Mixing space-time derivatives for video compressive sensing

Yi Yang*, Hayden Schaeffer*, Wotao Yin†, Stanley Osher‡

*Department of Mathematics, University of California, Los Angeles, CA 90095 USA
†Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005 USA
‡Level Set Systems, Pacific Palisades, CA 90272 USA

Abstract—With the increasing use of compressive sensing techniques for better data acquisition and storage, the need for efficient, accurate, and robust reconstruction algorithms continues to be in demand. In this work we present a fast total variation based method for reconstructing video compressive sensing data. Video compressive sensing systems store video sequences by taking a linear combination of consecutive spatially compressed frames. In order to recover the original data, our method regularizes both the spatial and temporal components using a total variation semi-norm that mixes information between dimensions. This mixing provides a more consistent approximation of the connection between neighboring frames with little to no increase in complexity. The algorithm is easy to implement since each iteration contains two shrinkage steps and a few iterations of conjugate gradient. Numerical simulations on real data show large improvements in both the PSNR and visual quality of the reconstructed frame sequences using our method.

I. INTRODUCTION

Compressive sensing (CS) techniques have a large range of applications from hardware development using 1 bit compressive sensing [1], [2] or coded apertures [3], [4] to methodologies using CS sampling methods [5]. Its popularity stems for the efficient way of handling large data sets - a growing issue in all fields and applications. For example in information science, large collections of images and videos are obtained at extremely high rates, thereby efficient transmission and storage accompanied by robust and accurate reconstruction is necessary. One recent device, the coded aperture compressive temporal imaging (CACTI) [6], shows promise on encoding optical data entering at rates approaching 1 exapixel/second. In fact, an emerging application of the CS paradigm, called video compressive sensing (VCS), has enabled high quality reconstruction of videos from very few observed frames.

In terms of the mechanics, VCS methods use physical techniques to code incoming optical data in order to compress either spatial or spectral information. For spatial compression, this is typically accomplished by using a mechanical grating (the coded aperture) to block the entering data stream, thus subsampling the incoming signal. There are several ways to compress the temporal data [7], [8], for this work we consider the methods employed in the compressive sensing framework. Here, we assume that the final stored measurement is defined as a linear combination of several spatially compressed frames, thereby removing both the spatial and temporal redundancy [9], [10].

The underlying forward model we consider is as follows: let $X$ be the compressed data set, $F = [F_1, \ldots, F_T]$ be the vector formed by all the original frames, where $T$ is the number of frames that are compressed into one. Define $A$ as the CS matrix containing the random frame by frame masks as well as the temporal compression. In other words, $A$ can be written as $[A_1, \ldots, A_T]$ with each $A_i$ being a random binary matrix, representing which pixels can store information. $X$ is acquired by taking the sum of $A_i F_i$ for all the $i$’s with the product understood as the Hadamard product. For simplicity, here we denote this compression process as $X = AF$. The CS mask can be generated in two ways, depending on the application. The first is to independently generate $A_i$ for each frame, a common technique in compressive sensing. The second way is to generate an initial mask $A_1$ for the first frame and shift it by one pixel (in a known direction) for each subsequent frame. In hardware, this amounts to translating the coded aperture between frames, which is easy to implement, requires less storage, and usually yields similar results as first way [6]. Fig. 1 illustrates the compression process for our problem.

Similar to other compressive sensing problems, the linear system $X = AF$ is underdetermined since there are much fewer measurements than unknown variables. In this way, we can greatly reduce the amount of information stored while still being able to recover the original signal with high accuracy. Our goal is to reconstruct $F$ given $A$ and $X$ by leveraging certain desired sparsity properties on the original video. For image decompression from CS measurements, recent research supports the use of total variation (TV) regularization because of its ability to preserve edge information. With the introduction of the Bregman iteration [11] and the split Bregman algorithm (related to the alternating direction method of multipliers (ADMM) algorithm) [12], the TV-based optimization problems can be solved in an extremely efficient way with sharper contrast in the recovered image. This is of particular importance.

Fig. 1. Illustration of the compression/encoding process of the video data.
to video reconstruction, since the temporal compression causes motion blur, adding to the overall difficulty of solving the inverse problem.

