Efficient Space-Time Sampling with Pixel-wise Coded Exposure for High Speed Imaging

Dengyu Liu, Jinwei Gu, Yasunobu Hitomi, Mohit Gupta, Tomoo Mitsunaga, Shree K. Nayar

Abstract—Cameras face a fundamental tradeoff between spatial and temporal resolution. Digital still cameras can capture images with high spatial resolution, but most high-speed video cameras have relatively 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 hardware constraints on sampling schemes imposed by architectures of current image sensors, which means that our sampling function can be implemented on CMOS image sensors with modified control units in the future. We evaluate components of our approach - sampling function and sparse representation by comparing them to several existing approaches. We also implement a prototype imaging system with pixel-wise coded exposure control using a Liquid Crystal on Silicon (LCoS) device. System characteristics such as field of view, Modulation Transfer Function (MTF) are evaluated for our imaging system. Both simulations and experiments on a wide range of scenes show that our method can effectively reconstruct a video from a single coded image while maintaining high spatial resolution.

Index Terms—Space-Time Sampling, Dictionary Learning, Sparse Reconstruction, Computational Camera

1 Introduction

Digital cameras are limited by a fundamental tradeoff between spatial resolution and temporal resolution. As the frame rate increases, spatial resolution decreases. This limitation is caused by hardware factors such as the readout and Analog-to-Digital (A/D) conversion time of image sensors. Although it is possible to increase the readout throughput by introducing parallel A/D converters and frame buffers [1], it often requires more transistors per pixel and thus lowers the fill factor and increases cost. As a compromise, many camera manufacturers implement a "thin-out" mode (*i.e.*, high speed draft mode [2]), which directly trades off the spatial resolution for higher temporal resolution, and thus degrades image quality, as shown in Fig. 1.

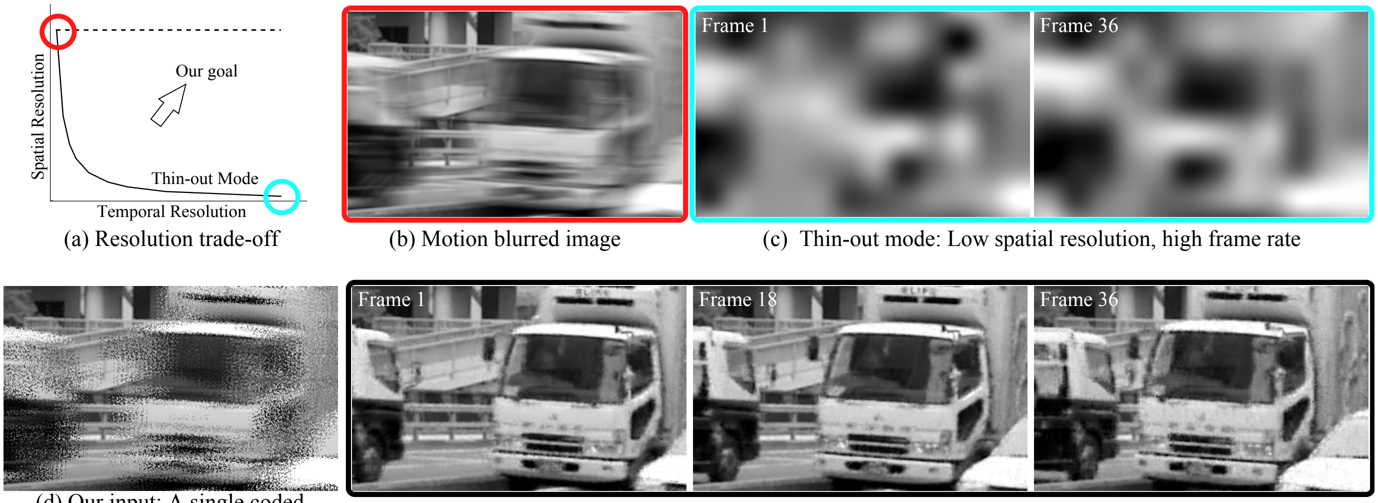
Can we go beyond this fundamental limitation and capture video more efficiently? Some recent works have addressed this problem. Gupta et al. [3] proposed to combine low-resolution videos and a few high-resolution still images to synthesize high-resolution videos. Bub et al. [4] implemented a camera with per-pixel exposure control using a Digital Micro-mirror Device (DMD), which can flexibly control the tradeoff between spatial and temporal resolution. Gupta et al. [5] proposed an adaptive, motion-aware algorithm which estimates motion from several frames and assigns high spatial but low temporal resolution exposure for still/slow-moving regions and vice versa for fast-moving regions. Reddy et al. [6] proposed a programmable pixel-wise compressive camera by exploiting spatial redundancy using sparse representation and temporal redundancy based on brightness constancy over time.

In this paper, we exploit statistical priors of time-varying appearance of natural scenes and propose a pixel-wise coded exposure to capture a video from a single photograph. Our key assumption is that the time-varying appearance of natural scenes can be represented as a sparse linear combination of the atoms of an over-complete dictionary learned from training data. Recent progress in compressive sensing has provided a general framework for efficient capture of sparse signals[7, 8]. Thus, by using pixel-wise coded exposure, we can obtain a 2D projection of the 3D space-time volume and reconstruct the volume via sparse reconstruction algorithms. Fig. 1(d) shows the captured image with coded pixel-wise exposure. Fig. 1(e) shows three frames of the reconstructed video. Notice that the image spatial resolution is preserved much better as compared to Fig. 1(c).

Different from previous work, our approach has two important contributions:

- We use a data-driven sparse representation for videos, which is more effective for sparse reconstruction. General analytical transforms, such as Discrete Cosine Transform (DCT) and Discrete Wavelets Transform (DWT), do not provide the desired level of compactness for sparse representation. Specific motion models, such as periodic motion[9], locally

- *D. Liu and J. Gu are with Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology, Rochester, NY 14623. E-mail: dxl5849@rit.edu and jwgu@cis.rit.edu*
- *Y. Hitomi and T. Mitsunaga are with SONY Corporation. E-mail: Yasunobu.Hitomi@jp.sony.com and Tomoo.Mitsunaga@jp.sony.com*
- *M. Gupta and S. K. Nayar are with the Department of Computer Science, Columbia University, New York, NY 10027. E-mail: mohitg@cs.columbia.edu and nayar@cs.columbia.edu*



(a) Resolution trade-off

(b) Motion blurred image

(c) Thin-out mode: Low spatial resolution, high frame rate

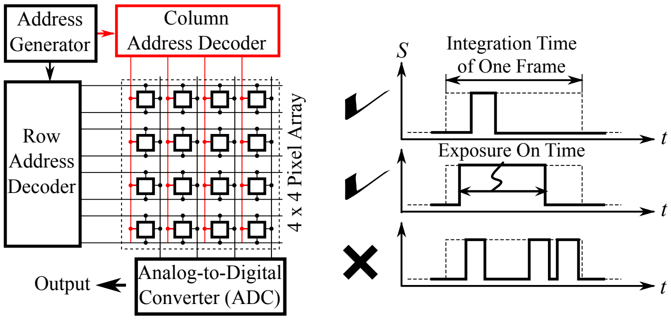
(d) Our input: A single coded exposure image

(e) Our result: High spatial resolution, high frame rate video

Fig. 1: Overcoming the space-time resolution tradeoff. (a) Digital cameras face a fundamental tradeoff between spatial resolution and temporal resolution. (b) Digital still cameras have high spatial resolution but low temporal resolution, which often results in motion blur. (c) The 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, (e) we achieve a high-spatial resolution and high frame rate video simultaneously.

rigid motion[10] and linear motion[11] are only applicable to specific scenarios. Instead, the sparsity of natural images on an over-complete dictionary has been well studied and applied for image denoising, inpainting, and compression [12]. In this paper, we exploit similar statistical priors for reconstructing natural videos from a single coded image.

- We impose a practical constraint — non-intermittent per pixel exposure — to the sampling function, which makes our approach easier to implement on real image sensors. Right now, most CMOS image sensors only have row addressing ability (Fig. 2(a)). But in the future, pixel-wise exposure control is achievable [13, 14]. Due to the readout time limit and the fact that most CMOS sensors have no frame buffer on chip, each pixel can only have one continuous exposure during the integration time of one shot (Fig. 2(b))[1]. For example, assume 0 represents “exposure off” and 1 represents “exposure on”, the exposure sequence $[0, 0, 1, 1, 1, 0]$ is realizable while the intermittent exposure sequence $[0, 1, 0, 1, 0, 1]$ is not. Therefore, it is important to adhere to this restriction to make our technique implementable on real CMOS sensors.



