

# Coded aperture compressive temporal imaging

Patrick Llull, Xuejun Liao, Xin Yuan, Jianbo Yang, David Kittle,  
Lawrence Carin, Guillermo Sapiro, and David J. Brady\*

Fitzpatrick Institute for Photonics, Department of Electrical and Computer Engineering  
Duke University, 129 Hudson Hall, Durham, North Carolina 27708, USA

*Corresponding author: dbrady@duke.edu*

**Abstract:** We use mechanical translation of a coded aperture for code division multiple access compression of video. We present experimental results for reconstruction at 148 frames per coded snapshot.

© 2013 Optical Society of America

**OCIS codes:** 110.6915

---

## References and links

1. D.J. Brady, M.E. Gehm, R.A. Stack, D.L. Marks, D.S. Kittle, D.R. Golish, E.M. Vera, and S.D. Feller. Multiscale gigapixel photography. *Nature*, 486(7403):386–389, 2012.
2. S. Kleinfelder, S.H. Lim, X. Liu, and A. El Gamal. A 10000 frames/s CMOS digital pixel sensor. *Solid-State Circuits, IEEE Journal of*, 36(12):2049–2059, 2001.
3. R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2, 2005.
4. D.J. Brady. *Optical Imaging and Spectroscopy*. Wiley-Interscience, 2009.
5. D.J. Brady, M. Feldman, N. Pitsianis, J.P. Guo, A. Portnoy, and M. Fiddy. Compressive optical MONTAGE photography. *Photonic Devices and Algorithms for Computing VII*, 5907(1):590708, 2005.
6. M. Shankar, N.P. Pitsianis, and D.J. Brady. Compressive video sensors using multichannel imagers. *Applied Optics*, 49(10):B9–B17, 2010.
7. 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 *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 287–294. IEEE, 2011.
8. D. Takhar, J.N. Laska, M.B. Wakin, M.F. Duarte, D. Baron, S. Sarvotham, K.F. Kelly, and R.G. Baraniuk. A new compressive imaging camera architecture using optical-domain compression. In *Electronic Imaging*, pages 606509–1–606509–10. International Society for Optics and Photonics, 2006.
9. M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K.F. Kelly, and R.G. Baraniuk. Compressive imaging for video representation and coding. In *Picture Coding Symposium*, 2006.
10. Y. Oike and AE Gamal. A  $256 \times 256$  CMOS image sensor with  $\delta\sigma$ -based single-shot compressed sensing. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, pages 386–388. IEEE, 2012.
11. M. Zhang and A. Bermak. CMOS image sensor with on-chip image compression: A review and performance analysis. *Journal of Sensors*, 2010, 2010.
12. A. Fish and O. Yadid-Pecht. *Circuits at the nanoscale: communications, imaging, and sensing*, chapter Low Power CMOS Imager Circuits. CRC Press, Inc., 2008.
13. V. Treeaporn, A. Ashok, and M.A. Neifeld. Space-time compressive imaging. *Applied Optics*, 51(4):A67–A79, 2012.
14. M.A. Neifeld and P. Shankar. Feature-specific imaging. *Applied Optics*, 42(17):3379–3389, 2003.
15. E.J. Candès and T. Tao. Reflections on compressed sensing. *IEEE Information Theory Society Newsletter*, 58(4):20–23, 2008.
16. D.L. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.
17. R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. *ACM Transactions on Graphics*, 25(3):795, 2006.
18. D. Reddy, A. Veeraraghavan, and R. Chellappa. P2C2: Programmable pixel compressive camera for high speed imaging. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 329–336. IEEE, 2011.

- 
19. A.C. Sankaranarayanan, C. Studer, and R.G. Baraniuk. CS-MUVI: Video compressive sensing for spatial-multiplexing cameras. In *Computational Photography (ICCP), 2012 IEEE International Conference on*, pages 1–10. IEEE, 2012.
  20. M.E. Gehm, R. John, D.J. Brady, R.M. Willett, T.J. Schulz, et al. Single-shot compressive spectral imaging with a dual-disperser architecture. *Opt. Express*, 15(21):14013–14027, 2007.
  21. X. Liao, H. Li, and L. Carin. Generalized alternating projection for weighted- $\ell_{2,1}$  minimization with applications to model-based compressive sensing. *Preprint submitted to SIAM Journal on Imaging Sciences*, 2012.
  22. J. M. Bioucas-Dias, and M. A. T. Figueiredo, “A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration,” *IEEE Transactions on Image Processing*, vol. 16, no. 12, 2992–3014 (2007).
- 

## 1. Introduction

Cameras capturing  $> 1$  gigapixel [1],  $> 10,000$  frames per second [2],  $> 10$  ranges [3] and  $> 100$  color channels [4] demonstrate that the optical data stream entering an aperture may approach 1 exapixel/second. While simultaneous capture of all optical information is not precluded by diffraction or quantum limits, this capacity is beyond the capability of current electronics. Previous compressive sampling proposals to reduce read-out bandwidth paradoxically increase system volume[5, 6] or operating bandwidth [7–9]. Here we use coded apertures to compress spatial and temporal sampling by  $> 4x$  and  $> 10x$ , respectively, without substantially increasing system volume or power. We describe computational estimation of 148 high-speed temporal frames from a single captured frame. By combining physical layer compression as demonstrated here with sensor-layer compressive sampling strategies [10, 11], and by using multiscale design to parallelize read-out, one may imagine streaming full exapixel data cubes over practical communications channels.

Under ambient illumination, cameras commonly capture 10-1000 pW per pixel, corresponding to  $10^9$  photons per second per pixel. At this flux, frame rates approaching  $10^6$  per second may still provide useful information. Unfortunately, frame rate generally falls well below this limit due to read-out electronics. The power necessary to operate an electronic focal plane is proportional to the rate at which pixels are read-out [12]. Current technology requires pW to n-W per pixel, implying 1-100W per sq. mm of sensor area for full data optical data cube capture. This power requirement cascades as image data flows through pipeline of read-out, processing, storage, communications, and display. Given the need to maintain low focal plane temperatures, this power density is unsustainable.

While interesting features persist on all resolvable spatial and temporal scales, the true information content of natural fields is much less than the photon flux because diverse spatial, spectral and temporal channels contain highly correlated information [13]. Under these circumstances, feature specific [14] and compressive [15, 16] multiplex measurement strategies developed over the past decade have been shown to maintain image information even when the number of digital measurement values is substantially less than the number of pixels resolved. Compressive measurement for visible imaging has been implemented using spatial light modulators (SLM) to code pixel values incident on a single detector [8, 9]. Unfortunately, this strategy increases, rather than decreases, operating power and bandwidth because the increased data load in the encoding signal is much greater than the decreased data load of the encoded signal. If the camera estimates  $F$  frames with  $N$  pixels per frame from  $M$  measurements, then  $MN$  control signals enter the modulator to obtain  $FN$  pixels. The bandwidth into the compressive camera exceeds the output bandwidth needed by a conventional camera by the factor  $NC$ , where  $C$  is the compression ratio. Unless  $C < 1/N$ , implying that the camera takes less than one measurement per frame, the control bandwidth exceeds the bandwidth necessary to fully read a conventional imager. This problem is slightly less severe in coding strategies derived from “flutter shutter” [17] motion compensation strategies. The flutter shutter uses full frame temporal modulation to encode motion blur for inversion. Several studies have implemented

per-pixel flutter shutter using spatial light modulators for video compression[7, 18, 19]. If we assume that these systems reconstruct at the full frame rate of the modulator, the control bandwidth is exactly equal to the bandwidth needed to read a conventional camera operating at the decompressed framerate. Alternatively, one may entirely avoid these problems by using parallel camera arrays with independent per pixel codes [5, 6] at the cost of increasing camera volume and cost by a factor of  $M$ .

Here we propose mechanical translation of a passive coded aperture for low power space-time compressive measurement. Coding is implemented by a chrome-on-glass binary transmission mask in an intermediate image plane. In contrast with previous approaches, modulation of the image data stream by harmonic oscillation of this mask requires no code transmission or operating power. We have previously used such masks for compressive imaging in coded aperture snapshot spectral imagers (CASSI) [20], which include an intermediate image plane before a spectrally dispersive relay optic. Here we demonstrate coded aperture compressive temporal imaging (CACTI). CASSI and CACTI share identical mathematical forward models. In CASSI, each plane in the spectral datacube is modulated by a shifted code. Dispersion through a grating or prism shifts spectral planes after coded aperture modulation. Detection integrates the spectral planes, but the datacube can be recovered by isolating each spectral plane based on its local code structure. This process may be viewed as code division multiple access (CDMA). In CACTI, translation of the coded aperture during exposure means that each temporal plane in the video stream is modulated by a shifted version of the code, thereby attaining per-pixel modulation using no additional sensor bandwidth.

