SIAM J. Imaging Sciences
Vol. 8, No. 1, pp. 373–402

Space-Time Regularization for Video Decompression[*]

Hayden Schaeffer[†], Yi Yang[‡], and Stanley Osher[§]

Abstract. We consider the problem of reconstructing frames from a video which has been compressed using the video compressive sensing (VCS) method. In VCS data, each frame comes from first subsampling the original video data in space and then averaging the subsampled sequence in time. This results in a large linear system of equations whose inversion is ill-posed. We introduce a convex regularizer to invert the system, where the spatial component is regularized by the total variation seminorm, and the temporal component is regularized by enforcing sparsity on the difference between the spatial gradients of each frame. Since the regularizers are L^1-like norms, the model can be written in the form of an easy-to-solve saddle point problem. The saddle point problem is solved by the primal-dual algorithm, whose implementation calls for nearly pointwise operations (i.e., no direct linear inversion) and has a simple parallel version. Results show that our model decompresses videos more accurately than other popular models, with PSNR gains of several dB.

Key words. video decompression, compressive sensing, total variation, temporal regularization

AMS subject classifications. 49M30, 49M29, 94A08, 90C46

DOI. 10.1137/140977400

1. Introduction.

Adaptive video compression and decompression is at the frontier of mathematical image processing. Since the dominant source of data is video-based and the demand for transmitting videos continues to grow, the importance of constructing reliable and fast tools to process such data is becoming increasingly necessary. In this work, we focus on the reconstruction of compressed sequences of frames acquired from what is known as video compressive sensing (VCS).

Classical compression algorithms, such as MPEG, acquire all incoming signals before compressing them in a given basis. For example, in MPEG-IV, the video is stored as a sequence, where the first frame of the sequence is compressed with respect to the wavelet basis [10, 9]. After the first frame is stored, the following frames are compressed by taking the difference between the current frame and the first frame with respect to the wavelet basis. If the difference is large, then the sequence is stored, and the process begins again with a new first frame and resulting sequence.

Recently, there have been works in the literature which propose new video compression schemes using the compressive sensing framework in order to increase the gains in storage

[*] Received by the editors July 14, 2014; accepted for publication (in revised form) October 17, 2014; published electronically February 12, 2015.
http://www.siam.org/journals/siims/8-1/97740.html

[†] Computing and Mathematical Sciences, Caltech, Pasadena, CA 91125 (schaeffer@caltech.edu). The research of this author was supported by NSF grant 1303892.

[‡] Skytree, San Jose, CA 95110 (yi@skytree.net). The research of this author was supported by NSF DMS 0835863, NSF DMS 0914561, and ONR N00014-11-0749.

[§] Department of Mathematics, University of California at Los Angeles, Los Angeles, CA 90095-1555 (sjo@math.ucla.edu).

and recovery. This is done by compressing the incoming data before the signal is sensed by the device, typically by means of a time-varying aperture coding function which prevents a fixed amount of information from reaching the lens. The coded aperture appears as a random or pseudorandom grating, which subsamples the signal in space. This differs from the classical framework since the incoming signal is not sensed directly, but rather a spatially subsampled version is acquired. Some recent work on the construction and implementation of coded apertures and compressive sensing cameras can be found in coded aperture compressive temporal image (CACTI) systems [22], coded aperture snapshot spectral imaging (CASSI) [35, 36], cooperative analog and digital signal processing transform imager (CADSP) [19], and sparse MRI [23]. For other examples of this methodology being implemented in hardware and software, see [14, 6, 37, 8, 16, 5, 24, 28]. A related and connected topic, in terms of hardware adaptation for image acquisition, can be found in [29, 1, 34], where a flutter shutter is used to better condition the resulting motion blur kernels.

A computational difficulty in the VCS framework is the reconstruction of this type of compressed data. Many of the mathematical methods found in the literature for recovering compressed sensing data focus on the reconstruction of single image data—for example, those working on the preservation of edges [13, 38, 17, 27, 3] or texture [11, 32, 18]. However, the temporal aspect of videos introduces the need for different types of regularization which leverage the particular structure of the data and the compression.

From the convex optimization perspective, the popular regularizers for video recovery rely on generalizing total variation to higher dimensions. Let $u = u(x, y, t)$ be a continuous spatiotemporal function over $\Omega \times [0, T]$, where $\Omega \subset \mathbb{R}^2$ and $T > 0$. Note that $u(x, t)$ will refer to $u(x, y, t)$ in some sections below. Recall that the two-dimensional (2D) isotropic total variation regularizer is defined as [31]

$$TV_2(u) := \int_\Omega \left| \sqrt{(\partial_x u)^2 + (\partial_y u)^2} \right|.$$

Total variation regularization and the related models are able to reconstruct piecewise constant images accurately and capture edge information found in most images. To extend this regularizer to video data, one can directly treat the time component as an additional space variable and form the corresponding seminorm [20, 21]

$$TV_3(u) := \int_{\Omega\times[0,T]} \left| \sqrt{(\partial_x u)^2 + (\partial_y u)^2 + (\partial_t u)^2} \right|.$$

In this form, all dimensions are coupled, and the expected behavior is uniform in all directions. An equivalent seminorm on the space of three-dimensional (3D) functions of bounded variation takes the form [14, 33]

$$TV_{2+1}(u) := \int_{\Omega\times[0,T]} \left| \sqrt{(\partial_x u)^2 + (\partial_y u)^2} \right| + \lambda \int_{\Omega\times[0,T]} |\partial_t u|$$

with λ positive. We denote this seminorm by TV_{2+1} since it decouples the two spatial dimensions from the temporal one. The TV_{2+1} regularizer takes into account the differing behaviors of the spatial and temporal components, which could allow for more flexibility in penalizing in-frame oscillations versus sporadic error between frames. Both seminorms provide sufficient regularization to reconstruct video data compressed in terms of a global basis. However, when the video data is spatially compressed using a subsampling operation, as is done in CACTI systems [22], this treatment of the temporal component is insufficient [39]. In this work, we propose a higher-order variation on the temporal component which promotes the correct behavior in the reconstructed solutions, which we verify with a variety of videos.

Related work on recovering videos from VCS data can be found in [40, 39]. The algorithm proposed in [40, 39] uses a Gaussian mixture model to represent each 3D patch in the data set, with the assumption that the space of patches lives on a union of subspaces and each patch is drawn from one subspace. Unlike other dictionary-based algorithms, the inversion of the compressive sensing operator in [40, 39] is analytic, making it more computationally efficient.

This paper is organized as follows. A description of our reconstruction model is given in section 2. Characterization of the model and minimizers along with some analytic remarks are provided in section 3. Section 4 details the primal-dual algorithm and its implementation for our model. Experiment results and comparisons on synthetic and real video sequences are provided in section 5. We conclude with several final remarks in section 6.

2. Description of the model.

In this section, we present a mathematical formulation of the linear operator which defines the data acquisition process and then detail our model for recovering VCS data.

2.1. Compressed data.

The data is acquired by first spatially subsampling each frame in the original video and then taking the running average [43, 22, 42, 33, 41]. In mathematical terms, we can define the corresponding projection operator $P : \mathbb{R}^{N\times N\times T} \to \mathbb{R}^{N\times N}$ over a video sequence as

$$P(u) = \frac{1}{T}\sum_{j=1}^{T} P_{S_j}(u_j),$$

where N^2 is the number of pixels in each frame and T is the number of frames. The sequence u is defined by the collection of frames u_j, i.e., $u := [u_1, u_2, \dots, u_T]$. The set S_j corresponds to the spatial locations in frame j where the mask is transparent (i.e., “on”). For each set, $S_j \subset \Omega$, where Ω is the spatial domain, the framewise projection is defined by the standard operation

$$P_{S_j}(u_j(x)) = \begin{cases} u_j(x) & \text{if} \quad x \in S_j, \\ 0 & \text{if} \quad x \in S_j^c. \end{cases}$$

We define the spatial compression rate (or ratio) as the size of S_j divided by the size of Ω, which is assumed to be (nearly) uniform in j. The data projection operator is typically normalized by T and hence represents the running average of the incoming compressed signal. Alternatively, the normalization can be done pointwise by dividing each pixel by the number of instances of acquired data:

$$P(u(x)) = \frac{\sum_{j=1}^{T} P_{S_j}(u_j(x))}{\sum_{j=1}^{T} P_{S_j}(1)},$$

where we take $\frac{0}{0} := 0$. For the mathematical exposition here, this normalization factor is not directly considered; however, our method readily applies in this case.

In practice, the data is acquired adaptively, in the sense that sequences with larger values of T have relatively less motion, while a sequence with smaller values of T may be more dynamic [22, 42, 33, 30]. For the model presented here, this fact is not directly used but could be incorporated via the parameter choices described in section 5. Overall, the inverse problem we consider here is as follows: given one compressed sequence $f \in \mathbb{R}^{N\times N}$, find $u \in \mathbb{R}^{N\times N\times T}$ such that $P(u) = f$.

2.2. Our model. Since it is not necessarily true that adjacent compressed frames carry similar information (for example, there could be an abrupt change in the motion of the objects in the foreground or a complete change in the scene), we will consider here the decompression of a sequence of frames given *one compressed measurement*. Within the sequence there are three main dimensions, two spatial and one temporal, and we must regularize in both space and time in order to fill in all missing information. The regularizer also must be strong enough to deal with the various types of issues caused by the compression: (1) blurring caused by the motion of objects between frames over the lapsed time (scene-based motion), (2) the blurring effect caused by averaging the frames together (camera-based motion), (3) noise in the sensor array during the acquisition process, and (4) the number of missing pixels and the overall distribution of information in each frame.

As a starting point, we seek a model similar to the Rudin–Osher–Fatemi (ROF) model for still images [31]:

$$\int_\Omega |\nabla u(x,0)| \;\; \mathrm{dx} + \frac{\mu}{2}\int_\Omega (u(x,0)-f)^2\,\mathrm{dx},$$

where μ is a positive parameter and f is the input data. So we assume that each individual frame can be well approximated by a piecewise constant function. Also, we take as a basic assumption that the sequence is defined over a finite time interval where the frames are made up of a stationary background and a moving foreground. Therefore, the appropriate spatial regularizer is the average total variation for each frame:

$$\text{Spatial:} \quad \frac{1}{T}\sum_{j=1}^{T} \|\nabla u_j\|_1 := \frac{1}{T}\sum_{j=1}^{T}\int_\Omega |\nabla u_j| \;\; \mathrm{dx}.$$

Unless otherwise stated, we consider all vector norms to be the standard l^2 norm. The spatial regularizer ensures that each reconstructed frame contains sharp edges and regions of homogeneous intensity, thereby reducing the effects of blur and noise in the sequence. Also, it has been shown that total variation regularized decompression provides a near-optimal stable method for image recovery [26].

Based on the assumption of piecewise constant frames, it is clear that the difference between frames $u_{j+1} - u_j$ must remain piecewise constant for consistency. Therefore, we regularize the temporal component by the average total variation applied to the difference between adjacent frames:

$$\text{Temporal:} \quad \frac{1}{T}\sum_{j=1}^{T} \|\nabla u_{j+1} - \nabla u_j\|_1 := \frac{1}{T}\sum_{j=1}^{T} \int_\Omega |\nabla u_{j+1} - \nabla u_j| \ \text{dx},$$

where $u_{T+1} = u_1$. The temporal regularizer acts partly like a *temporal fidelity* term which communicates gradient-based information between the frames and partly as a penalty term to decrease the sporadic intensity displacement that could occur. This helps mitigate the effects of subsampling and improves the reconstruction when the sampling rate in space is low and/or when T is large. The choice of this regularizer is also related to the assumption that the velocity field acts locally uniformly and can be discontinuous. This is applicable to real data, since many of the moving objects found in stationary videos will move at a locally uniform speed as long as the object is physically rigid in nature (i.e., nonexpansive and/or with low acceleration).

