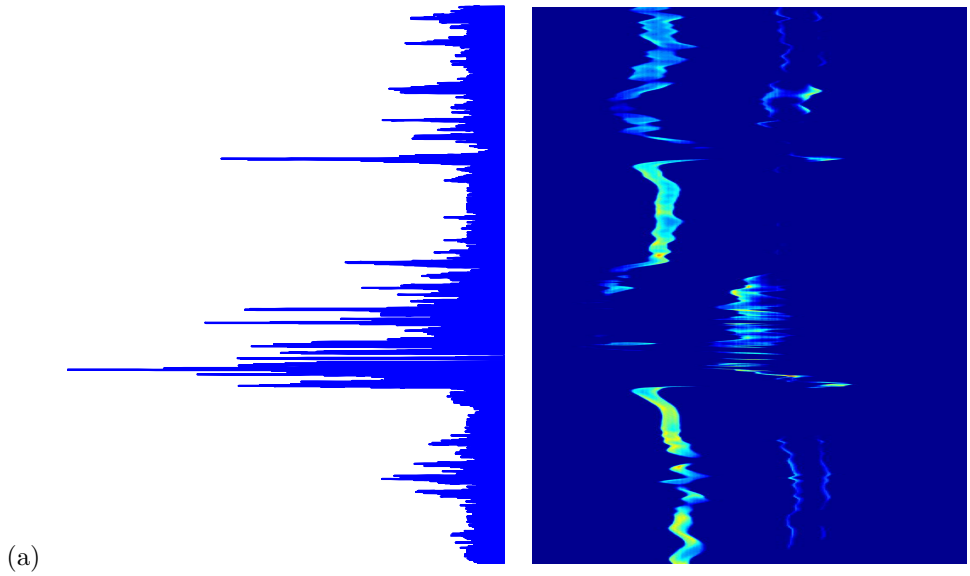
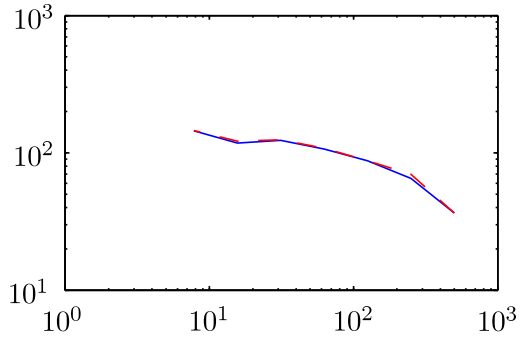


Figure 16: *Interpolation experiments for videos $f(x,t)$ containing a moving edge occluding a stationary background pattern; see Section 3.3.3 for details. For code, see* `mbwBLtests19int.m`.

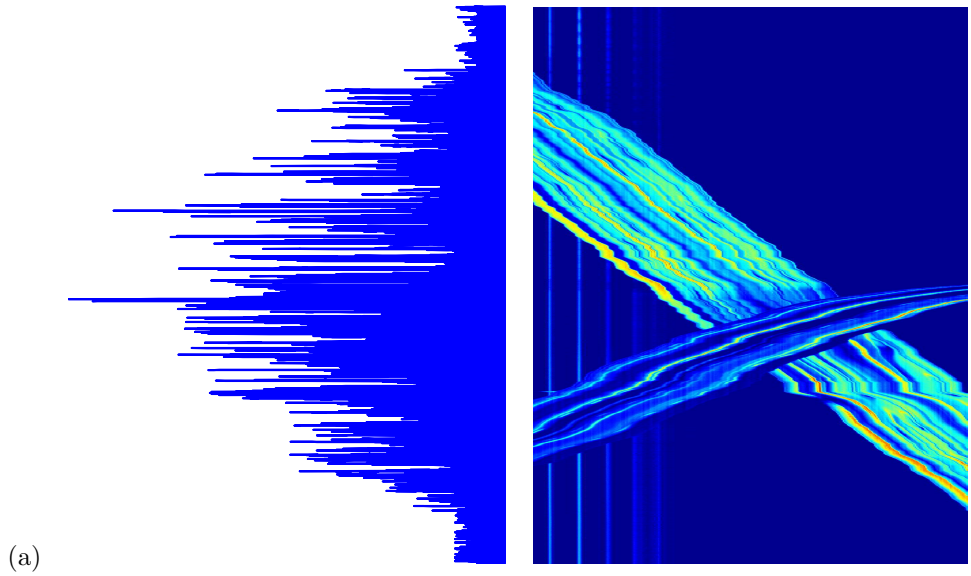


(a)

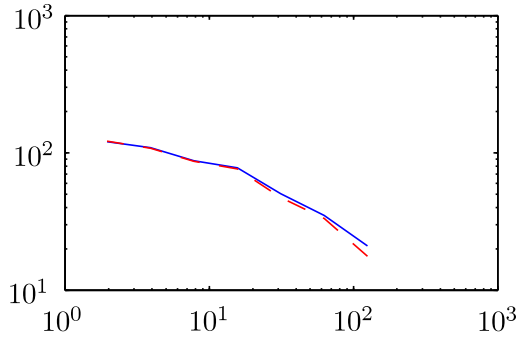


(b)

Figure 17: *Interpolation experiments for* Candle *video; see Section 3.3.3 for details. For code, see* `mbwRealVideoTests02int.m`.

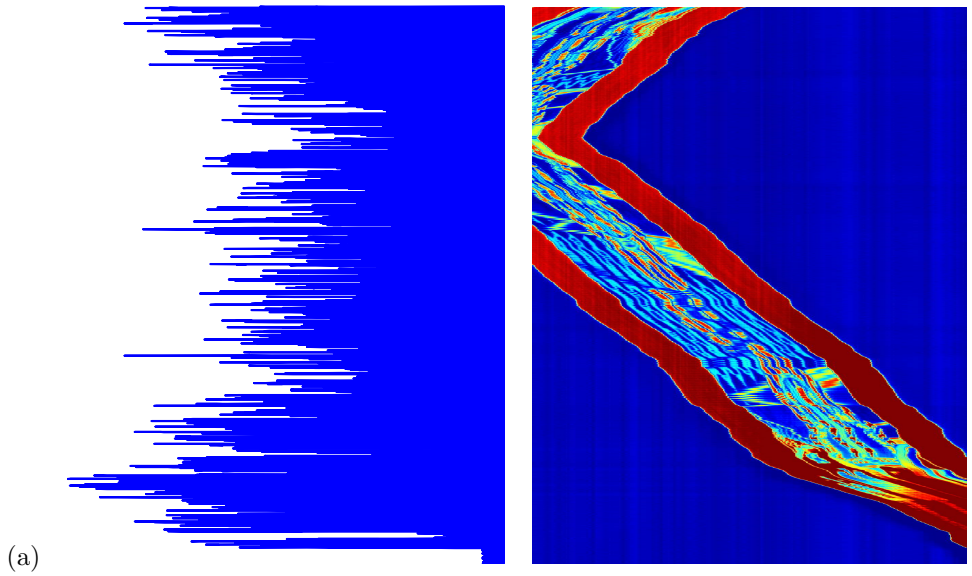


(a)

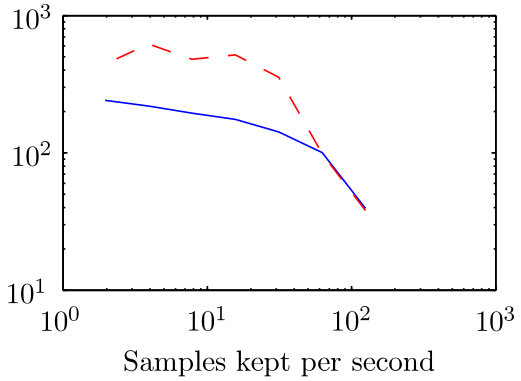


(b)

Figure 18: *Interpolation experiments for* Pendulum + Cars *video; see Section 3.3.3 for details. For code, see* `mbwRealVideoTests03int.m`.



(a)



(b)

Figure 19: *Interpolation experiments for* Card + Monster *video; see Section 3.3.3 for details. For code, see* `mbwRealVideoTests04int.m`.

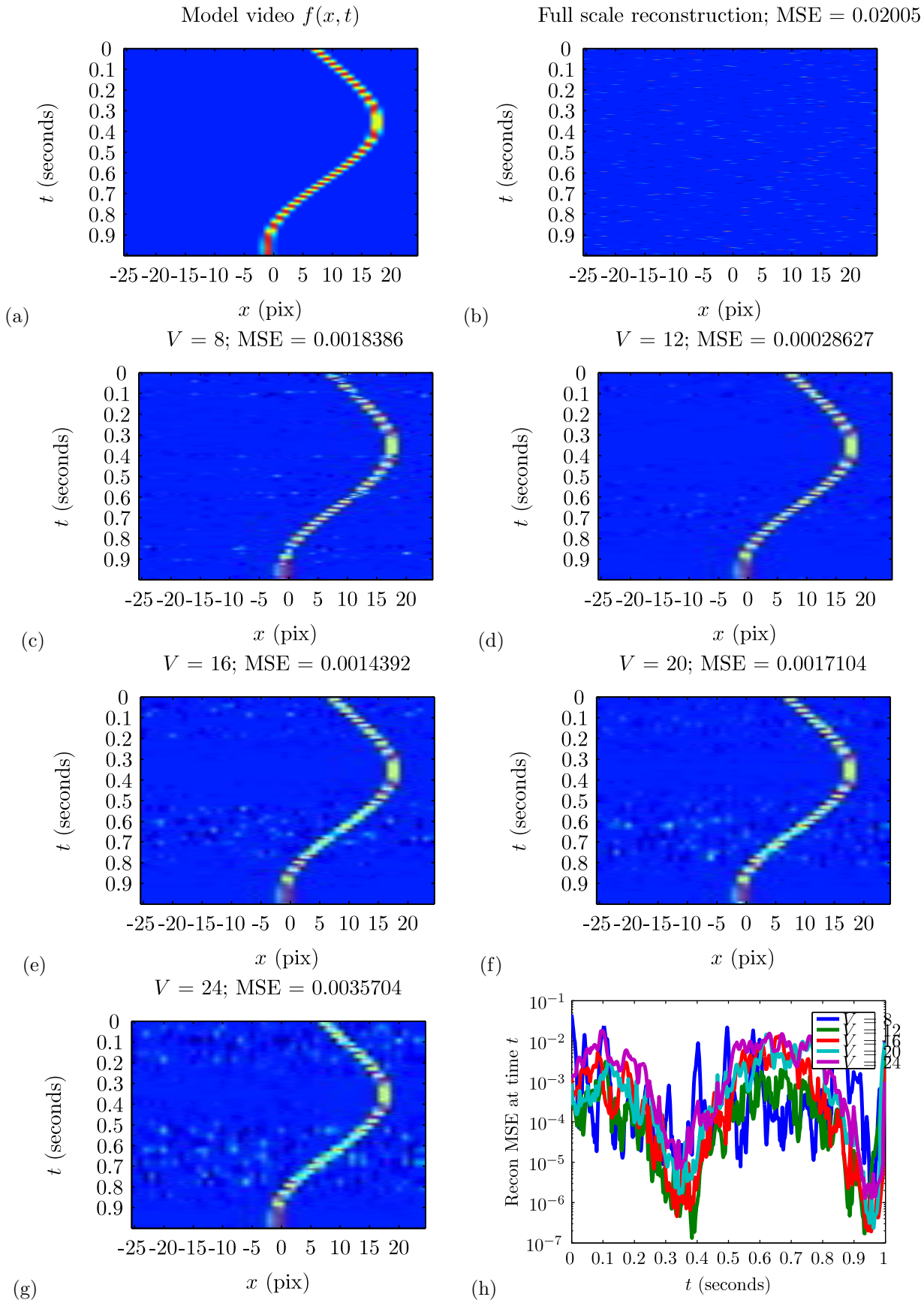


Figure 20: *CS reconstruction experiments for a translational video with a single moving pulse using a linear interpolation kernel; see Section 3.4.3 for details. For code, see* `mbwBLtests21.m`.

