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

Authors: Yasunobu Hitomi, Jinwei Gu, Mohit Gupta, Tomoo Mitsunaga,

Shree Nayar

Published in ICCV 2011

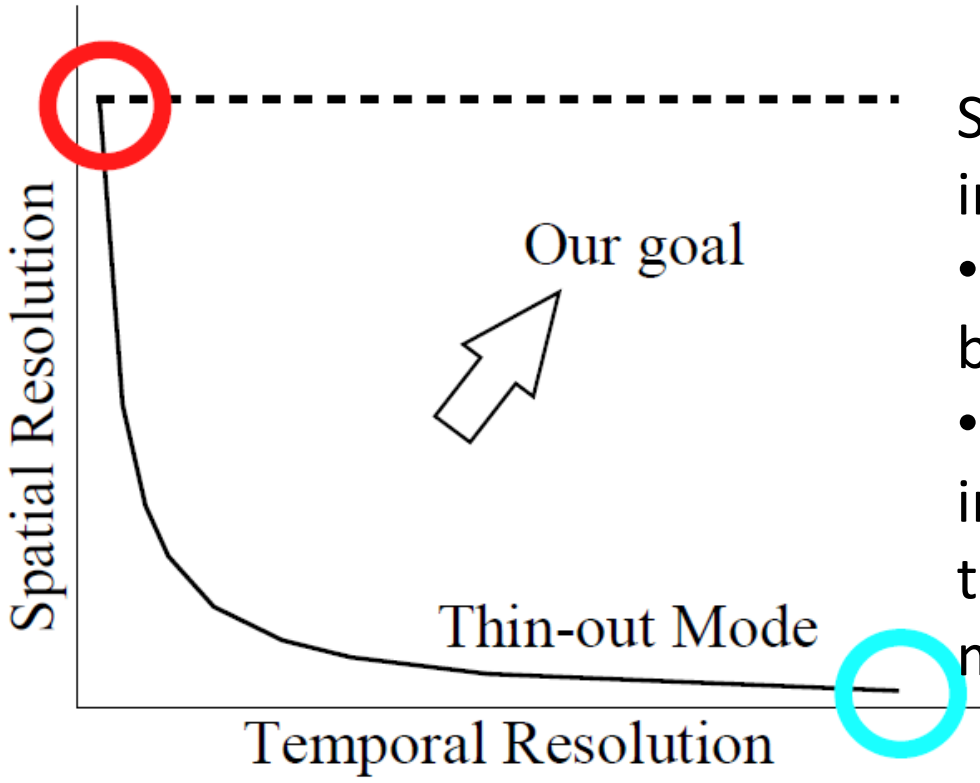
Presented by: Ajit Rajwade
(Feb 24, 2012)

Basic Goal of the Paper

- Application of “computational photography”

- Improve frame rate of a video camera by making appropriate changes to hardware WITHOUT sacrificing spatial resolution.

Space-Time Tradeoff

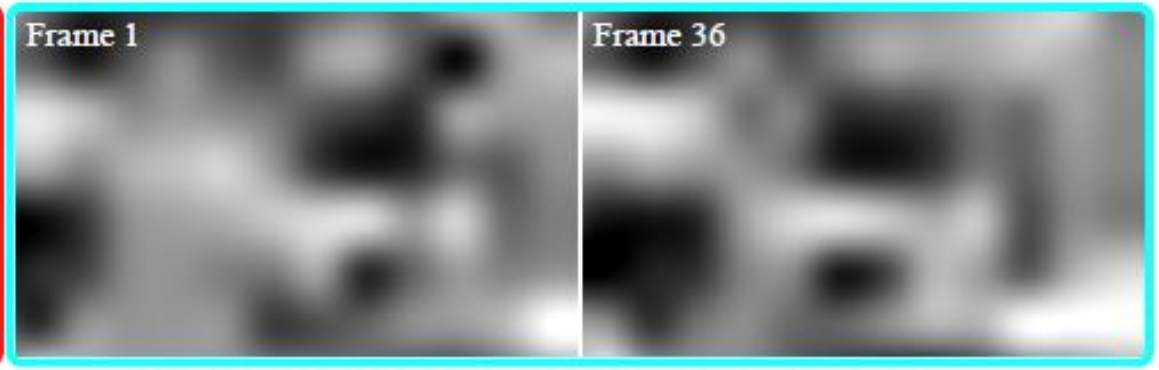


Sampling every k-th row of an image frame:

- Spatial resolution decreases by factor of k,
- Temporal resolution increases by factor of k (for the same number of measurements)

Can be overcome with more sophisticated hardware – but associated cost is HIGH

(b) Motion blurred image



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

Still camera

(d) Our input: A single coded exposure image

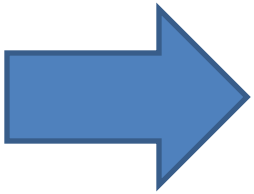
(e) Our result: High spatial resolution, high frame rate video

Coded Exposure Image

It is a coded superposition of ***N*** snapshots (sub-frames) within a unit integration time of the video camera.

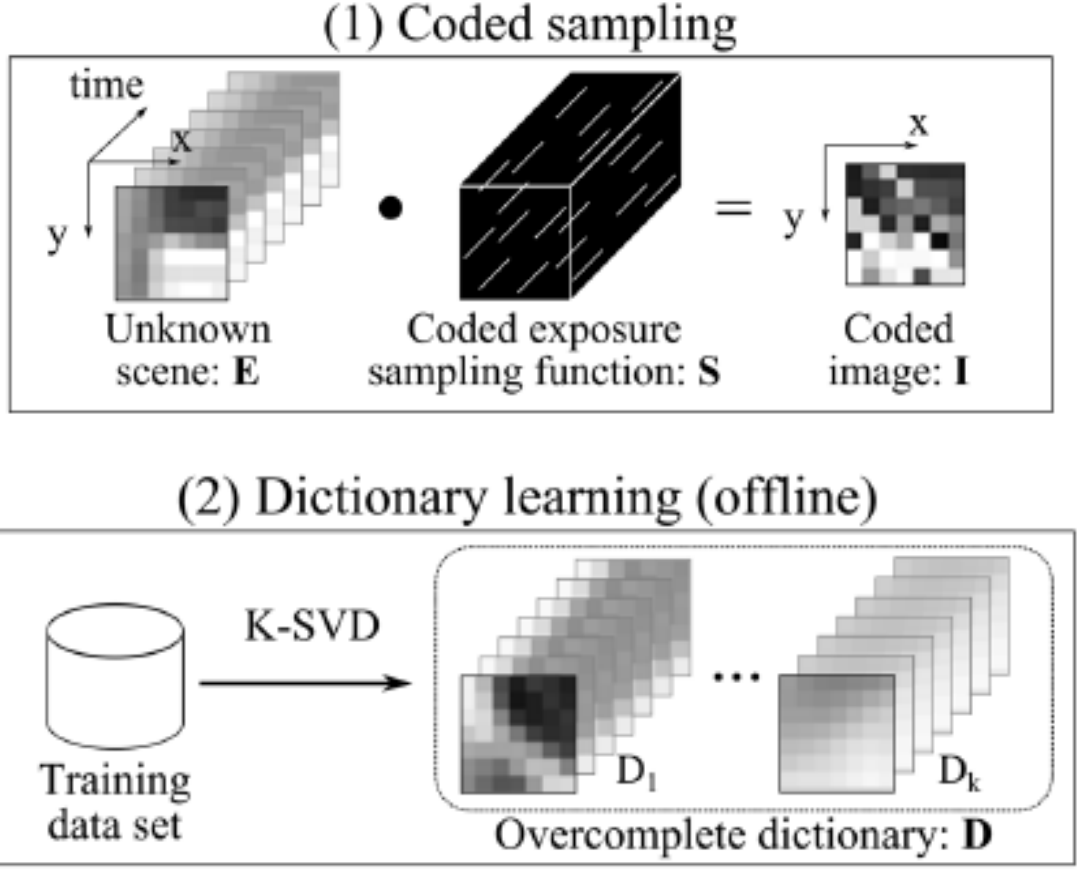
$$I(x,y) = \sum_{t=1}^{N} S(x,y,t) \cdot E(x,y,t).$$