(a) Architecture of CMOS image sensor

(b) Single-bump restriction due to sensor architecture

Fig. 2: CMOS sensor architecture and limitations. (a) Current CMOS sensors only have row addressing capability (black horizontal connections) which provides row-wise exposure control. One possible way to achieve per-pixel exposure control is adding column addressing (red vertical connections). (b) Most CMOS sensors do not have per-pixel frame-buffers on chip. Thus, each pixel can only have one single bump (one “exposure on” time) during a camera integration time.

Since it is still challenging to fabricate a CMOS image sensor with pixel-wise coded exposure control with current technology, we construct an emulated imaging system using Liquid Crystal on Silicon (LCoS) to prove our concept.

A shorter version of this work appeared on [15]. This journal version extends our work with more comparisons via extensive simulations and experiments, more details on hardware implementation and system evaluation . In the remainder of the paper, section 2 discusses related work. Section 3 discusses the overview of our approach, followed by the design of space-time sampling function (Section 4), sparse representation (Section 5) and comparison analysis (Section 6). Section 7 describes the hardware prototype. Experimental results are shown in Section 8. We show video reconstruction results for a variety of motions, ranging from simple linear translation to complex fluid motion and muscle deformation. We achieve temporal up-sampling factors of $9X - 18X$. This enables capturing videos with frame-rates up to 1000 fps with an off-the-shelf 30 fps machine vision camera while maintaining high spatial resolution.

[1] CCD image sensors allow multiple bumps (*i.e.*, several individual “exposure on” time within one integration time). However, they usually only have global shutter, and single readout amplifier, thus do not have per-pixel control.

2 Related Work

Efficient Video Capture: Several recent works have focused on efficient video capturing. Gupta et al. [3] proposed synthesizing high-resolution videos from low-resolution videos and a few high-resolution key frames. Wilburn et al. [16] proposed using camera arrays to capture high speed videos. Wakin et al. [17] built a single-pixel camera for video capturing by using the sparsity of 3D DWT coefficients. Ben-Ezra and Nayar [18] and Tai et al. [19] used hybrid camera systems to do motion-deblurring and temporal up-sampling. Sankaranarayanan et al. [20] improved the single-pixel camera by using a multi-scale video sensing and recovery framework. 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 [11, 21, 10].

Coded Exposure Photography: Coded exposure photography is an active research area in computational photography. Coded global shutter (*i.e.,* flutter shutter) has been used for motion deblurring [22] and reconstructing periodic high speed motion with compressive sensing [9]. Agrawal et al. [23] proposed temporal super resolution by multiplexing the exposure settings of four co-located video cameras. Gu et al. [24] proposed coded rolling shutter for CMOS image sensors for high speed imaging, high dynamic range imaging *etc.*. Nayar et al. [25] proposed programmable imaging system using DMD for high dynamic range imaging. Ri et al. [26] also built a DMD camera to do phase analysis and shape measurement. Bub et al. [4] implemented a pixel-wise coded exposure camera using DMD for high speed imaging. They designed an optimized sampling function to let pixels expose at different frames. They traded off the spatial resolution to obtain high speed videos by up-sampling. Gupta et al. [5] implemented a similar emulation system with a projector for motion-aware photography. Unlike Bub's method which applied the trade off on the whole scene, they only traded off at moving regions of the scene. Shu and Ahuja [27] proposed a 3D compressive sampling approach which reconstructed high spatial resolution video from low spatial resolution sensor. Reddy et al. [6] proposed a programmable pixel-wise compressive camera based on LCoS.

Different from previous approaches[10, 11, 6], our method exploits spatio-temporal sparsity of natural videos by training an over-complete dictionary, which has been verified on image processing by Aharon et al. [12]. We aim at reconstructing a video from a single photograph for a wide range of motions while maintaining high spatial-resolution. Our method does not rely on analytical motion models, and can handle challenging scenes where occlusions, deforming objects, flame and fluid flow are presented. Moreover, our sampling function is designed so that it is implementable in real hardware in the future.

3 Overview

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 an $M \times M$ image patch. 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

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

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

Figure 3 shows the flow-chart of our approach. Equation (1) can be written in a matrix form as $\mathbf{I} = \mathbf{SE}$, where $\mathbf{I}$ (observation) and $\mathbf{E}$ (unknowns) are vectors with M^2 and $N \times M^2$ 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 [7, 8] have shown that this system can be solved faithfully if the signal $\mathbf{E}$ has a sparse representation $\boldsymbol{\alpha}$ using a dictionary $\mathbf{D}$:

$$\mathbf{E} = \mathbf{D}\boldsymbol{\alpha} = \alpha_1\mathbf{D}_1 + \cdots + \alpha_k\mathbf{D}_k, \tag{2}$$

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, which means only few coefficients are non-zeros. 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 $\mathbf{I}$, $\mathbf{E}$ can be estimated using standard sparse reconstruction techniques.

4 Space-Time Sampling

We design sampling functions which satisfy the following restrictions imposed by image sensors:

- **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 a camera integration time, as shown in Fig. 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

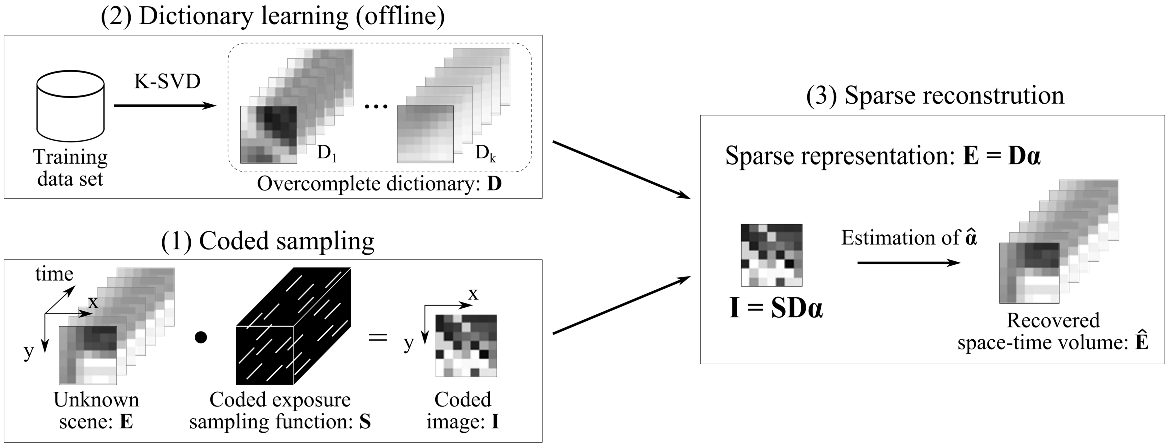

(2) Dictionary learning (offline)
K-SVD
Training data set
D_1
...
D_k
Overcomplete dictionary: **D**
(1) Coded sampling
time
x
y
Unknown scene: **E**
•
Coded exposure sampling function: **S**
=
x
y
Coded image: **I**
(3) Sparse reconstrution
Sparse representation: **E = Dα**
Estimation of $\hat{\boldsymbol{\alpha}}$
I = SDα
Recovered space-time volume: $\hat{\mathbf{E}}$

Fig. 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 a training video dataset, and (3) sparse reconstruction of the captured space-time volume from a single coded image.

would require a sensor to have a large dynamic range. We only consider 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. 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. Coded exposure with a long bump length attenuates high frequencies, while coded exposure with a short bump length collects less light, leading to poor signal-to-noise ratio. For each coded exposure 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. As expected, when the noise increases, codes with larger bump lengths are favored. In our experiments, we set the bump length to 2 (with 9X gain) or 3 (with 18X gain).

TABLE 1: Evaluating codes with different bump lengths.

Bump	Noise standard deviation σ (Grey-levels)					
length	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.

