

# **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.

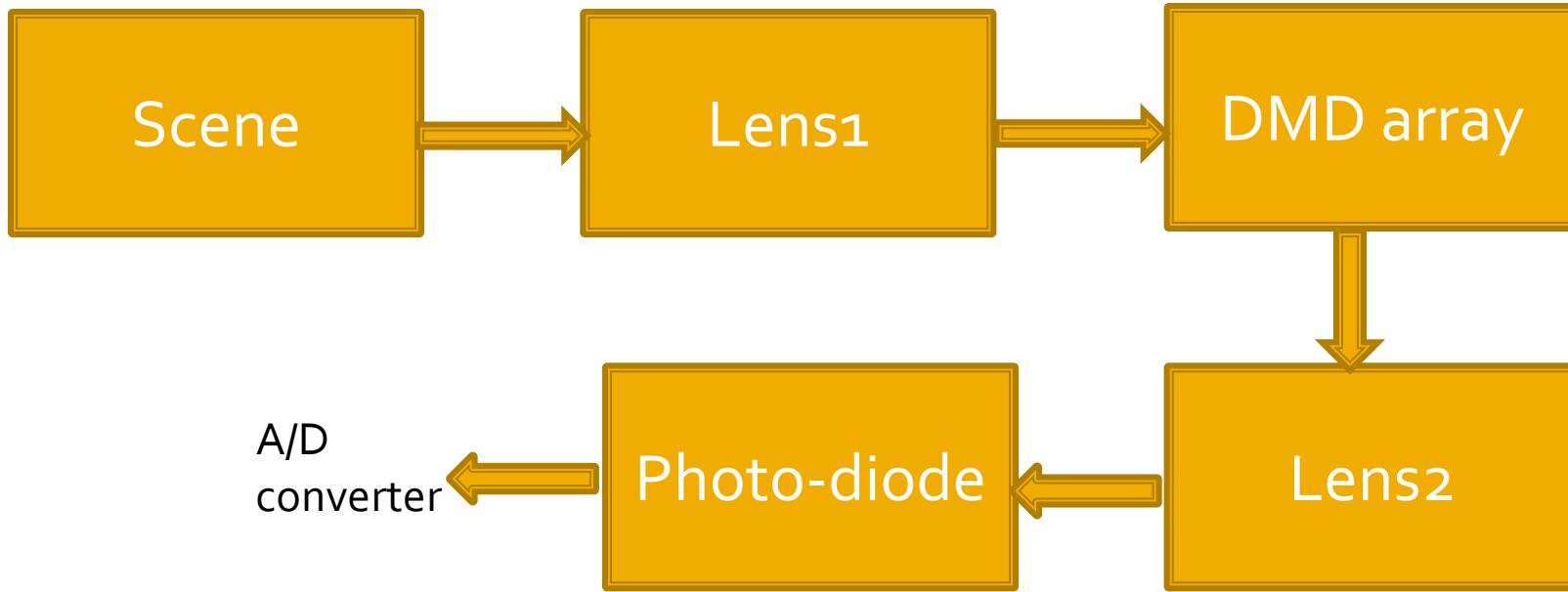
# Rice Single Pixel Camera: a CS-based camera

- 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:

$$\forall i, 1 \leq i \leq m, y_i = \mathbf{f}^T \boldsymbol{\Phi}_i$$

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

# 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  $\phi_i$ , 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  $\phi_i$  is used for each  $y_i$ ,  $1 \leq i \leq m$ .
- 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:

$$y_1 = \Phi_1 x$$

$$y_2 = \Phi_2 x$$

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

$\Phi_2$  contains a 1 wherever  $\Phi_1$  contains a 0, and  $\Phi_2$  contains a 1 wherever  $\Phi_1$  contains a 0.

# Rice Single Pixel Camera: a CS-based camera

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

$$\mathbf{y} = \Phi \mathbf{f}, \Phi = [\Phi_1 \mid \Phi_2 \mid \dots \mid \Phi_m]^T,$$

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

- 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)  $16 \times 16$ , and the dot products are computed separately for each block.
- The  $m \ll n$  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  $m \ll n$  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 ( $m \ll n$ ) than a conventional camera.

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

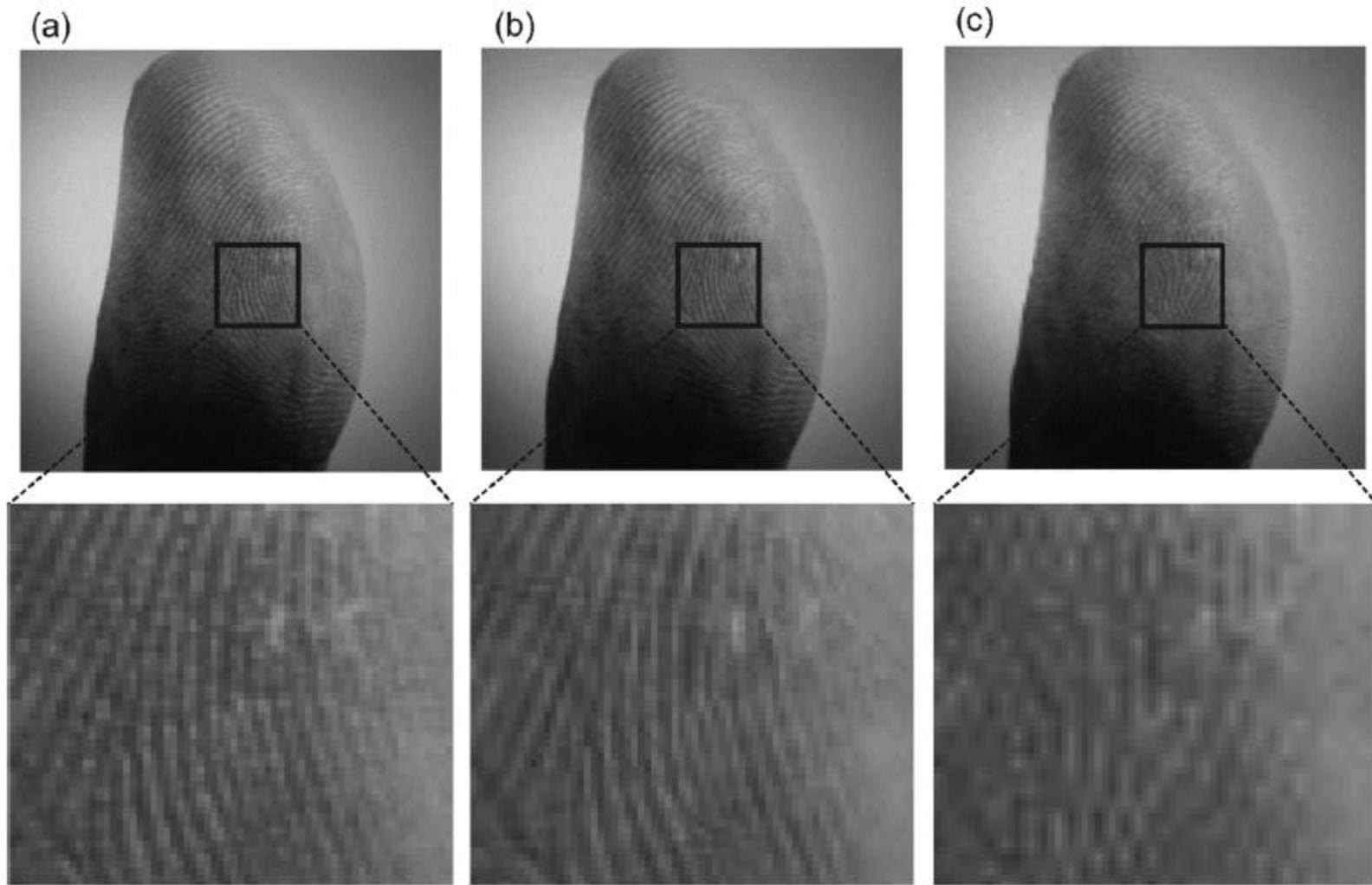


Fig. 16. Sample images captured in: (a) normal mode at 120 fps, (b) compressed sensing at  $\text{CR} = 1/4$  and 480 fps, (c) downsampling at 1/4 ratio.

