Video from a Single Coded Exposure Photograph using a Learned Over-Complete Dictionary

Yasunobu Hitomi[1], Jinwei Gu[2], Mohit Gupta[3], Tomoo Mitsunaga[1], Shree K. Nayar[3]
[1]Sony Corporation, [2]Rochester Institute of Technology, [3]Columbia University
{Yasunobu.Hitomi, Tomoo.Mitsunaga}@jp.sony.com, jwgu@cis.rit.edu, {mohitg, nayar}@cs.columbia.edu

Abstract

Cameras face a fundamental tradeoff between the spatial and temporal resolution – digital still cameras can capture images with high spatial resolution, but most high-speed video cameras suffer from low spatial resolution. It is hard to overcome this tradeoff without incurring a significant increase in hardware costs. In this paper, we propose techniques for sampling, representing and reconstructing the space-time volume in order to overcome this tradeoff. Our approach has two important distinctions compared to previous works: (1) we achieve sparse representation of videos by learning an over-complete dictionary on video patches, and (2) we adhere to practical constraints on sampling scheme which is imposed by architectures of present image sensor devices. Consequently, our sampling scheme can be implemented on image sensors by making a straightforward modification to the control unit. To demonstrate the power of our approach, we have implemented a prototype imaging system with per-pixel coded exposure control using a liquid crystal on silicon (LCoS) device. Using both simulations and experiments on a wide range of scenes, we show that our method can effectively reconstruct a video from a single image maintaining high spatial resolution.

1. Introduction

Digital cameras face a fundamental tradeoff between temporal and spatial resolution. As the frame rate increases, spatial resolution decreases. This limitation is due to hardware factors such as readout and analog-to-digital (AD) conversion time of sensors. Although it is possible to increase the throughput by introducing parallel AD convertors and frame buffers [1], this requires more transistors per pixel, thus lowering the fill-factor and increasing cost. As a compromise, many current camera manufacturers implement a "thin-out" mode (*i.e.*, high speed draft mode), which directly trades off the spatial resolution for higher temporal resolution and often degrades image quality, as shown in Figure 1.

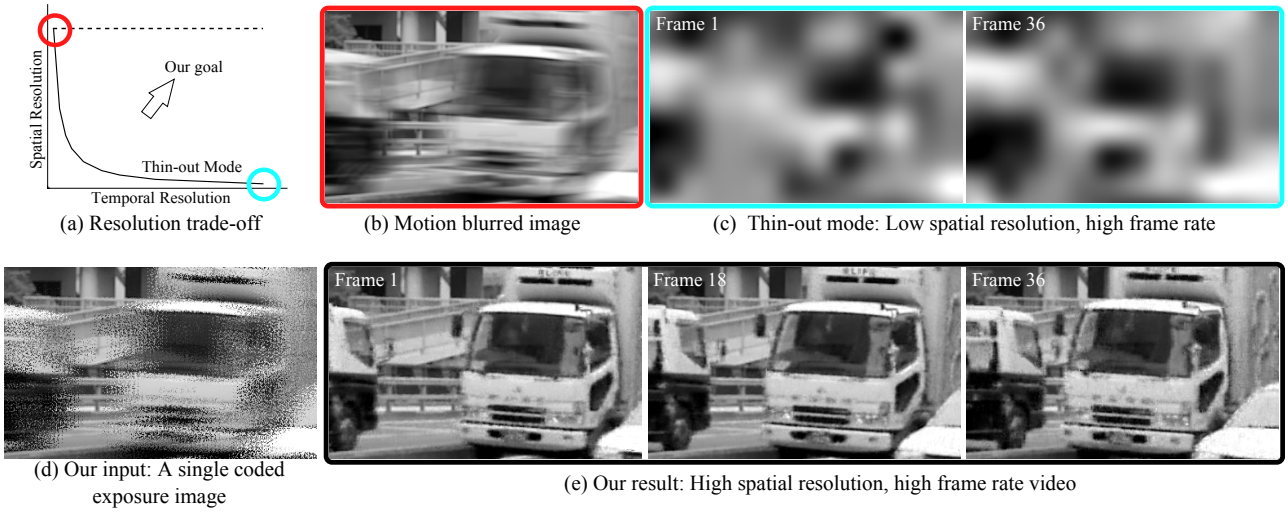
The goal of our work is to design an imaging system that can capture videos with both high spatial and temporal resolutions. In order to overcome the fundamental resolution tradeoff, such a system must exploit the sparsity of natural videos. Recent progress in the field of compressive sensing has provided a general framework for efficient capture of sparse signals [2, 3]. In case of capturing videos, however, there are specific challenges that must be addressed. In particular, the sampling function is limited by the hardware restrictions of image sensors. On the positive side, the structure of video signals can be used to build efficient representations which can increase the measurement efficiency. A practical video capture system must take into account such domain specific issues in the way it samples, represents, and reconstructs videos. In this paper, we focus on two problems: 1) sampling, and 2) representation of space-time volumes for designing practical compressive video acquisition systems.

