Compressive Imaging Systems

PART 1: RICE SINGLE PIXEL CAMERA

Standard Camera

- Consists of an aperture, a lens and a detector array.
- Light from the scene enters camera through aperture and is focussed onto detector array by the lens.
- Number of pixels on detector array = size of the image.

Rice Single Pixel Camera: a CS-based camera

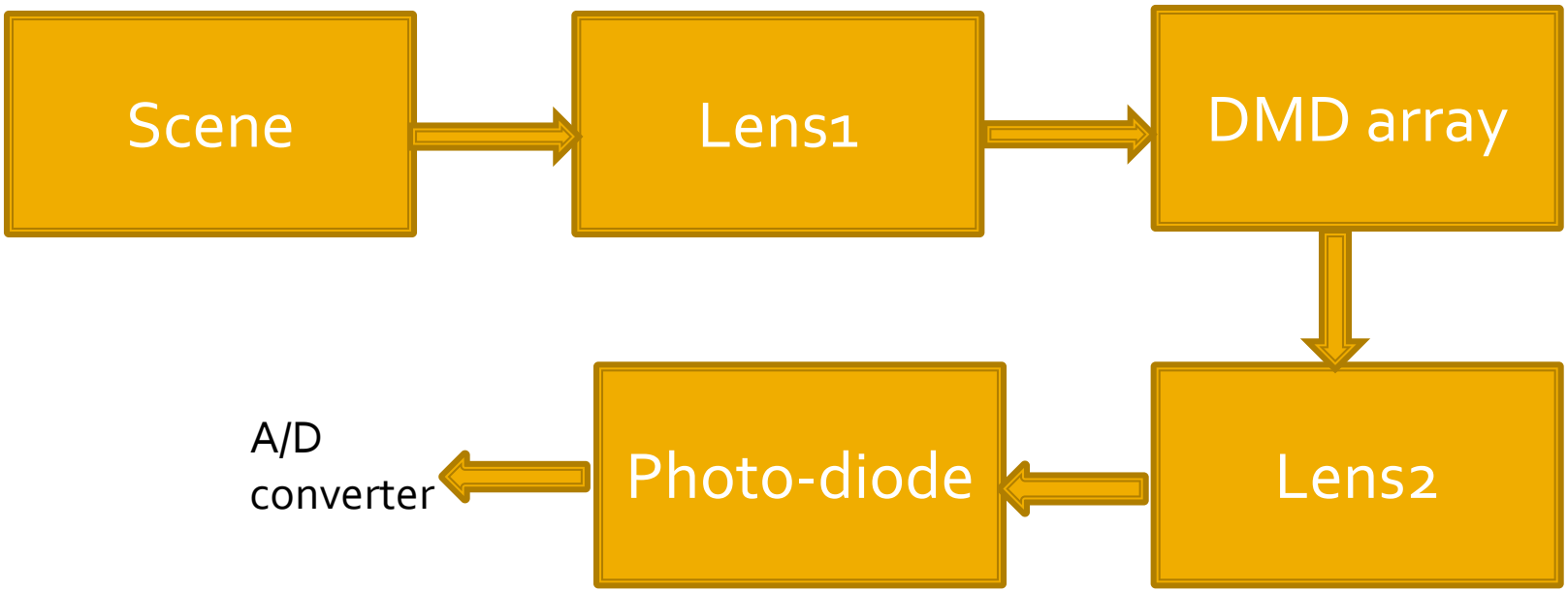
- It does not measure the image 2D array.
- Instead it directly measures the dot-product of the image with a set of random codes. This is mathematically shown below:

$$\forall i, 1 \le i \le m, y_i = \mathbf{f}^T \boldsymbol{\varphi}_\mathbf{i}$$

- We now want to reconstruct **f** given the set of measurements, i.e. $\{y_i\}$ and random codes.

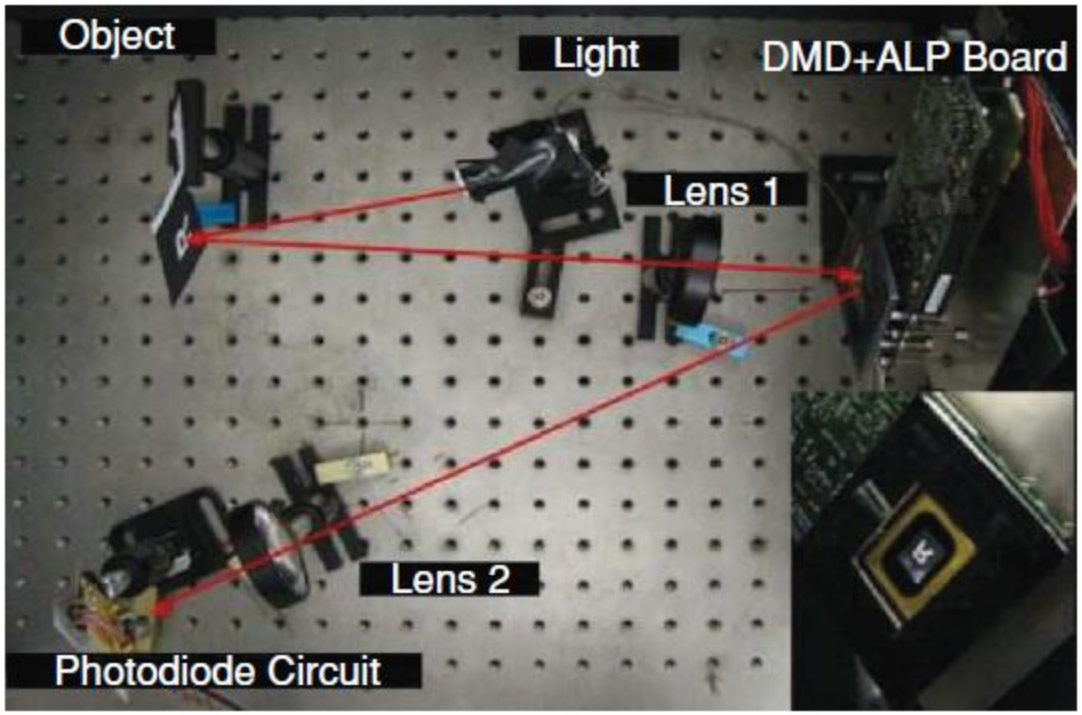
Ref: Duarte et al, "Single pixel imaging via compressive sampling", IEEE Signal Processing Magazine, March 2008.

Rice Single Pixel Camera: a CS-based camera



Ref: Duarte et al, "Single pixel imaging via compressive sampling", IEEE Signal Processing Magazine, March 2008.

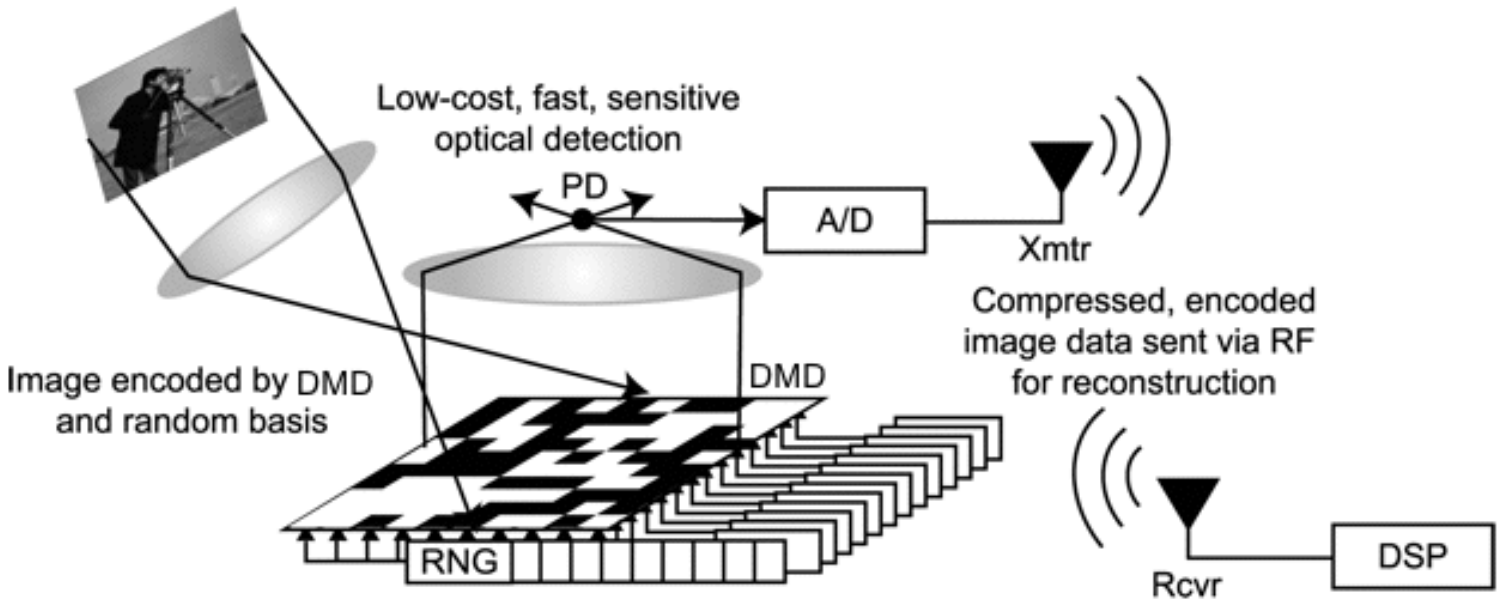
Rice Single Pixel Camera: a CS-based camera



[FIG1] Aerial view of the single-pixel CS camera in the lab [5].

Ref: Duarte et al, "Single pixel imaging via compressive sampling", IEEE Signal Processing Magazine, March 2008.

Rice Single Pixel Camera: a CS-based camera



Ref: Duarte et al, "Single pixel imaging via compressive sampling", IEEE Signal Processing Magazine, March 2008.

Rice Single Pixel Camera: a CS-based camera

- Contains no detector array.
- Light from the scene passes through Lens1 and is focussed on a digital micromirror device (DMD).
- DMD is a 2D array of thousands of very tiny mirrors.
- Light reflected from DMD passes through the second lens and to the photodiode.

Rice Single Pixel Camera: a CS-based camera

- The DMD acts as a random binary array of the same size as image **f**. Each mirror in the DMD corresponds to one pixel in **f**.
- A mirror can be either ON=1 (facing the Lens2) or OFF=0 (facing away from Lens2).
- The photodiode circuit acts as a photon counter – effectively, it measures the dot product between **f** and ϕ_i, i.e. it measures:

$$y_i = \sum_{j=1}^{n} f_j \phi_{ij}$$

Rice Single Pixel Camera: a CS-based camera

- These values $\{y_i\}$ are output in the form of a voltage which is then digitized by an A/D converter.
- Note that a different binary code vector ϕ_i is used for each y_i, $1 <= i <= m$.
- The random binary code is implemented by setting the orientation of the mirrors (facing toward or away from Lens2) randomly within the hardware.
- But these are codes with 0 and 1 and such a matrix does not obey RIP.
- So instead a matrix with -1 and +1 is "generated" using two measurements:

$$y_1 = \Phi_1 x$$
$$y_2 = \Phi_2 x$$
$$y_1 - y_2 = (\Phi_1 - \Phi_2) x$$

Φ_2 contains a 1 wherever Φ_1 contains a 0, and Φ_2 contains a 1 wherever Φ_1 contains a 0.

Rice Single Pixel Camera: a CS-based camera

- The basic measurement model can be written as follows (in vector notation):

$$\mathbf{y} = \mathbf{\Phi f}, \mathbf{\Phi} = [\boldsymbol{\varphi}_1 \,|\, \boldsymbol{\varphi}_2 \,|\, \ldots \,|\, \boldsymbol{\varphi}_m]^T,$$

$$\mathbf{y} = (y_1, y_2, \ldots, y_m)$$

- As per CS theory, there are guarantees of good reconstruction if the number of samples obeys (for K-sparse signals):

$$m \geq O(K \log(n/K))$$

Reconstruction Results

Refer to reconstruction results in the following article:

Duarte et al, "Single pixel imaging via compressive sampling", IEEE Signal Processing Magazine, March 2008

Optimization technique used:

$$\min_f TV(f) \text{ such that } y = \Phi f$$

$$TV(f) = \sum_x \sum_y \sqrt{f_x^2(x,y) + f_y^2(x,y)}$$

More Results:
http://dsp.rice.edu/cscamera

Original

4096 pixels, 800 measurements, i.e. 20% data

Informal description of Rice Single Pixel Camera:
http://terrytao.wordpress.com/2007/04/13/compressed-sensing-and-single-pixel-cameras/

Compressed sensing on the chip

- This is a compressive camera developed at Stanford, that uses the same mathematical model as the Rice SPC.
- The difference is that each video frame is divided into **non-overlapping blocks** of size (say) 16 x 16, and the dot products are computed separately for each block.
- The $m << n$ dot products are computed on a CMOS chip using m different binary random codes.
- For a single random code, the dot products are computed simultaneously for all the blocks.
- Per block, only the $m << n$ values are quantized (Analog to digital conversion), saving huge amounts of energy and time.
- Mounted on a mobile phone – led to 15 fold savings in battery power during acquisition.
- Reconstruction is performed offline.
- See here for more information.
- Yields excellent quality reconstruction with high frame rates (960 fps).
- Reason for being able to increase frame rate is that fewer measurements are made within each exposure time ($m << n$) than a conventional camera.

Image source: Oike and El-Gamal, "CMOS sensor with programmable compressed sensing", IEEE Journal of Solid State Electronics, January 2013

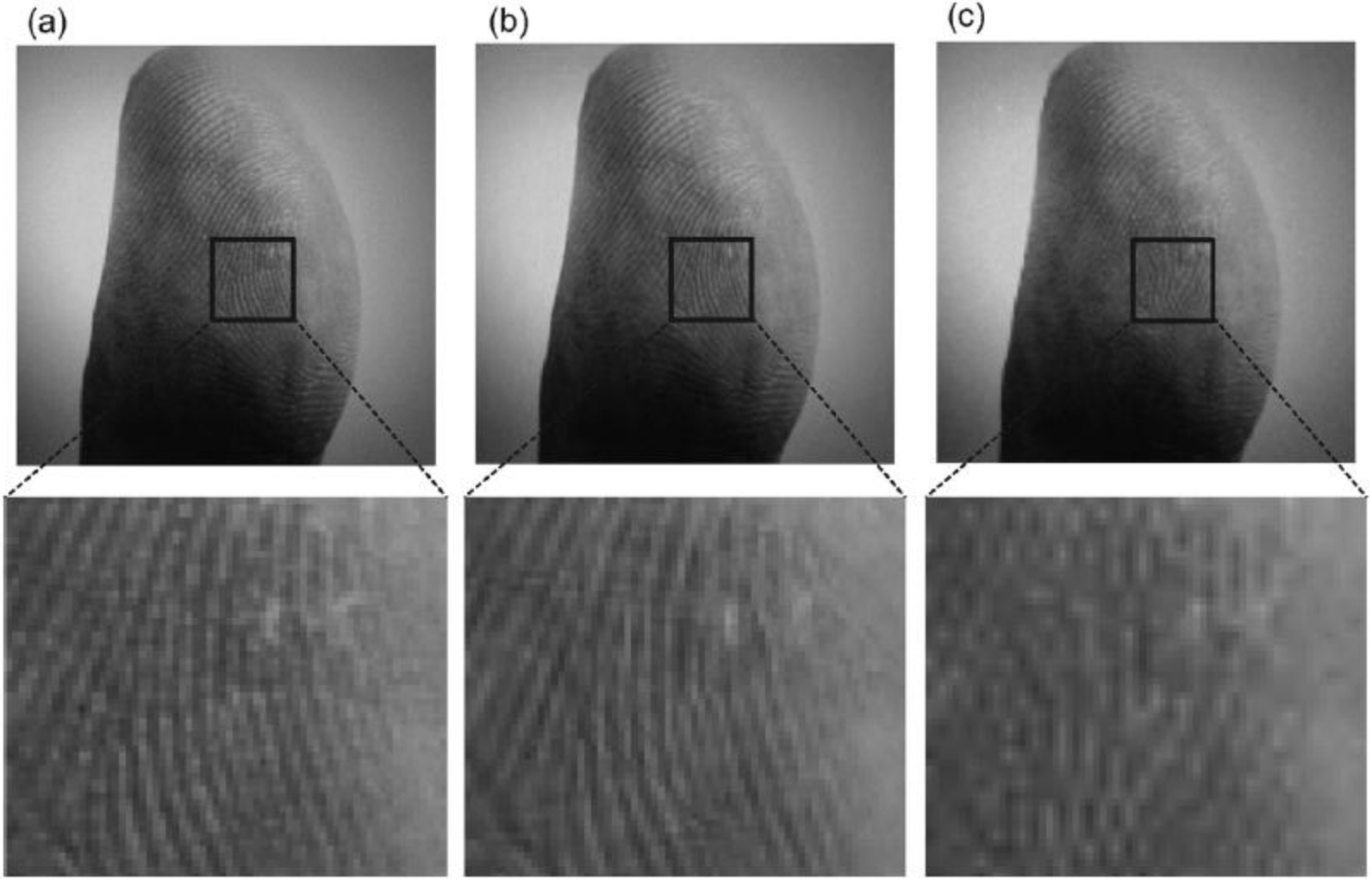


Fig. 16. Sample images captured in: (a) normal mode at 120 fps, (b) compressed sensing at CR = 1/4 and 480 fps, (c) downsampling at 1/4 ratio.

http://isl.stanford.edu/~abbas/papers/PDF1.pdf

Image source: Oike and El-Gamal, "CMOS sensor with programmable compressed sensing", IEEE Journal of Solid State Electronics, January 2013