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

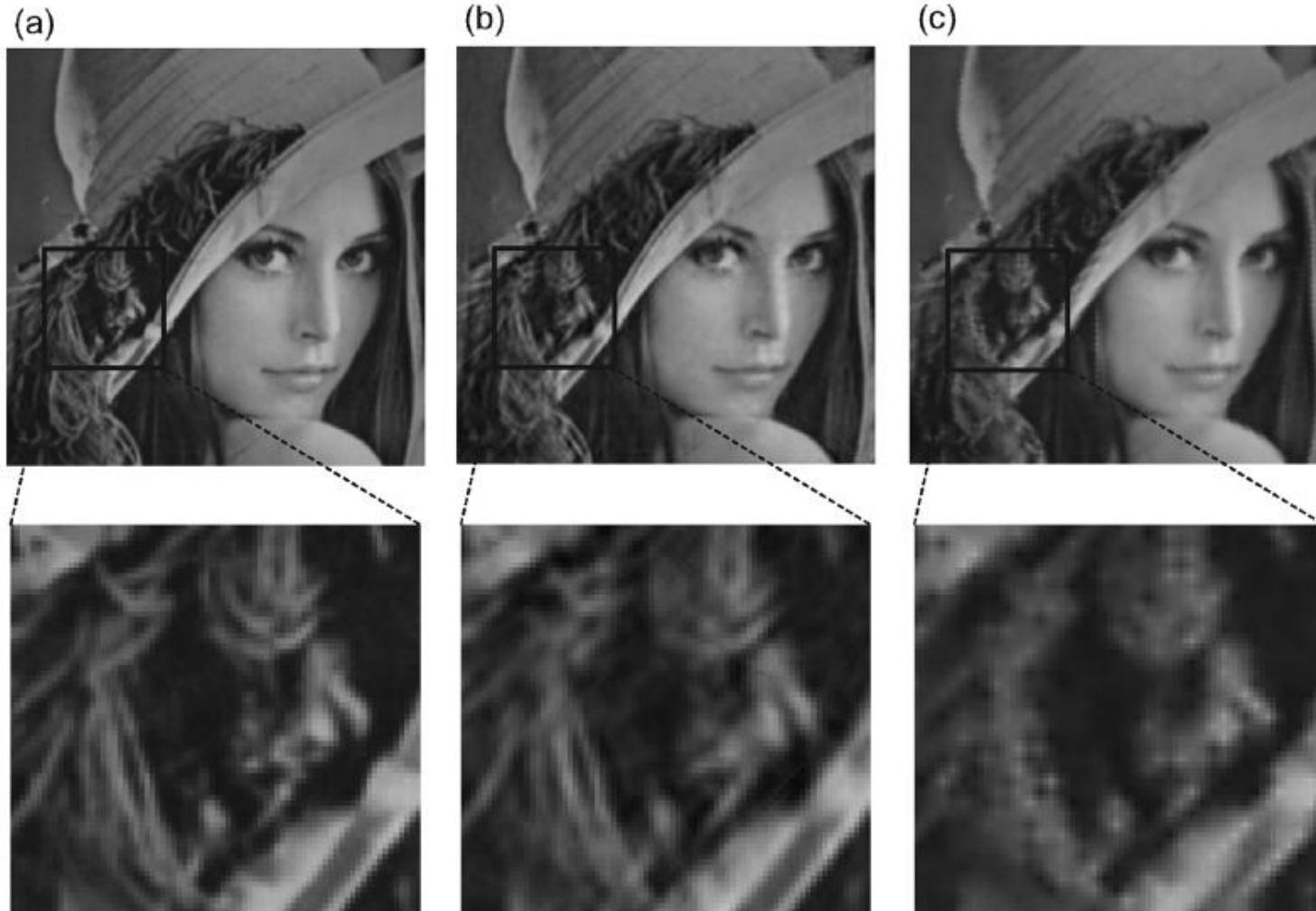


Fig. 17. Sample images captured in: (a) normal mode at 120 fps, (b) compressed sensing at  $\text{CR} = 1/8$  and 960 fps, (c) downsampling at 1/8 ratio.

<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  $\phi_i$  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  $\Psi$  and perform  $F$  independent (2D) frame-by-frame reconstructions, by solving:

$$\forall t \in \{1, \dots, F\}, \min_{\theta_t} |\theta_t|_1 \text{ such that } \mathbf{y}_t = \Phi_t \mathbf{f}_t = \Phi_t \Psi \theta_t,$$

$$\Phi_t \in R^{M/F \times n}, \Psi \in R^{n \times n}, \theta_t \in R^n, \mathbf{y}_t \in R^{M/F}$$

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

# Streaming Video Acquisition

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

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

$$\mathbf{y} = (\mathbf{y}_1 | \mathbf{y}_2 | \dots | \mathbf{y}_F), \mathbf{y}_i = \Phi_i \mathbf{f}_i$$

# Streaming Video Acquisition

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

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

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

- 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  $\Phi$  which operates on the full video, and with an associated 3D wavelet/DCT basis.

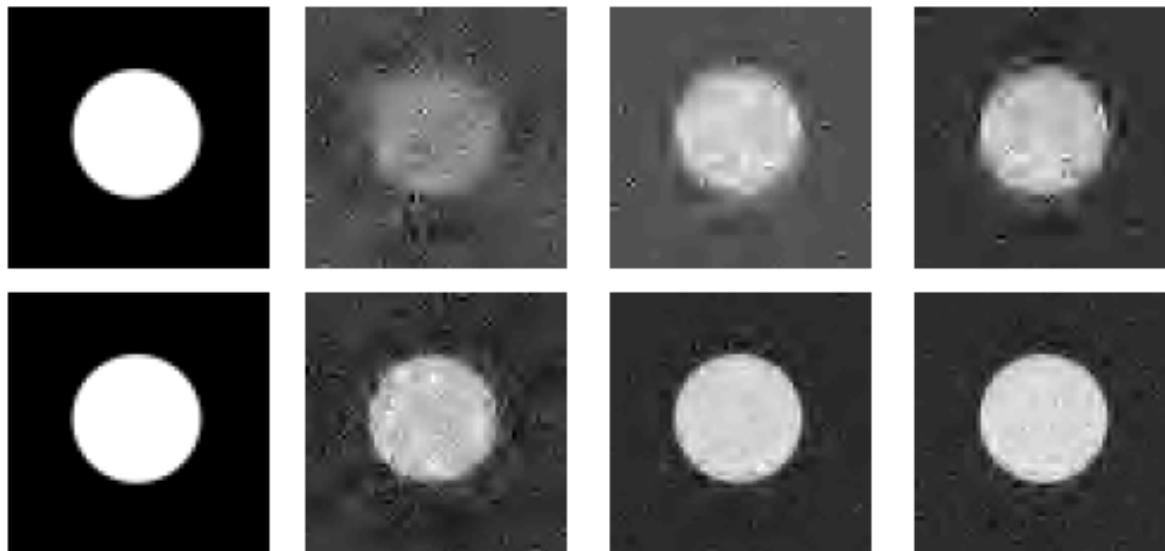
$$\min_{\theta} \|\theta\|_1 \text{ such that } \mathbf{y} = \Phi \mathbf{f} = \Phi \Psi \theta,$$

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

- Unlike method 2,  $\Phi$  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  $64 \times 64$  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.eecs.umass.edu/~mduarte/images/CSCamera\\_PCS.pdf](http://www.eecs.umass.edu/~mduarte/images/CSCamera_PCS.pdf)

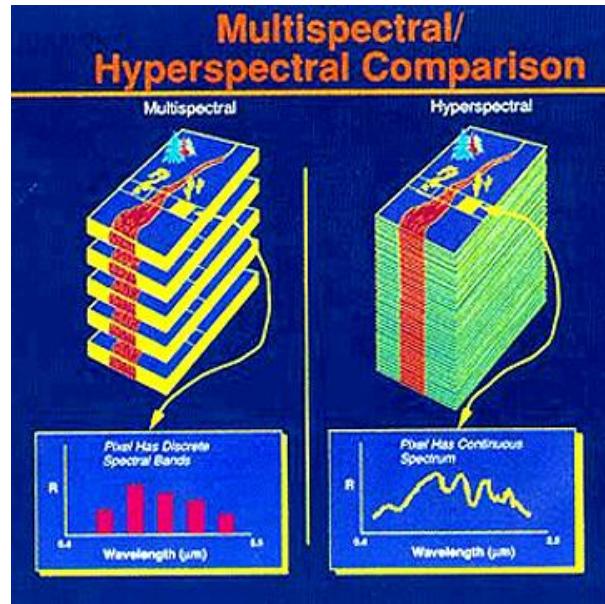
# PART 3: COMPRESSIVE ACQUISITION OF HYPERSPECTRAL DATA

# 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  $\sim 420$  nm to  $\sim 750$  nm.
- Finer division of wavelengths than possible in RGB!
- Can contain wavelengths in the infrared or ultraviolet regime.