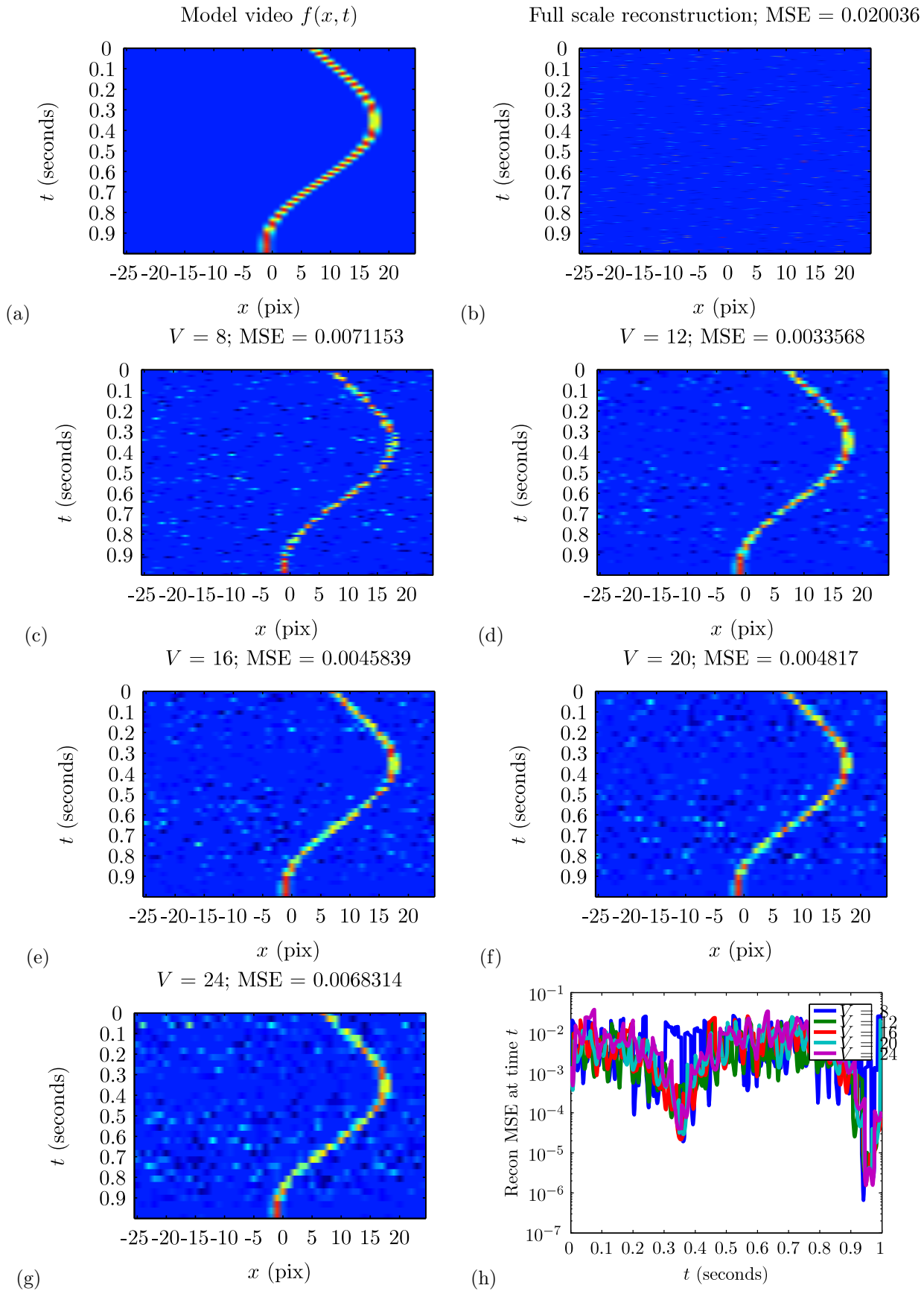


Figure 21: *CS reconstruction experiments for a translational video with a single moving pulse using a nearest neighbor (rectangular) interpolation kernel; see Section 3.4.3 for details. For code, see* `mbwBLtests21.m`.

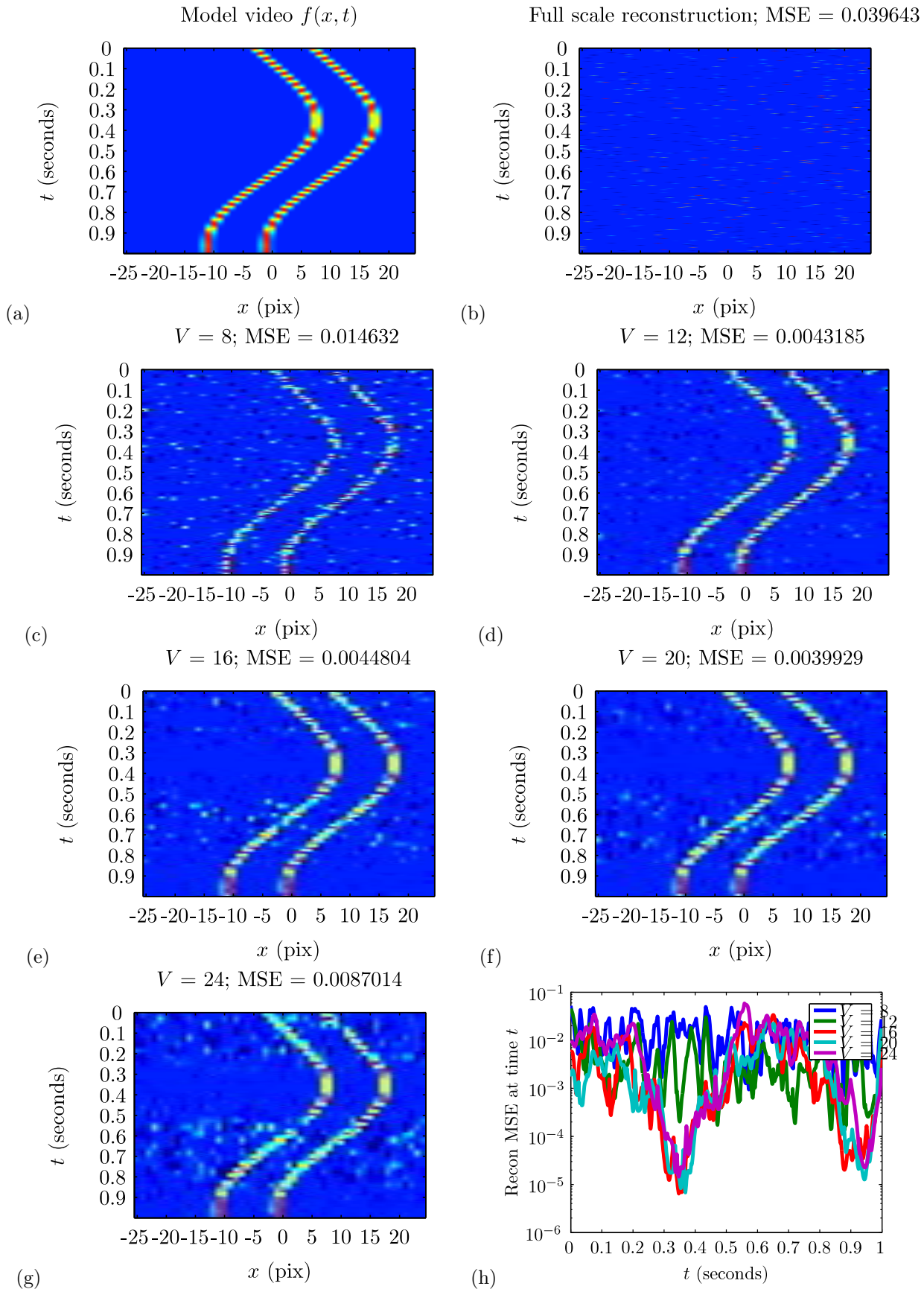


Figure 22: *CS reconstruction experiments for a translational video with a two moving pulses using a linear interpolation kernel; see Section 3.4.3 for details. For code, see* `mbwBLtests21.m`.

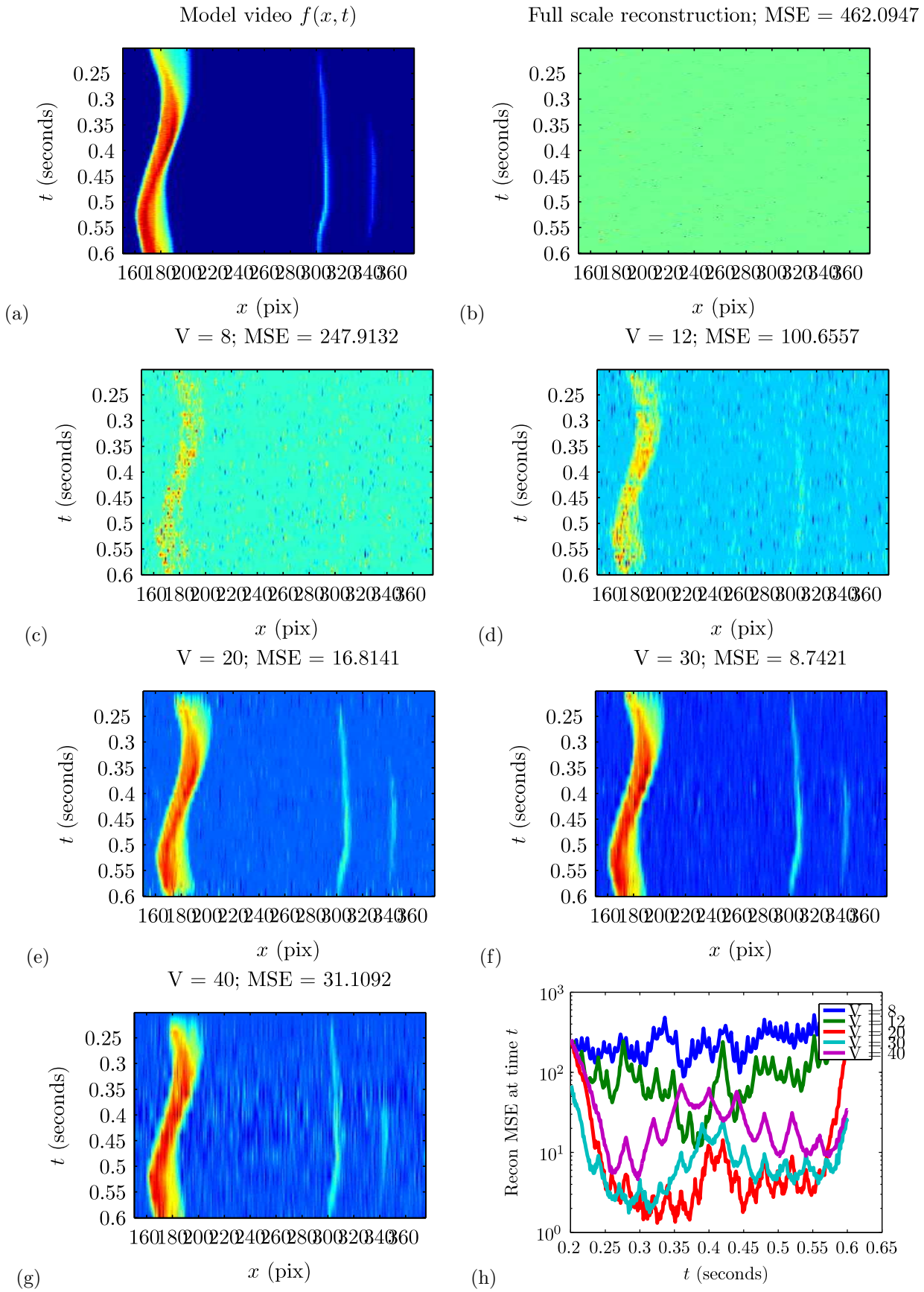


Figure 23: *CS reconstruction experiments for* Candle *video using a linear interpolation kernel; see Section 3.4.3 for details. For code, see* `mbwRealVideoTests02cs.m`.

