```c
#pragma omp parallel for
for (int yTile = 0; yTile < in.height(); yTile += 32)
    __m128i a, b, c, sum, avg;
    __m128i blurH[(256/8)*(32+2)]; // allocate tile blur
for (int xTile = 0; xTile < in.width(); xTile += 25)
    __m128i *blurHPtr = &blurH;
    for (int y = -1; y < 32+1; y++) {
```

### Computational Microscopy

Laura Waller  
Ted Van Duzer Associate Professor  
Electrical Engineering and Computer Sciences  
UC Berkeley
Computational imaging pipeline

Hardware design → Take picture → Crunch Data → Final result
The hard part is *integration*

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

- Atomic resolution 3D TEM
- Coded detection microscope
- Coded illumination microscope
- Neuron photostimulation
- Large-scale superresolution
- Data-driven system design
- DiffuserCam
DiffuserCam: stick a diffuser on a sensor

aperture

diffuser

sensor
Traditional cameras take direct measurements.
Computational cameras can multiplex

Need to know the forward model!
- Measure it?
- Model it?
Lenses map points to points

Point Spread Function (PSF)
Mask-based cameras multiplex

Point Spread Function (PSF)

Diffuser maps points to scatter patterns

Grace Kuo
Nick Antipa
Point spread function shifts with the object
DiffuserCam forward model is a convolution
raw sensor data → recovered scene

*solver is ADMM with TV reg in Halide*
Raw sensor data → recovered scene

*Solver is ADMM with TV reg in Halide*
raw sensor data

recovered scene

*solver is ADMM with TV reg in Halide*
El cheapo version – ScotchTapeCam!

Raspberry Pi + sensor = Scotch tape caustics

https://waller-lab.github.io/DiffuserCam/tutorial.html

Camille Biscarrat
Shreyas Parthasarathy

Reconstruction
2D → bigger FoV
Nominal Field of View

Extended Field of View
2D $\rightarrow$ 2D+time
Video from stills with rolling shutter

Raw data: 1 frame
2D \rightarrow 3D
The PSF scales with depth
3D is not so easy

Problems:
- **Calibration** (100M images?!?)
- **Computation**
- **Underdetermined**
Compressed sensing to the rescue!
solves under-determined problems via a sparsity prior

Image reconstruction is nonlinear optimization

\[ \arg \min_{x \geq 0} \| y - Ax \|_2^2 + \lambda \| \Phi x \|_1 \]

*solved with ADMM in Halide*

Towards lensless 3D microscopy

Lensless imager:
- small
- inexpensive
- enables tiling

with Adesnik Lab
Scanning Microscopy vs. Compressed Sensing

- Scanning Microscopy: speed scales with # voxels in image
- Compressed Sensing: speed scales with sparsity of sample
Confocal / SIM

Light sheet

DiffuserCam

Light field microscopy
Levoy et al., 2006

Resolution

Speed
3D neural activity tracking

Reconstructed neural activity

N. Pegard et al, Optica 2016

with Adesnik Lab
NASA: “Triple redundancy! We can’t afford to fail!”

Compressed Sensing: “Look at all this redundancy... I can fix that...”
Challenge: object-dependent resolution

Two-point resolution only predicts best case scenario.
Solution?: use condition number of sub-problem

Assume we know where non-zero elements are:
Solution?: use condition number of sub-problem

Now it is a small least squares problem
Solution?: use condition number of sub-problem

Local condition number sort of gives worst case scenario
Challenge #2: model mis-match
Solution #2: Local convolution model

Raw data

Reconstruction
Resolution: 7 um features
Miniscope version: diffuser in Fourier space

CMOS

Fluorescence filters

Gradient index lens

Random lenslets

Phase mask: Randomly spaced microlenses
Miniscope version: diffuser in Fourier space

- CMOS
- Fluorescence filters
- Gradient index lens
- Random lenslets
Inverse Problem Philosophies

Model-based
- gradient descent (FISTA, ADMM)
- Interpretable – based on physics
- Robust, guarantees on convergence
- Slow
- computationally intensive
- Model mismatch causes artifacts

Deep Learning
- Fast recon
- Higher-level tasks
- Need large training dataset
- Not interpretable
- No guarantees, not robust

CNNs, Unet, Resnet, etc.
Inverse Problem Philosophies

Model-based

- Efficient parametrization
- Uses known physics
- Learns unknowns

Physics-based learning

Deep Learning

Unrolled iterative algorithms make efficient networks

Unrolled Network

$\phi^{(n-2)}$  $\phi^{(n-1)}$  $\phi^{(n)}$  $\phi^{*}$

1st iteration  nth iteration  Nth iteration

$\Delta \phi = \mathcal{L}(\phi^{*}, \tilde{\phi})$

Acceleration Update  Gradient Update  Proximal Update

$\phi^{(n-1)}$  $\phi^{(n-2)}$  $\phi^{(n)}$

Michael Kellman
Emrah Bostan
Kristina Monakhova

Unrolled physics-based algorithm makes efficient network

\[ \mathcal{L} \]

After 1 iteration

After \( k \) iterations

Loss function

Physics-based learning improves speed + quality

Training

- dataset acquisition
  - lensed camera
  - DiffuserCam
  - screen
  - ground truth
- measurements
  - system model
  - loss function
  - reconstructions
  - backprop

Operation

- measurements
  - reconstructions
  - backprop

ML enables faster better-quality* reconstructions

input

1.5s/image

Physics-based

75ms/image

Deep Learning

10ms/image

Ours

Computers + Optics should talk more!
Reproducible = open-source + cheap + simple hardware

Computational CellScope

Quasi-dome LED array

ScotchTape Cam

www.laurawaller.com/opensource
Collaborators:
Hillel Adesnik (Neuro)
Ben Recht, Ren Ng (EECS)
David Schaffer, Lydia Sohn,
Dan Fletcher (BioE)

GigaPan: WallerLab_Berkeley
Open-source : www.laurawaller.com
Twitter: @oprickster
Github: Waller_Lab