# Sources of confusion 😊

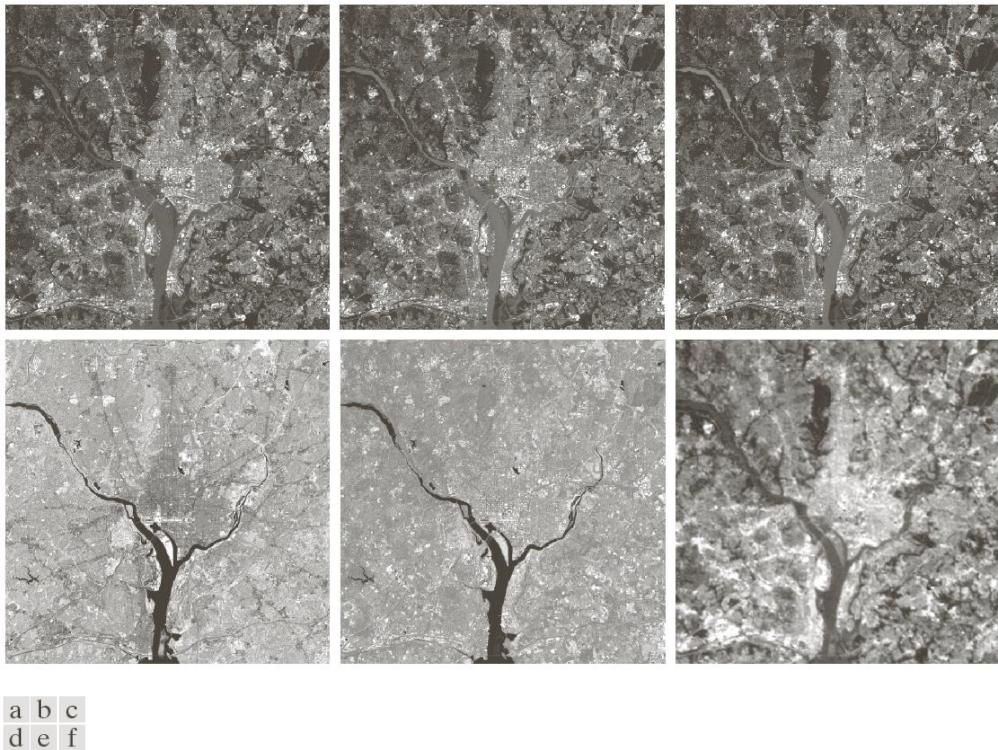
- Hyperspectral images are abbreviated as HSI!
- Hyperspectral images are different from multispectral images. The latter contain few, discrete and discontinuous wavelengths. The former contain many more wavelengths with continuity.



# Beyond color: Hyperspectral images

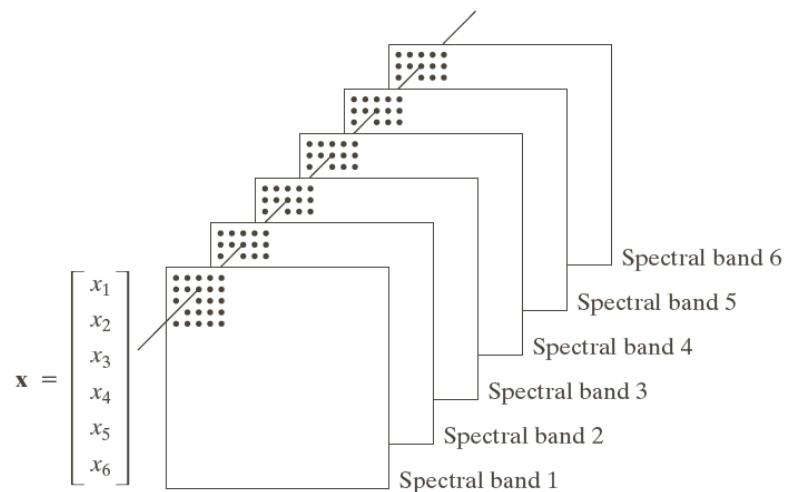
- Widely used in remote sensing (satellite images) – often different materials/geographical entities (soil, water, vegetation, concrete, landmines, mountains, etc.) can be detected/classified by spectral properties.
- Also used in chemistry, pharmaceutical industry and pathology for classification of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

## Example multispectral image with 6 bands



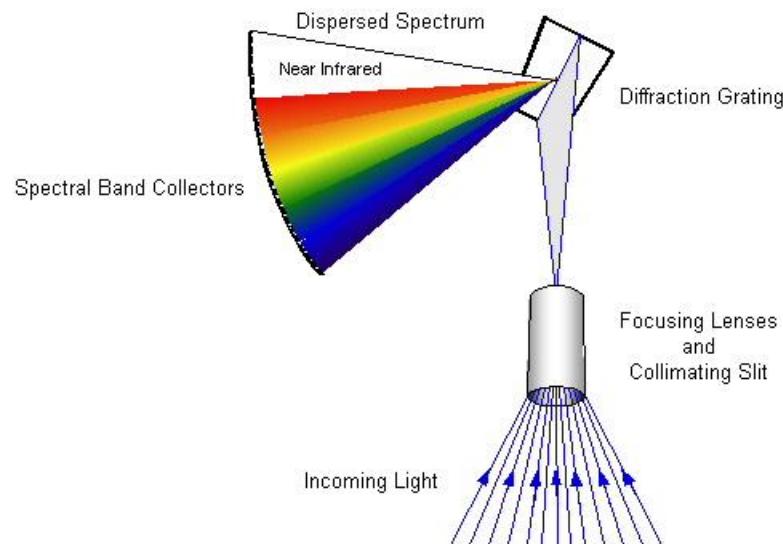
**FIGURE 11.38** Multispectral images in the (a) visible blue, (b) visible green, (c) visible red, (d) near infrared, (e) middle infrared, and (f) thermal infrared bands. (Images courtesy of NASA.)

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



# Conventional Hyperspectral Camera

Multiple sensor arrays  
– one per wavelength.  
Expensive!