Conventional capture



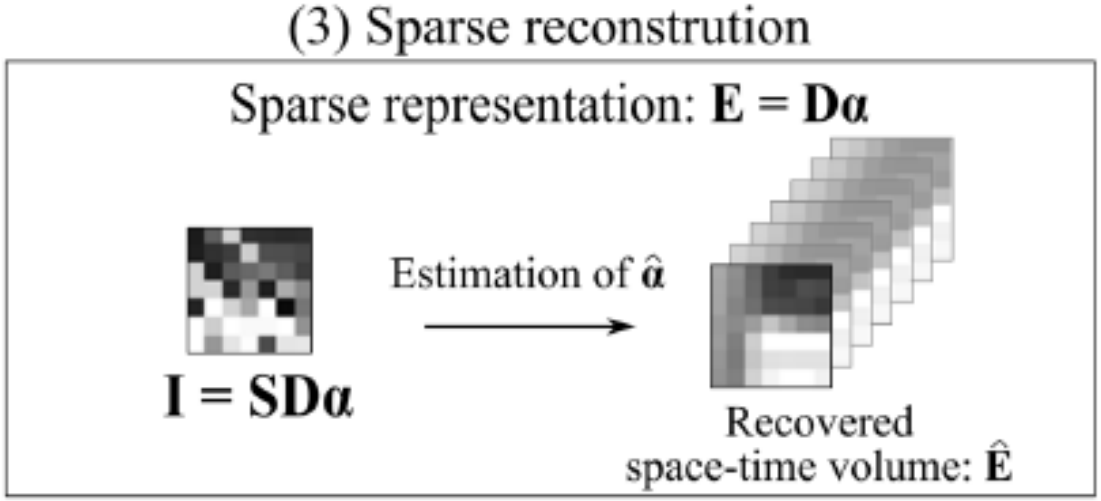
$$S(x,y,t) = 1 \ , \forall (x,y,t)$$

Dictionary Learning/Sparse Coding



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

Dictionary Learning/Sparse Coding



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

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

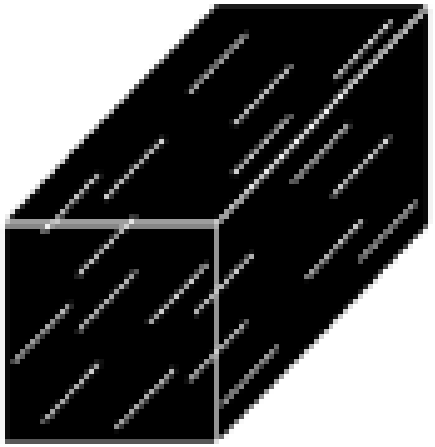
Code Design: S

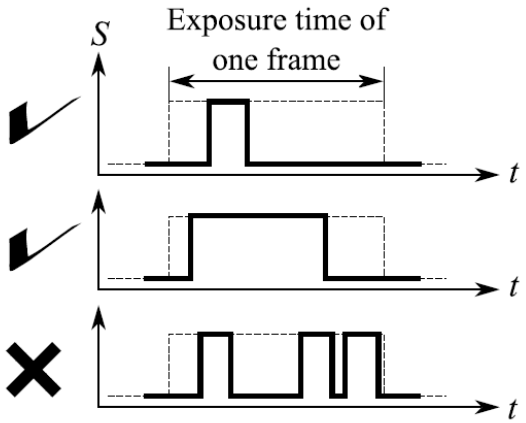
(1) S should be binary – at any point of time, a pixel (that collects light) is **either ON or OFF**

(2) Each pixel can have only one **continuous ON time** (called a **`bump'**) during the camera integration time (due to limitations of contemporary CMOS sensors)

(3) **Fixed bump length** for all pixels – but **different start times** for the bump at different pixels

(4) Union of bumps within an *M* x *M* spatial patch should **cover full integration time**





Optimal Bump Length?