Altogether, the noise-free decompression model can be written as a convex (constrained) optimization problem,

$$\min_u \ \frac{1}{T}\sum_{j=1}^{T} \|\nabla u_j\|_1 + \frac{\lambda}{T}\sum_{j=1}^{T} \|\nabla u_{j+1} - \nabla u_j\|_1 \qquad \text{s.t.} \quad P(u) = f, \tag{2.1}$$

and the noisy recovery model can be written as a convex (unconstrained) optimization problem,

$$\min_u \ \frac{1}{T}\sum_{j=1}^{T} \|\nabla u_j\|_1 + \frac{\lambda}{T}\sum_{j=1}^{T} \|\nabla u_{j+1} - \nabla u_j\|_1 + \frac{\mu}{2}\|P(u) - f\|_{L^2}^2, \tag{2.2}$$

where λ and μ are positive parameters. We impose a periodic boundary condition in space and time. This condition is consistent in time since we assume that the velocity field can be discontinuous. Also, since the vector norms used are l^2, the model is invariant under rotations of the video. It is also invariant to contrast, scaling, and translation of the video.

To verify the scaling between terms in our model, consider the continuous in time case where the lapse time between the frames, Δt, and the time interval, I, are shrinking. First, we will fix I and send $\Delta t \to 0^+$. The number of frames, T, is inversely proportional to the lapse time, i.e., $T = \frac{I}{\Delta t}$. For simplicity, we assume that the projection operators are identity (i.e., $S_j = \Omega$). Using the Taylor series, the difference between consecutive frames is

$$u_{j+1} - u_j := \Delta t \ \partial_t u_j + \mathcal{O}(\Delta t^2),$$

and thus,

$$\nabla u_{j+1} - \nabla u_j := \Delta t \ \nabla \partial_t u_j + \mathcal{O}(\Delta t^2).$$

Also, for any well-behaved sequence variables v_j, the running average can be approximated by

$$\frac{1}{T}\sum_{j=1}^{T} v_j = \frac{1}{I}\int_0^I v_j \mathrm{dt} + \mathcal{O}(\Delta t^2).$$

If the space-time mixed derivatives (i.e., $\nabla\partial_t u_j$) are bounded in L^1, the second term in (2.2) goes to zero as $\Delta t \to 0^+$. Therefore, for fixed $u(x,t)$, sending $\Delta t \to 0^+$ in the energy yields

$$\frac{1}{I}\int_0^I \int_\Omega |\nabla u_j| \ \mathrm{dx}\ \mathrm{dt} + \frac{\mu}{2}\int_\Omega \left(\frac{1}{I}\int_0^I u\ \mathrm{dt} - f\right)^2 \mathrm{dx}.$$

If we send the time interval $I \to 0^+$, by the Lebesgue differentiation theorem (and by switching some of the operations) we have

$$\int_\Omega |\nabla u(x,0)| \ \mathrm{dx} + \frac{\mu}{2}\int_\Omega (u(x,0)-f)^2 \,\mathrm{dx},$$

which is the ROF model on the initial frame $u(x,0)$. This argument is only formal and is used to verify that the choice of coefficients in the model (in terms of T) is correct with respect to our assumptions.

In the derivation above we take the difference $u_{j+1}-u_j$ to be an approximation to the time derivative. However, for discrete time, we consider it to be a measure of similarity between consecutive frames and not necessarily corresponding to close-in-time dynamics. It could be advantageous to take large time steps (Δt large) during the compression, thereby storing more information. In that case, only significant events in the video need to be captured, and the rest of the motion can be interpolated between frames.

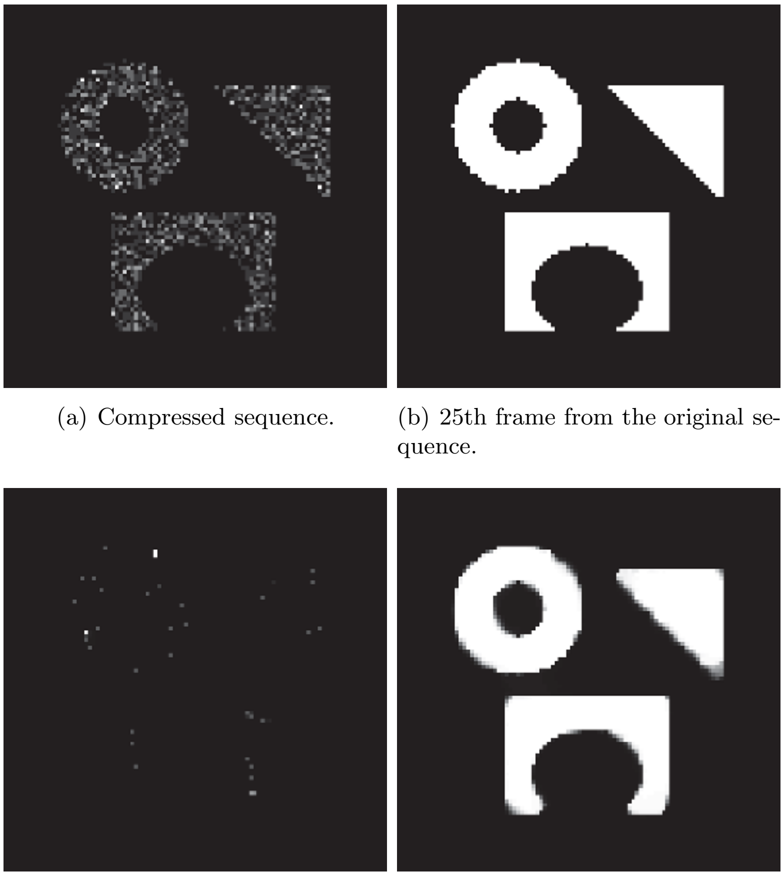
As an example, we compare our model with a model that only contains spatial regularization (by taking $\lambda = 0$). In Figure 1, a sequence with 50 repeated frames is compressed using the VCS method with an overall compression rate of 0.04% and then recovered with the total variation in space model (see Figure 1(c)) and our method (see Figure 1(d)). The average peak signal-to-noise ratio (PSNR) using our model is more than three times larger. Also based on visual comparison, we can see that it is necessary to use temporal regularization in order to reconstruct the objects in the video.

2.3. Relation to known models.

Since our model uses a total variation regularizer, it is related to other 3D extensions of the ROF model. In particular, we will consider the standard TV_3 model (for example, see [20, 21, 2]):

$$(TV_3)\quad \min_u \left\|\sqrt{(\partial_x u)^2 + (\partial_y u)^2 + (\partial_t u)^2}\right\|_{L^1} + \frac{\mu}{2}\|Au-f\|_{L^2}^2.$$

where A is a general (possibly ill-posed) linear operator. In this model the third dimension is treated in the same way as space and thus does not take advantage of the structure of the data. One benefit is that in this form the problem is easy to solve since one could make use of any algorithm that works in the 2D case.



(a) Compressed sequence. (b) 25th frame from the original sequence.

(c) 25th frame from the TV model. (d) 25th frame from our model.

Figure 1. *The standard Shapes image is repeated for* 50 *frames and has been compressed to* 2% *in space for a total compression rate of* 0.04%. *Each frame is compressed with an independently generated random mask. In this example, we show the benefits of a temporal regularizer. Our model provides a reconstructed sequence with an average peak signal-to-noise ratio (PSNR) of* 22.95, *while the spatial-only regularized model has an average PSNR of* 7.00. *Notice that this PSNR value for the spatial-only regularized model is not too low, since the background value in the original video is* 0.

Recall that the TV_{2+1} model separates the spatial and temporal terms as follows:

$$(TV_{2+1}) \quad \min_u \left\| \sqrt{(\partial_x u)^2 + (\partial_y u)^2} \right\|_{L^1} + \lambda \left\| \partial_t u \right\|_{L^1} + \frac{\mu}{2} \|Au - f\|_{L^2}^2.$$

Although the regularizer is an equivalent seminorm on $BV(\Omega \times [0,T])$, the minimizers will generally be different. This model also has the advantage that it can be solved using standard algorithms. In both of these models, the main assumption for the temporal term is that the optical velocity is sparse in time [14]. This is effective when the subsampling operation is done within a global basis, since the basis itself prevents localized effects such as pixelation (in space) or flickering (in time). However, when the data is randomly subsampled in space, the sparse in time assumption is no longer valid, as can be seen in numerical results presented in section 5.

3. Analytic remarks.

In this section, we introduce an alternative formulation of our regularizer in terms of duality. Using the dual form, we are able to characterize solutions in terms of the coefficients in the model. Also, we verify two basic properties that the reconstruction should contain in order to be a valid recovery model.

3.1. Characterization of semidiscrete minimizers.

We consider the semidiscrete function space, where the spatial variables are continuous and the temporal variable is discrete. The energy is defined for sequences of functions $u := [u_1, \dots, u_T] \in (BV)^T$. Since the regularizer is L^1-like, it can be written as a maximization against L^∞-like elements. Using this duality, we can provide a meaningful range for μ for fixed λ and T.

Lemma 3.1. *Consider the semidiscrete norm defined for $T > 0$ and $\lambda \geq 0$ on the sequence $u := [u_1, \dots, u_T] \in (BV(\Omega))^T$:*

$$||u|| := \frac{1}{T}\sum_{j=1}^{T} \|u_j\|_{TV} + \frac{\lambda}{T}\sum_{j=1}^{T} \|u_{j+1} - u_j\|_{TV}.$$

The norm has the equivalent representation

$$||u|| := \sup_{v \in V} \frac{1}{T}\sum_{j=1}^{T} \langle Hu_j, v_j \rangle ,$$

where the operator is defined as $H := [\nabla, \nabla S_+]$ and S_+ is the forward finite difference operator (in terms of the index j). The corresponding dual space can be identified as the space of functions w whose components can be written as $w_j = -\text{div}\ \phi_j$, with $\phi_j \in L^\infty(\Omega \times \Omega)$. The corresponding dual norm is as follows:

$$||w||_* = \inf_{\phi,\psi}\ \max\left\{ ||\phi_j - S_-\psi_j||_{L^\infty}, \frac{1}{\lambda}||\psi_j||_{L^\infty} \right\},$$

where S_- is the negative of the backward finite difference operator (i.e., $S_- = S_+^*$).

Proof. By considering the dual characterization of the TV seminorm, we can write our seminorm in a linear form [25]:

$$\begin{aligned} ||u|| &= \frac{1}{T}\sum_{j=1}^{T} \|u_j\|_{TV} + \frac{\lambda}{T}\sum_{j=1}^{T} \|u_{j+1} - u_j\|_{TV} \\ &= \sup_{||\phi_j||_\infty, ||\psi_j||_\infty \leq 1} \frac{1}{T}\sum_{j=1}^{T} \langle \nabla u_j, \phi_j \rangle + \frac{\lambda}{T}\sum_{j=1}^{T} \langle \nabla u_{j+1} - \nabla u_j, \psi_j \rangle . \end{aligned}$$

Next, by rescaling ψ_j (and reusing notation for simplicity) we have the following equation:

$$||u|| = \sup_{||\phi_j||_\infty \leq 1, ||\psi_j||_\infty \leq \lambda} \frac{1}{T}\sum_{j=1}^{T} \langle \nabla u_j, \phi_j \rangle + \frac{1}{T}\sum_{j=1}^{T} \langle \nabla u_{j+1} - \nabla u_j, \psi_j \rangle .$$

In this form, it is clear that by defining the operator $H := [\nabla, \nabla S_+]$ and the function space

$$V := \{v_j \mid v_j = [\phi_j, \psi_j],\ ||\phi_j||_\infty \leq 1,\ ||\psi_j||_\infty \leq \lambda\}\,,$$

the norm simplifies to

$$||u|| := \sup_{v \in V} \frac{1}{T} \sum_{j=1}^{T} \langle Hu_j, v_j\rangle\,.$$

Note also that the dual operator of H can be found easily via integration by parts and is $H^*[v_j^1, v_j^2] := -\text{div}\ v_j^1 - \text{div}\ S_-\ v_j^2$, where S_- is the negative of the backward finite difference operator.