Although the application of TV to imaging problems is clear, the extra dimension in videos makes extensions more ambiguous. The most direct way is to incorporate the temporal dimension as a third coordinate, which leads to the following optimization problem (TV3):

$$\min_F \sum_{i=1}^{T} |F_i|_{TV^3}, \text{ s.t. } \sum_{i=1}^{T} A_i F_i = X$$

(1)

where the 3d TV semi-norm $\cdot |_{TV^3}$ is defined as

$$|Y|_{TV^3} := |D_3 Y|_1 = \sum_{i,j,k} |D_3 Y(i,j,k)|$$

(2)

with the difference operator defined as $D_3 := [D_x,D_y,D_t]$ and the vector norm $|D_3 Y(i,j,k)| := \sqrt{(D_3 Y(i,j,k))_x^2 + (D_3 Y(i,j,k))_y^2 + (D_3 Y(i,j,k))_t^2}$. The TV3 model is used by TwIST (two-step iterative shrinkage/thresholding) [13] and TVAL3 (TV minimization by augmented Lagrangian and alternating direction algorithm) [14]. The above regularizer places equal importance in all directions and treats the pixel values as piecewise constant in time. The underlying assumption is that the movement between successive frames is small (sparse). Though this model achieves reasonable performance in application (with respect to PSNR, i.e. Peak Signal-to-Noise Ratio), we can often greatly improve both the PSNR and visual quality of the reconstruction by carefully exploring the temporal component. More examination of this model can be found in Section III.

Rather than treating the difference between successive frames as sparse, we assume that the difference is piecewise constant. This assumption is consistent with the spatial regularizer, since the typical hypothesis is that each frame in the video can be approximated by a piecewise constant image, then the difference image between two successive frames should also belong to that category. As a matter of fact, this is consistent with the dynamics captured in images: a piecewise constant object travels with a certain speed across the field of view. Therefore, instead of considering the $L^1$ norm of the difference image, here we propose the TV norm on the frame difference. As seen in the experimental results presented in Section III, our model provides better quality results regardless of the amount of motion present in the video.

This paper is organized as follows. Section II gives a detailed description of our model and the associated algorithm. In Section III, numerical simulations on real data are provided, which demonstrate the improvement in both numerical and visual quality of the reconstructed video under various circumstances. We also compare our model with several total variation based methods. Lastly, we conclude with a short discussion in Section IV.

II. MODEL AND ALGORITHM DESCRIPTION

The mathematical formulation of our model is as follows:

$$\min_F \sum_{i=1}^{T} |F_i|_{TV} + \lambda \sum_{i=1}^{T-1} |F_{i+1} - F_i|_{TV}, \text{ s.t. } \sum_{i=1}^{T} A_i F_i = X$$

(3)

where $|F_i|_{TV} = |DF_i|_1$ with $D = [D_x, D_y]$. The model essentially recovers the original data via deblurring the temporal component, reconstructing the compressed spatial component, as well as interpolating the missing frames.

The first term is the 2d TV semi-norm and the second term is the total variation of the difference between frames. The parameter $\lambda$ balances the importance between the two regularizers. In practice $\lambda$ has a direct relationship to $T$: when $T$ is large (small), a larger (smaller) $\lambda$ yields better results. This is related to the compression methodology, since large $T$ is used when successive frames are similar, while small $T$ is used for highly dynamic frames.

Since the model contains $L^1$ type regularizers, we can solve the minimization efficiently by applying the split Bregman technique. The convergence of this algorithm to the minimizer of (3) is guaranteed according to [15]. Introducing auxiliary variables $P_i$ for $i = 1, \ldots, T$, and $Q_i$ for $i = 1, \ldots, T-1$ to substitute for the spatial derivative and the mixed gradient respectively, the above problem is equivalent to

$$\min_{F,P,Q} \sum_{i=1}^{T} |P_i|_1 + \lambda \sum_{i=1}^{T-1} |Q_i|_1,$$

(4)

s.t. $P_i = DF_i$, $Q_i = DF_{i+1} - DF_i$, $\sum_{i=1}^{T} A_i F_i = X$

Let us define $\hat{D} F_i = DF_{i+1} - DF_i$, which will stand for the gradient of the time differences. Next adding back the constraint yields:

$$(F^k, P^k, Q^k) = \text{argmin}_{F,P,Q} \sum_{i=1}^{T} |P_i|_1 + \lambda \sum_{i=1}^{T-1} |Q_i|_1$$

