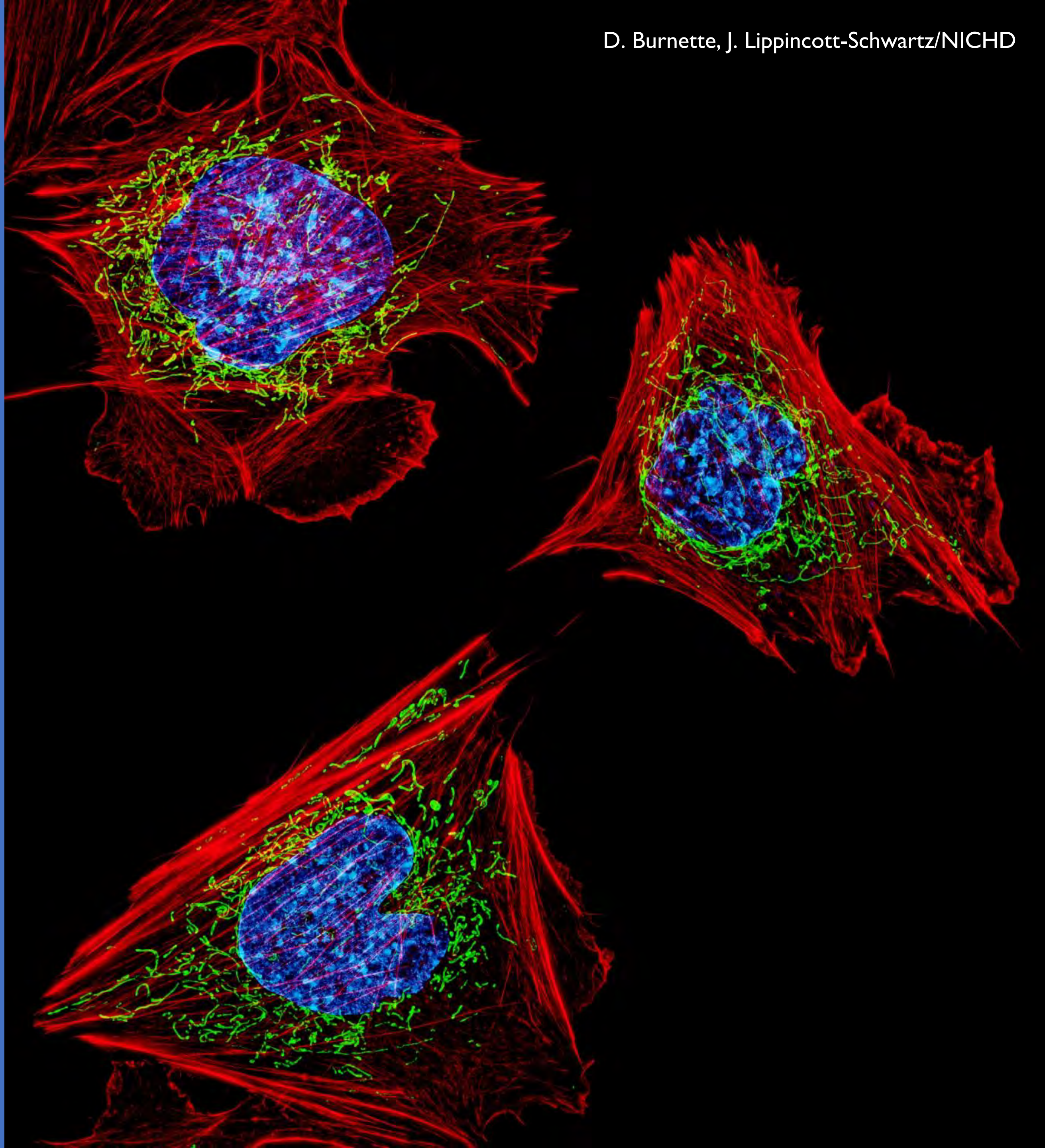
הטכניון
מכון טכנולוגי לישראל

Department of Biomedical Engineering, Technion
Computational optical imaging 336547

Tutorial 1 – Introduction to ImageJ

Elias Nehme & Yoav Shechtman

27 October 2020



Fiji is Just ImageJ

ImageJ – an open source Java-based image processing program.

Fiji – an image processing package based on **ImageJ**. Includes many useful plugins contributed by the community.

Fiji is a "batteries-included" distribution of ImageJ which facilitate scientific image analysis (life and material sciences).

Strengths of ImageJ

- **Intuitive** and easy to use.
- Can handle all **image formats**.
- Easy to **automate**.
- Bundles together many plugins into **one installation**.
- Automatically manages plugins dependencies and updating.
- Its plugin structure gives the flexibility to adapt it for **different needs**.

Plugins:

- https://imagej.net/Category:Plugins
- http://imagej.nih.gov/ij/plugins
- http://imagej.nih.gov/ij/plugins/mbf
- https://imagej.net/Cookbook
- And dozens of other lists and collections

Getting to know ImageJ

First steps

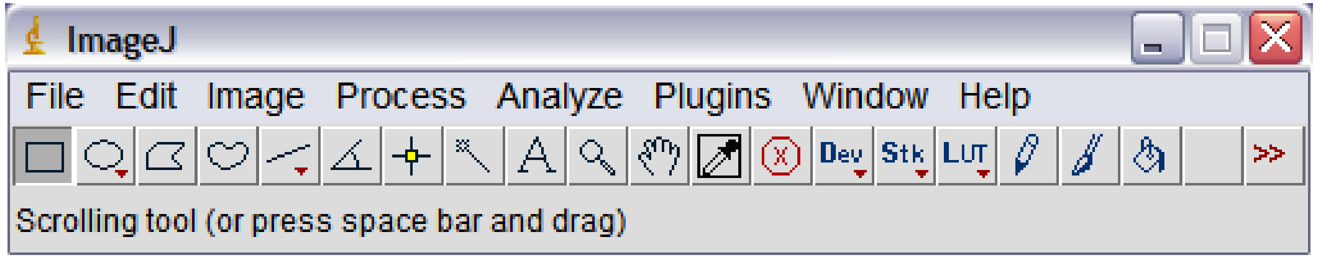
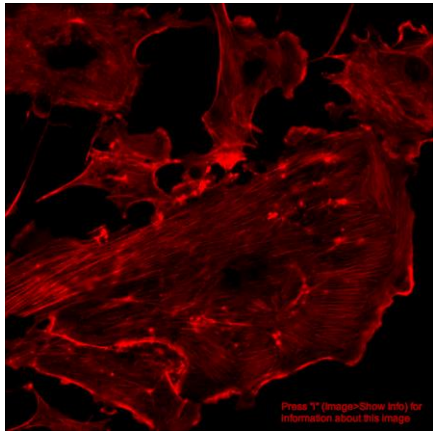
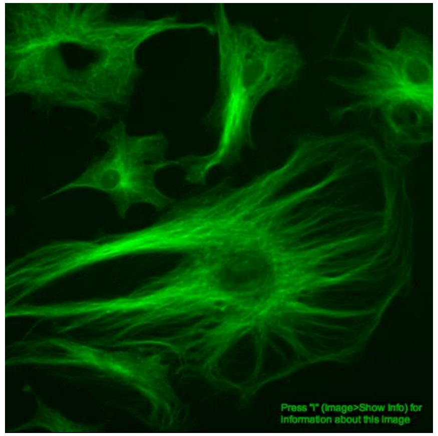
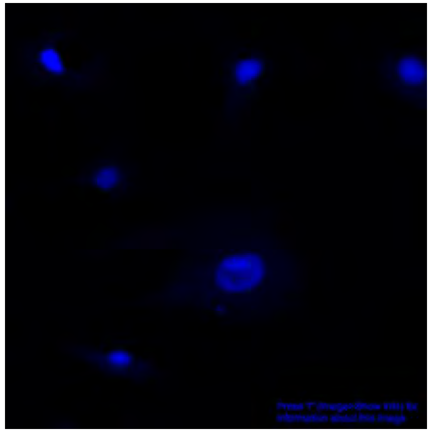


Image Processing Basics





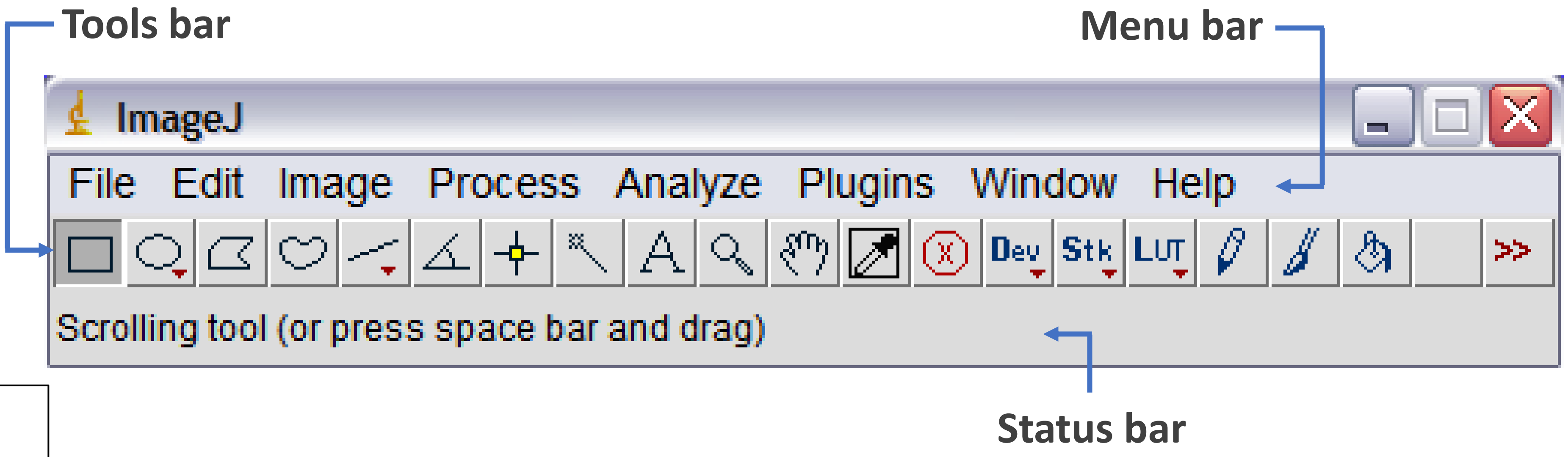


Advanced Tools - Plugins

First steps

Download: https://imagej.net/Fiji/Downloads

ImageJ main window:

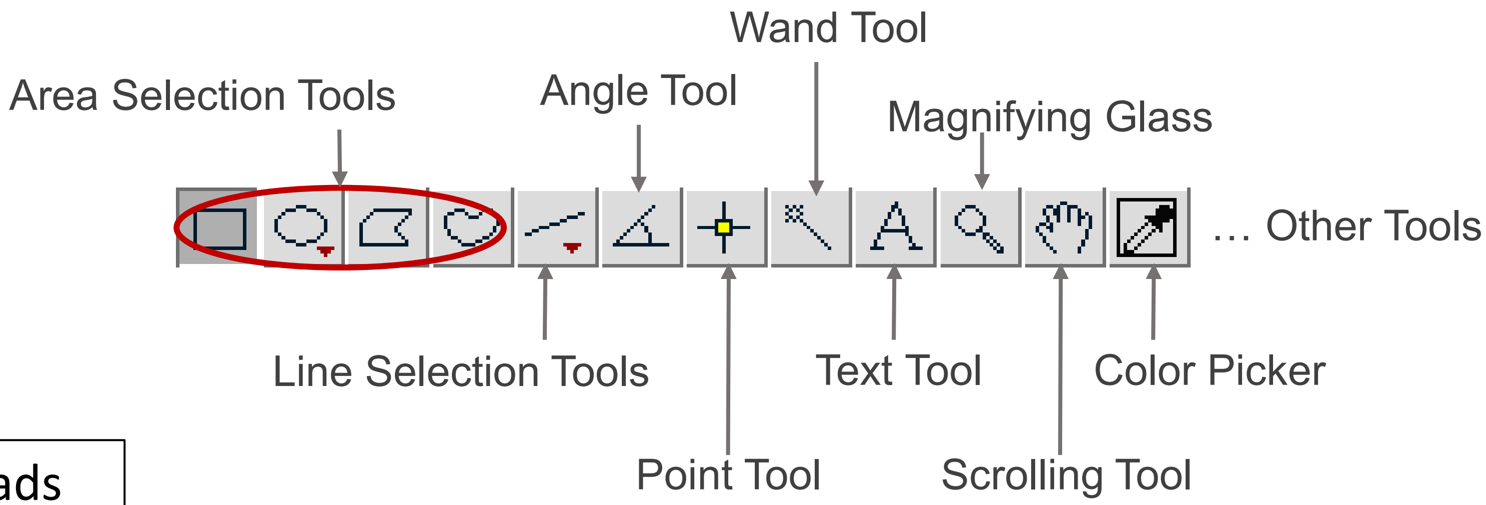


Measure (Ctrl+M)