To find the corresponding dual norm, we must first find the dual norms of both regularizers separately. For simplicity, we will define the separate norms as

$$||u||_a := \frac{1}{T}\sum_{j=1}^{T} \|u_j\|_{TV},$$

$$||u||_b := \frac{1}{T}\sum_{j=1}^{T} \|u_{j+1} - u_j\|_{TV}.$$

By definition, the dual norm of $||\cdot||_a$ is

$$||w||_{a,*} := \sup_{||u||_a \leq 1} \langle u, w\rangle = \sup_{||u||_a \leq 1} \frac{1}{T}\sum_{j=1}^{T} \langle u_j, w_j\rangle\,.$$

For all w in the corresponding dual space, we can identify (possibly nonunique) vector fields ϕ_j such that

$$w_j = -\text{div}\ \phi_j.$$

Therefore, the dual norm further simplifies to

$$\begin{aligned} ||w||_{a,*} &= \sup_{||u||_a \leq 1} \frac{1}{T}\sum_{j=1}^{T} \langle u_j, -\text{div}\ \phi_j\rangle \\ &= \sup_{||u||_a \leq 1} \frac{1}{T}\sum_{j=1}^{T} \langle \nabla u_j, \phi_j\rangle\,. \end{aligned}$$

By the Cauchy–Schwarz inequality, we have

$$\frac{1}{T}\sum_{j=1}^{T} \langle \nabla u_j, \phi_j\rangle \leq \frac{1}{T}\sum_{j=1}^{T} ||u_j||_{TV} ||\phi_j||_{L^\infty} \leq \max_j \left(||\phi_j||_{L^\infty}\right) ||u||_a,$$

which provides an upper bound to this dual norm (minimizing with respect to all choices of ϕ). This bound can be obtained by taking ∇u_J to be a (rescaled) Dirac delta function at a global maximum of ϕ_J, where J is the index which maximizes the L^∞ norms over the sequence $\{\phi_j\}$. Therefore, the dual norm is

$$||w||_{a,*} = \inf_{\phi} \max_{j} \left(||\phi_j||_{L^\infty}\right),$$

where $w_j = -\text{div } \phi_j$ for all $1 \leq j \leq T$. Similarly, for the dual norm of $||\cdot||_b$, we have by definition

$$||w||_{b,*} := \sup_{||u||_b \leq 1} \langle u, w \rangle = \sup_{||u||_b \leq 1} \frac{1}{T} \sum_{j=1}^{T} \langle u_j, w_j \rangle .$$

Following the same argument as above, the dual norm simplifies to

$$\begin{aligned} ||w||_{b,*} &= \sup_{||u||_b \leq 1} \frac{1}{T} \sum_{j=1}^{T} \langle u_j, -\text{div } S_- \phi_j \rangle \\ &= \sup_{||u||_b \leq 1} \frac{1}{T} \sum_{j=1}^{T} \langle \nabla S_+ u_j, \phi_j \rangle . \end{aligned}$$

By the Cauchy–Schwarz inequality, we have

$$\frac{1}{T} \sum_{j=1}^{T} \langle \nabla S_+ u_j, \phi_j \rangle \leq \frac{1}{T} \sum_{j=1}^{T} ||S_+ u_j||_{TV} ||\phi_j||_{L^\infty} \leq \max_{j} \left(||\phi_j||_{L^\infty}\right) ||u||_b,$$

which provides an upper bound as before. This bound can be obtained by taking $\nabla S_+ u_J$ to be a (rescaled) Dirac delta function at a global maximum of ϕ_J, where J is the index which maximizes the L^∞ norms over the sequence $\{\phi_j\}$. Therefore, the dual norm is

$$||w||_{b,*} = \inf_{\phi} \max_{j} \left(||\phi_j||_{L^\infty}\right),$$

where $w_j = -\text{div} S_- \ \phi_j$ for all $1 \leq j \leq T$.

Altogether, the dual norm of our regularizer, $||\cdot|| = ||\cdot||_a + \lambda ||\cdot||_b$, is given by

$$\begin{aligned} ||w||_* = \inf_{v,s} \ & s \\ \text{s.t. } & ||w - v||_{a,*} \leq s \ ||v||_{b,*} \leq \lambda s. \end{aligned}$$

To further simplify it, we can identify elements w and v in the dual space by $w_j = -\text{div } \phi_j$ and $v_j = -\text{div } S_- \ \psi_j$ so that the norm becomes

$$\begin{aligned} ||w||_* = \inf_{v,s} \ & s \\ \text{s.t. } & \inf_{\phi,\psi} ||\phi_j - S_- \psi_j||_{L^\infty} \leq s, \ \inf_{\psi} ||\psi_j||_{L^\infty} \leq \lambda s, \\ & \text{where } w_j = -\text{div } \phi_j \ \text{ and } \ v_j = -\text{div } S_- \ \psi_j, \end{aligned}$$

or (after some calculation), in a more compact form,

$$||w||_* = \inf_{\phi,\psi} \ \max\left\{ ||\phi_j - S_- \psi_j||_{L^\infty}, \frac{1}{\lambda}||\psi_j||_{L^\infty} \right\}$$
$$\text{s.t. } w_j = -\text{div } \phi_j,$$

where the direct dependence on v_j can be removed. ■

Next, recall the noisy model (2.2), and define the corresponding energy functional E over sequences $u \in (BV)^T$:

$$E(u) = \frac{1}{T}\sum_{j=1}^{T} \|u_j\|_{TV} + \frac{\lambda}{T}\sum_{j=1}^{T} \|u_{j+1} - u_j\|_{TV} + \frac{\mu}{2}\|P(u) - f\|^2_{L^2}. \tag{3.1}$$

Note that the adjoint operator of P is

$$P^*(u) = [P_{S_1} v, \ P_{S_2} v, \dots, P_{S_n} v],$$

which will be needed in the following theorem, whose proof is related to those found in [25, 12, 32].

Theorem 3.2. *Let E be the energy from (3.1) and $||\cdot||_*$ be the dual norm from Lemma 3.1, and define the following seminorm on $(BV)^T$:*

$$||u|| = \frac{1}{T}\sum_{j=1}^{T} \|u_j\|_{TV} + \frac{\lambda}{T}\sum_{j=1}^{T} \|u_{j+1} - u_j\|_{TV};$$

then the following relations hold:

- *If $||P^* f||_* \leq \frac{1}{\mu}$, then the optimal solution of $\min_u E(u)$ is $u = 0$.*
- *If $||P^* f||_* \geq \frac{1}{\mu}$, then the optimal solution of $\min_u E(u)$ satisfies $||P^*(f - Pu)||_* = \frac{1}{\mu}$ and $\mu \langle u, P^*(f - Pu)\rangle = ||u||$.*

Theorem 3.2 provides some guidelines for choosing μ given the compressed sequence f and a fixed scale λ. To determine λ, note that by the triangle inequality we have the following bound on the ratio between the two terms:

$$\frac{\frac{\lambda}{T}\sum_{j=1}^{T} \|u_{j+1} - u_j\|_{TV}}{\frac{1}{T}\sum_{j=1}^{T} \|u_j\|_{TV}} \leq 2\lambda. \tag{3.2}$$

Therefore, since we want to weigh their importance equally, it is best to choose $\lambda \in [0, 1]$, with typical values near $\frac{1}{2}$.

3.2. Further analytical remarks. We would like the model to exhibit certain properties when the data falls into two extremal cases. The first is that when the projection operator contains no subsampling in space, we prefer that the minimizing sequence not be able to distinguish information between frames. This is a modeling assumption to ensure that the reconstruction does not bias information between frames. Therefore, in this case we would like the optimal solution to be constant in time.

Property 3.3 (degeneracy). *If $S_j = \Omega$ for all $1 \leq j \leq T$, then a constant sequence is a minimizer.*

Proof. For any sequence $\{u_j\}_{j=1}^T$, define its average as $\bar{u} := \frac{1}{T}\sum_{j=1}^T u_j = P(u)$. The energy can be bounded from below:

$$\begin{aligned}
E(u) &= \frac{1}{T}\sum_{j=1}^{T} \|u_j\|_{TV} + \frac{\lambda}{T}\sum_{j=1}^{T} \|u_{j+1}-u_j\|_{TV} + \frac{\mu}{2}\|P(u)-f\|_{L^2}^2 \\
&\geq \left\| \frac{1}{T}\sum_{j=1}^{T} u_j \right\|_{TV} + \frac{\lambda}{T}\sum_{j=1}^{T} \|u_{j+1}-u_j\|_{TV} + \frac{\mu}{2}\|P(u)-f\|_{L^2}^2 \\
&\geq \left\| \frac{1}{T}\sum_{j=1}^{T} u_j \right\|_{TV} + \frac{\mu}{2}\|P(u)-f\|_{L^2}^2 \\
&= \|\bar{u}\|_{TV} + \frac{\mu}{2}\|\bar{u}-f\|_{L^2}^2 \\
&= E(\bar{u}).
\end{aligned}$$

Since the functional is convex, the minimizer must indeed be obtained by the constant sequence $\{\bar{u}\}_{j=1}^T$. ■

This may seem counterintuitive at first, within the framework of compressive sensing, since the more information one is given, the better the solution recovery rate should be. However, the subsampling in space provides us with the additional information corresponding to the pixel-frame relationship. By connecting some of the pixels to specific frames or subsets of frames, we can better identify structures and features in the entire sequence.

Second, we expect that if a sequence is nearly trivial (i.e., contains only one nontrivial frame), then minimizers should not depend on the location of the nontrivial frame.

Property 3.4 (consistency). *For any j, if $u_k = 0$ for all $k \neq j$, then u_j must solve*

$$\min_{u_j} \left\{ (1+2\lambda)\|u_j\|_{TV} + \frac{\mu}{2}\|u_j - f\|_{L^2}^2 \right\}.$$

This proof follows directly. This also provides further support for the use of the temporally periodic boundary condition, since with it the model directly inherits this property.

4. Numerical method.

Since the data can be large, both the storage and use of large linear operators may be expensive. Therefore, we use the simple first order method called the primal-dual algorithm [44, 7, 4, 15]. The primal-dual algorithm requires that the energy minimization problem can be written in the form of a saddle point problem. We outline the construction of the related saddle point problem below.

Using Lemma 3.1, (2.2) can be written as a saddle point problem:

$$\min_{u} \max_{||v||_\infty \leq 1, ||w||_\infty \leq \lambda} \sum_{j=1}^{T} \langle \nabla u_j, v_j \rangle + \sum_{j=1}^{T} \langle \nabla (u_{j+1}-u_j), w_j \rangle + \frac{\mu T}{2}\|P(u)-f\|_{L^2}^2,$$

where v and w are the dual variables. We can also introduce another variable z by taking the

Legendre dual of the fidelity term to get

$$\min_u \max_{||v||_\infty \le 1,\ ||w||_\infty \le \lambda,\ z} \sum_{j=1}^{T} \langle \nabla u_j, v_j \rangle + \sum_{j=1}^{T} \langle \nabla (u_{j+1} - u_j), w_j \rangle + \langle P(u) - f, z \rangle - \frac{1}{2\mu T} ||z||_{L^2}^2.$$

This equation can be written as an unconstrained saddle point problem by taking the L^∞ constraints and including them as a penalty to the energy:

$$\begin{aligned}(4.1)\qquad \min_u \max_{v,w,z} \sum_{j=1}^{T} \langle \nabla u_j, v_j \rangle + \lambda \sum_{j=1}^{T} \langle \nabla (u_{j+1} - u_j), w_j \rangle + \langle P(u) - f, z \rangle \\ - \frac{1}{2\mu T} ||z||_{L^2}^2 - \chi_B(v) - \chi_B\left(\frac{w}{\lambda}\right),\end{aligned}$$

where B is the L^∞ ball with radius 1 and $\chi_B(x)$ is 0 if $x \in B$ and ∞ otherwise. This is the form of the problem which we solve numerically and which is equivalent to (2.2).