Signal separation once again works by CDMA. We isolate the object's temporal channels from the compressed data by inverting a highly-underdetermined system of equations. By using an iterative reconstruction algorithm, we may estimate several high-speed video frames from a single coded measurement.

## 2. Theory

One may view CACTI's CDMA sensing process as uniquely patterning high-speed spatiotemporal object voxels  $f(x, y, t) \in \mathbb{R}^3$  with a transmission function that shifts in time (Fig. 1). Doing this applies distinct local coding structures to each temporal channel prior to integrating the channels as limited-framerate images  $g(x', y', t') \in \mathbb{R}^2$  on the  $N$ -pixel detector. An  $N_F$ -frame, high-speed estimate of  $f(x, y, t)$  may be reconstructed from each low-speed coded snapshot  $g(x', y', t')$ , with  $t' < t$ .

Considering only 1 spatial dimension ( $(x, y) \rightarrow x$ ) and respectively denoting object-and image-space coordinates with unprimed and primed variables, the sampled data  $g(x', t')$  consists of discrete samples of the continuous transformation [4]

$$g(x', t') = \int_1^{N_F} \int_1^N f(x, t) T(x - s(t)) \text{rect}\left(\frac{x - x'}{\Delta_x}\right) \text{rect}\left(\frac{t - t'}{\Delta_t}\right) dx dt, \quad (1)$$

where  $T(x - s(t))$  represents the transmission function of the coded aperture,  $\Delta_x$  is the detector pixel size,  $\text{rect}(\frac{x}{\Delta_x})$  is the pixel sampling function and  $\Delta_t$  is the temporal integration time.  $s(t)$  describes the coded aperture's spatial position during the camera's integration window.

One may analyze the expected temporal resolution of the coded data by considering the Fourier transform of Eq. (1). Assuming the coded aperture moves linearly during  $\Delta_t$  such that  $s(t) = vt$ , the image's temporal spectrum is given by

$$\hat{g}(u, v) = \text{sinc}(u\Delta_x) \text{sinc}(v\Delta_t) \int \hat{f}(u - w, v - vw) \hat{T}(w) dw, \quad (2)$$

where  $\hat{f}(u, v)$  is the 2D Fourier transform of the space-time datacube and  $\hat{T}(w)$  is the 1D

Fourier transform of the spatial code. Without the use of the coded aperture,  $\hat{g}(u, v) = \text{sinc}(u\Delta_x)\text{sinc}(v\Delta_t)\hat{f}(u, v)$  and the sampled data stream is proportional to the object video low-pass filtered by the pixel sampling functions. Achievable resolution is proportional to  $\Delta_x$  in  $x$  and  $\Delta_t$  in time. The moving code aliases higher frequency components of the object video into the passband of the detector sampling functions. The support of  $\hat{T}(w)$  extends to some multiple of the code feature size  $\Delta_c$  (in units of detector pixels), meaning that the effective passband may be increased by a factor proportional to  $1/\Delta_c$  in space and  $v/\Delta_c$  in time. In practice, finite mechanical deceleration times cause  $\hat{T}(w)$  to have significant DC and low-frequency components in addition to the dominant  $v = \frac{C}{\Delta_t}$ ; hence, high and low frequencies alike are aliased into the system's passband.

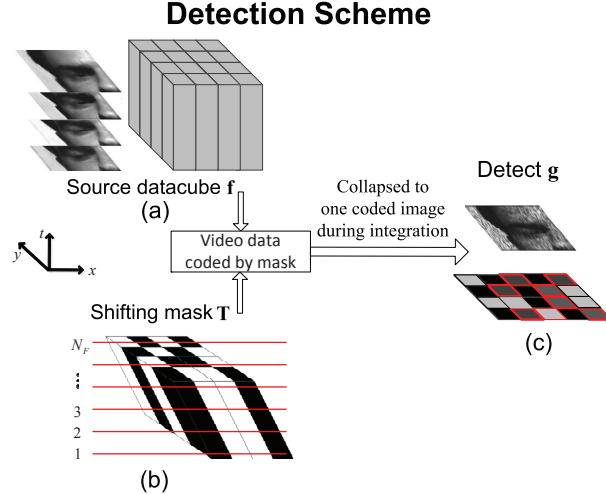


Fig. 1. Detection process. (a) A discrete space-time source datacube is (b) multiplied at each of  $N_F$  temporal channels with a shifted version of a coded aperture pattern. (c) Each detected frame  $g$  is the summation of the coded temporal channels and contains the object's spatiotemporal-multiplexed information. The dark grey (red-outlined) and black detected pixels in (c) pictorially depict the code's location at the beginning and the end of the camera's integration window, respectively.

Considering a square  $N$ -pixel active sensing area, the discretized form of the three-dimensional scene is  $\mathbf{f} \in \mathbb{R}^{\sqrt{N} \times \sqrt{N} \times N_F}$ , i.e. a  $(\sqrt{N} \times \sqrt{N}) \times N_F$ -voxel spatiotemporal datacube. In CACTI, a time-varying spatial transmission pattern  $\mathbf{T} \in \mathbb{R}^{\sqrt{N} \times \sqrt{N} \times N_F}$  uniquely codes each of the  $N_F$  temporal channels of  $\mathbf{f}$  prior to integrating them into one detector image  $\mathbf{g} \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}}$  during  $\Delta_t$ . These measurements at spatial indices  $(i, j)$  and temporal index  $k$  are given by

$$g_{i,j} = \sum_{k=1}^{N_F} T_{i,j,k} f_{i,j,k} + n_{i,j}, \quad (3)$$

where  $n_{i,j}$  represents imaging noise at the  $(i, j)^{th}$  pixel. One may rasterize the discrete object  $\mathbf{f} \in \mathbb{R}^{NN_F \times 1}$ , image  $\mathbf{g} \in \mathbb{R}^{N \times 1}$ , and noise  $\mathbf{n} \in \mathbb{R}^{N \times 1}$  to obtain the linear transformation given by

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}, \quad (4)$$

where  $\mathbf{H} \in \mathbb{R}^{N \times NN_F}$  is the system's *discrete forward matrix* that accounts for sampling factors including the optical impulse response, pixel sampling function, and time-varying transmission

function. The forward matrix is a 2-dimensional representation of the 3-dimensional transmission function  $\mathbf{T}$ :

$$\mathbf{H}_k \stackrel{\text{def}}{=} \text{diag} \left[ T_{1,1,k} \ T_{2,1,k} \ \cdots \ T_{\sqrt{N},\sqrt{N},k} \right], \quad k = 1, \dots, N_F; \quad (5)$$

$$\mathbf{H} \stackrel{\text{def}}{=} [\mathbf{H}_1 \ \mathbf{H}_2 \ \cdots \ \mathbf{H}_{N_F}], \quad (6)$$

where  $\mathbf{H}_k \in \mathbb{R}^{N \times N}$  is a matrix containing the entries of  $\mathbf{T}_k$  along its diagonal and  $\mathbf{H}$  is a concatenation of all  $\mathbf{H}_k$ ,  $k \in \{1, \dots, N_F\}$ . Fig. 2 underlines the role  $\mathbf{H}$  plays in the linear transformation.

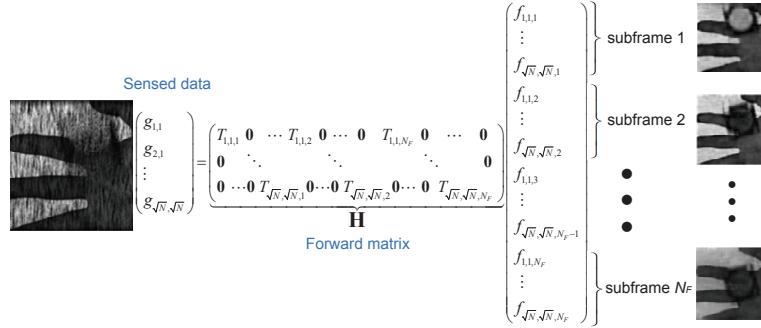


Fig. 2. Linear system model.  $N_F$  subframes of high-speed data  $\mathbf{f}$  are estimated from a single snapshot  $\mathbf{g}$ . The forward model matrix  $\mathbf{H}$  has many more columns than rows and has dimensions  $N \times (N \times N_F)$ .