First, how to **sample** space-time volumes while accounting for the restrictions imposed by imaging hardware? For the maximum flexibility in designing sampling schemes, it is important to have pixel-wise exposure control. Fortunately, most CMOS image sensors can provide pixel-wise controllability by making a straightforward modification to the control unit (Figure 2(a)). However, pixels on most CMOS image sensors allow only one continuous exposure during one camera integration time (Figure 2(b))[1]. This is because the pixels are "reset" at the end of each exposure *bump*. Multiple bumps during a single integration time would require the electrons to be stored, which in turn requires expensive per-pixel frame buffers. Next, the sampling function should ensure that the captured intensities are within the sensor's dynamic range. Finally, the sampling function is restricted to binary values (0 or 1)[2]. In this paper, we design sampling functions while adhering to these restrictions. Thus, it is feasible to implement our techniques on real sensors.

Second, how to efficiently **represent** space-time volumes for sparse reconstruction? General analytical transforms, such as discrete cosine transform (DCT) and wavelets often do not provide the desired level of

[1]CCD image sensors allow multiple bumps within a single integration time. However, they usually have only global shutter, and thus do not provide per-pixel exposure control.

[2]Fractional values can be simulated by fast binary modulation to achieve reconstructions at low temporal resolutions [4].



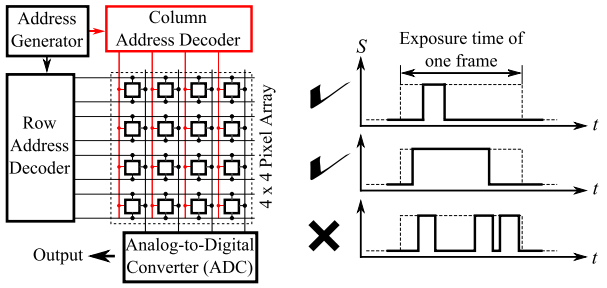
(a) Resolution trade-off (b) Motion blurred image (c) Thin-out mode: Low spatial resolution, high frame rate

(d) Our input: A single coded exposure image (e) Our result: High spatial resolution, high frame rate video

Figure 1: **Overcoming the space-time resolution tradeoff.** (a) Digital cameras face a fundamental tradeoff between spatial and temporal resolution. (b) Digital still cameras have high spatial resolution but low temporal resolution, which often results in motion blur. (c) Thin-out mode trades off spatial resolution to increase the frame rate. For large frame rates, the image quality is severely degraded. (d) By capturing a pixel-wise coded exposure image, and learning a sparse representation of videos, we achieve high-spatial resolution and frame rate video simultaneously (e).

sparsity. Specific motion models, such as periodic motion [5], locally rigid motion [6], and linear motion [7], are applicable only to specific scenarios. Instead, we propose learning an over-complete dictionary from a large collection of videos, and represent any given video as a sparse, linear combination of the elements from the dictionary. Since the dictionary is learned from video data itself, it captures common video features, for example, edges shifting in different orientations, as shown in Figure 5. Additionally, the redundant nature of these dictionaries leads to highly sparse representations [8, 9, 10]. We show that using a learned over-complete dictionary produces a significant improvement in video reconstruction as compared to previously used sparsity priors.

While we have not yet fabricated a CMOS image sensor chip with per-pixel exposure control, we constructed an emulation imaging system with an LCoS device to achieve pixel-wise exposure control [11]. We compared our approach to several existing techniques via extensive simulations and experiments. We show video reconstruction results for a variety of motions, ranging from simple linear translation to complex fluid motion and muscle deformations. We achieve temporal up-sampling factors (N) of $9X - 18X$. This enables capturing videos with frame-rates of up to 1000 fps with an off-the-shelf 60 fps machine vision camera, while maintaining high spatial resolution. **Please see the project web-page [24] for video results.** These results demonstrate the potential of using a learned over-complete dictionary based representation for compressive video sensing. We believe that this will motivate development of compressive imaging systems for a variety of other imaging domains.



(a) Architecture of CMOS image sensor (b) Single-bump restriction due to sensor architecture

Figure 2: **CMOS image sensor architecture and its limitations.** (a) Current CMOS image sensors have row addressing capability (black horizontal connections) which provides row-wise exposure control. Per-pixel exposure control can be implemented by adding column addressing (red vertical connections). (b) Most CMOS sensors do not have per-pixel frame-buffers on chip. Thus, each pixel can have only a single bump (on-time) during one camera exposure. The start and end times of the bump can be controlled.

2. Related Work

Efficient Video Capture: Several recent works have focused on efficient video capturing. Gupta et al. [12] proposed synthesizing high-resolution videos from low-resolution videos and a few high-resolution key frames. Wilburn et al. [13] proposed using camera arrays to capture high speed video. Wakin et al. [4] used the single-pixel camera for video capturing by using the sparsity of 3D discrete wavelet transform (DWT) coefficients. Other structural features in videos, such as the temporal coherence among multiple frames, sparsity in 2D DWT, and multi-scale representations have also been used for video reconstruction [7, 14, 6].

Coded Exposure Photography: An active research area in computational photography is coded exposure photography. Coded global shutter (*i.e.*, flutter shutter) has been used for motion deblurring [15] and reconstructing periodic high speed motion with compressive sensing [5]. Agrawal et al. [16] proposed temporal super resolution by multiplexing the exposure settings of four co-located video cameras. Gu et al. [17] implemented coded rolling shutter for CMOS image sensors for high speed imaging, HDR imaging, and image deblurring. Nayar et al. [18] implemented a pixel-wise coded exposure camera using a DMD (digital micro-mirror device) for HDR imaging, feature detection, and object recognition. Gupta et al. [19] implemented a similar emulation system with a projector for motion-aware photography. Recently, Reddy et al. [20] proposed a compressive video acquisition scheme using per-pixel coded exposure. Since this technique relies on optical-flow based regularization, it can not faithfully reconstruct scenes containing deforming objects, occlusions and specularities.