4.1. Primal-dual algorithm. Given a saddle point problem of the form

$$\min_u \max_v F(u) + \langle Au, v \rangle - G(v),$$

where F and G are convex functions and A is a linear operator, the iterative updates defined by the primal-dual algorithm are [44, 7, 4, 15]

$$\begin{aligned} u^{n+1} &= (I + \tau \partial F)^{-1}(u^n - \tau A^T v^n), \\ \bar{u}^{n+1} &= 2u^{n+1} - u^n, \\ v^{n+1} &= (I + \sigma \partial G)^{-1}(v^n + \sigma A \bar{u}^{n+1}), \end{aligned}$$

where ∂F and ∂G are the subdifferential of F and G, respectively. The inverse operators above are the proximal operators defined by

$$(I + \tau \partial F)^{-1}(z) = \min_v \left(f(v) + \frac{||v - z||^2}{2\tau} \right).$$

For our model, $F(u) := 0$ and $G(v, w, z) := \chi_B(v) + \chi_B\left(\frac{w}{\lambda}\right) + \frac{1}{2\mu T}||z||_{L^2}^2 + \langle f, z \rangle$, and $Au = [\nabla u, \nabla S_+ u, P(u)]$. The proximal operators are simple to calculate, thereby making the algorithm easy to implement. We summarize the update rules below.

For the u substep, since F is trivial, the update is

$$u^{n+1} = u^n + \tau \mathrm{div}\ v^n + \tau \mathrm{div}\ S_- w^n - \tau P^* z^n.$$

In terms of the dual variables, the function G is separable; therefore, the updates are completely decoupled. For the v substep, the update is

$$\begin{aligned} v^{n+1} &= (I + \sigma \partial \chi_B)^{-1}(v^n + \sigma \nabla u^{n+1}) \\ &= \mathrm{Proj}_B(v^n + \sigma \nabla u^{n+1}). \end{aligned}$$

Similarly for the w substep, the update is

$$w^{n+1} = \lambda \ \mathrm{Proj}_B(w^n + \sigma \nabla S_+ u^{n+1}),$$

and finally the z substep is given simply by

$$\begin{aligned} z^{n+1} &= \left(I + \sigma \partial \frac{1}{2\mu T} || \cdot ||_{L^2}^2 \right)^{-1} (z^n + \sigma(Pu^{n+1} - f)) \\ &= \frac{z^n + \sigma(Pu^{n+1} - f)}{1 + \frac{\sigma}{\mu T}}. \end{aligned}$$

The projection operator Proj_B is defined for vectors by

$$\mathrm{Proj}_B(x) := \frac{z}{|z|_{l^2}},$$

where $\frac{0}{0} := 0$. Altogether the algorithm is summarized below.

Algorithm 1 Primal-dual algorithm in the noisy case (equation (2.2)).

```
Parameters: $\tau$, $\sigma$, $\lambda$, $\mu$.
Initialize: $P$, $f$, $u^0$, $v^0$, $w^0$, $z^0$.
while $||u^{n+1} - u^n|| \geq tol$ do
    $u^{n+1} = u^n + \tau \mathrm{div}\ v^n + \tau\lambda \mathrm{div}\ S_- w^n - \tau P^* z^n,$
    $\bar{u}^{n+1} = 2u^{n+1} - u^n,$
    $v^{n+1} = \mathrm{Proj}_B(v^n + \sigma \nabla \bar{u}^{n+1}),$
    $w^{n+1} = \lambda \ \mathrm{Proj}_B(w^n + \sigma \nabla S_+ \bar{u}^{n+1}),$
    $z^{n+1} = \frac{z^n + \sigma(P\bar{u}^{n+1} - f)}{1 + \frac{\sigma}{\mu T}}$
end while
return $u$
```

Similarly, the constrained model (equation (2.1)) can be written as the following saddle point problem:

$$\min_u \max_{||v||_\infty \leq 1, \ ||w||_\infty \leq \lambda, \ z} \sum_{j=1}^{T} \langle \nabla u_j, v_j \rangle + \sum_{j=1}^{T} \langle \nabla (u_{j+1} - u_j), w_j \rangle + \langle P(u) - f, z \rangle .$$

The variable z is the Lagrange multiplier for the constraint $P(u) = f$. This saddle point problem can also be seen as the limit of (4.1) as $\mu \to \infty$. The algorithm is summarized below and is nearly identical to that of the unconstrained problem (Algorithm 1).

Algorithm 2 Primal-dual algorithm in the noise-free case (equation (2.1)).

```
Parameters: $\tau$, $\sigma$, $\lambda$.
Initialize: $P$, $f$, $u^0$, $v^0$, $w^0$, $z^0$.
while $||u^{n+1} - u^n|| \geq tol$ do
    $u^{n+1} = u^n + \tau \mathrm{div}\ v^n + \tau\lambda \mathrm{div}\ S_- w^n - \tau P^* z^n,$
    $\bar{u}^{n+1} = 2u^{n+1} - u^n,$
    $v^{n+1} = \mathrm{Proj}_B(v^n + \sigma \nabla \bar{u}^{n+1}),$
    $w^{n+1} = \lambda\ \mathrm{Proj}_B(w^n + \sigma \nabla S_+ \bar{u}^{n+1}),$
    $z^{n+1} = z^n + \sigma(P\bar{u}^{n+1} - f)$
end while
return $u$
```

Convergence of both algorithms above is guaranteed for σ and τ which satisfy $\sigma\tau \leq \frac{1}{||A||_{op}}$, where $||A||_{op}$ is the operator norm of A. For our algorithm, the bound $\sigma\tau \leq \frac{1}{\sqrt{41}}$ suffices. Also, we should note that the primal and dual residuals decay at an appropriate rate; thus residual balancing techniques do not accelerate the convergence of our model.

4.2. A parallel version. The algorithm is nearly pointwise since it requires only the knowledge of the neighboring pixel values (in space-time). This is particularly advantageous for implementation on graphics processing unit (GPU) arrays. Furthermore, we present a simple version of the algorithm that can be run in parallel. A naive patchwise parallel algorithm

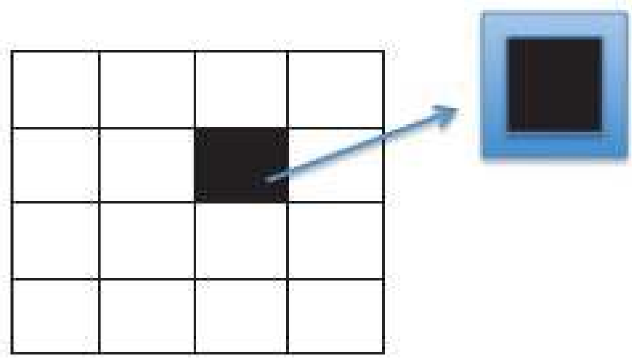


Figure 2. *For the parallel version of our algorithm, each frame is split into patches (on the left). Then each patch is extended to include some of the neighboring pixels (the blue boundary on the right). To recombine the patches, a spatial cutoff function is defined over each patch. The cutoff is equal to* 1 *in the patch (the black region), is equal to* 0 *outside of the patch, and is continuous on the overlap region (the blue region).*

would need to communicate with neighboring blocks in order to converge to the same solution as the nonparallel algorithm. Hence a good parallel algorithm should take into account the structure of the entire algorithm as well as the structure of the data to approximate the nonparallel solution.

In our approach, one generates a grid of nonoverlapping patches. Directly solving the algorithm on each patch is less expensive than working on the entire data set, since the time cost of Algorithms 1 and 2 depends directly on the number of pixels in the solution u. Therefore, each patch should be sent to different devices to be calculated in parallel. However, along the boundary of each patch, the solutions may not contain the proper continuity and thus have visual artifacts. To avoid patch-boundary artifacts, each patch is partially extended (see Figure 2). A spatial cutoff function $Q(p_i)$ is introduced as a function of each patch, p_i, which is defined as 1 in the patch, 0 outside the extended region of the patch, and continuous otherwise. Then, to recombine the patches into the frames, we take a weighted average with respect to the cutoff Q (i.e., the weight values are determined by the values of the cutoff function). In practice, we notice that a bilinear spline optimizes the visual quality of the recombination by not blurring around patch boundaries, as compared to higher-order splines. Also, the patch boundaries need only be extended by a few pixels to properly stitch the solution together.

For a simple example, consider a one-dimensional (1D) vector of data and extract two patches p_1 and p_2 which overlap on three points $\{1, 2, 3\}$. Then the linear spline extension is

$$Q(p_1)(x) = -\frac{x}{4} + 1 \quad \text{and} \quad Q(p_2)(x) = \frac{x}{4}$$

for $x \in \{0, 1, \dots, 4\}$. Given a function defined on the patches, which we will denote by u_{p_i}, the recombined value u is determined by

$$u(x) = \frac{Q(p_1)(x)u_{p_1}(x) + Q(p_1)(x)u_{p_2}(x)}{Q(p_1)(x) + Q(p_2)(x)} = Q(p_1)(x)u_{p_1}(x) + Q(p_1)(x)u_{p_2}(x),$$

since the interpolants sum to one, i.e., $Q(p_1)(x) + Q(p_2)(x) = 1$.

Note that the solution computed by this parallel algorithm is not necessarily the numerical solution of (2.2) but rather an approximation. The convergence rate, in terms of the number of iterations of either Algorithm 1 or Algorithm 2 required to reach a specified tolerance, does not significantly change depending on the patch size. Therefore, it is best to keep the number of patches close to the number of parallel devices.

As an example, in Figure 3 we compare Algorithm 1 to the parallel version described above using 49 patches. Both visual and PSNR comparisons show very little difference in the results. For the remaining examples, we will use Algorithm 1 directly and not the parallel version, since this is not the main focus of the work.

5. Experimental results.

In this section, we apply our algorithm to various compressed video sequences. The parameters are provided with each of the results and are chosen using the guidelines in section 3.1. The parameter λ should be larger when the motion present in the video sequence is relatively small and can be smaller when the motion in the video sequence is relatively large. In the experiments, we generate independent random masks for

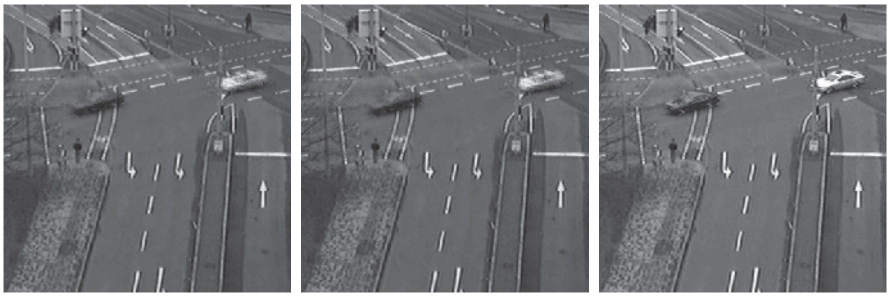

(a) Nonparallel. (b) Parallel. (c) Original frame.

Figure 3. *Comparison between Algorithm* 1 *and the parallel version on a sequence of four frames. The nonparallel method results in an average PSNR of* 34.3499 *with a PSNR of* 34.4915 *on the frame shown in* (a)*. The parallel method uses* 49 *patches since the size of each frame is* 350 × 350 *and the size of each extended patch is* 55 × 55 *(the extension is of length* 5*). This results in an average PSNR of* 34.3375 *and a PSNR of* 34.4887 *on the frame shown in* (b)*. The parameters are fixed at* $\lambda = 1$*,* $\mu T = 15$*. The compression rate in space is* 50%*, and the standard deviation of the added noise is* 5*.*

each frame, and we add Gaussian noise to each pixel before compressing the frames. The overall compression rate is given by the product between the compression rates in space and time. Also, each video is normalized to fall in the range of $[0, 255]$. For our model and the TV_{2+1} model, we apply the primal-dual algorithm outlined in the previous section. For the TV_3 model, we use the state-of-the-art implementation called TVAL3 found in [20, 21]. We also compare our model to the $L^1(DCT)$ model, which uses $||DCT(u)||_{L^1}$ as a regularizer (where DCT is the 3D discrete cosine transform). The reconstruction results using the TV_3 and the $L^1(DCT)$ models can be found in section 5.6. Since the results tend to be less visually accurate compared to TV_{2+1}, we do not include them in sections 5.1–5.5.