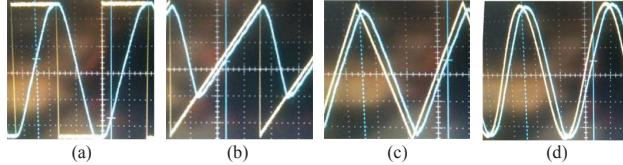


Fig. 3. Waveform choices for  $s(t)$ . Yellow: signal from function generator. Blue: actual hardware motion. Note the poor mechanical response to sharp rising/falling edges in (a) and (b). The sine wave (d) is unpreferable because of the nonuniform exposure time of different  $\mathbf{T}_k$ .

At the  $k^{\text{th}}$  temporal channel, the coded aperture's transmission function  $\mathbf{T}$  is given by

$$\mathbf{T}_k = \text{Rand}(\sqrt{N}, \sqrt{N}, s_k), \quad (7)$$

where  $\text{Rand}(m, n, p)$  denotes an  $m \times n$  random binary matrix shifted vertically by  $p$  pixels (optimal designs could be considered for this system as well).  $s_k$  discretely approximates  $s(t)$  at the  $k^{\text{th}}$  temporal channel by

$$s_k = C \text{Tri} \left[ \frac{k}{2\Delta_t} \right] \quad (8)$$

where  $C$ , the system's compression ratio, is the amplitude traversed by the code in units of detector pixels.  $\text{Tri} \left[ \frac{k}{2\Delta_t} \right]$  represents a discrete triangle wave signal of twice the integration time periodicity. The camera integrates during the  $C$ -pixel sweep of the coded aperture on the image plane and detects a linear combination of  $C$  uniquely-coded temporal channels of  $f$ .

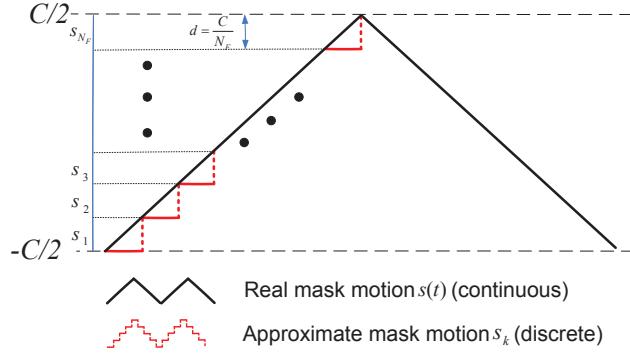


Fig. 4. Continuous motion and discrete approximation to coded aperture movement during integration time. The discrete triangle function  $s_k$  more-accurately approximates the continuous triangle wave driving the mask with smaller values of  $d$  but adds more columns to  $\mathbf{H}$ .

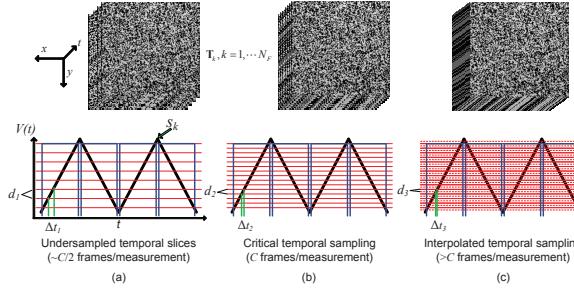


Fig. 5. Temporal channels used for the reconstruction. Red lines indicate which subset of transverse mask positions  $s_k$  were utilized to construct the forward matrix  $\mathbf{H}$ . Blue lines represent the camera integration windows. (a) Calibrating with fewer  $s_k$  results in a better-posed inverse problem but doesn't as closely approximate the temporal motion  $s(t)$ . (b) With  $d = 1$ , each pixel integrates several unique coding patterns with a temporal separation  $\Delta t = C^{-1}$ . (c) Constructing  $\mathbf{H}$  with large  $N_F$  ( $d < 1$ ) interpolates the motion occurring between the uniquely-coded image frames.

Periodic triangular motion lets us use the same discrete forward matrix  $\mathbf{H}$  to reconstruct any given snapshot  $\mathbf{g}$  within the acquired video while adhering to the hardware's mechanical acceleration limitations (Fig. 3). The discrete motion  $s_k$  (Eq. (8)) closely approximates the analog triangle waveform supplied by the function generator (Fig. 4).

Let  $d$  represent the number of *detector* pixels the mask moves between adjacent temporal channels  $s_k$  and  $s_{k+1}$ .  $N_F$  frames are reconstructed from a single coded snapshot given by

$$N_F = \frac{C}{d}, \quad (9)$$

thus, altering  $d$  will affect the number of reconstructed frames for given a compression ratio  $C$ .

In the case of  $d = 1$  (i.e.  $N_F = C$ ), the detector pixels that sense the continuous, temporally-modulated object  $f$  are *critically-encoded*; each pixel integrates a series of nondegenerate mask patterns (Fig. 5(b)) during  $\Delta_t$ .

When  $d < 1$  (i.e.  $N_F > C$ ), every  $\frac{1}{d}$  temporal channels of  $\mathbf{H}$  will contain nondegenerate temporal code information. These channels will reconstruct as if the sensing pixels are critically-encoded. The other temporal slices will interpolate the motion *between* critically-encoded temporal channels (Fig. 5(c)). Generally, this interpolation accurately estimates the direction of the motion between these critically-encoded states but retains most of the residual motion blur.

### 3. Experimental Hardware

The experimental prototype camera (Fig. 6) consists of a 50mm camera objective lens (Compuitar), a lithographically-patterned chrome-on-quartz coded aperture with anti-reflective coating for visible wavelengths [20] (Eq. (7)) mounted upon a piezoelectric stage (Newport Co.), an  $F/8$  achromatic relay lens (Edmund Optics), and a  $640 \times 480$  FireWire IEEE 1394a monochrome CCD camera (Marlin AVT).

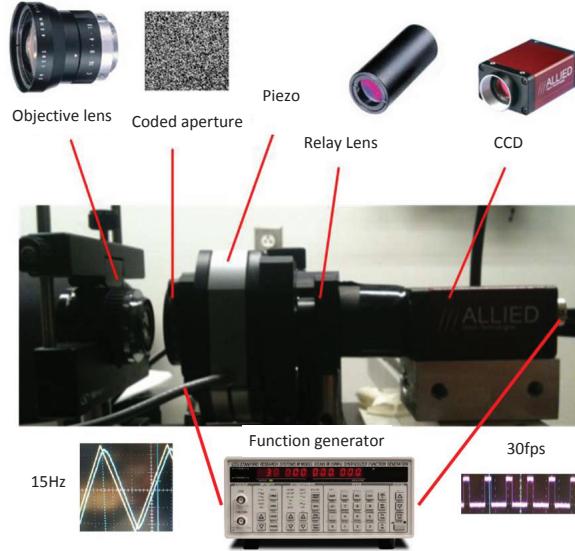


Fig. 6. CACTI Prototype hardware setup. The coded aperture is  $5.06\text{mm} \times 4.91\text{mm}$  and spans  $248 \times 256$  detector pixels. The function generator moves the coded aperture and triggers camera acquisition with signals from its function and SYNC outputs, respectively.

The objective lens images the continuous scene  $f$  onto the piezo-positioned mask. The func-

tion generator (Stanford Research Systems DS345) drives the piezo with a 10V pk-pk, 15Hz triangle wave to locally code the image plane while the camera integrates. We operate at this low frequency to accommodate the piezo's finite mechanical deceleration time.

To ensure the CDMA process remains time-invariant, we use the function generator's SYNC output to generate a 15Hz square wave. We frequency-double this signal using an FPGA device (Altera) to trigger camera integrations once along the mask's upward and downward motion (Fig. 5).

The relay lens images the spatiotemporally modulated scene onto the camera, which saves the 30fps coded snapshots to a local computer.  $N_F$  video frames of the discrete scene  $\mathbf{f}$  are later reconstructed from each coded image  $\mathbf{g}$  offline by the Generalized Alternating Projection (GAP) [21] algorithm.

During  $\Delta_t$ , the piezo can move a range of  $0 - 160\mu\text{m}$  vertically in the  $(x,y)$  plane. Using  $158.4\mu\text{m}$  of this stroke moves the coded aperture eight  $19.8\mu\text{m}$  elements (sixteen  $9.9\mu\text{m}$  detector pixels) during each camera integration period  $\Delta_t$ . Using larger strokes for a given modulation frequency is possible and would increase  $C$ .