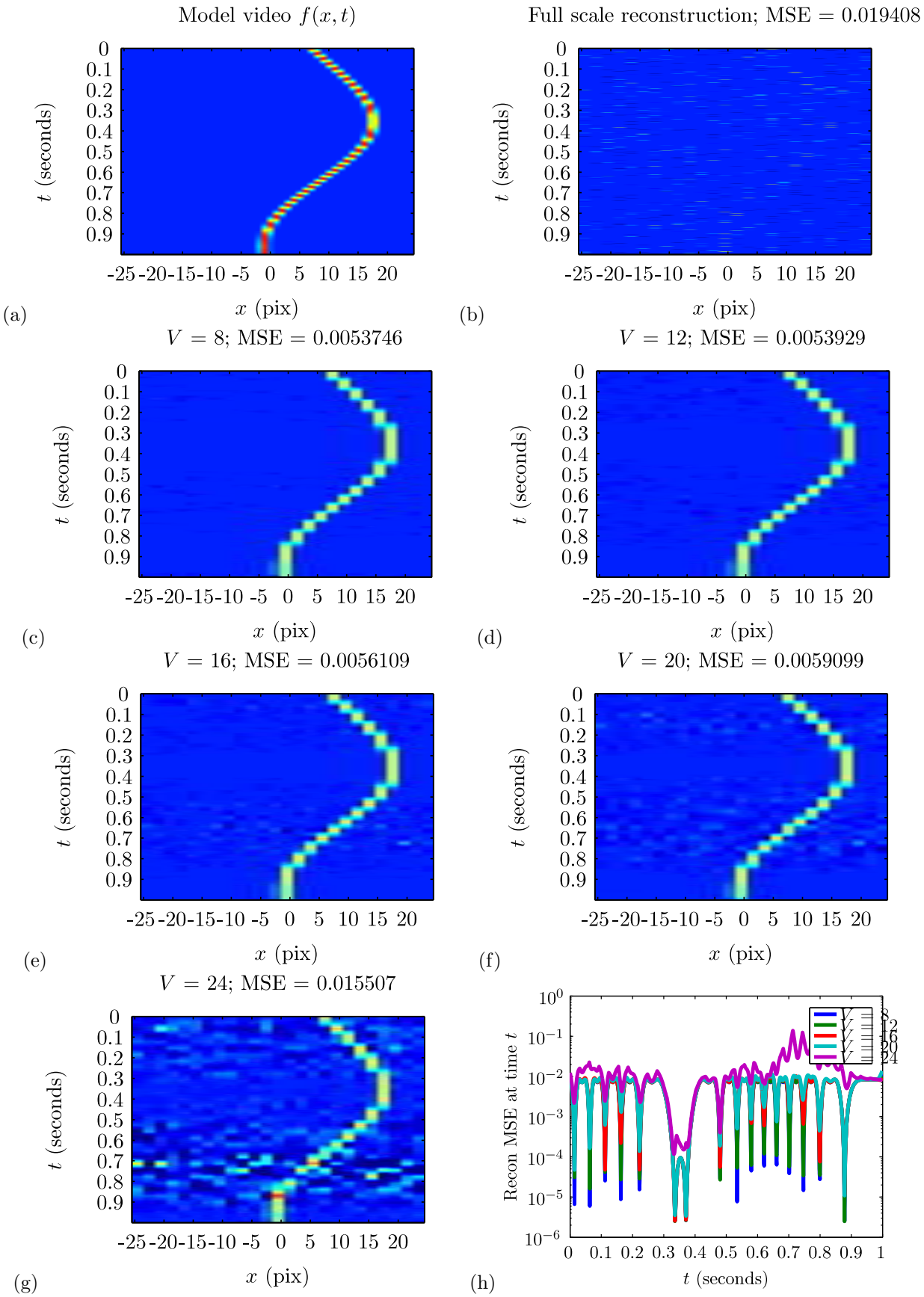


Figure 24: *CS reconstruction experiments for a translational video with a single moving pulse using a linear interpolation kernel, and with lowpass filtered measurement vectors; see Section 3.4.4 for details. For code, see* `mbwBLtests21.m`.

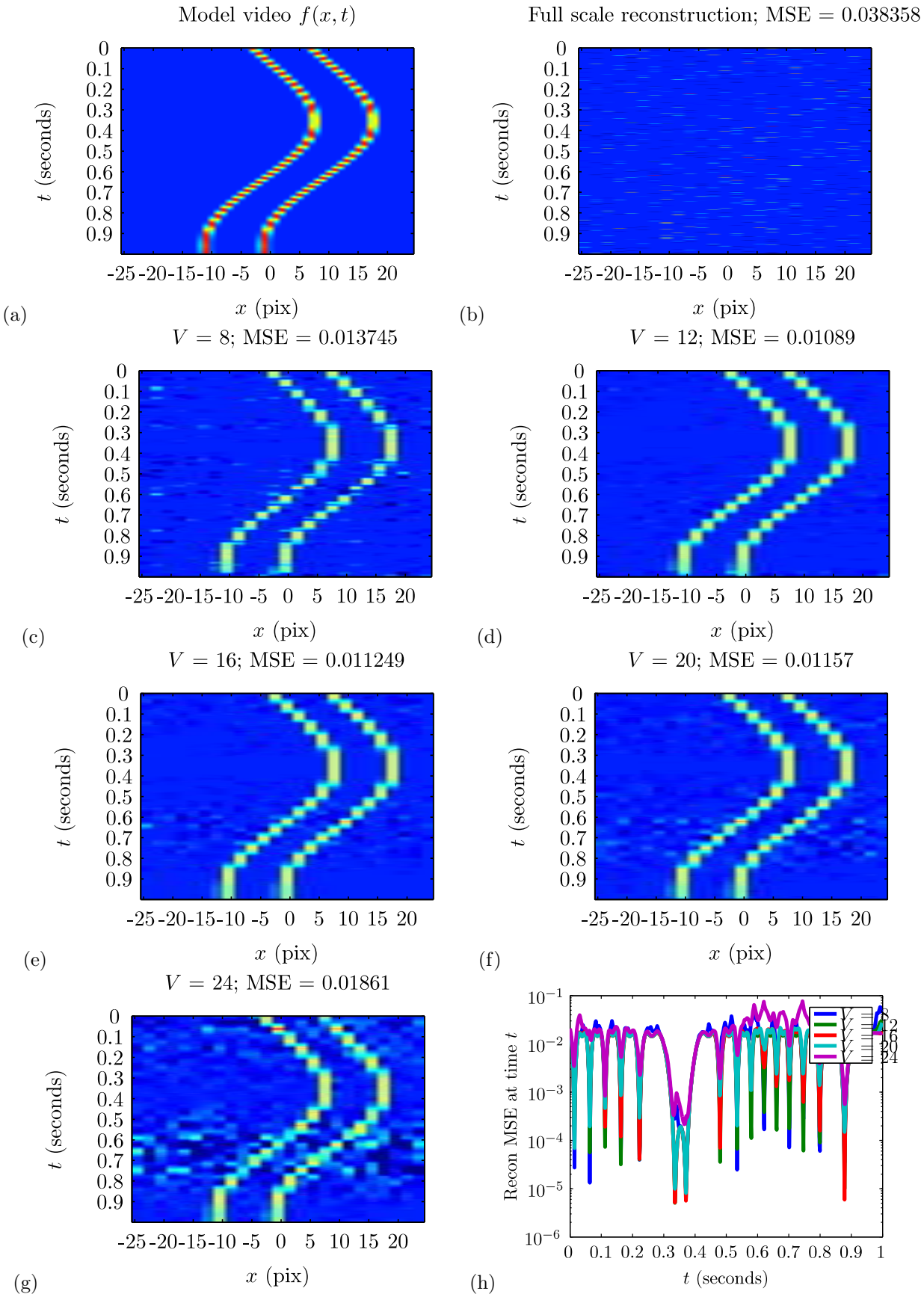


Figure 25: *CS reconstruction experiments for a translational video with two moving pulses using a linear interpolation kernel, and with lowpass filtered measurement vectors; see Section 3.4.4 for details. For code, see* `mbwBLtests21.m`.

4 Compressive acquisition of dynamic scenes

In this section, we detail the new compressive video acquisition framework we developed. *CS-LDS* performs video acquisition using a linear dynamical system (LDS) model coupled with sparse priors for the parameters of the LDS model. The core of the proposed framework is a two-step measurement strategy that enables the recovery of LDS parameters directly from compressive measurements. We solved for the parameters of the LDS using an efficient recovery algorithm that exploits structured sparsity patterns in the observation matrix. Finally, we demonstrated stable recovery of dynamic textures at very low measurement rates.

4.1 Background

4.1.1 Compressive sensing:

Consider a signal $\mathbf{y} \in \mathbb{R}^N$, which is K-*sparse* in an orthonormal basis Ψ; that is, $\mathbf{s} \in \mathbb{R}^N$, defined as $\mathbf{s} = \Psi^T \mathbf{y}$, has at most K non-zero components. Compressive sensing [1,16] deals with the recovery of $\mathbf{y}$ from undersampled linear measurements of the form $\mathbf{z} = \Phi\mathbf{y} = \Phi\Psi\mathbf{s}$, where $\Phi \in \mathbb{R}^{M\times N}$ is the measurement matrix. For $M < N$, estimating $\mathbf{y}$ from the measurements $\mathbf{z}$ is an ill-conditioned problem. Exploiting the sparsity of $\mathbf{s}$, CS states that the signal $\mathbf{y}$ can be recovered exactly from $M = O(K \log(N/K))$ measurements provided the matrix $\Phi\Psi$ satisfies the so-called *restricted isometry property* (RIP) [17].

In practical scenarios with noise, the signal $\mathbf{s}$ (or equivalently, $\mathbf{y}$) can be recovered from $\mathbf{z}$ by solving a convex problem of the form

$$\min \|\mathbf{s}\|_1 \text{ subject to } \|\mathbf{z} - \Phi\Psi\mathbf{s}\| \leq \epsilon \tag{17}$$

with ϵ a bound on the measurement noise. It can be shown that the solution to (17) is with high probability the K-sparse solution that we seek. The theoretical guarantees of CS have been extended to *compressible* signals [18]. In a compressible signal, the sorted coefficients of $\mathbf{s}$ decay rapidly according to a power-law.

There exist a wide range of algorithms that solve (17) under various approximations or reformulations [16, 19]. Greedy techniques such as CoSAMP [20] solve (17) efficiently with strong convergence properties and low computational complexity. It is also easy to impose structural constraints such as block sparsity into CoSAMP giving variants such as model-based CoSAMP [21].

4.1.2 Dynamic textures and linear dynamical systems:

Linear dynamical systems represent a class of parametric models for time-series data. including dynamic textures [22], traffic scenes [23], and human activities [24,25]. Let $\{\mathbf{y}_t, t = 0, \ldots, T\}$ be a sequence of frames indexed by time t. The LDS model parameterizes the evolution of $\mathbf{y}_t$ as follows:

$$\mathbf{y}_t = C\mathbf{x}_t + \mathbf{w}_t \quad \mathbf{w}_t \sim N(\mathbf{0}, R), R \in \mathbb{R}^{N\times N} \tag{18}$$

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + \mathbf{v}_t \quad \mathbf{v}_t \sim N(\mathbf{0}, Q), Q \in \mathbb{R}^{d\times d} \tag{19}$$

where $\mathbf{x}_t \in \mathbb{R}^d$ is the hidden state vector, $A \in \mathbb{R}^{d\times d}$ the transition matrix, and $C \in \mathbb{R}^{N\times d}$ is the observation matrix.

Given the observations $\{\mathbf{y}_t\}$, the truncated SVD of the matrix $[\mathbf{y}]_{1:T} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T]$ can be used to estimate both C and A. In particular, an estimate of the observation matrix C is obtained using the truncated SVD of $[\mathbf{y}]_{1:T}$. Note that the choice of C is unique only up to a $d \times d$ linear transformation. That is, given $[\mathbf{y}]_{1:T}$, we can define $\widehat{C} = UL$, where L is an invertible $d \times d$ matrix. This represents our choice of coordinates in the subspace defined by the columns of C. This lack of uniqueness leads to structured sparsity patterns which can be exploited in the inference algorithms.

4.2 Compressive acquisition of linear dynamical systems

For the rest of this section, we use the following notation. At time t, the image observation (the t-th frame of the video) is $\mathbf{y}_t \in \mathbb{R}^N$ and the hidden state is $\mathbf{x}_t \in \mathbb{R}^d$ such that $\mathbf{y}_t = C\mathbf{x}_t$, where $C \in \mathbb{R}^{N\times d}$ is the observation matrix. We use $\mathbf{z}$ to denote compressive measurements and Φ and Ψ to denote the measurement and sparsifying matrices respectively. We use ":" subscripts to denote sequences, such as $\mathbf{x}_{1:T} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ and $[\cdot]_{1:T}$ to denote matrices, such as $[\mathbf{y}]_{1:T}$ is the $N \times T$ matrix formed by $\mathbf{y}_{1:T}$ such that the k-th column is $\mathbf{y}_k$.

One of the key features of an LDS is that the observations $\mathbf{y}_t$ lie in the subspace spanned by the columns of the matrix C. The subspace spanned by C forms a static parameter of the system. Estimating C and the dynamics encoded in the state sequence $\mathbf{x}_{1:T}$ is sufficient for reconstructing the video. For most LDSs, $N \gg d$, thereby

making C much higher dimensional than the state sequence $\{\mathbf{x}_t\}$. In this sense, the LDS models the video using high information rate static parameters (such as C) and low information rate dynamic components (such as $\mathbf{x}_t$). This relates to our initial motivation for identifying signal models with parameters that are largely static. The subspace spanned by C is static, and hence, we can "pool" measurements over time to recover C.

