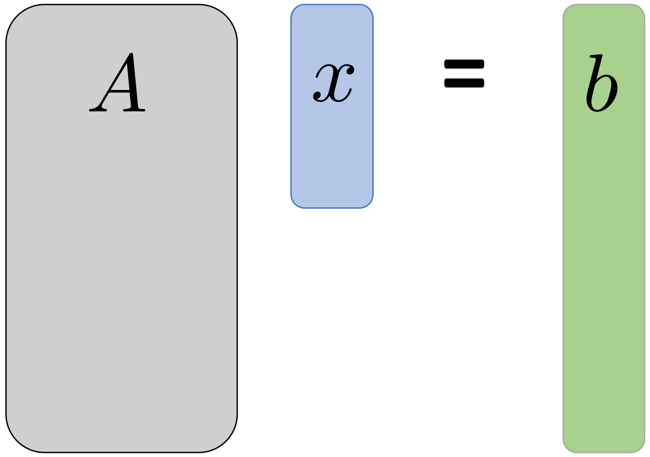
EECS 16A

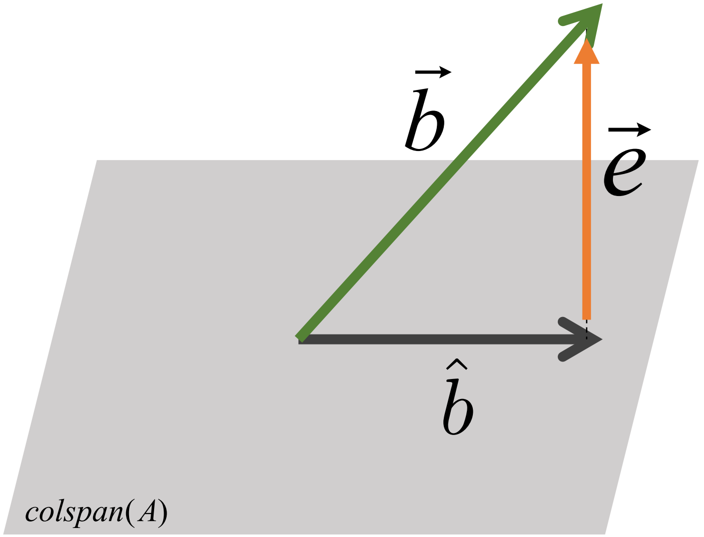
fun stuff: computational imaging

EECS16A: course evaluations

- course-evaluations.berkeley.edu

*Over*determined system: use least squares

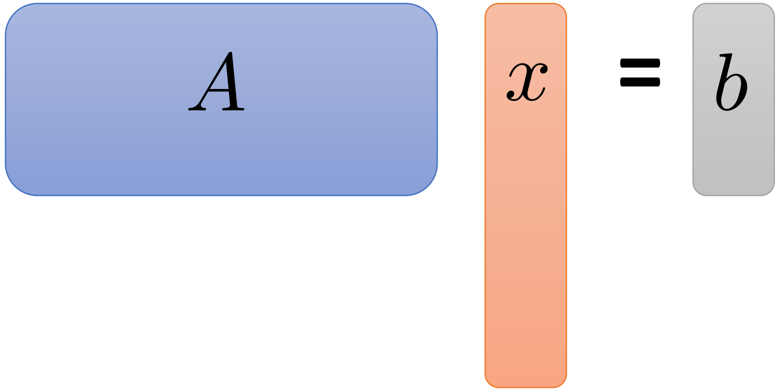


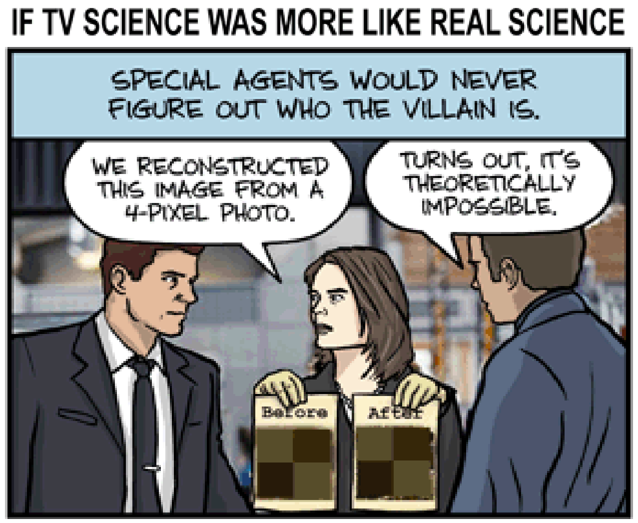


- the least-squares solution “minimally perturbs” b

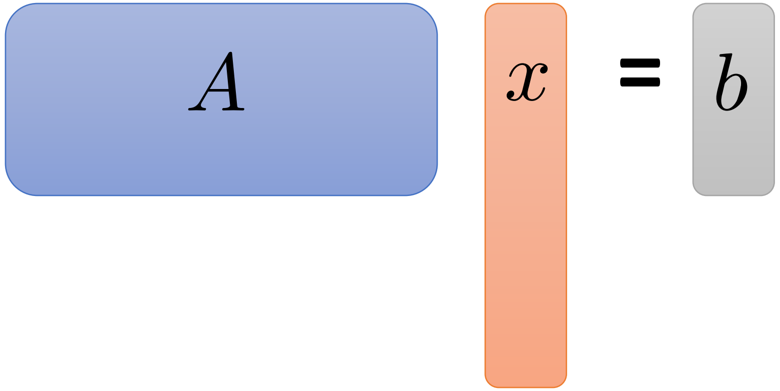
$$\hat{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \vec{b}$$

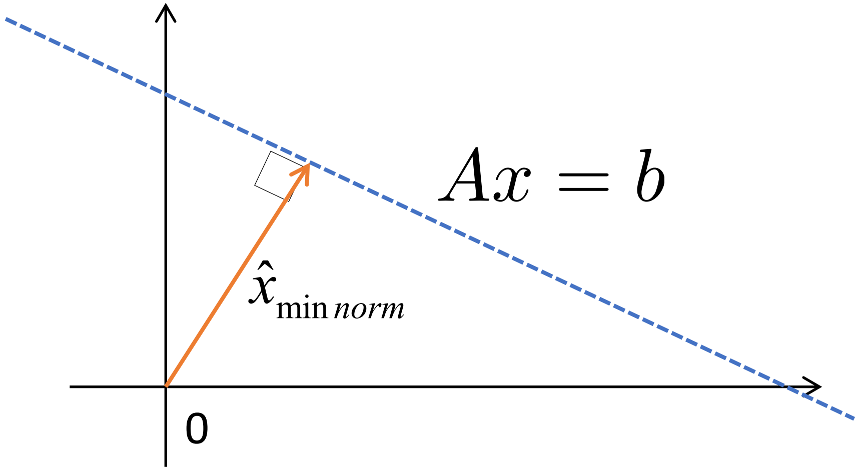
*Under*determined system: ????





*<u>Under</u>*determined system: ????

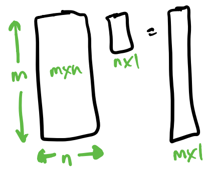




- Can be infinite valid solutions!
- Ideas: pick the ‘smallest’ one? The ‘sparsest’?
    - e.g. min norm:

$$\hat{x}_{\min norm} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\vec{b}$$

So far, we've only talked about 'overdetermined systems':



$m > n$ "overdetermined"

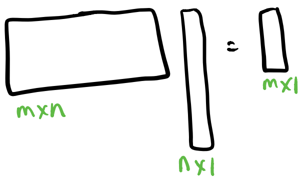
$A\vec{x} = \vec{b}$

Recall: for least squares sol'n we minimized error:

$\min \|\vec{e}\|^2 = \min_{\vec{x}} \|A\vec{x} - \vec{b}\|^2$

↳ $\hat{\vec{x}} = (A^TA)^{-1}A^T\vec{b}$

What if problem is "underdetermined"?



$m < n$

Infinite sol'ns (more unknowns than eqns)
↳ (though 0 sol'ns if inconsistent)

But, can we find a good estimate?
↳ Yes, in special cases!
e.g. sparsity (most entries are zero)

Example: We could look for 'smallest' → min norm solution: $\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \end{bmatrix} \rightarrow x_1 + x_2 = 1$
smallest sol'n:
$x_1 = 0.5,\ x_2 = 0.5$

$\min_{\vec{x}} \|\vec{x}\|^2$ such that $A\vec{x} = \vec{b}$ (constrained optimization) $\longrightarrow \hat{\vec{x}} = A^T(AA^T)^{-1}\vec{b}$

scalars (Lagrange multiplier)

to understand this, take EE127

$$\min_{\vec{x},\lambda} \|\vec{x}\|^2 + \vec{\lambda}^T(\vec{b} - A\vec{x})$$

take $\frac{\partial}{\partial x}\left(\vec{x}^T\vec{x} + \vec{\lambda}^T(\vec{b} - A\vec{x})\right) = 0$

$$A\,2\vec{x}^T - A\vec{\lambda}^T A = A\,0$$

$$\vec{\lambda} = (AA^T)^{-1}\,2A\vec{x}$$

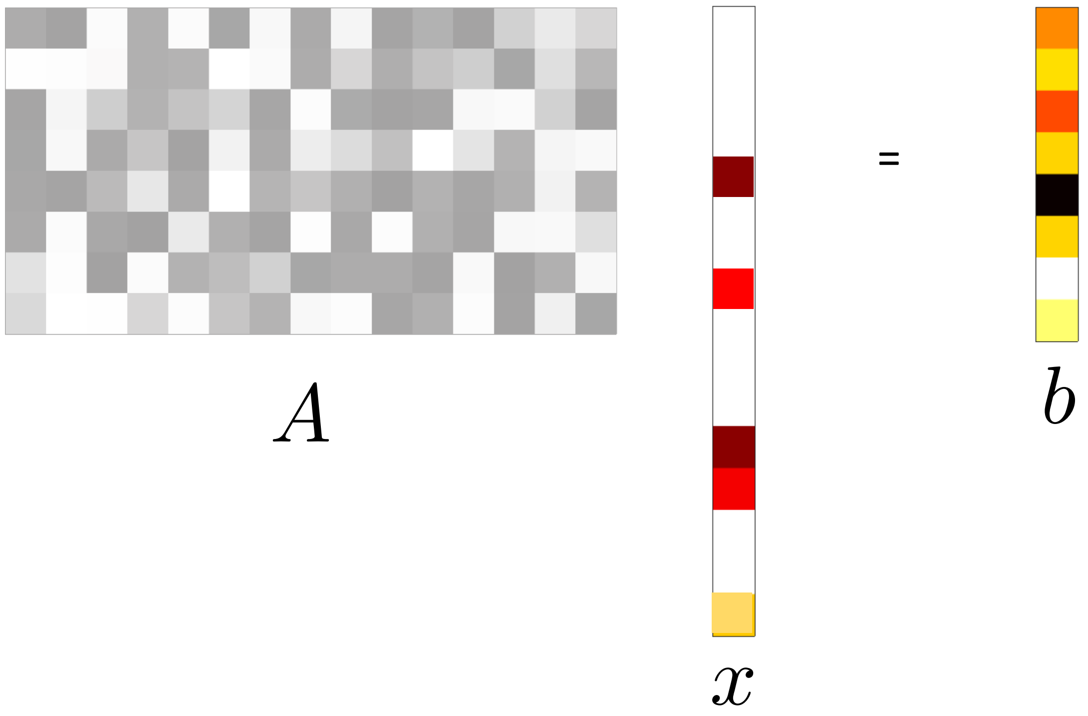
take deriv. : $A\vec{x} - \vec{b}$

$$\vec{\lambda} = (AA^T)^{-1}\,2\vec{b}$$

$$\boxed{\vec{x} = A^T(AA^T)^{-1}\vec{b}}$$

Minimum norm sol'n!

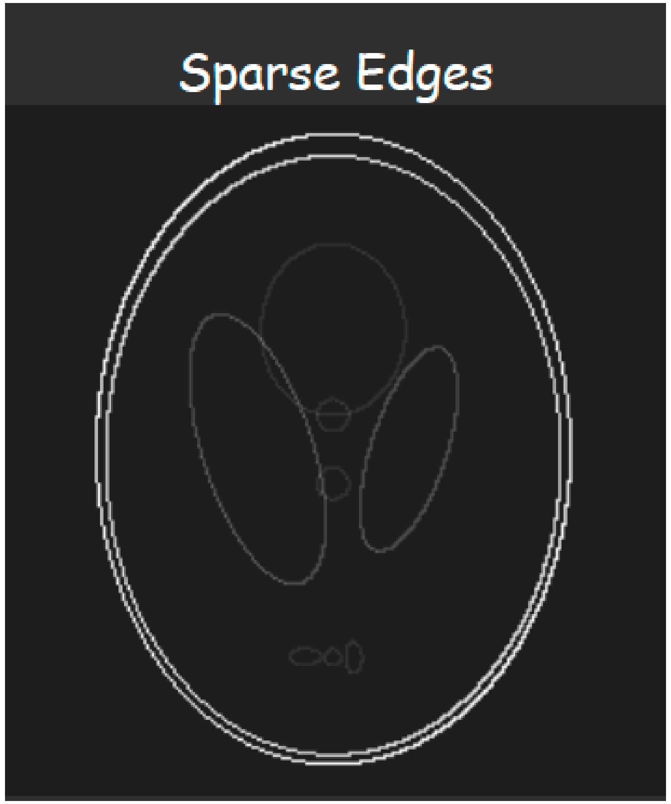
‘Sparsity’ tells us how ‘dense’ the solution is



The fraction of non-zero elements in a matrix is called the ***sparsity*** **(e.g. here k=5 for x vector)**

Sometimes not-sparse things are sparse in a different basis

Take |derivatives|



How can we use sparsity for good?

Example: image compression

Reduce memory by smartly choosing which information to throw away

No compression | 23:1 compression | 144:1 compression

sparsity

Image compression:

not sparse

sparse

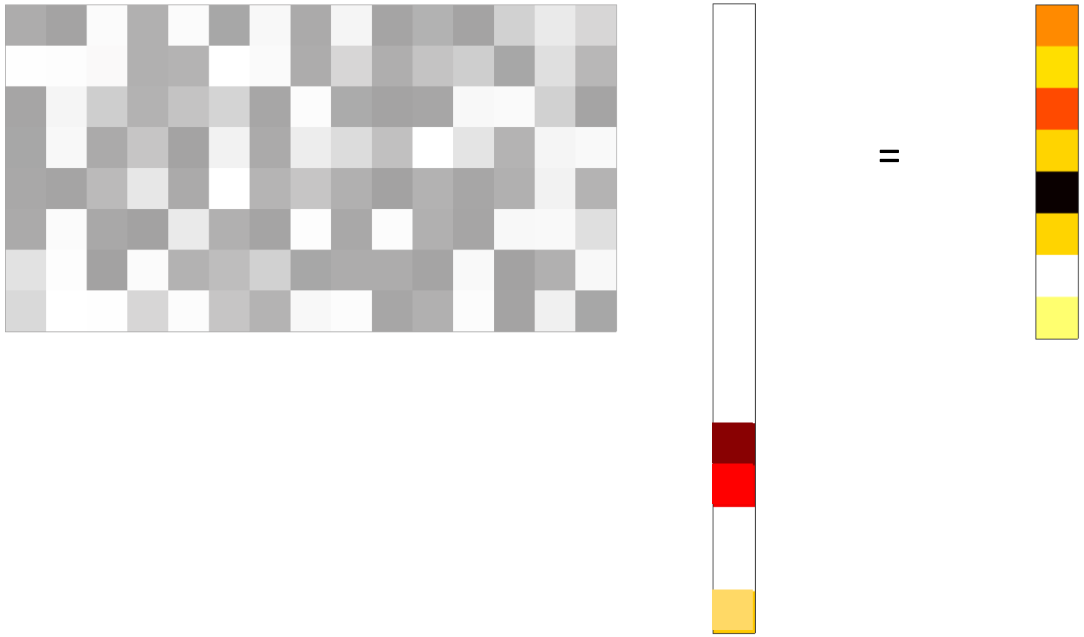
$$\underbrace{\begin{bmatrix} | \\ \vec{x} \\ | \end{bmatrix}}_{n\times 1} = \underbrace{\begin{bmatrix} & & \\ & \Psi & \\ & & \end{bmatrix}}_{n\times n} \underbrace{\begin{bmatrix} | \\ \vec{s} \\ | \end{bmatrix}}_{n\times 1}$$

but mostly zeros!