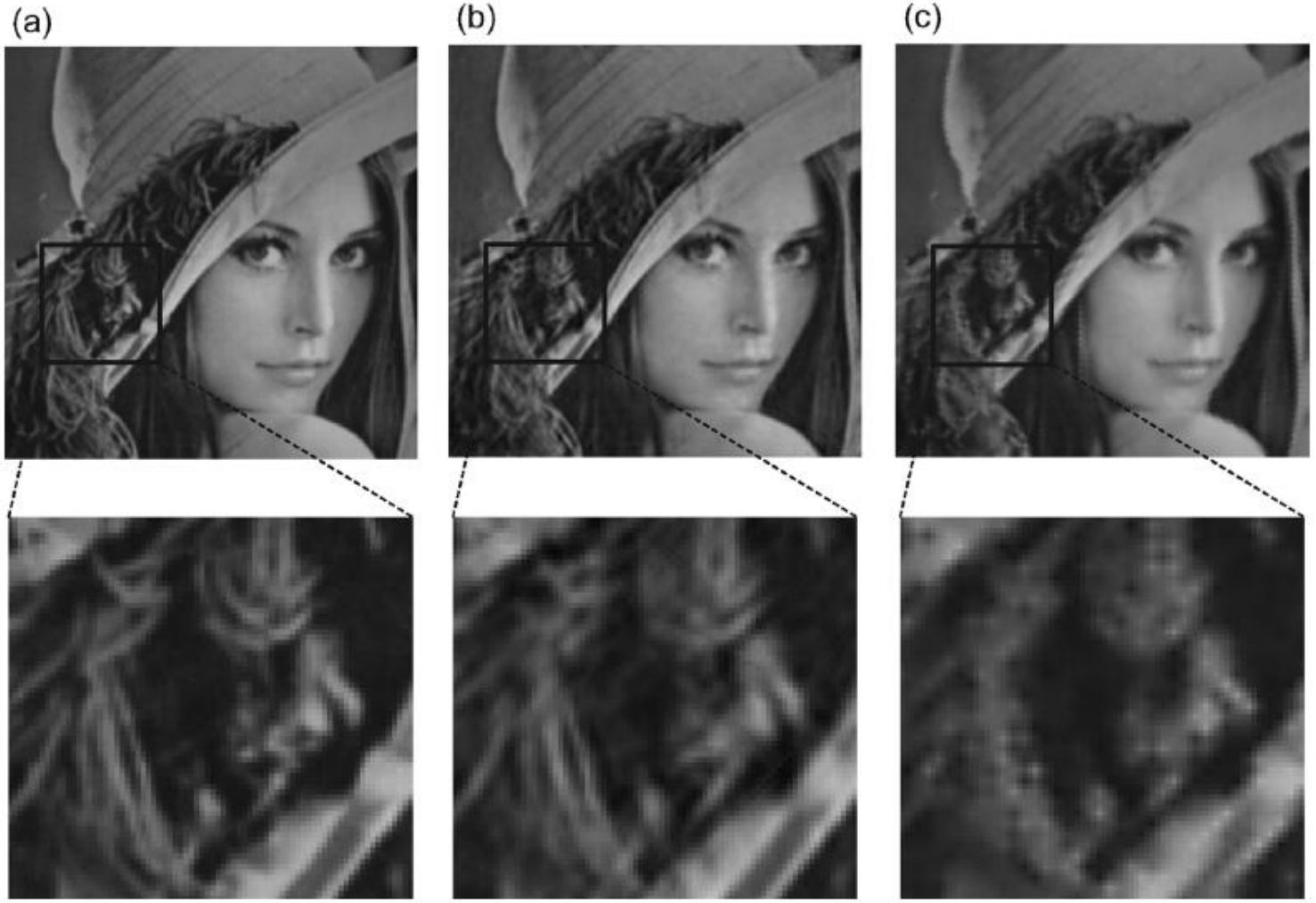


Fig. 17. Sample images captured in: (a) normal mode at 120 fps, (b) compressed sensing at CR = 1/8 and 960 fps, (c) downsampling at 1/8 ratio.

http://isl.stanford.edu/~abbas/papers/PDF1.pdf

PART 2: RICE SINGLE PIXEL CAMERA FOR VIDEO ACQUISITION

Streaming Video Acquisition

- SPC can be extended for video.
- Consider a video with a total of *F* (2D) images, each with *n* pixels.
- In the still-image SPC, an image was coded several times using different binary codes ϕ_i where i ranges from 1 to M.
- Note that in a video-camera, this reduces the **video frame rate**.
- Assume we take a total of *M* measurements, i.e. *M/F* measurements (**dot products**) per frame.
- We make the simplifying assumption that the **scene changes slowly or not at all** within the set of *M/F* dot products.

Streaming Video Reconstruction

- **Method 1:** To reconstruct the original video from the CS measurements, we could use a 2D DCT/wavelet basis Ψ and perform *F* independent (2D) frame-by-frame reconstructions, by solving:

$$\forall t \in \{1,\ldots,F\}, \min_{\boldsymbol{\theta}_t} |\boldsymbol{\theta}_t|_1 \text{ such that } \mathbf{y}_t = \boldsymbol{\Phi}_t \mathbf{f}_t = \boldsymbol{\Phi}_t \boldsymbol{\Psi} \boldsymbol{\theta}_t,$$

$$\boldsymbol{\Phi}_t \in R^{M/F \times n}, \boldsymbol{\Psi} \in R^{n \times n}, \boldsymbol{\theta}_t \in R^n, \mathbf{y}_t \in R^{M/F}$$

- This procedure fails to exploit the tremendous **inter-frame redundancy** in natural videos.

Streaming Video Acquisition

- **Method 2:** Create a joint measurement matrix Φ for the entire video sequence, as shown below. Φ is block-diagonal, with each of the diagonal blocks being the matrix Φ_t for measurement $\mathbf{y}_t$ at time *t*.

$$\Phi = \begin{vmatrix} \Phi_1 & 0 & & 0 \\ 0 & \Phi_2 & & 0 \\ & & & \\ 0 & & & \Phi_F \end{vmatrix}, \Phi \in \mathcal{R}^{M \times Fn}, \Phi_i \in \mathcal{R}^{M/F \times n}$$

$$\mathbf{y} = (\mathbf{y}_1; \mathbf{y}_2; \ldots; \mathbf{y}_F), \mathbf{y}_i = \Phi_i \mathbf{f}_i; \mathbf{y} \in \mathcal{R}^M$$

Streaming Video Acquisition

- **Method 2 (continued) :** Use a 3D DCT/wavelet basis Ψ (size *Fn* by *Fn*) for sparse representation of the video sequence:

$$\min_{\theta} \left| \theta \right|_1 \text{ such that } \mathbf{y} = \Phi\mathbf{f} = \Phi\Psi\theta,$$

$$\Phi \in R^{M \times Fn}, \Psi \in R^{Fn \times Fn}, \theta \in R^{Fn}, \mathbf{y} \in R^{M}$$

- Videos frames change slowly in time. The 3D-DCT/wavelet encourages smoothness in the time dimension.

Streaming Video Acquisition

- **Method 3 (Hypothetical):** Assume we had a 3D SPC with a full 3D sensing matrix $\mathbf{\Phi}$ which operates on the full video, and with an associated 3D wavelet/DCT basis.

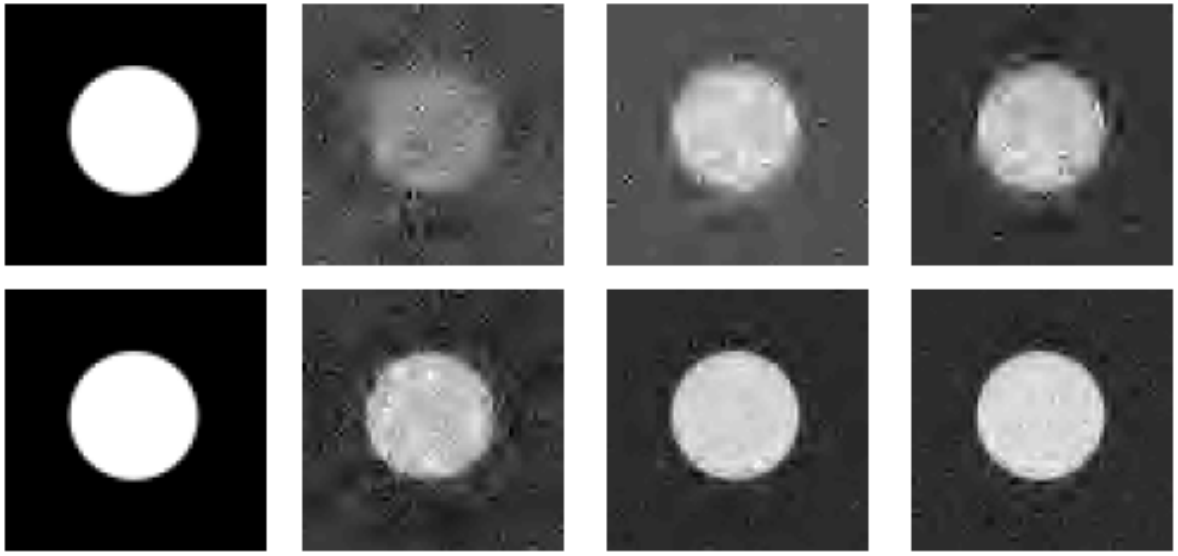
$$\min_{\boldsymbol{\theta}} \left| \boldsymbol{\theta} \right|_1 \text{ such that } \mathbf{y} = \mathbf{\Phi f} = \mathbf{\Phi \Psi \theta},$$

$$\mathbf{\Phi} \in R^{M \times Fn}, \mathbf{\Psi} \in R^{Fn \times Fn}, \boldsymbol{\theta} \in R^{Fn}, \mathbf{y} \in R^{M}$$

- Unlike method 2, Φ is not block-diagonal.
- Also, such a scheme is not realizable in practice – as dot products cannot be computed for an entire video.
- This method is purely for reference comparison.

Results

- Experiment performed on a video of a moving disk (against a constant background) - size 64 x 64 with $F = 64$ frames.
- This video is sensed with a total of M measurements with M/F measurements per frame.
- All three methods (frame-by-frame 2D, 2D measurements with 3D reconstruction, 3D measurements with 3D reconstruction) compared for $M = 20000$ and $M = 50000$.



(a) frame 32 (b) 2D meas 2D recon (c) 2D meas 3D recon (d) 3D meas 3D recon

Method 1 Method 2 Method 3

Fig. 3. *Frame 32 from reconstructed video sequence using (top row) $M = 20{,}000$ and (bottom row) $M = 50{,}000$ measurements. (a) Original frame. (b) Frame-by-frame 2D measurements; frame-by-frame 2D reconstruction; MSE = 3.63 and 0.82. (c) Frame-by-frame 2D measurements; joint 3D reconstruction; MSE = 0.99 and 0.24. (d) Joint 3D measurements; joint 3D reconstruction; MSE = 0.76 and 0.18. The results in (d) are comparable to the MSE obtained by wavelet thresholding with $K = 655$ and 4000 coefficients, respectively.*

Source of images:
Duarte et al,
"Compressive imaging for video representation and coding",
http://www.ecs.umass.edu/~mduarte/images/CSCamera_PCS.pdf

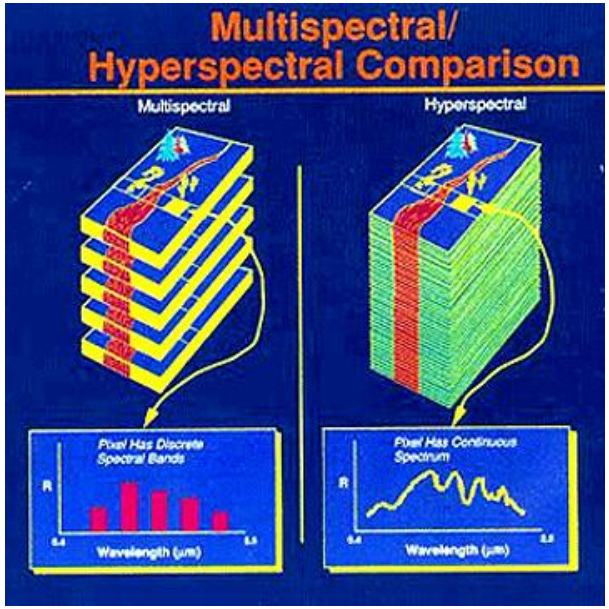
PART 3: COMPRESSIVE ACQUISITION OF HYPERSPECTRAL DATA

Beyond color: Hyperspectral images

- Hyperspectral images are images of the form $M \times N \times L$, where L is the number of channels. L can range from 30 to 30,000 or more.
- The visible spectrum ranges from ~420 nm to ~750 nm.
- Finer division of wavelengths than possible in RGB!
- Can contain wavelengths in the infrared or ultraviolet regime.

Sources of confusion ☺

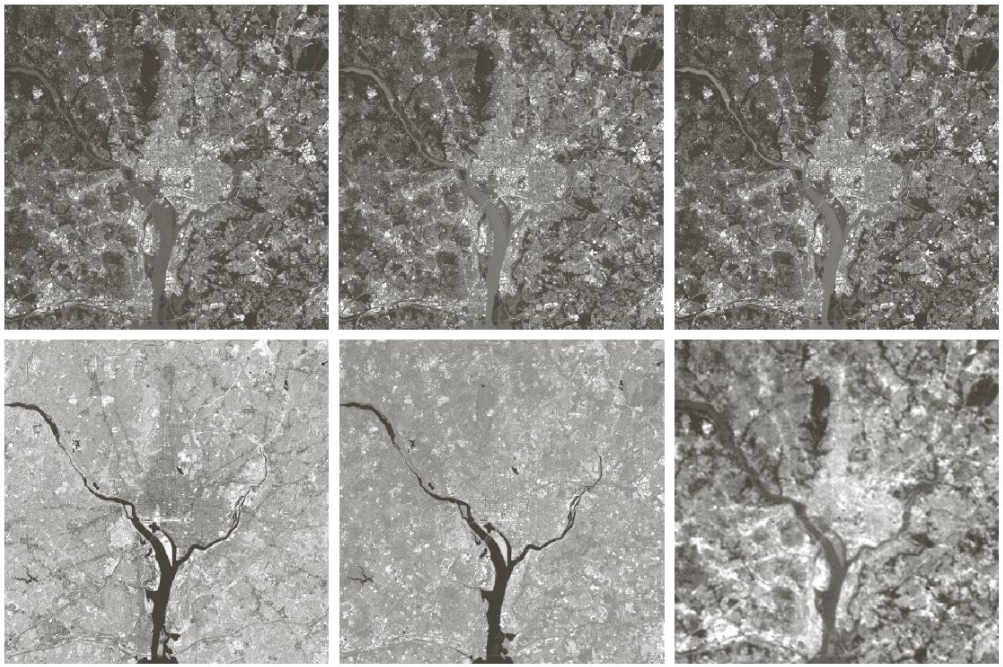
- Hyperspectral images are abbreviated as HSI!
- Hyperspectral images are different from multispectral images. The latter contain few, discrete and discontinuous wavelengths. The former contain many more wavelengths with continuity.



Beyond color: Hyperspectral images

- Widely used in remote sensing (satellite images) – often different materials/geographical entities (soil, water, vegetation, concrete, landmines, mountains, etc.) can be detected/classified by spectral properties.
- Also used in chemistry, pharmaceutical industry and pathology for classification of materials/tissues.

Example multispectral image with 6 bands



a b c
d e f

FIGURE 11.38 Multispectral images in the (a) visible blue, (b) visible green, (c) visible red, (d) near infrared, (e) middle infrared, and (f) thermal infrared bands. (Images courtesy of NASA.)

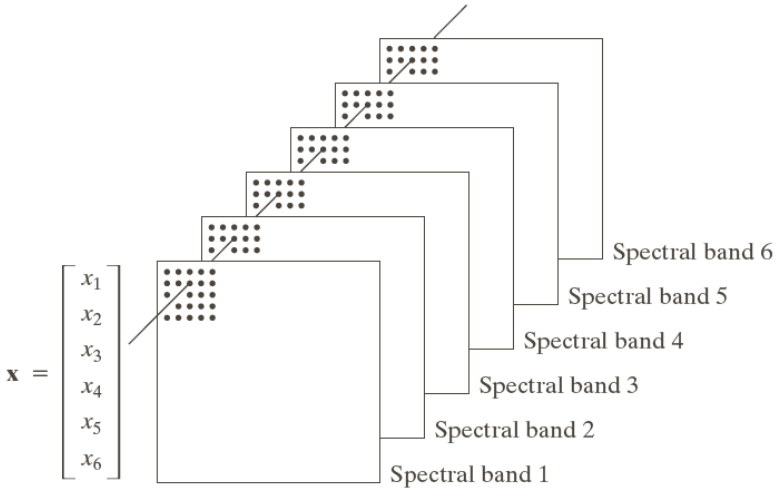
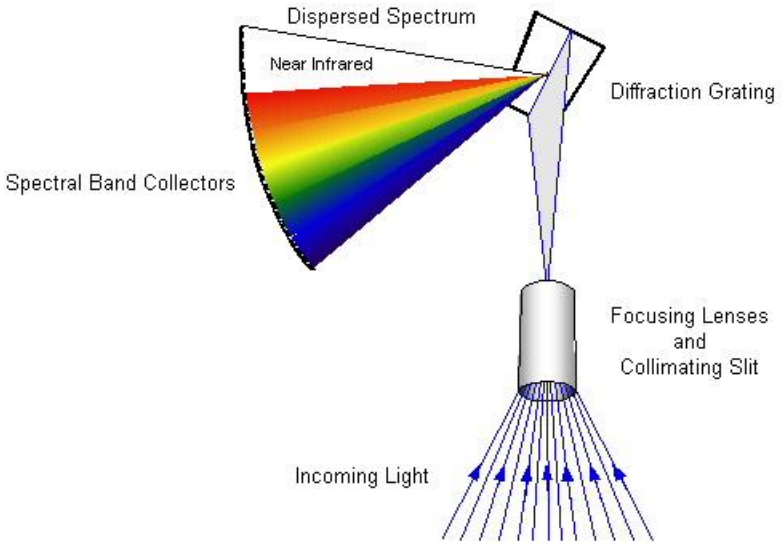


FIGURE 11.39 Formation of a vector from corresponding pixels in six images.