Too high: removes high frequency information

Too low: low SNR

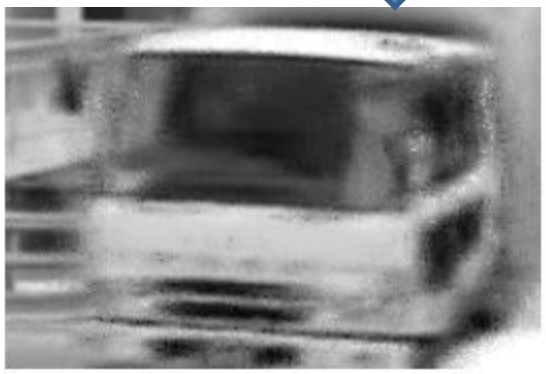
Bump length	Noise standard deviation σ (Grey-levels)					
	0	1	4	8	15	40
1	22.96	22.93	22.88	22.50	21.41	17.92
2	23.23	23.22	23.18	23.06	22.62	20.76
3	**23.37**	**23.37**	**23.35**	23.25	23.03	21.69
4	23.29	23.30	23.25	**23.27**	22.99	22.08
5	23.25	23.26	23.24	23.19	**23.07**	**22.34**
6	23.06	23.10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

Set to 2 for 9X frame rate gain, to 3 for 18X frame rate gain

Dictionary Learning

- Done offline – training set was 20 video sequences, each video rotated in 8 directions and played forward + backward = 320 videos.
- All videos had target frame rate (500 to 1000 fps, as we work with a 60 fps camera and want 9-18 fold gain).
- Video-patch size was 7 x 7 x 36 = 1764 x 1
- Offline learning: KSVD, K = 100,000 atoms
- Sparse coding done online (using OMP)