Duplicate ROI (Ctrl+Shift+D)

Duplicate area shape to another image (Ctrl+Shift+E)

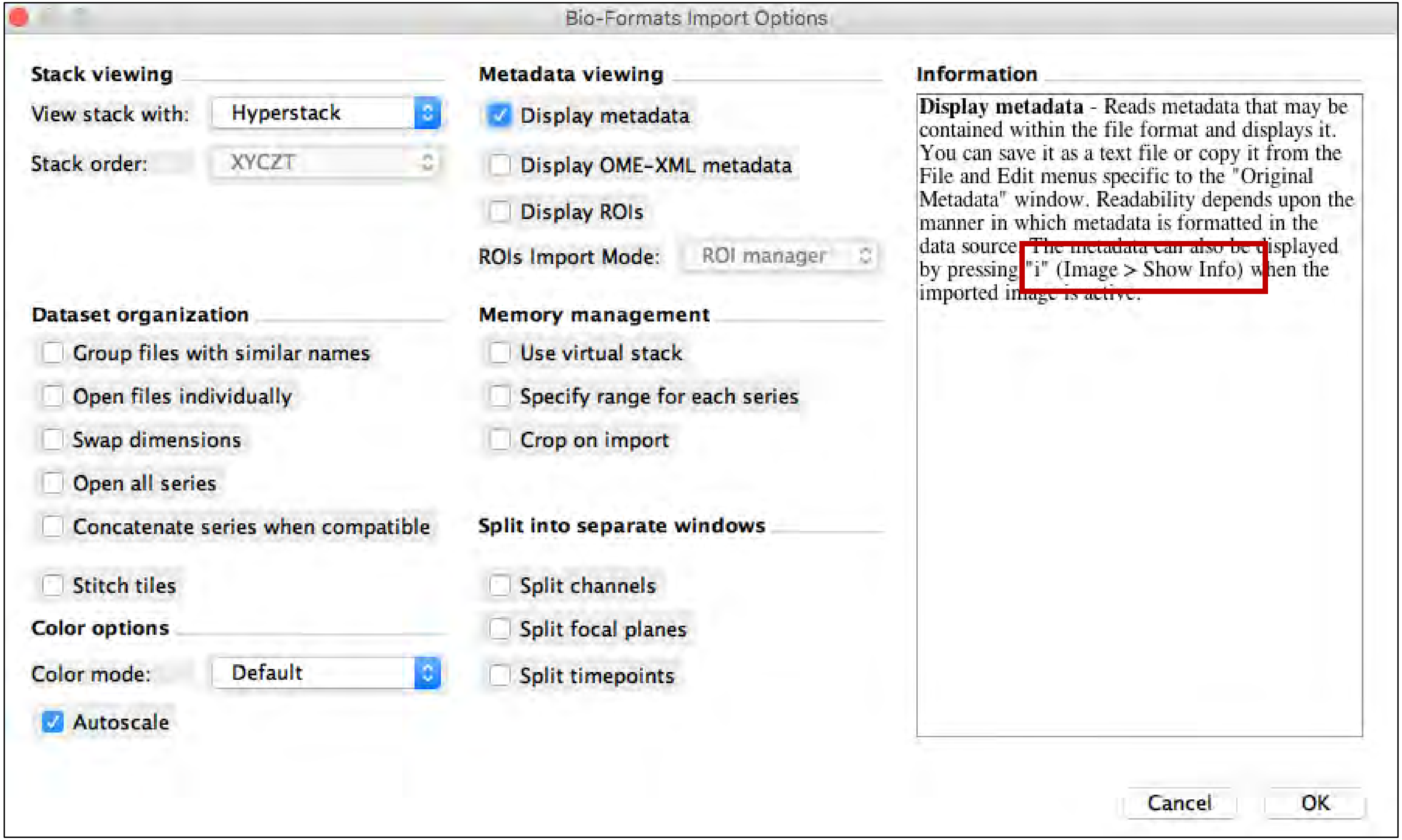
Gray value profile (Ctrl+K)



Memory management:

Edit → Options → Memory & Threads

Opening data: Drag & Drop | File → Open | File → Import → Bio-formats



Menu bar:

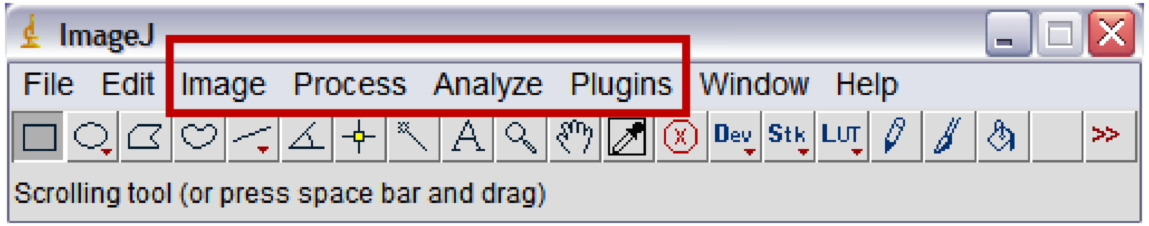


Image menu:

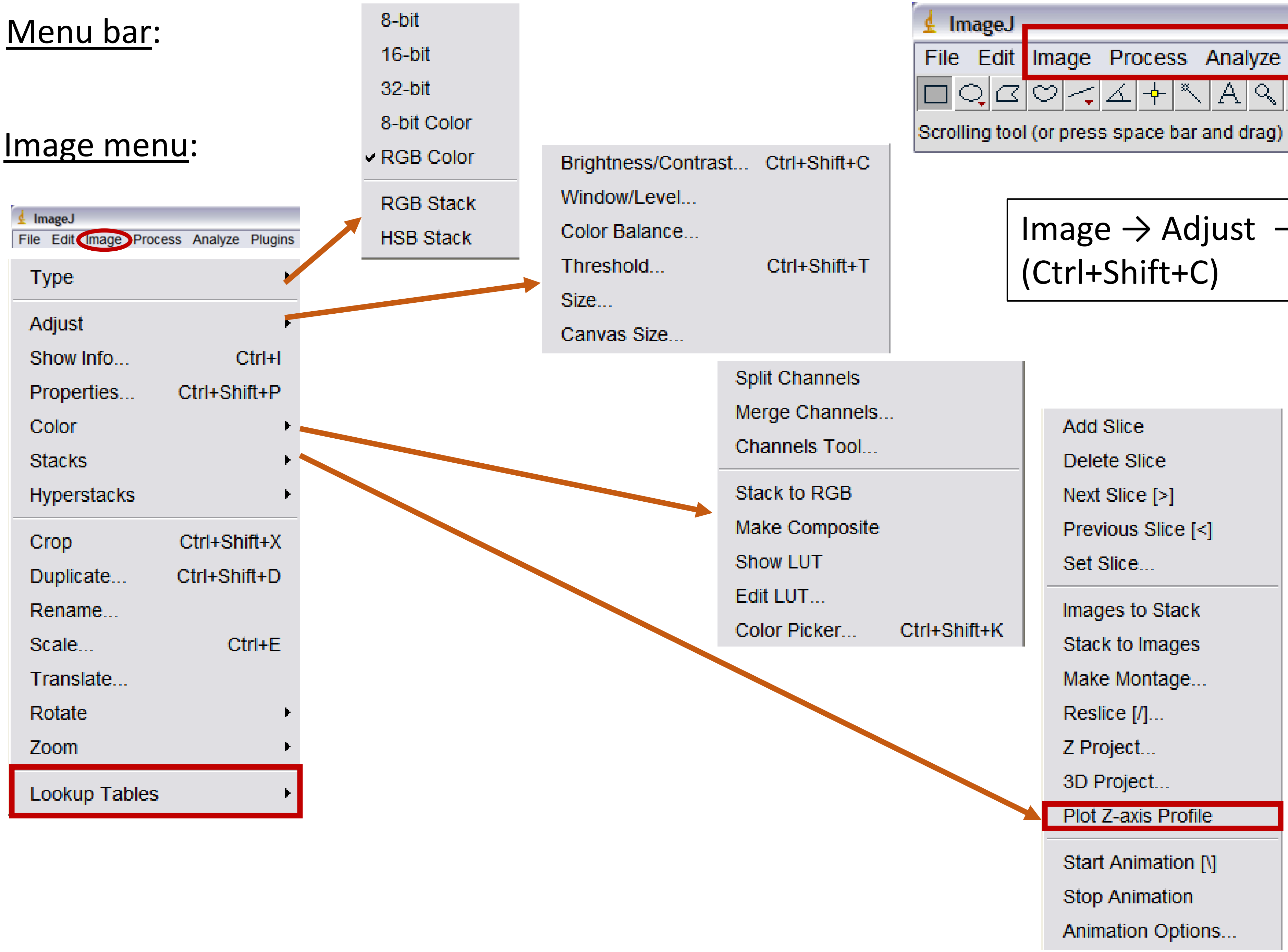
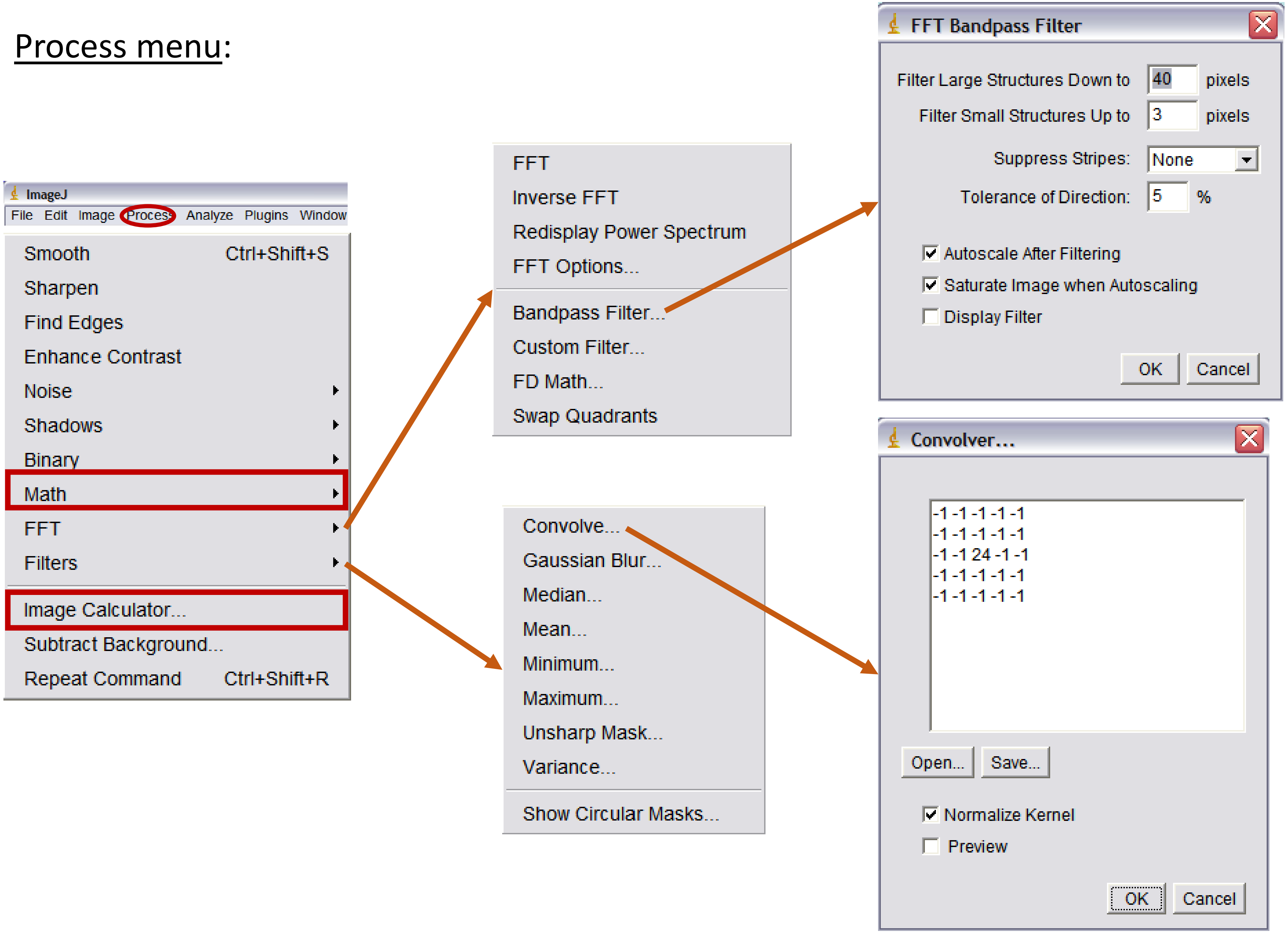
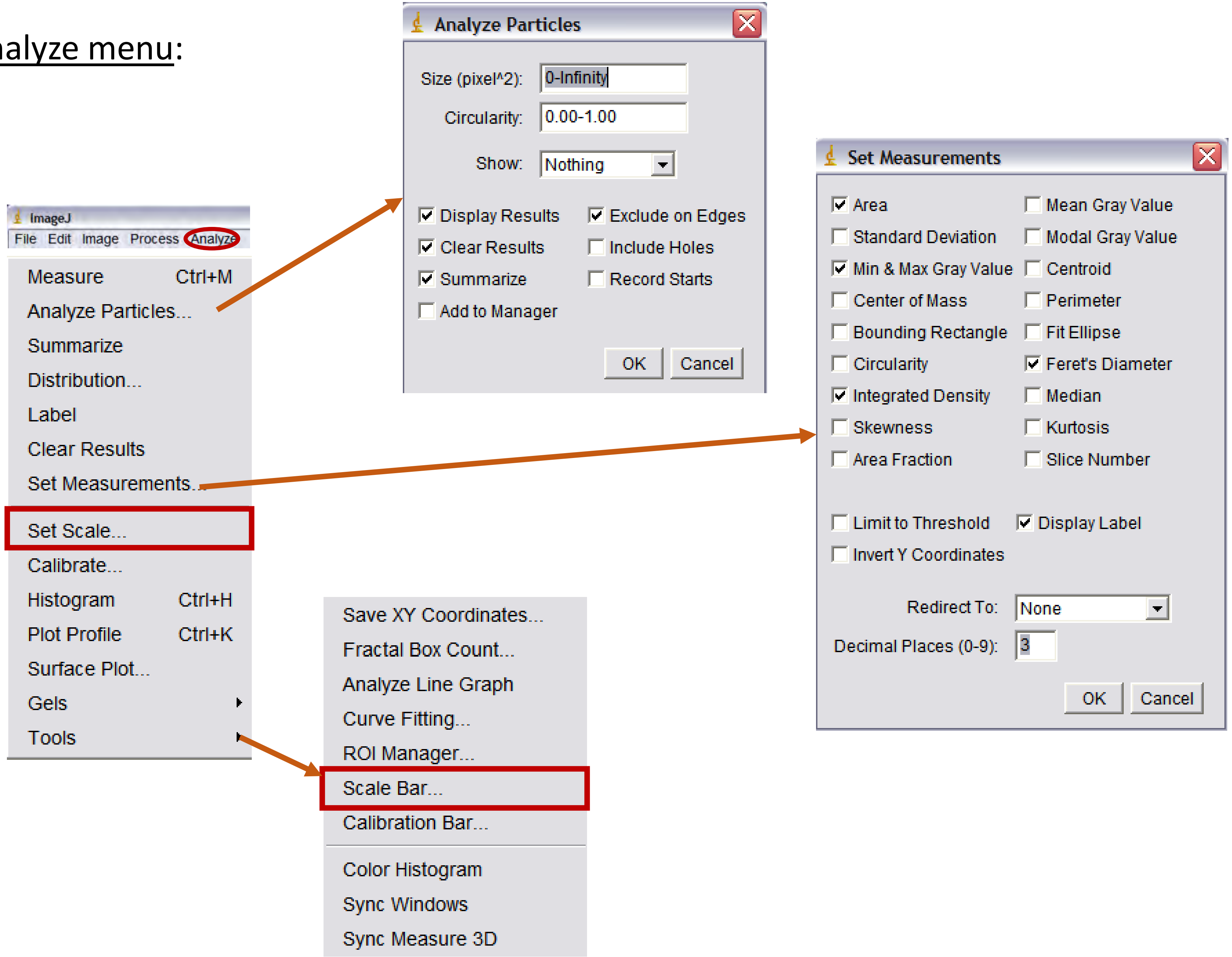