Conventional Hyperspectral Camera

Multiple sensor arrays – one per wavelength. Expensive!

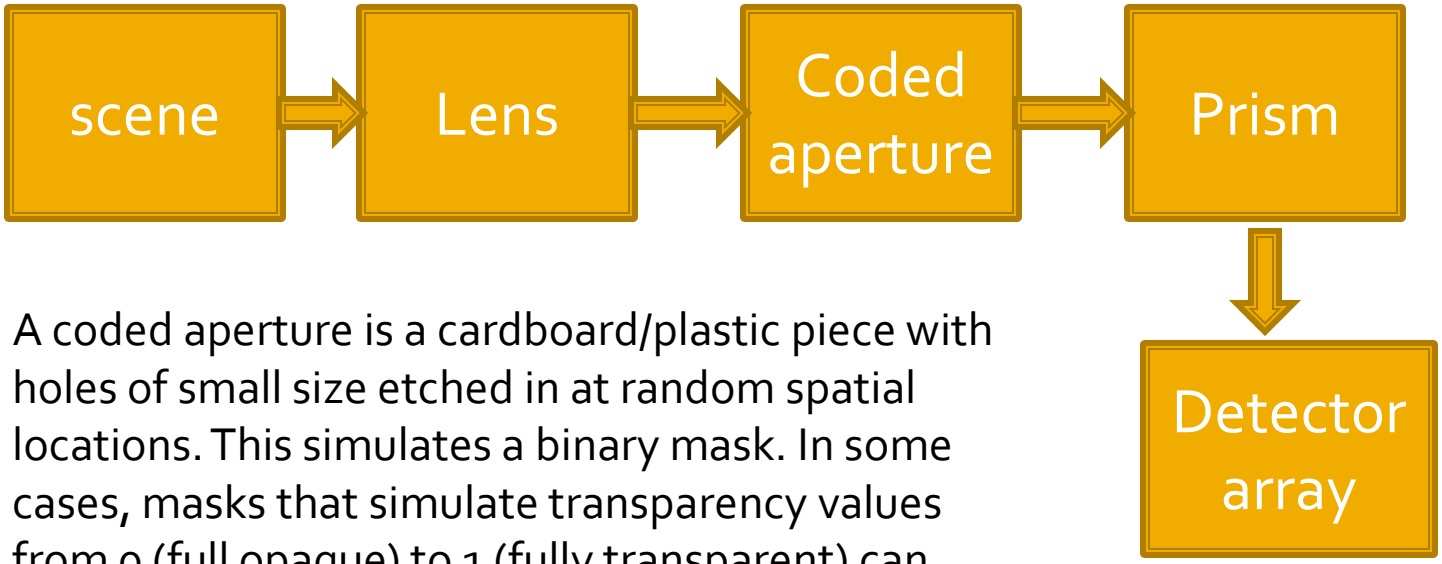


CASSI: Compressive Hyperspectral Image Acquisition

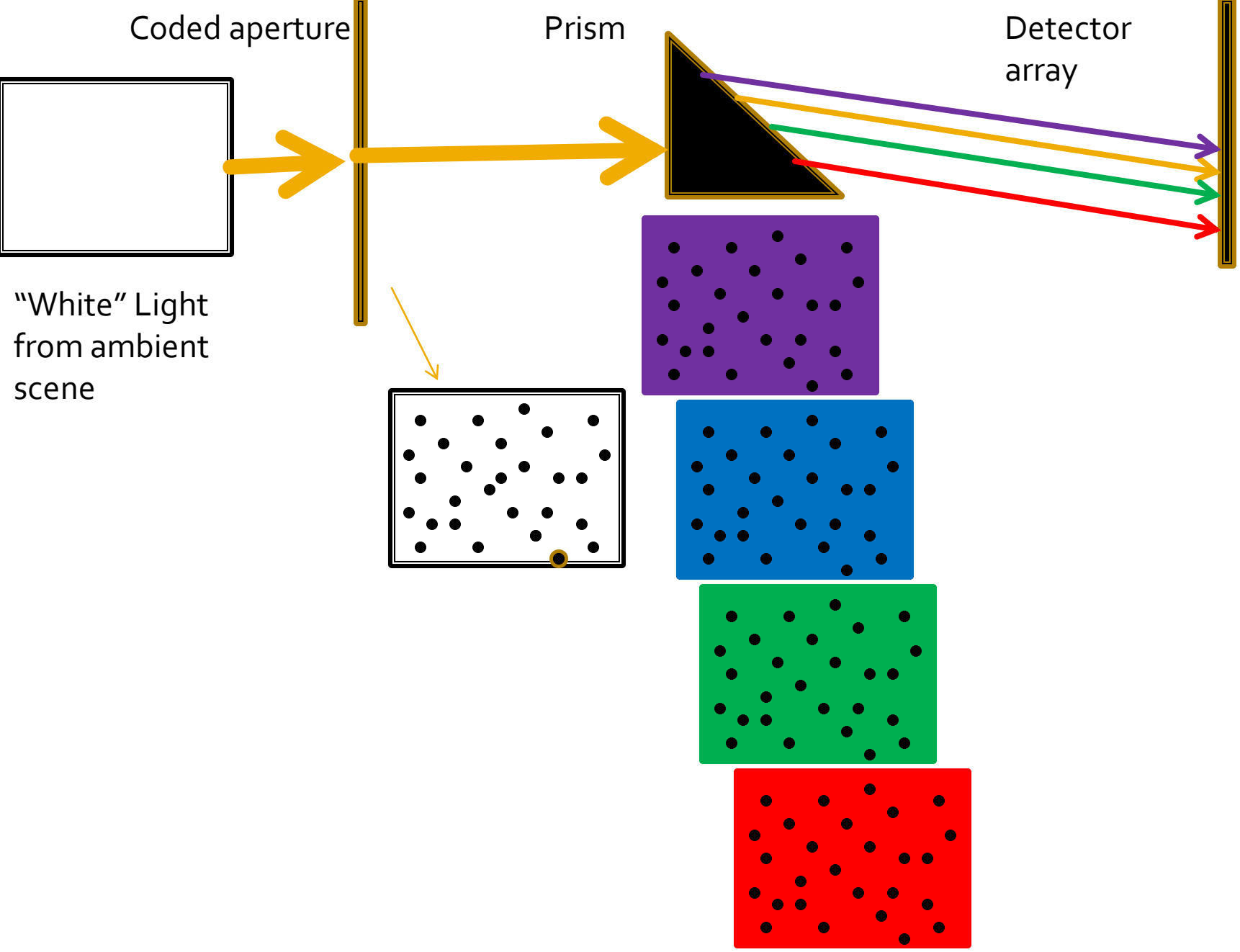
- Reconstruction of hyperspectral data imaged by a coded aperture snapshot spectral imager (CASSI) developed at the DISP (Digitial Imaging and Spectroscopy) Lab at Duke University.
- CASSI measurements are a superposition of aperture-coded wavelength-dependent data: ambient **3D hyperspectral datacube** is mapped to a **2D 'snapshot'**.
- **Task:** Given one or more 2D snapshots of a scene, recover the original scene (3D datacube).

CASSI: Description of Measured Data

Ref: A. Wagadarikar et al, "Single disperser design for coded aperture snapshot spectral imaging", Applied Optics 2008.



A coded aperture is a cardboard/plastic piece with holes of small size etched in at random spatial locations. This simulates a binary mask. In some cases, masks that simulate transparency values from 0 (full opaque) to 1 (fully transparent) can also be prepared.


Coded aperture
Prism
Detector array
"White" Light from ambient scene

CASSI: Description of Measured Data

- Assume we want to measure a hyperspectral data-cube given as $\mathbf{X} \in R^{N_x \times N_y \times N_\lambda}$ where data at each wavelength is a 2D image of size $N_x \times N_y$ where the number of wavelength is N_λ.

- In a CASSI camera, each image $\mathbf{X_j}, 1 \leq j \leq N_\lambda$, is multiplied by the same known (random)binary code given as $\mathbf{C} \in \{0,1\}^{N_x \times N_y}$ yielding an image

$$\hat{\mathbf{X}}_\mathbf{j} = \mathbf{X}_\mathbf{j} \bullet \mathbf{C}.$$

CASSI: Description of Measured Data

- Let the pixel at location (x, y) in image $\hat{\mathbf{x}}_j$ be denoted as $\hat{X}_j(x, y)$. The shifted version of $\hat{X}_j(x, y)$ is given as $S_j(x, y) = \hat{X}_j(x - l_j, y)$ where $l_j > 0$ denotes the shift in the pixels at wavelength $\lambda_j, l_j \neq l_{\hat{j}}, j \neq \hat{j}$.

- The wavelength-dependent shifts are implemented by means of a **prism** in the CASSI camera, whereas modulation by the binary code is implemented by means of a **mask**.

CASSI: Description of Measured Data

- The measurement by the CASSI system is a single 2D "snapshot" given as follows (superposition of coded data from all wavelengths):

$$M(x,y) = \sum_{j=1}^{N_\lambda} S_j(x,y) = \sum_{j=1}^{N_\lambda} \hat{X}_j(x-l_j,y) = \sum_{j=1}^{N_\lambda} X_j(x-l_j,y) \bullet C(x-l_j,y)$$

- Due to the wavelength-dependent shifts, the contribution to *M(x,y)* at different wavelengths corresponds to a different spatial location in each of the slices of the datacube **X**.
- Also the portions of the coded aperture contributing towards a single pixel value *M(x,y)* are different for different wavelengths.

Multi-frame CASSI

- The compression rate of CASSI is the number of wavelengths: 1.
- This compression rate can be reduced if $T > 1$ snapshots of the same scene are acquired in quick succession, denoted as $\{\mathbf{M_t}\}_{t=1}^{T}$ reducing the compression rate to $N_\lambda : T$.

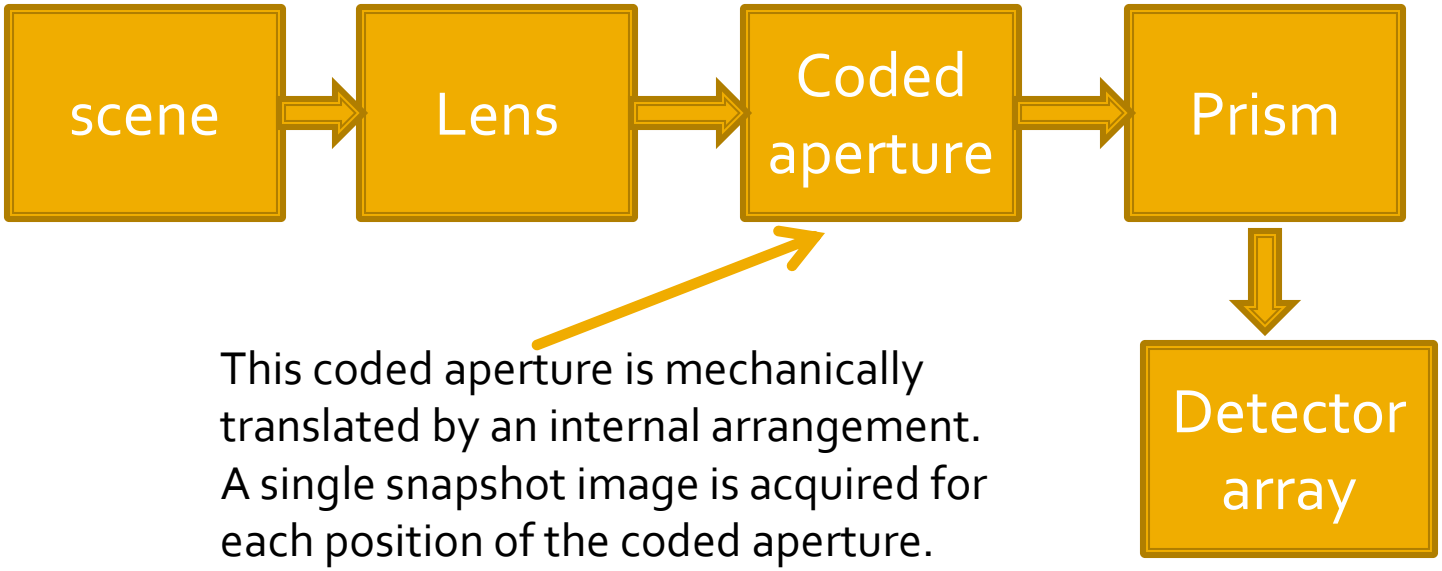
- Each snapshot is acquired using a **different aperture code**, i.e. a **different mask pattern** - implemented in hardware by **moving the position of the mask** using a piezo-electric mechanism.
- Reduction in compression rate = **less ill-posed problem** = scope for better reconstruction.

Multi-frame CASSI

Ref: A. Wagadarikar et al, "Single disperser design for coded aperture snapshot spectral imaging", Applied Optics 2008.

For $t = 1$ to T,

$$M_t(x,y) = \sum_{j=1}^{N_\lambda} S_{j,t}(x,y) = \sum_{j=1}^{N_\lambda} \hat{X}_{j,t}(x - l_j, y) = \sum_{j=1}^{N_\lambda} X_j(x - l_j, y) \bullet C_t(x - l_j, y)$$



Ref: D. Kittle at al, "Multiframe image estimation for coded aperture snapshot spectral imagers", Applied Optics, 2010.

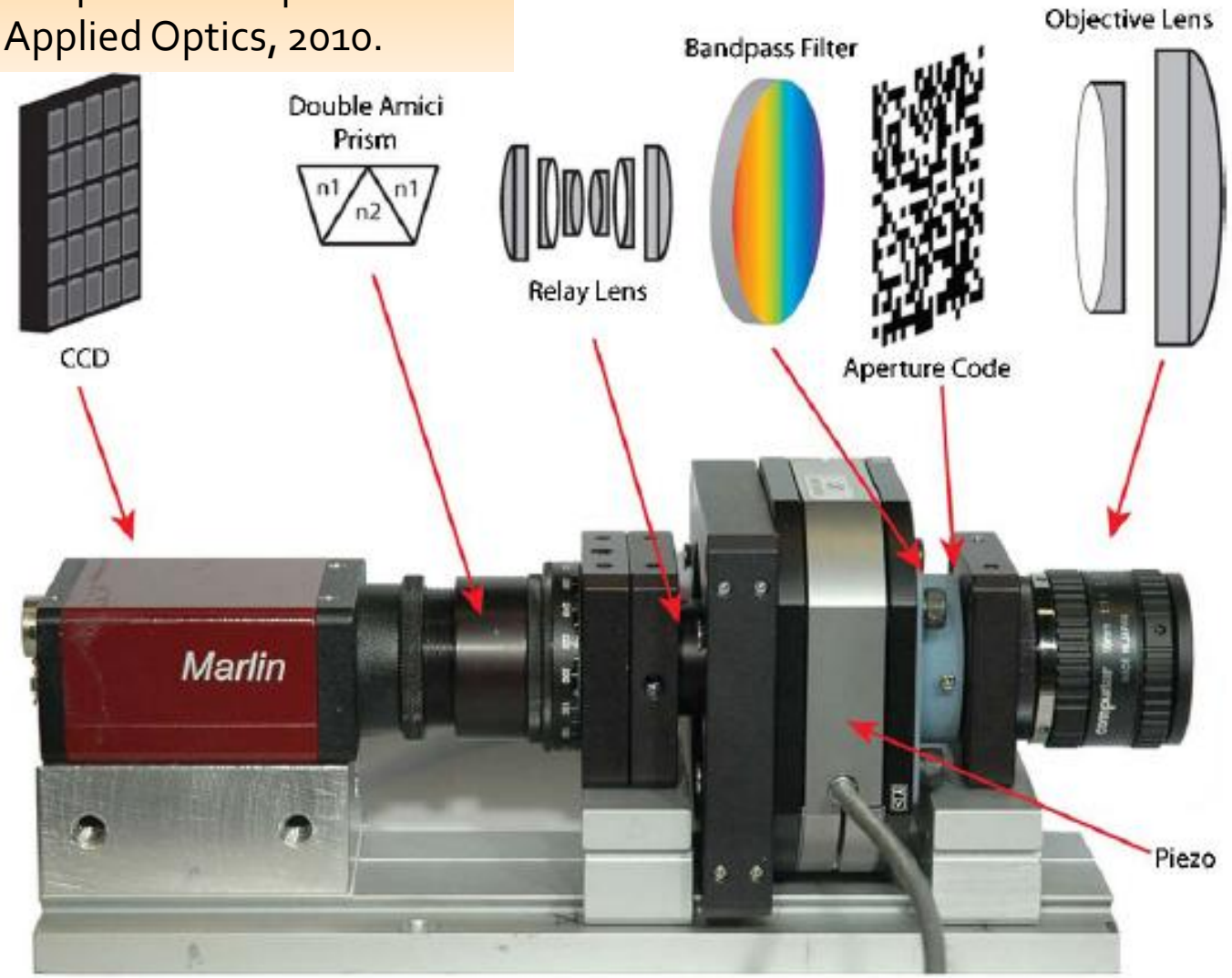
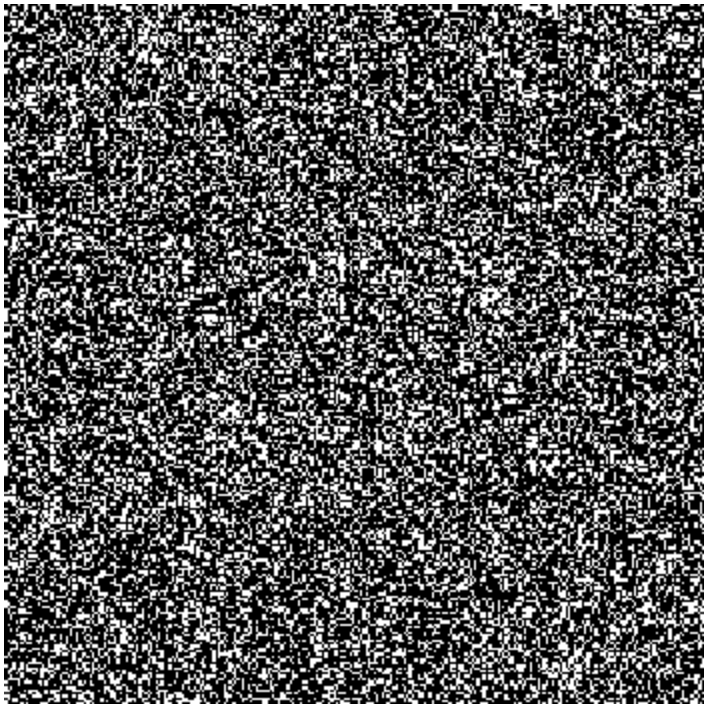
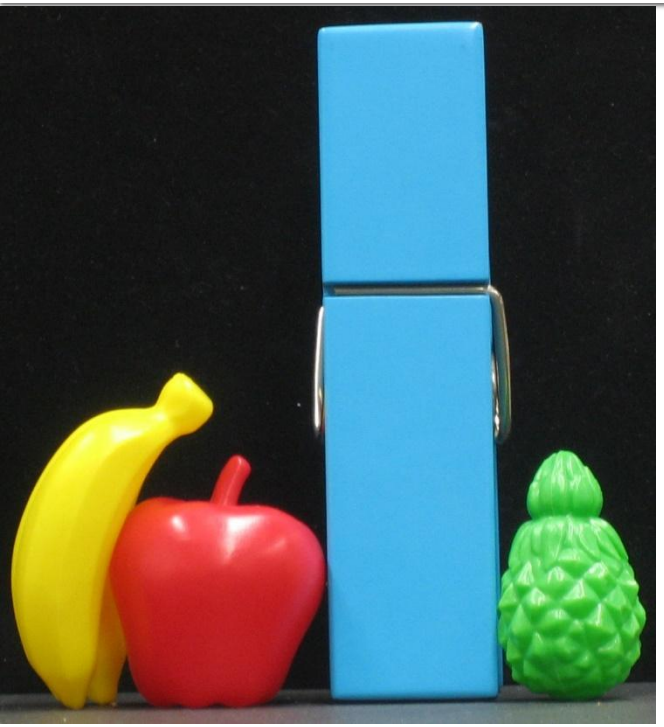


Fig. 2. (Color online) Schematic and photo showing the entire CASSI instrument. Left to right: CCD, double Amici prism, relay lens, piezo stage, bandpass filter, aperture code, and objective lens.