Further, given that the observations $\mathbf{y}_t$ are compressible in a wavelet/Fourier basis, we can argue that the columns of C need to be compressive as well, either in a similar wavelet basis. This is also motivated by the fact that columns of C encodes the dominant motion in the scene, and for a large set of videos, this is smooth and has sparse representation in a wavelet/DCT basis or in a dictionary learnt from training data. We can exploit this along the lines of the theory of CS. However, note that $\mathbf{y}_t = C\mathbf{x}_t$ is a bilinear relationship in C and $\mathbf{x}_t$ which complicates direct inference of the unknowns. Towards alleviating this non-linearity, we propose a two-step measurement process that allows to estimate the state $\mathbf{x}_t$ first and subsequently solve for a sparse approximation of C. We refer to this as the *CS-LDS* framework.

4.2.1 Outline of the CS-LDS framework

At each time instant t, we take two sets of measurements:

$$\mathbf{z}_t = \begin{pmatrix} \check{\mathbf{z}}_t \\ \widetilde{\mathbf{z}}_t \end{pmatrix} = \begin{bmatrix} \check{\Phi} \\ \widetilde{\Phi}_t \end{bmatrix} \mathbf{y}_t = \Phi_t \mathbf{y}_t, \tag{20}$$

where $\check{\mathbf{z}}_t \in \mathbb{R}^{\check{M}}$ and $\widetilde{\mathbf{z}}_t \in \mathbb{R}^{\widetilde{M}}$, such that the total number of measurements at each frame is $M = \check{M} + \widetilde{M}$. Consecutive measurements from an SPC [4] can be aggregated to provide multiple measurements at each t under the assumption of a quasi-stationary scene. We denote $\check{\mathbf{z}}_t$ as *common* measurements since the corresponding measurement matrix $\check{\Phi}$ is the same at each time instant. We denote $\widetilde{\mathbf{z}}$ as the *innovations* measurements.

The CS-LDS, first, solves for the state sequence $[\mathbf{x}]_{1:T}$ and subsequently, estimates the observation matrix C. The common measurements $[\check{\mathbf{z}}]_{1:T}$ are related to the state sequence $[\mathbf{x}]_{1:T}$ as follows:

$$[\check{\mathbf{z}}]_{1:T} = \begin{bmatrix} \check{\mathbf{z}}_1 & \check{\mathbf{z}}_2 & \cdots & \check{\mathbf{z}}_T \end{bmatrix} = \check{\Phi} C \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_T \end{bmatrix} = \check{\Phi} C [\mathbf{x}]_{1:T}. \tag{21}$$

The SVD of $[\check{\mathbf{z}}]_{1:T} = USV^T$ allows us to identify $[\mathbf{x}]_{1:T}$ up to a linear transformation. In particular, the columns of V corresponding to the top d singular values form an estimate of $[\mathbf{x}]_{1:T}$ up to a $d \times d$ linear transformation (the ambiguity being the choice of coordinate). When the video sequence is exactly an LDS of d dimensions, this estimate is exact provided $\check{M} > d$. The estimate can be very accurate, when the video sequence is approximated by a d-dimensional subspace as discussed later in this section.

Once we have an estimate of the state sequence, say $[\widehat{\mathbf{x}}]_{1:T}$, we can obtain C by solving the following convex problem:

$$(P1)\ \min \sum_{k=1}^{d} \|\Psi^T \mathbf{c}_k\|_1,\ \text{subject to } \|\mathbf{z}_t - \Phi_t C \widehat{\mathbf{x}}_t\|_2 \leq \epsilon, \forall t \tag{22}$$

where $\mathbf{c}_k$ is the k-th column of the matrix C, and Ψ is a sparsifying basis for the columns of C. In Section 4.2.3, we show that the specifics of our measurements induce a structured sparsity in the columns of C, and this naturally leads to an efficient greedy solution.

To summarize (see Figure 26), the design of the measurement matrix as in (20) enables the estimation of the state sequence using just the common measurements, and subsequently solving for C using the diversity present in the innovations measurements $[\widetilde{\mathbf{z}}]_t$.

4.2.2 Random projections of LDS data

As mentioned earlier, when $[\mathbf{y}]_{1:T}$ lies exactly in the (column) span of thematrix C, then $[\check{\mathbf{z}}]_{1:T}$ lies in the span of $\check{\Phi}C$. Hence, the SVD of $[\check{\mathbf{z}}]_{1:T}$ can be used to recover the state sequence up to a linear transformation, provided $\check{M} \geq d$

$$[\check{\mathbf{z}}]_{1:T} = USV^T, \quad [\widehat{\mathbf{x}}]_{1:T} = S_d V_d^T \tag{23}$$

where S_d is the $d \times d$ principal sub-matrix of S and V_d is the $T \times d$ matrix formed by columns of V corresponding to the largest d singular values. In practice, the observations $\mathbf{y}_t$ lie close to the subspace spanned by C such that projection of onto C makes for a highly accurate approximation of $\mathbf{y}_t$. In such a case, the estimate of the state sequence from the SVD of $[\check{\mathbf{z}}]_{1:T}$ is accurate only when the observations $\mathbf{y}_t$ are compressible [26]. In our case, this is equivalent to imposing a power-law decay on the singular values. Figure 27 shows the accuracy of the approximation of the estimated state sequence for various values of $\check{M}$. This suggests that, in practice, $\mathbf{x}_t$ can be reliably estimated with $\check{M} \propto d$.

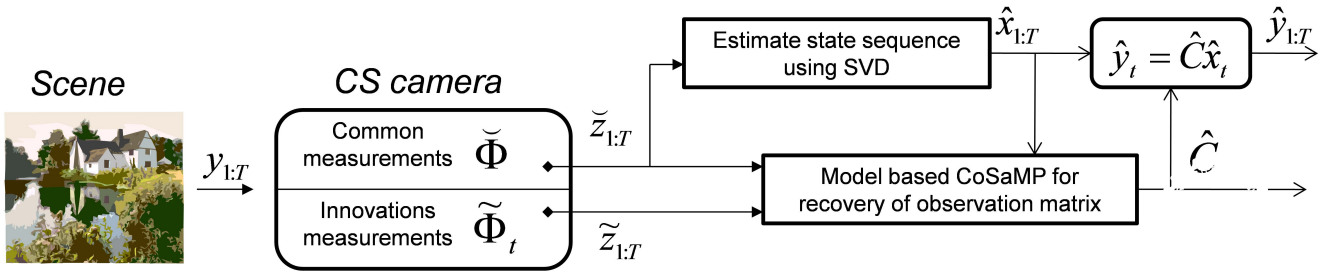


Figure 26: Block diagram of the CS-LDS framework.

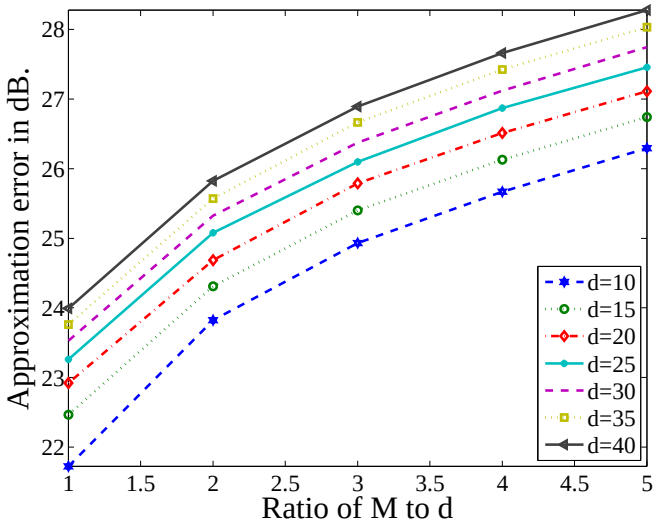


Figure 27: Average error in estimating the state sequence from common measurements for various values of state dimension d and the ratio $\breve{M}/d$. Statistics were computed using 114 videos of 250 frames taken from the DynTex database [27].

4.2.3 Structured sparsity and recovery with modified CoSAMP

The SVD of the common compressive measurements $\breve{\mathbf{z}}_t$ 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. Solving (P1) using this estimate will, at best, lead to an estimate $\widehat{C} = CL$ satisfying $\mathbf{z}_t = \Phi_t \widehat{C} \widehat{\mathbf{x}}_t$. This ambiguity introduces additional concerns in the estimation of C. 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 k-th 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. However, this apparent increase in sparsity is alleviated by the columns having identical supports. The property of identical supports on the columns of CL can be exploited to solve (P1) very efficiently using greedy methods.

Given the state sequence, we use a modified CoSAMP algorithm to estimate C. The modification exploits the structured sparsity induced by the columns of C having identical support. In this regard, the resulting algorithm is a particular instance of the model-based CoSAMP algorithm [21]. One of the key properties of model-based CoSAMP is that stable signal recovery requires only a number of measurements that is proportional to the model-sparsity of the signal, which in our case is equal to dK. Hence, we can recover the observation matrix from $O(dK \log(Nd))$ measurements [21]. Figure 28 summarizes the model-based CoSAMP algorithm used for recovering the observation matrix C.

4.2.4 Performance and measurement rate

For a stable recovery of the observation matrix C, we need in total $O(dK \log(Nd))$ measurements. In addition to this, for recovering the state sequence, we need a number of common measurements proportional to the dimensionality

$\widehat{C}$ = CoSaMP_Model_Sparsity($\Psi, K, \mathbf{z}_t, \widehat{\mathbf{x}}_t, \Phi_t, t = 1, \ldots, T$)

Notation:
supp($vec; K$) returns the support of K largest elements of vec
$A_{|\Omega,\cdot}$ represents the submatrix of A with rows indexed by Ω and all columns.
$A_{|\cdot,\Omega}$ represents the submatrix of A with columns indexed by Ω and all rows.

$\forall t, \Theta_t \leftarrow \Phi_t \Psi$
$\forall t, \mathbf{v}_t \leftarrow \mathbf{0} \in \mathbb{R}^M$
$\Omega_{\text{old}} \leftarrow \phi$
While (stopping conditions are not met)
$R = \sum_t \Theta_t^T \mathbf{v}_t \widehat{\mathbf{x}}_t^T$ $\quad (R \in \mathbb{R}^{N \times d})$
$k \in [1, \ldots, N], \mathbf{r}(k) = \sum_{i=1}^{d} R^2(k, i)$ $\quad (\mathbf{r} \in \mathbb{R}^N)$

$\Omega \leftarrow \Omega_{\text{old}} \bigcup \text{supp}(\mathbf{r}; 2K)$

Find $A \in \mathbb{R}^{|\Omega| \times d}$ that minimizes $\sum_t \|\mathbf{z}_t - (\Theta_t)_{|\cdot,\Omega} A \widehat{\mathbf{x}}_t\|_2$

$B_{|\Omega,\cdot} \leftarrow A$
$B_{|\Omega^c,\cdot} \leftarrow 0$

$k \in [1, \ldots, N], \mathbf{b}(k) = \sum_{i=1}^{d} B^2(k, i)$ $\quad (\mathbf{b} \in \mathbb{R}^N)$
$\Omega \leftarrow \text{supp}(\mathbf{b}; K)$
$S_{|\Omega,\cdot} \leftarrow B_{|\Omega,\cdot}$ $\quad S_{|\Omega^c,\cdot} \leftarrow 0$
$\Omega_{\text{old}} \leftarrow \Omega$
$\widehat{C} \leftarrow \Psi B$

$\forall t, \mathbf{v}_t \leftarrow \mathbf{z}_t - \Theta_t S \widehat{\mathbf{x}}_t$

Figure 28: Pseudo-code of the model-based CoSAMP algorithm for CS-LDS.

of the state vectors

$$MT \quad \propto \quad dK \log(Nd), \qquad \breve{M} \quad \propto \quad d. \tag{24}$$

Compared to Nyquist sampling, we obtain a measurement rate (M/N) given by

$$\frac{M}{N} \quad \propto \quad \frac{dK \log(Nd)}{NT}. \tag{25}$$

This indicates extremely favorable operating scenarios for the CS-LDS framework, especially when T is large (as in a high frame rate capture). Consider a segment of a video of *fixed* duration observed at various sampling rates. The effective number of frames, T, changes with the sampling rate, f_s (in frames per second), as $T \propto f_s$. However, the complexity of the video measured using the state space dimension d does not change. Hence, as the sampling rate f_s increases, $\widetilde{M}$ can be decreased while keeping Mf_s constant. This will ensure that (24) is satisfied, enabling a stable recovery of C. This suggests that as the sampling rate f_s increases our measurement rate decreases, a very desirable property for high-speed imaging.

4.2.5 Extensions

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 minimal modifications to the algorithm described above. First, the state sequence $[\hat{\mathbf{x}}]_{1:T}$ is obtained by performing SVD on the matrix $[\check{z}]_{1:T}$ modified such that the 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, we use model-based CoSAMP to estimate both C and μ simultaneously. However, only the columns of C enjoy the structured sparsity model. The support of μ is not constrained to be similar to that of C.