Image → Adjust → Brightness/Contrast (Ctrl+Shift+C)

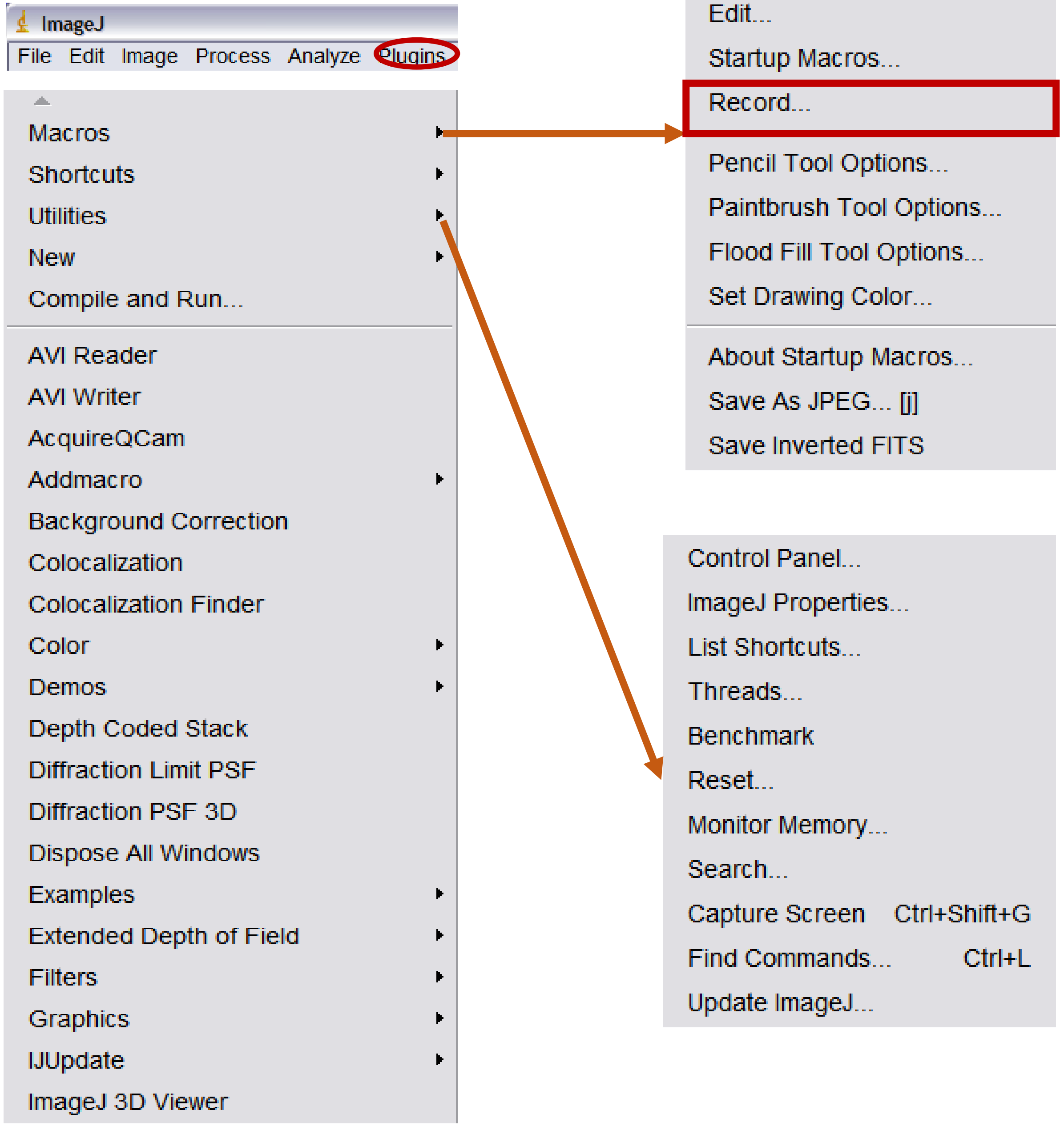
Process menu:



Analyze menu:



Plugins menu:



Getting to know ImageJ

First steps

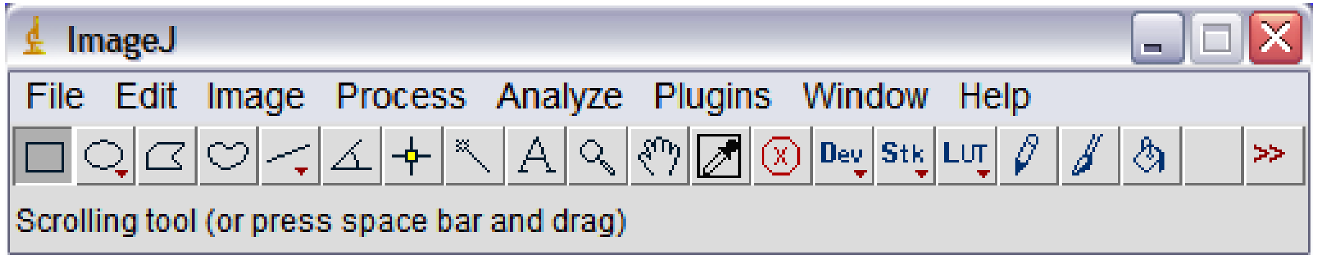
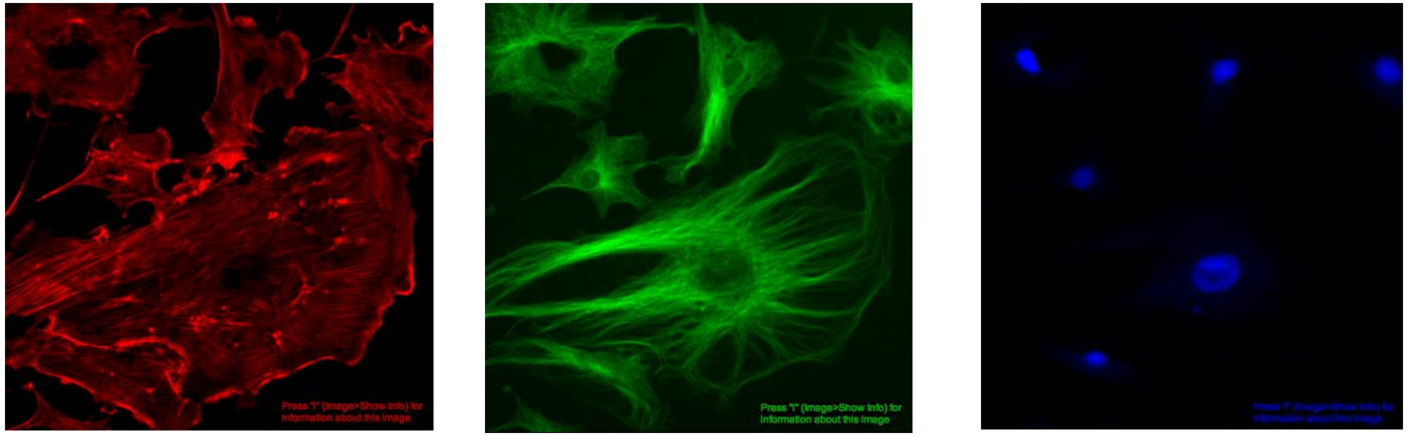