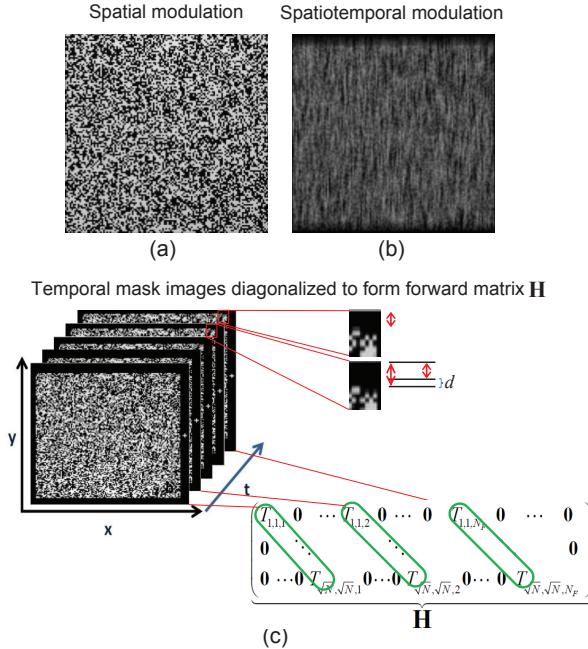


Fig. 7. Spatial and temporal modulation. (a) A stationary coded aperture spatially modulates the image. (b) Moving the coded aperture during the integration window applies local code structures to  $N_F$  temporal channels, effectively shearing the coded space-time datacube and providing per-pixel flutter shutter. (c) Imaging the (stationary) mask at positions  $d$  pixels apart and storing them into the forward matrix  $\mathbf{H}$  simulates the mask's motion, thereby conditioning the inversion.

Importantly, using a piezoelectric stage itself is not the optimal solution to translate a coded aperture during  $\Delta_t$ . This device was preferable for the hardware prototype because of its precision and convenient built-in Matlab interface. However, a low-resistance spring system could, in principle, serve the same purpose while using very little power.

### 3.1. Forward Model Calibration

We calibrate the system forward model by imaging the aperture code under uniform, white illumination at discrete spatial steps  $s_k$  according to Eq. (8). Steps are  $d$  detector pixels apart over the coded aperture's range of motion (Figs. 7(c), 5.). This accounts for system misalignments and relay-side aberrations. A Matlab routine controls the piezoelectric stage position during calibration. Since Matlab cannot generate a near-analog waveform for continuous motion, we connect the piezoelectric motion controller to the function generator via serial port during experimental capture.

We use an active area of  $281 \times 281$  detector pixels to account for the  $248 \times 256$ -pixel coded aperture's motion  $s_k$  with additional zero-padding. We choose  $d = \frac{\Delta x}{10}$  to provide a substantial basis with which to construct the forward model while remaining well above the piezo's 0.0048 pixel RMS jitter. Storing every temporal channel spaced  $d = 0.99\mu m$  apart into  $\mathbf{H}$  results in  $N_F = 160$  reconstructed frames.

When reconstructing, we may diagonalize any subset of temporal slices of the  $281 \times 281 \times 160$  set of mask images into the forward model (Fig. 7). We found the optimum subset of mask positions within this 160-frame set of  $s_k$  through iterative  $\|\mathbf{g} - \mathbf{H}\mathbf{f}_e\|_2$ -error reconstruction tests, where  $\mathbf{f}_e$  is GAP's  $N_F$ -frame estimate of the continuous motion  $f$ . From these tests, we chose and compared two numbers of frames to reconstruct per measurement,  $N_F = C = 14$  and  $N_F = 148$ .  $\mathbf{H}$  has dimensions  $281^2 \times (281^2 \times N_F)$  for both of these cases.

As seen in Figs. 10-13, decreasing  $d$  and estimating up to 148 frames from a single exposure  $\mathbf{g}$  does not significantly reduce the aesthetic quality of the inversion results, nor does it significantly affect the residual error (Fig. 8(b)). The reconstruction time increases approximately linearly with  $N_F$  as shown in Fig. 8(a).

## 4. Reconstruction Algorithm

Since  $\mathbf{H}$  multiplexes many local code patterns of the continuous object to the discrete-time image  $\mathbf{g}$ , inverting Eq. (4) for  $\mathbf{f}$  becomes difficult as  $N_F$  increases. Least-squares, pseudoinverse, and other linear inversion methods cannot accurately reconstruct such underdetermined systems. We use an iterative reconstruction algorithm called Generalized Alternating Projection (GAP) that exploits image and video models (priors) to effectively solve this ill-posed inverse problem [21].

GAP takes advantage of the structural sparsity of the subframes in transform domains such as wavelets and discrete cosine transform (DCT). Fig. 9 illustrates the underlying principle of GAP. It is worth noting that GAP is based on the volume's global sparsity and requires no training whatsoever. In other words, GAP is a universal reconstruction algorithm insensitive to data being inverted.

The GAP algorithm makes use of Euclidean projections on two convex sets, which respectively enforce data fidelity and structural sparsity. Please refer to [21] for details. Furthermore, GAP is an anytime algorithm; the results produced by the algorithm converge monotonically to the true value as the computation proceeds. The monotonicity has been generally observed in our extensive experiments and theoretically established under a set of sufficient conditions on the forward model. The reconstructed subframes continually improve over successive iterations and the user can halt computation at anytime to obtain intermediate results. The user may then continue improving reconstruction by resuming the computation. The following briefly reviews the main steps of the GAP algorithm [21].

### 4.1. The Linear Manifold

Data fidelity is ensured by projection onto the linear manifold  $\Pi = \mathbf{f} : \{\sum_{k=1}^{N_F} H_{i,j,k} f_{i,j,k} + n_{i,j} = g_{i,j}, \forall (i, j)\}$ , which consists of all legitimate high-speed frames of  $\mathbf{f}$  integrated onto the detector

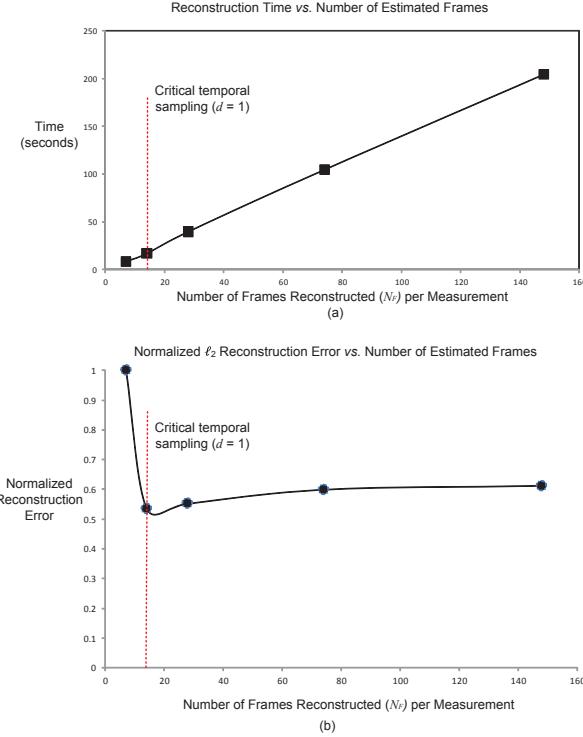


Fig. 8. Algorithm convergence times and relative residual reconstruction errors for various compression ratios. (a) GAP's reconstruction time increases linearly with data size. Tests were performed on a ASUS U46E laptop (Intel quad core I7 operated at 3.1GHz). (b) Normalized  $\ell_2$  reconstruction error vs. number of reconstructed frames. The residual error reaches a minimum at critical temporal sampling and gradually flattens out with finer temporal interpolation (lower  $d$ ).

via Eq. (4). In other words,  $\Pi$  is a set of solutions to the underdetermined system of linear equations which are disambiguated by using structural sparsity.

#### 4.2. The Weighted $\ell_{2,1}$ Ball

Structural sparsity is encoded by  $G = \{G_1, G_2, \dots, G_m\}$ , a partition of the indices  $\{(i, j, k)\}$  that span the voxels of  $\mathbf{f}$ , and the associated weights  $\beta = \{\beta_l : \beta_l > 0, l = 1, 2, \dots, m\}$ . The weighted  $\ell_{2,1}$  ball is defined as  $\Lambda(R) = \left\{ \mathbf{f} : \|Q(\mathbf{f})\|_{\ell_{2,1}^{G\beta}} \leq R \right\}$ , where  $Q(\mathbf{f})$  is an orthonormal transform (wavelet transformation or Discrete Cosine Transformation (DCT)) of the volume  $\mathbf{f}$ ,

$$Q(\mathbf{f})_{i,j,k} = \sum_{i',j',k'} Q_1(i, i') Q_2(j, j') Q_3(k, k') f_{i',j',k'},$$

with  $Q_1, Q_2, Q_3$  orthonormal matrices, and

$$\|\mathbf{w}\|_{\ell_{2,1}^{G\beta}} = \sqrt{\sum_{l=1}^m \beta_l \sum_{(i,j,k) \in G_l} w_{i,j,k}^2}$$