In contrast, our method aims at capturing videos from a single photograph while maintaining high spatial-resolution. Our method does not rely on an analytical motion model, and can handle challenging scenes, including occlusions, deforming objects and gas and fluid flow. Moreover, unlike previous approaches [7, 6], our sampling function is designed so that it is implementable in real hardware.

3. Overview of Our Approach

Let $E(x, y, t)$ denote the space-time volume corresponding to an $M \times M$ pixel neighborhood and one frame integration time of the camera. A conventional camera captures the projection of this volume along the time dimension, resulting in $M \times M$ measurements. Suppose we wish to achieve an N times gain in temporal resolution, *i.e.*, we wish to recover the space-time volume E at a resolution of $M \times M \times N$. Let $S(x, y, t)$ denote the per-pixel shutter function of the camera within the integration time ($S(x, y, t) \in \{0, 1\}$). Then, the captured image $I(x, y)$ is

$$I(x, y) = \sum_{t=1}^{N} S(x, y, t) \cdot E(x, y, t). \qquad (1)$$

For conventional capture, $S(x, y, t) = 1 \ , \forall (x, y, t)$. Our goal is to reconstruct the space time volume E from a single captured image I.

Equation 1 can be written in matrix form as $\mathbf{I} = \mathbf{SE}$, where $\mathbf{I}$ (observation) and $\mathbf{E}$ (unknowns) are vectors with $M \times M$ and $M \times M \times N$ elements, respectively. Clearly, the number of observations is significantly lower than the number of unknowns, resulting in an under-determined linear system. Recent advances in the field of compressive sensing [2, 3] have shown that this system can be solved faithfully if the signal $\mathbf{E}$ has a sparse representation $\boldsymbol{\alpha}$ using a dictionary $\mathbf{D}$:

$$\mathbf{E} = \mathbf{D}\boldsymbol{\alpha} = \alpha_1 \mathbf{D}_1 + \cdots + \alpha_k \mathbf{D}_k. \qquad (2)$$

where $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_k]^T$ are the coefficients, and $\mathbf{D}_1, \cdots, \mathbf{D}_k$ are the elements in the dictionary $\mathbf{D}$. The coefficient vector $\boldsymbol{\alpha}$ is sparse, i.e., only a few coefficients are non-zero. In this paper, the over-complete dictionary $\mathbf{D}$ is learned from a random collection of videos. At capture time, the space-time volume $\mathbf{E}$ is sampled with a coded exposure function $\mathbf{S}$ and then projected along the time dimension, resulting in a coded exposure image $\mathbf{I}$. Given $\mathbf{D}$, $\mathbf{S}$ and $\mathbf{I}$, $\mathbf{E}$ is estimated using standard sparse reconstruction techniques, as shown in Section 5.

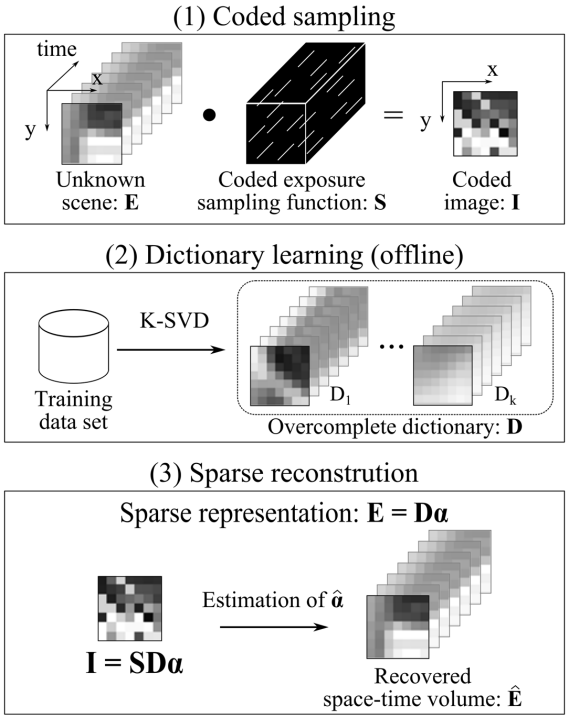


Figure 3: **Overview of our approach.** There are three main components of our approach: (1) Coded exposure sampling and projection of space-time volumes into images, (2) learning an over-complete dictionary from training video data, and (3) sparse reconstruction of the captured space-time volume from a single coded image.

Figure 3 shows the flow-chart of our approach. In the remainder of the paper, we discuss designing the coded exposure function (Section 4), the dictionary learning and sparse reconstruction (Section 5), and our hardware prototype and experimental results (Section 6).