Image Processing Basics



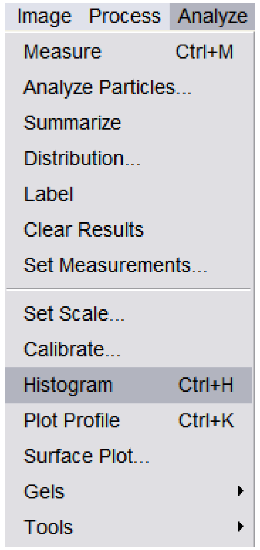
Advanced Tools - Plugins

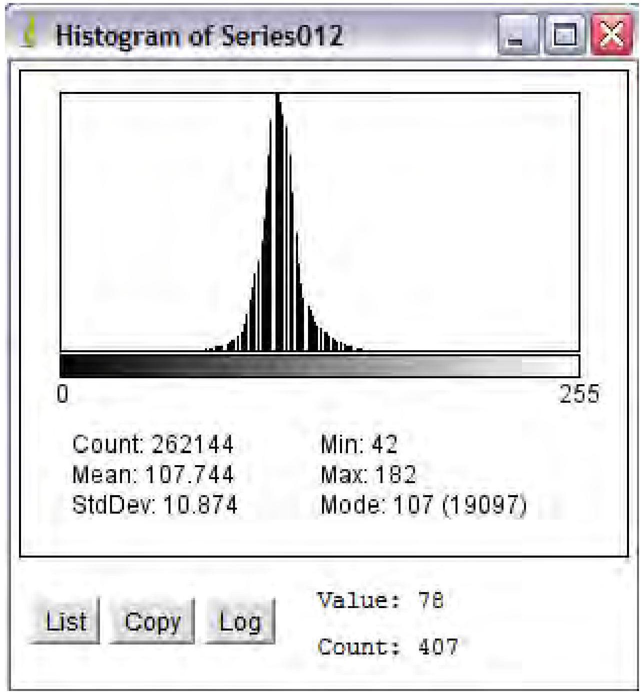
Image Processing Basics

<u>The image histogram</u>:

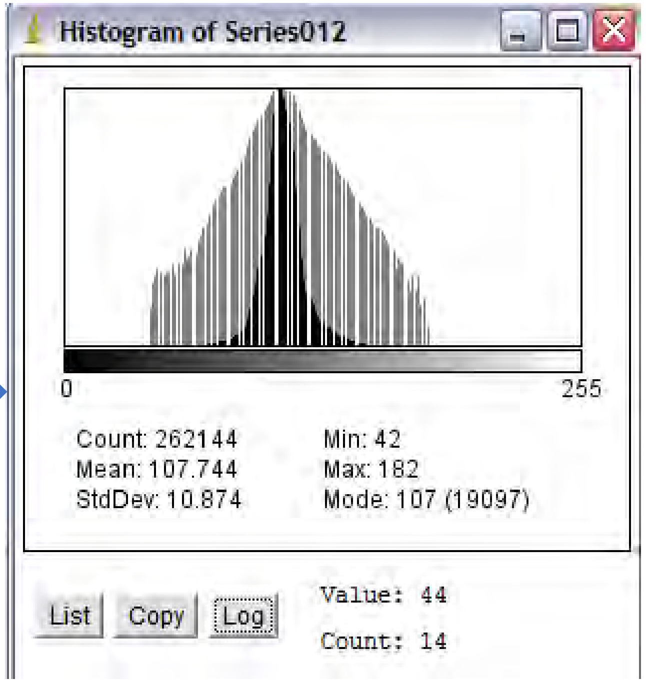
The histogram shows the number of pixels of each value, **regardless of location**.

Case 1:





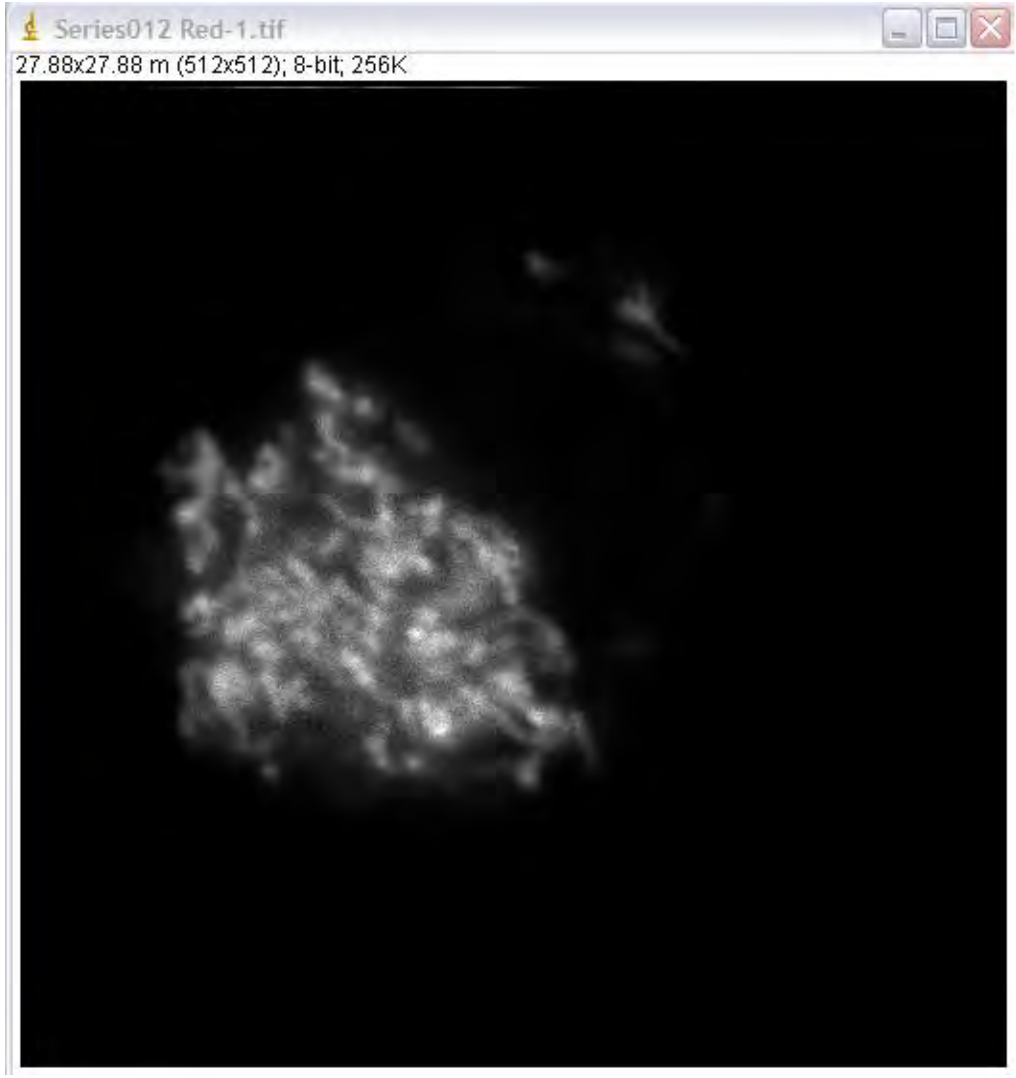
Log Scale

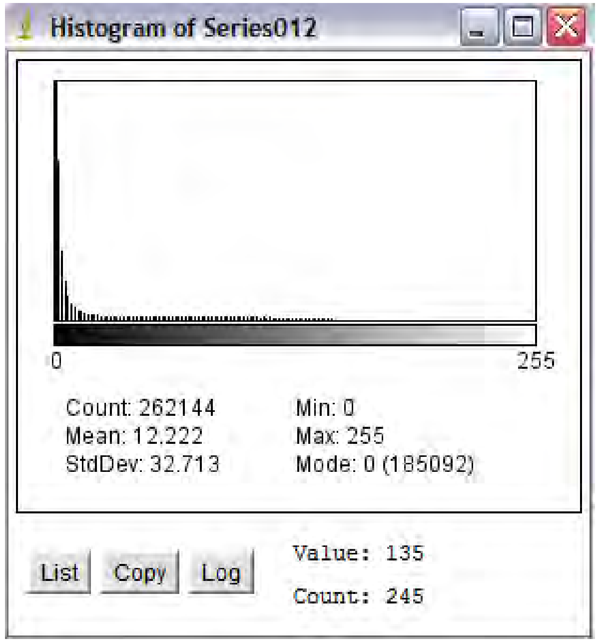


The log display allows for the visualization of **minor components**.
Note that there are **unused pixel values**.

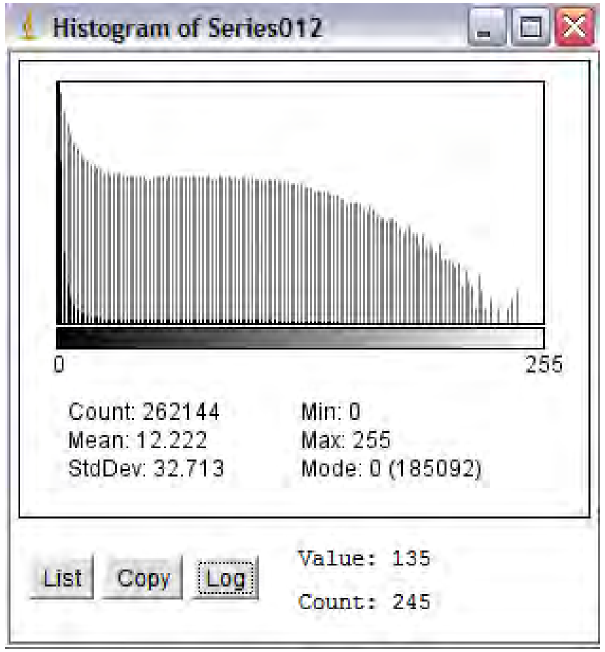
<u>The image histogram</u>:

Case 2:



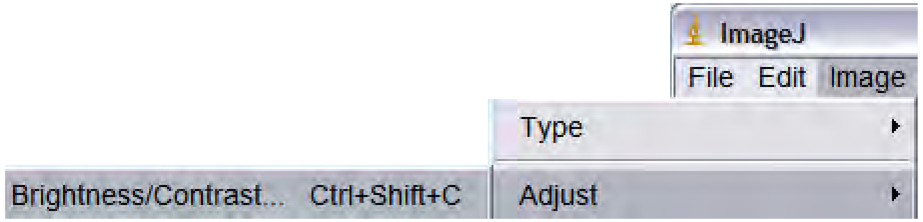


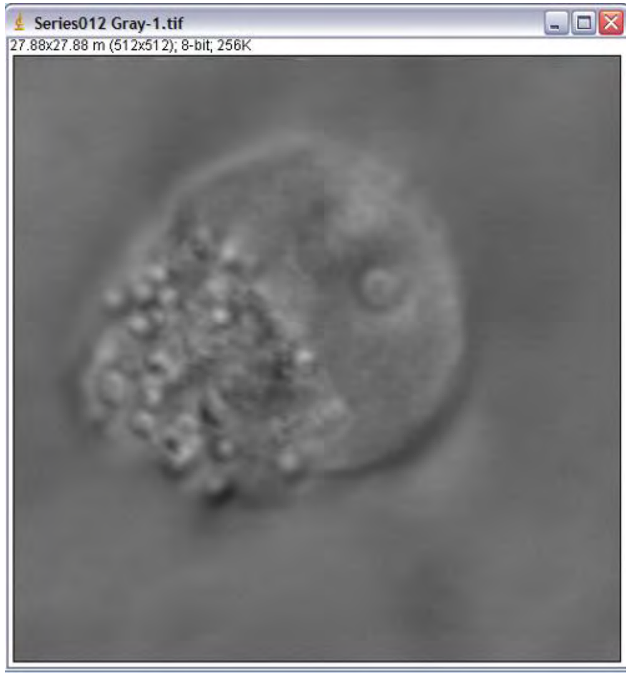
Log Scale

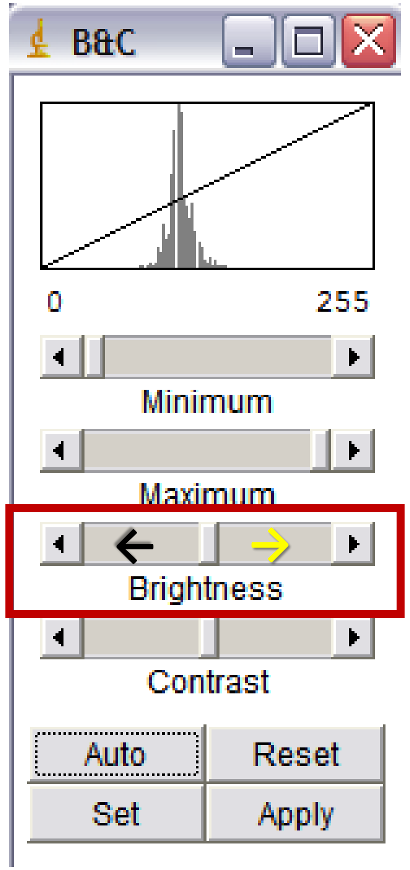


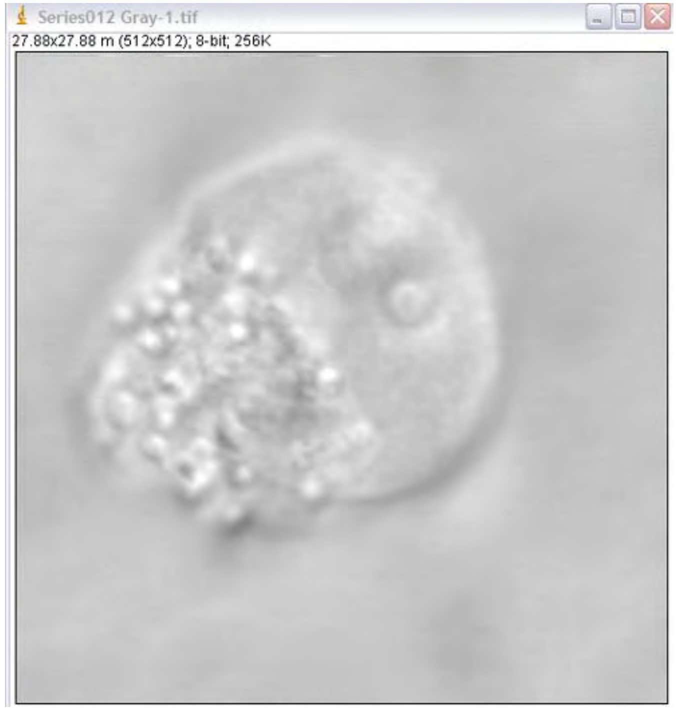
In this case, the log display indicates that **virtually all pixel values are used**, even though they are a **small percentage of the total**.