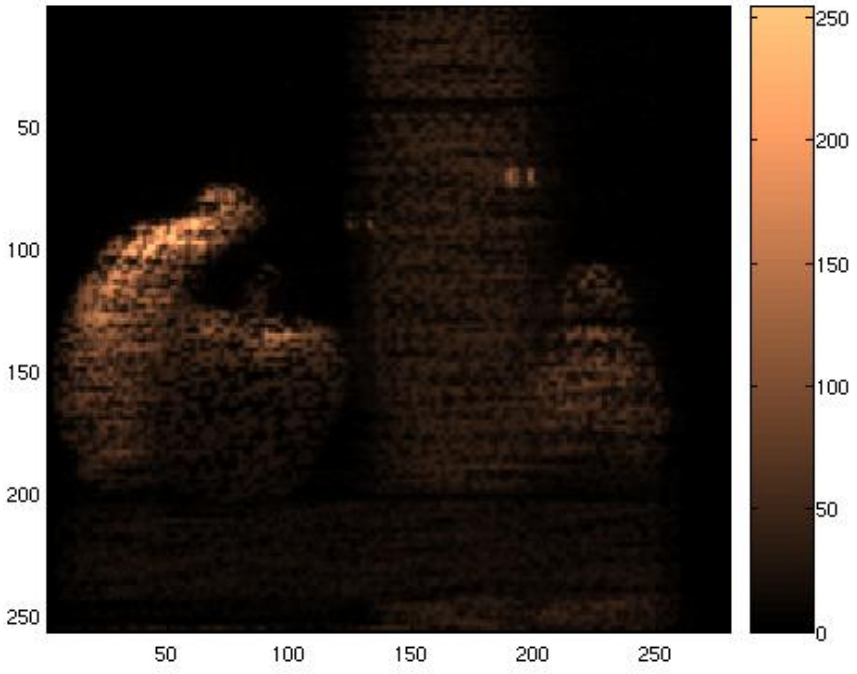


- Aperture Code: created randomly (random binary, [0,1] uniform random also possible)
- The aperture code pattern has holes of size 2 x 2 pixels (smaller holes give rise to diffraction artifacts).
- The pattern is projected onto the detector array in a magnified form.
- Note: Random mask pattern is needed as per CS theory.

Sample CASSI Measurements versus RGB representation of underlying hyperspectral scene



Reference color image (only for reference – NOT acquired by the camera)



Snapshot spectral image acquired by CASSI camera

http://www.disp.duke.edu/projects/CASSI/experimentaldata/index.ptml

Reconstruction Method

- A total-variation based CS solver called as TwIST was used (ref: Bioucas-Dias and Figuereido, A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration", *IEEE Transactions on Image Processing, 2007.*)
- The inversion is performed by solving the following:

$$E(\mathbf{f}^*) = \min_f \sum_t \left\| \mathbf{m}_t - \mathbf{\Phi}_t \mathbf{f} \right\|^2 + \tau TV(\mathbf{f}),$$

$$TV(\mathbf{f}) = \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} \sum_{\lambda=1}^{N_\lambda} \sqrt{(f(x+1,y,\lambda) - f(x,y,\lambda))^2 + (f(x,y+1,\lambda) - f(x,y,\lambda))^2}$$

Reconstruction Method

$$E(\mathbf{f}^*) = \min_f \sum_t \left\| \mathbf{m_t} - \mathbf{\Phi_t f} \right\|^2 + \tau TV(\mathbf{f}),$$

Known forward model (sensing matrix) for the *t*-th snapshot measurement, i.e. $\mathbf{m_t}$ (governed by several factors – the exact aperture code and its position relative to the scene, plus any blurring effects due to the hardware)

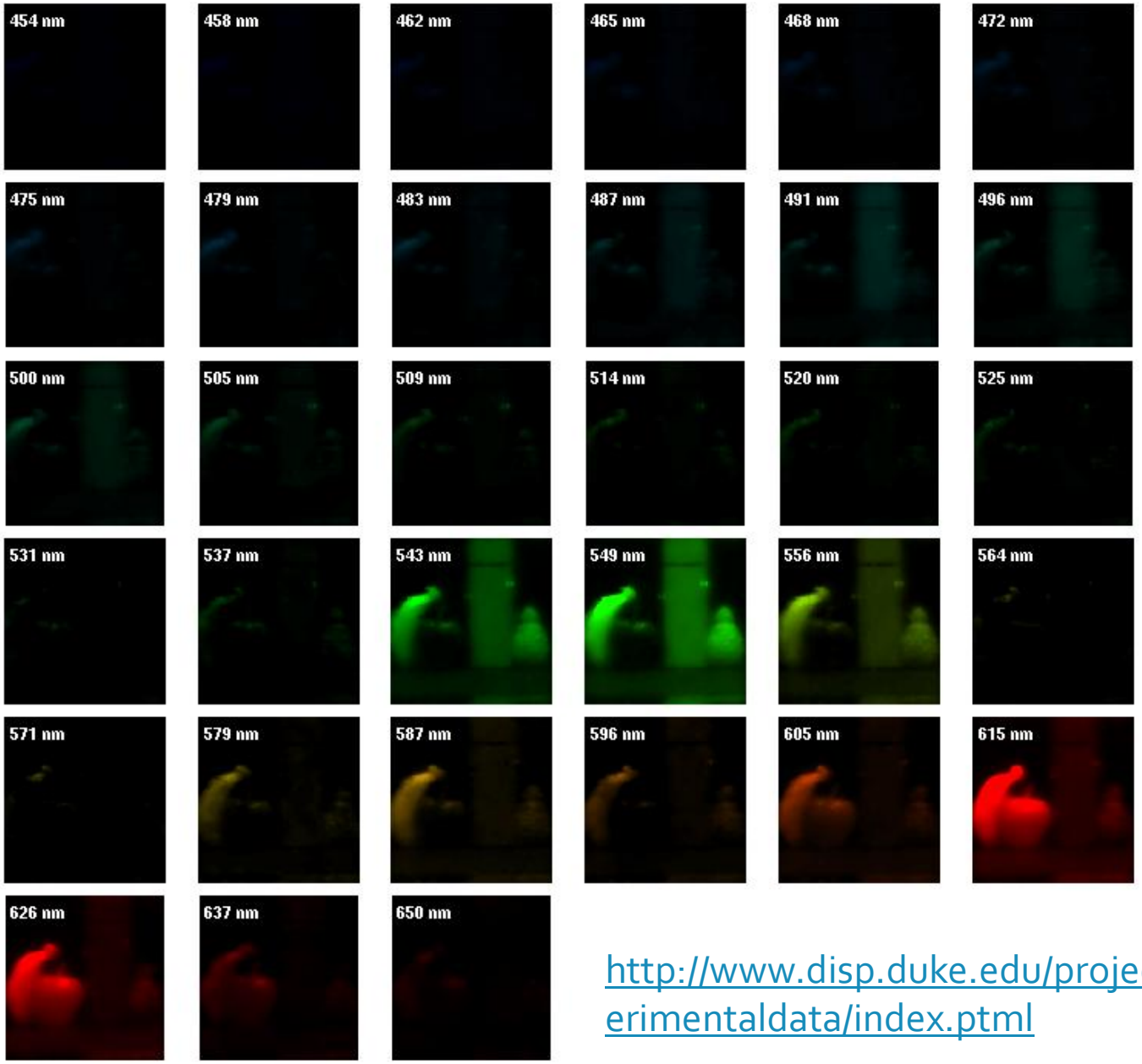
$$\mathbf{\Phi_t} = \left(diag(C_{1,t}) \quad diag(C_{2,t}) \quad . \quad . \quad diag(C_{N_\lambda,t}) \right) \rightarrow \text{size } N_x N_y \times N_x N_y N_\lambda$$

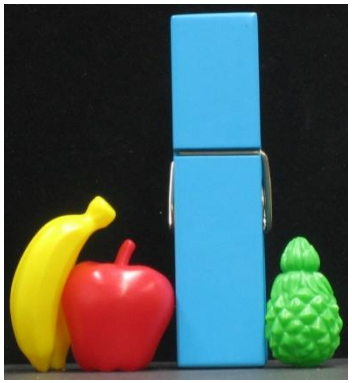
$$\forall l = 1 \text{ to } N_\lambda, \; diag(C_{l,t}) \rightarrow \text{size } N_x N_y \times N_x N_y$$

$$\mathbf{f} = \text{vectorized form of hyperspectral datacube} \rightarrow \text{size } N_x N_y N_\lambda \times 1$$

$$\mathbf{m_t} = \text{vectorized form of snapshot image} \rightarrow \text{size } N_x N_y \times 1$$

Diagonal matrix – whose diagonal is equal to a vectorized form of the coded aperture for the *t*-th snapshot at the shift for the *l*-th spectral band.

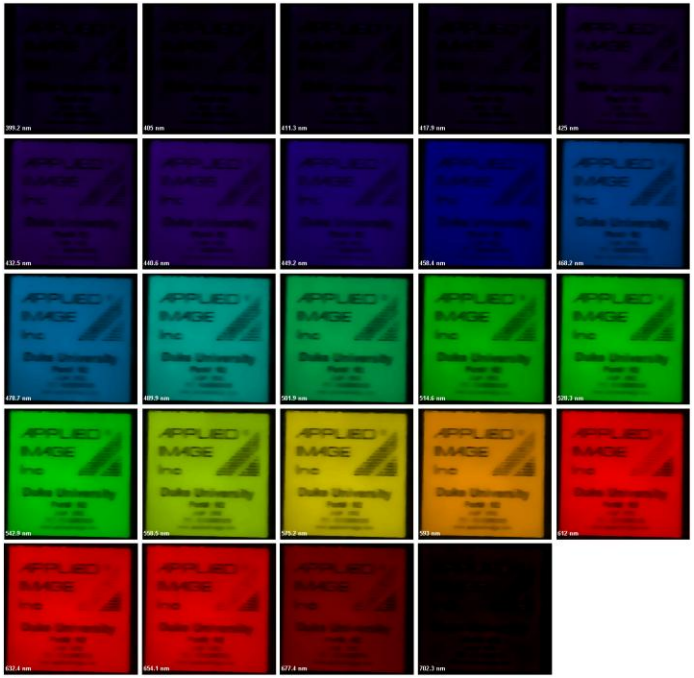


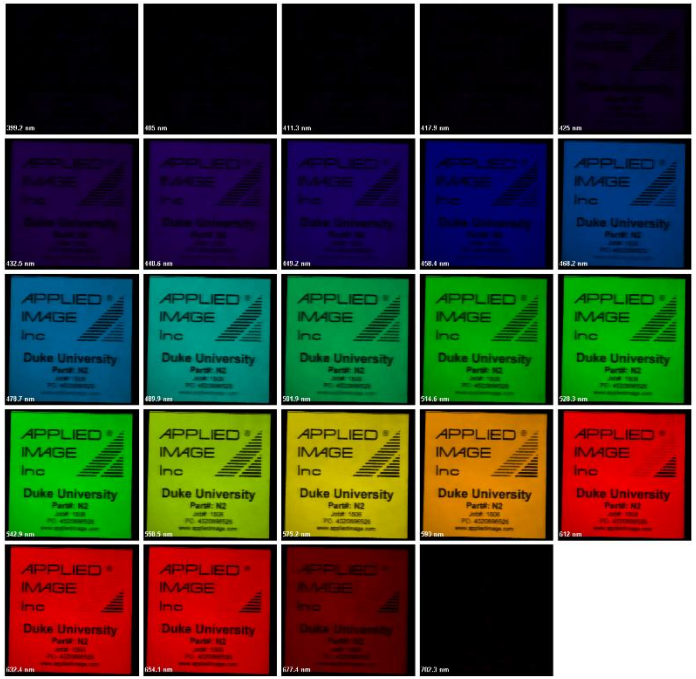


Reference color (RGB) image

http://www.disp.duke.edu/projects/CASSI/experimentaldata/index.ptml

Single versus multi-frame

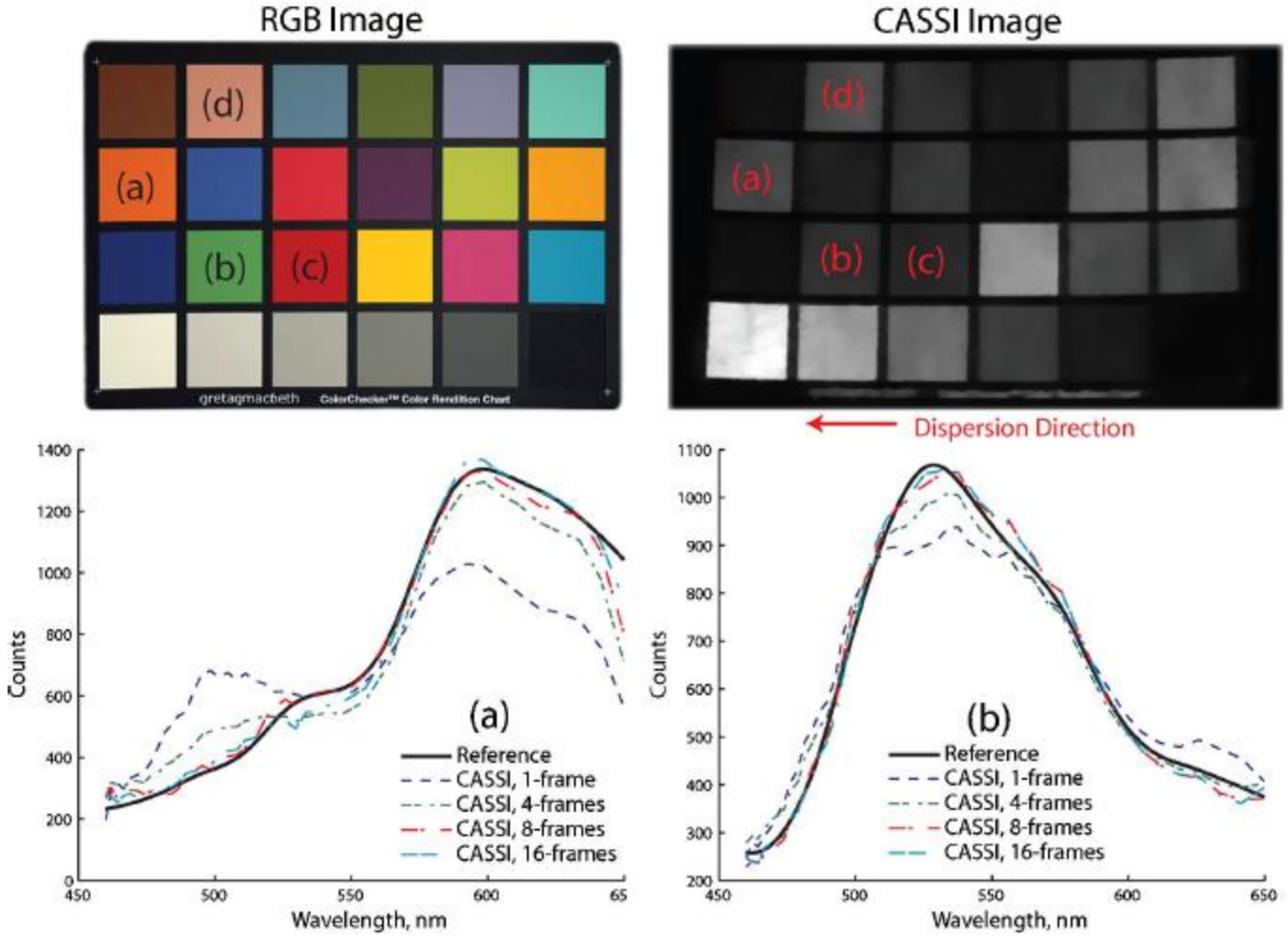




http://www.disp.duke.edu/projects/Multi_CASSI/index.ptml

More results

- Take a look at the following papers:
- Kittle et al, "Multiframe image estimation for coded aperture snapshot spectral imagers"
- *Ajit Rajwade*, David Kittle, Tsung-Han Tsai, David Brady and Lawrence Carin, Coded Hyperspectral Imaging and Blind Compressive Sensing, SIAM Journal on Imaging Sciences (2013)