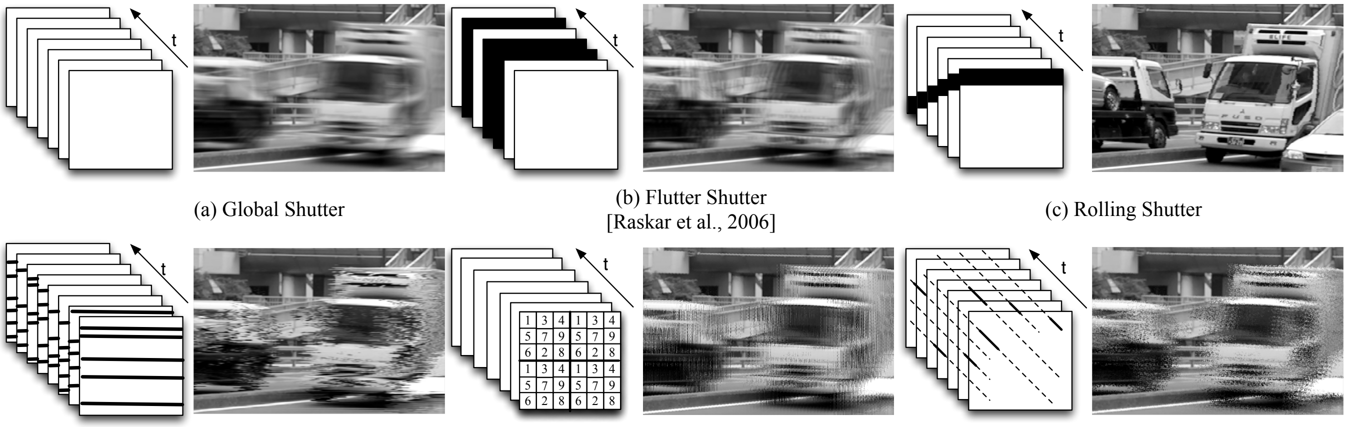
5 Sparse Representation via Learning

In this section, we discuss the details of building the sparse representation of videos and reconstructing videos from a single exposure coded image. To obtain the sparse representation of videos, we choose to learn an over-complete dictionary from videos covering a wide range of scenes, such as racing cars, horse running, skiing, boating and facial expression.

We then model a given video as a *sparse*, *linear* combination of the elements from the learned dictionary (Equation (2)). The over-completeness guarantees the sparsity of the representation [7, 12]. The learning is used to find a dictionary that captures most common structures and features in videos for compact, sparse decomposition [28, 12].

In our study, we learn a dictionary on video patches of size $7 \times 7 \times 36$, derived from a random selection of videos (20 sequences), using the K-SVD algorithm [12]. The frame rates of the training videos are close to our target frame rate (500 $\sim$ 1000 fps). To add variation, we perform rotations on the sequences in eight directions, and play the sequences forward and backward. We learn $5000 \times 20 = 100$K dictionary elements. Fig. 5 shows a part of the learned dictionary. The dictionary captures features such as shifting edges in various orientations. Please refer to supplemental materials for the video of the learned dictionary.

Once we learn the dictionary, we apply a standard sparse estimation technique [7] 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 dictionary $\mathbf{D}$ are known. We use Orthogonal Matching Pursuit



(a) Global Shutter

(b) Flutter Shutter [Raskar et al., 2006]

(c) Rolling Shutter

(d) Coded Rolling Shutter [Gu et al., 2010]

(e) Grid Pixel-wise Shutter [Gupta et al., 2010]

(f) Random Pixel-wise Shutter [Hitomi et al., 2011]

Fig. 4: Sampling functions and corresponding coded images. (a) All pixels on the image sensor are exposed for a continuous period of time, which generates a motion-blurred image. Instead of keeping the shutter open for the entire exposure duration, (b) opens and closes the shutter in a binary pseudo-random sequence. (c) is widely used in CMOS image sensor, which reads out data row-by-row, sequentially from top to bottom. Since pixels in different rows are exposed to light at different times, this causes skew effect for moving objects. (d) adds row-wise controllability on existing rolling shutter. (e) divides the whole image areas into several blocks and applies an optimized exposure pattern on each block. Four 3×3 blocks are shown here. Each pixel is assigned to a specific number which indicates the subframe where the pixel is exposed. (f) Instead of assigning the exposure sequence in a specific order, pixels are randomly exposed for one or several continuous frames in a whole integration period.

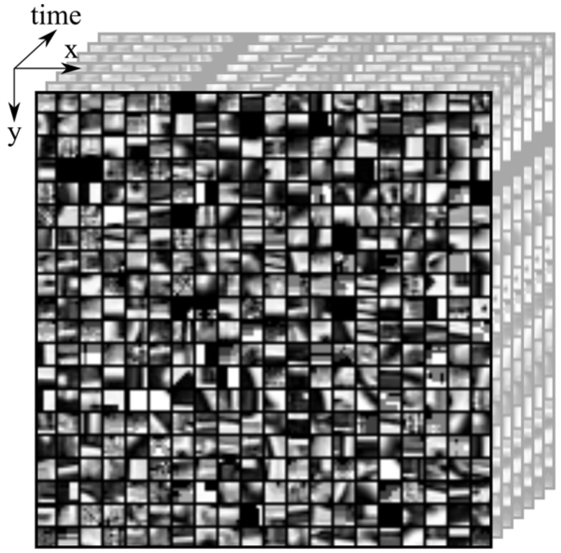


Fig. 5: An over-complete dictionary is learned from 20 videos of resolution 384×216, rotated into 8 different orientations and played forward and backward. The frame rate of the training videos matches the target frame rate ($500 - 1000$ fps). The learned dictionary captures various local features and structures in videos, such as edges shifting in different orientations. Please see supplemental materials for video of the learned dictionary.

(OMP) algorithm [29] to recover sparse estimate of the vector $\hat{\boldsymbol{\alpha}}$:

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

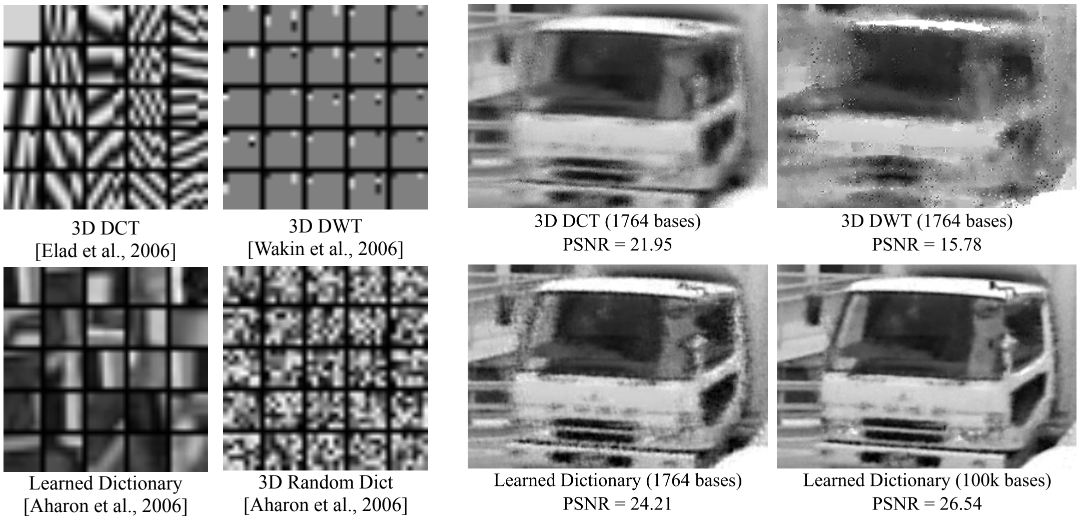
The space-time volume is computed as $\hat{\mathbf{E}} = \mathbf{D}\hat{\boldsymbol{\alpha}}$. We perform the reconstruction for all the $M \times M$ patches in the image. Every pixel (x, y) lies in M^2 patches and thus its time-varying appearance $\mathbf{E}(x, y, \mathbf{t})$ is reconstructed M^2 times.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

6 Evaluation and Comparison

In this section, we evaluate the influence factors including sampling function, representation (dictionary), dictionary patch size and noise, which contribute to the final performance of reconstruction.

6.1 Sampling Function

Figure 4 shows six sampling functions and their corresponding coded images. We choose a scene with *moving trucks* in this figure. Global shutter is the ordinary sampling function which exposes the whole image in the integration period. As expected, the moving truck is blurred. Flutter shutter [22] opens and closes the shutter many times in an optimized pattern during a single integration time. It preserves some high frequency details, as shown at edges of the moving trucks. Conventional rolling shutter is applied in most CMOS sensors. With a rolling shutter, the whole image is readout row-by-row under the control of the row address decoder. One disadvantage of the rolling shutter is the skew effect which tilts the moving truck as shown in the image. Coded rolling shutter [24] is based on the scheme of rolling shutter, but changes the conventional readout sequence. It achieves row-wise exposure control as shown in the coded image. By using a spatial light modulator, pixel-wise exposure pattern can be implemented. Grid pixel-wise shutter [5] divides the whole image area into several blocks. In each block (*e.g.*, 3×3), an optimized sampling function is applied. Pixel-wise exposure patterns can be designed to have single bump or multiple bumps in an integration period. In order to adhere to the hardware restriction, we choose single bump exposure pattern for comparison.