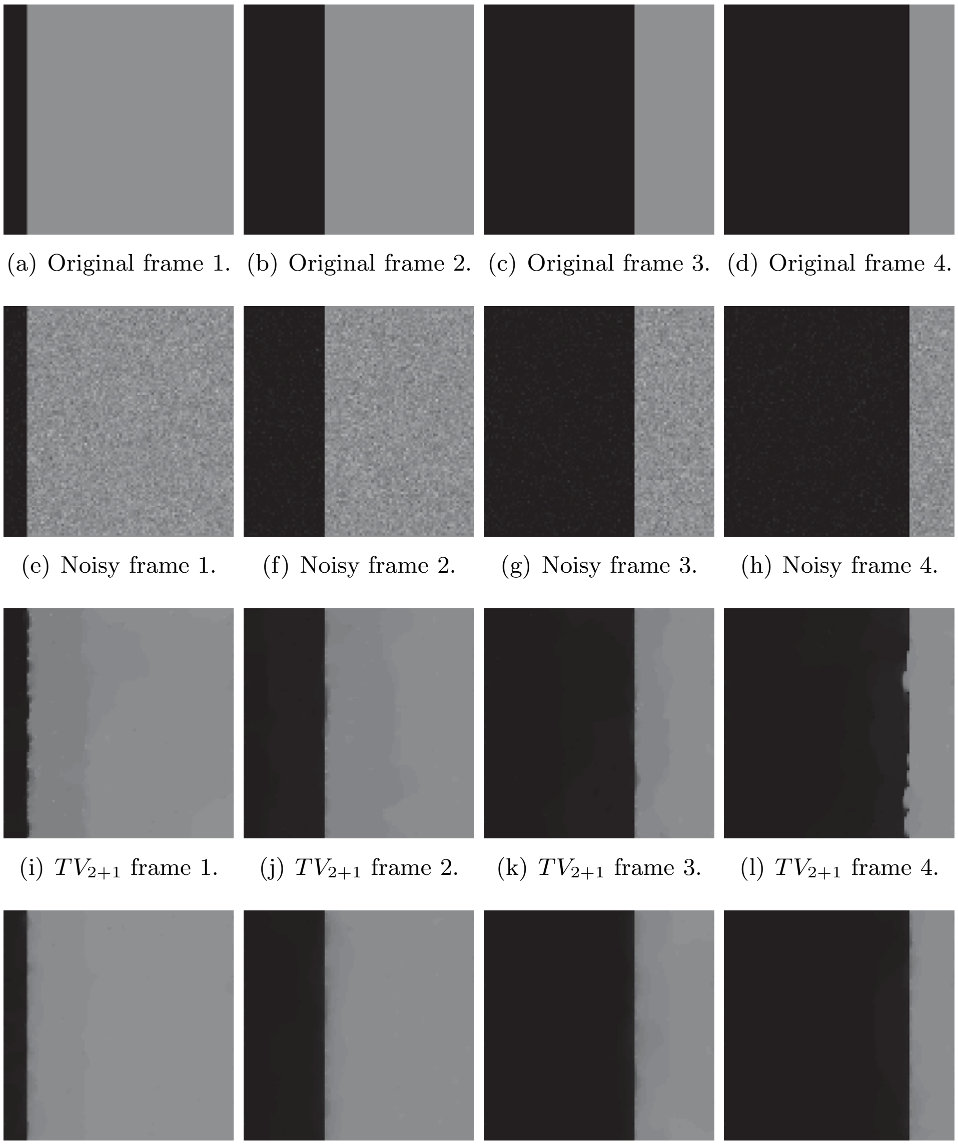
5.1. Synthetic example.

Since our model is closely related to a video extension of the ROF model, we will show that it too captures the correct edge set. In Figure 4, a simple synthetic sequence consisting of a jump moving to the right is compressed with an overall compression rate of 5%. Our model provides a reconstructed sequence with an average PSNR of 28.46, while the TV_{2+1} model has an average PSNR of 23.21. Since the frames' only feature is a high-contrast jump, the main error in TV_{2+1} is along the edge set. In Figure 5 (b) and (c), a Canny edge detector is used to locate the edges of the two reconstructed versions of frame 4, for visual comparison of the regularity of the edge set.

5.2. Parking lot data.

For the remaining examples, our algorithm is applied to real video sequences containing various types of motion. Some parameter optimization is necessary to obtain optimal results. To do so, we first optimize the parameters on a small part of the video. Then new data is generated, and the optimized parameters are applied to the new sequence.

First, we consider a stationary camera observing a parking lot. In Figure 6 one frame from a noisy sequence of 16 frames is shown. Since each frame is nearly piecewise constant, total variation regularized models should be effective. The average PSNR for our method applied to this sequence is 37.08, and the PSNR of the shown frame (Figure 6 (c)) is 37.35. The average PSNR for the TV_{2+1} method applied to this sequence is 30.74, and the PSNR of



(a) Original frame 1. (b) Original frame 2. (c) Original frame 3. (d) Original frame 4.

(e) Noisy frame 1. (f) Noisy frame 2. (g) Noisy frame 3. (h) Noisy frame 4.

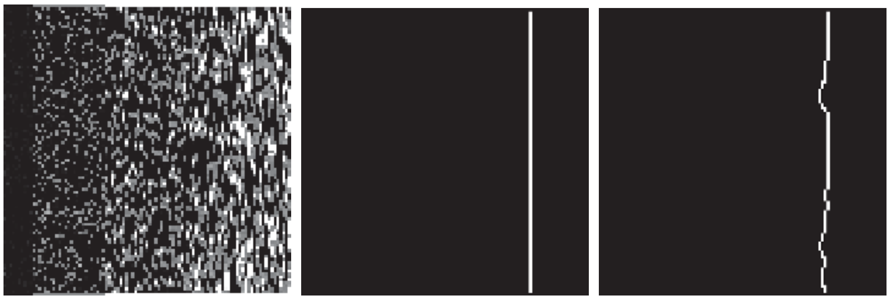
(i) TV_{2+1} frame 1. (j) TV_{2+1} frame 2. (k) TV_{2+1} frame 3. (l) TV_{2+1} frame 4.

(m) Our frame 1. (n) Our frame 2. (o) Our frame 3. (p) Our frame 4.

Figure 4. *This synthetic sequence demonstrates the reconstruction of edge (high contrast) features. The sequence is compressed to* 5% *of the original data. The noise has a standard deviation of* 24.5*. Our model provides a reconstructed sequence with an average PSNR of* 28.46*, while the* TV_{2+1} *regularized model has an average PSNR of* 23.21*.*

the shown frame (Figure 6 (d)) is 31.18.

Our model has a higher PSNR since it is able to communicate the spatial continuity between frames better than the TV_{2+1} model. This can be seen in the homogeneous intensity along the light pole and paint lines in Figure 6 (c) as opposed to the false edges in Figure 6 (d). In Figure 7, the difference between the computed frames and the exact frame is shown (both



(a) Compressed sequence. (b) Edges of our frame 4. (c) Edges of TV_{2+1} frame 4.

Figure 5. *In* (b) *and* (c)*, a Canny edge detector is used to locate the edges of two reconstructed versions of frame* 4*, for visual comparison of the regularity of the edge set.*

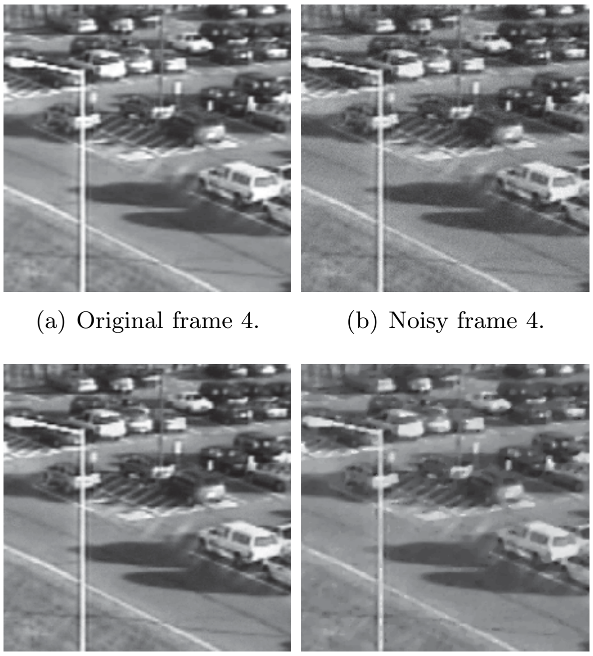
on the same scale). The error between our approximation and the true solution is mainly made up of fine scale details and a smooth component, both of which are known not to be preserved by total variation terms. On the other hand, the error between the TV_{2+1} approximation and the true solution contains edge information and discontinuous details, which are likely an artifact due to the inconsistency between the two regularizers.

In Figure 8, a four-frame sequence is extracted from the parking lot video which contains car and shadow motion. The average PSNR for our method applied to this sequence is 33.66, and the average PSNR for the TV_{2+1} method applied to this sequence is 31.32. The TV_{2+1} reconstruction of the car (located on the bottom right corner of the frames) creates a fuzzy boundary as compared to our reconstruction. One can also see the staircasing and pixelation effect along solid lines in the TV_{2+1} reconstruction.

5.3. Toy car data. Next, we look at the toy car video, which contains a large range of motion. In Figure 9, we compare our method to the TV_{2+1} solution and the original sequence. In this sequence three scales of movement appear: the slow car in the background, the medium speed of the car on the left, and the faster car on the right. The velocity field is close to piecewise constant and can be very discontinuous due to the large jumps between frames. The average PSNR for our method applied to this sequence is 37.96, and the average PSNR for the TV_{2+1} method applied to this sequence is 35.77.

In the zoomed in versions of frames 1 and 2 (see Figure 10), we can see the flickering effect present in the TV_{2+1} reconstruction. The difference between the two TV_{2+1} reconstructed frames will still be sparse since it contains pixelwise peaks. However, this does not appear as the visually correct representation of the true frame. Also, the reconstruction contains a "ghost," or a faint copy of moving objects from neighboring frames, appearing because of the method's inability to correctly separate the frame-by-frame behavior. These effects are diminished in our reconstruction, thereby leading to higher PSNR values.

5.4. Traffic data. In Figure 11, we apply our method to a sequence of traffic/surveillance frames. In this video, multiple scales of objects appear in motion—both cars and people. The average PSNR for our method applied to this sequence is 32.19, and the average PSNR for the TV_{2+1} method applied to this sequence is 29.66. In the zoomed in frame, Figure 12,


(a) Original frame 4. (b) Noisy frame 4.

(c) Our frame 4. (d) TV_{2+1} model's frame 4.

Figure 6. *Our algorithm applied to a parking lot sequence of* 16 *frames with slow car and shadow motion in the bottom right corner. The average PSNR of our reconstruction is* 37.08 *with a PSNR of* 37.35 *in the frame shown in* (c). *The average PSNR of the* TV_{2+1} *reconstruction is* 30.74 *with a PSNR of* 31.18 *in the frame shown in* (d). *The parameters are set to* $\lambda_{ours} = 4$, $\lambda_{TV_{2+1}} = 0.5$, $\mu T = 20$. *The total number of iterations is fixed to a maximum of* 1200. *The compression rate in space is* 50%, *and the standard deviation for the additive noise is* 5.