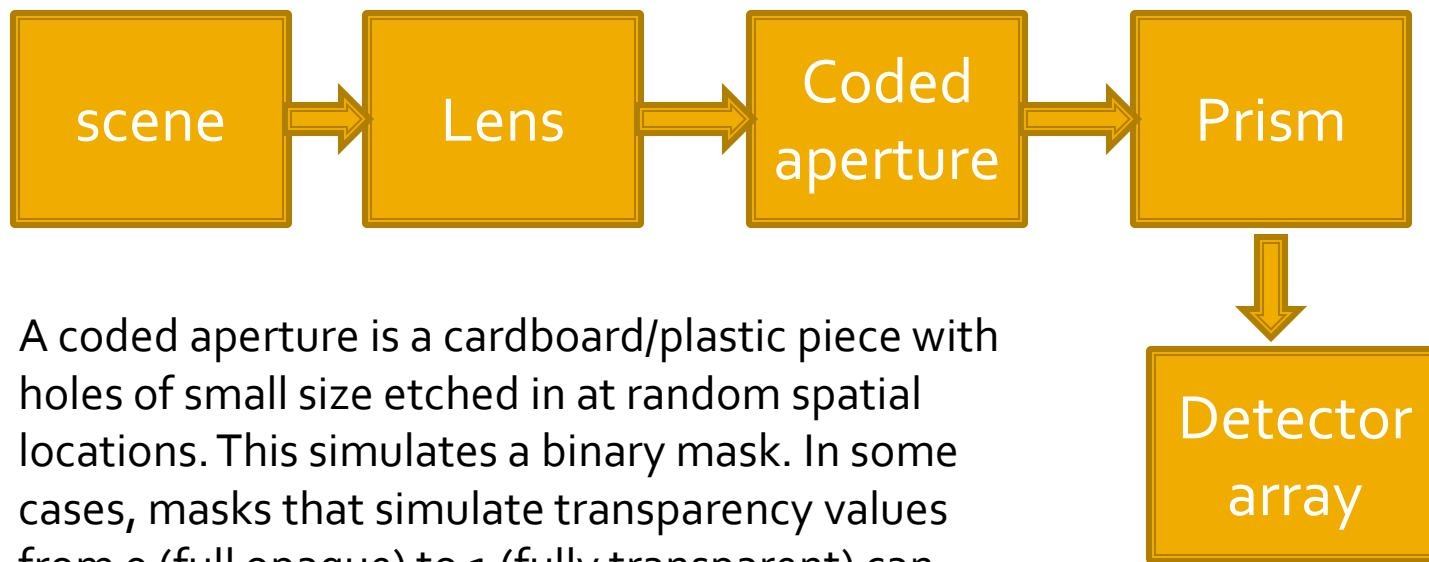


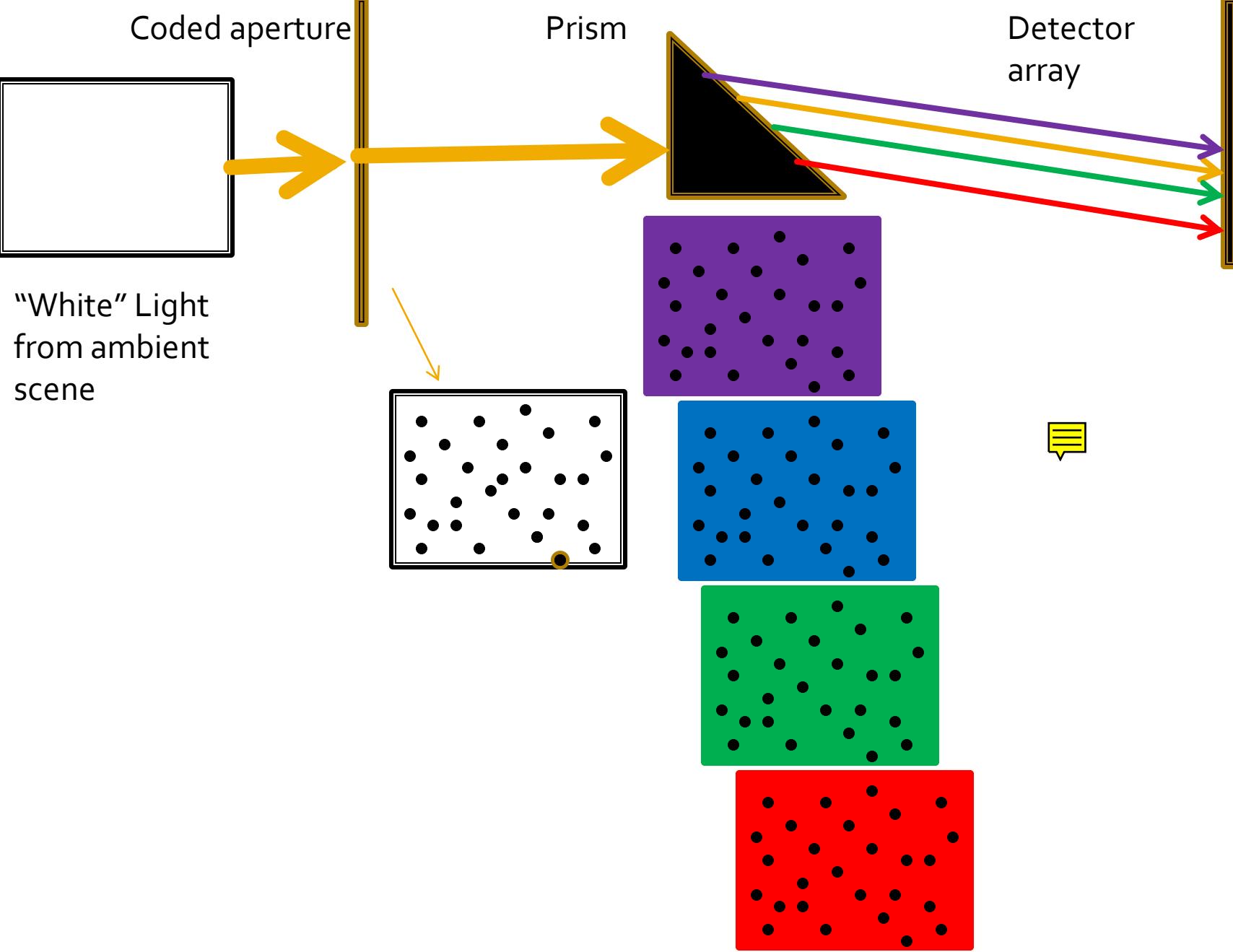
# CASSI: Compressive Hyperspectral Image Acquisition

- Reconstruction of hyperspectral data imaged by a coded aperture snapshot spectral imager (CASSI) developed at the DISP (Digital 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.





# 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  $N_\lambda$ .
- 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

$$\hat{\mathbf{X}}_j = \mathbf{X}_j \bullet \mathbf{C}.$$

# 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  $\mathbf{X}$ .
- Also the portions of the coded aperture contributing towards a single pixel value  $M(x, y)$  are different for different wavelengths.

# Multi-frame CASSI

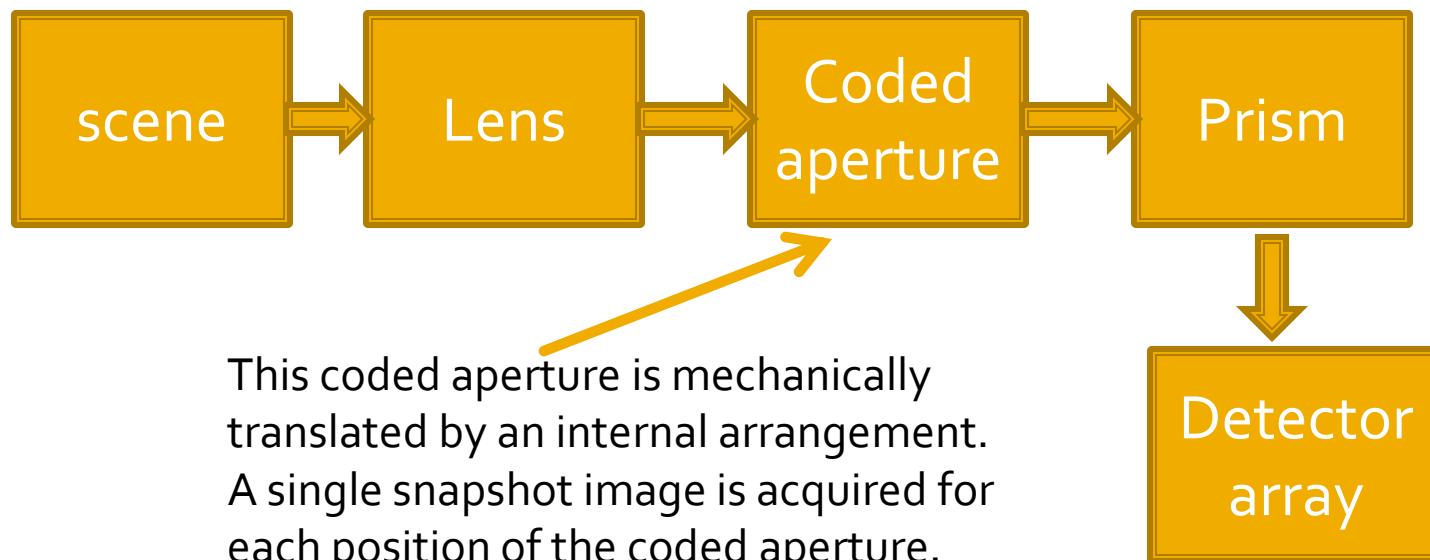
- The compression rate of CASSI is the number of wavelengths: 1.
- This compression rate can be reduced if  $T > 1$  snapshots of the same scene are acquired in quick succession, denoted as  $\{\mathbf{M}_t\}_{t=1}^T$  reducing the compression rate to  $N_\lambda : T$ .
- Each snapshot is acquired using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

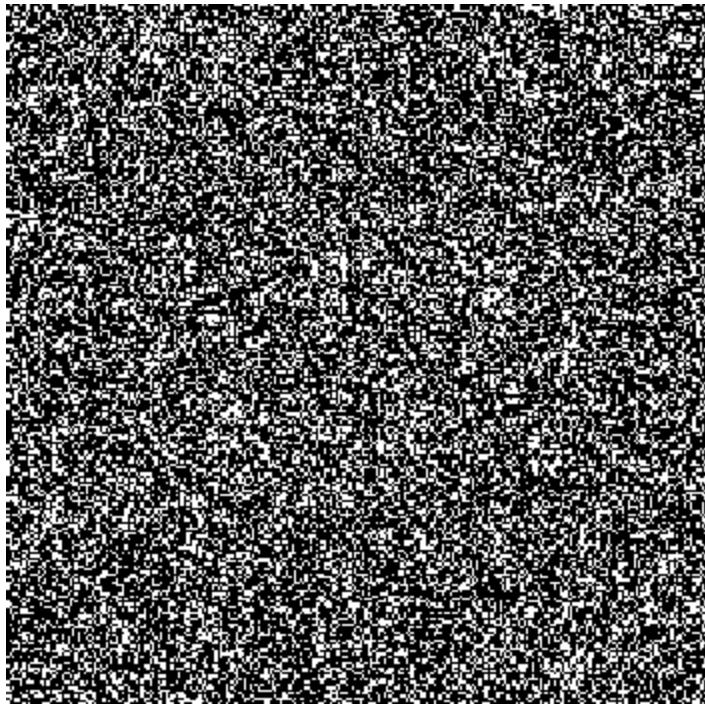
# Multi-frame CASSI

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

For  $t = 1$  to  $T$ ,

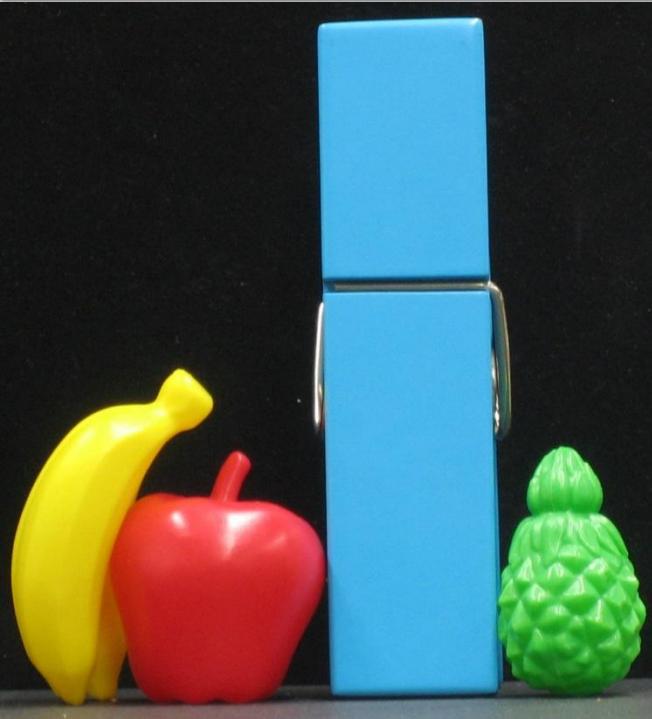
$$M_t(x, y) = \sum_{j=1}^{N_\lambda} S_{j,t}(x, y) = \sum_{j=1}^{N_\lambda} \hat{X}_{j,t}(x - l_j, y) = \sum_{j=1}^{N_\lambda} X_j(x - l_j, y) \bullet C_t(x - l_j, y)$$