Results on toy-data

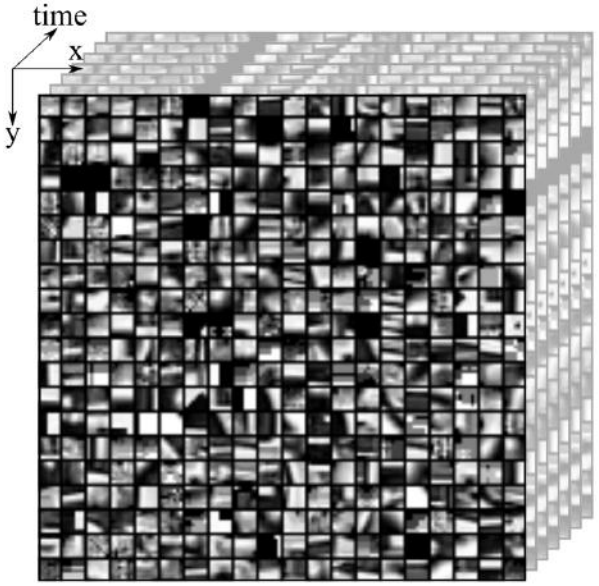


3D DCT (1764 bases)
PSNR = 21.95

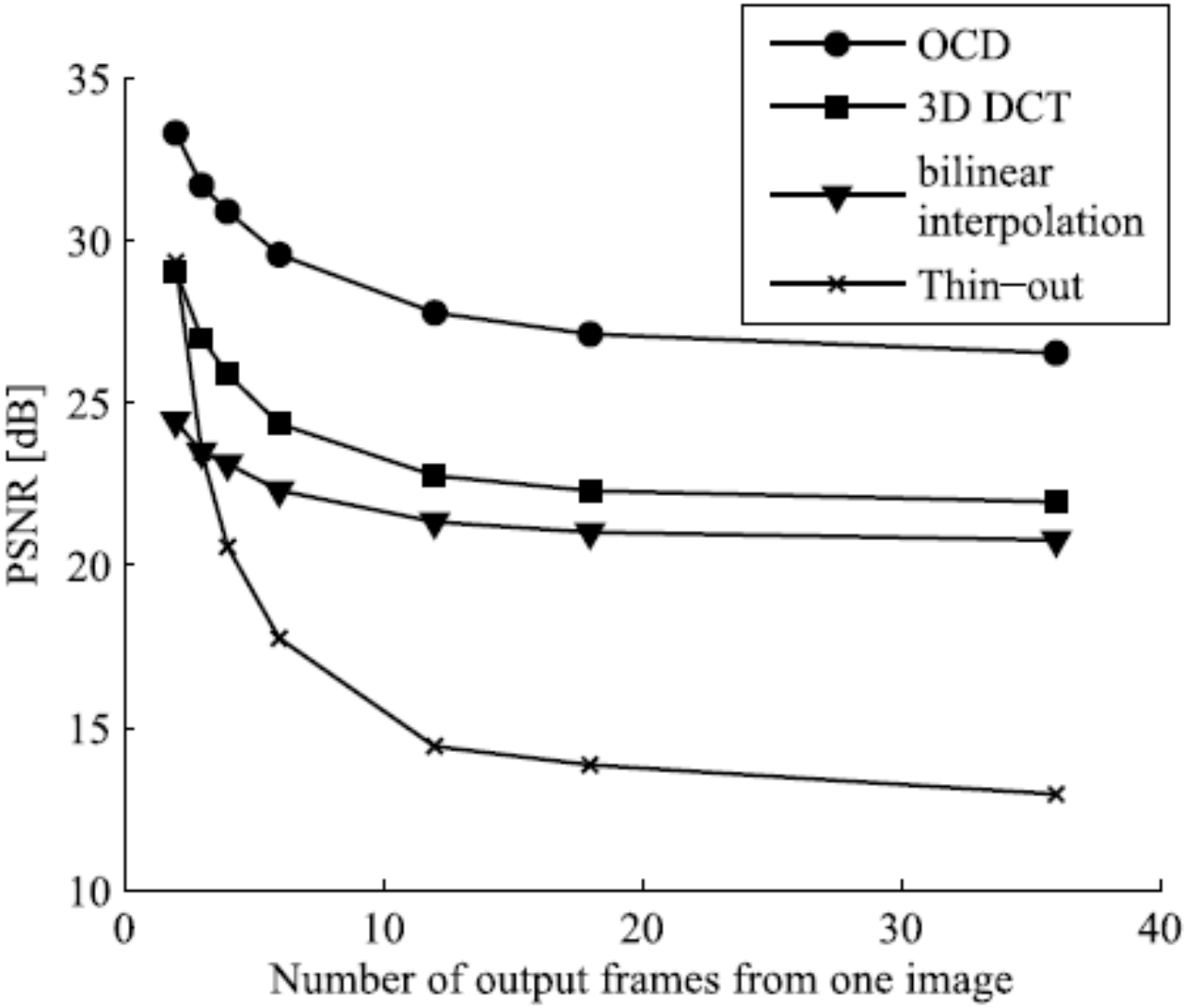
3D DWT (1764 bases)
PSNR = 15.78

Learned Dictionary (1764 bases)
PSNR = 24.21

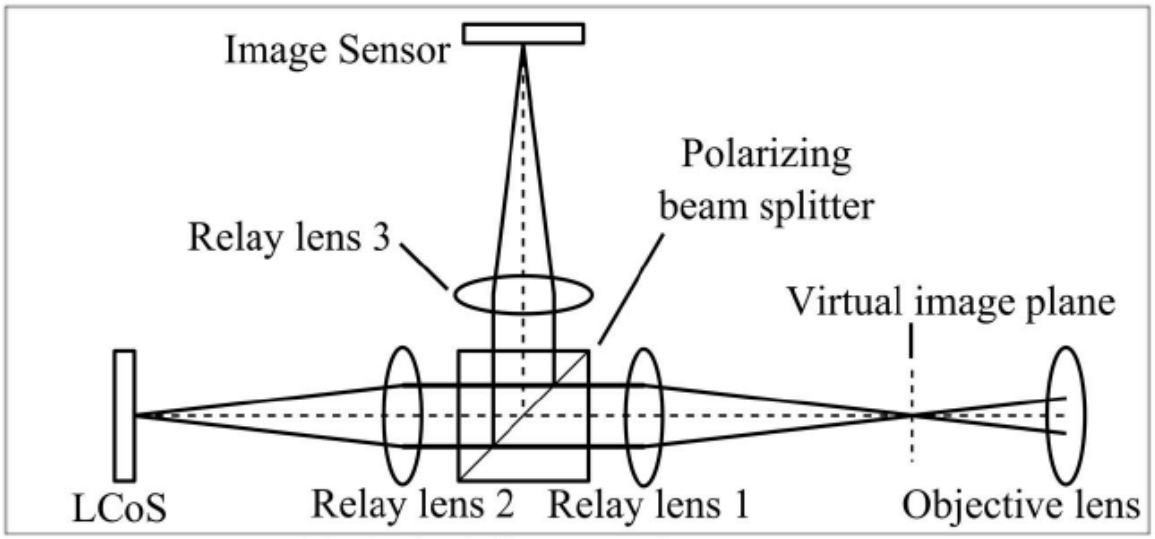
Learned Dictionary (100k bases)
PSNR = 26.54