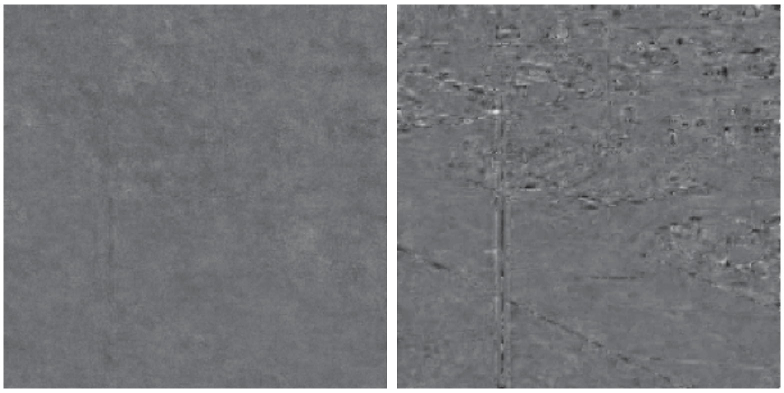
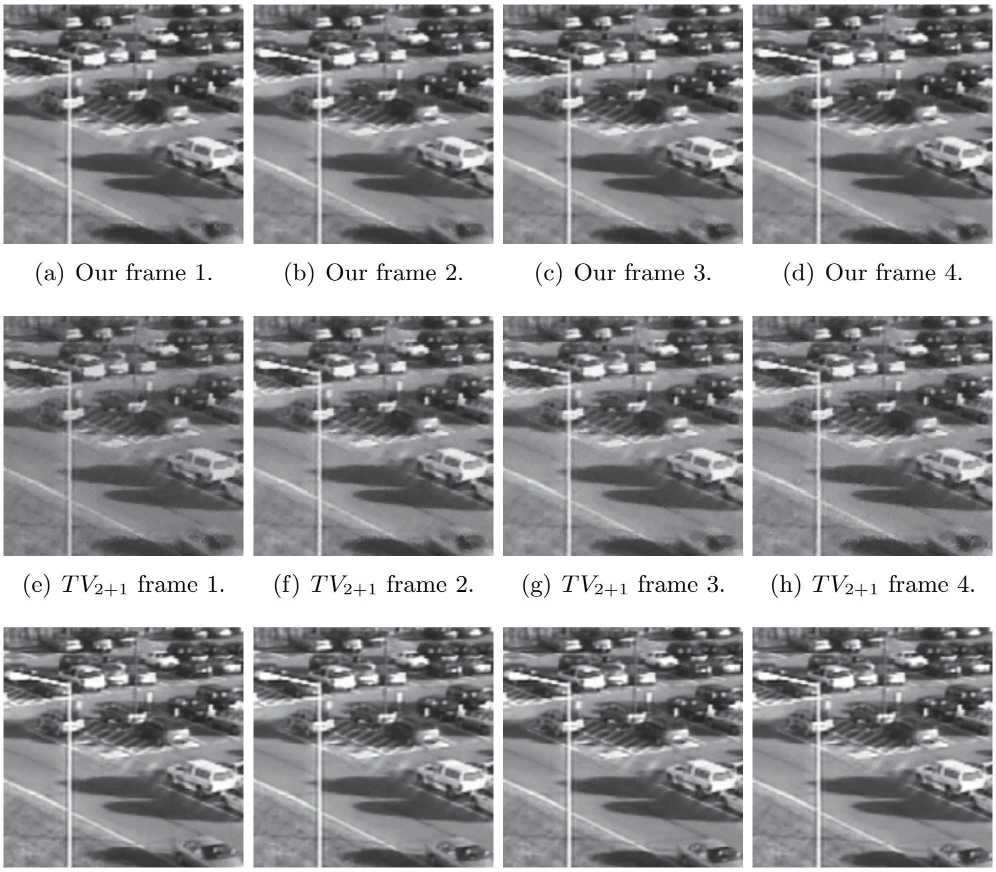

(a) Our difference. (b) TV_{2+1} difference.

Figure 7. *Comparison of the difference between the recovered frame and the original frames corresponding to the solutions in Figure* 6 (c) *and* (d).

we observe the reconstruction of a person walking. Our model is able to recover the shape of the person as well as the continuous linear structures appearing on the road, while the



(a) Our frame 1. (b) Our frame 2. (c) Our frame 3. (d) Our frame 4.

(e) TV_{2+1} frame 1. (f) TV_{2+1} frame 2. (g) TV_{2+1} frame 3. (h) TV_{2+1} frame 4.

(i) Original frame 1. (j) Original frame 2. (k) Original frame 3. (l) Original frame 4.

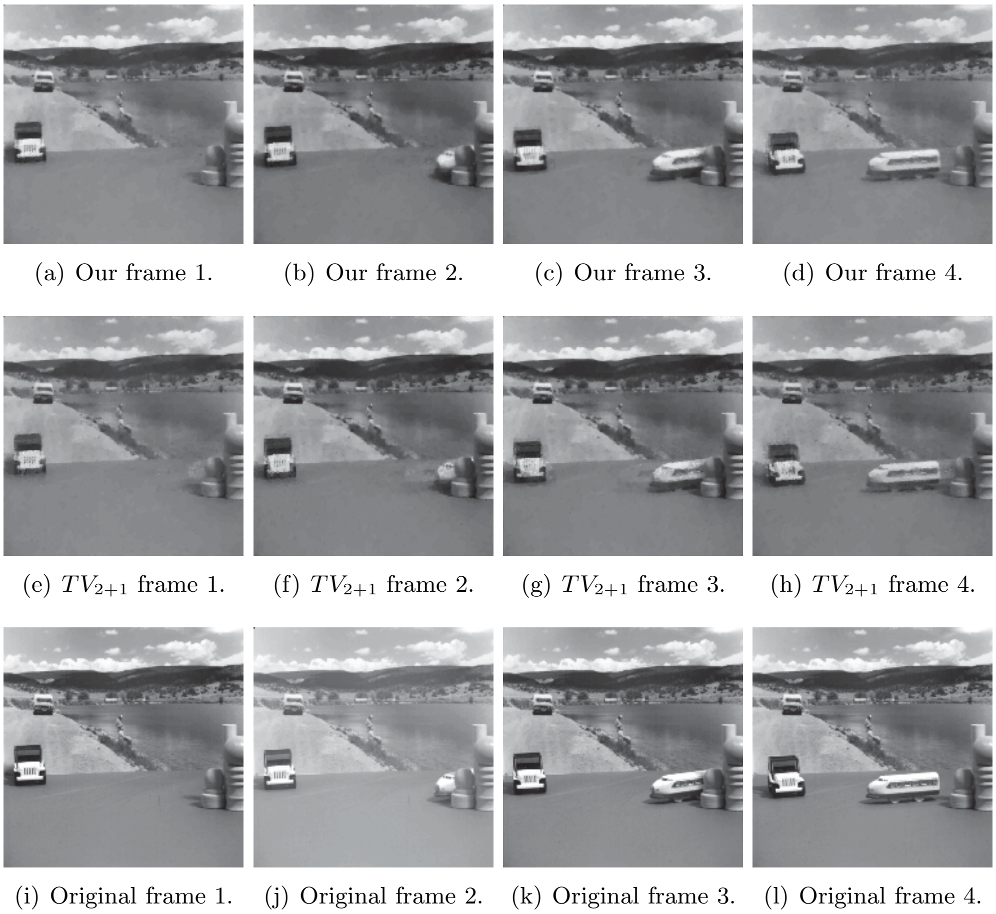
Figure 8. *Our algorithm applied to a parking lot sequence of four frames with moderate car and shadow motion in the bottom right corner. The average PSNR of our reconstruction is* 33.66. *The average PSNR of the* TV_{2+1} *reconstruction is* 31.32. *The parameters are set to* $\lambda_{ours} = 1.4$, $\lambda_{TV_{2+1}} = 1.2$, $\mu T = 0.5$. *The total number of iterations is fixed to a maximum of* 750. *The compression rate in space is* 35%, *and the standard deviation for the additive noise is* 5.

comparable model (TV_{2+1}) has difficulty. Although some objects in the video may be moving by only one pixel between frames, which is typically not considered to be well represented by the total variation seminorm, it seems that our model is still able to approximate the correct continuous behavior along the boundary of those moving objects.

5.5. Facial data.

For a challenging video, we test our method on a facial sequence which contains small movements and texture. In Figure 13, we observe that each frame consists of small scale details in the stationary and moving objects as well as subtle facial movements which rotate outside of the visual plane. The average PSNR for our method applied to this sequence is 32.19, and the average PSNR for the TV_{2+1} method applied to this sequence is 30.38. The smaller difference between the PSNRs is due in part to the difficulty both methods have in capturing all the details of the scenes.

A zoomed in comparison is provided in Figure 14. Visually, our method produces a more


(a) Our frame 1. (b) Our frame 2. (c) Our frame 3. (d) Our frame 4.

(e) TV_{2+1} frame 1. (f) TV_{2+1} frame 2. (g) TV_{2+1} frame 3. (h) TV_{2+1} frame 4.

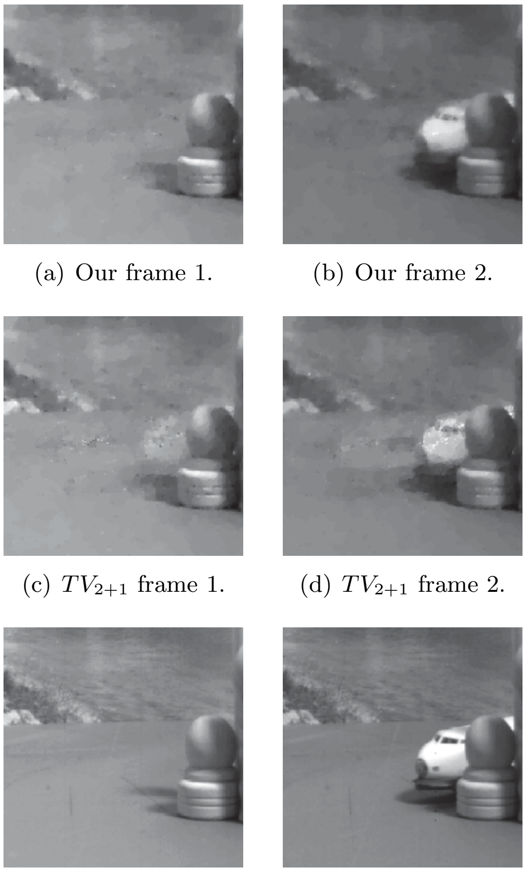
(i) Original frame 1. (j) Original frame 2. (k) Original frame 3. (l) Original frame 4.

Figure 9. *Our algorithm applied to a toy car sequence of four frames with three varying levels of motion. The average PSNR of our reconstruction is* 37.96*. The average PSNR of the* TV_{2+1} *reconstruction is* 35.77*. The parameters are set to* $\lambda_{ours} = 0.75$*,* $\lambda_{TV_{2+1}} = 0.4$*,* $\mu T = 0.25$*. The total number of iterations is fixed to a maximum of* 3000*. The compression rate in space is* 45%*, and the standard deviation for the additive noise is* 5*.*

realistic approximation. Our model contains smoother level lines on the face, better representing the contours in the data, while the TV_{2+1} reconstruction contains fuzzy boundaries.

5.6. Comparison of results. We compare our results to three models, the TV_{2+1} model, which we have used for visual comparisons in the previous sections, the $L^1(DCT)$ model, which uses $||DCT(u)||_{L^1}$ as a regularizer, and the TV_3 model implemented by the TVAL3 algorithm. In Table 1, we compare the methods using different sequences, different levels of noise, and various compression ratios. The sequences listed with the same name are from different parts of the given video. Also, by varying the compression ratio in time from 25% to 4%, we see that the PSNR values from our reconstruction vary less than those of the other two models, which shows that our model has a more stable performance.

In all cases listed here, our method has higher PSNR and produces more visually appealing results as compared to the other methods. In Figure 15, we plot the results from Table 1.