4. Designing the Sampling Function

We design sampling functions while adhering to the following restrictions imposed by imaging architecture:

- **Binary shutter:** The sampling function S is binary *i.e.*, $S(x, y, t) \in \{0, 1\}$. At any time t, a pixel is either collecting light (1-on) or not (0-off).
- **Single bump exposure:** Since CMOS sensors do not have per-pixel frame buffers on chip, each pixel can have only one continuous "on" time (*i.e.*, a single bump) during one camera integration time, as shown in Figure 2(b).
- **Fixed bump length for all pixels:** Image sensors have a limited dynamic range. A sampling function with a large range of bump lengths among pixels would require a sensor to have a large dynamic range. We consider only the sampling functions with a fixed bump length.

We use the following scheme to assign the bump-start time for all pixels. First, we randomly select the bump-start time of the pixels within a $M \times M$ patch on the top left corner of an image sensor (denoted as p_0), such that the union of the "on" time of these M^2 pixels will cover the entire camera integration time, *i.e.*, $\sum_{(x,y)\in p_0} S(x, y, t) \geq 1$, for $t = 1, \cdots, N$ where N is the number of frames we want to reconstruct from a coded exposure image. Next, consider the adjacent $M \times M$ patch p_1 to the right of p_0. Since there are $M - 1$ overlapped columns, we keep the bump-start times for these overlapped pixels, and randomly assign the bump-start times for pixels in the new column in p_1, according to the same constraint for p_0. This process iterates until all pixels have been assigned.

We use simulations to find the optimal bump length[3]. Codes with a long bump length attenuate high frequencies, while codes with a short bump length collect less light, leading to poor signal-to-noise ratio. For each code with a given bump length, we simulate coded image capture using real high-speed video data. Signal-independent noise is added to the simulated coded exposure image. From the coded image, we recover the space-time volume using the proposed sparse reconstruction technique (Section 5). Table 1 shows the peak signal-to-noise-ratio (PSNR) values as a function of the bump length and noise level, averaged over a wide range of scenes. As expected, as the noise increases, codes with larger bump lengths are favored. In our experiments, we set the bump length to be 2 (for 9X gain) or 3 (for 18X gain).

4.1. Comparison of Sampling Schemes

We compare our random pixel-wise sampling with random row-wise sampling [17] using simulations on high-speed video data. Note that the comparison here is for the sampling scheme only - we used the same reconstruction method (proposed in this paper) for both the sampling schemes. Results are shown in Figure 4. Images in the left column are one sub-frame of the sampling function $S(x, y, t)$ for $t = 1$. White implies $S(x, y, t) = 1$, and black implies $S(x, y, t) = 0$. Images in the middle column are the coded captured images, and the right column shows one sub-frame of reconstructed space-time volume $E(x, y, t)$. Reconstruction using row-wise exposure is of poor quality. Pixel-wise sampling provides more flexibility compared to row-wise sampling, resulting in better reconstructions.

[3]The focus of this paper is not on finding the optimal space-time sampling scheme. Because of this, we restrict our sampling design to a 1-D search over only the bump-lengths. The resulting sampling functions are not necessarily optimal. Finding the optimal space-time sampling function would require a rigorous theoretical analysis, which is beyond the scope of this paper.

Bump length	Noise standard deviation σ (Grey-levels)					
	0	1	4	8	15	40
1	22.96	22.93	22.88	22.50	21.41	17.92
2	23.23	23.22	23.18	23.06	22.62	20.76
3	**23.37**	**23.37**	**23.35**	23.25	23.03	21.69
4	23.29	23.30	23.25	**23.27**	22.99	22.08
5	23.25	23.26	23.24	23.19	**23.07**	**22.34**
6	23.06	23.10	23.07	23.06	22.85	22.32
7	22.93	22.92	22.89	22.85	22.80	22.29
8	22.80	22.81	22.77	22.78	22.69	22.23
9	22.63	22.62	22.61	22.59	22.53	22.09
10	22.49	22.48	22.50	22.49	22.43	22.06

* The highest PSNR value in each column is highlighted in bold.

Table 1: **Evaluating codes with different bump lengths:** For $N = 36$, we generate codes with bump lengths from 1 to 10. For each code, we simulate coded exposure image capture using high-speed video data and add signal-independent noise of varying levels. Peak signal-to-noise-ratio (PSNR) values are computed by comparing the reconstructed space-time volume with the ground-truth.

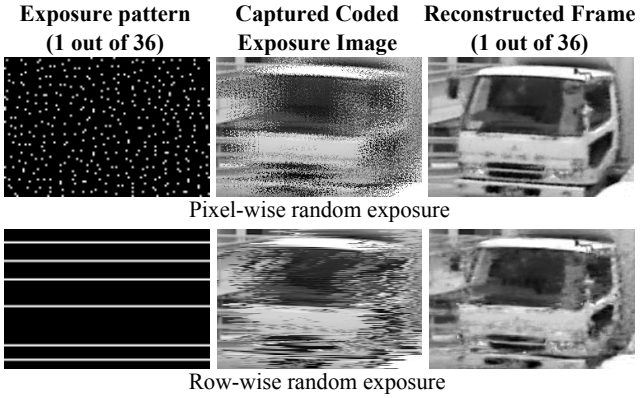


Figure 4: **Comparison of pixel-wise and row-wise sampling using simulations.** We used the reconstruction method proposed in this paper for both the sampling techniques. 36 frames were reconstructed from a single coded image. **Left column:** One sub-frame of the exposure pattern. **Middle column:** Input coded exposure images. **Right column:** One sub-frame of the reconstructed space-time volume. Pixel-wise sampling provides more flexibility compared to row-wise sampling, resulting in higher quality reconstructions.