4.3 Experimental validation

We present a range of experiments validating various aspects of the CS-LDS framework. Our test dataset comprises of videos from DynTex [27] and data we collected using high speed cameras. For most experiments, we chose $\breve{M} = 2d$, with d and K chosen appropriately. We used the mean+LDS model for all the experiments with the 2D DCT as the sparsifying basis for the columns of C as well as the mean. Finally, the entries of the measurement

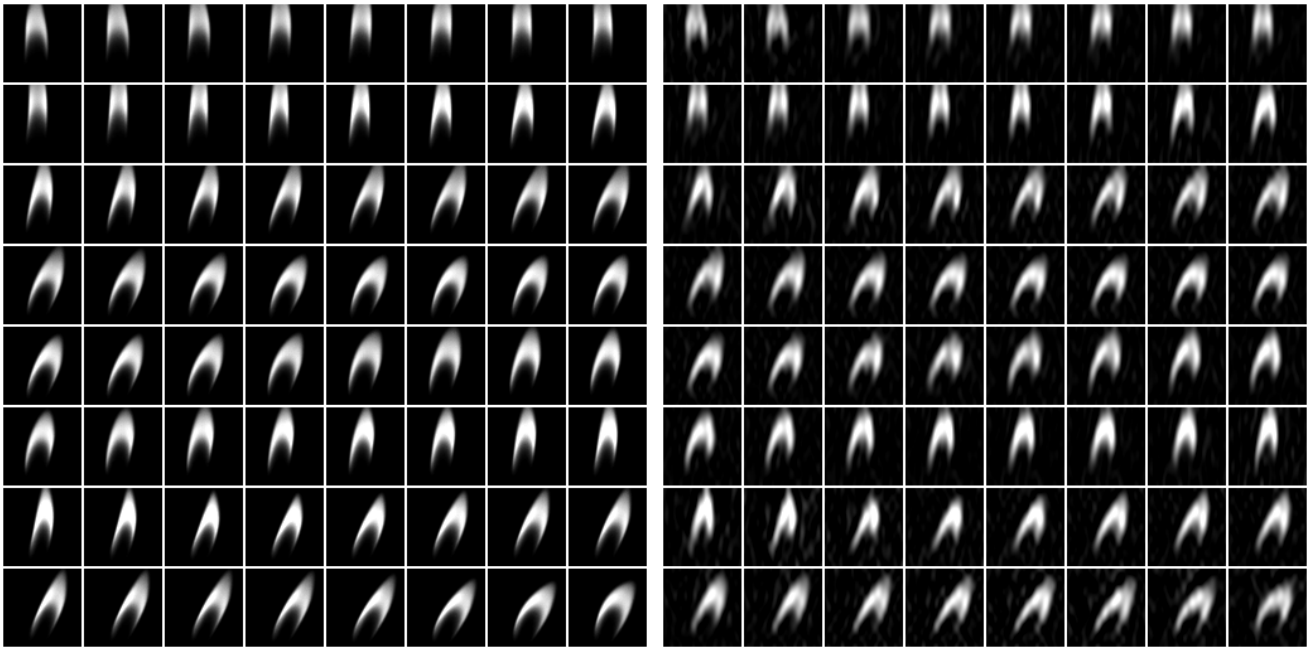


Figure 29: Reconstruction of $T = 1024$ frames of a scene of resolution $N = 64 \times 64$ pixels shown as a mosaic. The original data was collected using a high speed camera operating at 1000 fps. Compressive measurements were obtained with $\check{M} = 30$ and $\widetilde{M} = 20$, thereby giving a measurement rate $M/N = 1.2\%$. Reconstruction was performed using an LDS with $d = 15$ and $K = 150$. Shown above are 64 uniformly sampled frames from the ground truth (left) and the reconstruction (right).

matrix were sampled from iid standard Gaussian distribution. We compare against *frame-by-frame* CS where each frame of the video is recovered separately using conventional CS techniques. We use the term *oracle LDS* for parameters and video reconstruction obtained by operating on the original data itself. The oracle LDS estimates the parameters using a rank-d approximation to the ground truth data. The reconstruction SNR of the oracle LDS gives an upper bound on achievable SNR. Finally, the ambiguity in observation matrix (due to non-uniqueness of the SVD based factorization) as estimated by oracle LDS and CS-LDS is resolved for visual comparison in Figures 30 and 31.

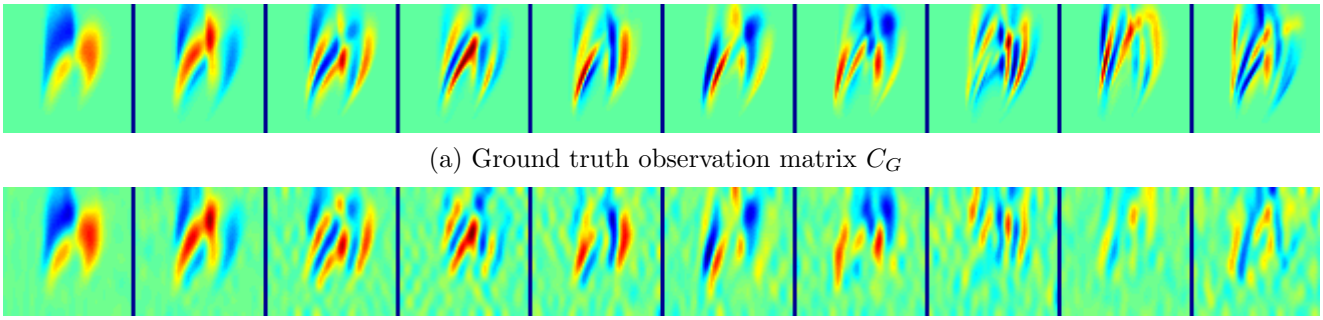
Reconstruction: Figure 29 shows reconstruction results from data collected from a high speed camera of a candle flame. Figure 30 shows the estimated observation matrix as well as the state sequence.

Figure 31 shows video reconstruction of a dynamic texture from the DynTex dataset [27]. Reconstruction results are under a measurement rate $M/N = 1/234$ (about 0.42%), an operating point where a 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 SNR of the reconstructions shown are as follows: Oracle LDS = 24.97 dB, frame-to-frame CS: 11.75 dB and CS-LDS: 22.08 dB.

Performance with measurement noise: It is worth noting that the video sequences used in the experiments have moderate model fit error at a given value of d. The columns of C with larger singular values are, inherently, better conditioned to deal with this model error. The columns corresponding to the smaller singular values are invariably estimated at higher error. This is reflected in the estimates of the C matrix in Figures 30 and 31.

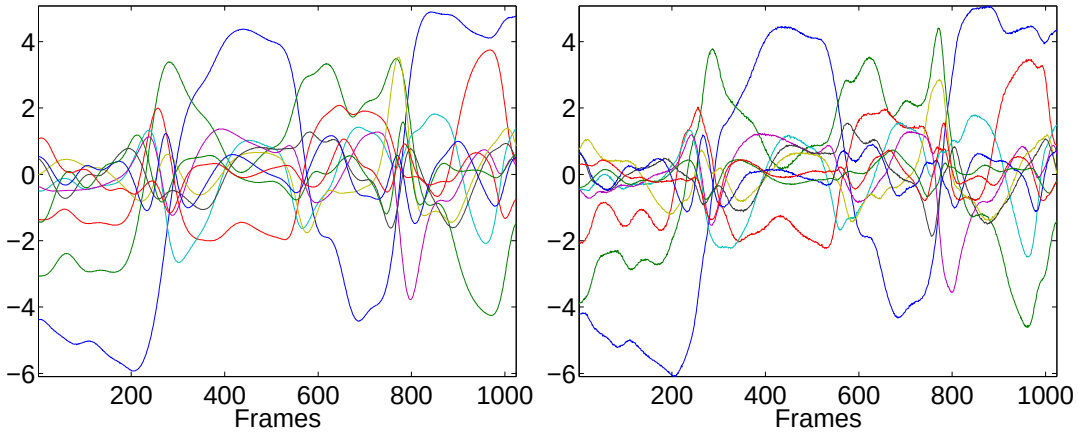
Figure 32 shows the performance of the recovery algorithm for various levels of measurement noise. The effect of the measurement noise on the reconstructions is perceived only at much lower SNR. This is, in part, due to the model fit error dominating the performance of the algorithm when the measurement noise SNR is very high. As the measurement SNR drops significantly below the model fit error, predictably, it starts influencing the reconstructions more. This provides a certain amount of flexibility in the design of potential CS-LDS cameras especially in scenarios where we are not primarily interested in visualization of the sensed video.

Sampling rate: Figure 33 shows reconstruction plots of the candle sequence (of Figure 29) for 1 second of video at various sampling rates. We use (25) to predict the required measurement rates at various sampling rates to maintain a constant reconstruction SNR. As expected, the reconstruction SNR remains the same, while the measurement rate decreases significantly with a linear increase in the sampling rate. This makes the CS-LDS framework extremely promising for high speed capture applications. In contrast, most existing video CS algorithms have measurement rates that, at best, remain constant as the sampling rate increases.



(a) Ground truth observation matrix C_G

(b) Estimated observation matrix $\widehat{C}L \quad (L = \widehat{C}^{\dagger} C_G)$



(c) Ground truth state sequence (d) Estimated state sequence

Figure 30: Ground truth and estimated parameters corresponding to Figure 29. Shown are the top 10 columns of the observation matrix and state sequences. Matlab's "jet" colormap (red= +large and blue= −large) is used in (a) and (b).

Table 2: Classification results (in %) on the traffic databases for two different values of state space dimension d. Results are over a database of 254 videos, each of length 50 frames at a resolution of 64×64 pixels under a measurement rate of 4%.

(a) $d = 10$

	Expt 1	Expt 2	Expt 3	Expt 4
Oracle LDS	85.71	85.93	87.5	92.06
CS-LDS	84.12	87.5	89.06	85.71

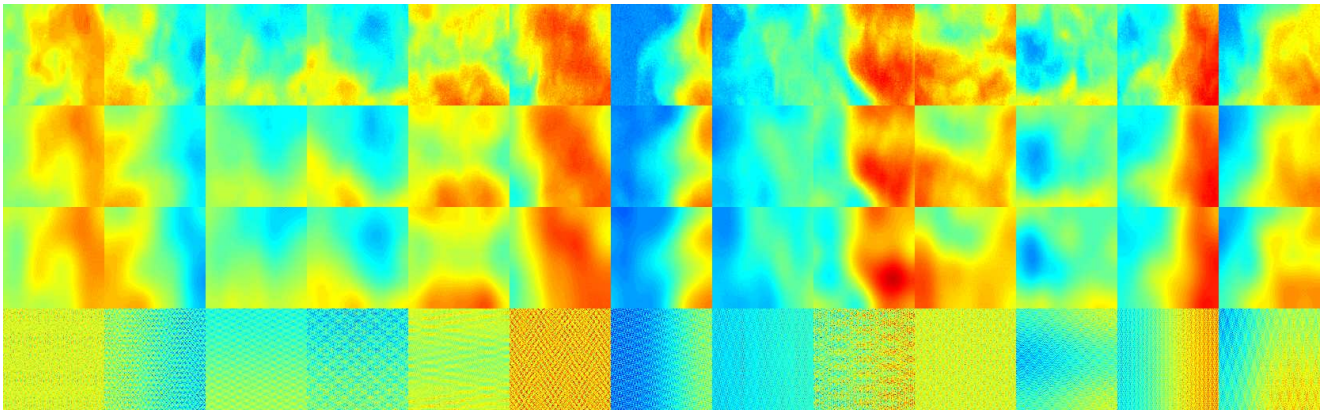
(b) $d = 5$

	Expt 1	Expt 2	Expt 3	Expt 4
Oracle LDS	77.77	82.81	92.18	80.95
CS-LDS	85.71	73.43	78.1	76.1

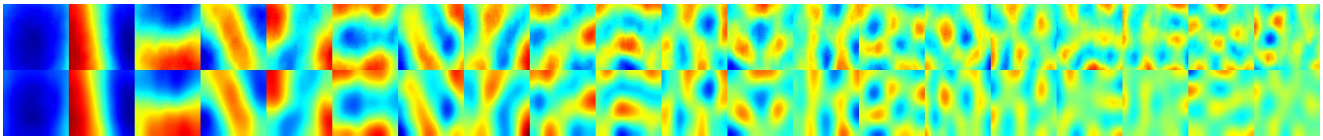
Application to scene classification: In this experiment, we study feasibility of classification problems on the videos sensed and reconstructed under the CS-LDS framework. We consider the UCSD traffic database used in [23]. The dataset consists of 254 videos of length 50 frames capturing traffic of three types: light, moderate, heavy. Figure 34 shows reconstruction results on a traffic sequence from the dataset. We performed a classification experiment of the videos into these three categories. There are 4 different train-test scenarios provided with the dataset. Classification is performed using the subspace-angles based metric with a nearest-neighbor classifier on the LDS parameters [28]. The experiment was performed using the parameters estimated directly without reconstructing the frames. For comparison, we also perform the same experiments with fitting the LDS model on the original frames (oracle LDS). Table 2 shows classification results. We see that we obtain comparable classification performance using the proposed CS-LDS recovery algorithm to the oracle LDS. This suggests that the CS-LDS camera is extremely useful in a wide range of applications not tied to video recovery.