(a) Dictionary patches (7x7) (b) Performance Comparison using Different Dictionaries

Fig. 6: (a) First five rows and columns of four dictionaries (7×7 patch size). The patch on the top left corner of 3D DCT is the DC component, thus it only has gray intensity. When it goes to the bottom right, the frequency of the patch pattern becomes higher. 3D DWT is based on Haar wavelet. Learned dictionary is trained from 20 different scenes using K-SVD. 3D Random dictionary is generated using i.i.d. uniformly distributed entries. (b) Performance 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 function (pixel-wise exposure) and sparse reconstruction algorithm are used.

6.2 Representation

Figure 6(a) shows the first five rows and columns of the four dictionaries (7×7 patch size). In 3D DCT, the patch on the top left corner is the DC component, thus it only has gray intensity; patches near bottom right corner represent higher frequency components. Other patches show patterns with different frequencies. 3D DWT is built with Haar wavelet bases. The learned over-complete dictionary is trained from 20 different scenes using K-SVD algorithm. 3D random dictionary is generated based on i.i.d uniformly distributed entries.

Figure 6(b) shows the performance comparison for different representations. In this comparison, the same sampling function and reconstruction method are used for all the representations. The comparisons are performed using simulations on a high-speed video. The learned over-complete dictionary has higher PSNR as compared to the analytical bases for the same number of bases elements.

6.3 Coded Sampling vs. Sparse Representation

As shown in the diagram of our approach, both coded sampling function and sparse representation (dictionary) are needed for reconstruction. But which is more important — coded sampling or sparse representation? To answer this question, we perform a thorough comparison analysis on different combinations of sampling functions and sparse representations.

We select four dictionaries, six sampling functions and five different size of dictionary patches for comparison analysis, which are 120 configurations in total for one scene. All reconstructions are done using the algorithm mentioned in section 5. For time efficiency, we use high performance computing resources from National Institute for Computational Science (NICS).

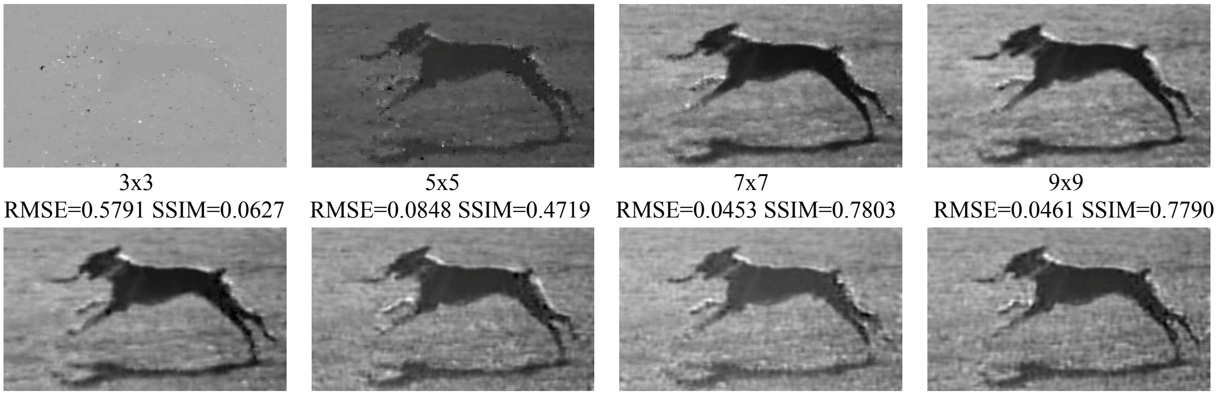
Figure 14 shows the grid reconstruction results for six sampling functions and four dictionaries with 7×7 patch size. The results are the reconstructions of 36 frames from a single coded image. We calculate averaged Root Mean Squared Error (RMSE[2]) and Structural SIMilarity [30] (SSIM).[3] Notice that the combination of pixel-wise coded exposure and learned dictionary yields the smallest RMSE and the largest SSIM among all configurations. Although the numerical difference in RMSE and SSIM evaluation between grid pixel-wise shutter and random pixel-wise shutter (using learned dictionary) is small, we can still see visual difference in the reconstruction result. There are jagged artifacts along the edge, which may be caused by the repetitive structure in grid pixel-wise shutter. Whereas the edge is smoother in the result using random pixel-wise shutter. Besides, coded sampling (either row-wise or pixel-wise) generally results in better reconstruction irrespective of the choice of sparse representation. We run the same simulation on several test videos in our database and observe similar trend. Thus, we conclude that both coded sampling and sparse representation are important for reconstruction, but coded sampling contributes more.

6.4 Dictionary Patch Size

We analyze the reconstruction results for different dictionary patch sizes using pixel-wise sampling function

[2] RMSE is calculated as $RMSE(y - \hat{y})/(\hat{y}_{max} - \hat{y}_{min})$, ranging from 0 to 1. The lower value, the better image quality.

[3] SSIM is a measurement for perceptual similarity between two images, ranging from 0 to 1. The higher value, the better image quality.



3x3 RMSE=0.5791 SSIM=0.0627

5x5 RMSE=0.0848 SSIM=0.4719

7x7 RMSE=0.0453 SSIM=0.7803

9x9 RMSE=0.0461 SSIM=0.7790

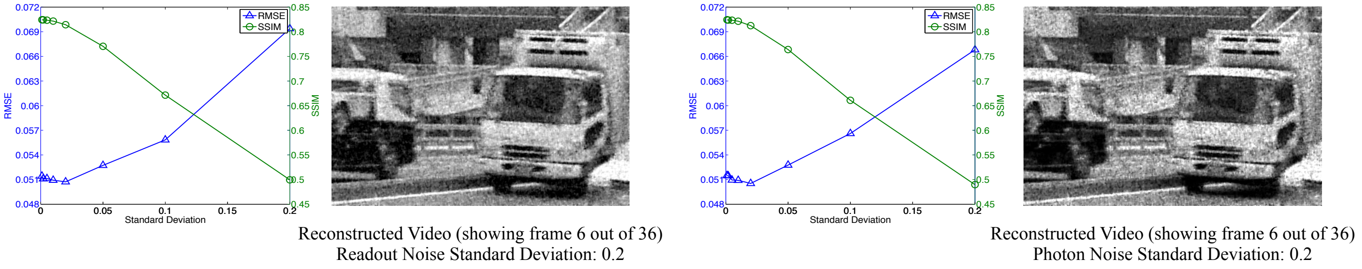
11x11 RMSE=0.0638 SSIM=0.6218

13x13 RMSE=0.0589 SSIM=0.6813

15x15 RMSE=0.0619 SSIM=0.6702

17x17 RMSE=0.0621 SSIM=0.6649

Fig. 7: Reconstruction results (36X gain) based on eight dictionary patch sizes(showing frame 7 out of 36 video frames). When the patch size is too small (*e.g.*, 3×3), the learned dictionary patches contain no detail information for the input sources, only gray intensity left, thus the reconstructed result appears gray. When the patch size is too large(*e.g.*, 17×17), the learned dictionary patches only contain general features and lost high frequency information, which can be seen from the grass and dog's back feet.



Reconstructed Video (showing frame 6 out of 36)
Readout Noise Standard Deviation: 0.2

(a) Readout Noise Evaluation

Reconstructed Video (showing frame 6 out of 36)
Photon Noise Standard Deviation: 0.2

(b) Photon Noise Evaluation

Fig. 8: Noise performance evaluation. We show averaged RMSE and SSIM curves for the reconstructed video (36X gain), with readout noise and photon noise evaluation. When the standard deviation of noise is 0.2, the RMSE is less than 0.07, and the SSIM is about 0.5.

and learned dictionary, as shown in Fig. 7. When the dictionary patch size is too small, the learned dictionary patches do not contain any detail information of the source video dataset, but only gray intensity, thus fail to represent other videos. When the dictionary patch size is too large, it is not efficient to reconstruct detail features of the scene. As shown in the results for 17×17 dictionary patch, the figure shows the block artifact on dog's legs. At the same time, a larger dictionary patch size also requires much longer reconstruction time . Considering the performance and time cost, we choose a dictionary patch size as 7×7.

6.5 Noise Performance

We simulate reconstruction with photon and readout (Gaussian) noise. Fig. 8 shows the averaged RMSE and SSIM plot for the *truck* scene. We evaluate the noise performance with mean of the signal power (for photon noise, the square root of signal power), and standard deviation range from 0.001 to 0.2. One frame of the reconstructed video are shown with noise standard deviation of 0.2. The results show that our method is robust to photon noise and readout noise in a relative scale.