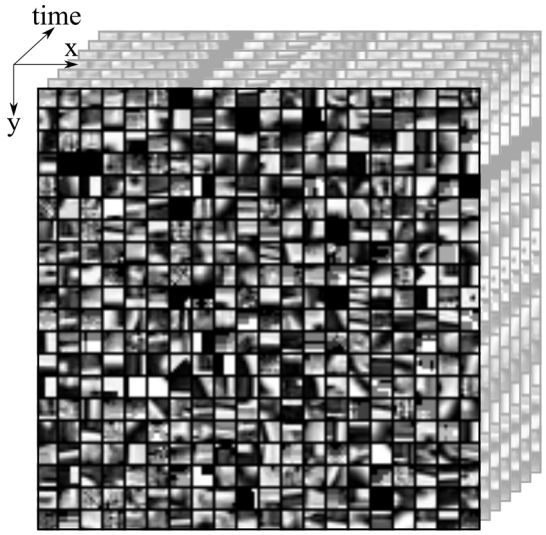


Figure 5: **Over-complete dictionary** is learned from 20 videos of resolution 384x216, rotated into 8 different orientations and played forward and backward. The frame rate of the training videos matches the target frame rate ($300 - 1000$ fps). The learned dictionary captures various local features and structures in videos, such as edges shifting in different orientations. Please see the project webpage [24] for videos of the learned dictionary.

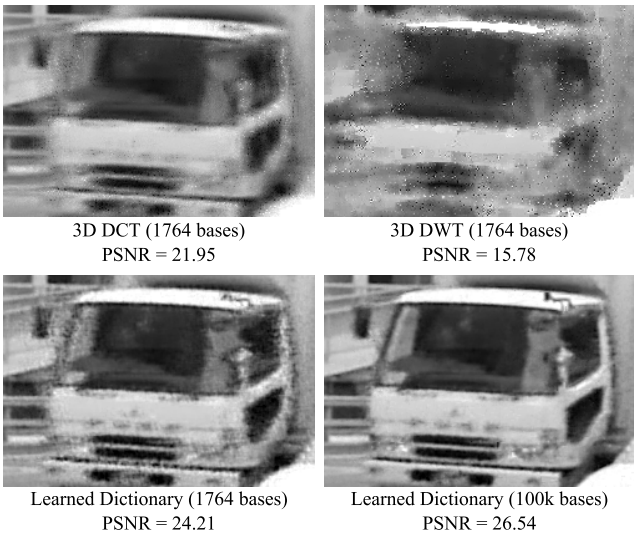


Figure 6: **Comparison of different representations.** Learned dictionaries (bottom row) capture the sparsity in signal more effectively as compared to analytical bases (top row), resulting in better reconstructions. Increasing the number of bases (over-complete dictionary) further improves the reconstruction quality. For this comparison, same sampling scheme (pixel-wise exposure) and sparse reconstruction was used.

5. Sparse Representation via Learning

In this section, we discuss the details of building a learned over-complete dictionary based sparse representation of videos, and reconstructing videos from a single coded exposure image. In general, there is a tradeoff between the generalizability and the compactness of learned dictionary based representations. While compact representations can be achieved by constructing scene class specific dictionaries (such as sports, outdoors, indoors), in order to achieve more generalizability, we choose to learn a dictionary from videos covering a wide range of scene, such as racing cars, horse running, skiing, boating and facial expression.

We model a given video as a *sparse*, *linear* combination of the elements from the learned dictionary (Equation (2)). The over-complete nature of the dictionary, and the fact that the dictionary captures most common structures and features in videos, results in highly compact and sparse representations [2, 9, 21].

In our work, we trained an over-complete dictionary on video patches of size $= M \times M \times N$ (in our experiments, we chose $M = 7, N = 36$), derived from a random selection of videos (20 sequences), using the K-SVD algorithm [21]. The frame rates of the training videos are close to our target frame rate ($500 \sim 1000$ fps). To add variation, we performed rotations on the sequences in eight directions, and played the sequences forward and backward. We learned $5000 \times 20 = 100$K dictionary elements. Figure 5 shows a part of the learned dictionary. As shown, the dictionary captures features such as shifting edges in various orientations. Please refer to the project web-page [24] for the video of the learned dictionary.

5.1. Sparse Reconstruction

Once we learn the over-complete dictionary, we apply a standard sparse estimation technique [2] to recover the space-time volume from a single captured image. Combining Equation (1) (for sampling) and Equation (2) (for sparse representation), we get $\mathbf{I} = \mathbf{S}\,\mathbf{D}\,\boldsymbol{\alpha}$, where the captured coded image $\mathbf{I}$, the shutter function $\mathbf{S}$, and the over-complete dictionary $\mathbf{D}$ are known. We use the orthogonal matching pursuit (OMP) algorithm [22] to recover sparse estimate of the vector $\hat{\boldsymbol{\alpha}}$:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \quad \text{subject to} \quad \|\mathbf{SD}\boldsymbol{\alpha} - \mathbf{I}\|_2^2 < \epsilon. \quad (3)$$

The space-time volume is computed as $\hat{\mathbf{E}} = \mathbf{D}\hat{\boldsymbol{\alpha}}$. We perform the reconstruction for all the $M \times M$ patches in the image. Every pixel (x, y) lies in M^2 patches and thus its time-varying appearance $\mathbf{E}(x, y, \mathbf{t})$ is reconstructed M^2 times. We average these M^2 reconstructions to obtain the final estimate of $\mathbf{E}(x, y, \mathbf{t})$.