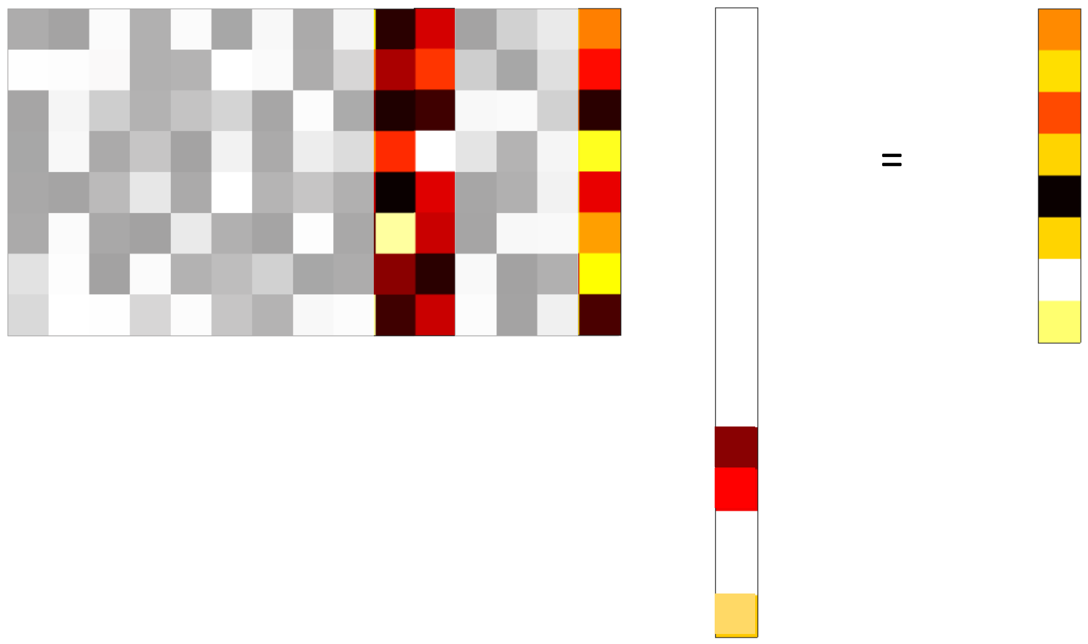
transform matrix
change of basis
↳ DCT, FFT, etc.

ML: dictionary learning is all about finding a good basis transform to make sparse.

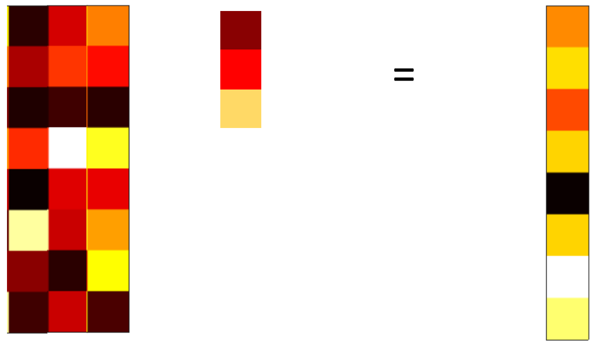
Sparse x means only a few columns of A ‘matter’



Sparse x means only a few columns of A ‘matter’

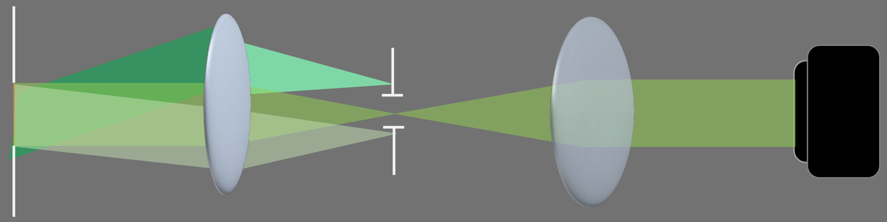


If we knew which elements were non-zero, we could solve a small least squares problem:



Computational Imaging

Hardware Design



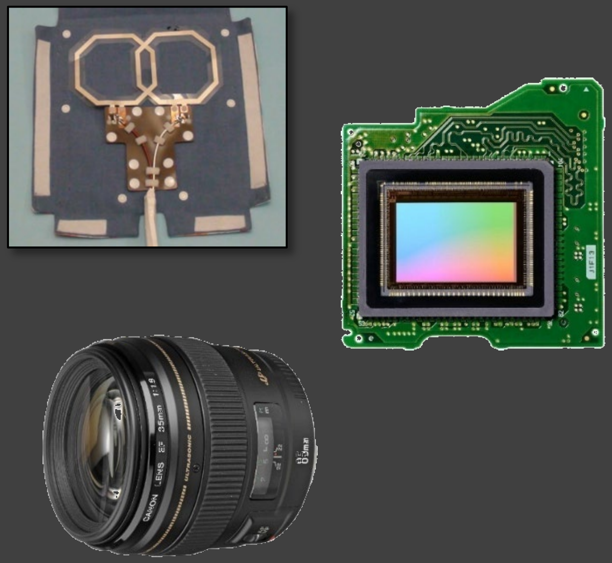
Computation Design

find $\mathbf{x}$

such that $\mathbf{y} = |\mathbf{Ax}|^2$

The hard part is *integration*

Hardware Toolbox

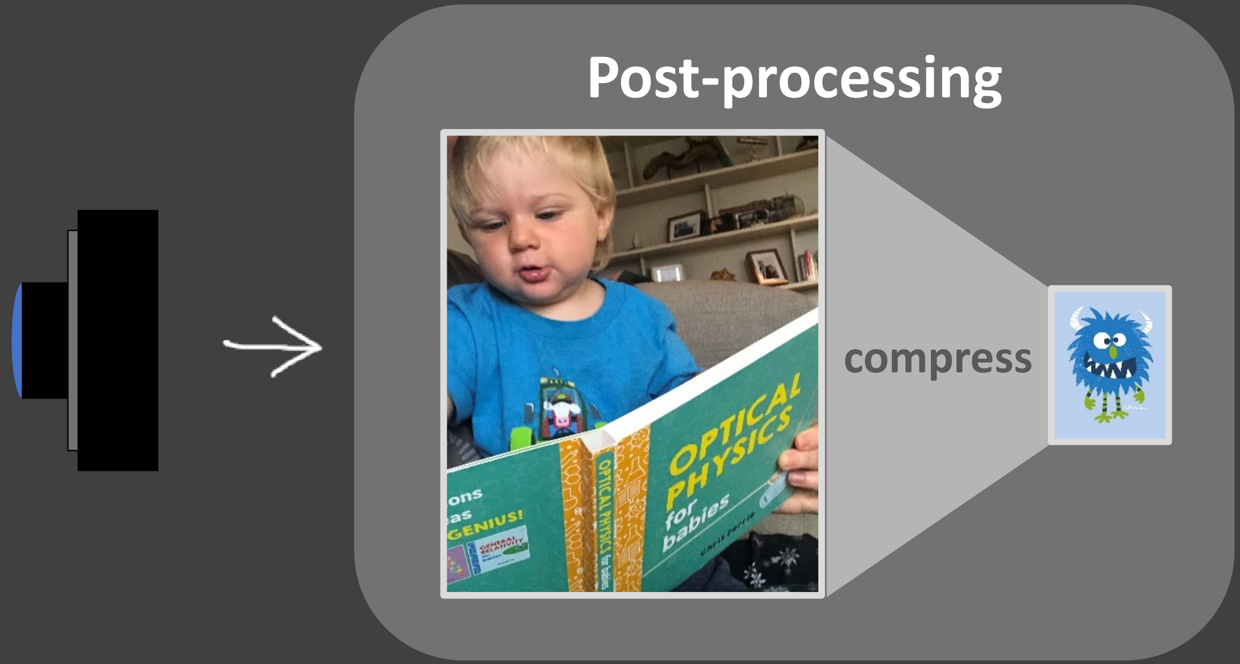


Computational Toolbox

A new way to approach imaging system design

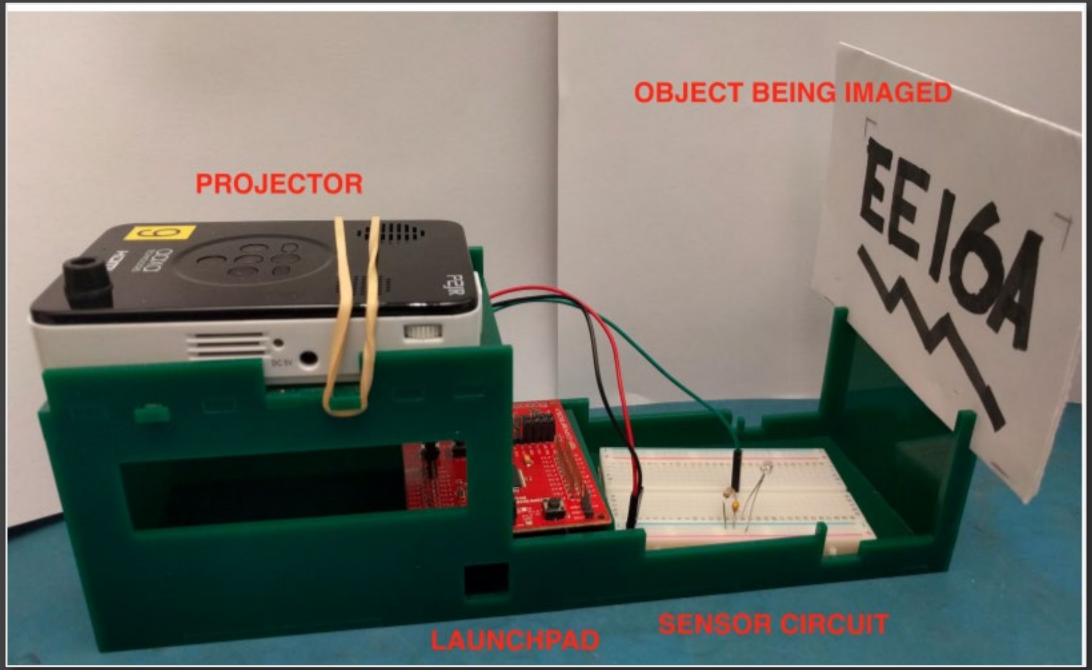
- leverage commodity hardware
- pave the way for design of new devices

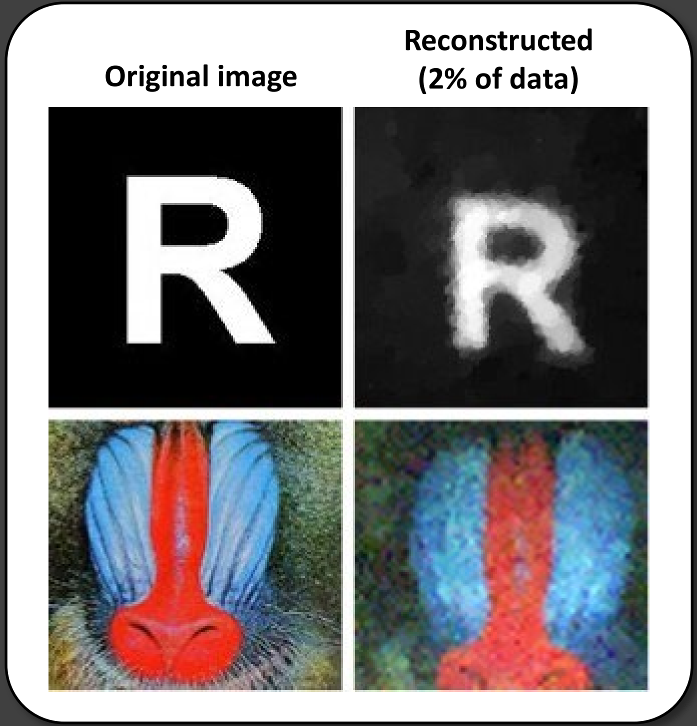
Compressible images can be represented with less data



Can we compress data at the capture stage?

Yes! With compressed sensing! Example: single-pixel camera





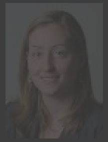
If you design the patterns on your imaging lab well, and images are compressible, you could solve with very little data!

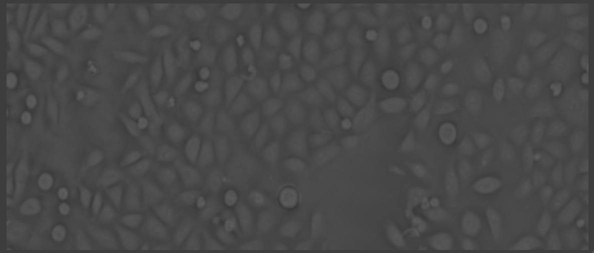
Barniauk et al., Rice University.