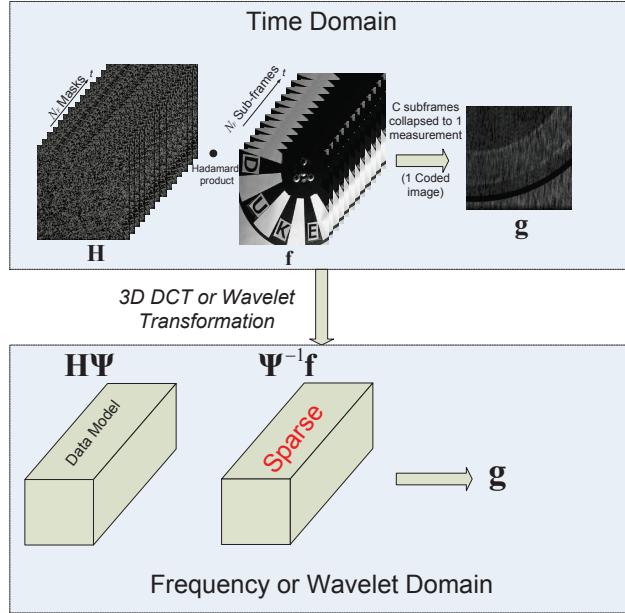


Fig. 9. Illustration of the GAP algorithm.

is a weighted  $\ell_{2,1}$  norm of  $\mathbf{w}$ . Note that  $\Lambda(R)$  is constructed as a weighted  $\ell_{2,1}$  ball in the space of transform coefficients  $\mathbf{w} = Q(\mathbf{f})$ , since structural sparsity is desired for the coefficients instead of voxels. The ball is rotated in the voxel space due to the orthonormal transform  $Q(\cdot)$ .

#### 4.3. Euclidean Projections

The Euclidean projection of  $\theta$  onto  $\Pi$ ,

$$P_\Pi(\theta) = \arg \min_{f: f \in \Pi} \sum_{i,j,k} (\theta_{i,j,k} - f_{i,j,k})^2,$$

is given by

$$f_{i,j,k} = \theta_{i,j,k} + \frac{H_{i,j,k}}{\sum_{k'=1}^N H_{i,j,k'}^2} \left( g_{i,j} - \sum_{k'=1}^N \theta_{i,j,k'} H_{i,j,k'} \right). \quad (10)$$

The Euclidean projection of  $\mathbf{f}$  onto  $\Lambda(R)$ ,

$$P_{\Lambda(R)}(\mathbf{f}) = \arg \min_{\theta: \theta \in \Lambda(R)} \sum_{i,j,k} (\theta_{i,j,k} - f_{i,j,k})^2,$$

can be equivalently written as

$$P_{\Lambda(R)}(\mathbf{f}) = Q^{-1} \left( \arg \min_{\theta: \|\theta\|_{\ell_{2,1}^G} \leq R} \sum_{i,j,k} (\theta_{i,j,k} - Q(\mathbf{f})_{i,j,k})^2 \right).$$

using that fact that Euclidean distance is invariant to the orthonormal transform  $Q(\cdot)$ . We are only interested in  $P_{\Lambda(R)}(\mathbf{f})$  when  $R$  takes the special values considered below.

#### 4.4. Alternating Projection between $\Pi$ and $\Lambda(R)$ when $R$ Systematically Changes

The GAP algorithm is a sequence of Euclidean projections between a linear manifold and a weighted  $\ell_{2,1}$  ball that undergoes a systematic change in size. Starting from  $\theta^{(0)} = 0$ , the GAP algorithm iterates between the following two steps, until  $\|\mathbf{f}^{(t)} - \theta^{(t)}\|_{\ell_2}$  converges in  $t$ .

- 1) Projection on the linear manifold,

$$\mathbf{f}^{(t)} = P_{\Pi}(\theta^{(t-1)}), \quad t \geq 1,$$

with the solution given in Eq. (10).

- 2) Projection on the weighted  $\ell_{2,1}$  ball of changing size,

$$\theta^{(t)} = P_{\Lambda(R^{(t)})}(\mathbf{f}^{(t)}), \quad t \geq 1,$$

where

$$R^{(t)} = \sum_{q=1}^{m^*} \beta_{l_q}^2 \left( \frac{\sqrt{\sum_{(i,j,k) \in G_{l_q}} Q(\mathbf{f}^{(t)})_{i,j,k}^2}}{\beta_{l_q}} - \frac{\sqrt{\sum_{(i,j,k) \in G_{l_{m^*+1}}} Q(\mathbf{f}^{(t)})_{i,j,k}^2}}{\beta_{l_{m^*+1}}} \right),$$

$(l_1, l_2, \dots, l_m)$  is a permutation of  $(1, 2, \dots, m)$  such that

$$\frac{\sqrt{\sum_{(i,j,k) \in G_{l_q}} Q(\mathbf{f}^{(t)})_{i,j,k}^2}}{\beta_{l_q}} \geq \frac{\sqrt{\sum_{(i,j,k) \in G_{l_{q+1}}} Q(\mathbf{f}^{(t)})_{i,j,k}^2}}{\beta_{l_{q+1}}}$$

holds for any  $q \leq m-1$ , and  $m^* = \min\{z : \text{cardinality}(\cup_{q=1}^z G_{l_q}) \geq \text{cardinality}(g)\}$ . The projection is given by  $\theta^{(t)} = Q^{-1}(\vartheta^{(t)})$  where

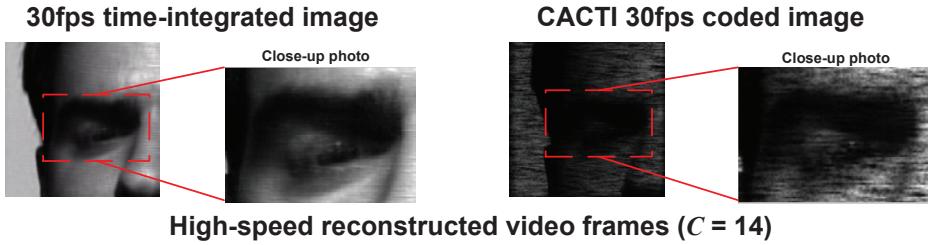
$$\vartheta_{i,j,k}^{(t)} = Q(\mathbf{f}^{(t)})_{i,j,k} \max \left\{ 1 - \frac{\beta_l \sqrt{\sum_{(i,j,k) \in G_{l_{m^*+1}}} Q(\mathbf{f}^{(t)})_{i,j,k}^2}}{\beta_{l_{m^*+1}} \sqrt{\sum_{(i,j,k) \in G_l} Q(\mathbf{f}^{(t)})_{i,j,k}^2}}, 0 \right\}, \quad \forall (i,j,k) \in G_l.$$

## 5. Results

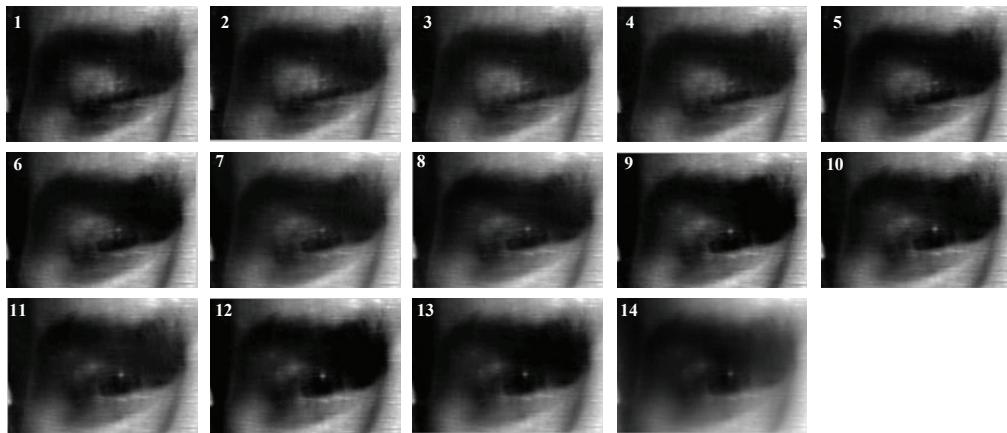
CACTI's experimental temporal superresolution results are shown in Figs. 10-13 (see Media 1-8 for the complete videos). An eye blinking, a lens falling in front of a hand, a chopper wheel with the letters 'DUKE' placed on the blades, and a bottle pouring water into a cup are captured at 30fps and reconstructed with  $N_F = C = 14$  and  $N_F = 148$ . There is little aesthetic difference between these reconstructions. In some cases, as with the lens and hand reconstruction,  $N_F = 148$  appears to yield additional temporal information over  $N_F = 14$ . The upper-left images depict the sum of the reconstructed frames, showing the expected time-integrated snapshots acquired with a 30fps video camera lacking spatiotemporal image plane modulation.