(5)

$$+ \frac{\mu_1}{2} \sum_{i=1}^{T} \|A_i F_i - X + X^{k-1}\|^2$$

(6)

$$+ \frac{\mu_2}{2} \sum_{i=1}^{T-1} \|P_i - \hat{D} F_i + B^{k-1}_i\|^2$$

(7)

$$+ \frac{\mu_3}{2} \sum_{i=1}^{T-1} \|Q_i - \hat{D} F_i + b^{k-1}_i\|^2$$

(8)

$$X^k = \sum_{i=1}^{T} A_i F_i - X + X^{k-1}$$

(9)

$$B^{k}_i = P^{k}_i - \hat{D} F^{k}_i + B^{k-1}_i$$

(10)

$$b^k_i = Q^k_i - \hat{D} F^k_i + b^{k-1}_i$$

(11)

where the variables $X^k$, $B^k$ and $b^k$ are the Bregman variables which are used to enforce the equality constraint. Although the choice of $\mu_1$ will not affect the final solution, their values relate to the convergence rate. Also, in practice we can simply set $\mu_2 = \mu_3$ since they both control the gradient terms.
update for the variables $P, Q$ and $F$ are done in an alternating fashion by the equations below:

$$P^k_i = \text{shrink}(DF_i^{k-1} - b_i^{k-1}, 1/\mu_2)$$  \hspace{1cm} (12)  
$$Q^k_i = \text{shrink}(\hat{DF}_i^{k-1} - b_i^{k-1}, \lambda/\mu_3)$$  \hspace{1cm} (13)  
$$F^k = \arg\min_F \frac{\mu_1}{2} \sum_{i=1}^T A_i F_i - X + X^{k-1} \|2^2$$  \hspace{1cm} (14)  
$$+ \frac{\mu_2}{2} \sum_{i=1}^T \|P_i - DF_i + B_i^{k-1}\|2^2$$  \hspace{1cm} (15)  
$$+ \frac{\mu_3}{2} \sum_{i=1}^{T-1} \|Q_i - \hat{DF}_i + b_i^{k-1}\|2^2$$  \hspace{1cm} (16)

where the shrink function above is defined pointwise for a 2d vector $Y$ as $\text{shrink}(Y, \tau) := \max(\|Y\| - \tau, 0) \frac{Y}{\|Y\|}$. For the $F$ variable update, since the associated subproblem is differentiable, taking the first derivative and rearranging terms leads to

$$(\mu_1 A^* A + \mu_2 D^* D + \mu_3 \hat{D}^* \hat{D}) F = \text{RHS}$$ \hspace{1cm} (17)

where $(\cdot)^*$ stands for the adjoint of the operator and $\text{RHS} = \mu_1 A^* (X - X^{k-1}) + \mu_2 D^*(P^k + B^{k-1}) + \mu_3 \hat{D}^*(Q^k + \hat{b}^k)$. In each iteration of split Bregman algorithm, (17) only needs to be approximately solved with a few steps of conjugate gradient. Faster convergence can be obtained by using a preconditioner.

Altogether the method reduces to two explicit shrinkage steps and solving a linear system for each iteration, making the algorithm fast and easy to implement.

III. NUMERICAL EXPERIMENTS AND DISCUSSION

In this section we use various numerical experiments to illustrate how our model and the associated algorithm can improve the reconstruction results compared with TV3. In order to show the importance of temporal consistency in video reconstruction, we also include the following model (TV2) in our comparison:

$$\min_F \frac{T}{2} \sum_{i=1}^T \|F_i\|_{TV}, \text{ s.t. } \sum_{i=1}^T A_i F_i = X$$ \hspace{1cm} (18)

where the regularizer is decoupled. The TV2 model is able to reconstruct a reasonable result since the CS mask is random. Numerically, (18) is solved via the split Bregman method, similar to the algorithm described in Section II. Our method converges in under a minute for all the experiments described below.

As discussed in [6], shifting the CS masks usually leads to reconstruction results that are comparable with generating a random masks for each frame. Hence we simulate this type of mask generation during our experiments. This is done by first generate a random binary mask with a certain amount of zero elements, and translating the mask in a fixed direction after each frame. We also provide an experiment using random mask, which are generated for each frame independently. In all cases, the parameters are chosen to maximize the PSNR values.

In the first experiment (Setting 1), 4 frames are compressed into one – each with 50% spatial downsampling. According to the first row of results displayed in Table I, we can see that our model achieves a much higher PSNR value compared with the results obtained from TV2 and TV3. The model without temporal regularizers provides the worst results among all the candidates with respect to the PSNR values. This justifies the importance of leveraging the relationship between neighboring frames. Note that all the methods yield a relatively good PSNR. Since the region of motion is relatively small compared to the overall size of the image, one could poorly recover the dynamic object and still have a good PSNR.

Next, in Setting 2 we keep the same conditions as in Setting 1 but input a more dynamic video (see Fig. 2). From Table I, we see that our model yields higher PSNRs than the other models. Although all models give a relatively high PSNR, as seen in Fig. 2 there are clear qualitative differences. Our reconstruction has less point structures commonly associated with poor CS reconstruction. It is also able to recover the details of the background and capture the moving structures. The TV2 model provides a piecewise constant reconstruction of each frame, but it fails to recover the fine scale details found in the background. The TV3 model is able to reconstruct the background well and some of the textures, but is unable to resolve the moving object. From this example, it is clear that the temporal regularizer helps promote texture reconstruction.