Berkeley Center for Computational Imaging

Computational Microscopy

Laura Waller



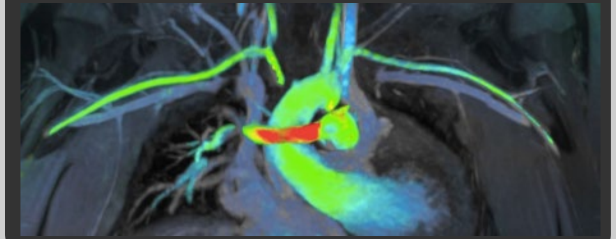


Computational Photography

Ren Ng

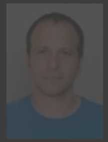
Computational MRI

Michael Lustig



Algorithms & Optimization

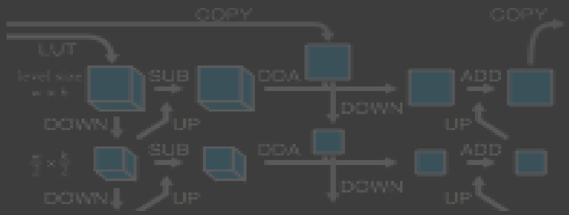
Ben Recht



$$x(s) = \mathrm{PSF} \circledast \left\{ \sum_{j=1}^{k} c_j \delta(s - s_j) \right\}$$

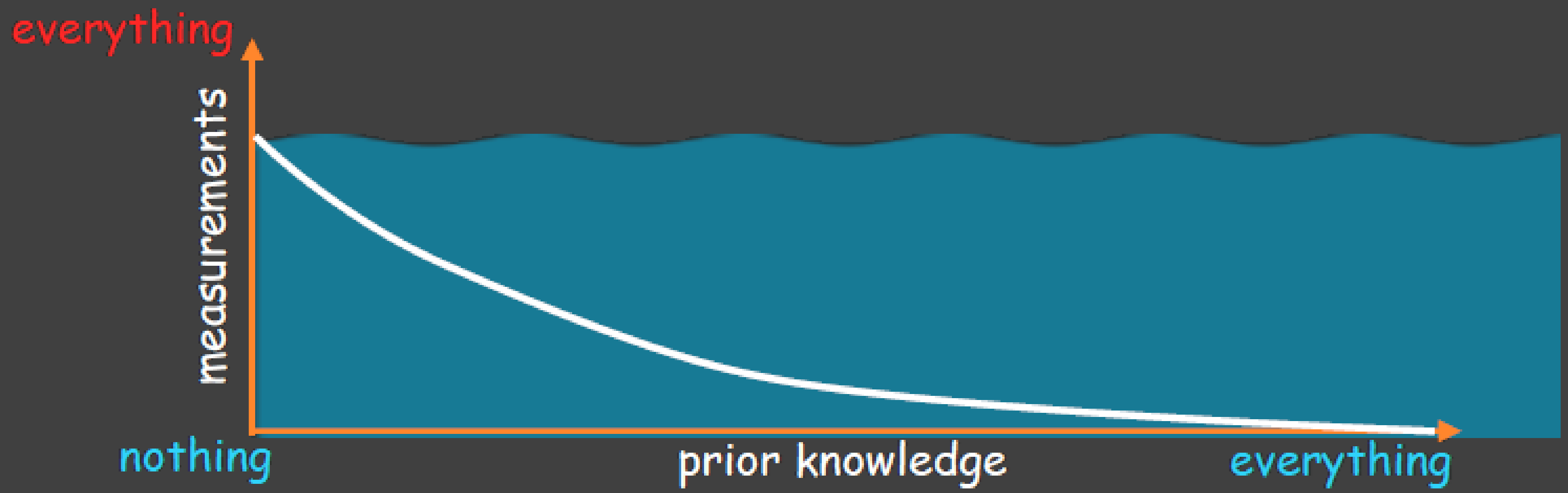
High Performance Visual Computing

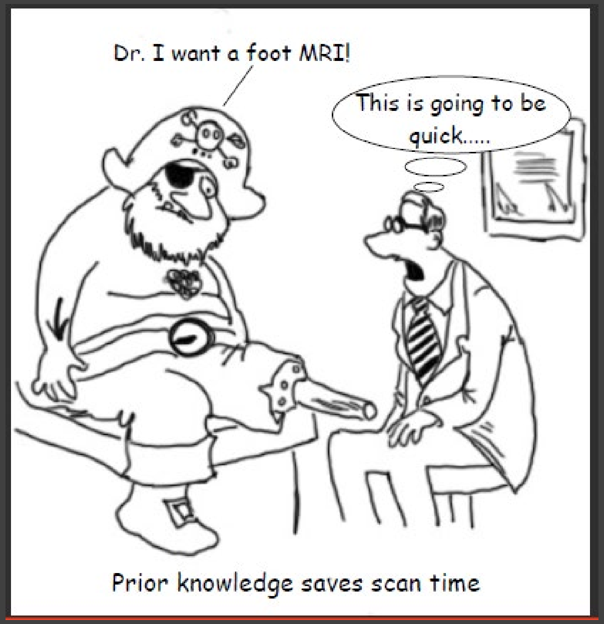
Jonathan Ragan-Kelley



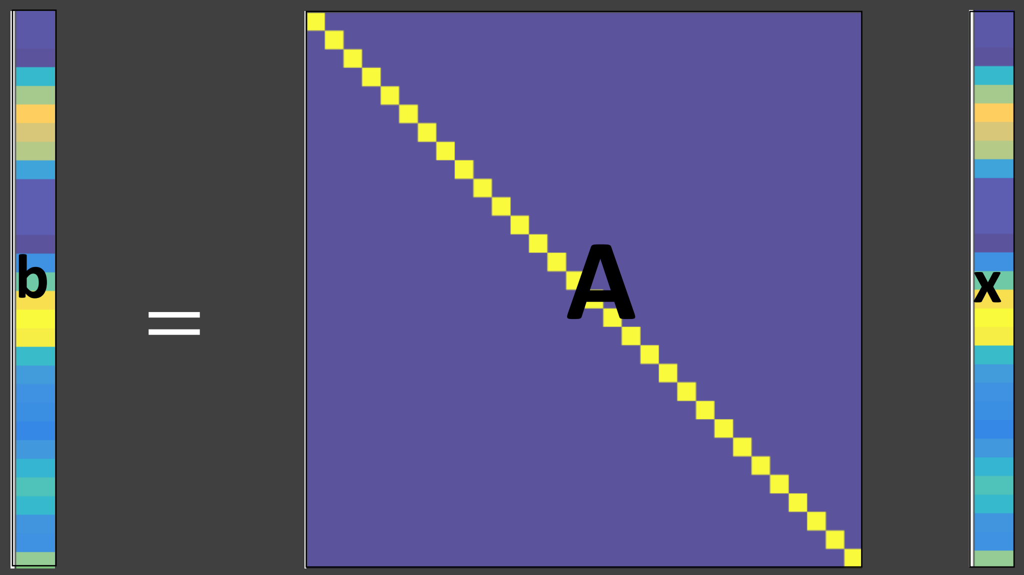
Compressed sensing is all about using prior knowledge

- Redundancy reduces sampling requirements
  (The more you know, the less you need)

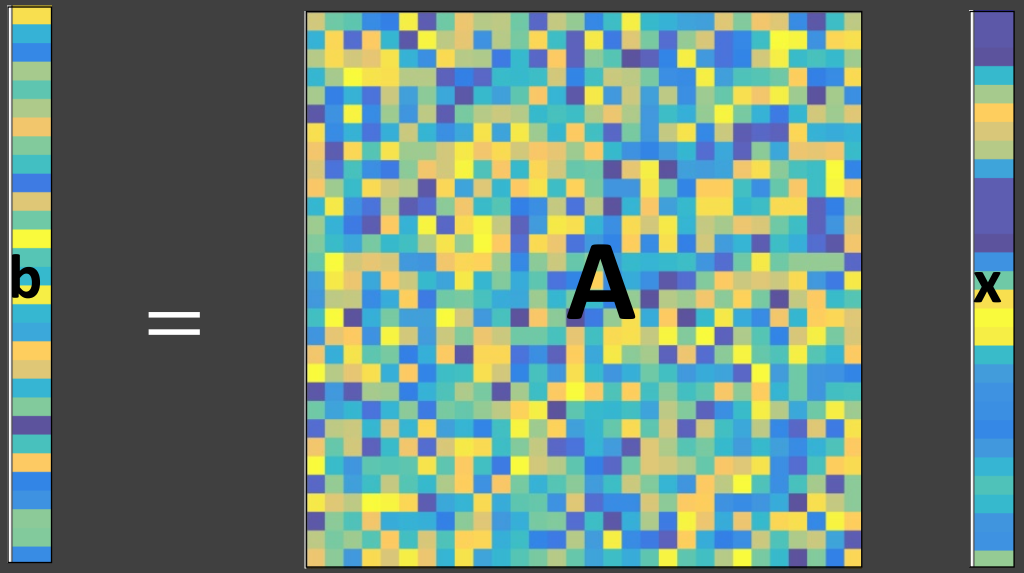


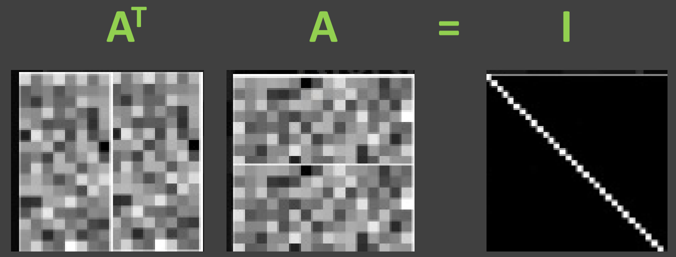


Traditional sensing: direct measurements



Multiplexed measurements

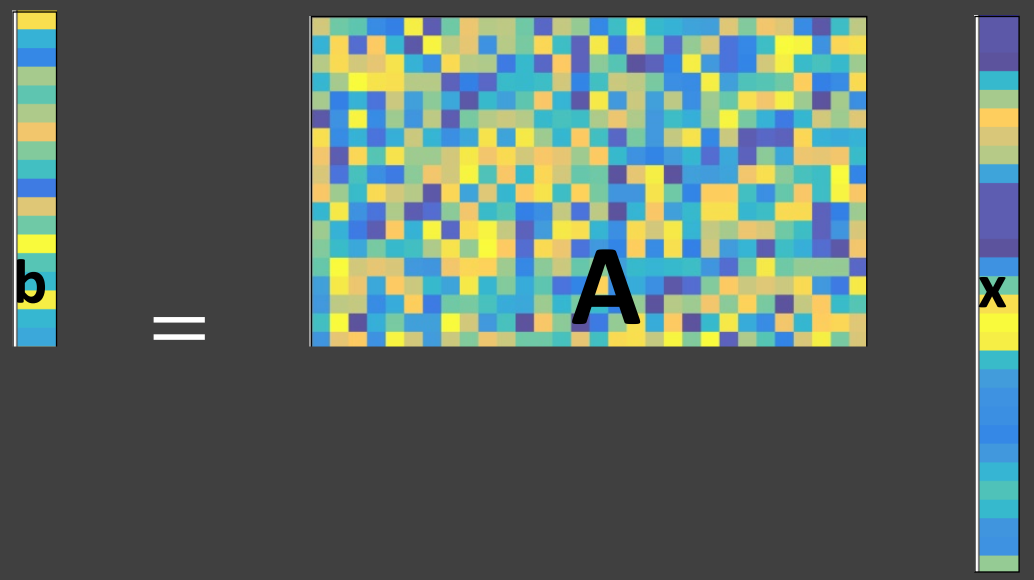




What makes a good sensing matrix?

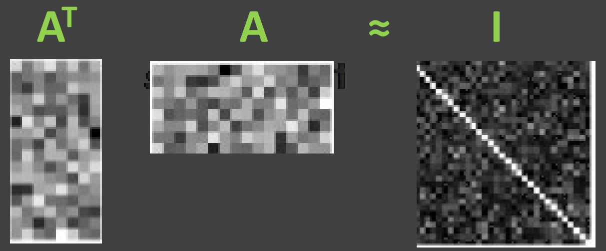
A is orthogonal

Compressed sensing solves underdetermined problems using sparsity



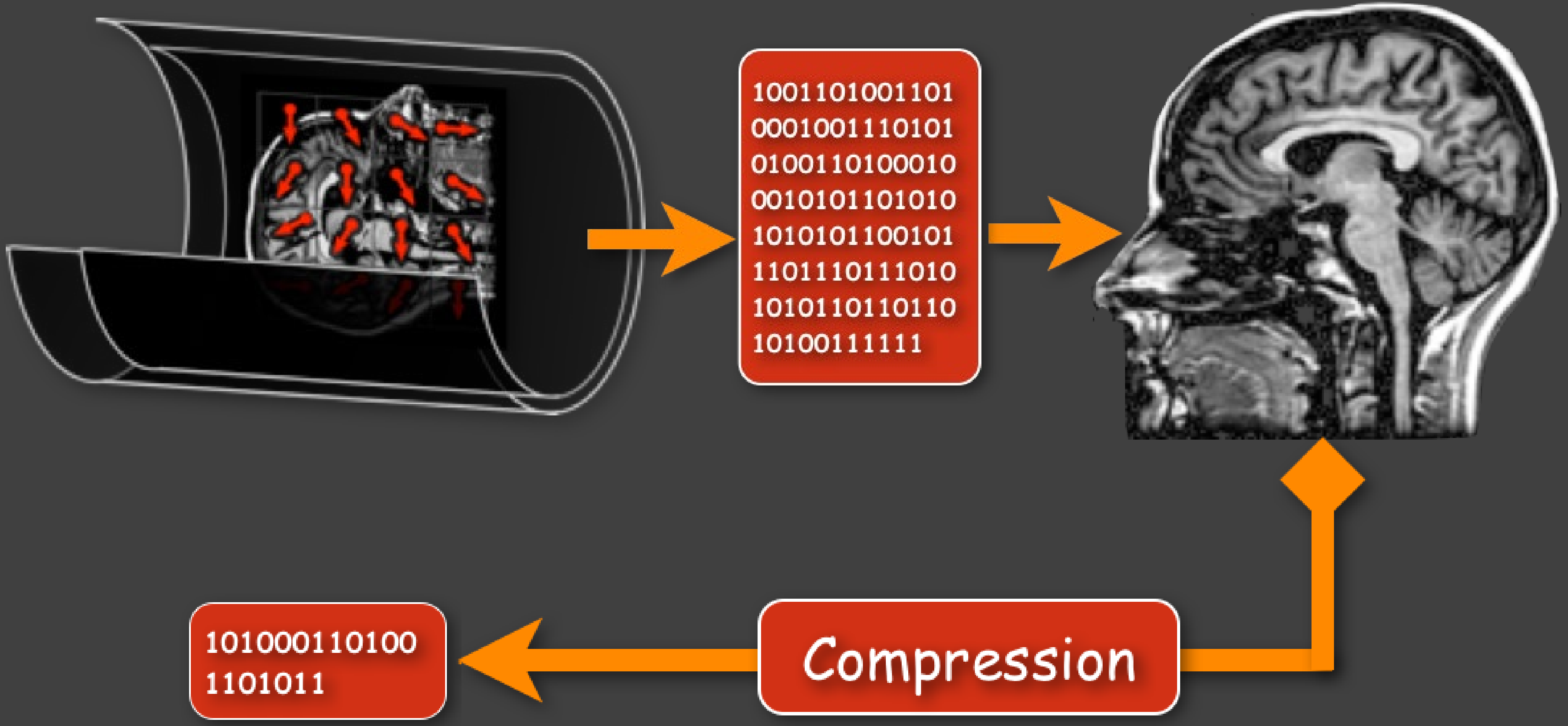
What makes a good sensing matrix?