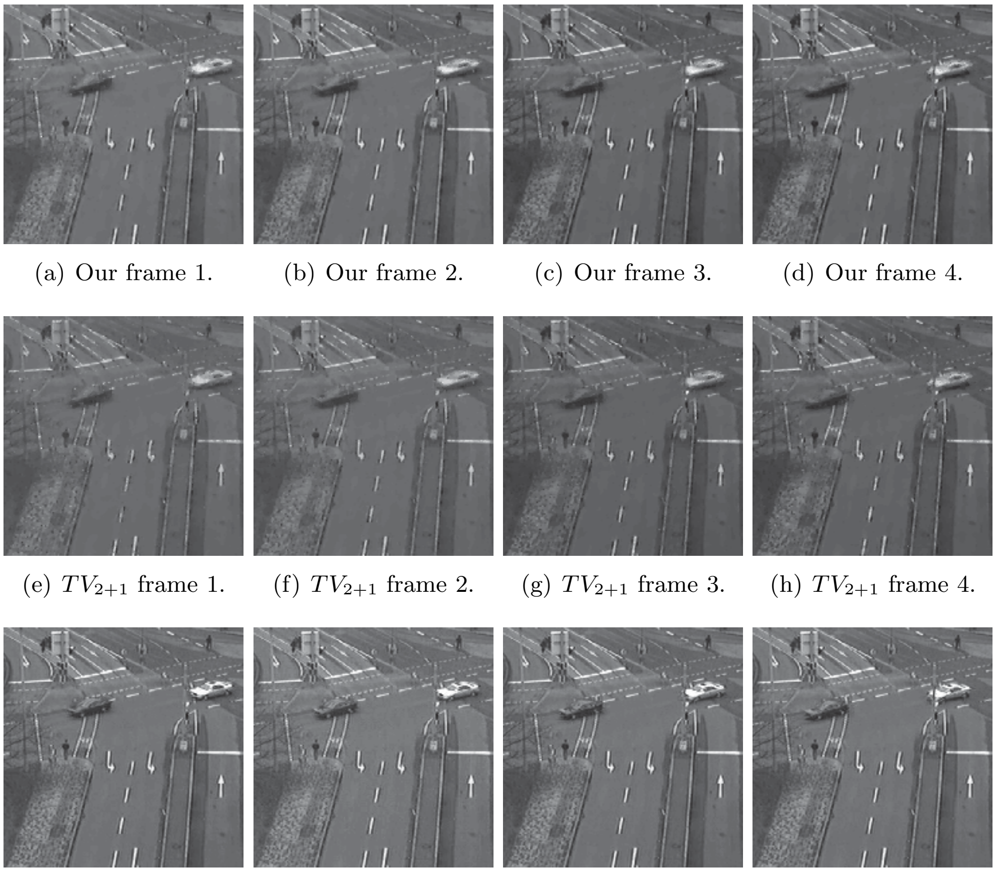
(a) Our frame 1. (b) Our frame 2.

(c) TV_{2+1} frame 1. (d) TV_{2+1} frame 2.

(e) Original frame 1. (f) Original frame 2.

Figure 10. *A zoomed in visual comparison between the reconstructions in Figure 9.*

The first column in Figure 15 corresponds to the first row in Table 1, the second column in Figure 15 corresponds to the tenth row in Table 1, and the third column in Figure 15 corresponds to the last row in Table 1. The TV_{2+1} and TV_3 both have the flickering effect seen in the previous examples, while the solutions generated by the $L^1(DCT)$ model suffers from global high-frequency oscillations. It should be noted that the TVAL3 method is not specifically tuned and could have results more comparable to the other methods if significantly altered, although this is not within the scope of this work.

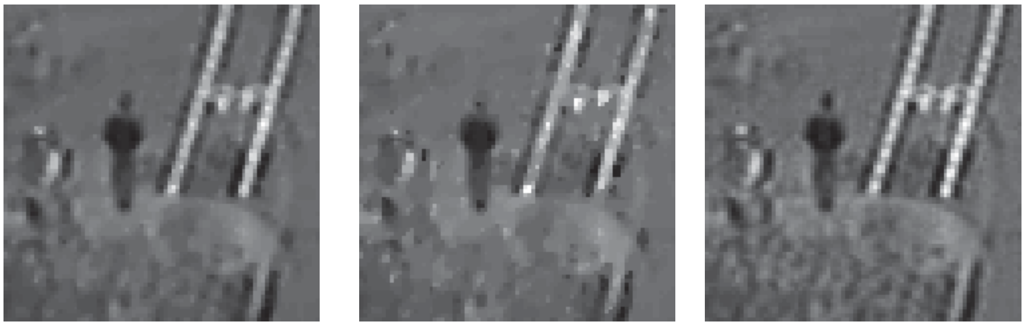


(a) Our frame 1. (b) Our frame 2. (c) Our frame 3. (d) Our frame 4.

(e) TV_{2+1} frame 1. (f) TV_{2+1} frame 2. (g) TV_{2+1} frame 3. (h) TV_{2+1} frame 4.

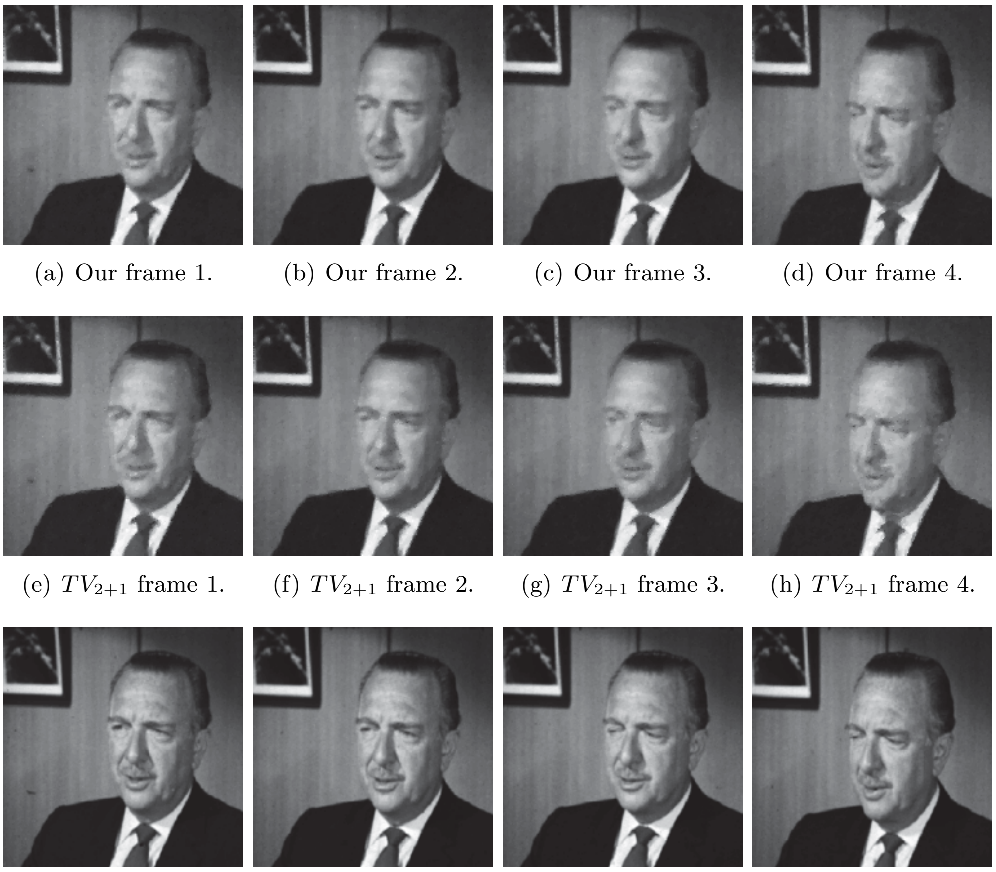
(i) Original frame 1. (j) Original frame 2. (k) Original frame 3. (l) Original frame 4.

Figure 11. *Our algorithm applied to a traffic sequence of five frames (four of which are shown here) with varying levels of motion. The average PSNR of our reconstruction is* 32.19. *The average PSNR of the* TV_{2+1} *reconstruction is* 29.66. *The parameters are set to* $\lambda_{ours} = 0.65$, $\lambda_{TV_{2+1}} = 0.2$, $\mu T = 1$. *The total number of iterations is fixed to a maximum of* 3500. *The compression rate in space is* 50%.



(a) Our frame 3. (b) TV_{2+1} frame 3. (c) Original frame 3.

Figure 12. *A zoomed in visual comparison between the solutions from Figure* 11.

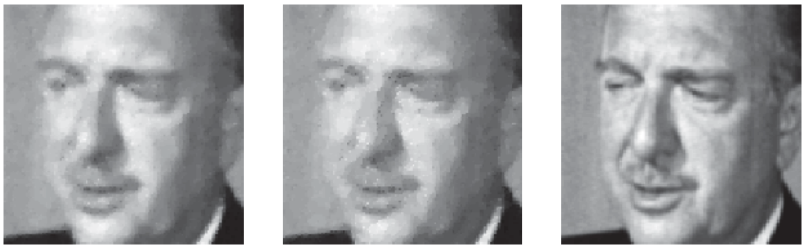


(a) Our frame 1. (b) Our frame 2. (c) Our frame 3. (d) Our frame 4.

(e) TV_{2+1} frame 1. (f) TV_{2+1} frame 2. (g) TV_{2+1} frame 3. (h) TV_{2+1} frame 4.

(i) Original frame 1. (j) Original frame 2. (k) Original frame 3. (l) Original frame 4.

Figure 13. *Our algorithm applied to a facial sequence of four frames. The average PSNR of our reconstruction is* 32.19. *The average PSNR of the* TV_{2+1} *reconstruction is* 30.38. *The parameters are set to* $\lambda_{ours} = 0.5$, $\lambda_{TV_{2+1}} = 0.35$, $\mu T = 1$. *The total number of iterations is fixed to a maximum of* 3500. *The compression rate in space is* 50%, *and the standard deviation for the additive noise is* 5.



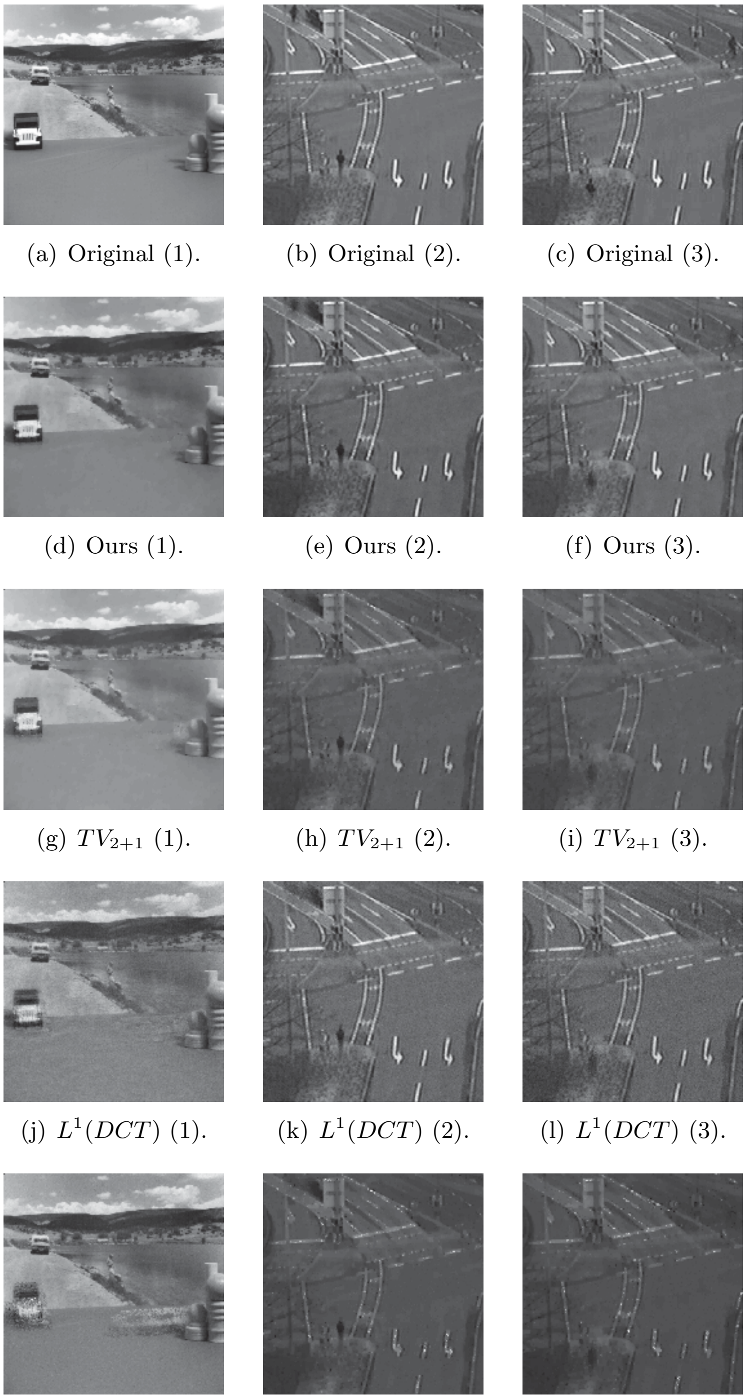
(a) Our frame 3. (b) TV_{2+1} frame 3. (c) Original frame 3.