Brightness Adjustment:

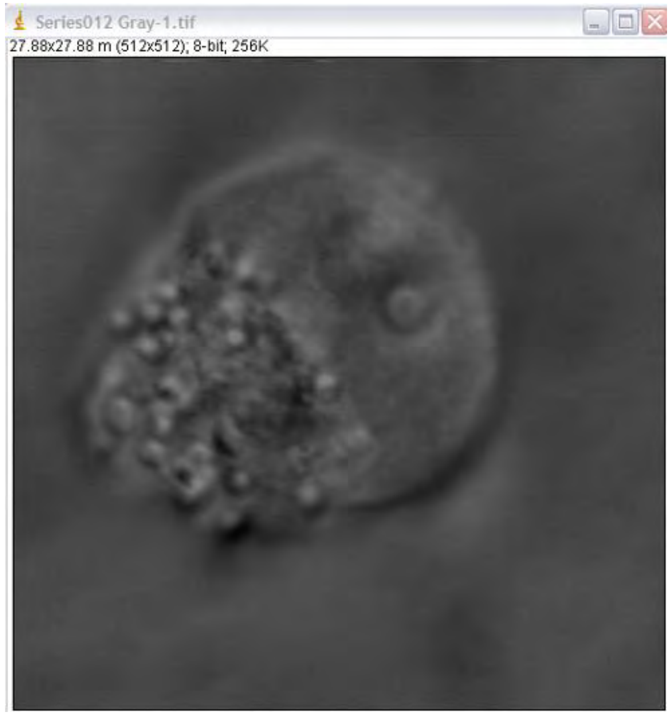






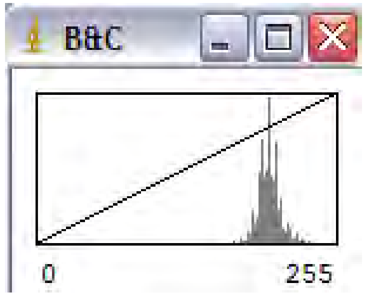


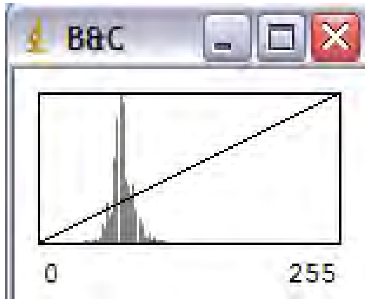
Adding a constant → Brighter



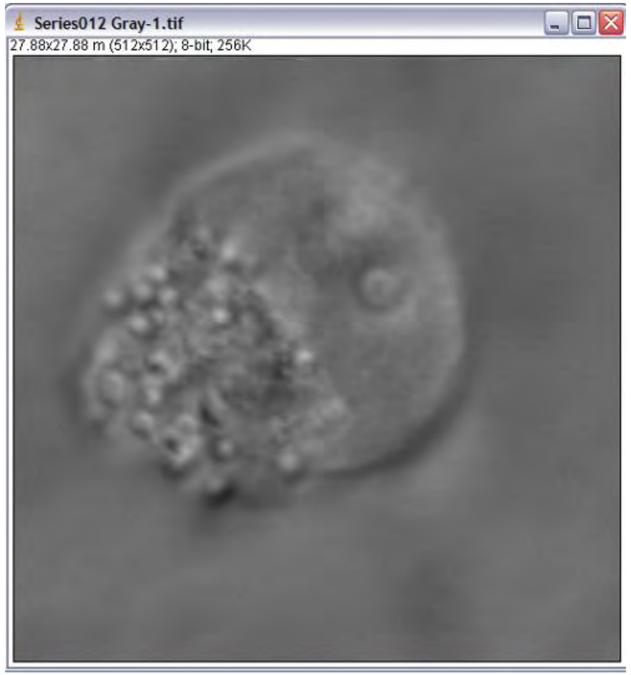
Subtracting a constant → Darker

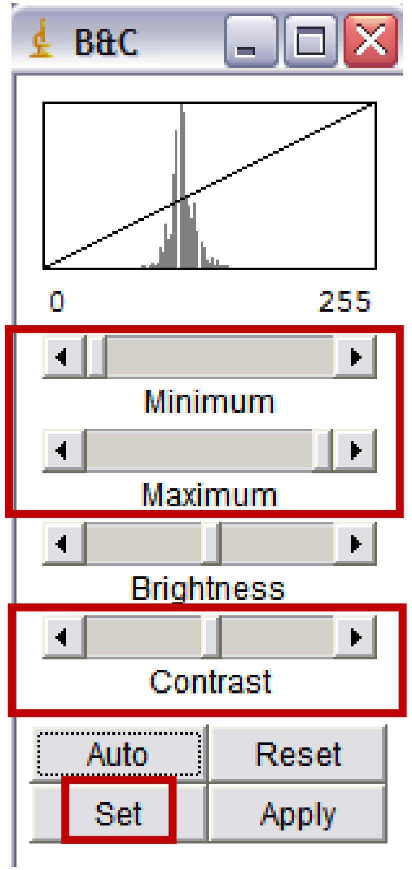
The brightness adjustment essentially **adds or subtracts a constant to every pixel**, causing a shift in the histogram along the x axis, but **no change in the distribution**.





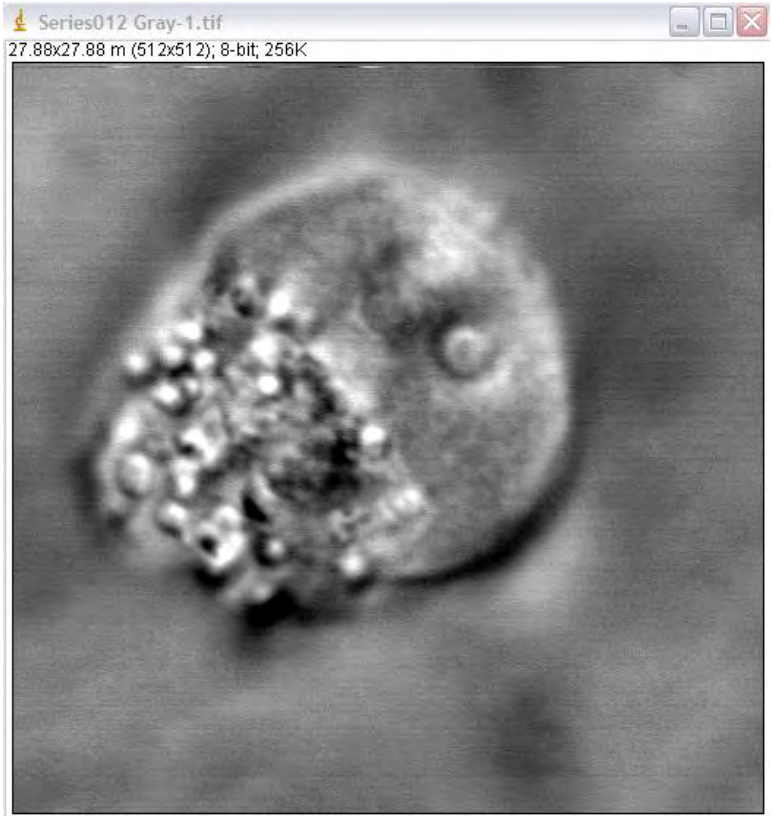
Contrast Enhancement:

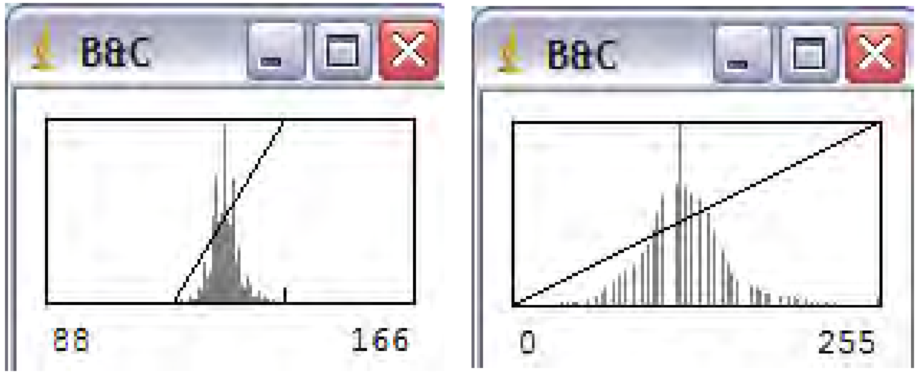




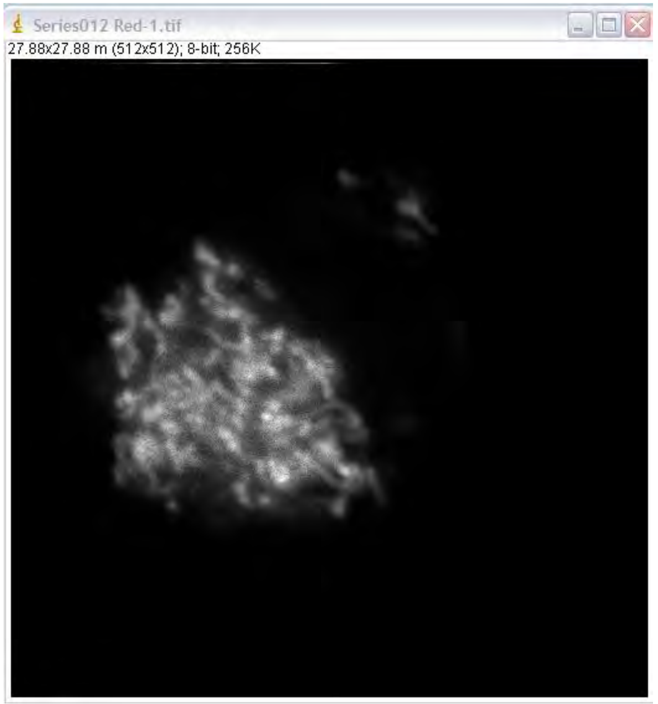
For contrast enhancement, a lower value, in this case, 88, is set at zero, and a higher value, 166, is set at 255.

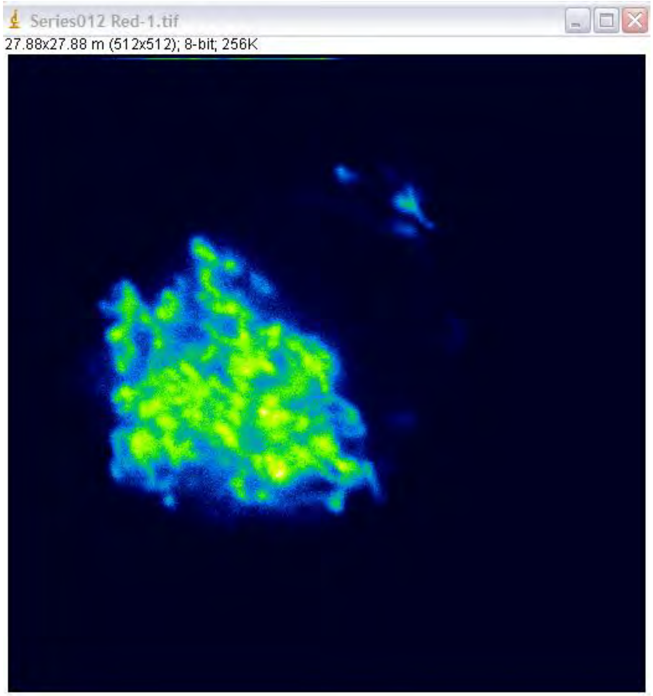
The values of each of the pixels are adjusted proportionately. Note that because of the integer values, not all pixel values are used.