Note that several of these features, particularly the water pouring, are hardly visible among the moving code pattern. Objects exhibiting large temporal motion blur were reconstructed with GAP using DCT bases, while wavelet bases were used for the stationary reconstructions. Click on the media objects files to view the complete reconstructed videos.

The compression ratio  $C$  is 14 rather than 16 because the triangle wave's peak and trough ( $\mathbf{T}_{s1}$  and  $\mathbf{T}_{s16}$ ) are not accurately characterized by linear motion due to the mechanical deceleration time and were hence not placed into  $\mathbf{H}$  to reduce model error.



**14 frames reconstructed**



**148 frames reconstructed**

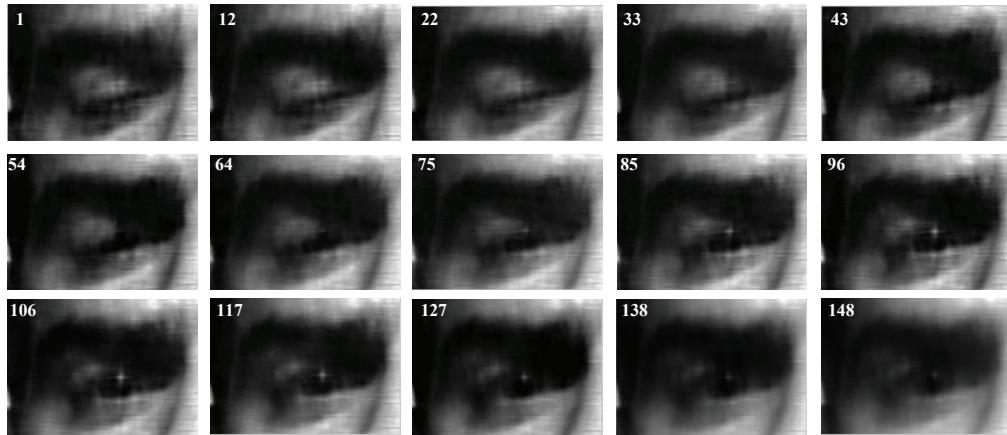
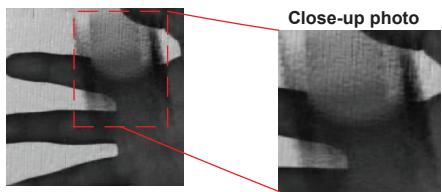
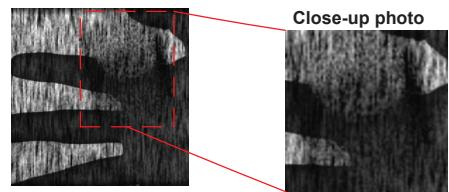


Fig. 10. High-speed ( $C = 14$ ) video of an eye blink, from closed to open, reconstructed from a single coded snapshot for  $N_F = 14$  ([Media1.mov](#)) and  $N_F = 148 ([Media2.mov](#)). The numbers on the bottom-right of the pictures represent the frame number of the video sequence. Note that the eye is the only part of the scene that moves. The top left frame shows the sum of these reconstructed frames, which approximates the motion captured by a 30fps camera without a coded aperture modulating the focal plane.$

**30fps time-integrated image**

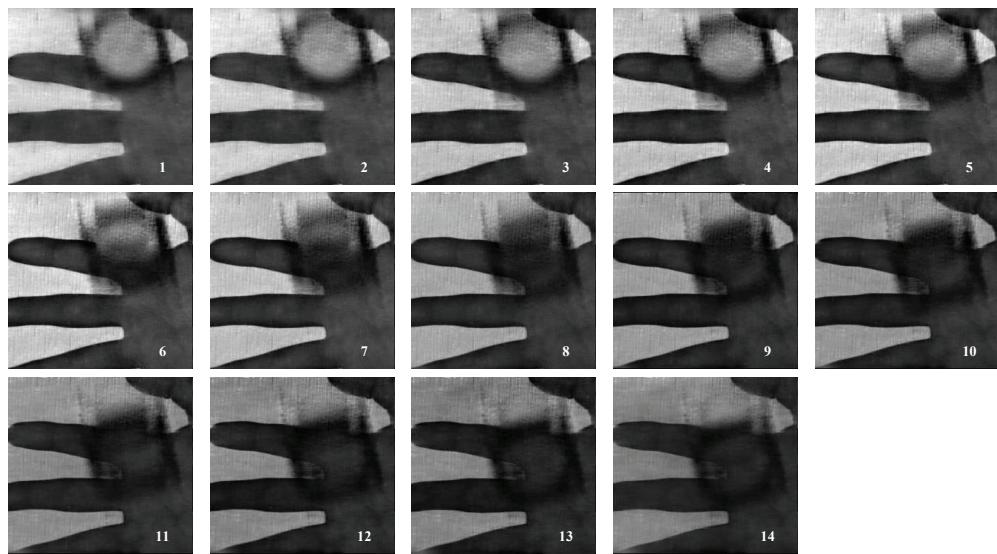


**CACTI 30fps coded image**



**High-speed reconstructed video frames ( $C = 14$ )**

**14 frames reconstructed**



**148 frames reconstructed**

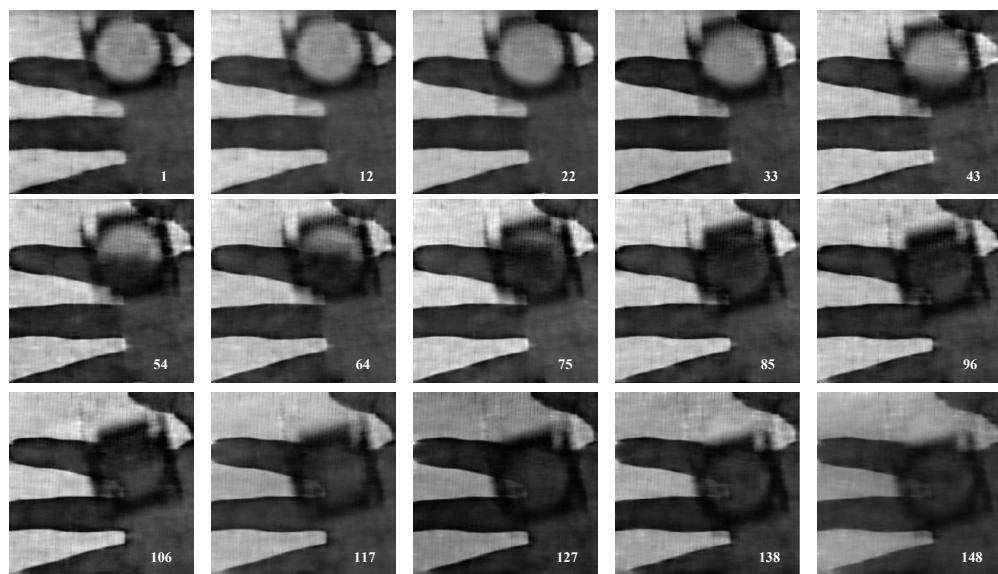
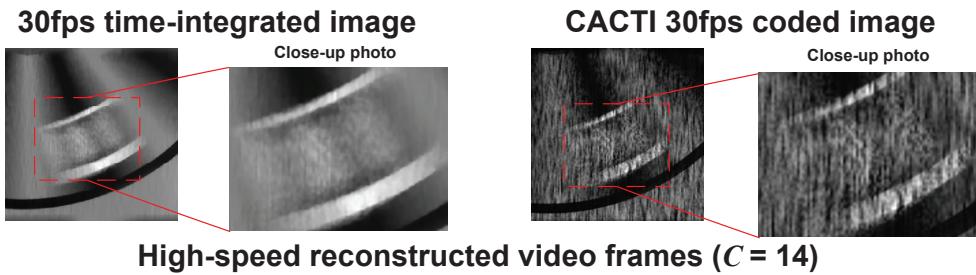
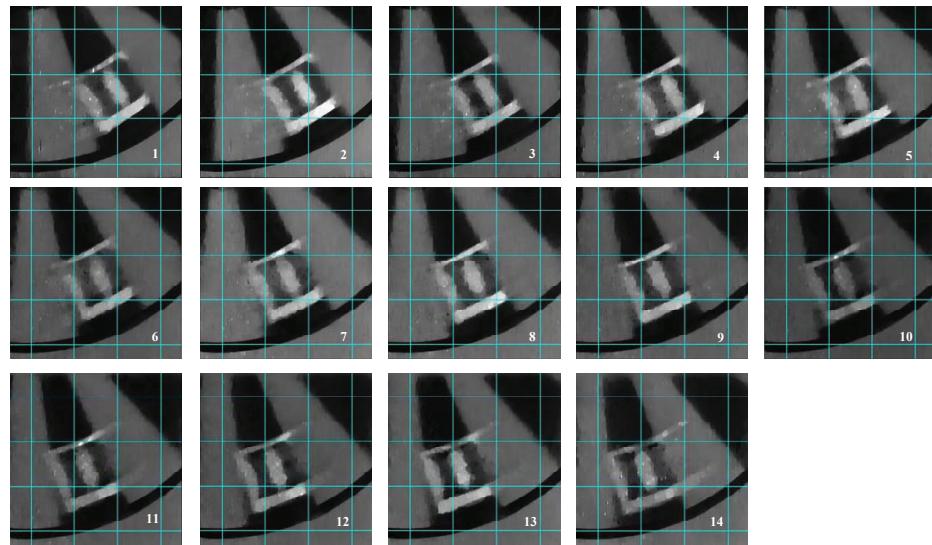