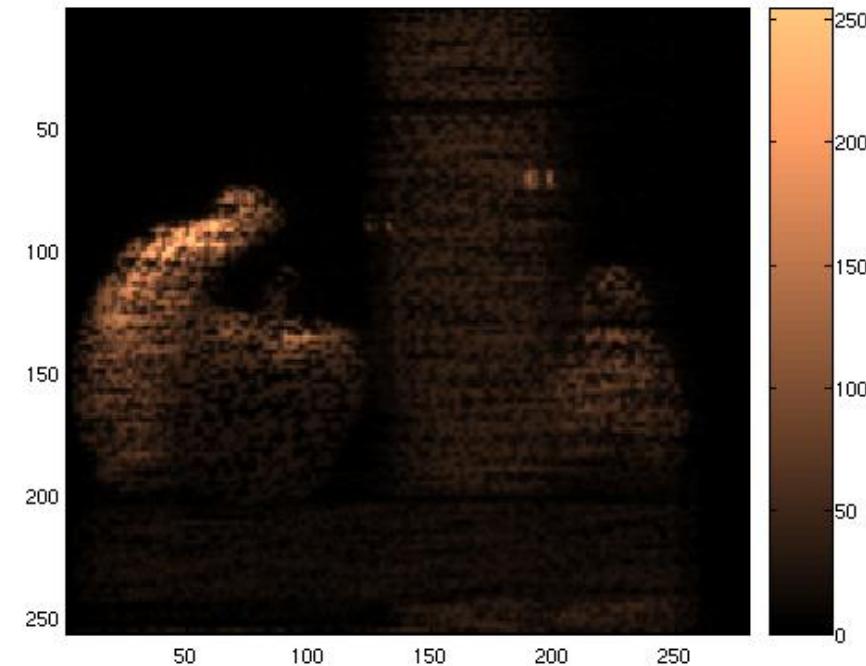
- Aperture Code: created randomly (random binary, [0,1] uniform random also possible)
- The aperture code pattern has holes of size  $2 \times 2$  pixels (smaller holes give rise to diffraction artifacts).
- The pattern is projected onto the detector array in a magnified form.
- Note: Random mask pattern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

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



Snapshot spectral image  
acquired by CASSI camera

# 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:

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

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

# Reconstruction Method

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



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)

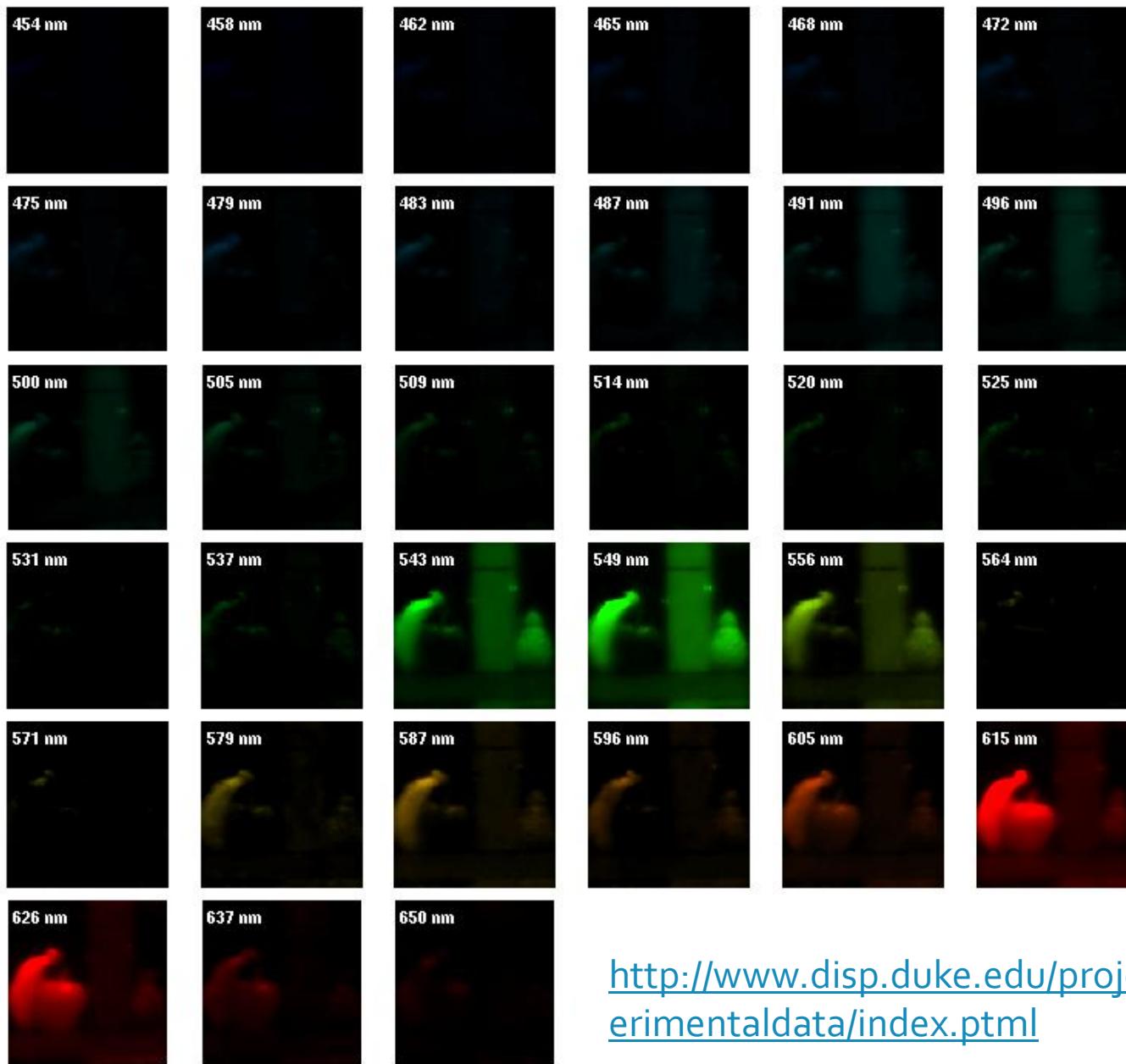
$$\Phi_t = \begin{pmatrix} \text{diag}(C_{1,t}) & \text{diag}(C_{2,t}) & \dots & \text{diag}(C_{N_\lambda,t}) \end{pmatrix} \rightarrow \text{size } N_x N_y \times N_x N_y N_\lambda$$

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

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

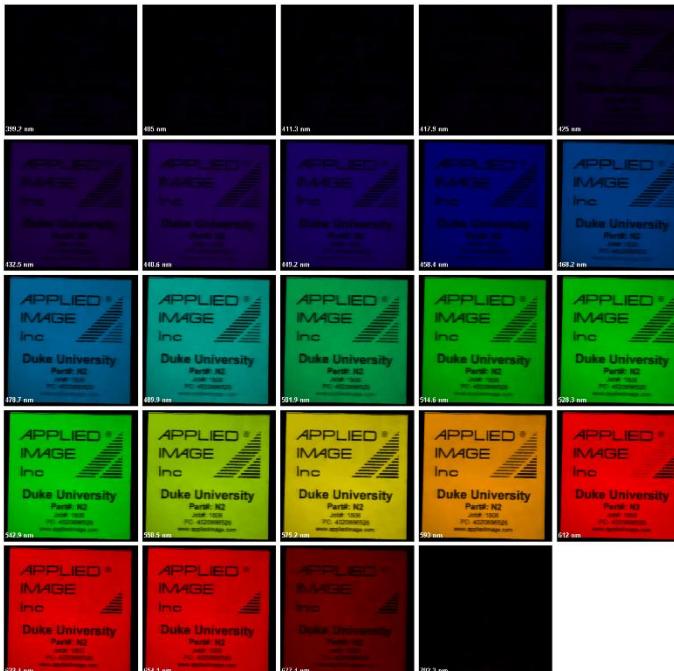
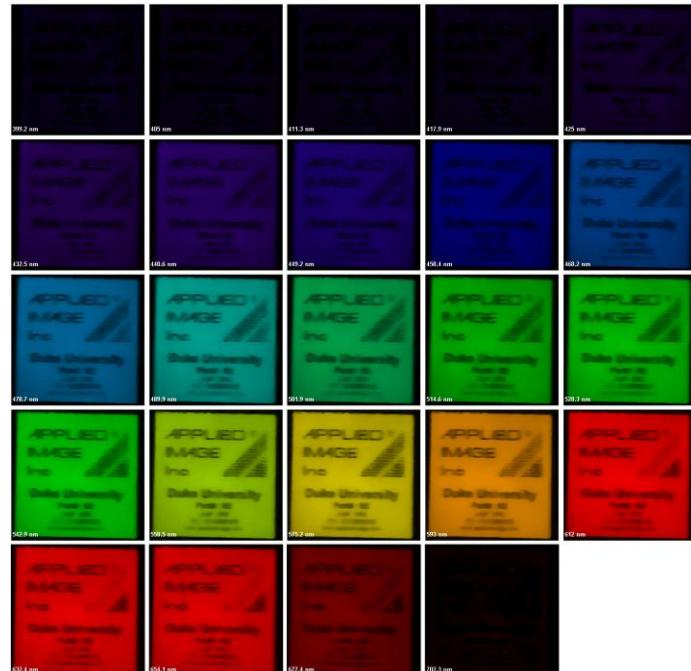
$\mathbf{m}_t$  = vectorized form of snapshot image  $\rightarrow$  size  $N_x N_y \times 1$