A is (almost) orthogonal



Compressed Sensing MRI

Medical images are compressible
  Standard approach: First collect, then compress

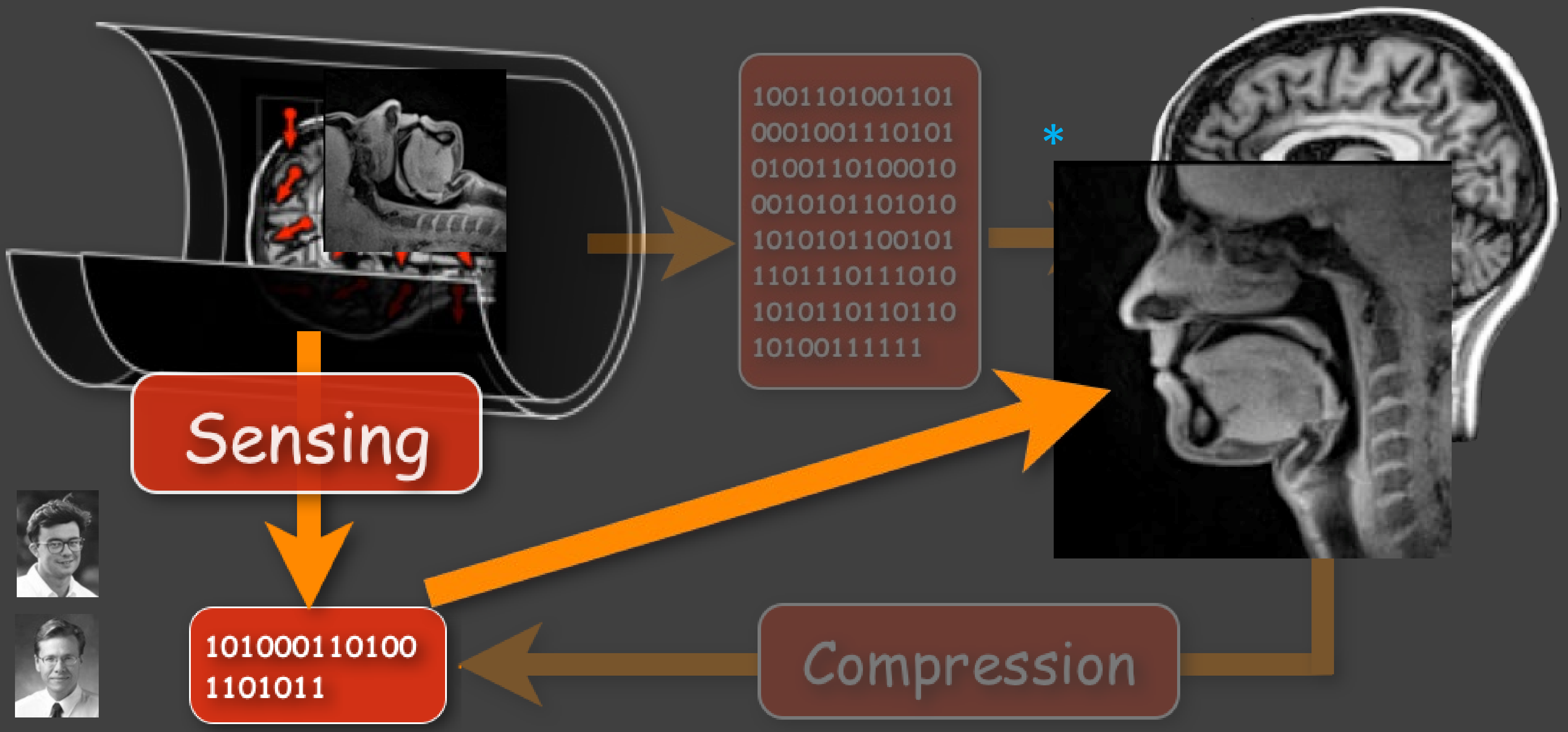


Michael Lustig's Lab 

Compressed Sensing MRI

Medical images are compressible

New approach: Acquire "compressed" data directly!



*Courtesy, M. Uecker, J. Frahm, Max Planck

Michael Lustig's Lab

12 month male (114 bpm)

Scan time: 11:05 min; Total acceleration

Resolution: 0.9 x 0.9 x 1.4 mm^3

VENC: (150, 150, 150) cm/s

Contrast: ferumoxytol

Generated using Arterys (San Francisco, CA)

powered by BART

Lucile Packard Children's Hospital Stanford

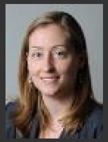
Prof. Shreyas Vasanawala

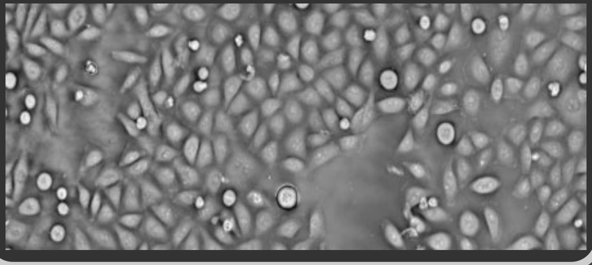
J.Y. Cheng et al., JMRI 2016. doi: 10.1002/jmri.25106

Berkeley Center for Computational Imaging

Computational Microscopy

Laura Waller



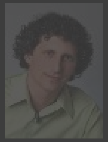


Computational Photography

Ren Ng

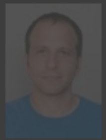
Computational MRI

Michael Lustig



Algorithms & Optimization

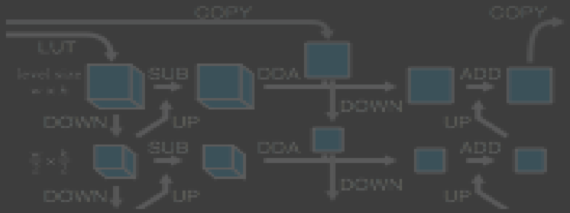
Ben Recht



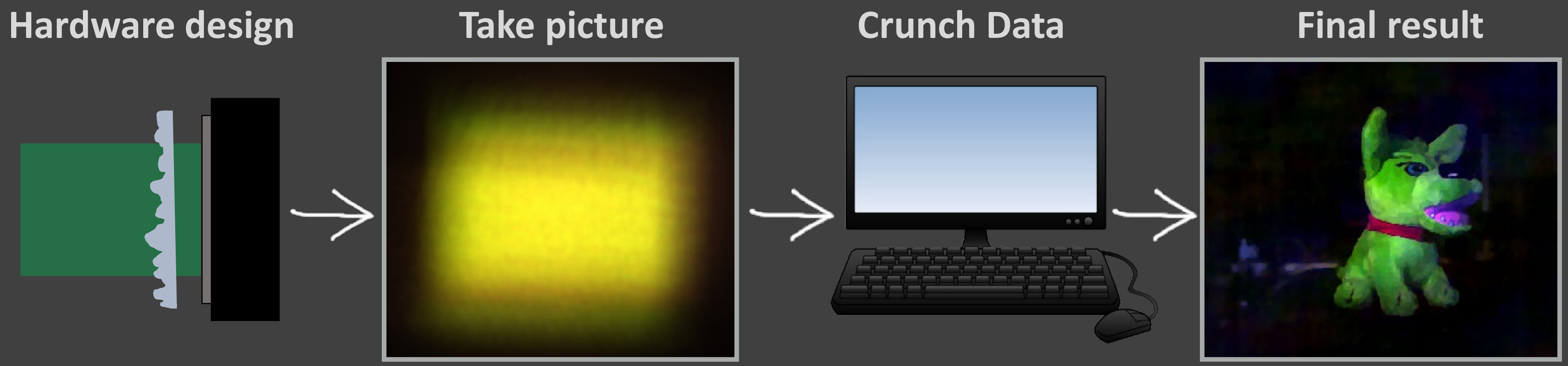
$$x(s) = \mathrm{PSF} \circledast \left\{ \sum_{j=1}^{k} c_j \delta(s - s_j) \right\}$$