RGB Image
CASSI Image
(d)
(a)
(b)
(c)
gretagmacbeth
Dispersion Direction
Counts
Wavelength, nm
(a)
(b)
Reference
CASSI, 1-frame
CASSI, 4-frames
CASSI, 8-frames
CASSI, 16-frames

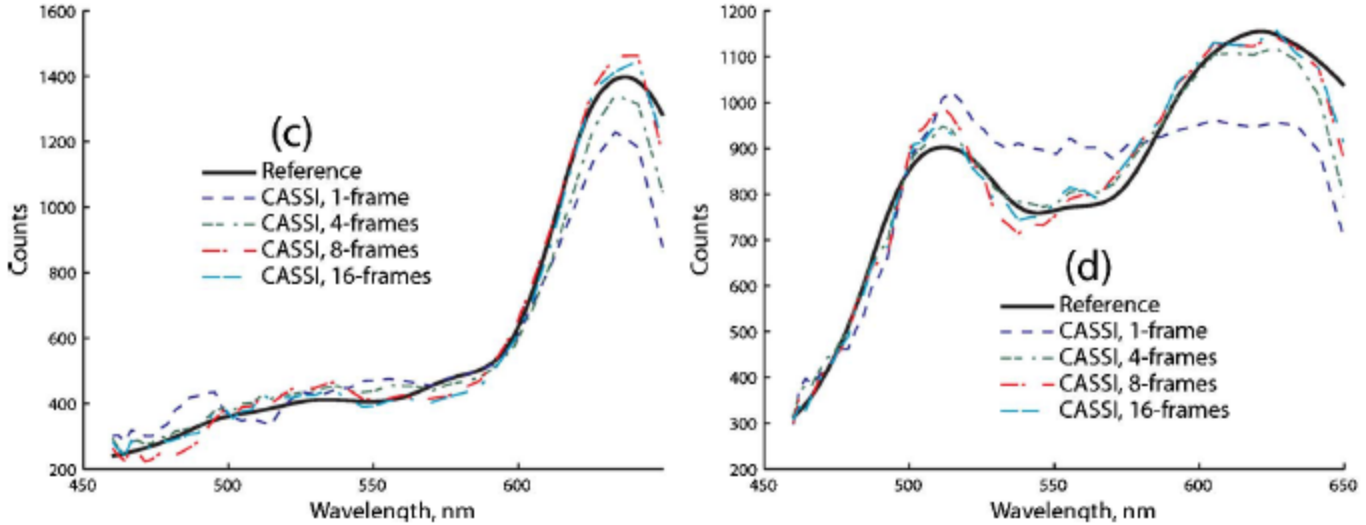


Fig. 5. (Color online) Using the GretagMacbeth ColorChecker as a baseline, comparisons were made with white-light sunlight emulation. (a)–(d) compare snapshot versus 4, 8, and 16 frame reconstructions. Nearby colors can bleed into the spectra, as seen in the RGB image, where the blue just to the right of the orange square was dispersed into the orange. Multiple frames eliminate this problem.

Table 2. RMSE of CASSI Spectra versus Reference Spectrometer

	Center				Edge			
Frames	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
1	9.6%	6.9%	4.1%	17.4%	14.0%	11.7%	8.4%	34.3%
4	2.7%	3.6%	2.4%	4.5%	5.3%	8.1%	2.7%	10.2%
8	3.7%	3.3%	2.6%	5.4%	2.7%	3.6%	2.3%	8.9%
16	1.9%	3.4%	2.8%	5.0%	2.9%	4.1%	2.7%	6.8%

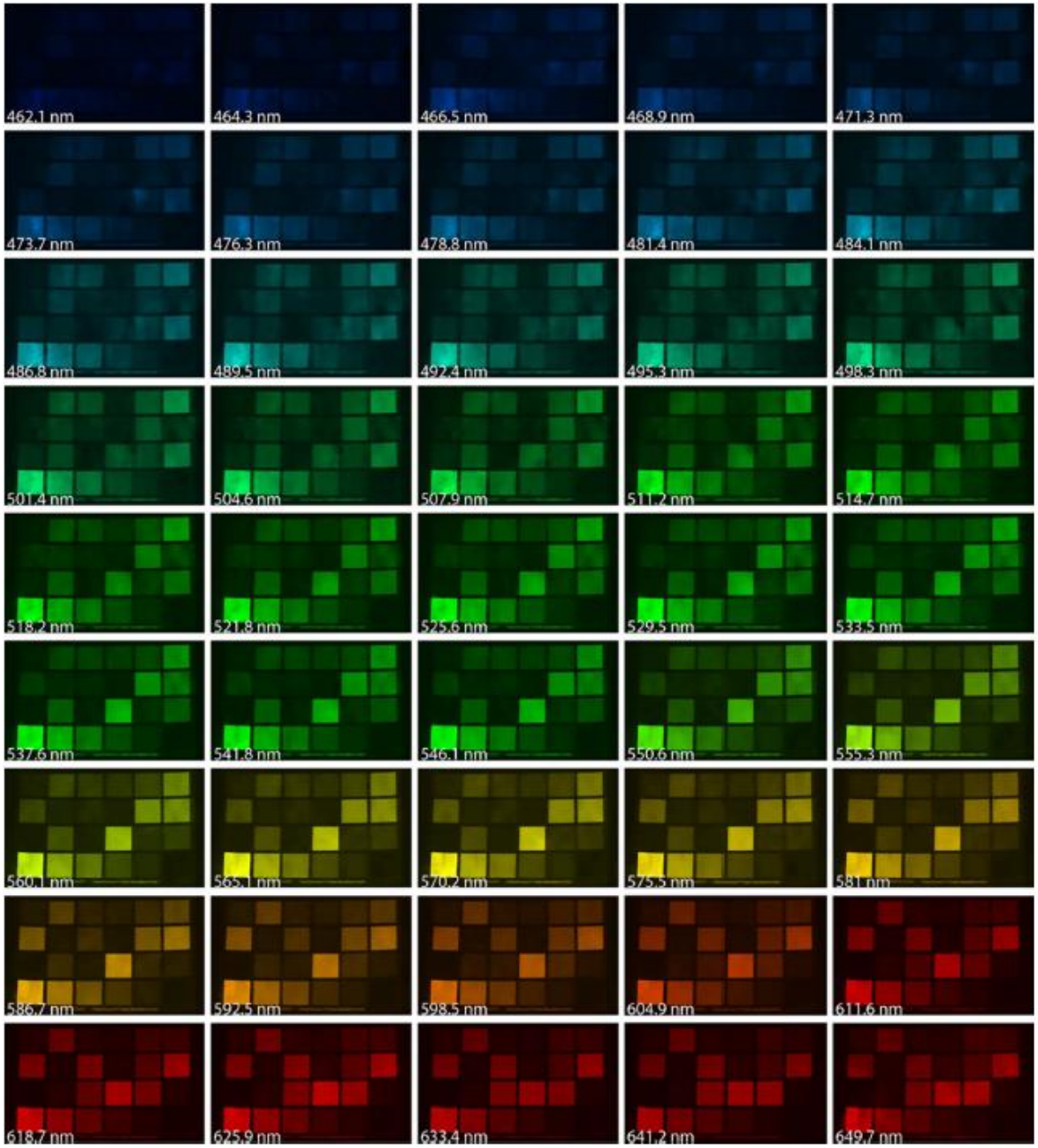


Fig. 6. (Color online) Multiframe CASSI reconstruction of the GretagMacbeth ColorChecker, showing all the channels from 460–650 nm.

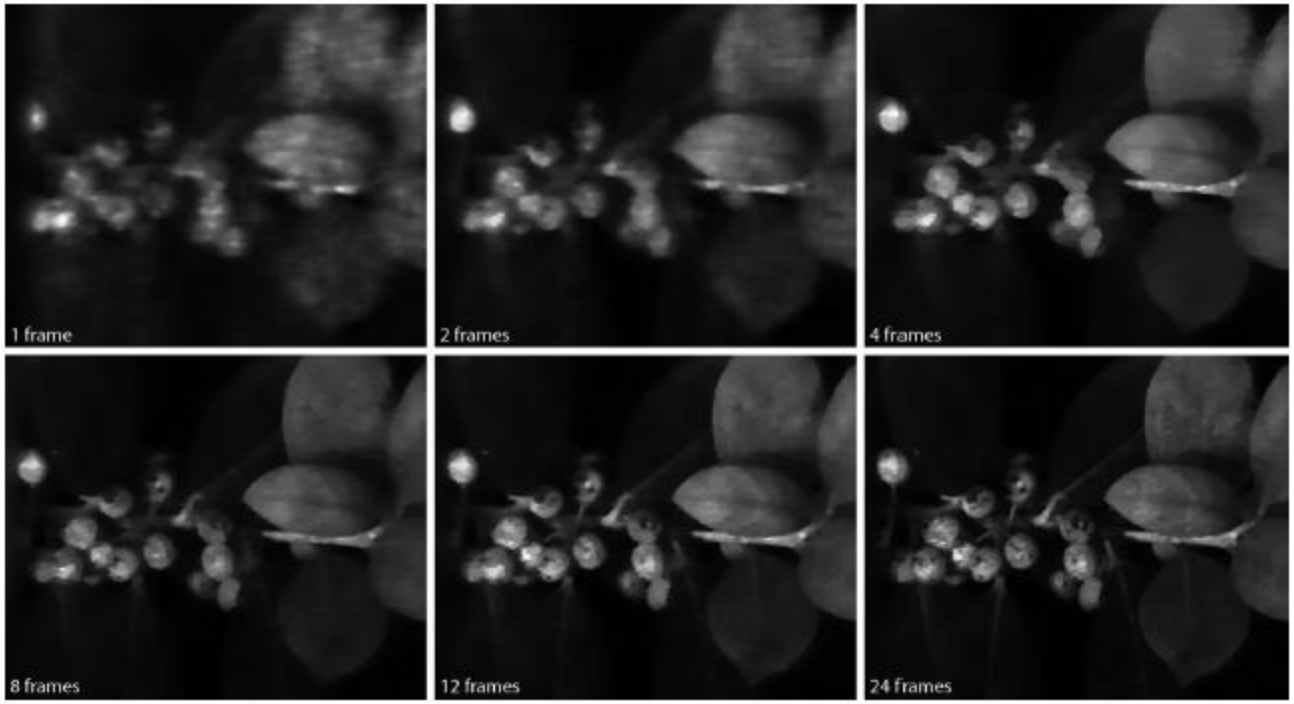


Fig. 7. Comparison between a single snapshot and various multiframe reconstructions for a single channel at 605 nm.

Table 3. PSNR versus Number of CASSI Frames for Actual Data, Referenced to a 24-Frame Image in Fig. 7

Frames	PSNR
1	23.2
2	27.3
4	29.0
8	30.7
12	35.7

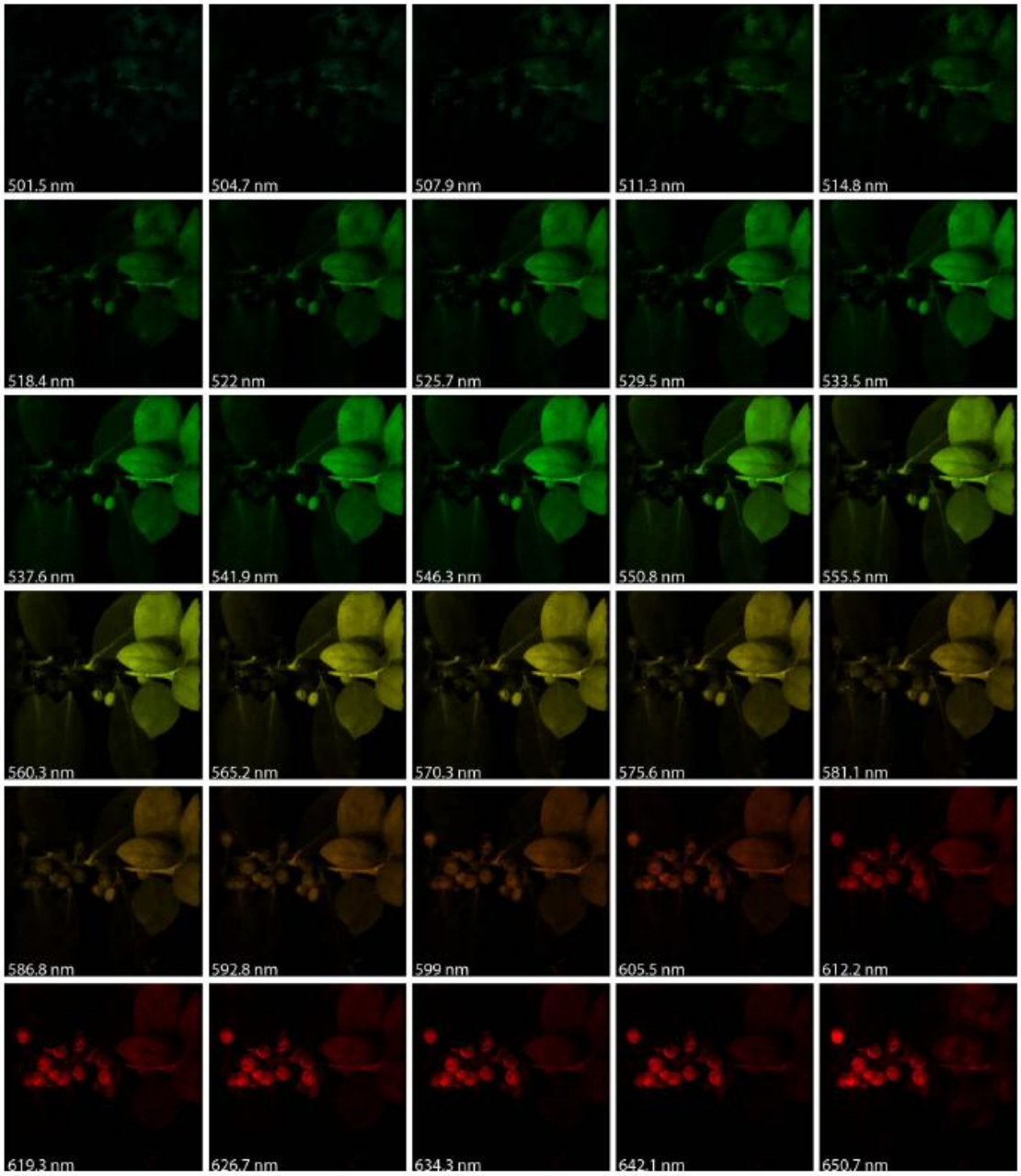


Fig. 8. (Color online) All channels from 500–650 nm from a 24-frame reconstruction of the data cube.

Importance of coded aperture

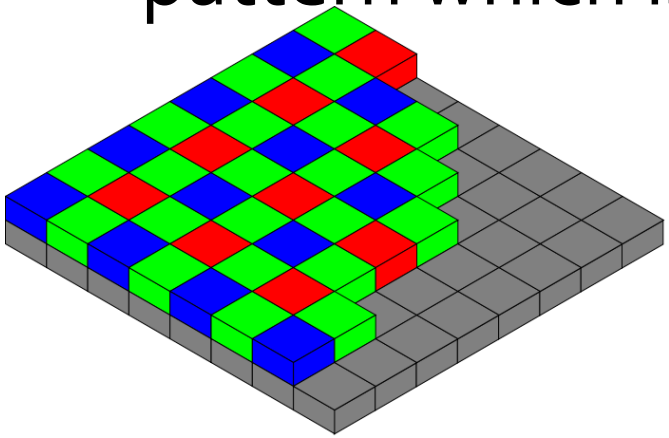
- The coded aperture allows for lower coherence values between the sensing matrix Φ and the orthonormal basis Ψ.
- No coded aperture = $\forall t, x, y\, C_t(x, y) = 1$.
- No coded aperture = multi-frame CASSI not possible.

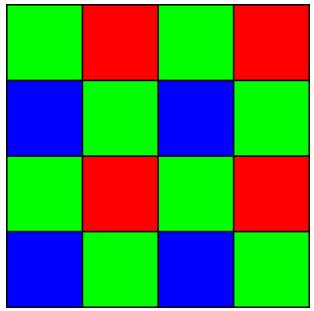
Related topic: (RGB) Color Filter Arrays

- A color image camera typically does not measure R,G,B values of a pixel – it measures just one of them!
- A color filter array is an array of tiny color filters, each filter placed before one sensor element, from the image sensor array of a camera.
- The resolution of this array is the same as that of the image sensor array.
- Each color filter may allow a different wavelength of light to pass – this is pre-determined during the camera design.