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.



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

# Single versus multi-frame



[http://www.disp.duke.edu/projects/Multi\\_CASSI/index.html](http://www.disp.duke.edu/projects/Multi_CASSI/index.html)

# 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)*

# Importance of coded aperture

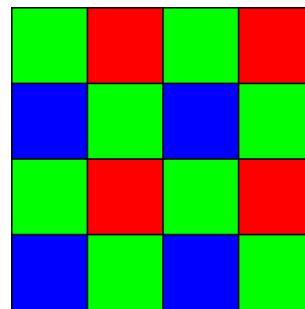
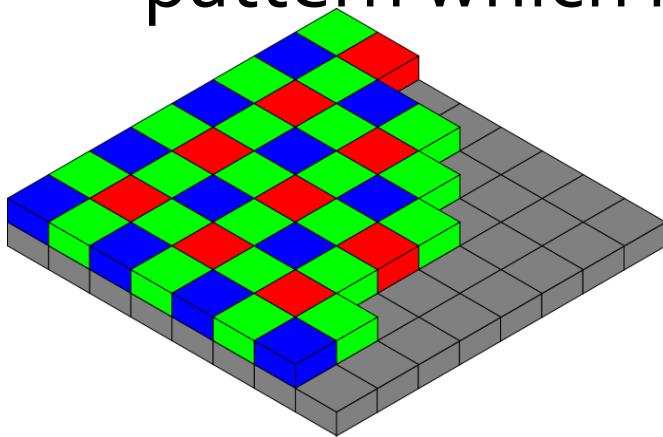
- The coded aperture allows for lower coherence values between the sensing matrix  $\Phi$  and the orthonormal basis  $\Psi$ .
- No coded aperture =  $\forall t, x, y C_t(x, y) = 1$ .
- No coded aperture = multi-frame CASSI not possible.

# 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:

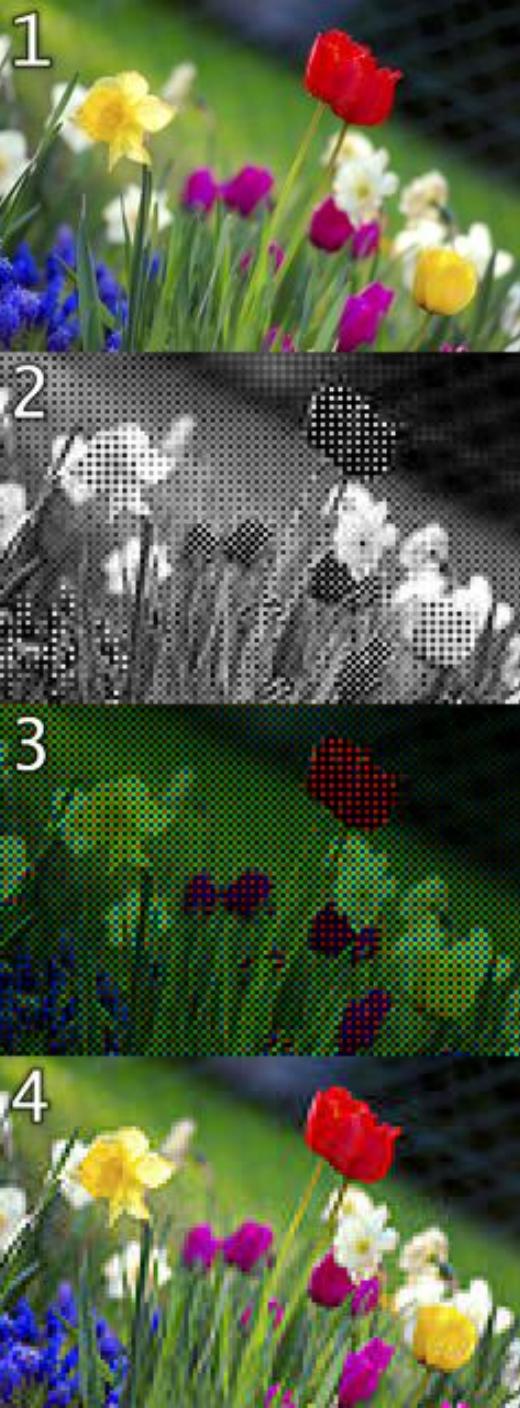


[https://en.wikipedia.org/wiki/Color\\_filter\\_array](https://en.wikipedia.org/wiki/Color_filter_array)

- 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](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

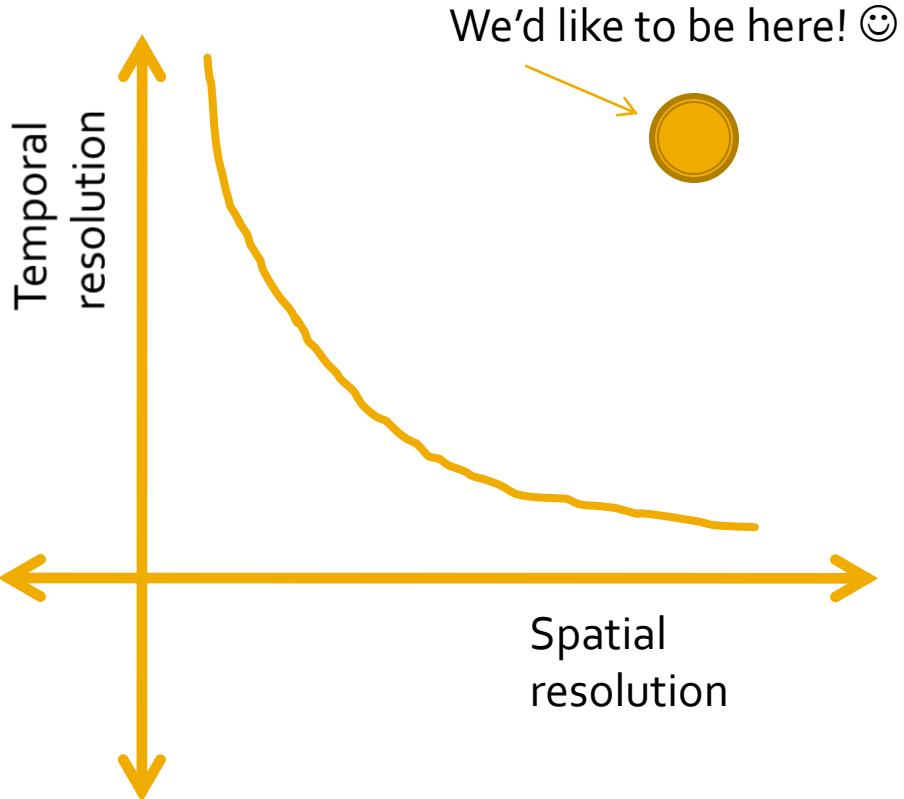
# **Video from a Single Coded Exposure Photograph using a Learned Over-Complete Dictionary**

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

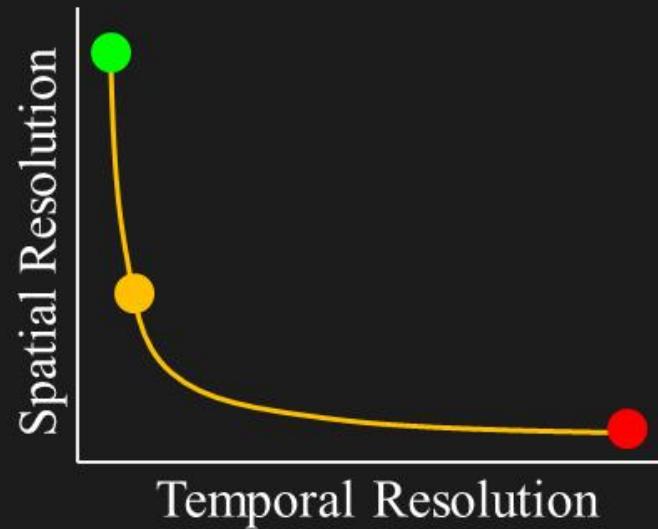
# Spatio-Temporal Resolution Tradeoff



Temporal Resolution: 1X



Temporal Resolution: 4X



Temporal Resolution



Temporal Resolution: 36X

# Coded Exposure Image

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

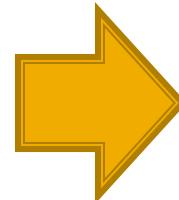
Coded exposure image  
(captured in one unit  
integration time of the  
camera)