LookUp Tables:





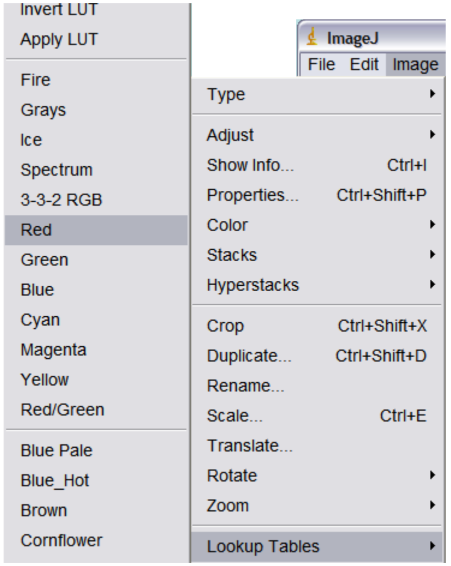
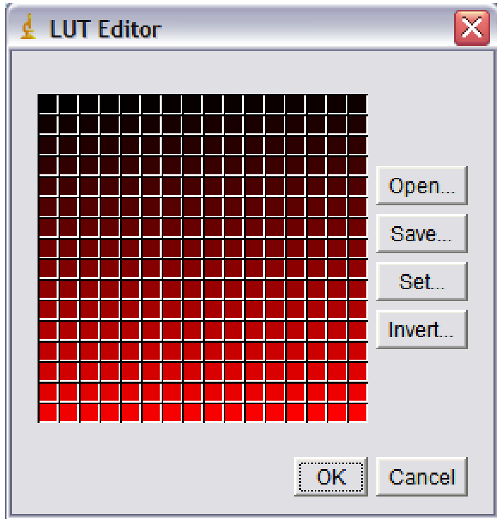
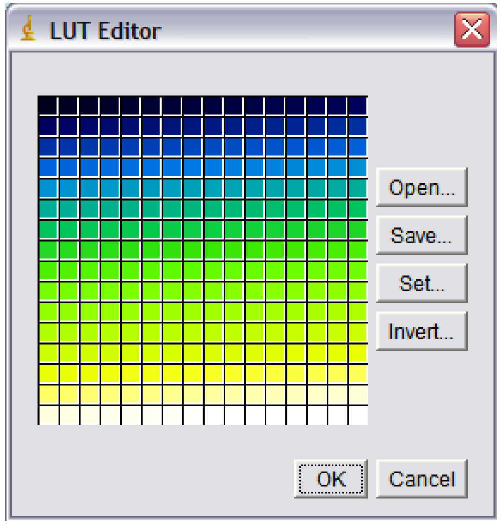


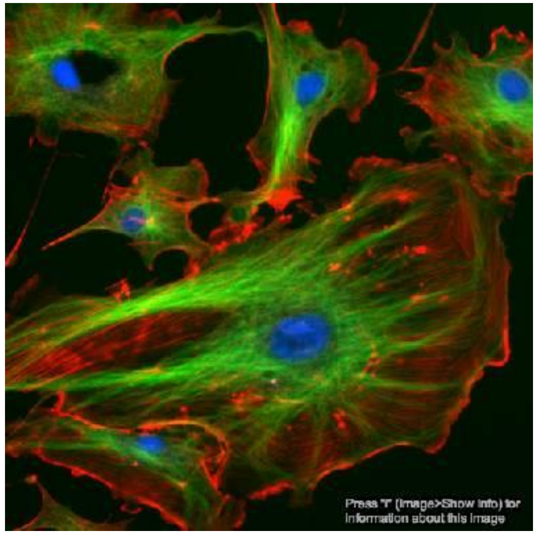
Image → Color → Edit LUT



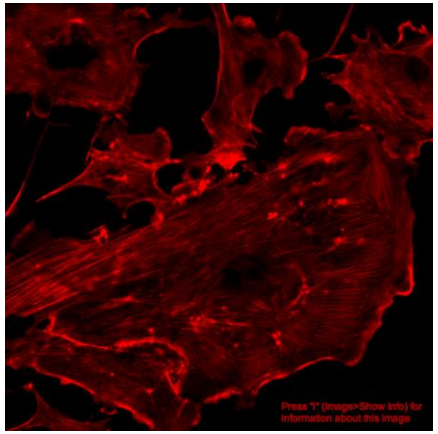


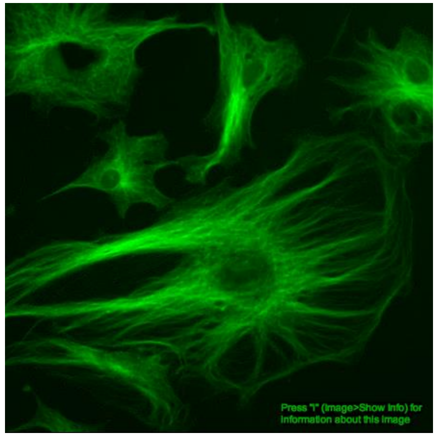
Color channels:

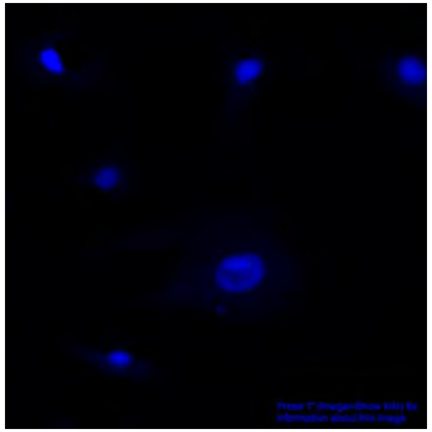
The other way to treat color is to **assign a set of 3 values**, for Red, Green and Blue to each pixel.
For common color images, each of the three colors is represented as an **8-bit value**.



One can think of a color image as consisting of **three channels**, one for each of the primary colors.

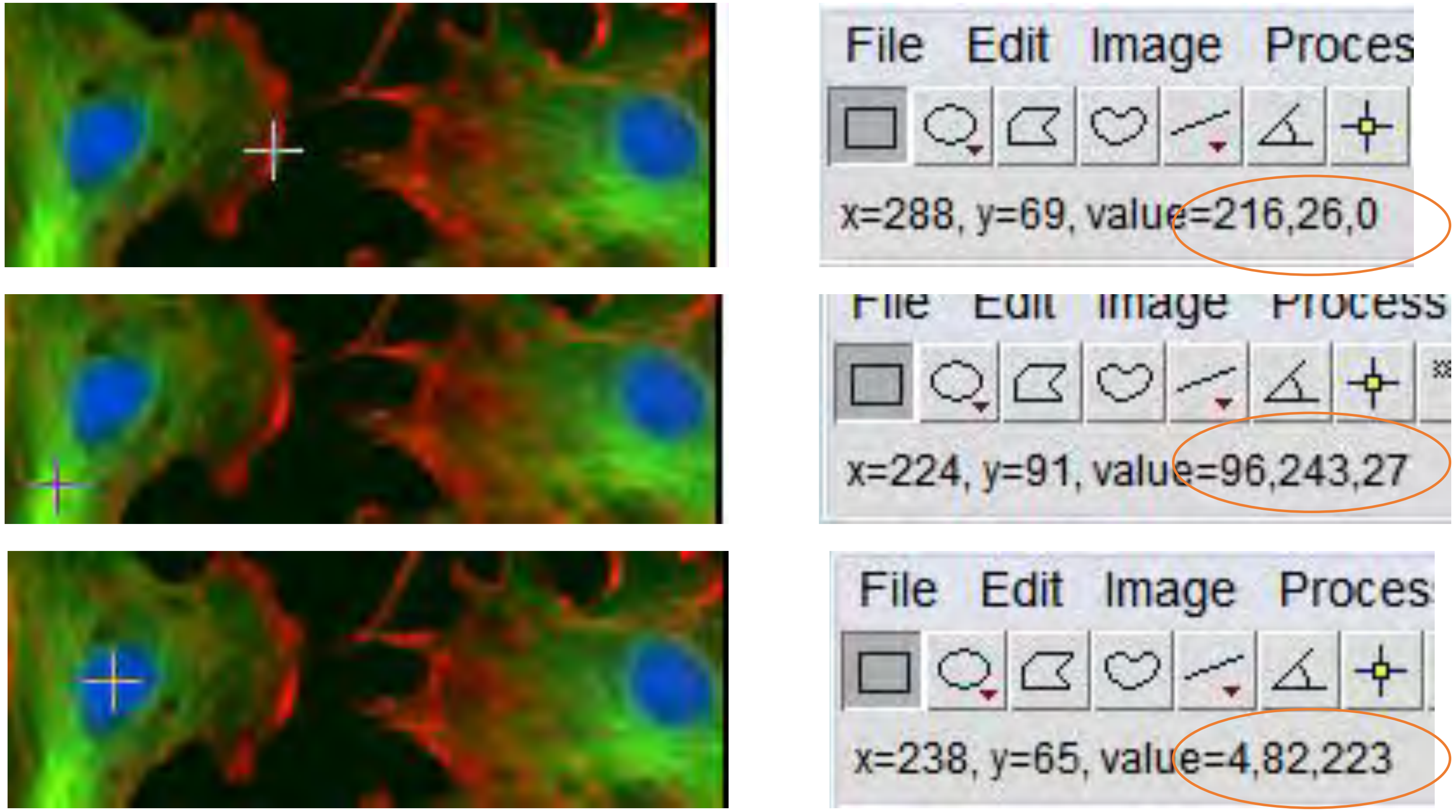


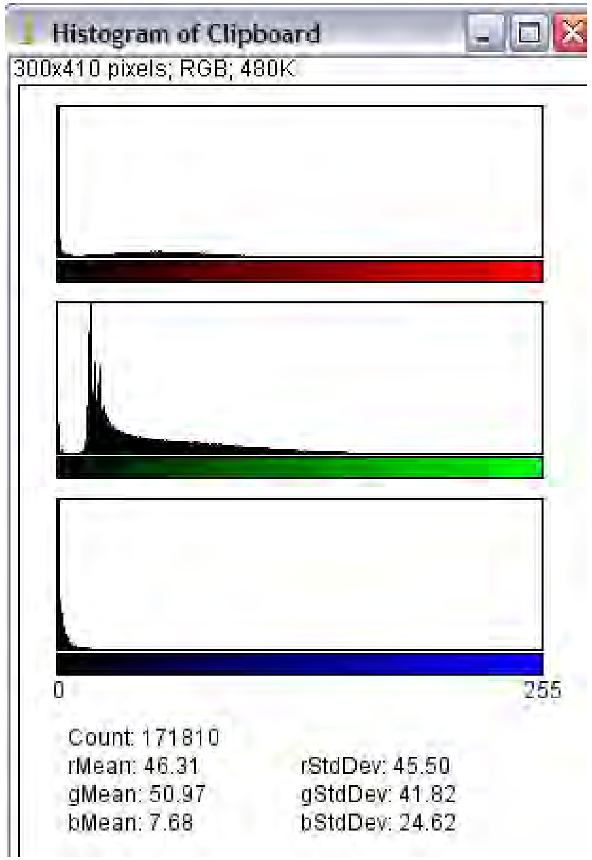




Color channels:

As we move the cursor over different parts of the image, the color values appear in the status bar of the program.