4.4 Conclusions and future work

We proposed a framework for the compressive acquisition of dynamic scenes modeled as LDSs. We showed that the strong scene model for the video enables stable reconstructions at very low measurement rates. In particular, this emphasizes the power of video models that are predictive as well as static.

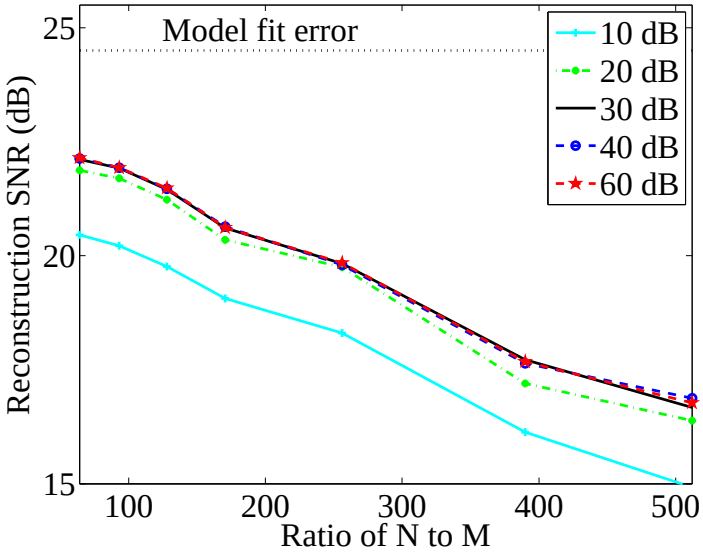


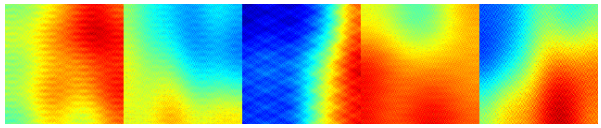
(a) Mosaic of frames of a video, with each column a different time instant, and each row a different algorithm. (top row to bottom) ground truth, oracle LDS, CS-LDS, and frame-by-frame CS.



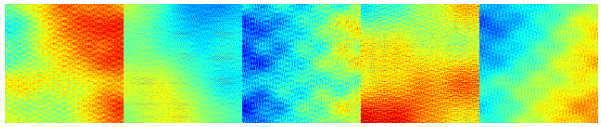
(b) Mosaic of ground truth (top) and estimated (bottom) observation matrix

Figure 31: Reconstruction of a fire texture of length 250 frames and resolution of $N = 128 \times 128$ pixels. Compressive measurements were obtained at $\tilde{M} = 30$ and $\widetilde{M} = 40$ measurements per frame, there by giving a measurement rate of 0.42% of Nyquist. Reconstruction was performed with $d = 20$ and $K = 30$. Frames of the videos are shown in false-color for better contrast.

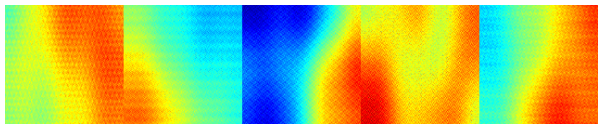




(a) M/N : 0.25%, Input SNR: 10 dB



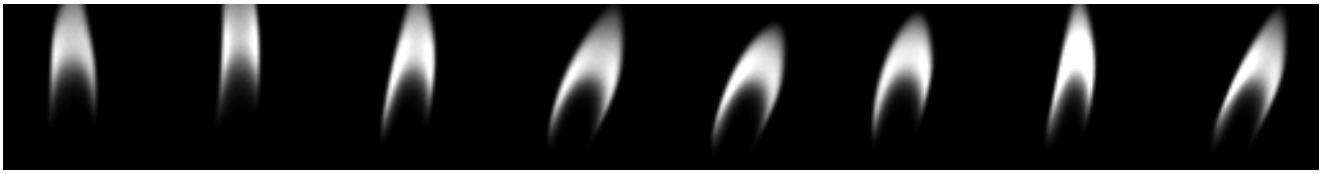
(b) M/N : 0.19%, Input SNR: 10 dB



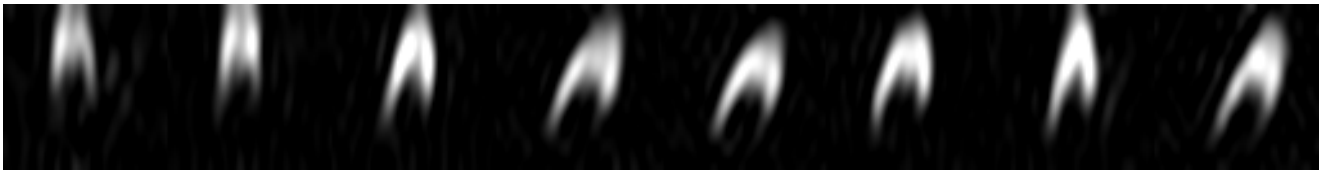
(c) M/N : 0.19%, Input SNR: 30 dB

Figure 32: Resilience of the CS-LDS framework to measurement noise. (Left) Reconstruction SNR as a function of measurement rates and input SNR levels computed using 32 Monte-Carlo simulations. The "black-dotted" line shows the reconstruction SNR for an $d = 20$ oracle LDS. (Right) Snapshots at various operating points. The dynamic texture of Figure 31 was used for this result.

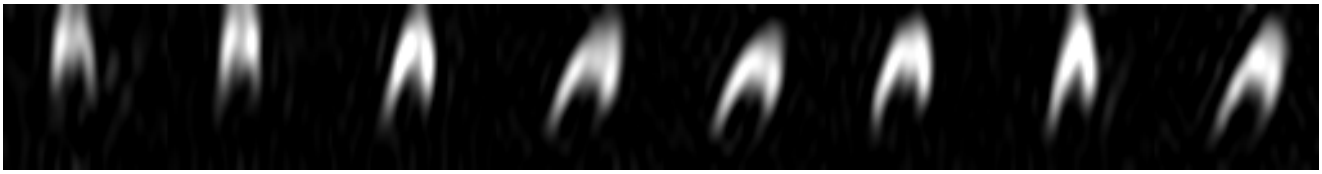
Extensions of the CS-LDS framework: The CS-LDS algorithm proposed requires, at best, $O(d)$ measurements per time instant. This roughly corresponds to the number of degrees of freedom in the dynamics of the video under a d-dimensional LDS model. However, the state transition model of the LDS further constrains the dynamics by providing a model for the evolution of the signal. Incorporating this might help in reducing the number of



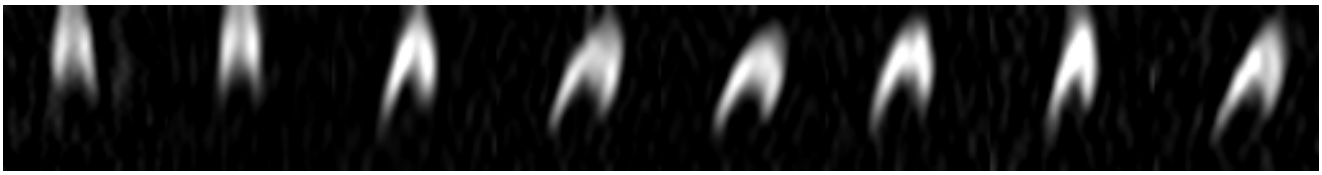
(a) Ground truth



(b) $f_s = 256$ Hz, $\check{M} = 30, \widetilde{M} = 170$, Meas. rate = 5%, SNR = 13.73 dB.



(c) $f_s = 512$ Hz, $\check{M} = 30, \widetilde{M} = 70$, Meas. rate = 2.44%, SNR = 13.73 dB.



(d) $f_s = 1024$ Hz, $\check{M} = 30, \widetilde{M} = 20$, Meas. rate = 1.22%, SNR = 12.63 dB.

Figure 33: As the sampling frequency f_s increases, we maintain the same reconstruction capabilities for significantly lesser number of measurements. Shown are reconstructions for $N = 64 \times 64$ and various sampling frequencies, achieved measurement rates, and reconstruction SNRs.

measurements required at each time instant. Another direction for future research is in fast recovery algorithms that operate at multiple spatio-temporal scales, exploiting the fact that a global LDS model induces a local LDS model as well. Finally, much of the proposed algorithm relies on sparsity of the observation matrix C. Wavelets and Fourier (DCT) bases do not sparsify videos where the motion is localized in space. This suggests the use of dyadic partition methods such as platelets [29], which have been shown to have success in modeling bounded shapes.

Newer models for video CS: While the CS-LDS framework makes a compelling case study of LDSs for video CS, its applicability to an arbitrary video is limited. The LDS model is well-matched to a large class of dynamic textures such as flames, water, traffic etc. but does not extend to simple non-stationary scenes such as people walking. The importance of video models for CS motivates the search for models that are more general than LDS. In this regard, a promising line of future research is to leverage our new understanding of video models for compression algorithm-based CS recovery.

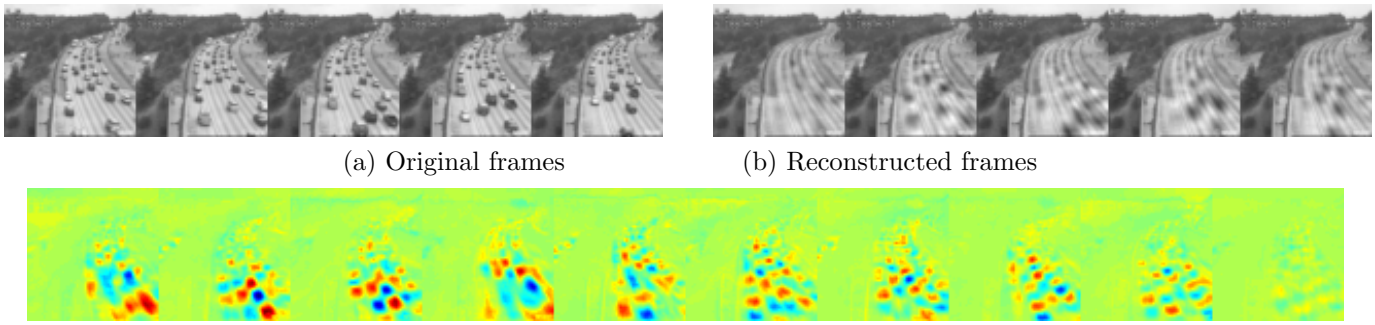
5 Joint manifolds for data fusion

In this section we would like to highlight our results in applying our developed JMM theory to relevant image processing problems.

5.1 Joint manifolds for binary classification

We applied the random projection-based fusion algorithm of [30] to perform binary classification. Suppose a number of synthetic cameras, each with resolution N, observe the motion of a truck along a straight road.[10] This forms a

[10] Our synthetic images were generated using POVRAY (*http://www.povray.org*), an open-source ray tracing software package.



(a) Original frames (b) Reconstructed frames

(c) Estimates of observation matrix C

Figure 34: Reconstructions of a traffic scene of $N = 64 \times 64$ pixels at a measurement rate 4%, with $d = 15$ and $K = 40$. The quality of reconstruction and LDS parameters is sufficient for capturing the flow of traffic.