Figure 14. *A zoomed in visual comparison between the solutions from Figure* 13.

Table 1

Comparing the PSNRs of our model, the TV_{2+1} model, the $L^1(DCT)$ model, and the TV_3 model implemented by TVAL3. The first column contains the name of the video used to generate the sequence, and the sequences differ between rows. The second column contains the standard deviation of the added noise. The third, fourth, and fifth columns contain the compression ratios. The sixth, seventh, eighth, and ninth columns contain the PSNR values for the four methods.

Sequence	Noise	Overall	Time	Space	Ours	TV_{2+1}	$L^1(DCT)$	TVAL3
Toy car	10	11.25%	25%	45%	37.96	35.74	33.24	30.98
Toy car	10	11.25%	25%	35%	35.48	33.63	29.54	29.59
Toy car	5	8.75%	25%	35%	37.19	35.10	33.49	30.91
Parking	5	7.00%	20%	35%	34.99	33.10	31.28	29.69
Parking	10	7.00%	20%	35%	32.12	30.69	27.57	28.57
Parking	10	9.00%	20%	45%	32.50	31.64	28.97	30.44
Parking	10	10.00%	20%	50%	32.50	31.91	29.50	30.48
Traffic	5	6.25%	25%	25%	29.74	27.25	27.74	24.67
Traffic	5	3.12%	12.5%	25%	31.58	29.07	30.27	27.40
Traffic	5	2.08%	8.3%	25%	33.51	28.90	31.06	27.04
Traffic	5	1.56%	6.25%	25%	33.59	28.69	29.60	26.85
Traffic	5	0.40%	4%	10%	31.51	24.37	24.49	23.62



(a) Original (1).
(b) Original (2).
(c) Original (3).
(d) Ours (1).
(e) Ours (2).
(f) Ours (3).
(g) TV_{2+1} (1).
(h) TV_{2+1} (2).
(i) TV_{2+1} (3).
(j) $L^1(DCT)$ (1).
(k) $L^1(DCT)$ (2).
(l) $L^1(DCT)$ (3).
(m) TV_3 (1).
(n) TV_3 (2).
(o) TV_3 (3).

Figure 15. *A visual comparison of the results found in Table 1. The first row,* (a)–(c)*, contains the original frames, while the remaining rows contain the solutions generated by each model. The first column corresponds to the first row in Table 1, the second column corresponds to the tenth row in Table 1, and the third column corresponds to the last row in Table 1.*

6. Conclusion. In this work, we introduce a convex regularizer for video recovery, which is made up of a total variation term on each frame and a total variation term on the difference between frames. The model can be easily solved using the primal-dual algorithm and has a simple parallel version. Based on visual comparisons and PSNR values, the model outperforms other popular models. Since our model is developed for sequences of compressive frames with randomly generated masks, it may be applicable to other compressive sensing paradigms—for example, compressive hyperspectral imaging.

Acknowledgments. The authors would like to thank Wotao Yin and Tom Goldstein for their useful discussions. The authors would also like to thank Lawrence Carin, Guillermo Sapiro, Giang Tran, Jianbo Yang, Xin Yuan, and the anonymous reviewers for their helpful discussions and comments.

REFERENCES

[1] A. Agrawal and R. Raskar, *Resolving objects at higher resolution from a single motion-blurred image*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Washington, DC, 2007, pp. 1–8.

[2] J. M. Bioucas-Dias and M. A. T. Figueiredo, *A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration*, Proc. IEEE Trans. Image Process., 16 (2007), pp. 2992–3004.

[3] K. Bredies, K. Kunisch, and T. Pock, *Total generalized variation*, SIAM J. Imaging Sci., 3 (2010), pp. 492–526.

[4] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging Vision, 40 (2011), pp. 120–145.

[5] W. L. Chan, M. L. Moravec, R. G. Baraniuk, and D. M. Mittleman, *Terahertz imaging with compressed sensing and phase retrieval*, Optics Lett., 33 (2008), pp. 974–976.

[6] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F Kelly, and R. G. Baraniuk, *Single-pixel imaging via compressive sampling*, IEEE Signal Process. Mag., 25 (2008), pp. 83–91.

[7] E. Esser, X. Zhang, and T. F. Chan, *A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science*, SIAM J. Imaging Sci., 3 (2010), pp. 1015–1046.

[8] R. Fergus, A. Torralba, and W. T. Freeman, *Random Lens Imaging*, available online from http://hdl.handle.net/1721.1/33962, 2006.

[9] D. J. Le Gall, *The MPEG video compression algorithm*, Signal Process. Image Commun., 4 (1992), pp. 129–140.

[10] D. Le Gall, *MPEG: A video compression standard for multimedia applications*, Commun. ACM, 34 (1991), pp. 46–58.

[11] G. Gilboa and S. Osher, *Nonlocal operators with applications to image processing*, Multiscale Model. Simul., 7 (2008), pp. 1005–1028.

[12] J. Gilles and Y. Meyer, *Properties of BV- G structures + textures decomposition models. Application to road detection in satellite images*, IEEE Trans. Image Process., 19 (2010), pp. 2793–2800.

[13] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM J. Imaging Sci., 2 (2009), pp. 323–343.

[14] T. Goldstein, L. Xu, K. F. Kelly, and R. Baraniuk, *The STONE Transform: Multi-Resolution Image Enhancement and Real-Time Compressive Video*, arXiv preprint arXiv:1311.3405, http://arxiv.org/abs/1311.3405, 2013.

[15] T. Goldstein, E. Esser, and R. Baraniuk, *Adaptive Primal-Dual Hybrid Gradient Methods for Saddle-Point Problems*, arXiv preprint arXiv:1305.0546, http://arxiv.org/abs/1305.0546, 2013.

[16] J. Gu, S. Nayar, E. Grinspun, P. Belhumeur, and R. Ramamoorthi, *Compressive structured light for recovering inhomogeneous participating media*, in Proceedings of Computer Vision–ECCV 2008, Marseille, France, 2008, pp. 845–858.

[17] W. Guo and W. Yin, *EdgeCS: Edge guided compressive sensing reconstruction*, in Proceedings of Visual Communication and Image Processing, SPIE Proc. 7744, SPIE, Bellingham, WA, 2010, 77440L.

[18] W. Guo, J. Qin, and W. Yin, *A New Detail-Preserving Regularity Scheme*, UCLA CAM Technical report 13-04, University of California Los Angeles, Los Angeles, CA, 2013.

[19] P. Hasler and D. V. Anderson, *Cooperative analog-digital signal processing*, in Proceedings of the 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Volume 4, IEEE, Washington, DC, 2002, pp. IV-3972–IV-3975.

[20] C. Li, W. Yin, and Y. Zhang, *User's Guide for TVAL3: TV Minimization by Augmented Lagrangian and Alternating Direction Algorithms*, CAAM Report, Rice University, Houston, TX, 2009.

[21] C. Li, W. Yin, H. Jiang, and Y. Zhang, *An efficient augmented Lagrangian method with applications to total variation minimization*, Comput. Optim. Appl., 56 (2013), pp. 507–530.

[22] P. Llull, X. Liao, X. Yuan, J. Yang, D. Kittle, L. Carin, G. Sapiro, and D. J. Brady, *Coded Aperture Compressive Temporal Imaging*, arXiv preprint arXiv:1302.2575, http://arxiv.org/abs/1302.2575, 2013.

[23] M. Lustig, D. Donoho, and J. M. Pauly, *Sparse MRI: The application of compressed sensing for rapid MR imaging*, Magnetic Resonance in Medicine, 58 (2007), pp. 1182–1195.

[24] S. Marchesini, *Ab Initio Compressive Phase Retrieval*, arXiv preprint arXiv:0809.2006, http://arxiv.org/abs/0809.2006, 2008.

[25] Y. Meyer, *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations: The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures*, Univ. Lecture Ser. 22, AMS, Providence, RI, 2001.

[26] D. Needell and R. Ward, *Stable image reconstruction using total variation minimization*, SIAM J. Imaging Sci., 6 (2013), pp. 1035–1058.

[27] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An iterative regularization method for total variation-based image restoration*, Multiscale Model. Simul., 4 (2005), pp. 460–489.

[28] A. Poglitsch, C. Waelkens, N. Geis, H. Feuchtgruber, B. Vandenbussche, L. Rodriguez, O. Krause, E. Renotte, C. Van Hoof, P. Saraceno, et al., *The photodetector array camera and spectrometer (PACS) on the Herschel space observatory*, Astronomy Astrophys., 518 (2010), L2.

[29] R. Raskar, A. Agrawal, and J. Tumblin, *Coded exposure photography: Motion deblurring using fluttered shutter*, ACM Trans. Graphics, 25 (2006), pp. 795–804.

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

[31] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.

[32] H. Schaeffer and S. Osher, *A low patch-rank interpretation of texture*, SIAM J. Imaging Sci., 6 (2013), pp. 226–262.

[33] H. Schaeffer, Y. Yang, and S. Osher, *Real-Time Adaptive Video Compressive Sensing*, UCLA CAM Tech. Report, University of California Los Angeles, Los Angeles, CA, 2013.

[34] Y. Tendero, J.-M. Morel, and B. Rougé, *The flutter shutter paradox*, SIAM J. Imaging Sci., 6 (2013), pp. 813–847.

[35] A. Wagadarikar, R. John, R. Willett, and D. Brady, *Single disperser design for coded aperture snapshot spectral imaging*, Appl. Optics, 47 (2008), pp. B44–B51.

[36] A. A. Wagadarikar, N. P. Pitsianis, X. Sun, and D. J. Brady, *Video rate spectral imaging using a coded aperture snapshot spectral imager*, Opt. Express, 17 (2009), pp. 6368–6388.

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

[38] Y. Wang, J. Yang, W. Yin, and Y. Zhang, *A new alternating minimization algorithm for total variation image reconstruction*, SIAM J. Imaging Sci., 1 (2008), pp. 248–272.

[39] J. Yang, X. Yuan, X. Liao, P. Llull, G. Sapiro, D. J. Brady, and L. Carin, *Gaussian mixture model for video compressive sensing*, in Proceedings of the IEEE International Conference on Image Processing, IEEE, Washington, DC, 2013, pp. 19–23.

[40] J. Yang, X. Yuan, X. Liao, P. Llull, D. J. Brady, G. Sapiro, and L. Carin, *Video compressive sensing using Gaussian mixture models* , IEEE Trans. Image Process., 23 (2014), pp. 4863–4878.

[41] Y. Yang, H. Schaeffer, W. Yin, and S. Osher, *Mixing space-time derivatives for video compressive sensing*, in Proceedings of the 2013 Asilomar Conference on Signals, Systems and Computers, IEEE, Washington, DC, 2013, pp. 158–162.

[42] X. Yuan, J. Yang, P. Llull, X. Liao, G. Sapiro, D. J. Brady, and L. Carin, *Adaptive Temporal Compressive Sensing for Video*, arXiv preprint arXiv:1302.3446, http://arxiv.org/abs/1302.3446, 2013.

[43] J. Zheng and E. L. Jacobs, *Video compressive sensing using spatial domain sparsity*, Optical Engrg., 48 (2009), 087006.

[44] M. Zhu and T. Chan, *An Efficient Primal-Dual Hybrid Gradient Algorithm for Total Variation Image Restoration*, UCLA CAM Tech. Report 08-34, University of California Los Angeles, Los Angeles, CA, 2008.