High Performance Visual Computing

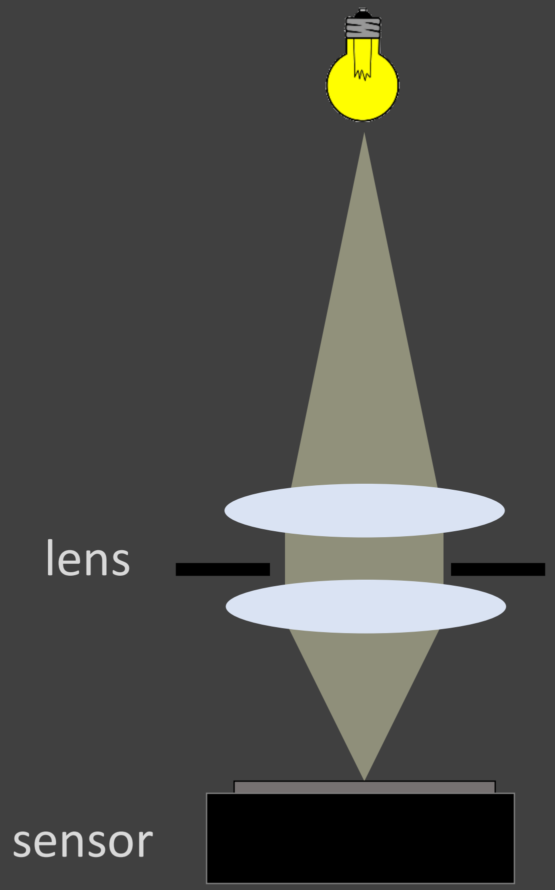
Jonathan Ragan-Kelley



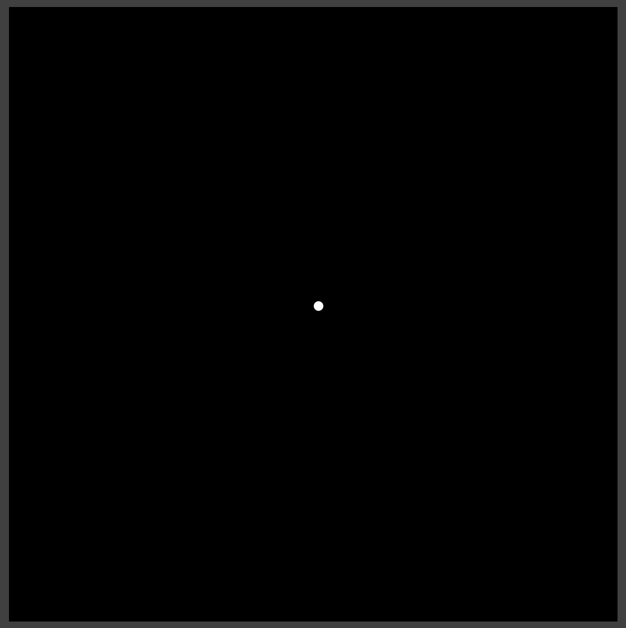
Computational imaging pipeline



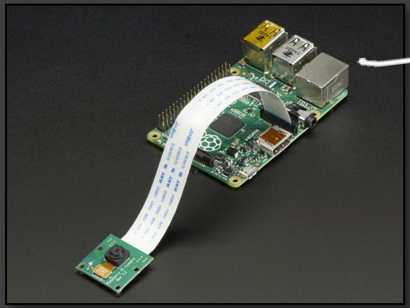
Lenses map points to points



System response to point source

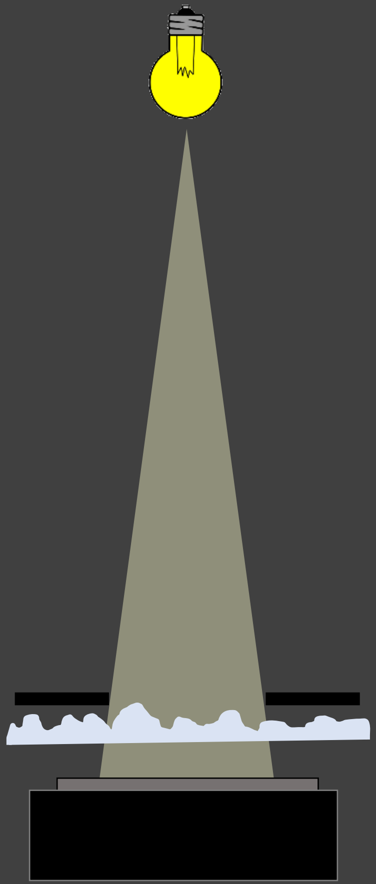


DiffuserCam: stick a scatterer on a sensor

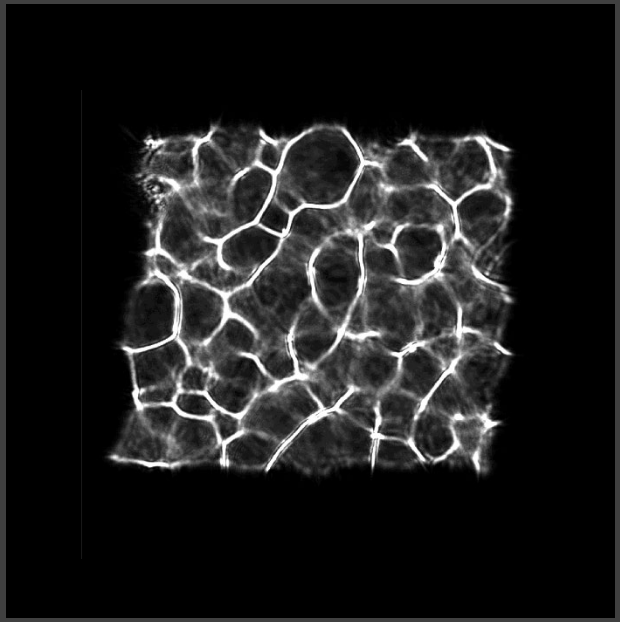


diffuser

sensor



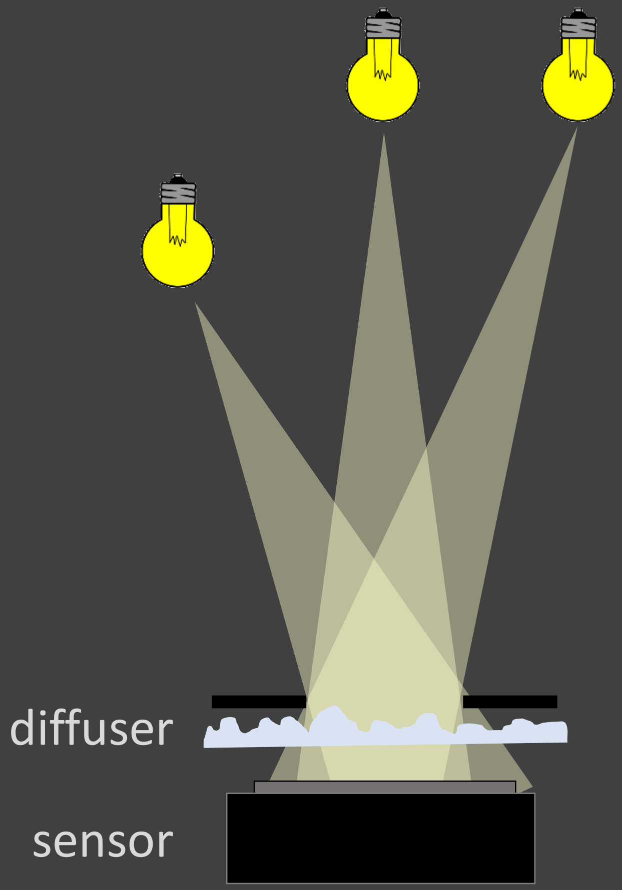
System response to point source

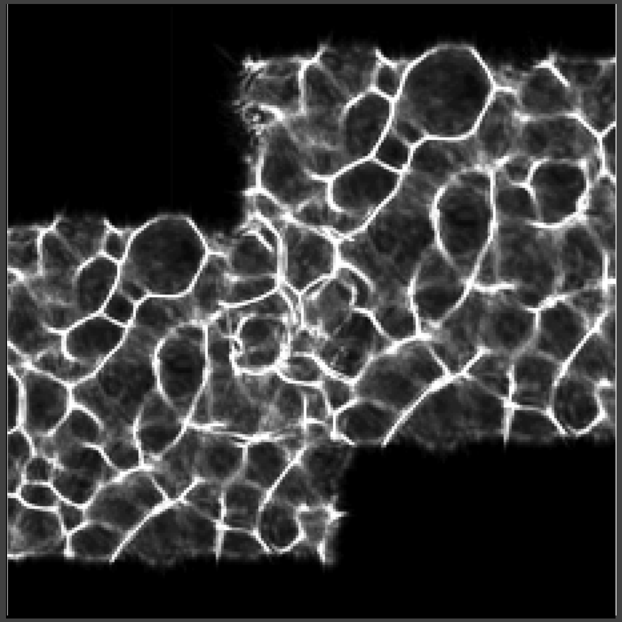


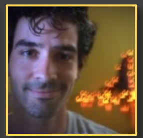
Camille Biscarrat
Shreyas Parthasarathy


https://laurawaller.com/opensource

DiffuserCam: stick a scatterer on a sensor

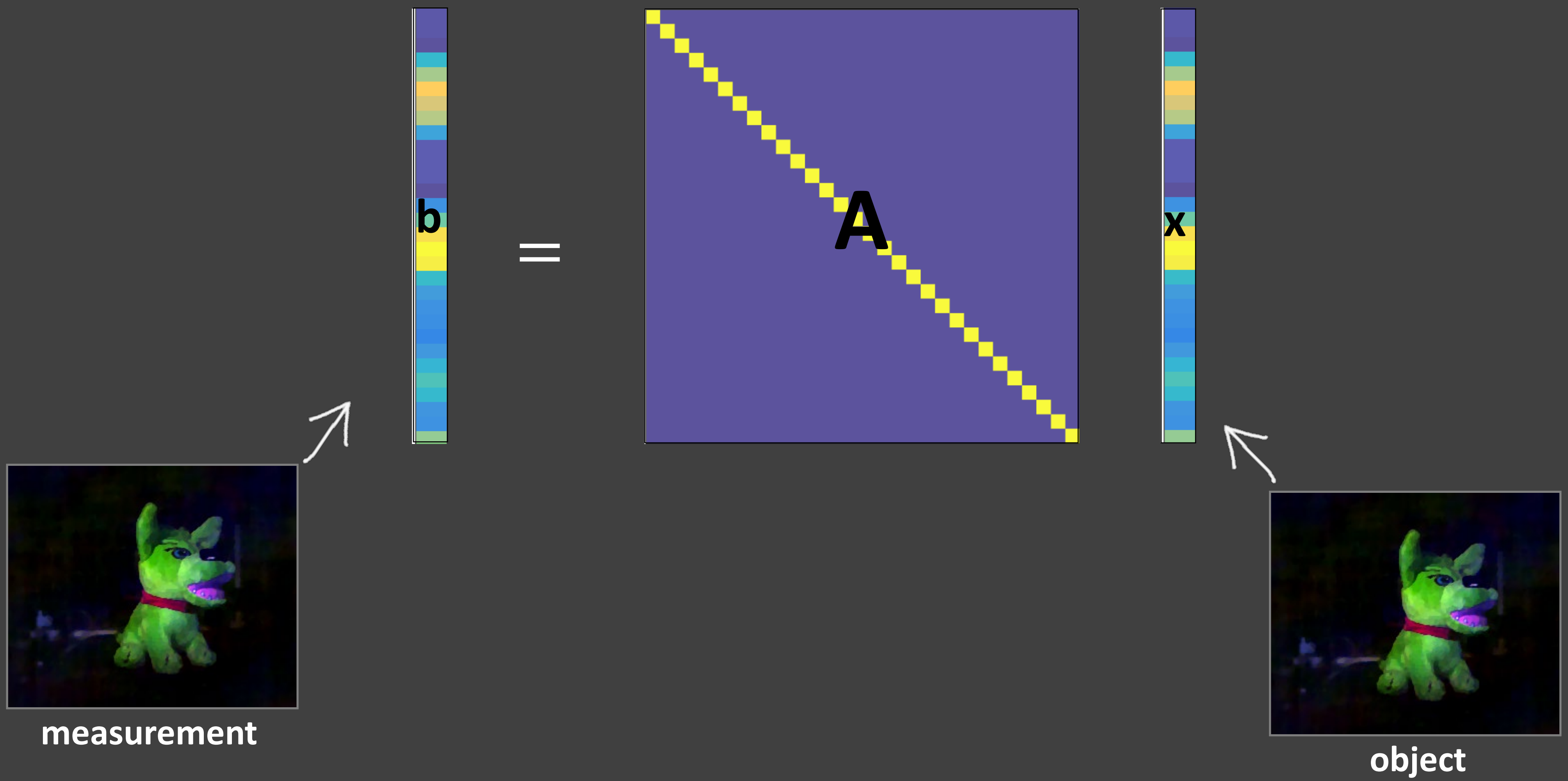




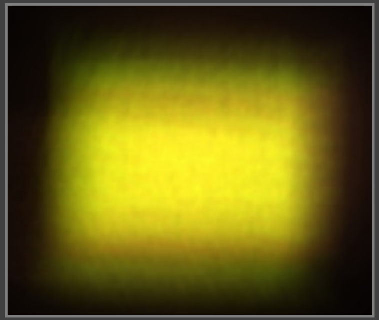


Grace Kuo
Nick Antipa

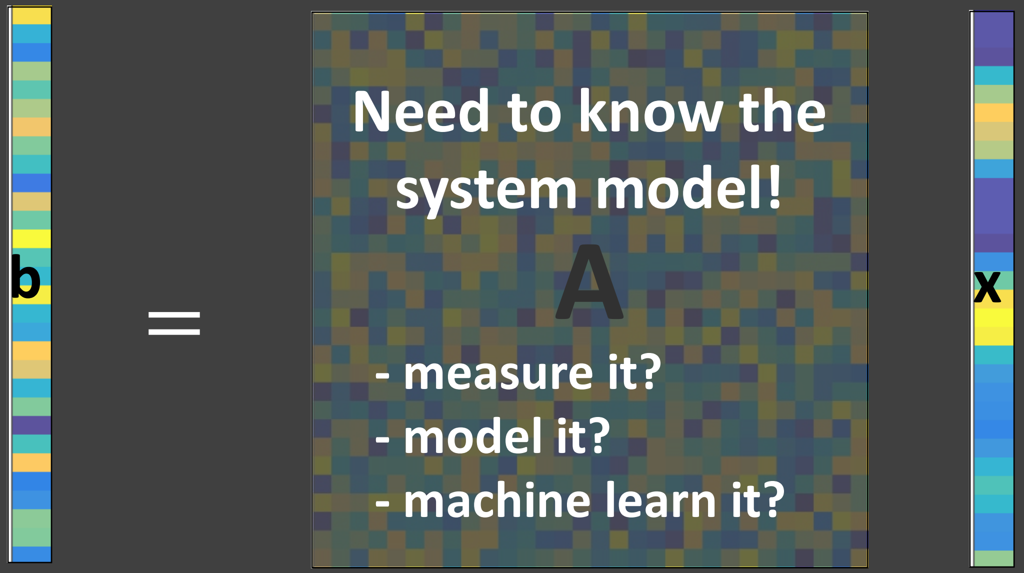
Traditional cameras take direct measurements



Computational cameras can multiplex

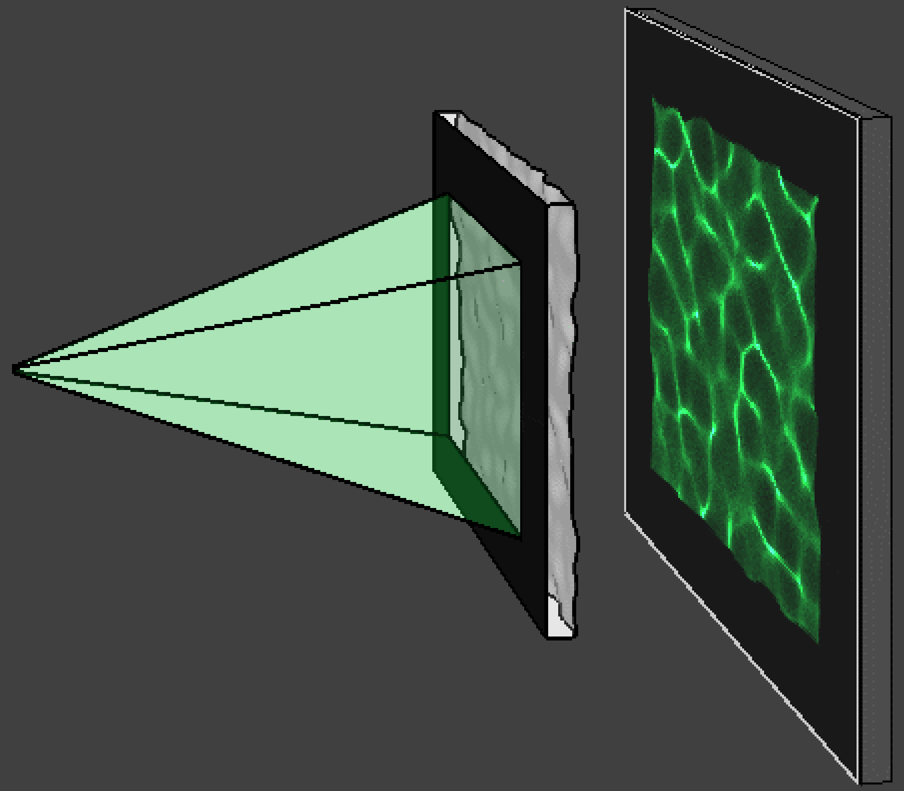


measurement

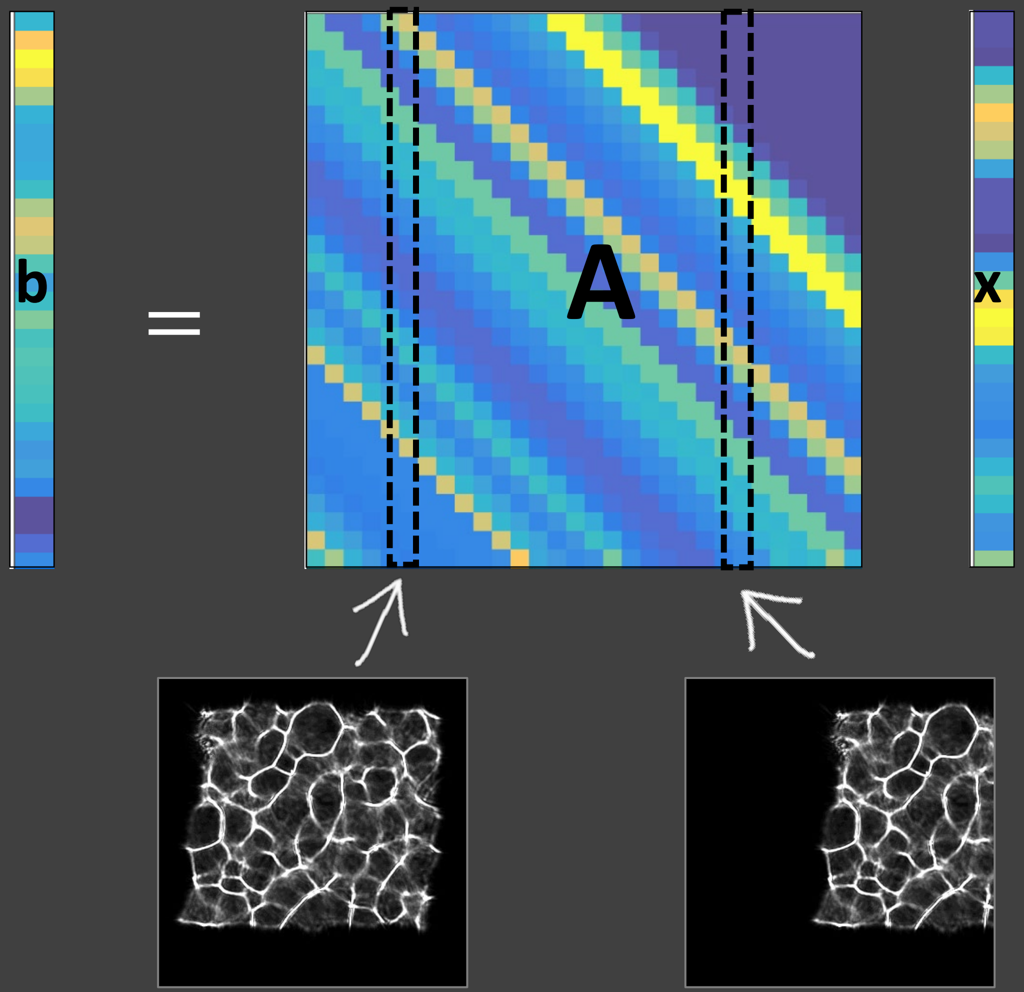


object

System response shifts with position

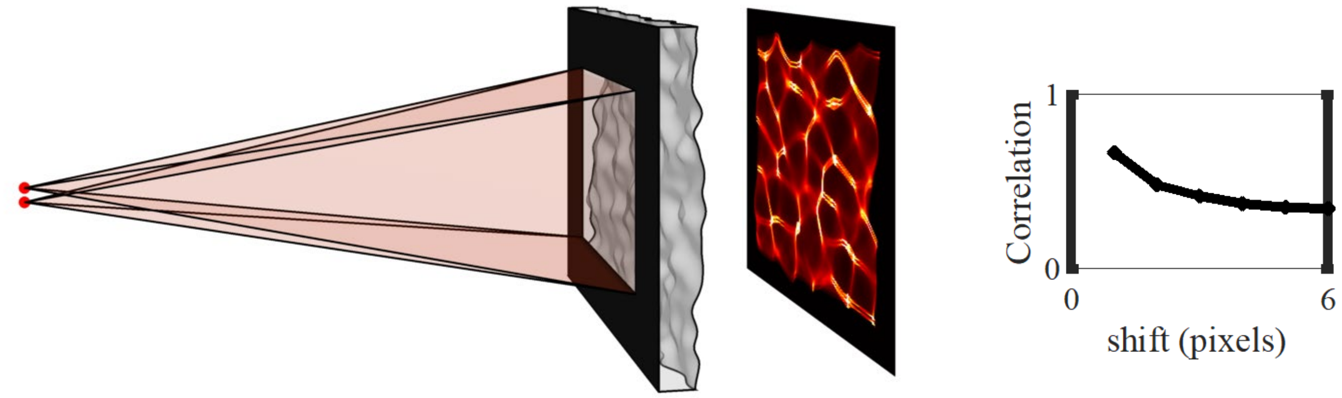


DiffuserCam system model is a shift-invariant

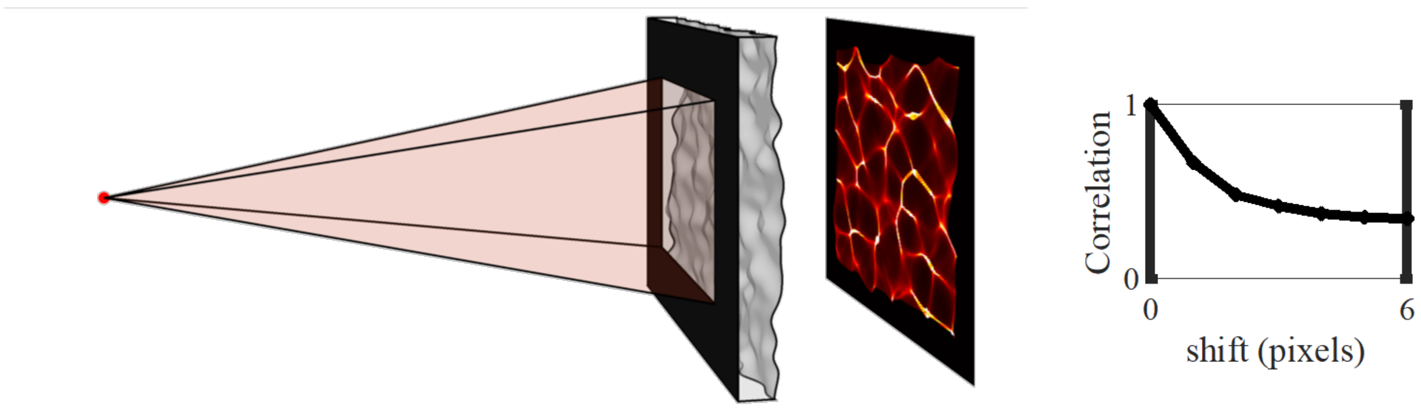


System response is same but shifted for different image pixels

We could find location of a point by correlating image captured with shifts in system response!