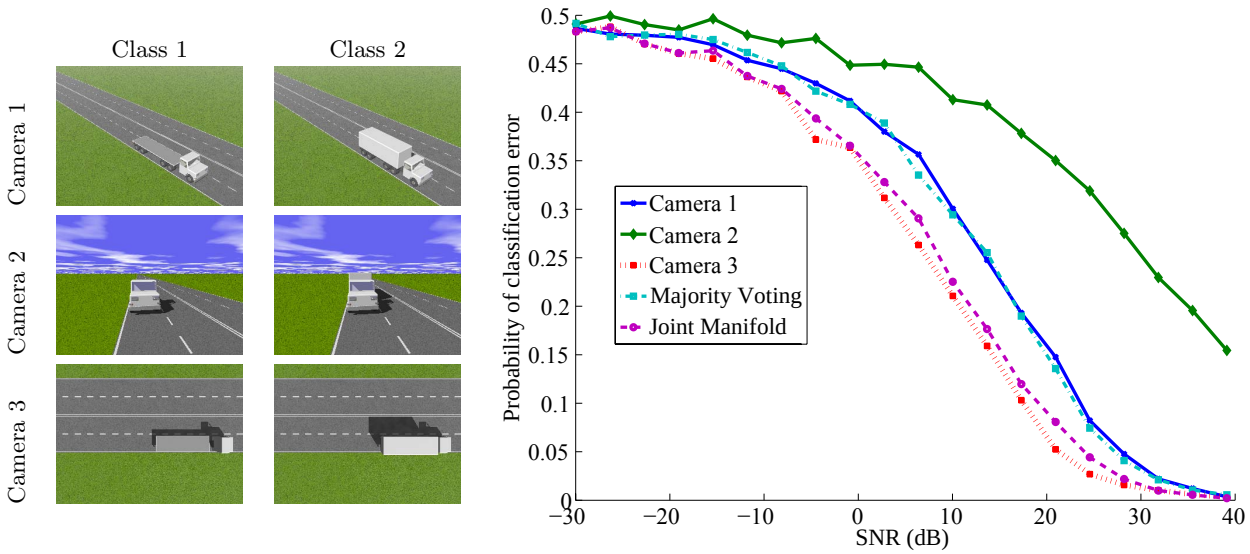


Figure 35: *Sample images of 2 different trucks from multiple camera views and SNR vs. probability of error for individual cameras, the joint manifold, and majority voting. The number of pixels in each camera image $N = 240 \times 360 = 76800$. Joint manifold-based classification outperforms majority voting and performs nearly as well as the best camera.*

1-D manifold in the image space $\mathbb{R}^N$ pertaining to each camera; the joint manifold is also a 1-D manifold in $\mathbb{R}^{JN}$. Suppose now that we wish to classify between two types of trucks. Example images from three camera views for the two classes are shown in Fig. 35. The resolution of each image is $N = 240 \times 320 = 76800$ pixels. In our experiment, we convert the images to grayscale and sum $M = 200$ random projections for the three camera views. The sample camera views suggest that some views make it easier to distinguish between the classes than others. For instance, the head-on view of the two trucks is very similar for most shift parameters, while the side view is more appropriate for discerning between the two classes of trucks.

The probability of error, which in this case is given by $\frac{1}{2}P_{\mathcal{MN}} + \frac{1}{2}P_{\mathcal{NM}}$, for different manifold-based classification approaches as a function of the signal-to-noise ratio (SNR) is shown in Fig. 35. It is clear that the joint manifold approach performs better than majority voting and is comparable in performance to the best camera. While one might hope to be able to do even better than the best camera, our theoretical work in [30] suggests that in general this is only possible when no camera is significantly better than the average camera. Moreover, in the absence of prior information regarding how well each camera truly performs, the best strategy for the central processor would be to fuse the data from *all* cameras. Thus, joint manifold fusion proves to be more effective than high-level fusion algorithms like majority voting.

This example highlights two scenarios when our proposed approach should prove useful. First, our method acts as a simple scheme for data fusion in the case when most cameras do not yield particularly reliable data (and thus decision fusion algorithms like majority voting are ineffective.) Second, due to the high dimensionality of the data,

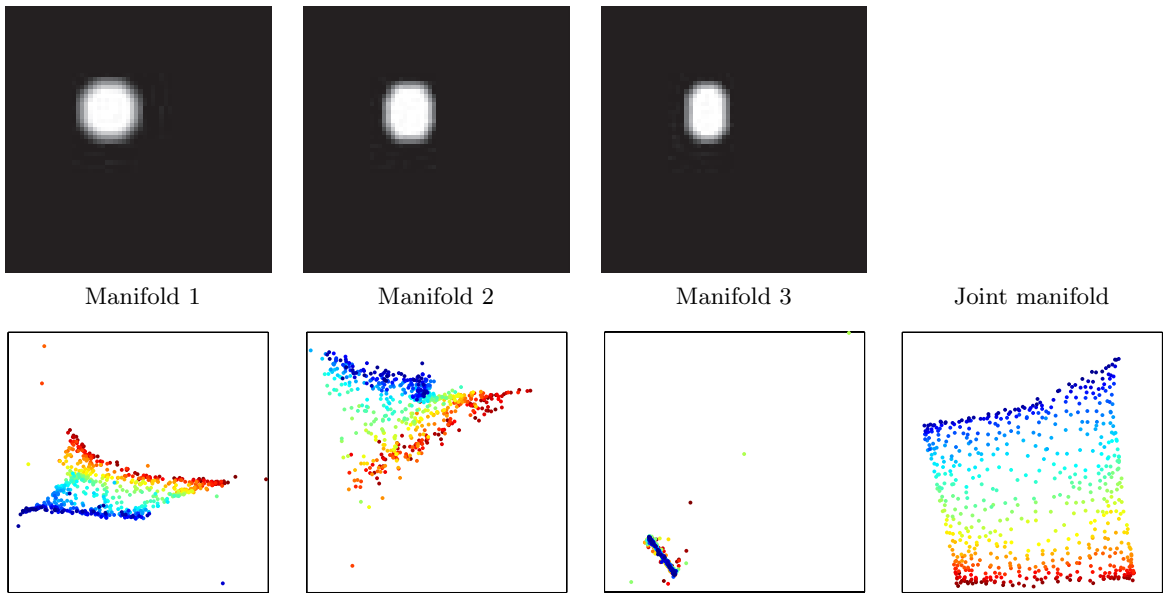


Figure 36: *(top) Articulation manifolds sharing a common 2-D parameter space Θ. Images simulate viewing a translating disc from $J = 3$ viewing angles. (bottom) 2-D embedding of individual and joint manifolds learned via Isomap.*

transmitting the images could be expensive in terms of communication bandwidth. Our method ensures that the communication cost is reduced to be proportional only to the number of degrees of freedom of the signal.

5.2 Joint manifold learning to image processing problems

We used our methods in [30] towards image processing problems in order to demonstrate the significant gains obtained by using the joint manifold model, both with and without the use of random projections. The manifold learning results have been generated using Isomap [31]. For ease of presentation, all of our experiments are performed on 2-D image manifolds isomorphic to a closed rectangular subset of $\mathbb{R}^2$. Thus, ideally the 2-D embedding of the data should resemble a rectangular grid of points that correspond to the samples of the joint manifold in high dimensional space.

Manifolds isometric to Euclidean space

As a first example, we considered three different manifolds formed by $N = 64 \times 64 = 4096$ pixel images of an ellipse with major axis a and minor axis b translating in a 2-D plane, for $(a, b) = (7, 7)$, $(7, 6)$ and $(7, 5)$; an example point is shown in Fig. 36. The eccentricity of the ellipse directly affects the condition number $1/\tau$ of the image articulation manifold; in fact, it can be shown that manifolds associated with more eccentric ellipses exhibit higher values for the condition number. Consequently, we expect that it is "harder" to learn such manifolds. Figure 36 shows that this is indeed the case. We add a small amount of i.i.d. Gaussian noise to each image and apply Isomap to both the individual datasets as well as the concatenated dataset. We observe that the 2-D rectangular embedding is poorly learnt for each of the component manifolds but improves visibly for the joint manifold.

Gaussian noise in realistic images

We then demonstrated how using joint manifolds can help ameliorate imaging artifacts such as Gaussian noise in a more realistic setting. We tested our proposed joint manifold learning approach on a set of synthetic truck images. The data comprises a set of 540 views of a truck on a highway from 3 vantage points. Each image is of size $N = 90 \times 120 = 10800$. The images are parametrized by the 2-D location of the truck on the road; thus, each of the image data sets can be modeled by a 2-D manifold. Sample views are shown in Fig. 35; for this experiment, we only use images from Class 2. We convert the images to grayscale, so that the ambient dimension of the data from each camera lies in $\mathbb{R}^{10800}$. Next, we add i.i.d. Gaussian noise to each image and attempt to learn the 2-D manifold. The noise level is quite high (PSNR = 3.5dB), as evidenced by the sample images in Fig. 37. It is visually clear from the 2-D embedding results that the learning performance improves markedly when the data is modeled using a joint manifold, thus providing numerical evidence for our findings in [30].

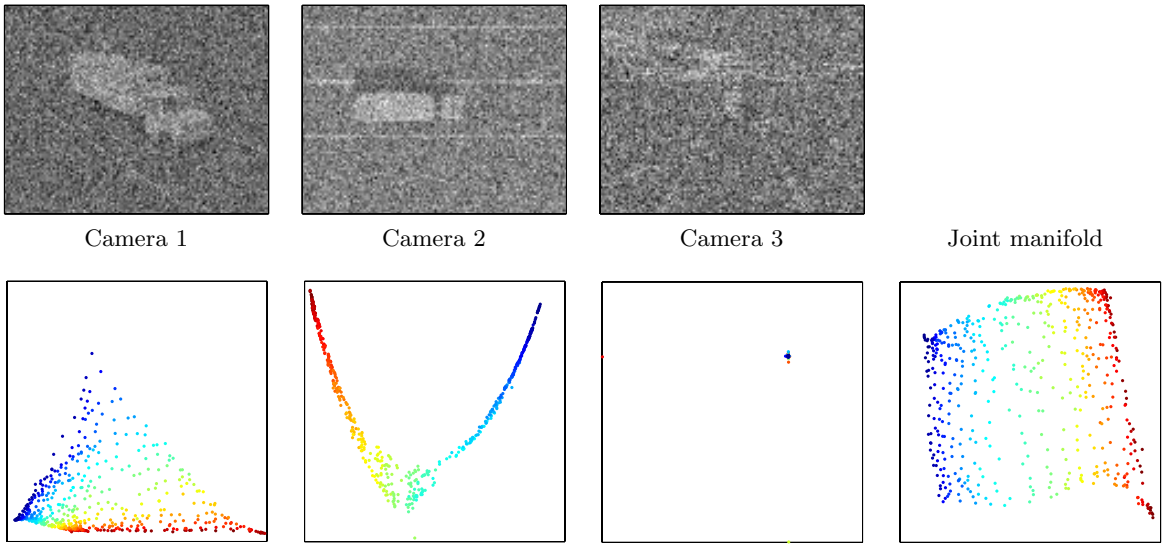


Figure 37: *(top) Noisy captured images. SNR ≈ 3.5 dB ; (bottom) 2-D embeddings learned via Isomap from noisy images. The joint manifold model helps ameliorate the effects of noise.*

Real data experiment—learning with occlusions

We then tested our methods on data from a camera network; the images are obtained from a network of four Unibrain Fire-i$^{\text{TM}}$ OEM Firewire board cameras. Each camera has resolution $N = 320 \times 240 = 76800$ pixels. The data comprises $J = 4$ different views of the independent motions of 2 toy koalas along individual 1-D paths, yielding a 2-D combined parameter space. This data suffers from real-world artifacts such as fluctuations in illumination conditions and variations in the pose of the koalas; further, the koalas occlude one another in certain views or are absent from certain views depending on the particular vantage point. Sample images and 2-D embedding results are displayed in Fig. 38. We observe that the best embedding is obtained by using the modified version of Isomap for learning the joint manifold. To test the effectiveness of the data fusion method described in [30], we compute $M = 2400$ random projections of each image and sum them to obtain a randomly projected version of the joint data and repeat the above experiment. The dimensionality of the projected data is only 3% of the original data; yet, we see very little degradation in performance, thus displaying the effectiveness of random projection-based fusion.

5.3 Towards video fusion: joint manifold learning for target trajectory recovery

Finally, we considered a situation where we are given a set of training data consisting of images of a target moving through a region along with a set of test images of the target moving along a particular trajectory. We do not explicitly incorporate any known information regarding the locations of the cameras or the parameter space describing the target's motion. The training images comprise $J = 4$ views of a coffee mug placed at different positions on an irregular rectangular grid. Example images from each camera are shown in Fig. 39. For the test data, we translate the coffee mug so that its 2-D path traces out the shape of the letter "R". We aim to recover this shape using both the test and training data. To solve this problem, we attempt to learn a 2-D embedding of the joint manifold using the modified version of Isomap detailed in [30]. The learned embedding for each camera is shown in Fig. 39. As is visually evident, learning the data using any one camera yields very poor results; however learning the joint manifold helps discern the 2-D structure to a much better degree. In particular, the "R" trajectory in the test data is correctly recovered only by learning the joint manifold. Finally, we repeat the above procedure using $M = 4800$ random projections of each image, and fuse the data by summing the measurement vectors. While the recovered trajectory of the anomalous (test) data suffers some degradation in visual quality, we observe comparable 2-D embedding results for the individual and joint manifolds as with the original data set. Since the dimensionality of the projected data is merely 6% that of the original data set, this would translate to significant savings in communication costs in a real-world camera network.