Fig. 11. Capture and reconstruction of a lens falling in front of a hand for  $N_F = 14$  ([Media3.mov](#)) and  $N_F = 148$  ([Media4.mov](#)). Notice the reconstructed frames capture the magnification effects of the lens as it passes in front of the hand.



**14 frames reconstructed (grid added to help visualize movement)**



**148 frames reconstructed (grid added to help visualize movement)**

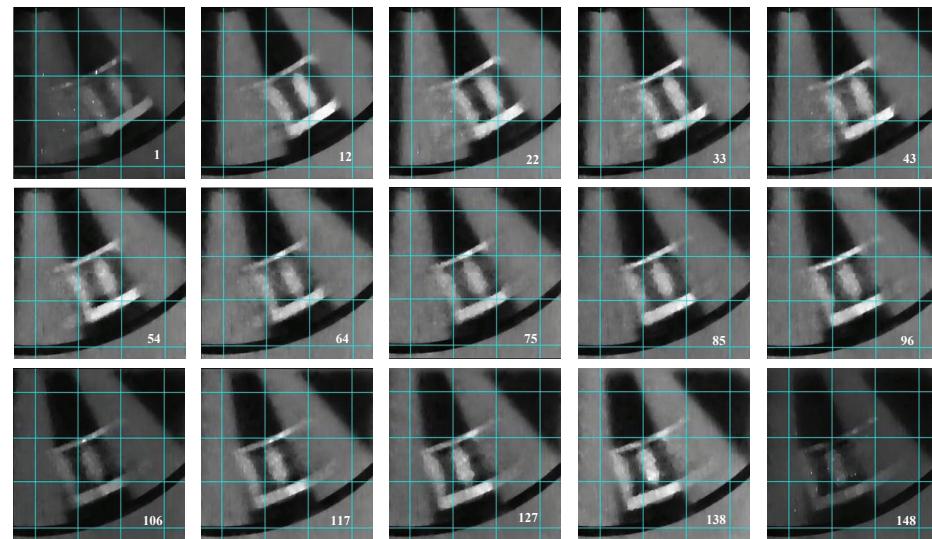
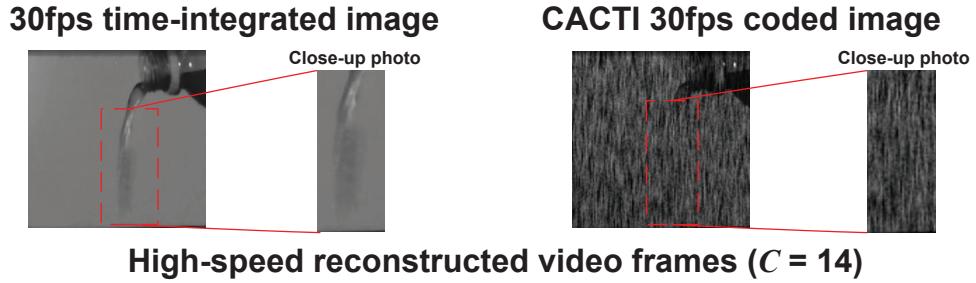
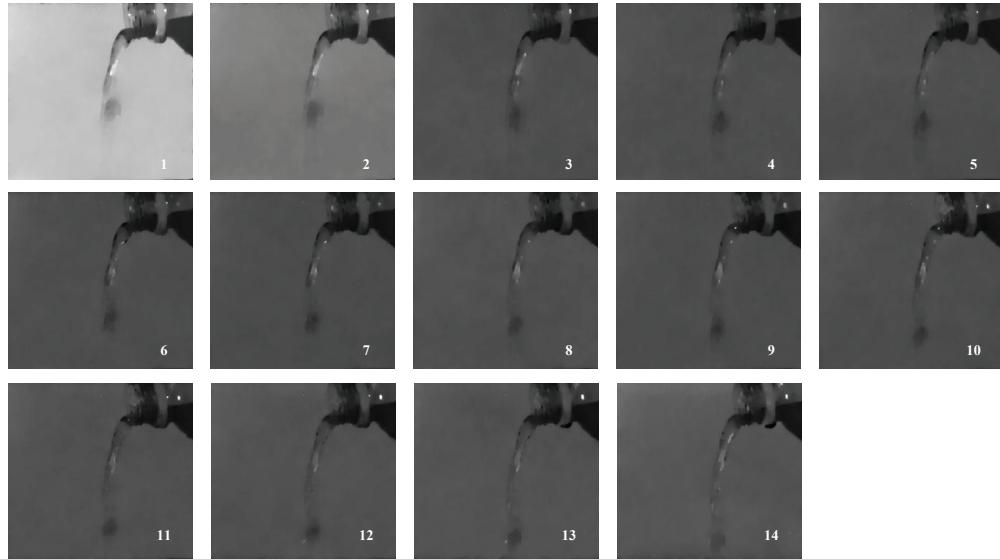


Fig. 12. Capture and reconstruction of a letter 'D' placed at the edge of a chopper wheel rotating at 15Hz for  $N_F = 14$  ([Media5.mov](#)) and  $N_F = 148$  ([Media6.mov](#)). The white part of the letter exhibits ghosting effects in the reconstructions due to ambiguities in the solution. The TwIST algorithm was used to reconstruct this data [22].



**14 frames reconstructed**



**148 frames reconstructed**

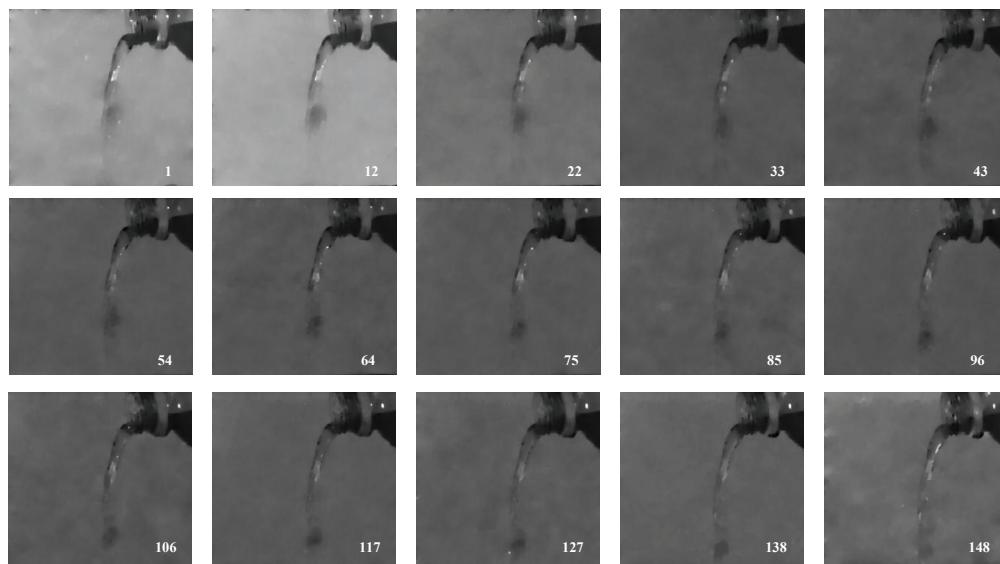


Fig. 13. Capture and reconstructed video of a bottle pouring water for  $N_F = 14$  ([Media7.mov](#)) and  $N_F = 148$  ([Media8.mov](#)). Note the time-varying specularities in the video. The TwIST algorithm was used to reconstruct this data [22].