5.2. Performance Comparison

Figure 6 shows the performance comparison for different representations. In this comparison, the same sampling function and reconstruction method was used

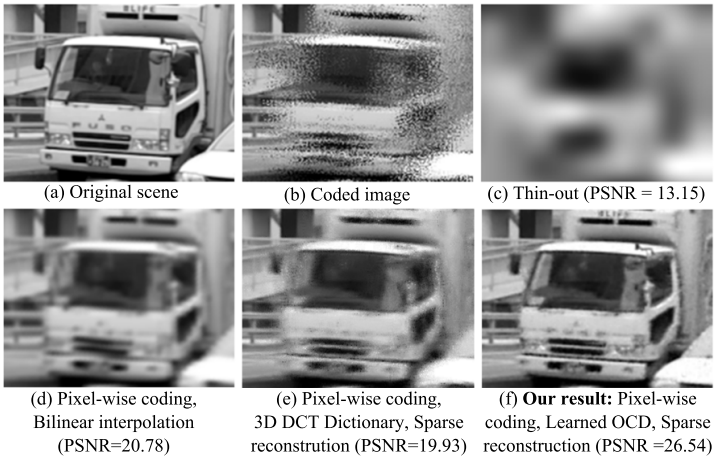


Figure 7: **Comparison of video reconstruction techniques using simulations. Left:** (a) One frame of original sequence. (b) Pixel-wise coded exposure image. (c) One sub-frame of the thin-out movie (row-wise exposure). (d) 3D bilinear interpolation [19]. (e) 3D DCT dictionary and sparse reconstruction. (f) Our result using the learned over-complete dictionary and sparse reconstruction. In this comparison, 36 frames were reconstructed from a single coded image. **Right:** As the number of frames reconstructed from a single image increases, the performance of all the techniques degrades.

for all the representations. The comparisons were performed using simulations on high-speed video data. The trained over-complete dictionary has higher PSNR as compared to the analytical bases for the same number of bases elements. Increasing the number of bases elements further improves the reconstructed image quality as well as the PSNR values.

Figure 7 shows the comparison of the performance of video acquisition between our approach (*i.e.*, a learned over-complete dictionary + sparse reconstruction) and several previous techniques: the thin-out mode (Figure 7(c)), the pixel-wise grid exposure and 3D bilinear interpolation [19](Figure 7(d)), and the pixel-wise random exposure and reconstruction using a 3D DCT dictionary (Figure 7(e)). Our result has more details, both on the moving objects and background. The right side of Figure 7 shows the PSNR of the reconstructed video using these techniques, as the target temporal resolution increases. As expected, the performance degrades as more frames are reconstructed from a single captured image. Our approach consistently outperforms the other methods. Please refer to the project webpage [24] for more comparison results in videos.

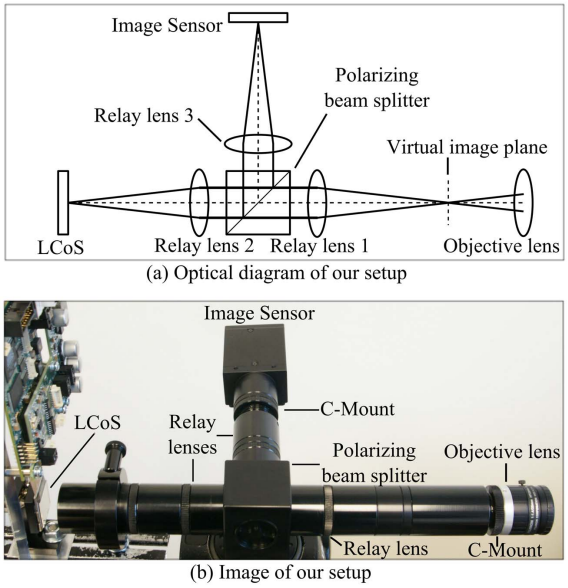


Figure 8: **Our hardware setup:** Optical diagram (top) and image (bottom) of our setup. Our system emulates fast per-pixel shutter using a liquid crystal on silicon device.

6. Hardware Prototype and Experiments

Our sampling scheme (Section 4) requires fast per-pixel modulation. We simulate fast per-pixel shutter using a liquid crystal on silicon (LCoS) device. These devices have been used as spatial light modulators in various imaging applications [23, 11], and can display binary patterns at fast rates (up to $3000Hz.$), making them ideally suited to our application. Figure 8 illustrates our hardware setup. It consists of an image sensor (Point Grey Grasshopper), an LCoS chip (Forth Dimension Displays, SXGA-3DM, spatial resolution 1280×1024), a polarizing beam-splitter, relay lenses and an objective lens. The scene is first imaged on a virtual image plane through the objective lens, then on the LCoS, and finally on the image sensor. The camera and LCoS are synchronized using a trigger signal from the LCoS. During a single camera exposure, the LCoS displays several binary images, corresponding to the sampling function. We typically run the LCoS at $9 \sim 18$ times the frame-rate corresponding to the integration time of the camera. For example, for an 18ms camera integration time ($55Hz.$), we operated the LCoS at $1000Hz.$, resulting in 18 video frames from