6.6 Comparison Results with Other Methods

We compare our reconstruction results with recent methods using flexible voxel [5] and P2C2 [6]. We use only one coded image as the input. Fig. 9 shows one comparison result (Please see other results in supplementary material) for one frame of the reconstructed video with error evaluation. Flexible voxel method generates different spatial-temporal interpretations from the coded image, and then do motion-aware post-processing interpolation. It preserves high spatial resolution features in the static region, but trades off spatial resolution for high speed motion, as we can see blurry features on the dog. P2C2 does a good job when using multiple coded images to calculate optical flow, but if there is only one coded image, the reconstruction result is degraded. In summary, flexible voxel is simple and fast, but limited to simple scenes with few features. P2C2 needs several coded images to better exploit the temporal redundancy. Our method exploits natural video priors by using a dictionary learning based algorithm instead of interpolation or optical flow. Although the time cost is relative high, it outperforms other two methods in most scenarios.

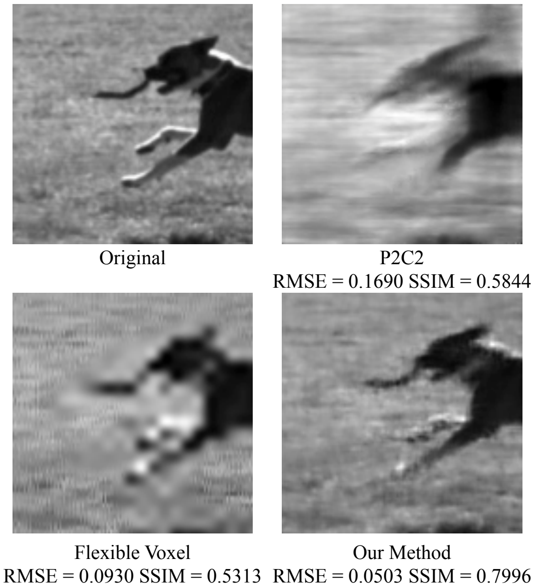

Original
P2C2
RMSE = 0.1690 SSIM = 0.5844
Flexible Voxel
RMSE = 0.0930 SSIM = 0.5313
Our Method
RMSE = 0.0503 SSIM = 0.7996

Fig. 9: Reconstruction results (32X gain) compared with other two methods, showing frame 6 out of 32. Compared with other two methods, our method can preserve more features both in the background and foreground.

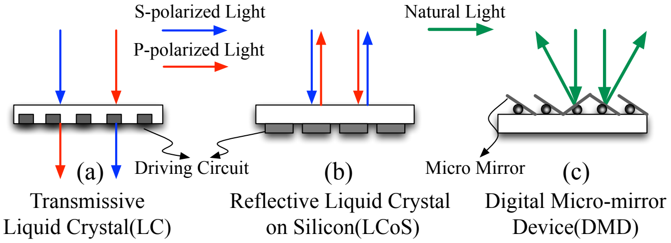


Fig. 10: Spatial Light Modulator (SLM). (a) and (b) modulate the light by changing its polarization state, (c) modulates the light by changing its direction.

7 Hardware Implementations

In this section, we show the details of our prototype imaging system. We first briefly describe the Spatial Light Modulator (SLM) and compare three types of popular SLMs. Then we introduce our prototype camera with per-pixel exposure control. Finally, we evaluate system characteristics for our prototype camera.

7.1 Overview of Spatial Light Modulator (SLM)

Our sampling function requires fast per-pixel modulation. Although we are not able to build a real image sensor with per-pixel exposure control due to the hardware limitation, we build an emulated imaging system using Spatial Light Modulator (SLM). SLM is a device that imposes spatially varying modulation on a beam of light. SLMs are extensively used in projection displays, but they can also be used as a component in optical computing. There are basically three types of SLMs, as shown in Fig. 10.

TABLE 2: Comparison of SLMs

	Transmissive LC	LCoS	DMD
Light Throughput	Low	Medium	High
Frame Rate	Low	Medium	High
Contrast	Low	Medium	High
Polarization	Yes	Yes	No
Pixel-wise Control	Difficult	Capable	Capable
Cost	Low	Medium	High

Figure 10(a) shows the transmissive Liquid Crystal (LC). It modulates the light by changing its polarization state, *i.e.*, when a pixel is turned “ON”, S-polarized light will be changed to a P-polarized light after going through that pixel. Nayar and Branzoi [31] build an adaptive dynamic range imaging system based on LCD. But this kind of device has some limitations. Because the device is transparent, and the driving circuits are located between the liquid crystal elements, this will reduce the fill factor for each pixel. Besides, the pattern generated on the LC is optically defocused by the imaging system and thus pixel-wise control could not be achieved. Finally, due to the diffraction effect produced by the LC cells, the captured images will also be blurred [25].

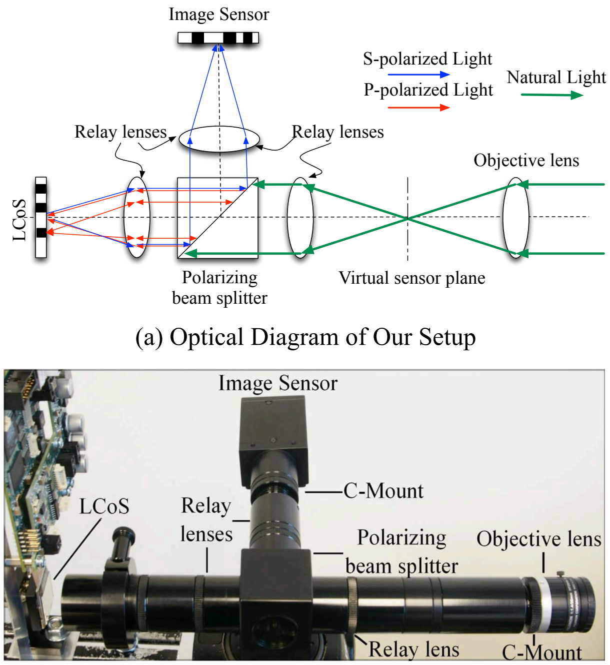
Another kind of LC device is called Liquid Crystal on Silicon (LCoS), which is a reflective liquid crystal device. Light modulation on this device is also based on polarization, but it is reflective instead of transmissive. As shown in Fig. 10(b), the driving circuit is located on the back side of the LC, thus the fill factor and contrast ratio are increased. By locating the LCoS on the virtual sensor plane of image sensor, pixel-wise control can be achieved in a relative compact imaging system [32, 6].

In order to modulate the light, both transmissive LC and LCoS need a polarizer. A polarizer will reduce the light by half. Combined with other optical components, the light throughput can be greatly reduced [33]. A DMD invented by Texas Instruments (TI) is a Micro-Electro-Mechanical System (MEMS) device that has a tiled micro mirror array, as shown in Fig. 10(c). Those mirrors can be individually tilted $\pm 10^\circ$ to an “ON” or “OFF” state. Therefore, light modulation is implemented by controlling the direction of the reflected light from those mirrors. The advantage of using DMD is that no polarizer is needed, and also the reflectivity of DMD mirror is higher than that of LCoS, so the light throughput of DMD should be higher. But since the modulation is achieved by tilting the micro mirror, DMD plane may be not parallel to the image sensor plane, thus lens aberration increases markedly[26].

Table 2 summarizes these three SLMs in different aspects including light throughput, frame rate, contrast *etc.*. In general, LCoS and DMD would be good choices for pixel-wise exposure control.

7.2 Our Prototype

In our prototype, we emulate the pixel-wise exposure control using an LCoS device. Fig. 11 illustrates our hardware setup. It consists of an image sensor (Point Grey Grasshopper 2, 1384×1036), an LCoS chip (ForthDD SXGA-3DM, 1280×1024), a polarizing beam-splitter, three relay lenses (Edmund Optics), and an objective lens



(a) Optical Diagram of Our Setup

(b) Imaging System Layout

Fig. 11: Our hardware prototype: optical diagram (top) and image (bottom) of our setup. Our system emulates fast per-pixel shutter using LCoS. The incident light is focused after the objective lens, and then becomes collimated after a relay lens and hits the polarizing beam splitter. S-polarized light is reflected away, only P-polarized light passes through. P-polarized light gets focused on the LCoS and then reflects back. The polarization state of the light changes (S-polarized becomes P-polarized and vice versa) when the LCoS pixel is “ON” (shown in white) and keeps the same when the LCoS pixel is “OFF” (shown in black). At last, only S-polarized light is reflected towards the image sensor and P-polarized light passes through.