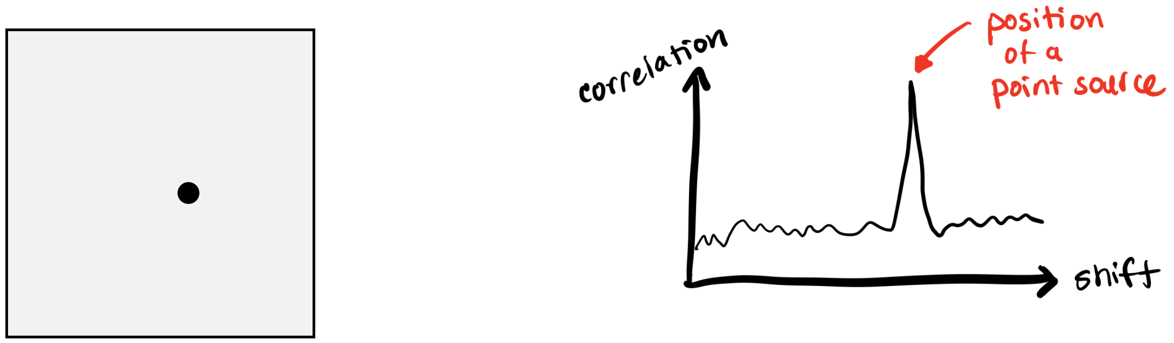


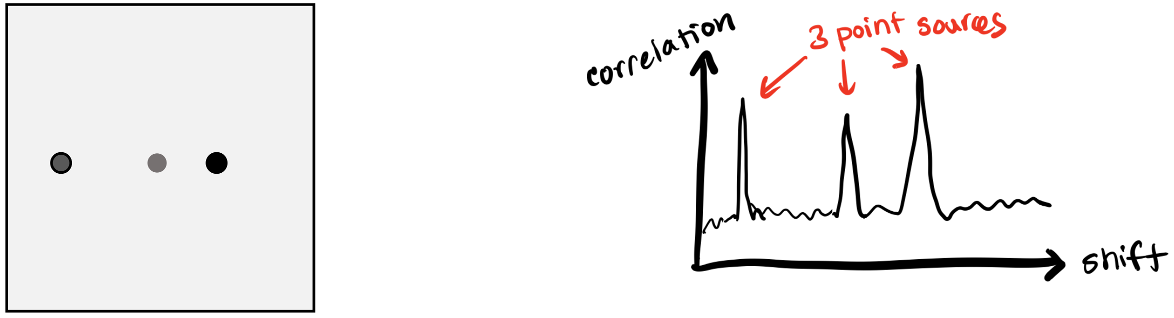
We could find location of a point by correlating image captured with shifts in system response!



Reconstructing image amounts to finding strength of each 'point source':

Looks a lot like our GPS problem! (especially if image is sparse)





raw sensor data

recovered scene

*solver is ADMM with TV reg in Halide

Grace Kuo
Nick Antipa

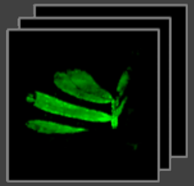
raw sensor data

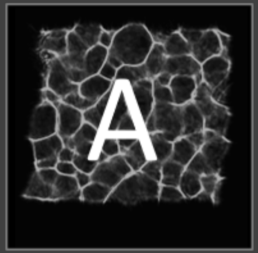
recovered scene

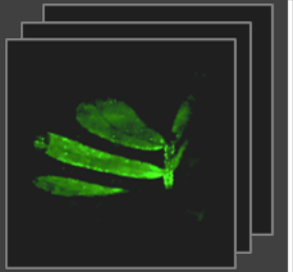
Grace Kuo
Nick Antipa

*solver is ADMM with TV reg in Halide

Image reconstruction is nonlinear optimization







$$\underset{x \geq 0}{\arg\min} \left\| b - A * x \right\|_2^2 + \lambda \left\| \Phi x \right\|_1$$

Sparsity basis

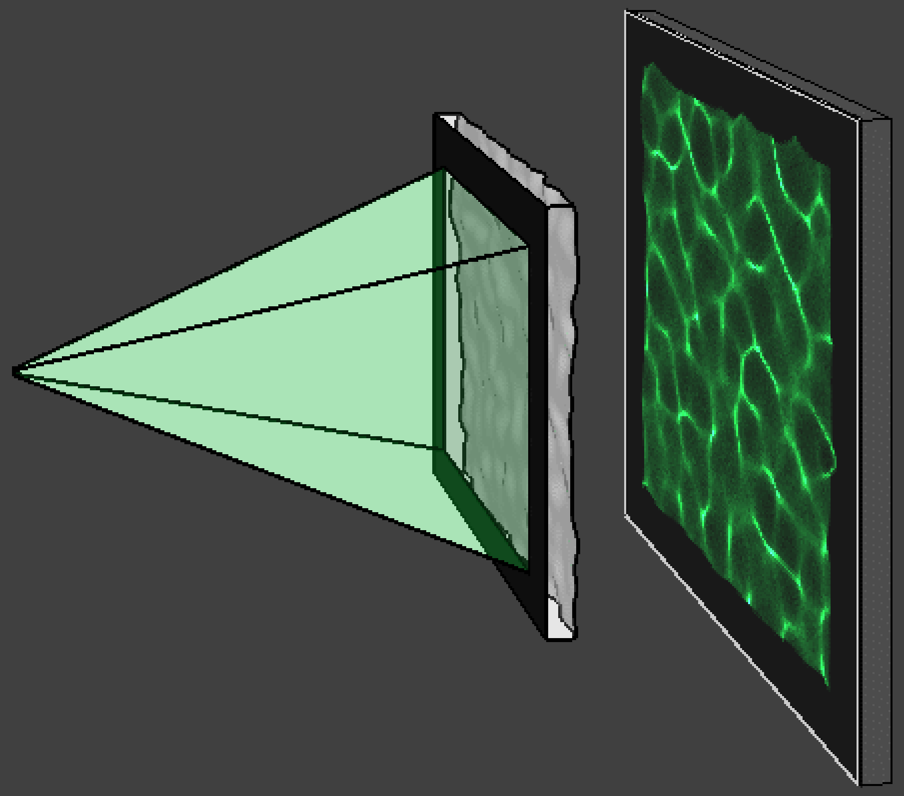
***solved with ADMM in Halide**

S. Boyd, et al. *Foundations and Trends in Machine Learning* (2011)

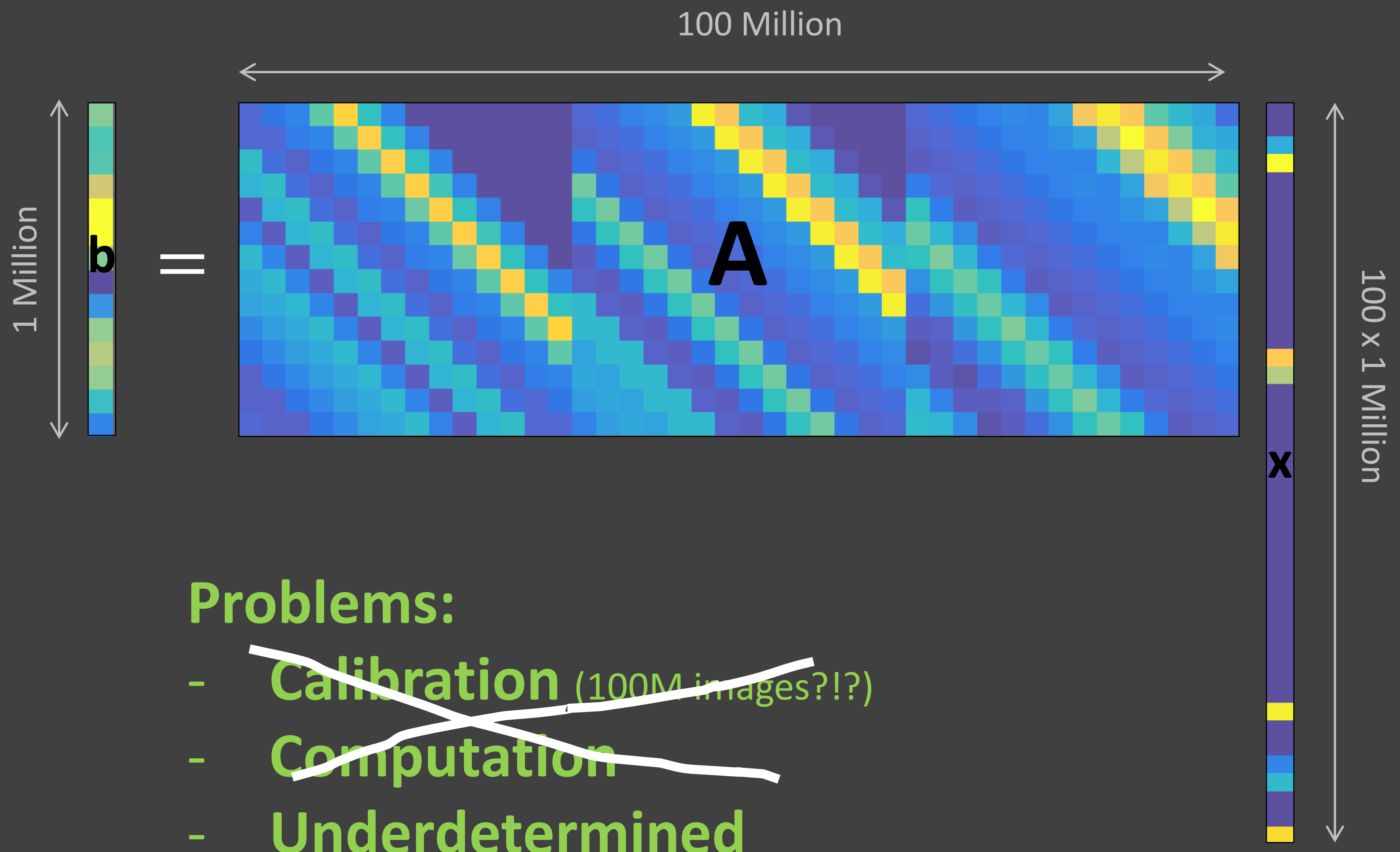
J. Ragan-Kelley, et al. *AMC SIGPLAN* (2013)

2D →

Point spread function scales with depth



Single-shot 3D is underdetermined

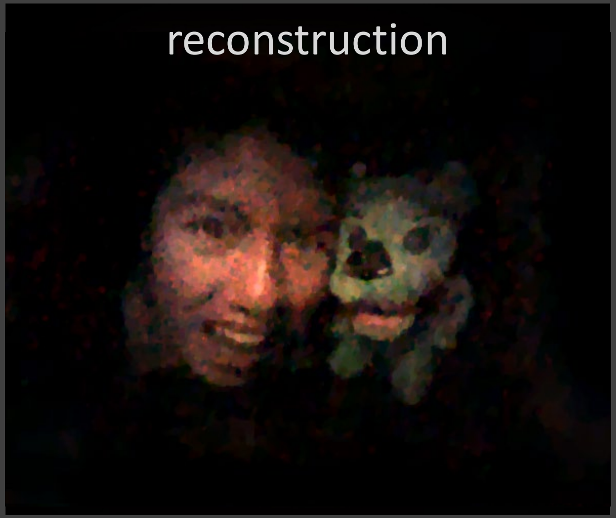


Problems:

- ~~**Calibration** (100M images?!?)~~
- ~~**Computation**~~
- **Underdetermined**

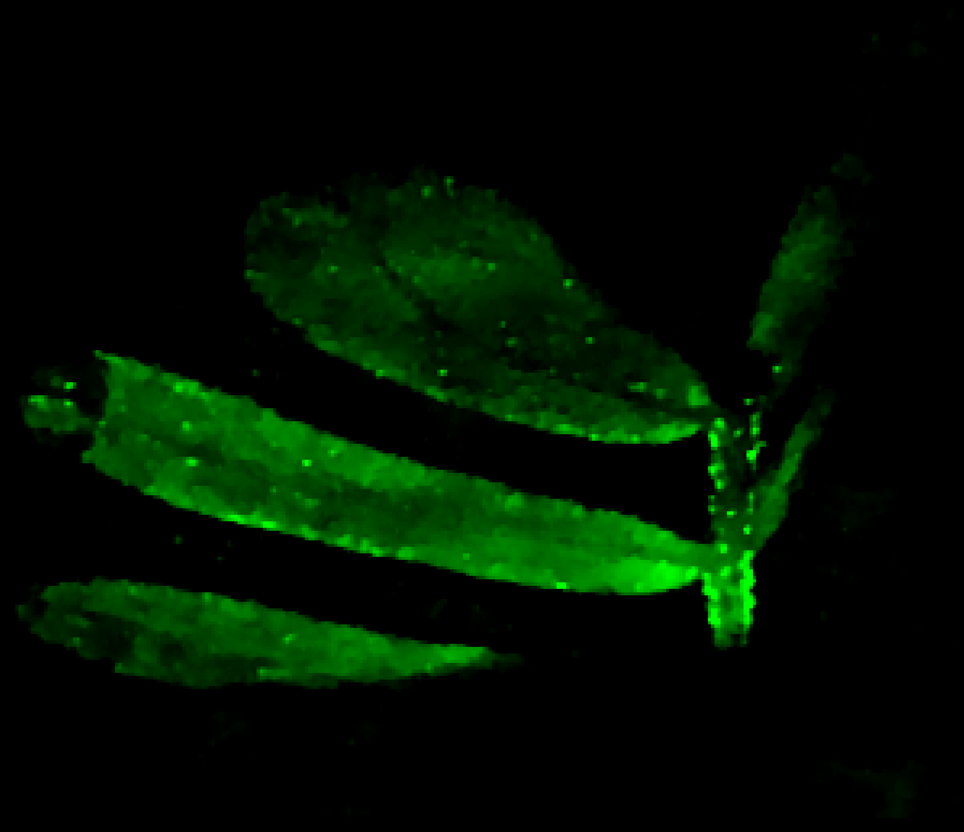
Compressed sensing to the rescue!