36X frame rate gain

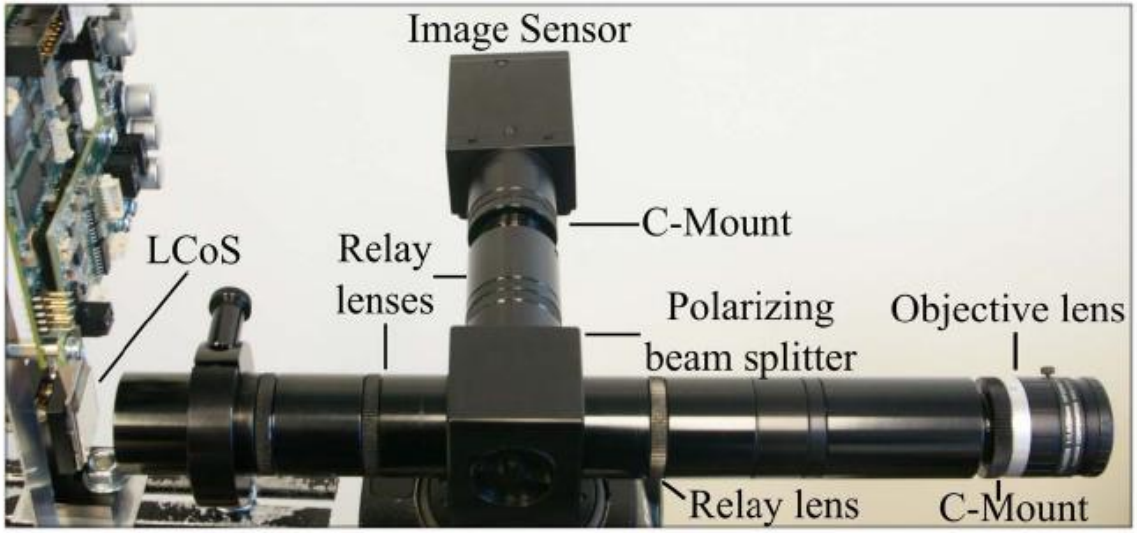


Bilinear interpolation – Uses simple grid-based down-sampling of space-time volume.



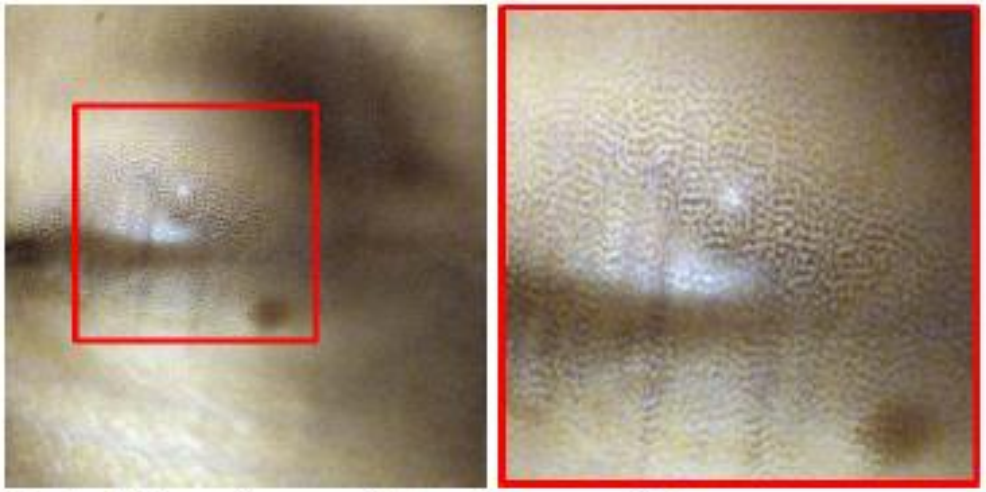
(a) Optical diagram of our setup

Per-pixel binary codes were implemented using a liquid crystal on silicon device (LCoS)