(Computar M1614 16mm F1.4)(Please see supplemental material for detail system specification). The scene is first imaged on a virtual sensor plane through the objective lens, after passing through the polarizing beam splitter, S-polarized light is reflected downward, only P-polarized light passes through. An image is focused on the LCoS plane and reflected back. When an LCoS pixel is turned “ON”, the P-polarized light will be changed to S-polarized. When “OFF”, the polarization state will be the same (P-polarized). For the reflected light, only S-polarized light will be directed to the image sensor, P-polarized light will pass through the beam splitter. Therefore, the incident light is modulated by the LCoS.

The camera and LCoS are synchronized using a trigger signal from the LCoS. During a single camera exposure time, the LCoS displays several binary images, corresponding to the sampling function. We typically run the LCoS at $9 \sim 18$ times of the camera frame rate. For example, for an 18ms camera integration time ($55Hz.$), we operate the LCoS at $1000Hz.$, resulting in 18 video frames from a single coded exposure image.

7.3 System Characteristics

7.3.1 *Field of View*

As shown in Fig. 11, the relay system transfers the imaging sensor plane to the virtual sensor plane and also to the LCoS plane for light modulation. Since all the relay lenses have the same focal length, the magnification ratio is 1:1. Therefore, Field Of View (FOV) of our imaging system is the same as if the sensor were placed at the virtual sensor plane. The FOV can be calculated based on the sensor size and focal length of the objective lens:

$$FOV \approx 2arctan\frac{d}{2f_o}, \tag{4}$$

where d is the diagonal size of the image sensor, and f_o is the focal length of the objective lens.

Our prototype camera uses a 16mm objective lens and a 2/3“($8.8mm \times 6.6mm$) CCD sensor, so the FOV along horizontal and vertical directions are $30.75°$ and $23.31°$.

7.3.2 *Light Efficiency*

Light efficiency characterizes how much light is received by the image sensor after passing through the imaging system. Ideally, according to the specification of the LCoS and beam splitter, the light efficiency of the imaging system is:

$$27.5\% = 50\%(Polarization) \times 55\%(Reflectivity). \tag{5}$$

However, other components such as the relay lenses may also attenuate the intensity of the incident light. Therefore, the actual light efficiency would be lower than 27.5%. To measure the light efficiency of the imaging system, we capture two images of a uniform white scene. One with only the objective lens and relay lenses, and the other adds the polarizing beam splitter and LCoS. The ratio of the averaged pixel value of those two captured images is 21.88%, which represents the real light efficiency of the system.

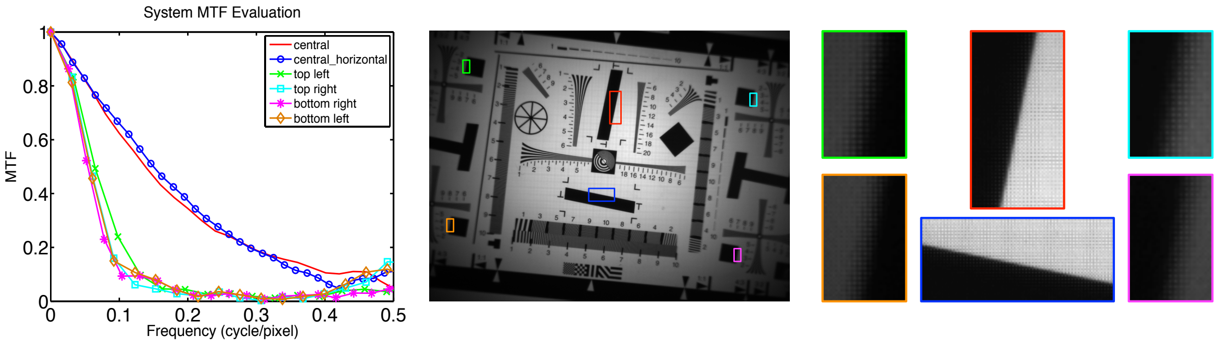
7.3.3 *MTF*

Modulation Transfer Function (MTF) is one of the most important index for an imaging system. MTF is the spatial frequency response of an imaging system, which describes how well the system is able to resolve image details as a function of spatial frequency. MTF can be calculated using the following equation:

$$MTF = \frac{M_o}{M_i}, M = \frac{I_{max} - I_{min}}{I_{max} + I_{min}}, \tag{6}$$

where M_o is the output modulation of the image, and M_i is the modulation of the input target.

We evaluate MTF by capturing an ISO 12233 target image and using the slanted edge method [34] to calculate MTF, as shown in Fig. 12. We select several regions of the image plane to calculate MTF (central region and corner region). The MTF curve in the central region is higher than those in the corner regions. The difference is caused by lens aberration. From the zoom in edge regions, we can clearly see that the edges near the corner are blurred, and the contrast of the edges are decreased.



(a) System MTF Plot (b) ISO 12233 Target (c) Zoom In Slanted Edge Regions

Fig. 12: MTF evaluation using the slanted edge method. (a) MTF curves on different regions of image plane. Central regions of the image plane have higher MTF. (b) ISO 12233 target that we use for measurement. (c) Zoom in of all six regions of edges. The edges in the central part are sharper compared with those in the corner.

7.3.4 Vignetting and Distortion

Vignetting is evaluated by taking an image of a white scene with uniform illumination. Vignetting is caused by insufficient light coming from the peripheral region. One way to reduce vignetting is to reduce the size of the aperture. The geometric distortion is calibrated using the camera calibration toolbox for Matlab. Detail results can be found in supplemental materials.

8 Experimental Results

Using our hardware prototype, we capture and reconstruct scenes comprising a wide range of motions, as shown in Fig. 13. The first example demonstrates the motion of an eye-lid during blinking. This motion is challenging as it involves occlusion and muscle deformations. The input frame is 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.

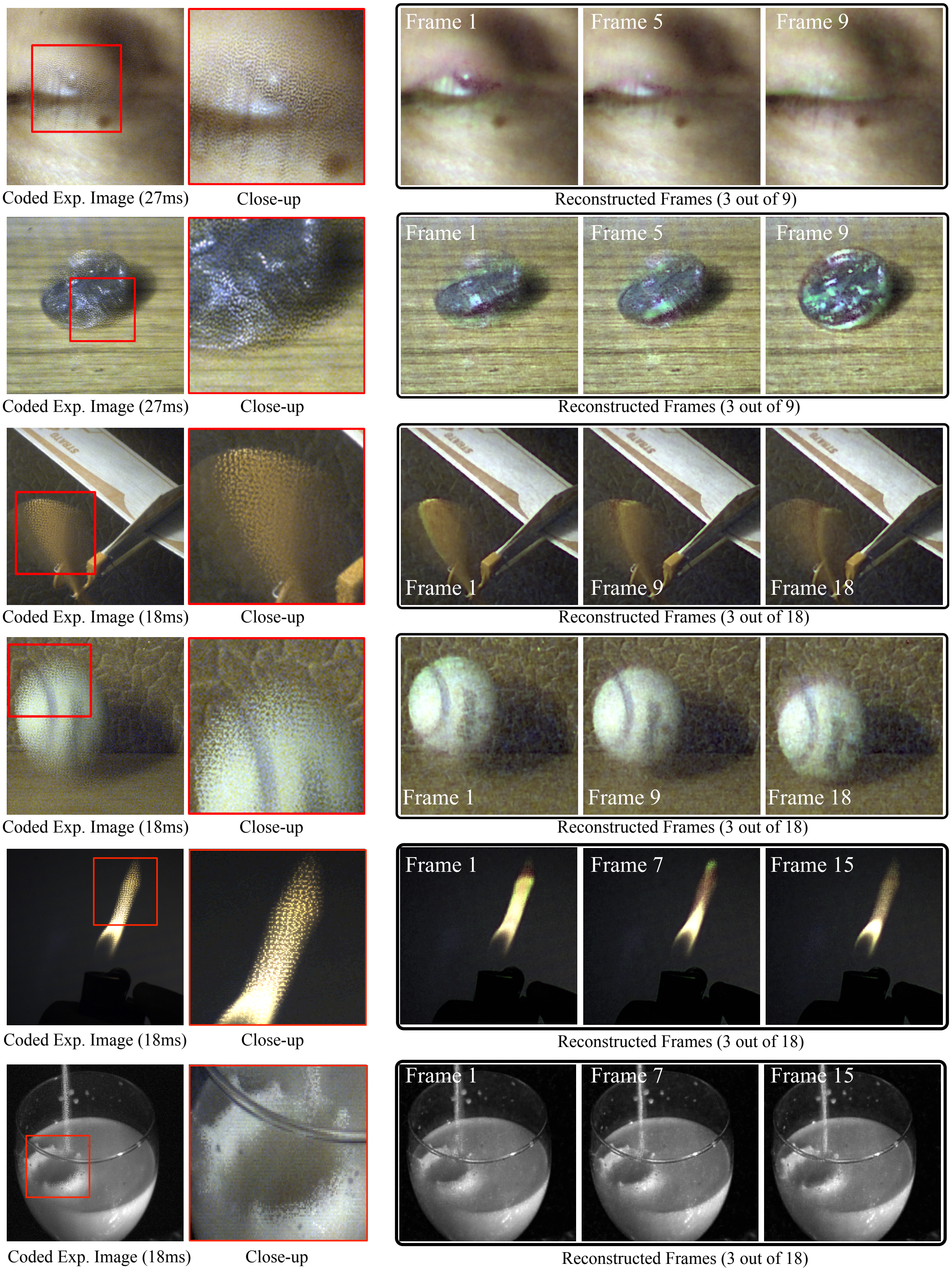
The second example shows a coin rotating on a table. This motion is challenging due to occlusions; as the coin rotates, one face of the coin becomes visible to the camera. From the single captured image, 9 output frames are reconstructed, while maintaining high spatial resolution, both on the coin and 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 perform the reconstruction at 1000 fps (18 output frames). The sharp edges of the blade and the texture on the ball are reconstructed in the output frames. The spatial detail on the static wings of the toy-plane are nearly the same as the input image. The fifth and sixth examples show the tongue of a flame and the milk drop crown. The subtle change of the flame tongue, as well as the complex fluid motion shown in milk drop, is faithfully reconstructed.