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

Binary code at  
time instant  $t$

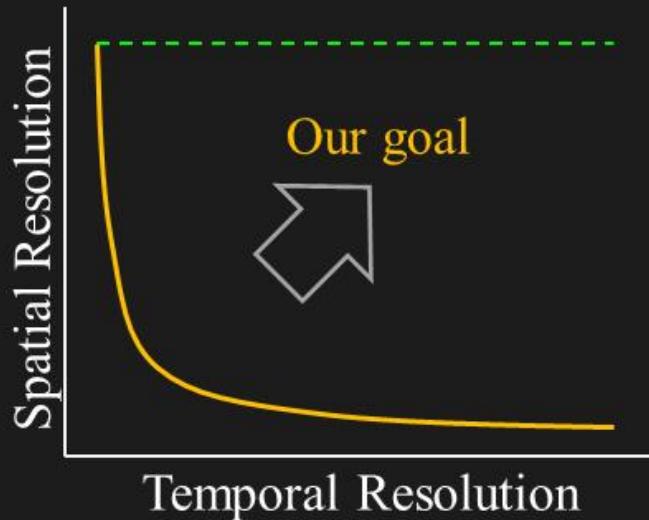
Sub-frame at  
time instant  $t$

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



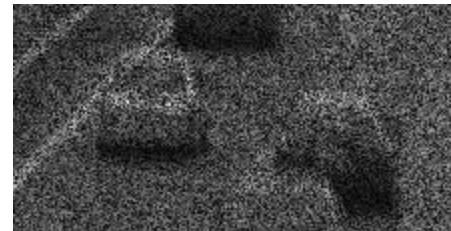
Coded Single Image



[http://www.cs.columbia.edu/CAVE/projects/single\\_shot\\_video/](http://www.cs.columbia.edu/CAVE/projects/single_shot_video/)



$T=3$



# 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

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

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

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

$\mathbf{y}$  is a vectorized form of the snapshot image  $\mathbf{I}$  of  $n$  pixels.

$\mathbf{f}$  is a vectorized form of the underlying video with  $T$  frames and each frame having  $n$  pixels, hence  $\mathbf{f}$  has  $nT$  pixels in total.

$\Psi$  is a 3D DCT basis – the basis can also be “learned offline on a representative set of videos”

$\Phi$  is a matrix of size  $n \times nT$  containing values from the binary code  $\mathbf{S}$ . See below:

$$\Phi = (\text{diag}(S_1) | \text{diag}(S_2) | \dots | \text{diag}(S_T))$$

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

$\text{Diag}(S_t)$  represents a  $n \times n$  diagonal matrix. The diagonal contains elements from the mask  $S_t$ .

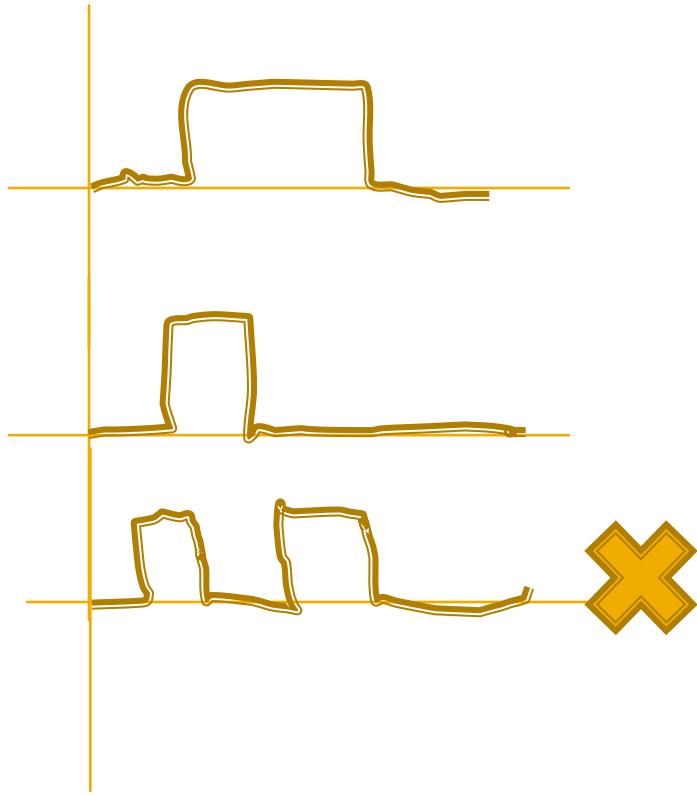
# Code Design: S

(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  $M \times M$  spatial patch should **cover full integration time**

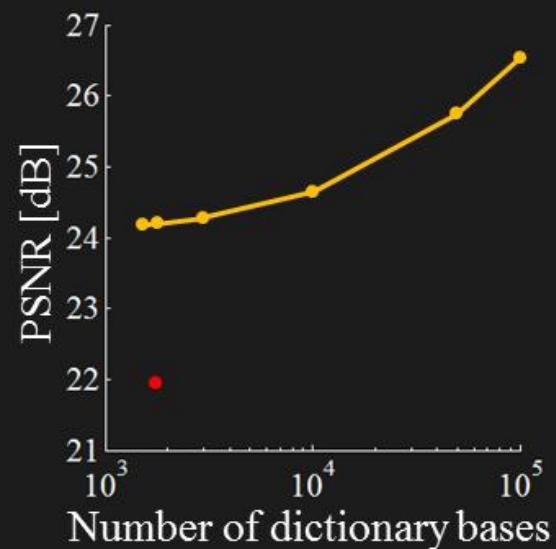
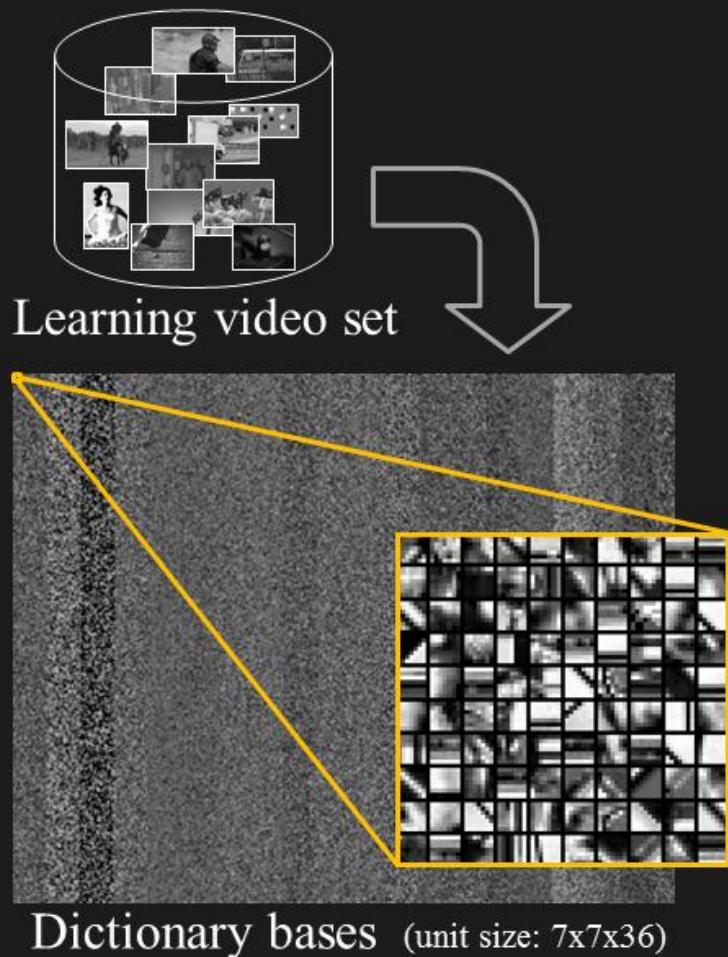


# Dictionary Learning

- 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  $7 \times 7 \times 36 = 1764 \times 1$
- 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  $n \times N$ , it learns a set of vectors ( $n \times K$ ), 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).

# Learned Over-complete Dictionary



- Learned over-complete dictionary
- DCT based dictionary

[http://www.cs.columbia.edu/CAVE/projects/single\\_shot\\_video/](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>

# 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:

$$F(t) = \int_R f(\mathbf{r}) \exp(-j2\pi \langle \mathbf{k}(t), \mathbf{r} \rangle) d\mathbf{r}$$

- 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  $\mathbf{k}(t)$ .