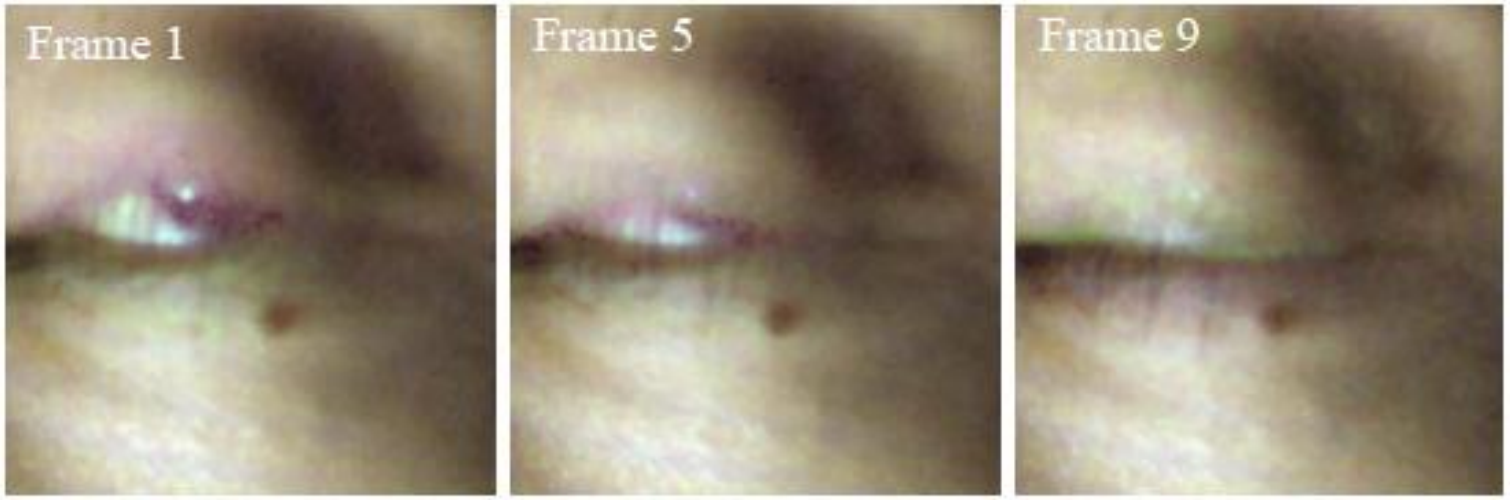


(b) Image of our setup

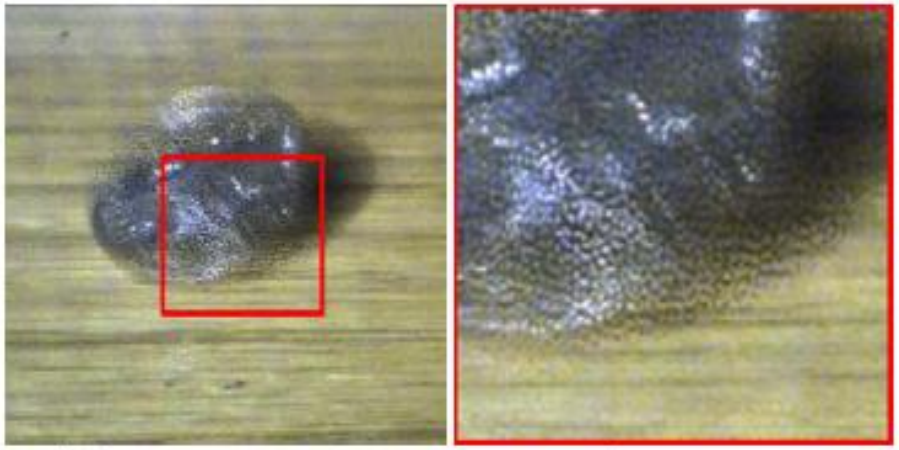
LCoS is synced with the camera and operates at 9-18 times camera frame rate