9 Discussion

In this paper, we propose an efficient way of capturing videos from a single photograph using pixel-wise coded exposure. We incorporate the hardware restrictions of existing image sensors into the design of the sampling functions, and implement 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 achieve sparse representations of space-time volumes for efficient reconstructions. We demonstrate the effectiveness of our method via extensive simulation comparison analysis and experiments. However, the proposed method has several limitations.

Software: First, the temporal resolution of the over-complete dictionary has to be pre-determined (*e.g.*, 36 frames). To do different scales of temporal upsampling, we have to train different dictionaries. Second, the reconstruction time for a video sequence of $450 \times 300 \times 36$ using a 10k dictionary bases is about 5 hours (HP Z600 workstation), which means that we cannot do real-time high-speed imaging. However, it would be beneficial for some applications such as collision detection where reconstruction can be done off-line.

Hardware: First, the maximum frame rate of LCoS determines the maximum temporal resolution of the reconstructed high speed video. For example, if the maximum frame rate of LCoS is 1000fps, we can only reconstruct a video of 1000fps at maximum. Second, since the image sensor and LCoS have different pixel sizes, one-to-one correspondence requires accurate geometric and photometric calibration. Incorrect calibration can cause artifact (ghosting) and also reduce the contrast of LCoS patterns. These artifacts can be significantly reduced once the per-pixel exposure control is implemented on image sensors in the near future. However, in order to achieve pixel-wise exposure control, the readout scheme needs to be redesigned, which is still challenging with current technology.



Coded Exp. Image (27ms) Close-up Reconstructed Frames (3 out of 9)

Coded Exp. Image (27ms) Close-up Reconstructed Frames (3 out of 9)

Coded Exp. Image (18ms) Close-up Reconstructed Frames (3 out of 18)

Coded Exp. Image (18ms) Close-up Reconstructed Frames (3 out of 18)

Coded Exp. Image (18ms) Close-up Reconstructed Frames (3 out of 18)

Coded Exp. Image (18ms) Close-up Reconstructed Frames (3 out of 18)

Fig. 13: 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-sixth 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, the stripe on the ball, the tongue of a flame and the milk drop crown.

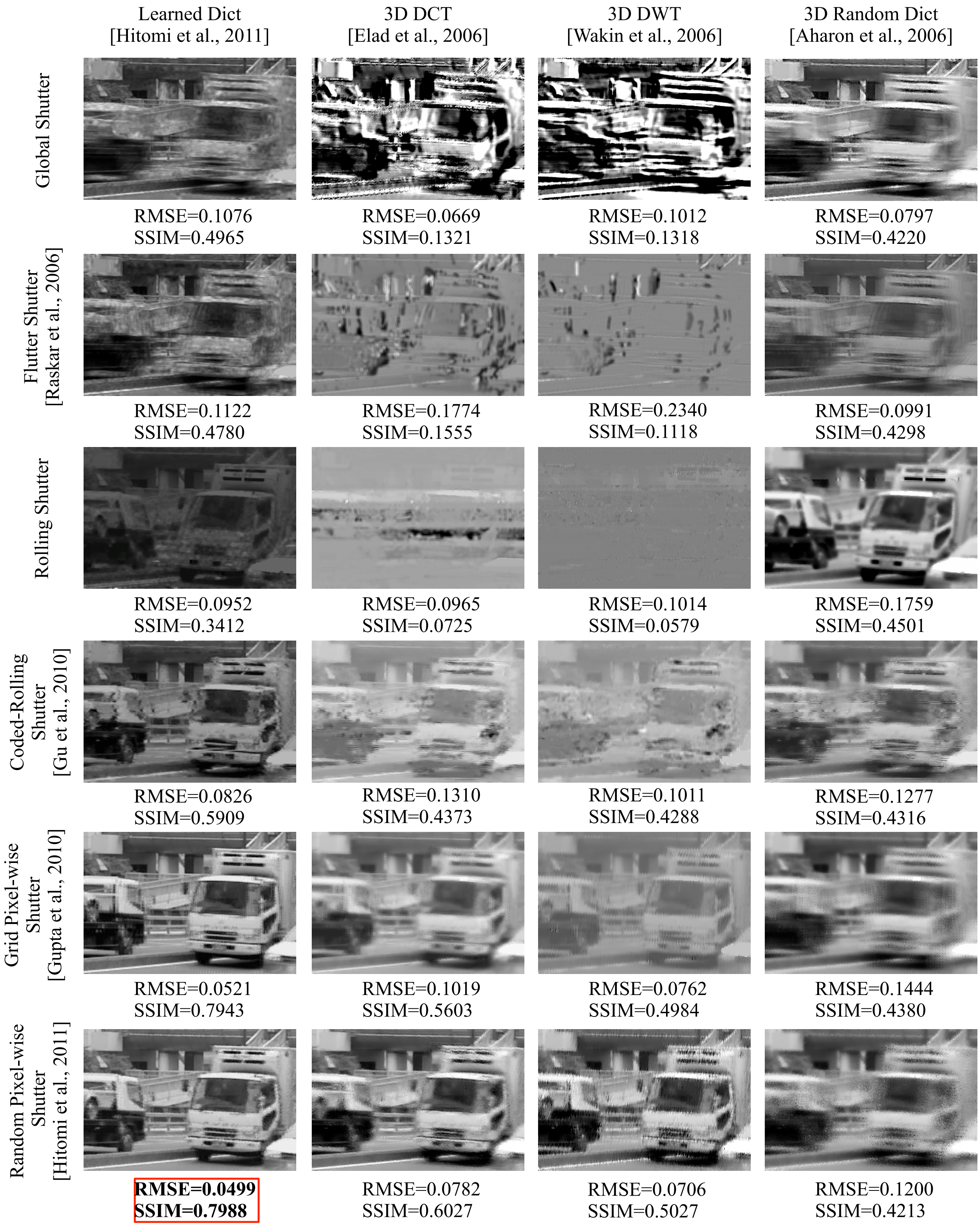


Fig. 14: Sampling functions versus representations . Horizontal direction shows reconstruction results (36X gain, frame 9 out of 36) for four dictionaries, combined with six exposure patterns along the vertical direction. Numerical analysis is given based on RMSE and SSIM. Notice that the combination of random pixel-wise shutter and learned dictionary has the best performance.

REFERENCES

[1] S. Kleinfelder, S. Lim, X. Liu, and A. El Gamal, "A 10000 Frames/s CMOS Digital Pixel Sensor," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 12, pp. 2049 –2059, 2001.

[2] "Teli Cameras." [Online]. Available: http://www.southimg.com/teli.html

[3] A. Gupta, P. Bhat, M. Dontcheva, O. Deussen, B. Curless, and M. Cohen, "Enhancing and Experiencing Space-Time Resolution with Videos and Stills," in *IEEE International Conference on Computational Photography (ICCP)*, 2009, pp. 1–9.

[4] G. Bub, M. Tecza, M. Helmes, P. Lee, and P. Kohl, "Temporal Pixel Multiplexing for Simultaneous High-Speed, High-Resolution Imaging," *Nature Methods*, vol. 7, 2010.