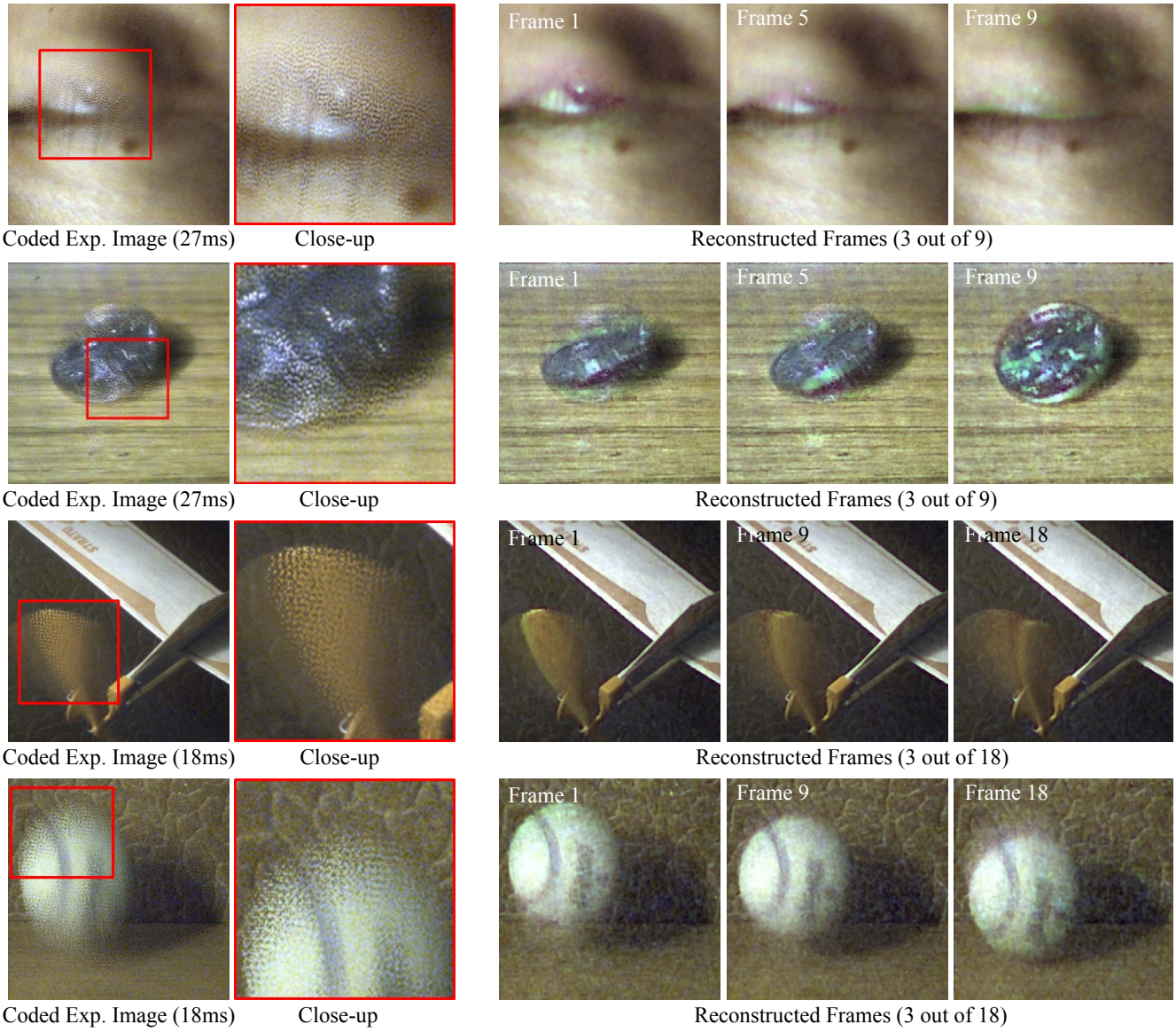


Figure 9: **Experimental results: First column:** Input coded exposure images. Numbers in parentheses denote the camera integration time for the input image. **Second column:** Close-ups illustrate the coded motion blur. **Third-fifth columns:** The reconstructions maintain high spatial resolution despite a significant gain in temporal resolution ($9X - 18X$). Notice the spatial details inside the eye, on the coin and the table, wing of the plane and the stripe on the ball.

a single coded exposure image.

Experimental Results: Using our hardware prototype, we capture and reconstruct scenes comprising of a wide range of motions. Figure 9 shows the results. The first example demonstrates the motion of an eyelid during blinking. This motion is challenging as it involves occlusion and muscle deformations. The input frame was captured with an exposure time of 27ms. Notice the coded motion blur on the input frame. We recover 9 video frames from the captured image, equivalent to an output frame rate of 333 fps. Please see the project web-page [24] for the complete video.