→ solves under-determined problems via a sparsity prior



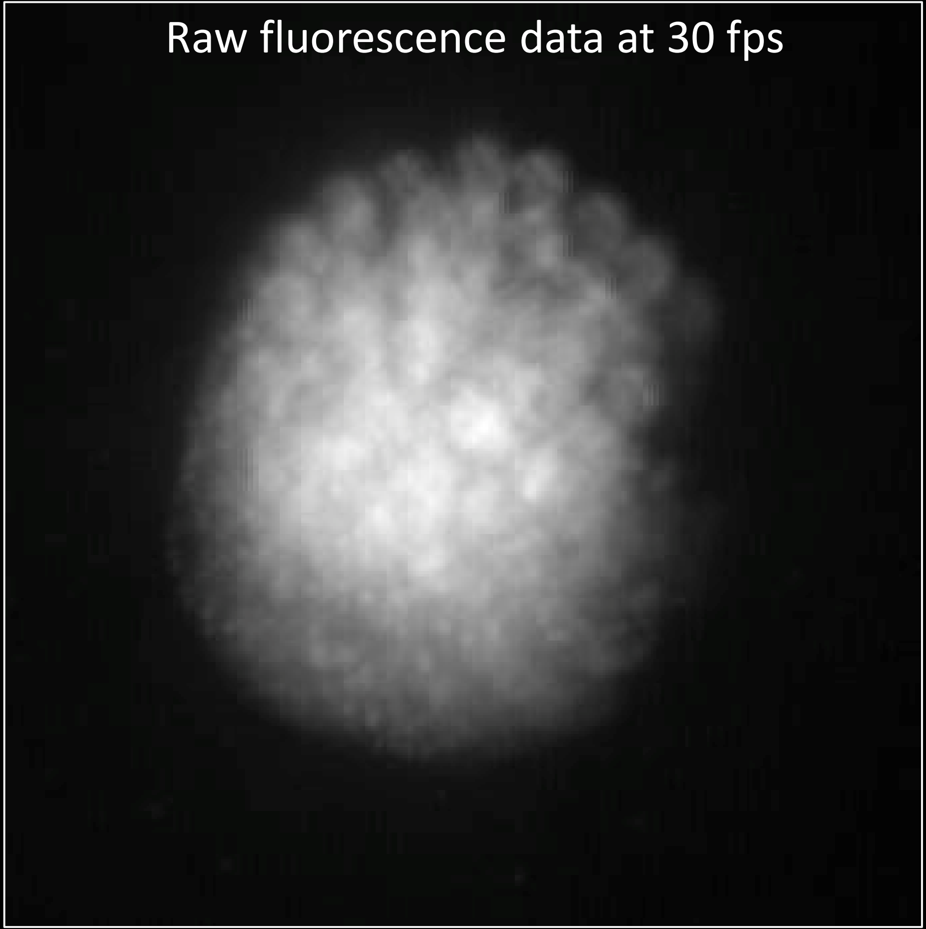
E. J. Candes, et al. *IEEE Signal Processing* (2008)
L. I. Rudin, et al. *Physica D: Nonlinear Phenomena* (1992)

128x more voxels for ***FREE!***

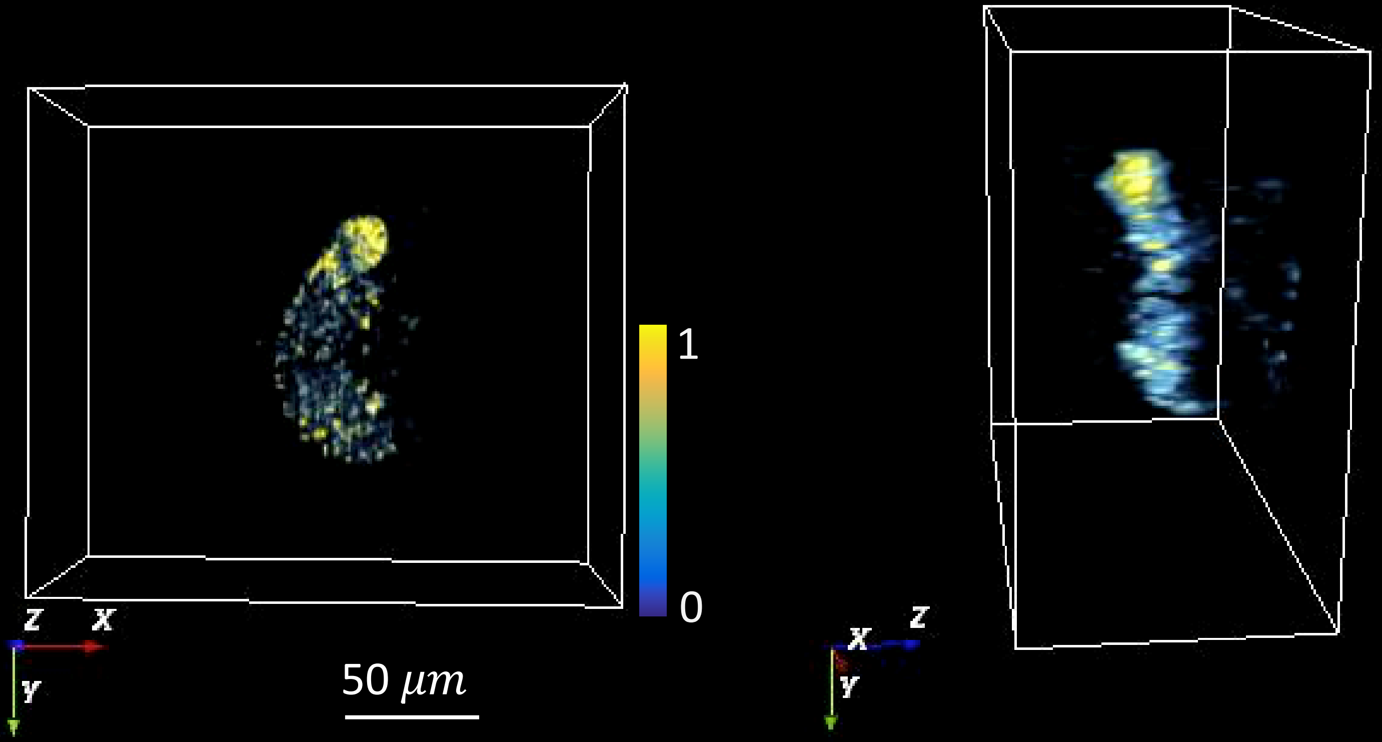


N. Antipa, G. Kuo, R. Heckel, E. Bostan, B. Mildenhall, R. Ng, L. Waller, *Optica* 5(1) (2017).

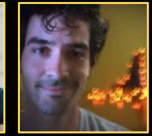
Raw fluorescence data at 30 fps



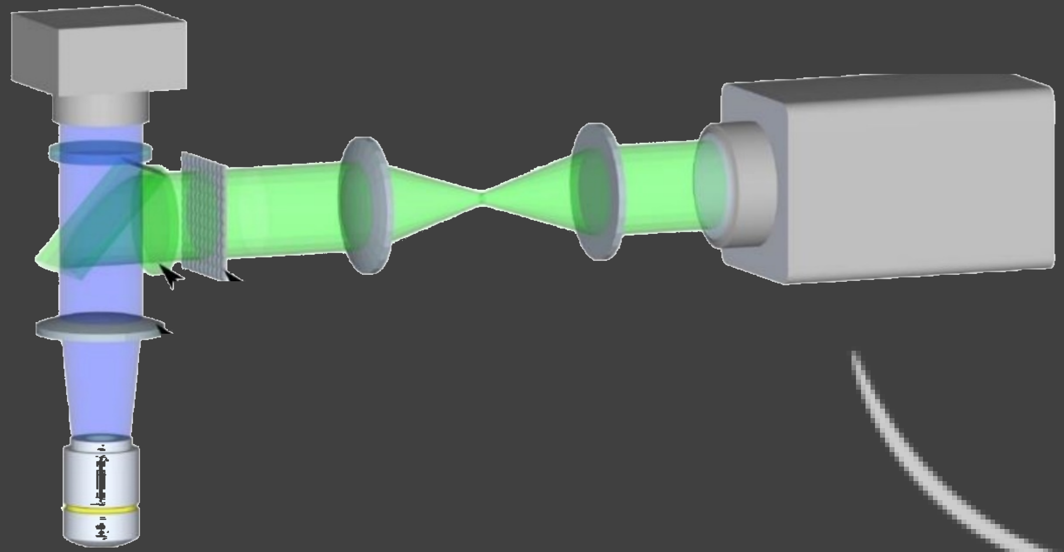
3D video reconstruction

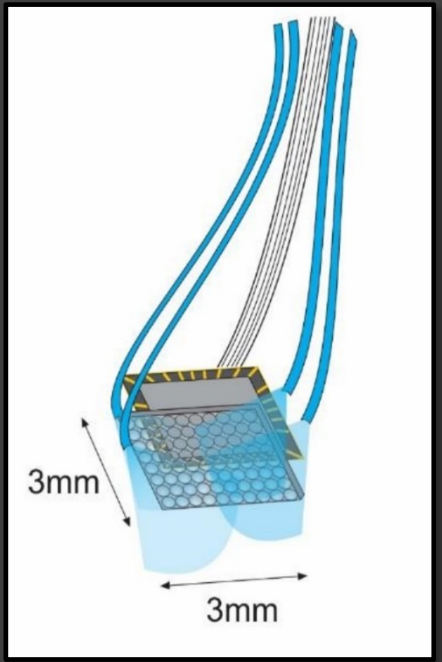


Kyrollos Yanny
Nick Antipa

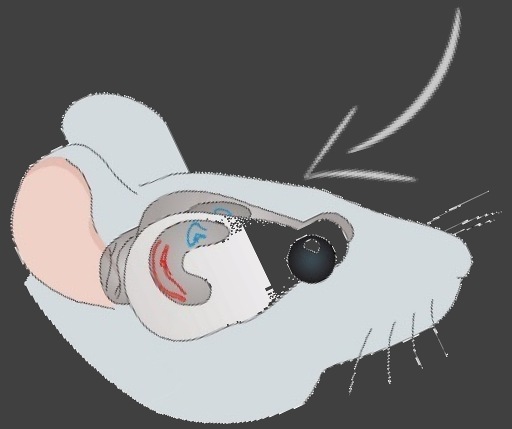


Towards lensless 3D microscopy

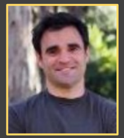




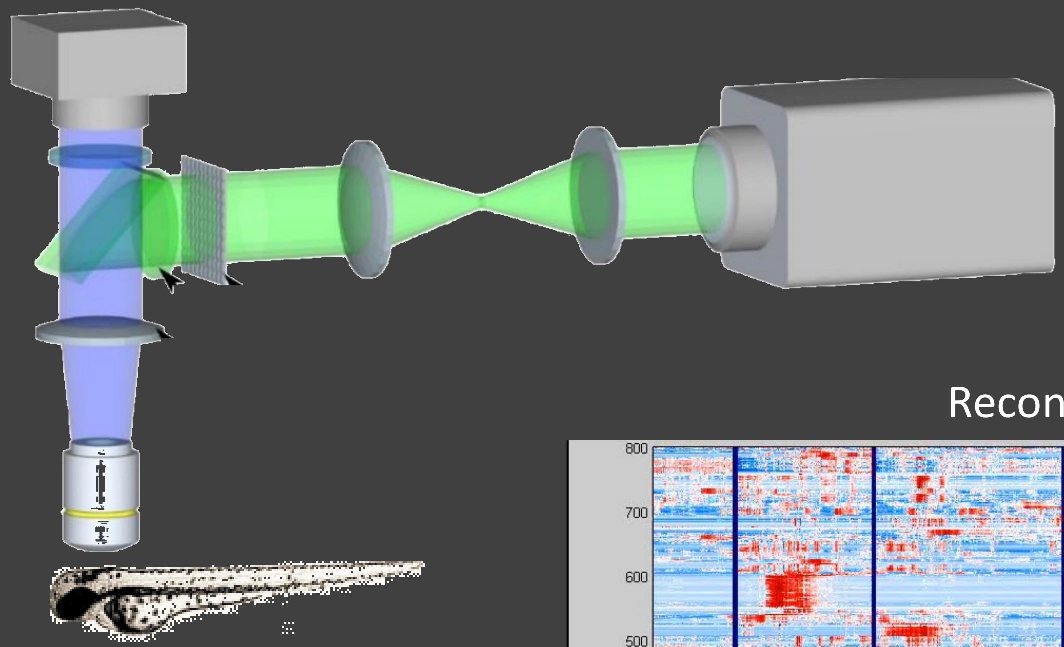
Lensless imager:
- small
- inexpensive
- enables tiling



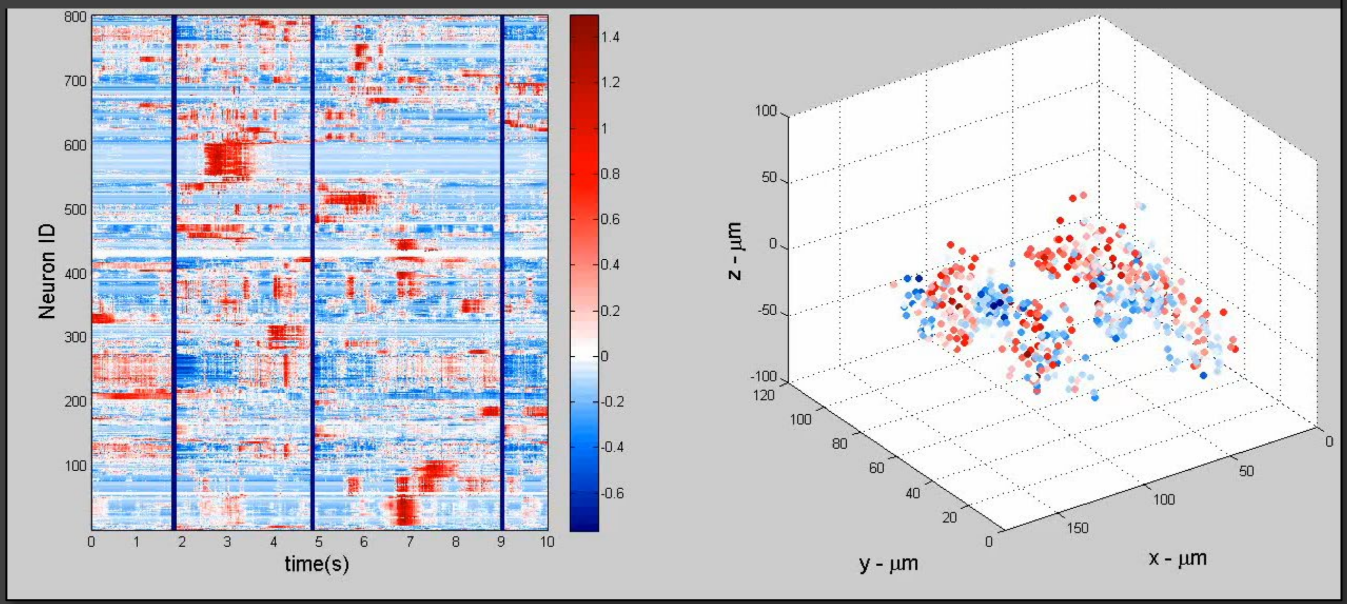
with Adesnik Lab



3D neural activity tracking



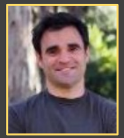
Reconstructed neural activity



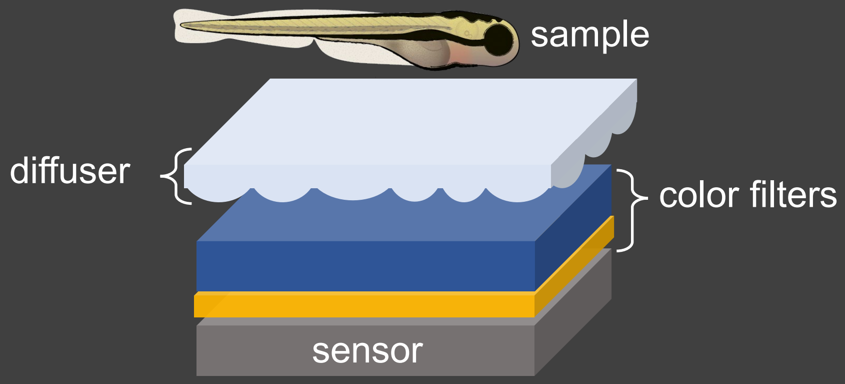
Nico Pegard

N. Pegard et al, *Optica* 2016

with Adesnik Lab



Neural activity tracking with flat DiffuserScope



Grace Kuo

G. Kuo, F. Liu, I. Grossrubatscher, R. Ng, L. Waller, *Optics Express* (2020).

Outlook: Computers and optics should talk more

Hardware Toolbox

Computational Toolbox

Pushing the limits of imaging can be done with existing physics and commodity hardware.