# 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.

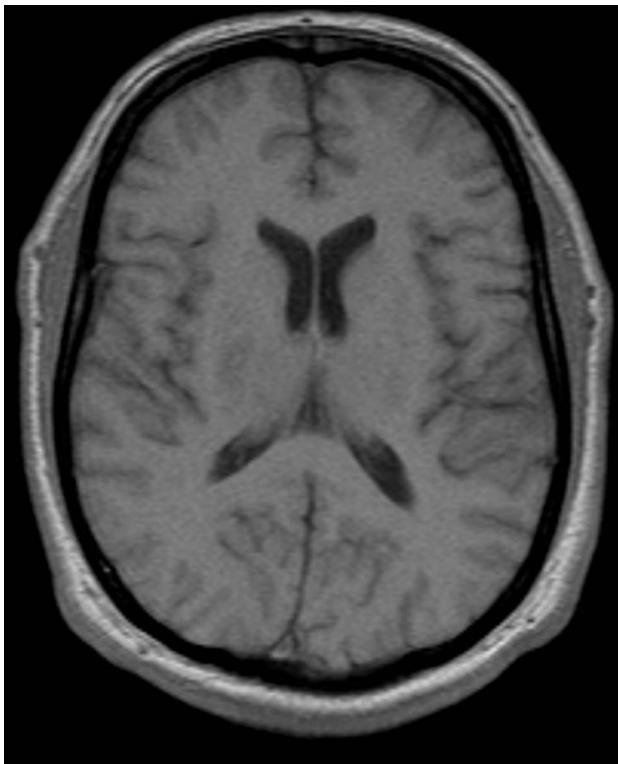


Image sources:

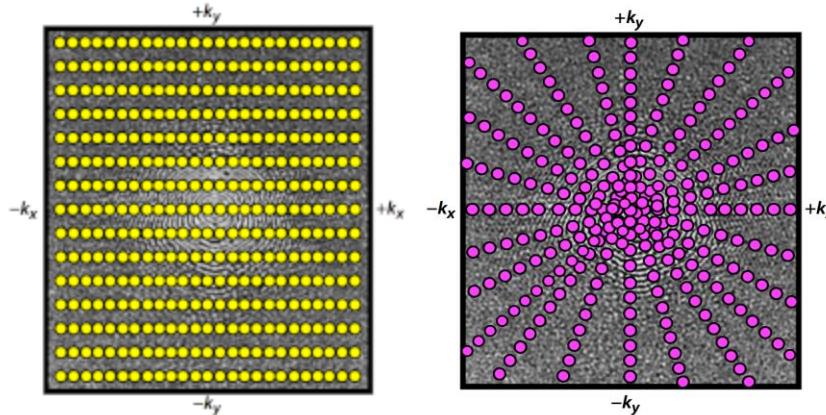
<http://www.mridoc.com/cases/neuro/005.html>

# Nature of acquisitions

- A random set of Fourier measurements, i.e. a random choice of the frequency  $k(t)$  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)



# Application: MR Angiography

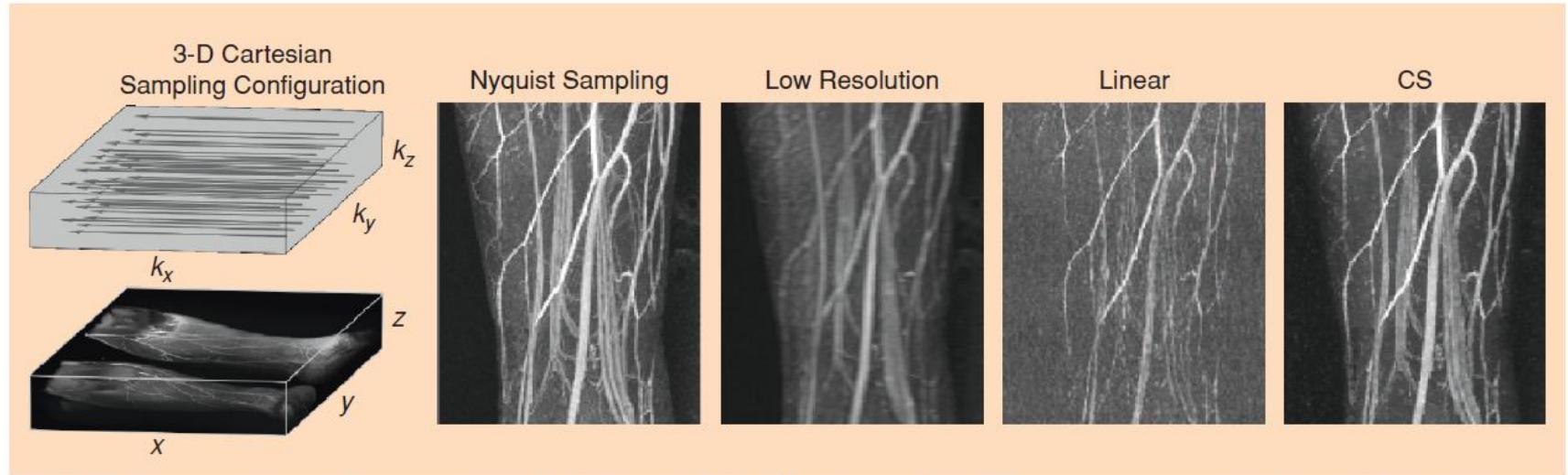
- In MR angiography, the aim is to observe the structure of the blood vessels (arteries/veins) after injecting a contrast agent into the blood.
- The dynamics of the structure is also very important in many applications.
- So this application requires reconstruction at high spatial and temporal resolution – 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  $\mathbf{k}$ -space with non-uniform gaps in between consecutive lines.
- The corresponding CS reconstruction proceeds by solving:  
$$\min \|\mathbf{x}\|_1 \text{ such that } \|\mathbf{y} - \Phi \mathbf{x}\|_2^2 \leq \varepsilon$$
- CS produces better results than low resolution centric  $\mathbf{k}$ -space acquisition (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

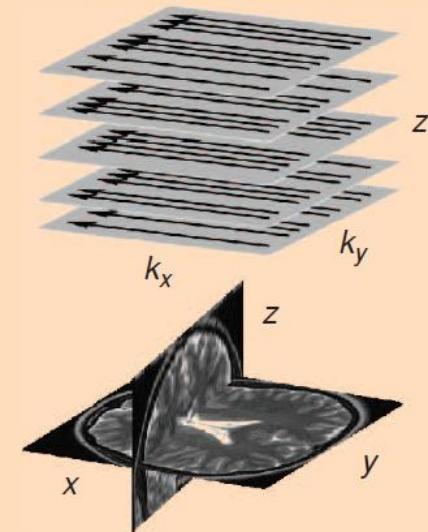
Lower resolution = reconstruction using the same number of k-space values but sampled densely from the lower-frequency region of the k-space

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

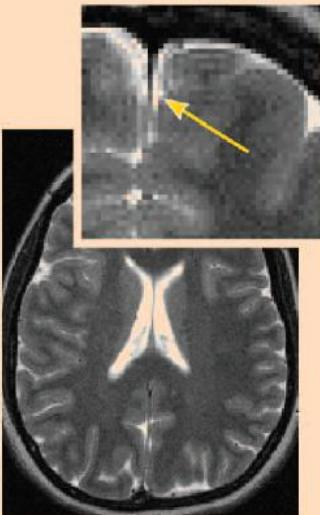
# Application: Brain MR Imaging

- 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.

Multislice 2-D Cartesian Sampling Configuration



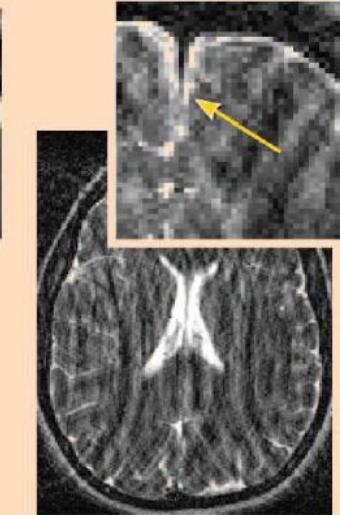
Nyquist Sampling



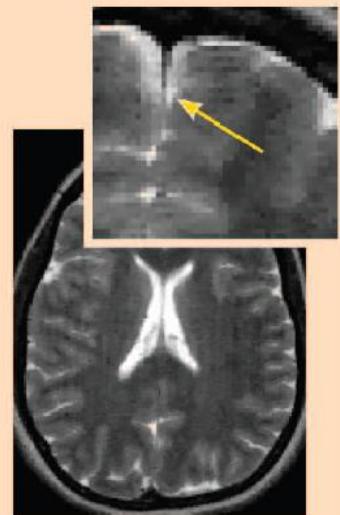
Low-Resolution Sampling



Linear



CS  
Wavelet + TV



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