[5] M. Gupta, A. Agrawal, and A. Veeraraghavan, "Flexible Voxels for Motion-Aware Videography," in *European Conference on Computer Vision (ECCV)*, vol. 3, 2010, p. 6.

[6] D. Reddy, A. Veeraraghavan, and R. Chellappa, "P2C2: Programmable Pixel Compressive Camera for High Speed Imaging," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 329–336.

[7] D. L. Donoho, M. Elad, and V. N. Temlyakov, "Stable Recovery of Sparse Overcomplete Representations in the Presence of Noise," *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 6 – 18, 2006.

[8] E. J. Candes, J. Romberg, and T. Tao, "Stable Signal Recovery from Incomplete and Inaccurate Measurements," *Communications On Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[9] A. Veeraraghavan, D. Reddy, and R. Raskar, "Coded Strobing Photography: Compressive Sensing of High Speed Periodic Videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 33, no. 4, pp. 671–686, 2011.

[10] J. Y. Park and M. B. Wakin, "A Multiscale Framework for Compressive Sensing of Video," in *Picture Coding Symposium*, 2009, pp. 1–4.

[11] A. C. Sankaranarayanan, P. K. Turaga, R. G. Baraniuk, and R. Chellappa, "Compressive Acquisition of Dynamic Scenes," in *European Conference on Computer Vision (ECCV)*, vol. 6311, 2010, pp. 129–142.

[12] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[13] K. Yonemoto and H. Sumi, "A numerical analysis of a CMOS image sensor with a simple fixed-pattern-noise-reduction technology," *IEEE Transactions on Electron Devices*, vol. 49, pp. 746–753, 2002.

[14] "Sony Develops Next-generation Back-Illuminated CMOS Image Sensor." [Online]. Available: http://sony.net/SonyInfo/News/Press/201201/12-009E/

[15] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S. K. Nayar, "Video from a Single Coded Exposure Photograph using a Learned Over-Complete Dictionary," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 287–294.

[16] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz, "High-Speed Videography using a Dense Camera Array," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2004, pp. 294–301.

[17] 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 *Picture Coding Symposium*, 2006.

[18] M. Ben-Ezra and S. K. Nayar, "Motion-based motion deblurring," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2004.

[19] Y.-W. Tai, H. Du, M. Brown, and S. Lin, "Correction of spatially varying image and video motion blur using a hybrid camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 32, no. 6, pp. 1012 –1028, 2010.

[20] A. C. Sankaranarayanan, C. Studer, and R. G. Baraniuk, "CS-MUVI: Video Compressive Sensing for Spatial-Multiplexing Cameras," in *IEEE International Conference on Computational Photography (ICCP)*, 2012, pp. 1–10.

[21] R. Marcia, R. Willett, R. Marcia, and R. Willett, "Compressive Coded Aperture Video Reconstruction," in *European Signal Processing Conference*, vol. 2, 2008.

[22] R. Raskar, A. Agrawal, and J. Tumblin, "Coded Exposure Photography: Motion Deblurring using Fluttered Shutter," in *SIGGRPAH*, vol. 3, 2006.

[23] A. Agrawal, M. Gupta, A. Veeraraghavan, and S. G. Narasimhan, "Optimal Coded Sampling for Temporal Super-Resolution," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 599–606.

[24] J. Gu, Y. Hitomi, T. Mitsunaga, and S. K. Nayar, "Coded Rolling Shutter Photography: Flexible Space-Time Sampling," in *IEEE International Conference on Computational Photography (ICCP)*, 2010, pp. 1–8.

[25] S. K. Nayar, V. Branzoi, and T. E. Boult, "Programmable Imaging: Towards a Flexible Camera," *IEEE International Journal of Computer Vision*, vol. 70, no. 1, pp. 7–22, 2006.

[26] S. Ri, Y. Matsunaga, M. Fujigaki, T. Matui, and Y. Morimoto, "Development of DMD Reflection-Type CCD Camera for Phase Analysis and Shape Measurement," *Journal of Robotics and Mechatronics*, vol. 18, no. 6, p. 728, 2006.

[27] X. Shu and N. Ahuja, "Imaging via three-dimensional compressive sampling (3DCS)," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 439–446.

[28] M. Elad and M. Aharon, "Image Denoising Via

Learned Dictionaries and Sparse Representation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2006, pp. 895–900.

[29] J. A. Tropp and A. C. Gilbert, "Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

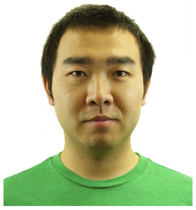
[30] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," in *IEEE Transactions on Image Processing*, vol. 13, no. 4, Apr 2004, pp. 600–612.

[31] S. K. Nayar and V. Branzoi, "Adaptive Dynamic Range Imaging: Optical Control of Pixel Exposures over Space and Time," in *IEEE International Conference on Computer Vision (ICCV)*, vol. 2, 2003, pp. 1168–1175.

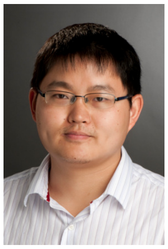
[32] H. Mannami, R. Sagawa, Y. Mukaigawa, T. Echigo, and Y. Yagi, "High Dynamic Range Camera using Reflective Liquid Crystal," in *IEEE International Conference on Computer Vision (ICCV)*, 2007, pp. 1–8.

[33] H. Nagahara, C. Zhou, T. Watanabe, H. Ishiguro, and S. K. Nayar, "Programmable Aperture Camera Using LCoS," in *European Conference on Computer Vision (ECCV)*, vol. 6316, 2010, pp. 337–350.

[34] P. D. Burns, "Slanted-Edge MTF for Digital Camera and Scanner Analysis," in *Proc. PICS Conf., IS & T*, 2000, p. 135.



Dengyu Liu received his B.E. degree in Electronic Science and Technology and M.E. degree in Optical Engineering from Beijing Institute of Technology, China, in 2008 and 2010, respectively. He is currently a doctoral student in the Center for Imaging Science, Rochester Institute of Technology. His research interests include computational imaging, computer vision.



Jinwei Gu is currently an assistant professor in the Munsell Color Science Laboratory in the Center for Imaging Science at Rochester Institute of Technology. He received his PhD degree from Columbia University in May 2010, and his bachelor and master degree from Tsinghua University, China in 2002 and 2005. His research interests are computer vision and computer graphics. His current research focuses on computational photography, physics-based computer vision, and data-driven computer graphics.

Yasunobu Hitomi received his B.E. in mechanical engineering and M.E. in information science from Tohoku University, Japan in 2001 and 2003, respectively. He has been working for Sony Corporation, Tokyo, Japan, since 2003. He was a visiting scholar with Prof. Shree Nayar at Columbia University from 2009 to 2011. His research area is computational photography, computer vision, and image processing.

Mohit Gupta is a research scientist in the CAVE Lab, Columbia University. He received his B.Tech. in Computer Science from the Indian Institute of Technology, New Delhi in 2003, M.S. in Computer Science from the Stony Brook University in 2005 and Ph.D. in Robotics from the Robotics Institute, CMU. His research interests are in physics-based computer vision, computational illumination and imaging and light transport. Details about his research can be found at http://www.cs.columbia.edu/ mohitg/Research.html

Tomoo Mitsunaga received the B.E. and M.E. degree in biophysical engineering from Osaka University, Japan, in 1989 and 1991, respectively. He has been working for Sony Corporation since 1991. He studied as a visiting scholar with Prof. Shree Nayar in Columbia University from 1997 to 1999. His interests include computer vision, digital image processing and computational cameras.

Shree K. Nayar received his PhD degree in Electrical and Computer Engineering from the Robotics Institute at Carnegie Mellon University in 1990. He is currently the T. C. Chang Professor of Computer Science at Columbia University. He co-directs the Columbia Vision and Graphics Center. He also heads the Columbia Computer Vision Laboratory (CAVE), which is dedicated to the development of advanced computer vision systems. His research is focused on three areas; the creation of novel cameras, the design of physics based models for vision, and the development of algorithms for scene understanding. His work is motivated by applications in the fields of digital imaging, computer graphics, and robotics. He has received best paper awards at ICCV 1990, ICPR 1994, CVPR 1994, ICCV 1995, CVPR 2000 and CVPR 2004. He is the recipient of the David Marr Prize (1990 and 1995), the David and Lucile Packard Fellowship (1992), the National Young Investigator Award (1993), the NTT Distinguished Scientific Achievement Award (1994), the Keck Foundation Award for Excellence in Teaching (1995), the Columbia Great Teacher Award (2006) and Carnegie Mellon University's Alumni Achievement Award. In February 2008, he was elected to the National Academy of Engineering.