A color histogram plugin is available

Getting to know ImageJ

First steps

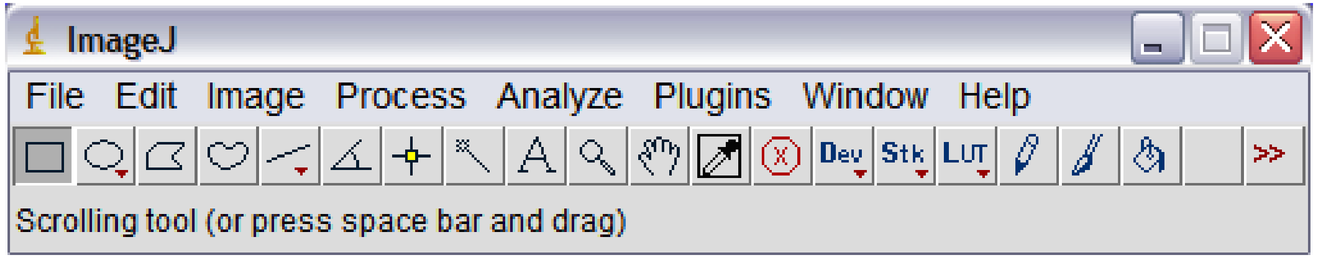
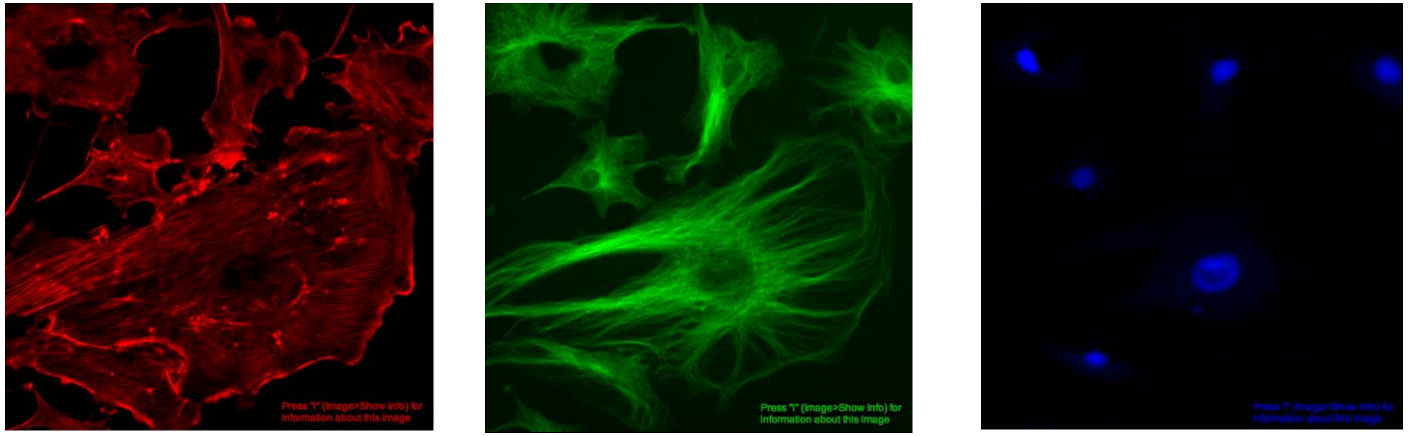


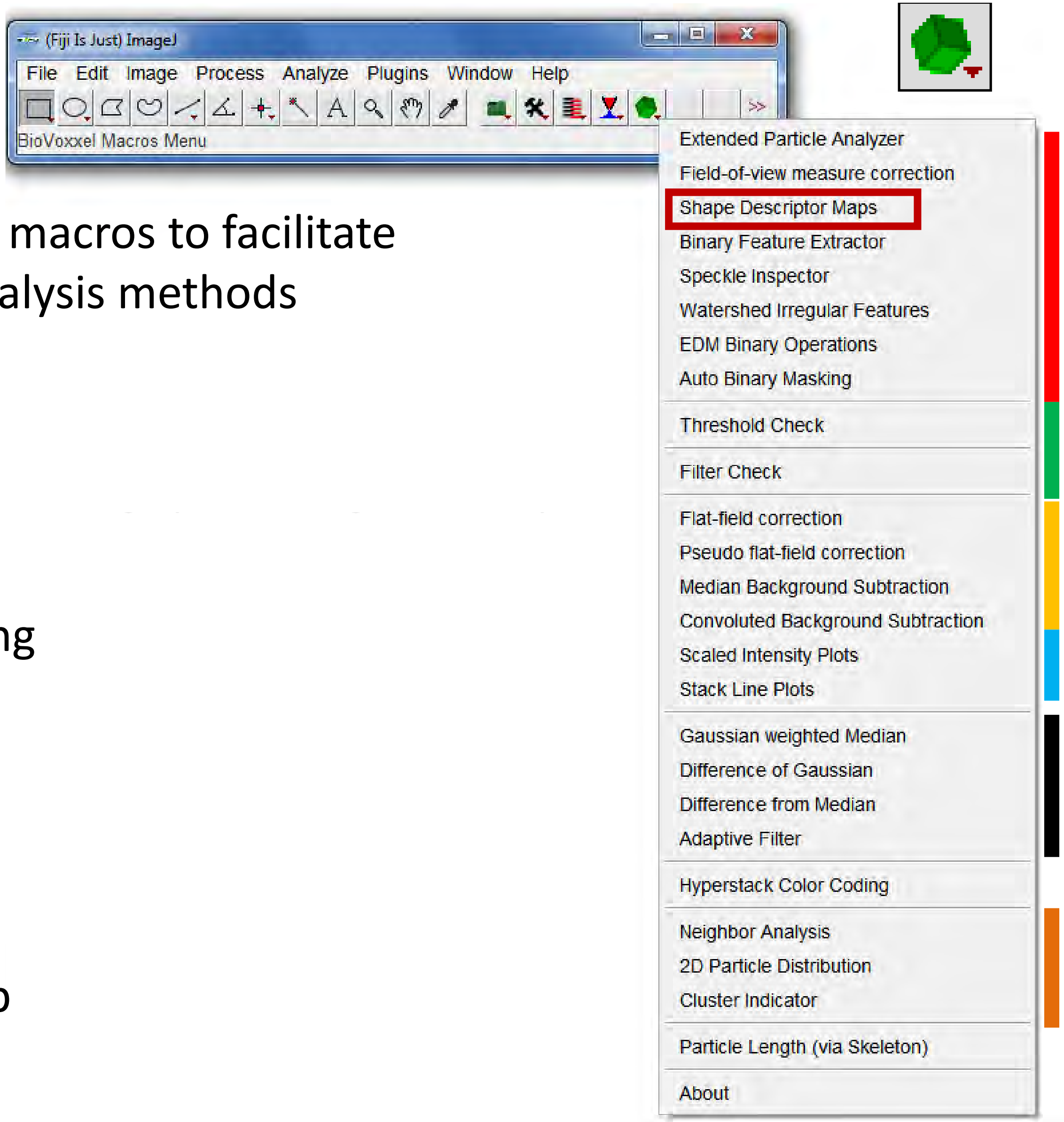
Image Processing Basics



Advanced Tools - Plugins

Advanced Tools - Plugins

BioVoxxel Toolbox:



binary operations and analysis tools

filter and threshold comparison

background and lighting correction

diverse line plots

image filters

neighbour and cluster tools

Collection of plugins and macros to facilitate image processing and analysis methods

Pre-processing
- Background filters
- Image filters

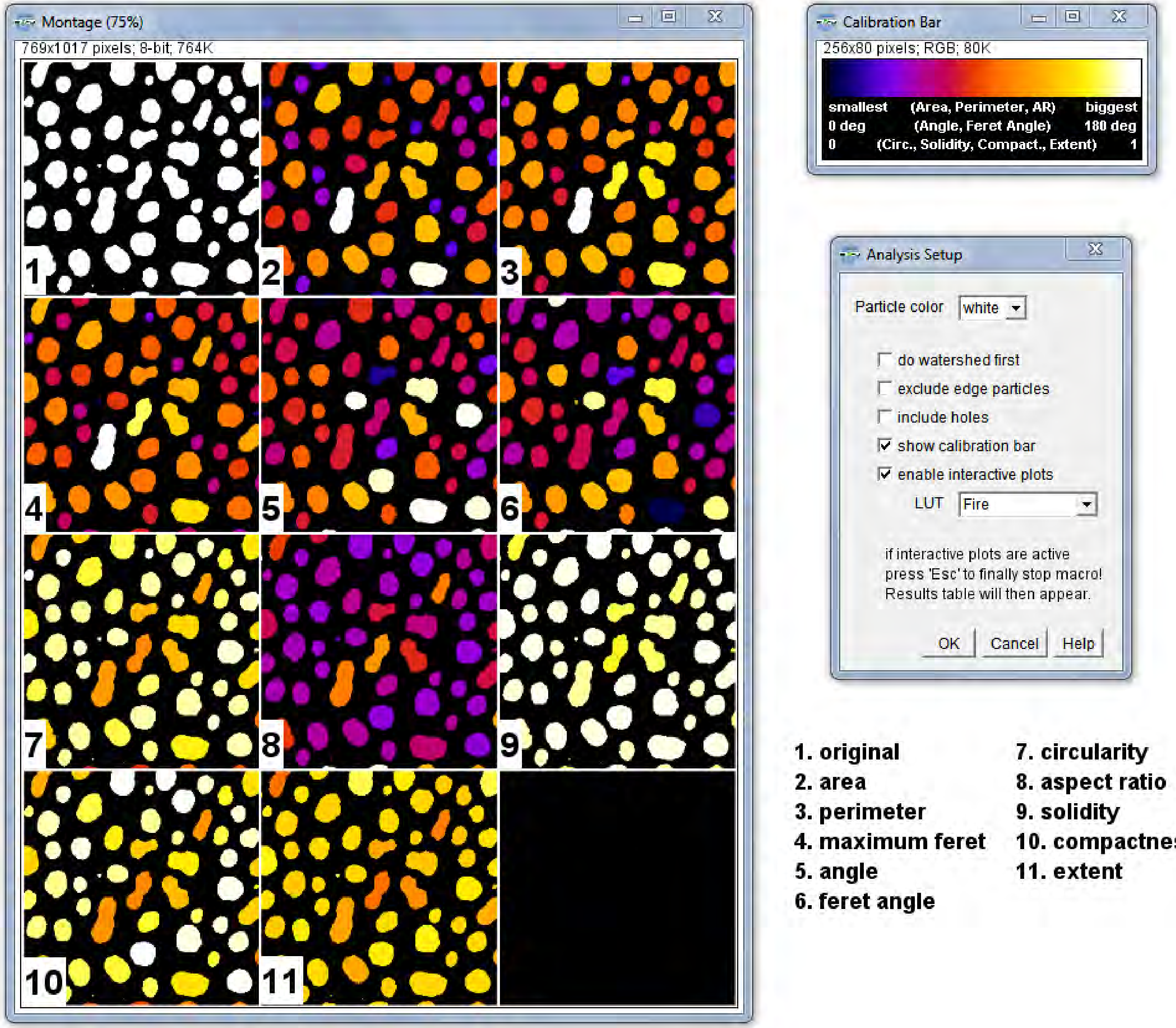
Feature Extraction
- Optimized thresholding

Post-processing
- Binary operations

Analysis
- Speckle inspector
- Particle Analyzer
- Shape Descriptor Map
- Clustering Analysis