Coded Exp. Image (27ms) Close-up

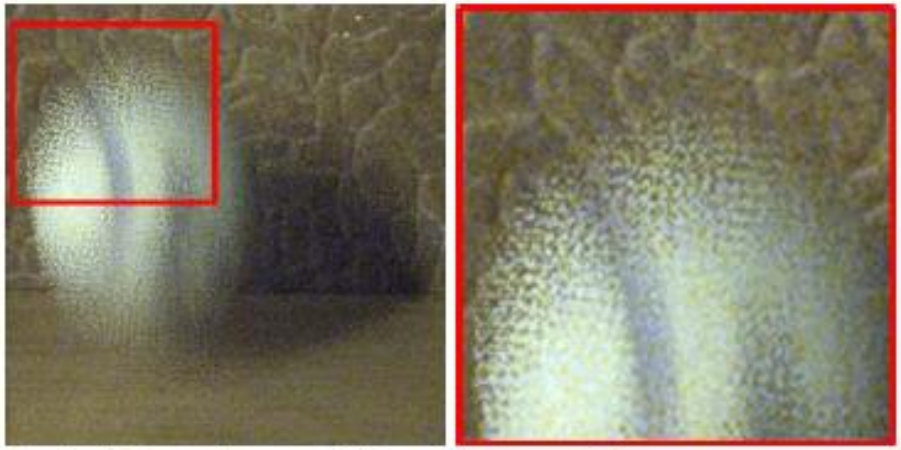


Reconstructed Frames (3 out of 9)

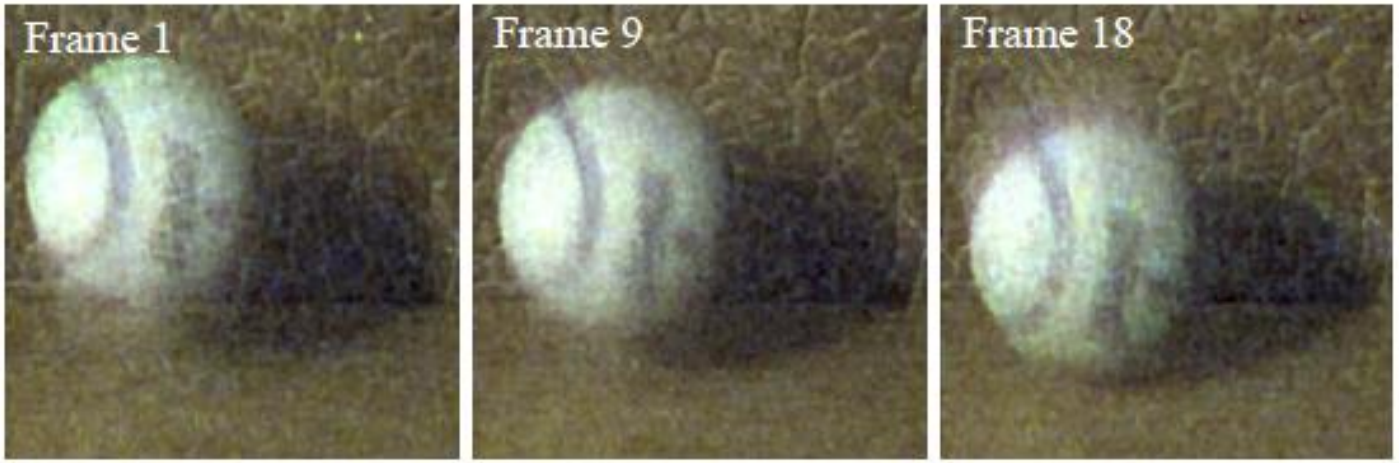


Coded Exp. Image (27ms) Close-up

Reconstructed Frames (3 out of 9)



Coded Exp. Image (18ms) Close-up



Reconstructed Frames (3 out of 18)

Prior Work

- Related hardware prototype in *"P2C2: Programmable Pixel Compressive Camera for High Speed Imaging"*, by Reddy *et al*, CVPR 2011.
- One major difference – reconstruction technique: sparsity on the transform coefficients of each *sub-frame* +

  brightness constancy assumption / optical flow for temporal redundancy

Conclusion

- Overcomes space-time tradeoff using per-pixel coded exposure pattern
- Hardware prototype developed
- Works well for varied complex motions (does not require analytical motion model)
- Limitation 1: Maximum target frame-rate must be fixed (e.g. 36X)
- Limitation 2: Requires training videos (which are hopefully `representative') at target frame rate.