Color Filter Arrays

- The most common type of CFA is the Bayer pattern which is shown below:



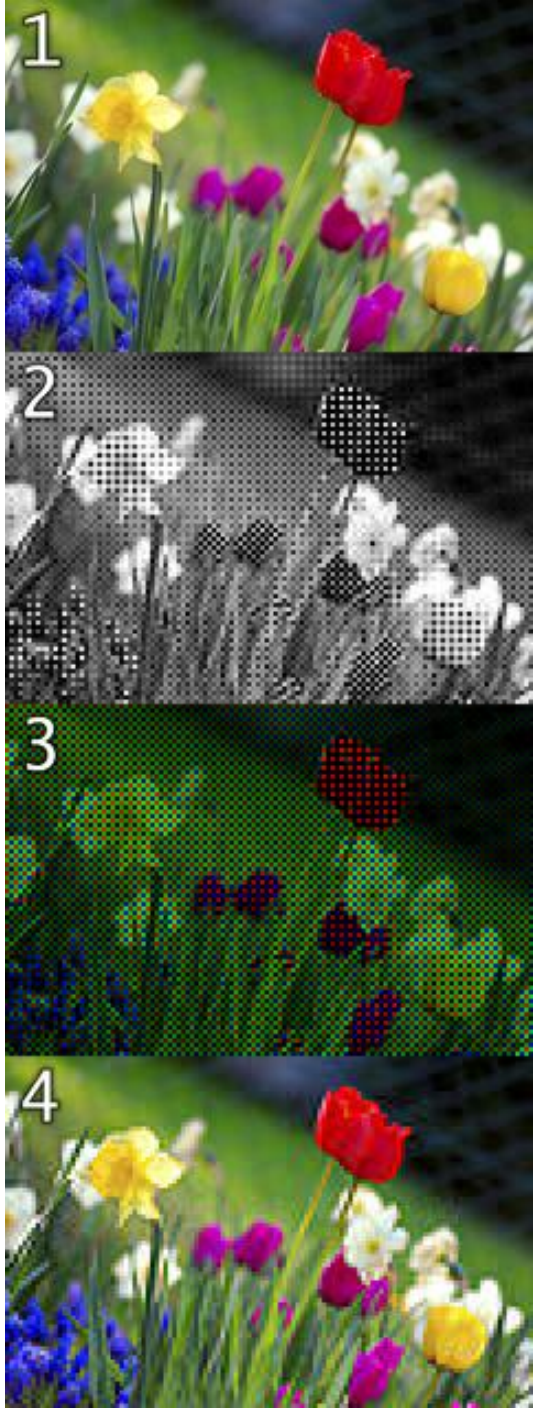


https://en.wikipedia.org/wiki/Color_filter_array

- The Bayer pattern collects information at red, green, blue wavelengths only as shown above.

Color Filter Arrays

- The Bayer pattern uses twice the number of green elements as compared to red or blue elements.
- The **raw** (uncompressed) output of the Bayer pattern is called as the **Bayer pattern image** or the **mosaiced** (*) **image**.
- The mosaiced image needs to be converted to a normal RGB image by a process called color image **demosaicing**.



"original scene"

https://en.wikipedia.org/wiki/Bayer_filter

Mosaiced image

Mosaiced image – just coded with the Bayer filter colors

"Demosaiced" image – obtained by interpolating the missing color values at all the pixels

CASSI versus CFAs

- The CASSI camera has a prism and this causes wavelength-dependent shifts.
- A CFA operates using per-pixel filters, and it has no prisms in it. There are no pixel-dependent shifts.
- A CASSI camera operates for a very large number of wavelengths.

PART 4: COMPRESSIVE VIDEO ACQUISITION USING CODED SNAPSHOTS

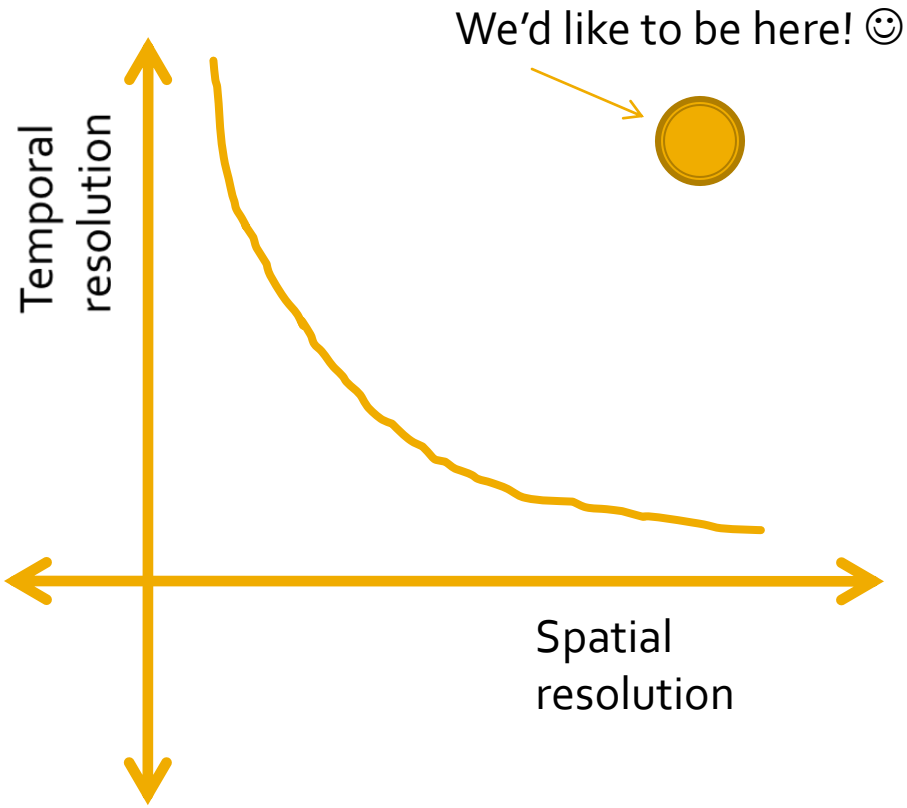
Video from a Single Coded Exposure Photograph using a Learned Over-Complete Dictionary

Authors: Yasunobu Hitomi, Jinwei Gu, Mohit Gupta, Tomoo Mitsunaga,
Shree Nayar
Published in ICCV 2011

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



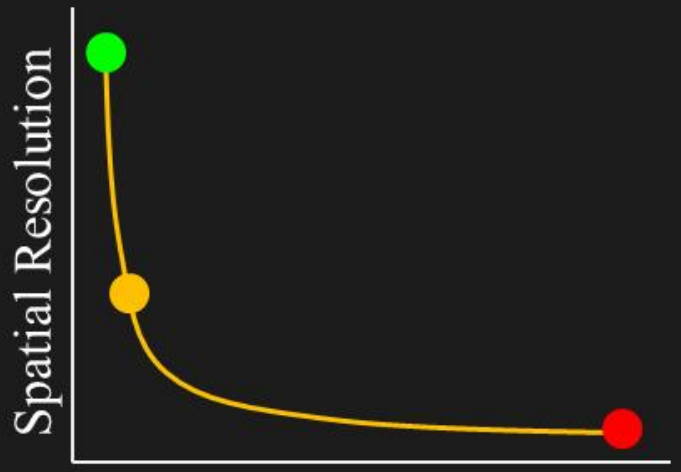
Sampling every *k*-th row of an image frame:

- 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

Spatio-Temporal Resolution Tradeoff

Temporal Resolution: 1X



Temporal Resolution: 4X

Temporal Resolution: 36X

http://www.cs.columbia.edu/CAVE/projects/single_shot_video/

Coded Exposure Image

It is a coded superposition (i.e. summation) of *T* sub-frames within a unit integration time of the video camera.

Coded exposure image (captured in one unit integration time of the camera)

$$I(x,y)=\sum_{t=1}^{T} S(x,y,t)\bullet E(x,y,t)$$

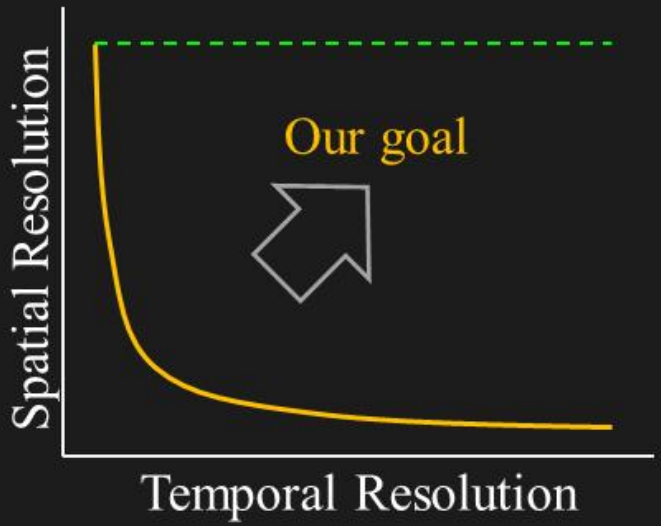
Binary code at time instant *t*

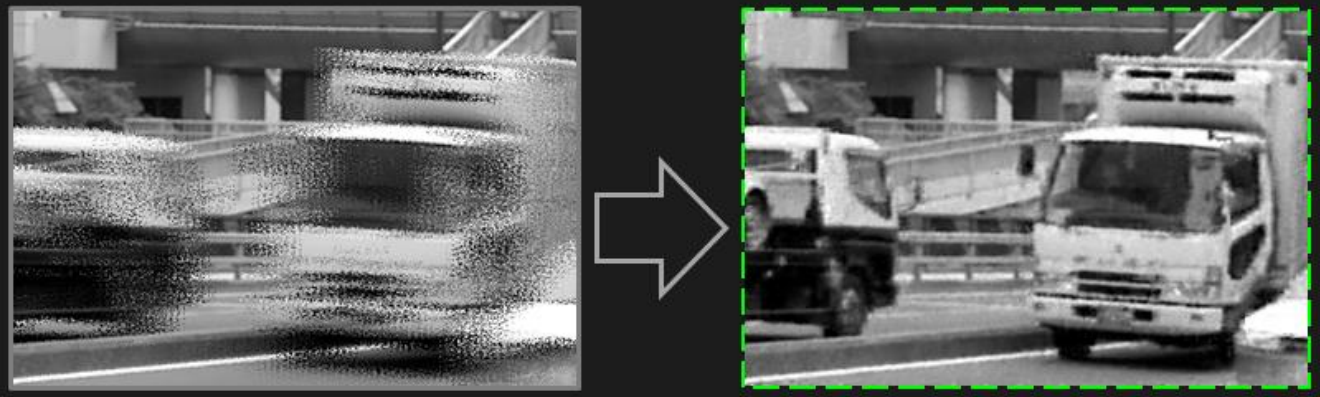
Sub-frame at time instant *t*

Conventional capture (simple integration across time, without modulation by binary codes)

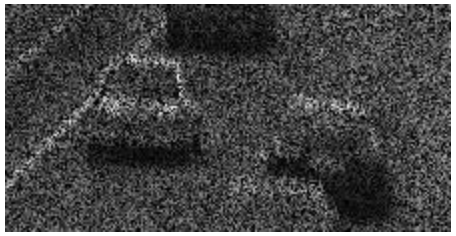
$$\forall x,y,t\ S(x,y,t)=1$$

Our Goal: Overcome the Resolution Tradeoff

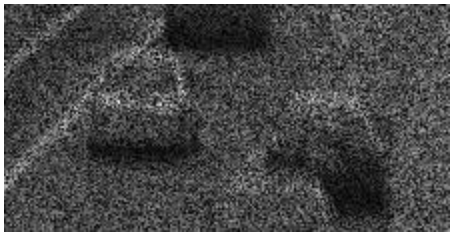




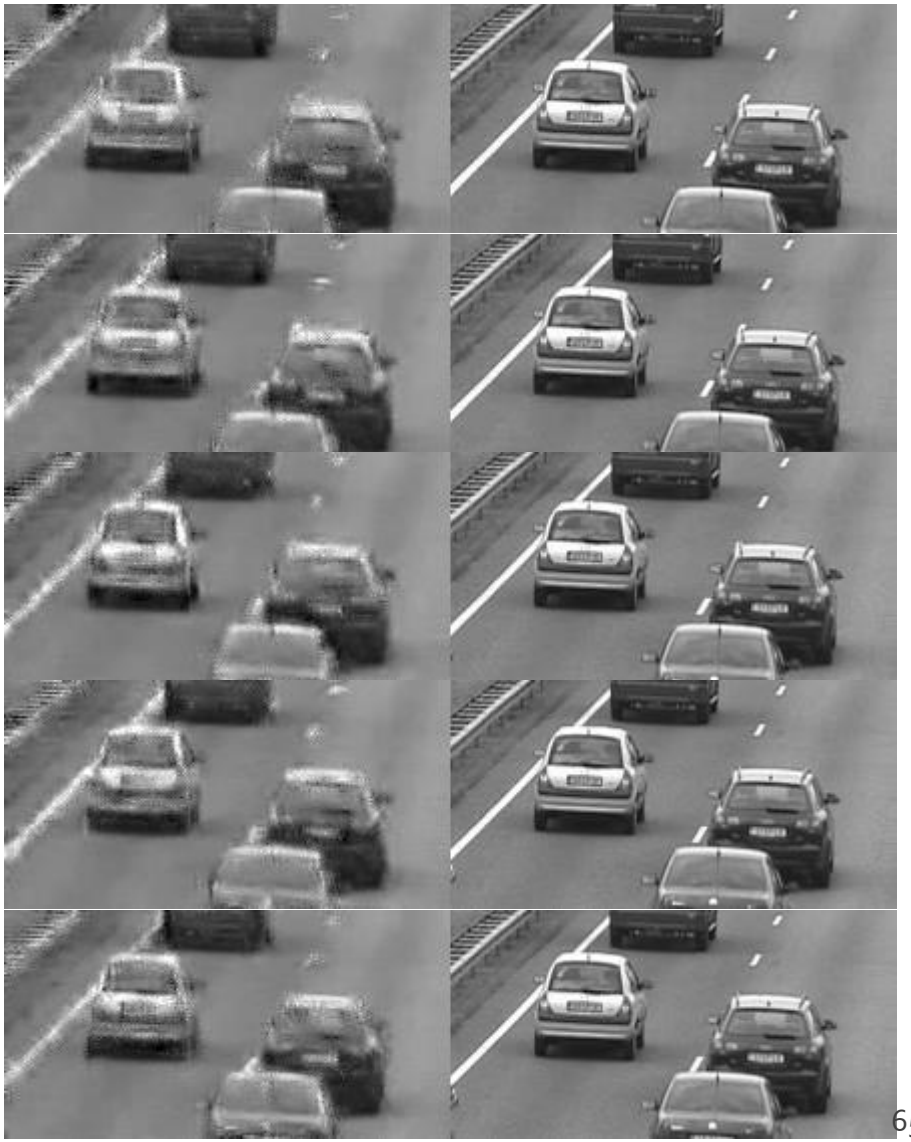
http://www.cs.columbia.edu/CAVE/projects/single_shot_video/



T=3



T=5



Explanation

- Imagine a **30 fps** off the shelf video-camera. It acquires one frame in **1/30 seconds** (this is the **unit integration time of the video camera**).
- The camera model in this paper will acquire a **coded exposure image** *I(x,y)* in the **same** amount of time.
- From this coded exposure image, we will be able to **reconstruct *N* = *20* sub-frames** (all showing changes that occurred in the scene **within the 1/30 seconds period**), using a standard Compressive Sensing Reconstruction algorithm.
- Thus we are doing **20-fold temporal super-resolution**, and that too **without sacrificing spatial resolution**.
- Effectively we are **increasing the camera frame rate from 30 fps to 30 x 20 = 600 fps!**
- Note that such a camera acquires a sequence of such coded exposure images.

Sparse Coding

$$I(x,y) = \sum_{t=1}^{T} S(x,y,t) \bullet E(x,y,t) \rightarrow \mathbf{y} = \mathbf{\Phi f}$$

$$\mathbf{f} = \mathbf{\Psi \theta}$$

$$\therefore \mathbf{y} = \mathbf{\Phi \Psi \theta}$$

y is a vectorized form of the snapshot image **I** of *n* pixels.
f is a vectorized form of the underlying video with *T* sub-frames and each sub-frame having *n* pixels, hence **f** has *nT* pixels in total.
Ψ is a 3D DCT basis – the basis can also be "learned offline on a representative set of videos"
Φ Is a matrix of size *n* x *nT* containing values from the binary code **S**. See below:

$$\mathbf{\Phi} = (diag(S_1) \,|\, diag(S_2) \,|\, \ldots \,|\, diag(S_T))$$

$$S_t(x,y) = S(x,y,t) \text{ for } t = 1 \text{ to } T$$

Diag(S_t) represents a *n* x *n* diagonal matrix. The diagonal contains elements from the mask S_t.