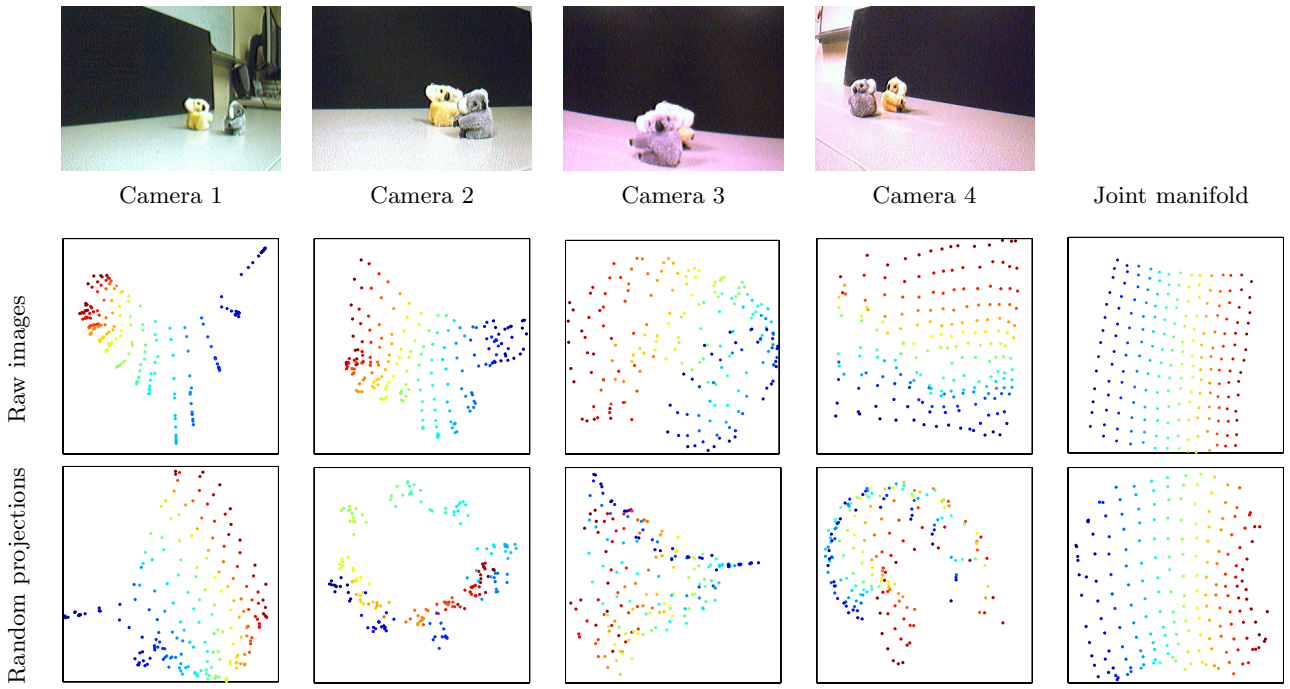


Figure 38: *(top) Sample images of 2 koalas moving along individual 1-D paths, yielding a 2-D manifold; (middle) 2-D embeddings of the dataset learned via Isomap from $N = 76800$ pixel images; (bottom) 2-D embeddings of the dataset learned from $M = 2400$ random projections. Learning the joint manifold yields a much improved 2-D embedding.*

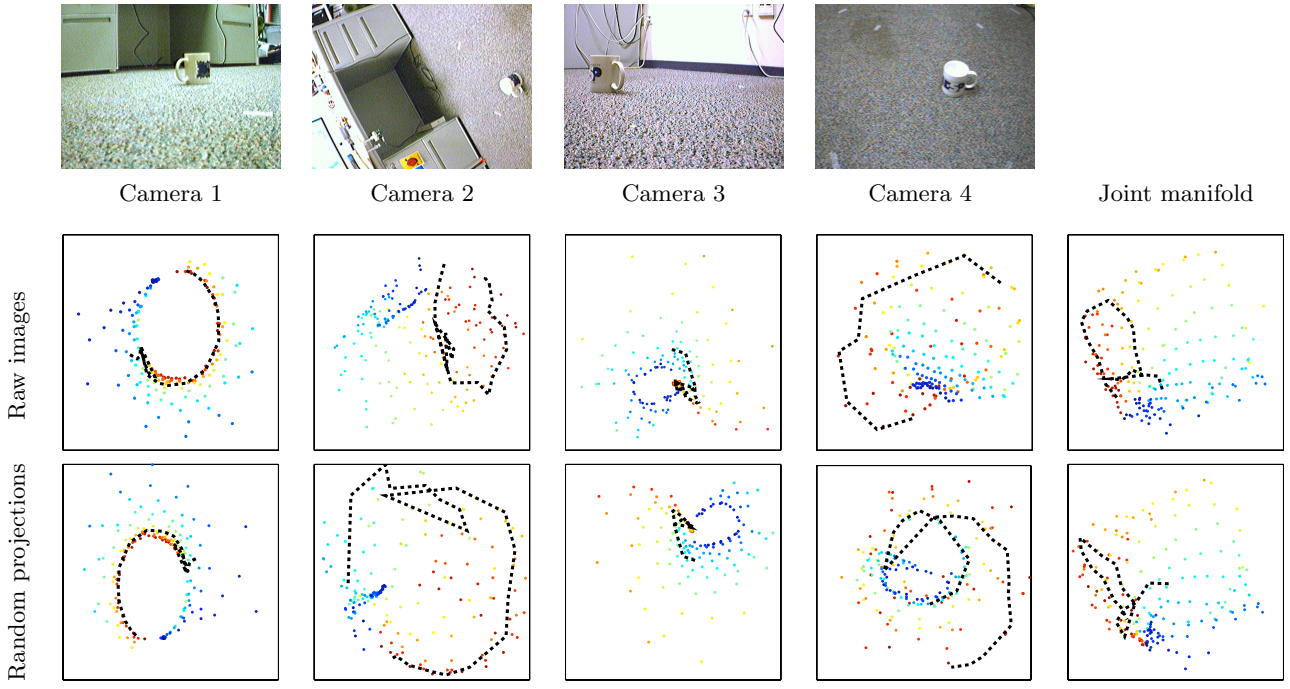


Figure 39: *(top) Sample images of the 2-D movement of a coffee mug; (middle) 2-D embeddings of the dataset learned via Isomap from $N = 76800$ pixel images; (bottom) 2-D embeddings of the dataset learned via Isomap from $M = 4800$ random projections. The black dotted line corresponds to an "R"-shaped trajectory in physical space. Learning the joint manifold yields a much improved 2-D embedding of the training points, as well as the "R"-shaped trajectory.*

5.4 Conclusions and future work

Joint manifolds naturally capture the structure present in the data produced by camera networks. We have provided some basic examples of our studies of joint manifolds and how they can improve the performance of common signal

processing algorithms. We have also introduced a simple framework for data fusion for camera networks that employs independent random projections of each image, which are then accumulated to obtain an accurate low-dimensional representation of the joint manifold. Our fusion scheme can be directly applied to the data acquired by such devices. Furthermore, while we have focused primarily on camera networks in this paper, our framework can be used for the fusion of signals acquired by many generic sensor networks, as well as multimodal and joint audio/video data.

6 Publications supported by this grant

M. A. Davenport, C. Hegde, M. F. Duarte, and R. G. Baraniuk, "Joint manifolds for data fusion," *IEEE Transactions on Image Processing*, vol. 19, pp. 2580–2594, Oct. 2010.

M. A. Davenport, C. Hegde, M. F. Duarte, and R. G. Baraniuk, "High-dimensional data fusion via joint manifold learning," in *AAAI Fall 2010 Symposium on Manifold Learning*, (Arlington, VA), Nov. 2010.

A. C. Sankaranarayanan, P. K. Turaga, R. G. Baraniuk, and R. Chellappa, "Compressive acquisition of dynamic scenes," in *European Conf. on Computer Vision*, Sept. 2010.

7 Deliverables

Included with this submission is a ZIP file containing the Matlab source code necessary to reproduce the bandlimited video research.

8 Professional personnel

Principal Investigator
Richard G. Baraniuk

Co-Principal Investigator
Rama Chellappa

Consultant
Michael B. Wakin

Postdoctoral Research Associates
Jarvis Haupt, Aswin Sankaranarayanan

Graduate Student Research Assistants
Mark Davenport, Chinmay Hegde, Jason Laska, Manjari Narayan, Mona Sheikh, Andrew Waters

References

[1] D. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, Apr. 2006.

[2] E. Candès and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. Inform. Theory*, vol. 52, pp. 5406–5425, Dec. 2006.

[3] R. G. Baraniuk, "Compressive sensing," *Lecture Notes in IEEE Signal Processing Magazine*, vol. 24, pp. 118–120, Jul. 2007.

[4] 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 Processing Magazine*, vol. 25, pp. 83–91, Mar. 2008.

[5] 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 (PCS)*, (Beijing, China), Apr. 2006.

[6] A. Secker and D. Taubman, "Motion-compensated highly scalable video compression using an adaptive 3d wavelet transform based on lifting," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 2, pp. 1029 –1032 vol.2, Oct. 2001.

[7] M. B. Wakin, D. Donoho, H. Choi, and R. G. Baraniuk, "The multiscale structure of non-differentiable image manifolds," in *Wavelets XI in SPIE International Symposium on Optical Science and Technology*, (San Diego), SPIE, SPIE, Aug. 2005.

[8] M. F. Duarte, M. A. Davenport, M. B. Wakin, J. N. Laska, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "Multiscale random projections for compressive classification," in *IEEE International Conference on Image Processing (ICIP)*, (San Antonio, TX), Sep. 2007.

[9] M. A. Davenport, C. Hegde, M. F. Duarte, and R. G. Baraniuk, "A theoretical analysis of joint manifolds," *Rice University, Department of Electrical and Computer Engineering, Technical Report TREE 0901*, Jan. 2009.

[10] Y. Wang, J. Ostermann, and Y. Q. Zhang, *Video processing and communications*. Prentice Hall, 2002.

[11] R. E. Ziemer and W. H. Tranter, *Principles of Communication: Systems, Modulation and Noise*.

[12] E. Meijering and M. Unser, "A note on cubic convolution interpolation," *IEEE Transactions on Image Processing*, vol. 12, pp. 477–479, April 2003.

[13] A. C. Sankaranarayanan, P. K. Turaga, R. G. Baraniuk, and R. Chellappa, "Compressive acquisition of dynamic scenes," in *European Conf. on Computer Vision*, Sept. 2010.

[14] J. Y. Park and M. B. Wakin, "A multiscale framework for compressive sensing of video," in *Proc. Picture Coding Symp. (PCS)*, 2009.

[15] M. B. Wakin, "A manifold lifting algorithm for multi-view compressive imaging," in *Proc. Picture Coding Symp. (PCS)*, 2009.

[16] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[17] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, 2008.

[18] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4036–4048, 2006.

[19] E. van den Berg and M. P. Friedlander, "Probing the pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008.

[20] D. Needell and J. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[21] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, "Model-Based Compressive Sensing," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982–2001, 2010.

[22] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto, "Dynamic textures," *International Journal of Computer Vision*, vol. 51, no. 2, pp. 91–109, 2003.

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

[24] A. Veeraraghavan, A. K. Roy-Chowdhury, and R. Chellappa, "Matching shape sequences in video with applications in human movement analysis," *TPAMI*, vol. 27, pp. 1896–1909, 2005.

[25] P. Turaga, A. Veeraraghavan, and R. Chellappa, "Unsupervised view and rate invariant clustering of video sequences," *CVIU*, vol. 113, no. 3, pp. 353–371, 2009.

[26] J. Fowler, "Compressive-projection principal component analysis,," *IEEE Transactions on Image Processing*, vol. 18, October 2009.

[27] R. Péteri, S. Fazekas, and M. Huiskes, "DynTex: A Comprehensive Database of Dynamic Textures," *(to appear)*, p. URL: http://projects.cwi.nl/dyntex/, 2010.

[28] P. Saisan, G. Doretto, Y. Wu, and S. Soatto, "Dynamic texture recognition," in *CVPR*, vol. 2, pp. 58–63, December 2001.

[29] R. Willett and R. Nowak, "Platelets: a multiscale approach for recovering edges and surfaces in photon-limited medical imaging," *IEEE Transactions on Medical Imaging*, vol. 22, no. 3, pp. 332–350, 2003.

[30] M. A. Davenport, C. Hegde, M. F. Duarte, and R. G. Baraniuk, "Joint manifolds for data fusion," *IEEE Transactions on Image Processing*, vol. 19, pp. 2580–2594, Oct. 2010.

[31] J. Tenenbaum, V. Silva, and J. Landford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.