The second example shows a coin rotating on a table. This motion is challenging due to occlusions; as the coin rotates, one face of the coin becomes visible to the camera. The input frame was captured with an exposure time of 27ms. From the single captured image, 9 output frames were reconstructed, while maintaining high spatial resolution, both on the coin and the table. The third and the fourth examples consist of rotating rotor-blades on a toy plane and a ball falling vertically, respectively. The input frames, captured with an exposure time of 18ms, show large motion blur. In order to recover the high-speed motion, we performed the

reconstruction at 1000 fps (18 output frames). Notice the sharp edges of the blade and the texture on the ball in the output frames. The spatial detail on the static wings of the toy-plane are nearly the same as the input image. **Please see the project web-page [24] for more examples, ranging from simple linear translation to complex fluid motion.**

7. Discussion and Limitations

In this paper, we proposed an efficient way of capturing videos from a single photograph using pixel-wise coded exposure. We incorporated the hardware restrictions of existing image sensors into the design of the sampling schemes, and implemented a hardware prototype with an LCoS device that has pixel-wise exposure control. By using an over-complete dictionary learned from a large collection of videos, we achieved sparse representations of space-time volumes for efficient reconstruction. We demonstrated the effectiveness of our method via extensive simulation and experiments.

The proposed method has several limitations. First, the maximum temporal resolution of the over-complete dictionary has to be pre-determined (*e.g.*, 36 frames). To reconstruct videos at different temporal resolutions, we have to train different dictionaries. The hardware setup requires precise alignment of the camera and the LCoS. Imperfect alignment can cause artifacts (ghosting), as is visible in some results. We believe that these artifacts will be reduced significantly once per-pixel coded exposure is implemented on chip.

Acknowledgments: This research was supported in parts by Sony Corporation, NSF (grant number IIS 09-64429) and ONR (grant number N00014-08-1-0638).

References

[1] S. Kleinfelder, S. Lim, X. Liu, and A. Gamal. A 10,000 frames/s CMOS digital pixel sensor. *IEEE Journal of Solid-State Circuits*, 36, 2001. 1

[2] D. Donoho, M. Elad, and V. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans. on Information Theory*, 52(1), 2006. 1, 3, 5

[3] E. Candes, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59, 2006. 1, 3

[4] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk. Compressive imaging for video representation and coding. In *Proc. of Picture Coding Symposium*, 2006. 1, 2

[5] A. Veeraraghavan, D. Reddy, and R. Raskar. Coded strobing photography: Compressive sensing of high-speed periodic events. *IEEE Trans. on PAMI*, 99, 2010. 2, 3

[6] J. Park and M. Wakin. A multiscale framework for compressive sensing of video. In *Proc. Picture Coding Symposium*, 2009. 2, 3

[7] A. Sankaranarayanan, P. Turaga, R. Baraniuk, and R. Chellappa. Compressive acquisition of dynamic scenes. In *ECCV*, 2010. 2, 3

[8] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *ICCV*, 2009. 2

[9] M. Elad and M. Aharon. Image denoising via learned dictionaries and sparse representation. In *CVPR*, 2006. 2, 5

[10] M. Protter and M. Elad. Image sequence denoising via sparse and redundant representations. In *IEEE Trans. on Image Processing*, 2007. 2

[11] H. Nagahara, C. Zhou, T. Watanabe, H. Ishiguro, and S. Nayar. Programmable aperture camera using LCoS. In *ECCV*, 2010. 2, 6

[12] A. Gupta, P. Bhat, M. Dontcheva, O. Deussen, B. Curless, and M. Cohen. Enhancing and experiencing spacetime resolution with videos and stills. In *ICCP*, 2009. 2

[13] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz. High speed video using a dense camera array. In *CVPR*, 2004. 2

[14] R. Marcia and R. Willett. Compressive coded aperture video reconstruction. In *European Signal Processing Conference*, 2008. 2

[15] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: Motion deblurring using fluttered shutter. In *SIGGRPAH*, 2006. 3

[16] A. Agrawal, M. Gupta, A. Veeraraghavan, and S. Narasimhan. Optimal coded sampling for temporal super-resolution. In *CVPR*, 2010. 3

[17] J. Gu, Y. Hitomi, T. Mitsunaga, and S. Nayar. Coded rolling shutter photography: Flexible space-time sampling. In *ICCP*, 2010. 3, 4

[18] S. Nayar, V. Branzoi, and T Boult. Programmable Imaging: Towards a Flexible Camera. *Int'l J. Computer Vision*, 70, 1, 7-22 3

[19] M. Gupta, A. Agrawal, A. Veeraraghavan, and S. Narasimhan. Flexible voxels for motion-aware videography. In *ECCV*, 2010. 3, 6

[20] D. Reddy, A. Veeraraghavan, and R. Chellappa. P2C2: Programmable Pixel Compressive Camera for High Speed Imaging. In *CVPR*, 2011. 3

[21] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. In *IEEE Trans. on Signal Processing*, volume 54, 2006. 5

[22] J. Tropp and A. Gilbert. Signal Recovery From Random Measurements via Orthogonal Matching Pursuit. In *IEEE Trans. on Information Theory*, volume 53, 2007. 5

[23] H. Mannami, R. Sagawa, Y. Mukaigawa, T. Echigo, and Y. Yagi. Adaptive dynamic range camera with reflective liquid crystal. *Journal of Visual Communication and Image Representation*, 18(5), 2007. 6

[24] Web-page: *http://www.cs.columbia.edu/CAVE/projects/DictonaryBasedCompressiveVideo* 2, 5, 6, 7, 8