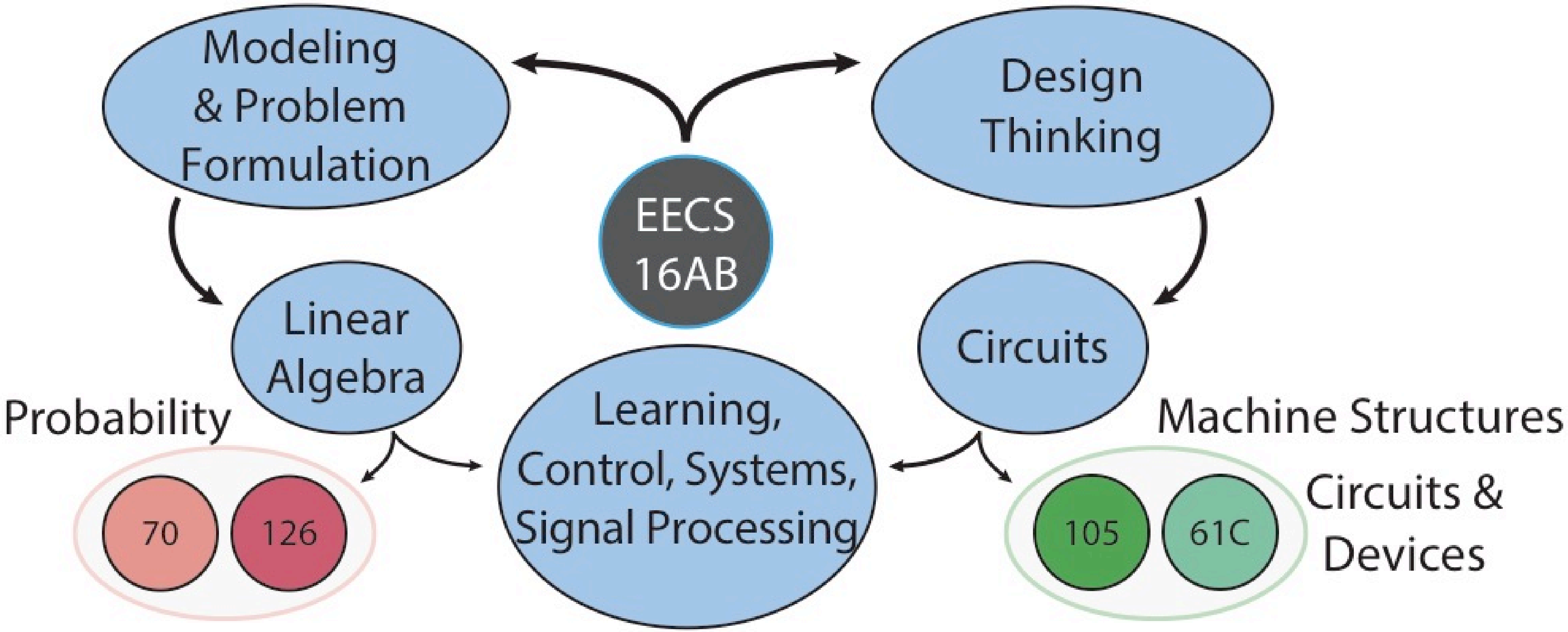
Enough about me…

- Congrats! This year has not been easy!

- What you have accomplished this semester:
  - Built a camera
  - Built two types of touchscreens
  - Built your own GPS system

- If you liked the class, please:
  - Thank your TAs!
  - apply to become one! (see piazza post, due May 2)

- This is just the beginning…

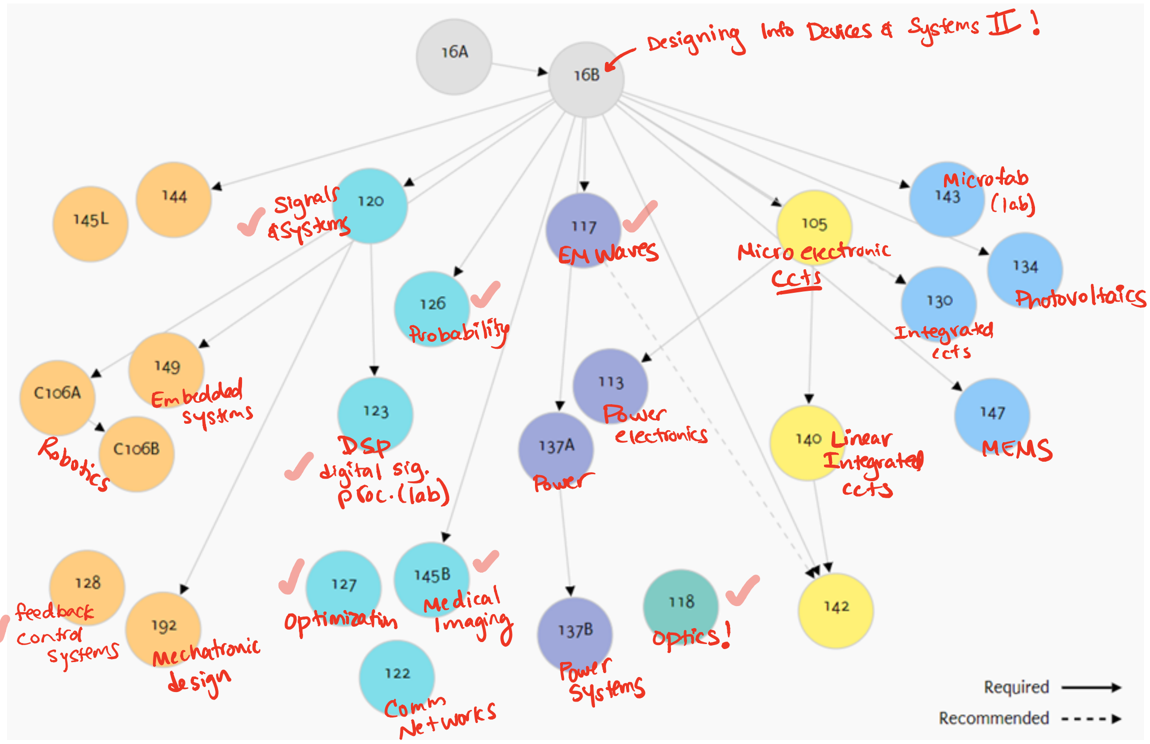


How to approach something unfamiliar
and systematically build understanding

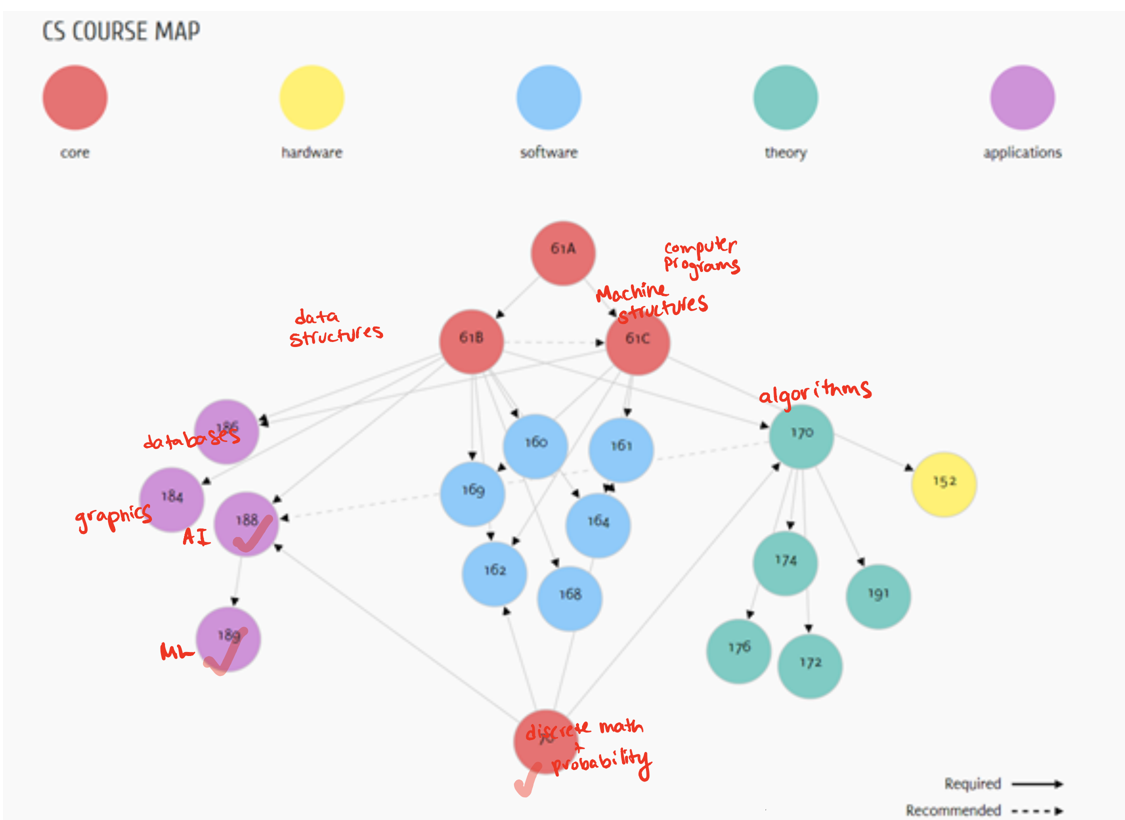
Linear Algebra: conceptual tools to model
Circuits: How to go from model to design, grounded in physical world

Intro to foundational concepts in Machine Learning

EECS course maps:



classes useful to my research



If you liked Linear Algebra, you'll love:
(M1, M3)

EECS 70: probability

EE120: signals & systems

↳ EE123 Digital Signal Processing (lab) → e.g. gold codes, correlations communications

EECS 126: graph transitions, wireless systems, using probability

↳ e.g. Page Rank, pumps ↳ like GPS

EE127: optimization theory → how to make things optimal! (mathematical)

CS 188:
CS 189: } AI/ML ← practical implementations for big data uses optimization,

EECS C106AB:
C128: } controls + Robotics ← eigenvalues!

If you liked circuits, you'll LOVE:

EE105:

EE 140:

CS 61C:

EE 130:

EE 134:

EE 137AB:

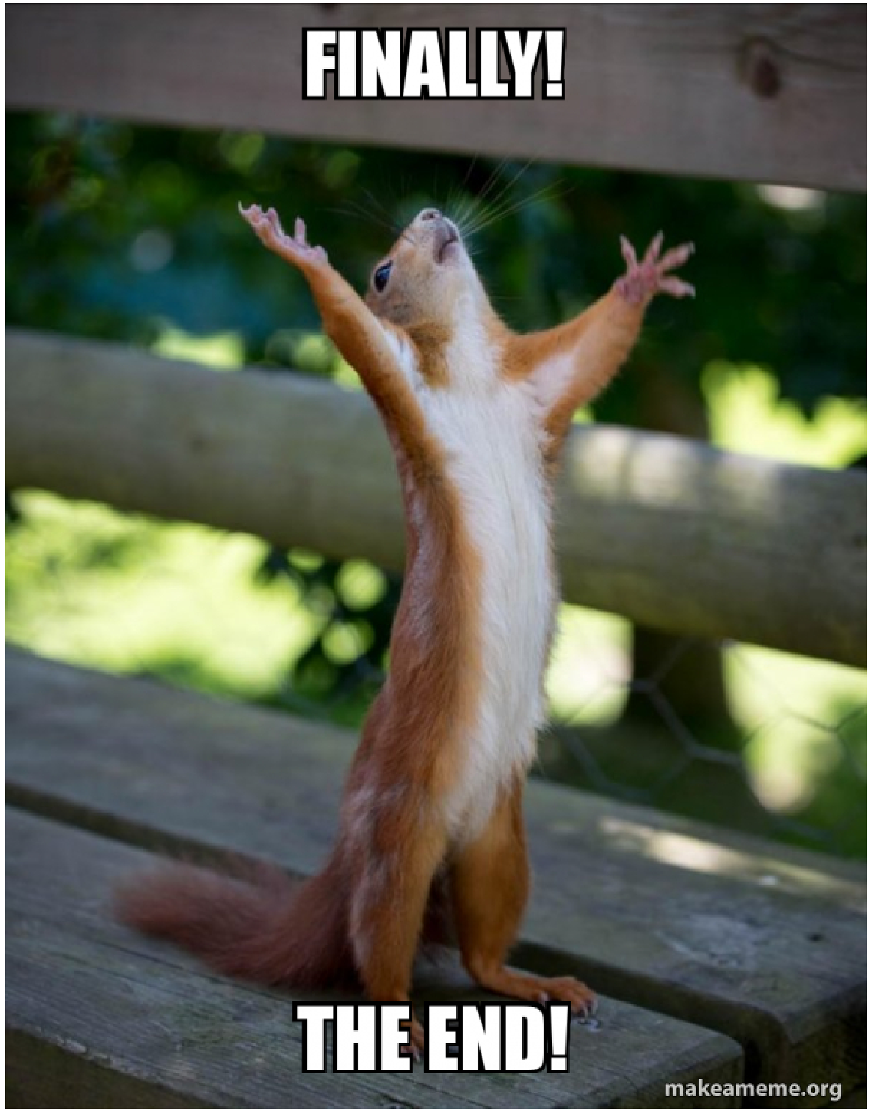
EE 143:

Physical laws behind the devices { Phys 7B:
EE117:
EE118:

↑ also, imaging!

If you liked my jokes, you'll LOVE:

↳ sorry, not teaching next year



The End

Oh, except for the final exam…