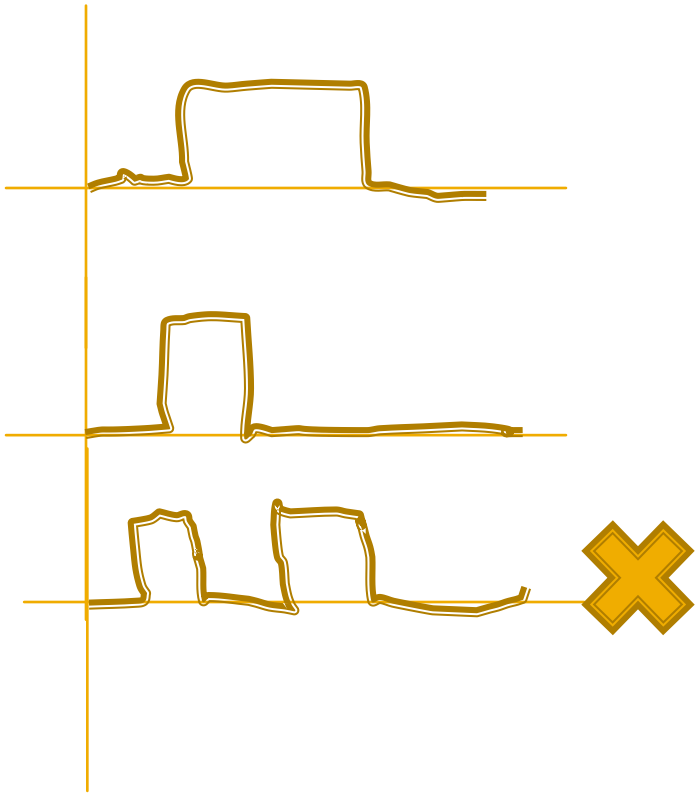
Code Design: S

(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

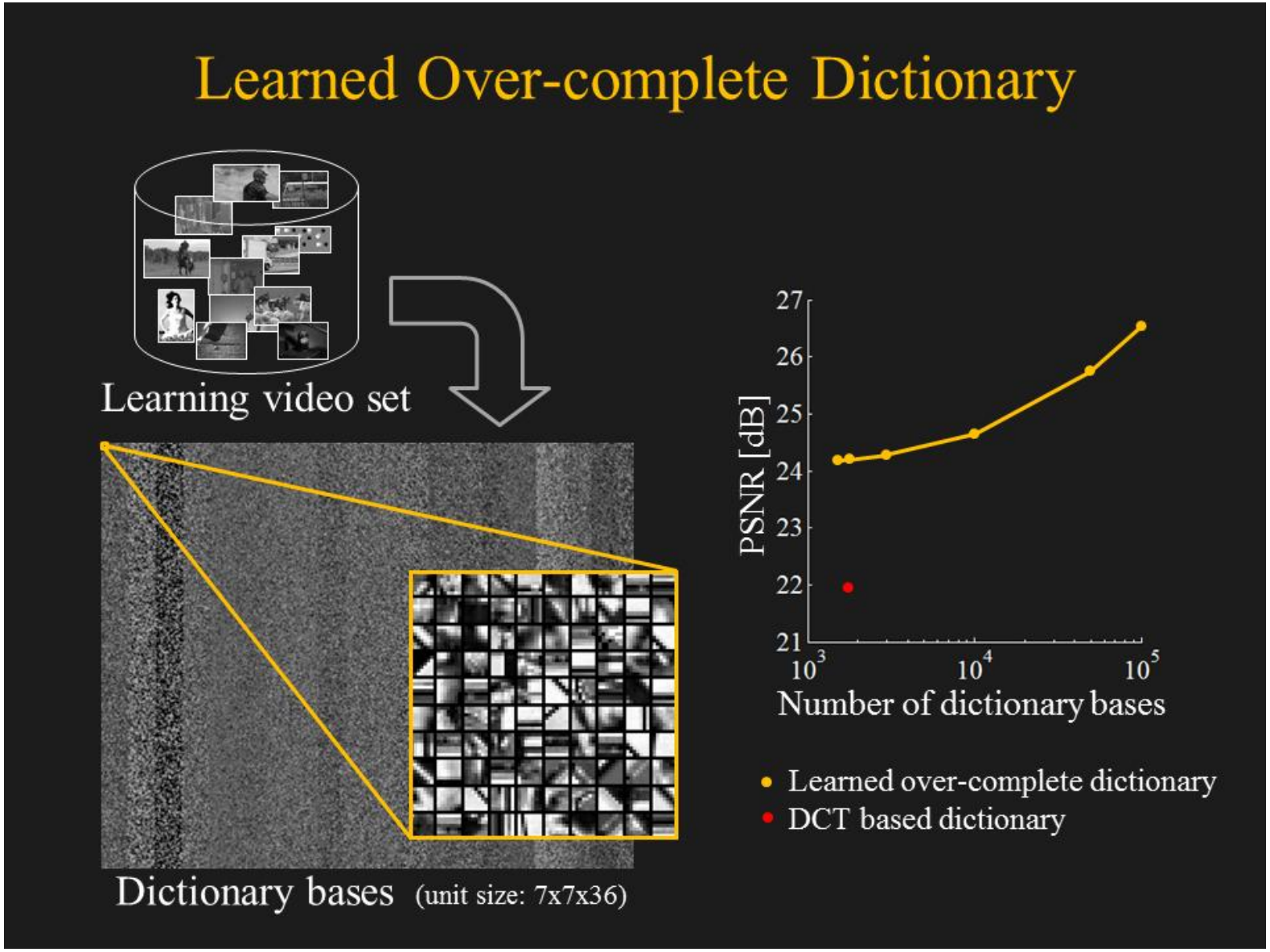
(4) Union of bumps within an *M* x *M* spatial patch should **cover full integration time**



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)

KSVD is a dictionary learning technique. Given a set of input patches in n X N, it learns a set of vectors (n X K), such that a sparse linear combination of these vectors approximates each patch as closely as possible. There are K coefficients per patch, out of which most are encouraged to be 0 (sparse).



http://www.cs.columbia.edu/CAVE/projects/single_shot_video/

Video reconstruction results on real data available below:

https://www.youtube.com/watch?v=JAYCoC3NIdY

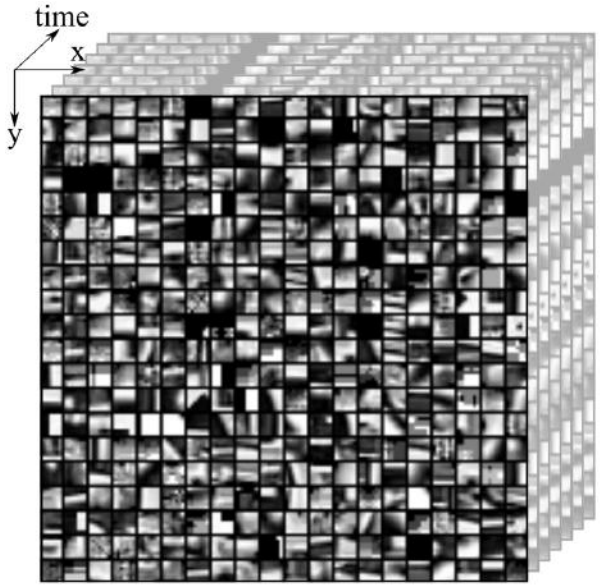
Results on toy-data

3D DCT (1764 bases)
PSNR = 21.95

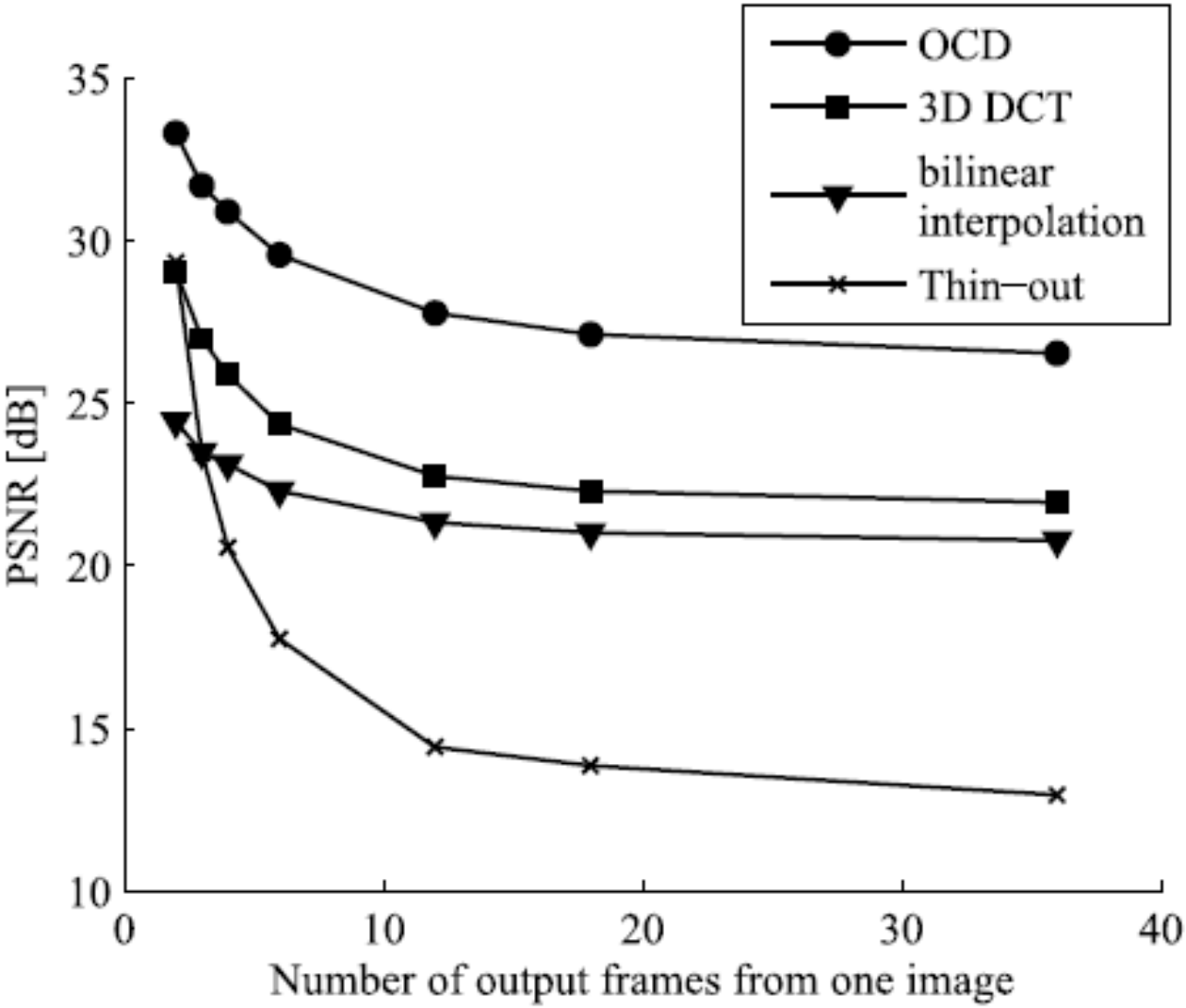
3D DWT (1764 bases)
PSNR = 15.78

Learned Dictionary (1764 bases)
PSNR = 24.21

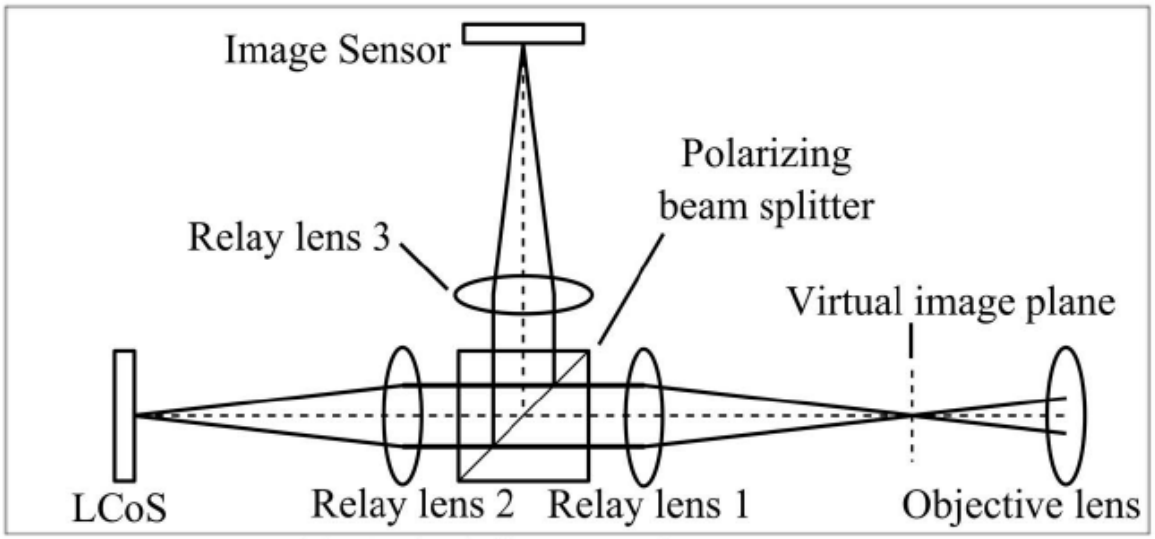
Learned Dictionary (100k bases)
PSNR = 26.54



36X frame rate gain

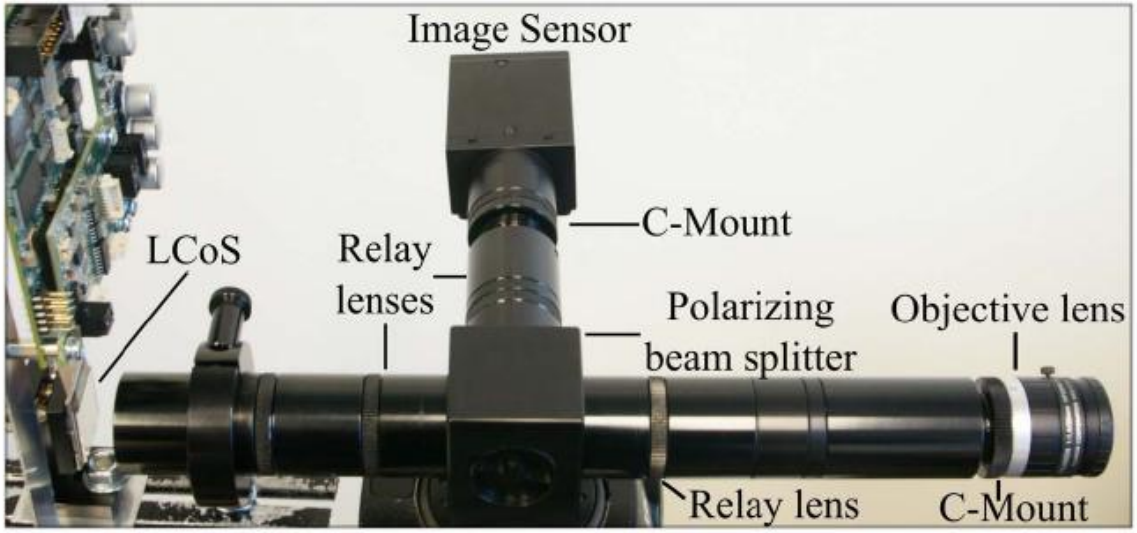


Bilinear interpolation – Uses simple grid-based down-sampling of space-time volume.



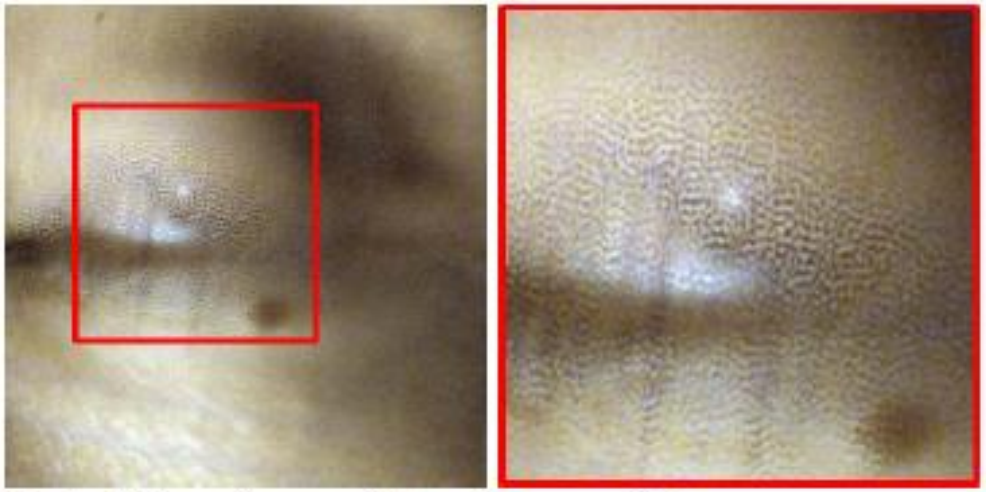
(a) Optical diagram of our setup

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

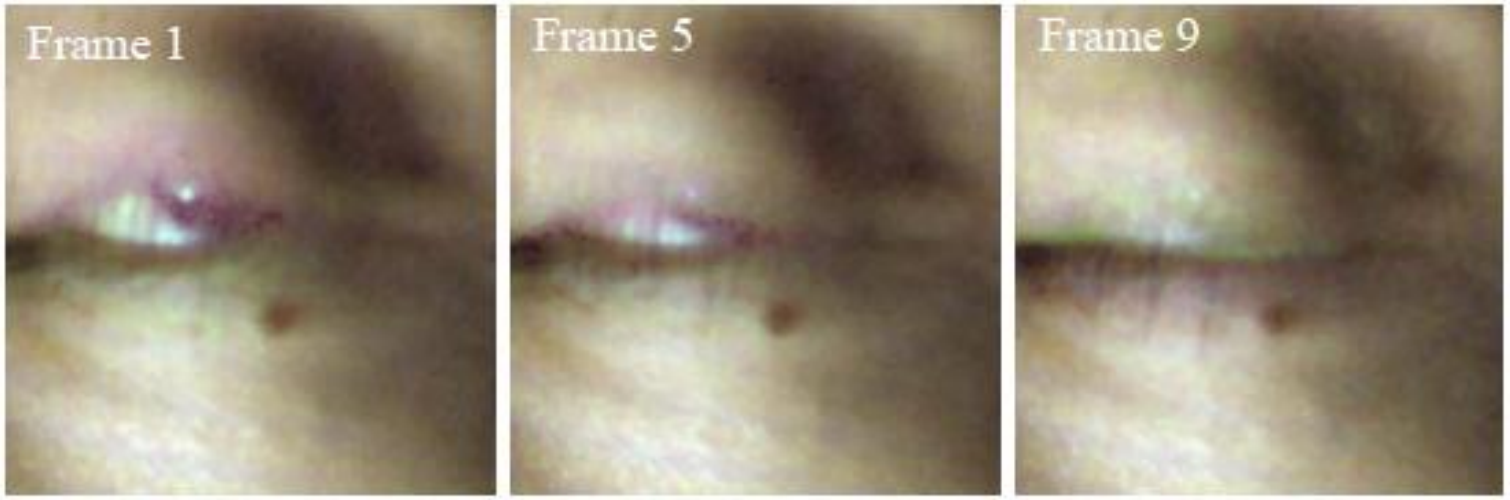


(b) Image of our setup

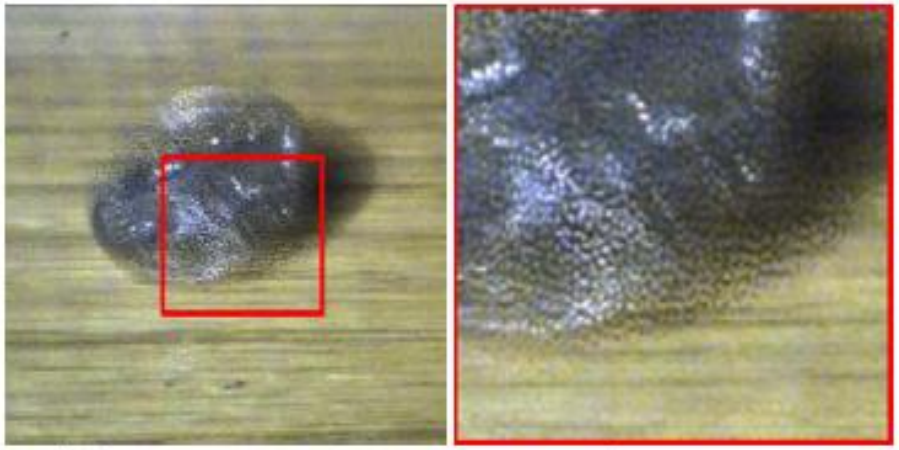
LCoS is synced with the camera and operates at 9-18 times camera frame rate