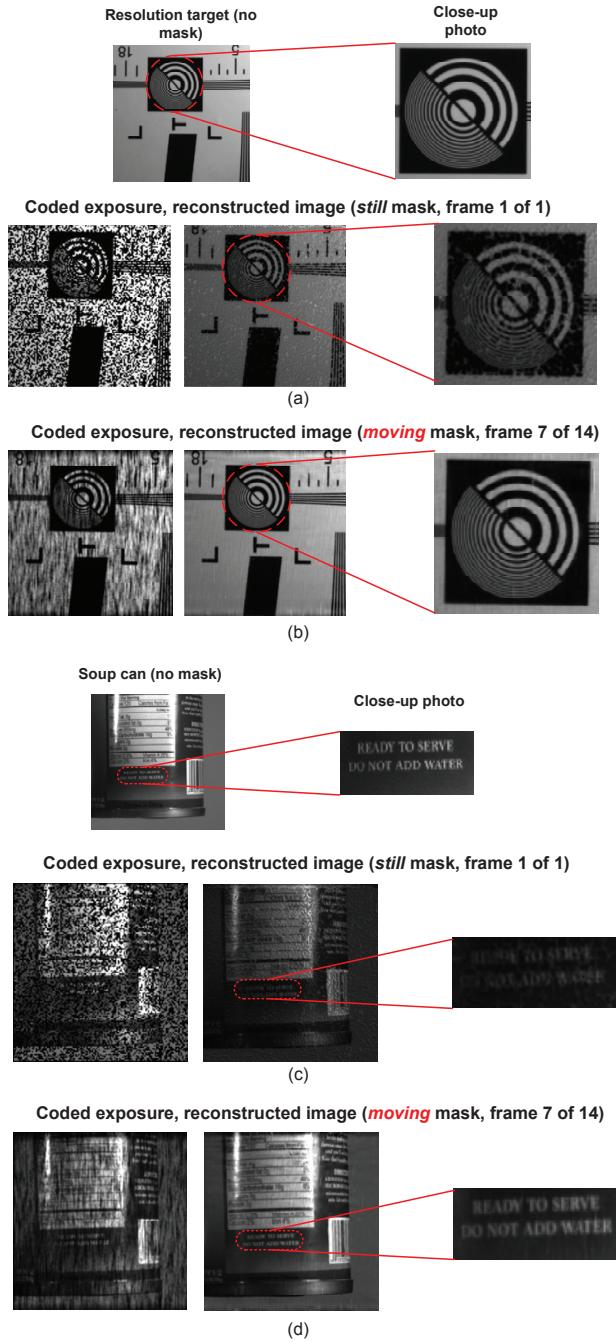


Fig. 14. Spatial resolution tests of (a,b) an ISO 12233 resolution target and (c,d) a soup can. These objects were kept stationary several feet away from the camera. (a,c) show reconstructed results without temporally moving the mask; (b,d) show the same objects when reconstructed with temporal mask motion.

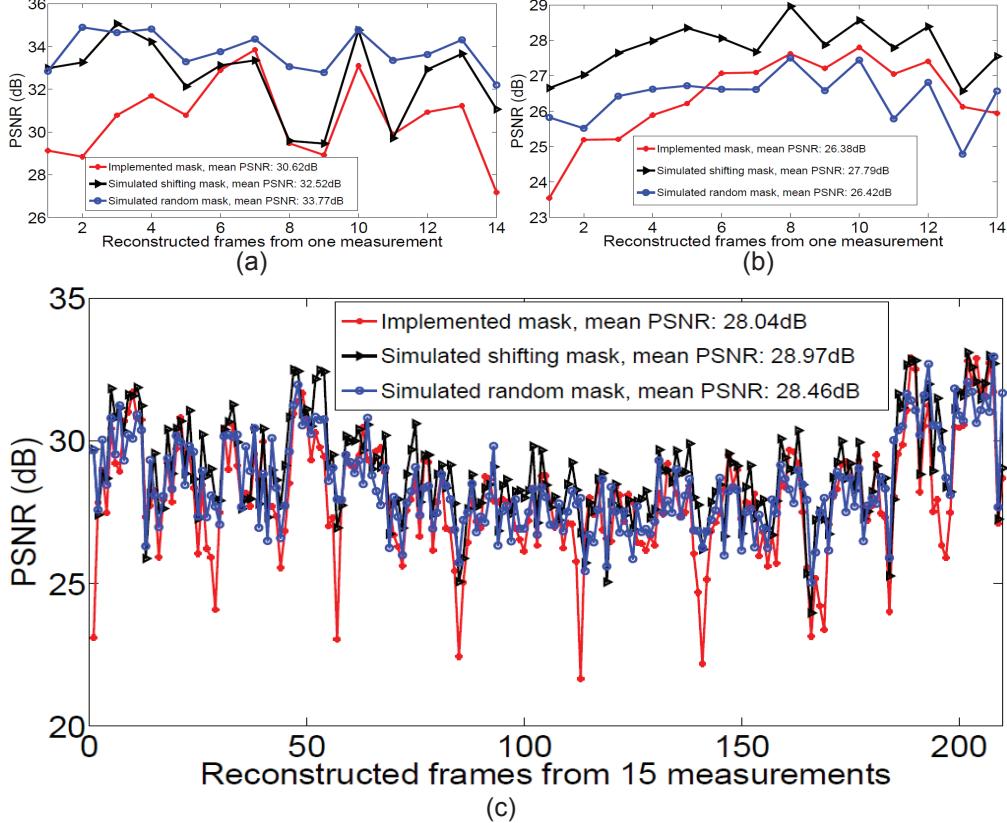


Fig. 15. Simulated and actual reconstruction PSNR by frame. (a), (b), and (c) show PSNR by high-speed, reconstructed video frame for 14 eye blink frames ([Media9.mov](#)), 14 fan frames ([Media10.mov](#)), and 210 fan frames ([Media11.mov](#)), respectively, from snapshots g. Implemented mechanically-translated masks (red curves), simulated translating masks (black curves), and simulated LCoS coding (blue curves) were applied to high-speed ground truth data and reconstructed using GAP.

The CACTI system captures more unique coding projections of scene information when the mask moves  $C$  pixels during the exposure (Fig. 14(b,d)) than if mask is held stationary (Fig. 14(a,c)), thereby improving the reconstruction quality for detailed scenes. The stationary binary coded aperture may completely block small features, rendering the reconstruction difficult and artifact-ridden.

Completely changing the coding patterns  $C$  times during  $\Delta_t$  in hardware is only possible with adequate fill-factor employing a reflective LCoS device to address each pixel  $C$  times during the integration period. To compare the reconstruction fidelity of the low-bandwidth CACTI transmission function (Eq. (7)) with this modulation strategy, we present simulated PSNR values of videos in Fig. 15(a,b,c) (see Media 9-11 for the complete videos). For these simulations, reconstructed experimental frames at  $N_F = 14$  were summed to emulate a time-integrated image. The high-speed reconstructed frames were used as ground truth. We reapply (1) the actual mask pattern; (2) a simulated CACTI mask moving with motion  $s_k$ ; and (3) a simulated, re-randomized coding pattern to each high-speed frame used as ground truth. The reconstruction performance difference between translating the same code and re-randomized the code for each of the  $N_F$

reconstructed frames is typically within 1dB.

## 6. Discussion and Conclusion

CACTI presents a new framework to uniquely code and decompress high-speed video exploiting conventional sensors with limited bandwidth. This approach benefits from mechanical simplicity, large compression ratios, inexpensive scalability, and extensibility to other frameworks.

We have demonstrated GAP, a new reconstruction algorithm that can use one of several bases to compactly represent a sparse signal. This fast-converging algorithm requires no prior knowledge of the target scene and will scale easily to larger image sizes and compression ratios. This algorithm was used for all reconstructions except Figs. 12 and 13.

Despite GAP’s computational efficiency, large-scale CACTI implementations will seek to minimize the data *reconstructed* in addition to that transmitted to sufficiently represent the optical datastream. Future work will adapt the compression ratio  $C$  such that the resulting reconstructed video requires the fewest number of computations to depict the motion of the scene with high quality.

Since coded apertures are passive elements, extending the CACTI framework onto larger values of  $N$  only requires use of a larger mask and a greater detector sensing area, making it a viable choice for large-scale compressive video implementations. As  $N$  increases, LCoS-driven temporal compression strategies must modulate  $N$  pixels  $C$  times per integration. Conversely, translating a passive transmissive element attains  $C$  times temporal resolution without utilizing any additional bandwidth relative to conventional low-framerate capture.

Future large-scale imaging systems may employ CACTI’s inexpensive coding strategy in conjunction with higher-dimensional imaging modalities, including spectral compressive video. Integrating CACTI with the current CASSI system [20] should provide preliminary reconstructions depicting 4-dimensional datasets  $\mathbf{f}(x, y, \lambda, t)$ .