In Setting 3, we increase the $T$ value to 8 while keep all other conditions the same. The PSNR gap between our model and the other two grows, showing that our model is more robust to larger compression rates. Fig. 3 provides a visual comparison between our model, TV2 and TV3. Fig. 3 (a) shows an original frame from the video sequence and (b) shows the compressed frame $X$. Comparing the reconstructions (c-e), the zoomed in images (f-h) and difference images (i-k), we can easily see that our model gives a smoother reconstruction with sharper contrast. The loss of contrast in (d) is primarily due to the lack of temporal consistency. The point structure effect in (e) is caused by the inaccuracy of $L^1$ regularization for large motion. In the difference images (i-k), it is clear that our model produces the least amount of errors in the background component while recovers a piecewise smooth dynamic foreground. The PSNR gain of our model for this particular frame is more than 3.

To examine the effect of both the random mask simulation and the sampling ratio, in the forth setting we generate an independent random mask for each frame and lowered the sampling ratio to 45%. The temporal compression rate, $T$, is set at 8 for this case. We can see that all the PSNRs have decreased as expected, while the values from our model are much higher than the others. The PSNRs for TV2 are robust in the sense that they do not change dramatically between the settings; however, they are consistently lower than the ones from other algorithms. Lastly, when $T = 12$ and the spatial compression ratio returns to 50%, again our model provides the best PSNR values among all the models.

In general, adding two regularizers for the same variable does not necessarily increase the quality of the reconstruction. In our model, since the regularizers act on the different coordinates, both are necessary. In Fig. 4, we compare our
### TABLE I. MEAN AND MAXIMUM PSNR COMPARISON BETWEEN DIFFERENT VIDEO RECONSTRUCTION MODELS

<table>
<thead>
<tr>
<th>Setting</th>
<th>Our model</th>
<th>TV2</th>
<th>TV3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean PSNR</td>
<td>max PSNR</td>
<td>mean PSNR</td>
</tr>
<tr>
<td>Setting 1</td>
<td>38.6716</td>
<td>39.3424</td>
<td>32.3084</td>
</tr>
<tr>
<td>Setting 2</td>
<td>34.4394</td>
<td>34.9082</td>
<td>30.8161</td>
</tr>
<tr>
<td>Setting 3</td>
<td>36.6830</td>
<td>37.4991</td>
<td>30.0898</td>
</tr>
<tr>
<td>Setting 4</td>
<td>33.4960</td>
<td>34.1152</td>
<td>28.8421</td>
</tr>
<tr>
<td>Setting 5</td>
<td>33.1679</td>
<td>33.6097</td>
<td>27.8260</td>
</tr>
</tbody>
</table>

![Fig. 2](image1.png)

**Fig. 2.** Figure illustration for Setting 2. The PSNR of this particular reconstruction with our model is 34.9082, while TV2 gives 31.1694 and TV3 gives 32.3697.

![Fig. 3](image2.png)

**Fig. 3.** Figure illustration for Setting 3. The PSNR of this particular reconstruction with our model is 35.4635, compared with 29.7760 for TV2 and 32.4518 for TV3.

Our model shows dramatic improvement using a more consistent regularizer. The main function of our temporal regularizer is to better capture both fast moving objects and temporal relationships. Numerical simulations show large improvements in both the PSNR and visual quality of the reconstructions using our method, regardless of the type of dynamics present in the video. In this work, the problem we are considering derives from VCS; however, the method presented here can easily be applied to other compression types.

IV. CONCLUSION

Our model shows dramatic improvement using a more consistent regularizer. The main function of our temporal regularizer is to better capture both fast moving objects and temporal relationships. Numerical simulations show large improvements in both the PSNR and visual quality of the reconstructions using our method, regardless of the type of dynamics present in the video. In this work, the problem we are considering derives from VCS; however, the method presented here can easily be applied to other compression types.
ACKNOWLEDGMENT

H. Schaeffer is supported by the Department of Defense (DoD) through the National Defense Science and Engineering Graduate Fellowship (NDSEG). Y. Yang is supported by NSF DMS 0835863, NSF DMS 0914561, and ONR N00014-11-0749. W. Yin is supported by NSF grants DMS-0748839, and ECCS-1028790, and ARO/ARL MURI grant FA9550-10-1-0567. S. Osher is supported by DARPA KeCom.

REFERENCES