Coded Exp. Image (27ms) Close-up

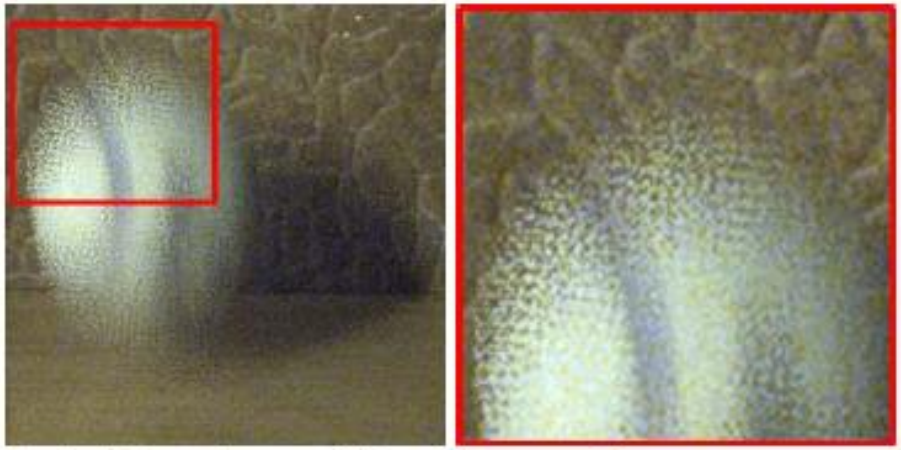


Reconstructed Frames (3 out of 9)

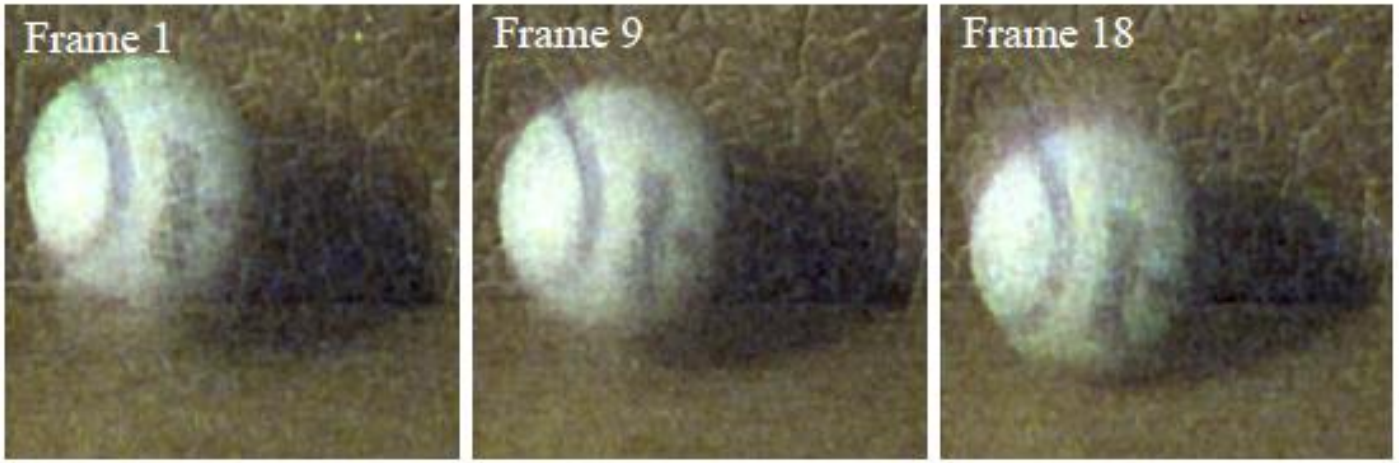


Coded Exp. Image (27ms) Close-up

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.

CS in MRI

Recall: pre-requisites for CS

- Signal should be sparse in some orthonormal basis
- The measurement system should be incoherent with the representation basis

Magnetic Resonance Imaging (MRI)

- A popular imaging modality – gives images with high level of detail
- Particularly popular for brain or spinal imaging, dynamic cardiac imaging, and angiography (images of blood vessels).
- Operates in still image – as well dynamic (video) – mode
- Operates in 2D (slice) as well as 3D (volume) mode.
- Relies on the interaction between a strong magnetic field and hydrogen atoms in the water molecules inside the human body.

Magnetic Resonance Imaging (MRI)

- The signal acquisition in an MR machine has the form of a Fourier integral:

$$\mathcal{F}[f](\mathbf{k}(t)) = \int_{R} f(\mathbf{r}) \exp\left(-j2\pi\langle \mathbf{k}(t), \mathbf{r}\rangle\right) d\mathbf{r}$$

r = spatial location in 2D or 3D
k(*t*) = frequency tuple in 2D or 3D

Fourier transform of f at frequency **k**(*t*)

- This is different from traditional pixel-level measurements in other CS systems.

- Here *t* denotes a time instant, at which the measurement is made at frequency **k**(*t*).

Magnetic Resonance Imaging (MRI)

- The software in the machine reconstructs the signal from these Fourier (often called **k**-space) measurements.

- The frequencies are specified usually along some trajectory (curve or set of curves) in the frequency plane (for 2D measurements) or frequency volume (for 3D measurements).

Representation basis

- Most MR images are sparse in some well-known basis.
- Example: MR Angiography images are sparse in the pixel domain or after spatial finite-differencing

Image sources:
http://www.cedars-sinai.edu/

Representation basis

- Example: Brain MR images are sparse in the DCT or wavelet domain.

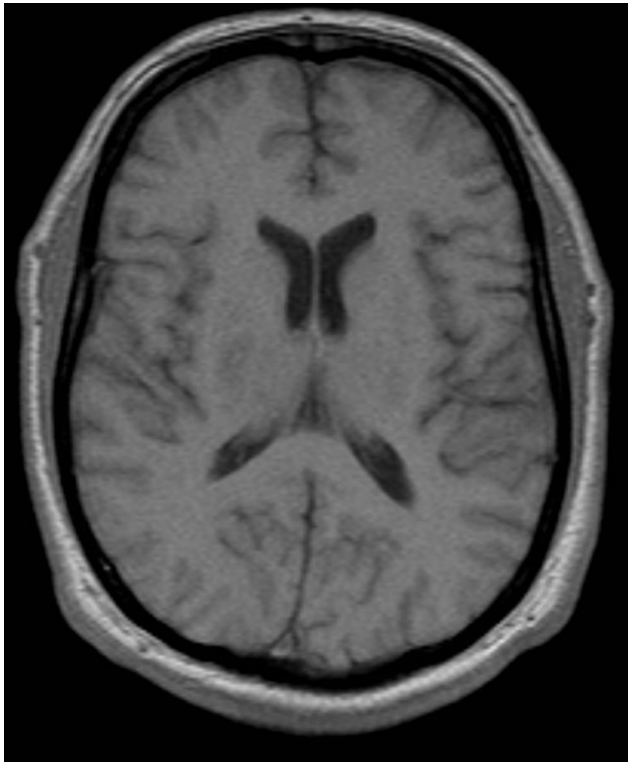


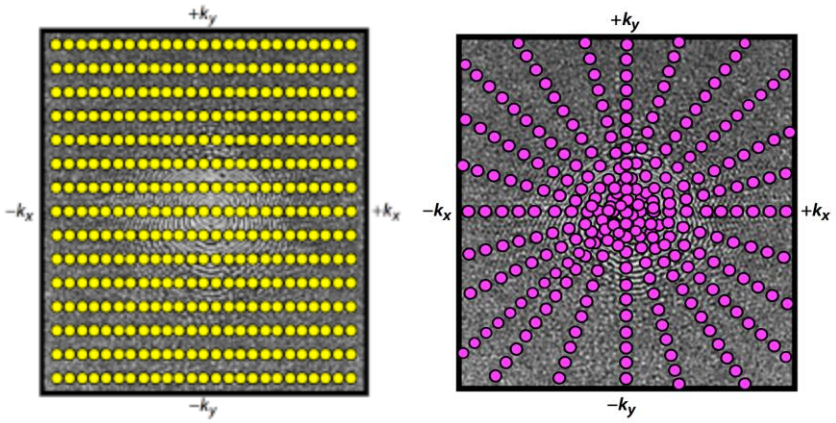
Image sources:
http://www.mridoc.com/cases/neuro/005.html

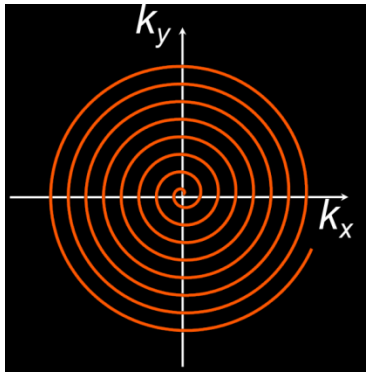
Nature of acquisitions

- A random set of Fourier measurements, i.e. a random choice of the frequency $\mathbf{k}(t)$ at every time instant, is ideal from a CS perspective.
- Why? Because the random Fourier measurement matrix is highly incoherent with the canonical basis – with high probability.
- It also obeys the RIP with high probability.

Nature of acquisitions

- However a choice of random frequencies is not practicable in actual hardware acquisitions.
- So various other types of trajectories are in use – Cartesian (left) and radial (right)





http://mriquestions.com/radial-sampling.html

https://mrrl.ucla.edu/file/59203/M229_Lecture11_FastImagingNC2.pdf

Application: MR Angiography

- In MR angiography, the aim is to observe the structure of the blood vessels (arteries/veins) after injecting a contrast agent into the blood.
- The dynamics of the structure is also very important in many applications.
- So this application requires reconstruction at high spatial and temporal resolution – a good case for application of CS.

Application: MR Angiography

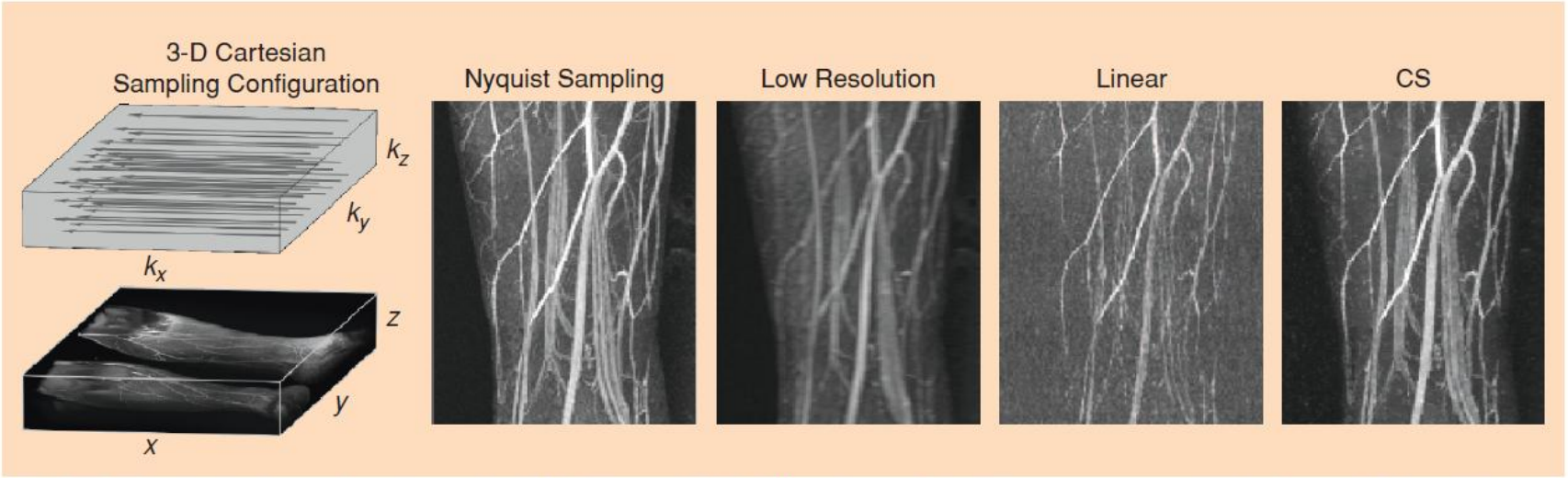
- Recall: A truly random **k**-space sampling will provide very high incoherence with the Euclidean basis, but it is not in tune with MR hardware or health-related constraints.

Application: MR Angiography

- Instead one resorts to sampling along a small number (say 10%) parallel lines in **k**-space with non-uniform gaps in between consecutive lines.
- The corresponding CS reconstruction proceeds by solving:

$$\min \|\mathbf{x}\|_1 \text{ such that } \|\mathbf{y} - \mathbf{\Phi x}\|_2^2 \leq \varepsilon$$

- CS produces better results than low resolution centric **k**-space acquisition or linear reconstruction (see next slide).



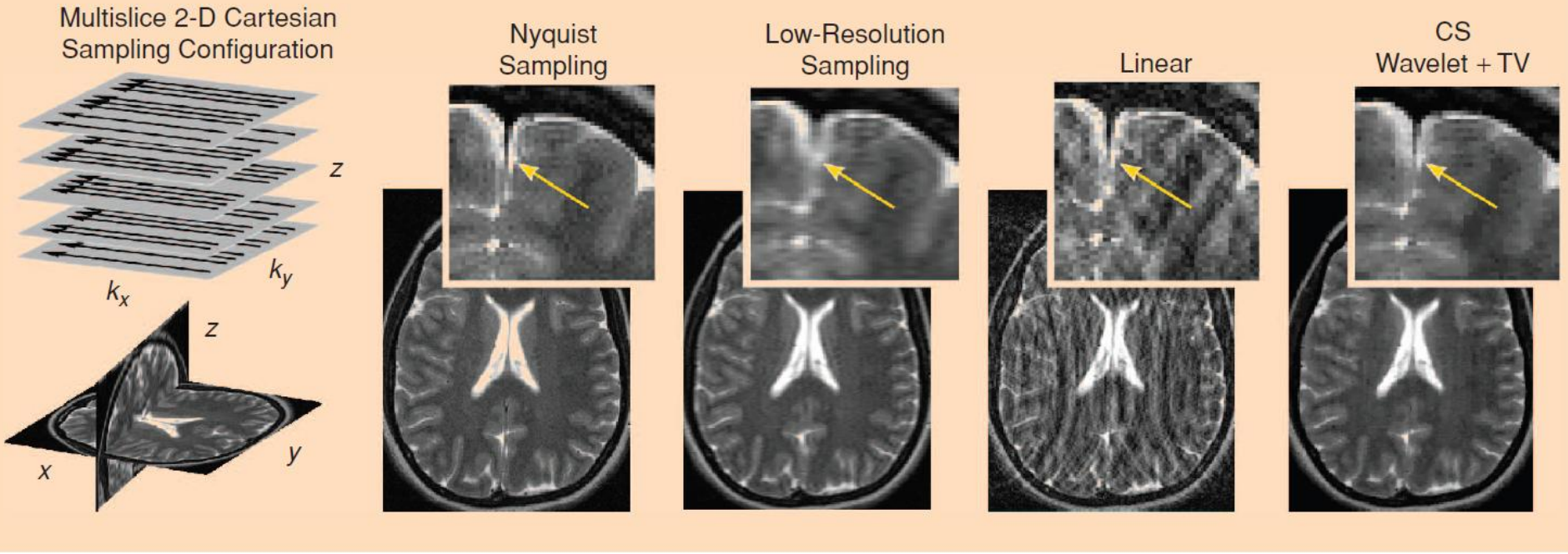
[FIG8] 3-D Contrast enhanced angiography. Right: Even with 10-fold undersampling CS can recover most blood vessel information revealed by Nyquist sampling; there is significant artifact reduction compared to linear reconstruction; and a significant resolution improvement compared to a low-resolution centric k-space acquisition. Left: The 3-D Cartesian random undersampling configuration.

CS = reconstruction using non-linear method using 10% of the k-space values
Linear = reconstruction using linear method using 10% of the k-space values
Lower resolution = reconstruction using the same number of k-space values but sampled densely from the lower-frequency region of the k-space

https://people.eecs.berkeley.edu/~mlustig/CS/CSMRI.pdf

Application: Brain MR Imaging

- Brain slices are sparse in wavelet/DCT domain.
- Sampling strategy for each slice can be spiral or random parallel lines.
- But adjacent slices can be quite similar to each other.
- Instead of independent reconstruction, one can reconstruct in a coupled manner by exploiting the fact that finite differences will be sparse in the third dimension.
- In the case of such coupled reconstruction, it is useful to use different sampling schemes in every slice.



https://people.eecs.berkeley.edu/~mlustig/CS/CSMRI.pdf

In the CS method, each 256 x 192 slice was sampled using 80 random Cartesian trajectories (compression ratio = 2.4) – different trajectories for each slice. The reconstruction quality is comparable to Nyquist-sampled measurements, but with the CS method, fewer measurements are required for the same quality. The CS method produces better quality results than linear reconstruction (i.e. using penalized pseudo-inverse). The "low-resolution sampling" method indicates a lower spatial resolution.

Compressed Sensing: Success story!

- In MRI: https://t.co/30776nzj4T

  Thanks to "compressed sensing" technology, which was developed in part at Rice, **scans of the beating heart can be completed in as few as 25 seconds** while the patient breathes freely. In contrast, in an MRI scanner equipped with conventional acceleration techniques, patients must lie still for four minutes or more and **hold their breath for as many as seven to 12 times** throughout a cardiovascular-related procedure.

- Some material from the website of Siemens: https://www.siemens-healthineers.com/en-in/magnetic-resonance-imaging/clinical-specialities/compressed-sensing