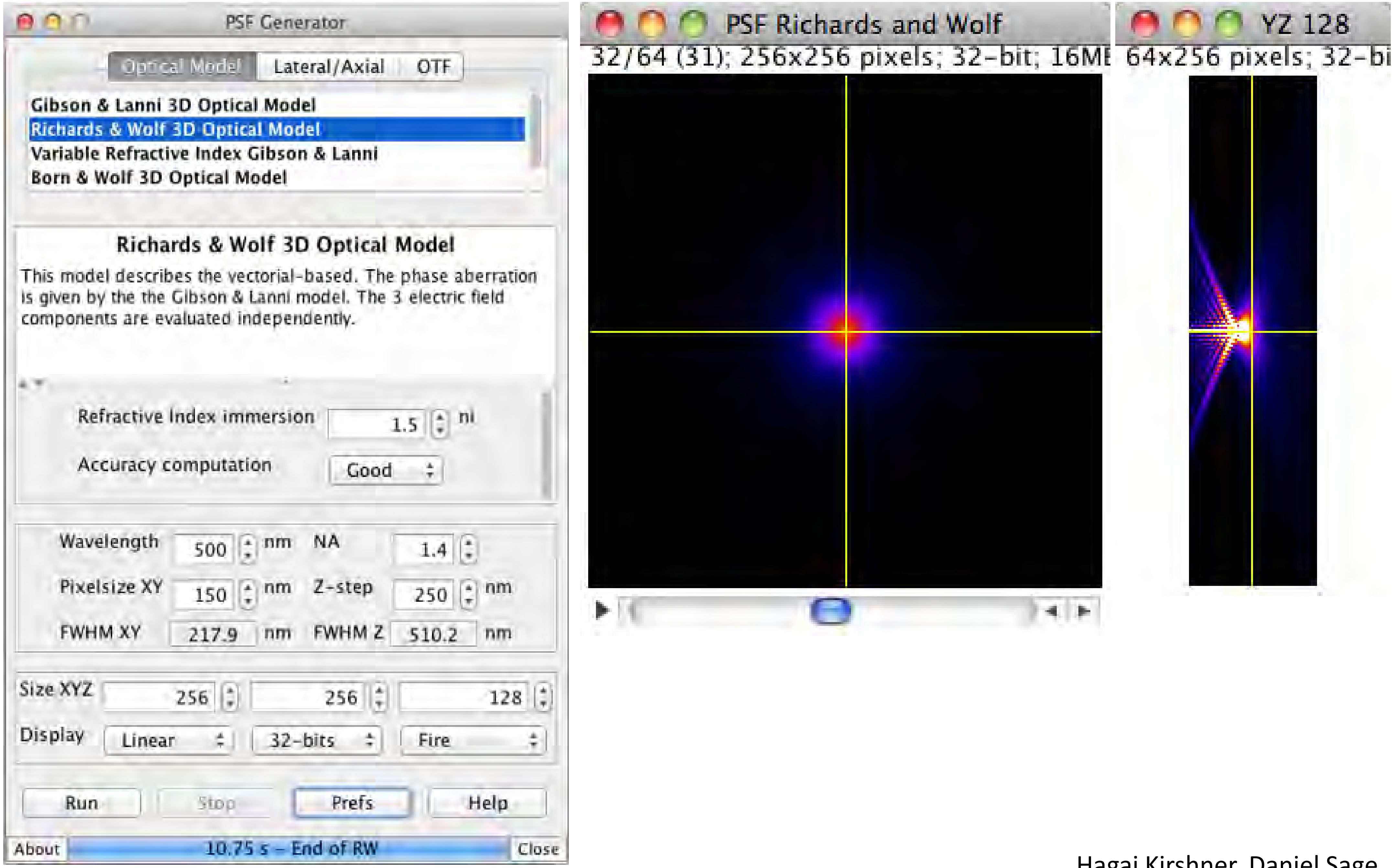
Jan Brocher, Thorsten Wagner

BioVoxxel Toolbox – Shape Descriptors:



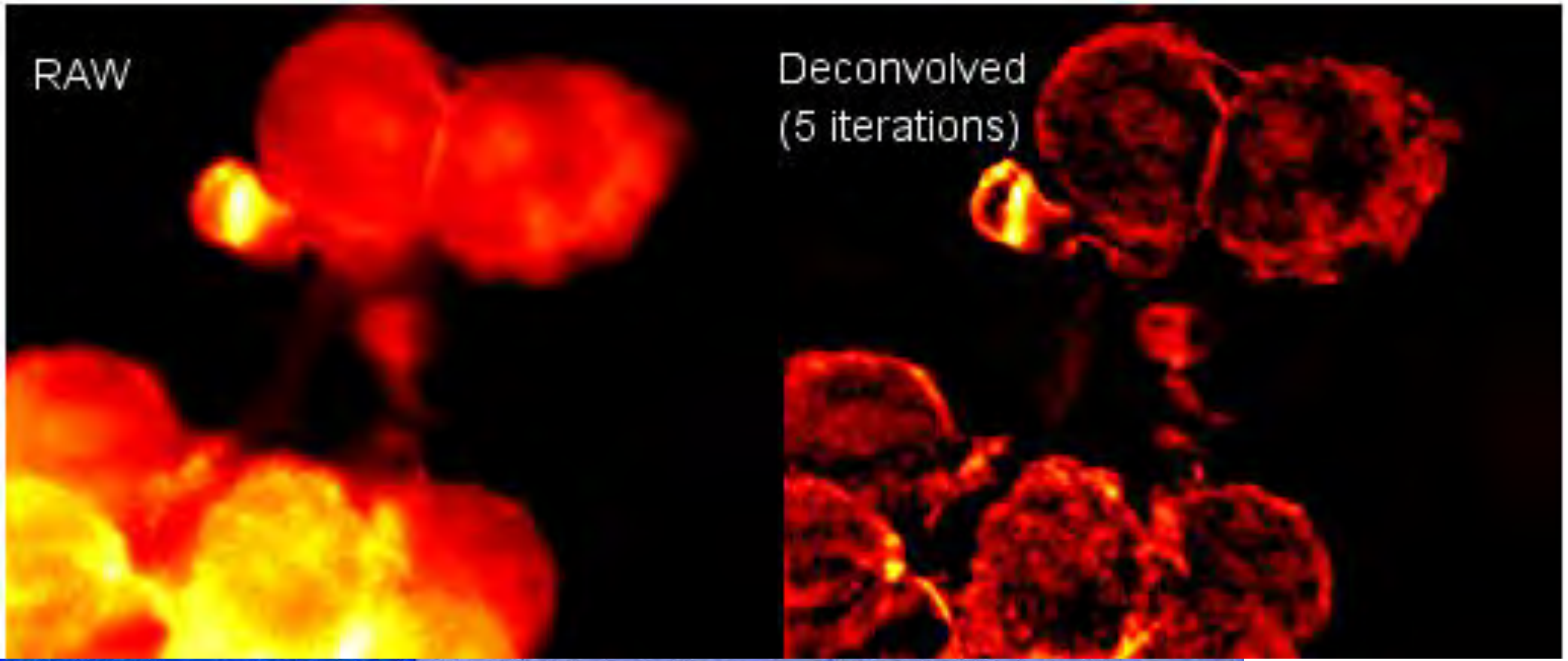
1. original
2. area
3. perimeter
4. maximum feret
5. angle
6. feret angle
7. circularity
8. aspect ratio
9. solidity
10. compactness
11. extent

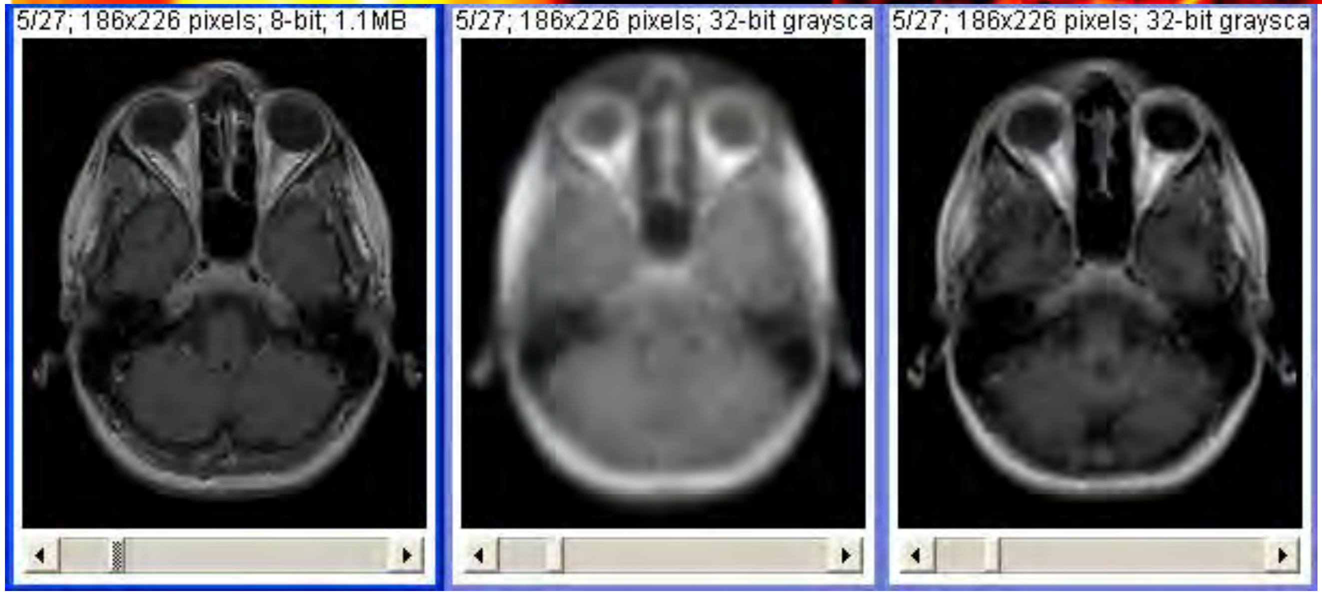
PSF Generator:



Hagai Kirshner, Daniel Sage

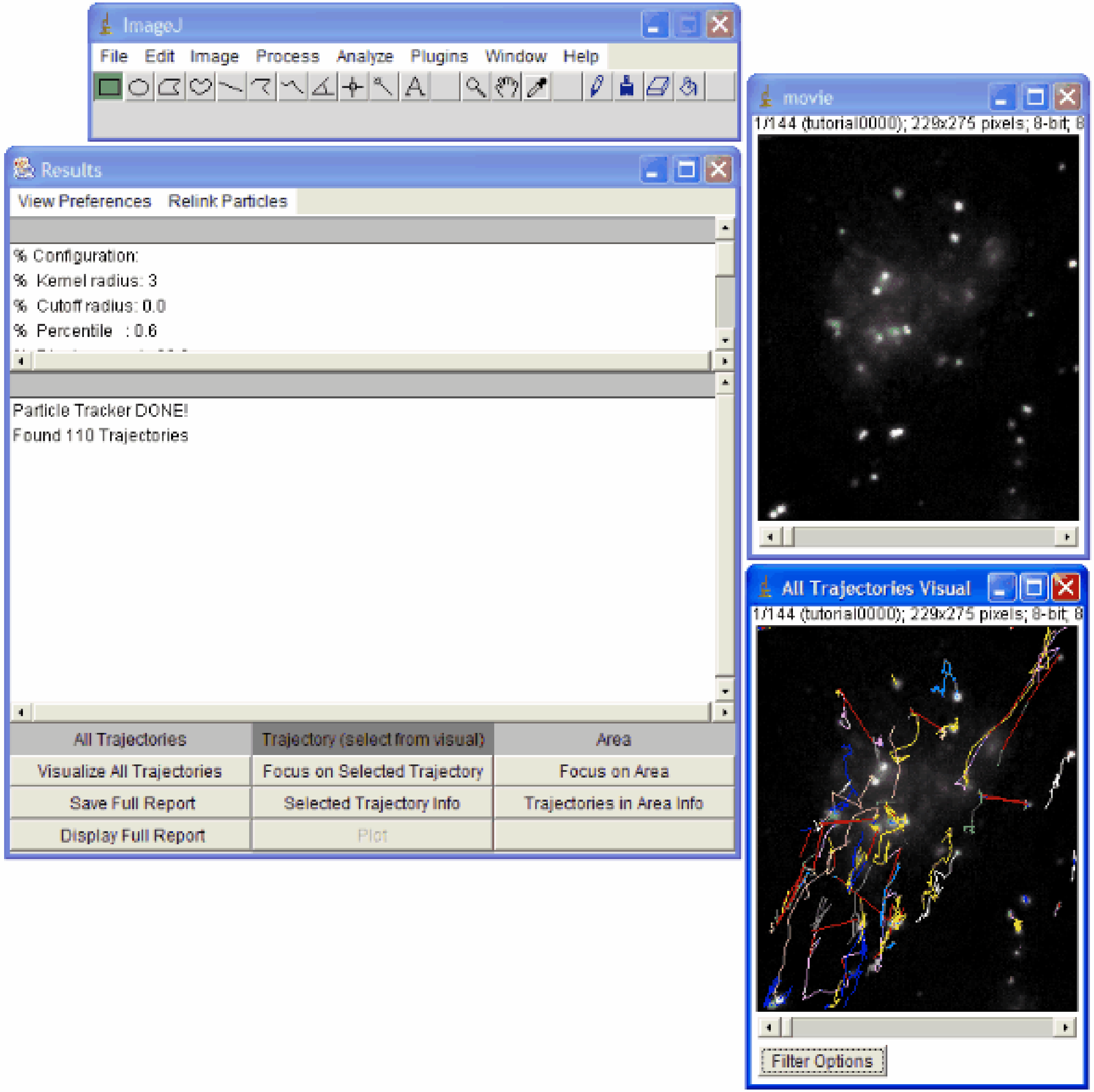
Iterative Deconvolution 3D – Cookbook:





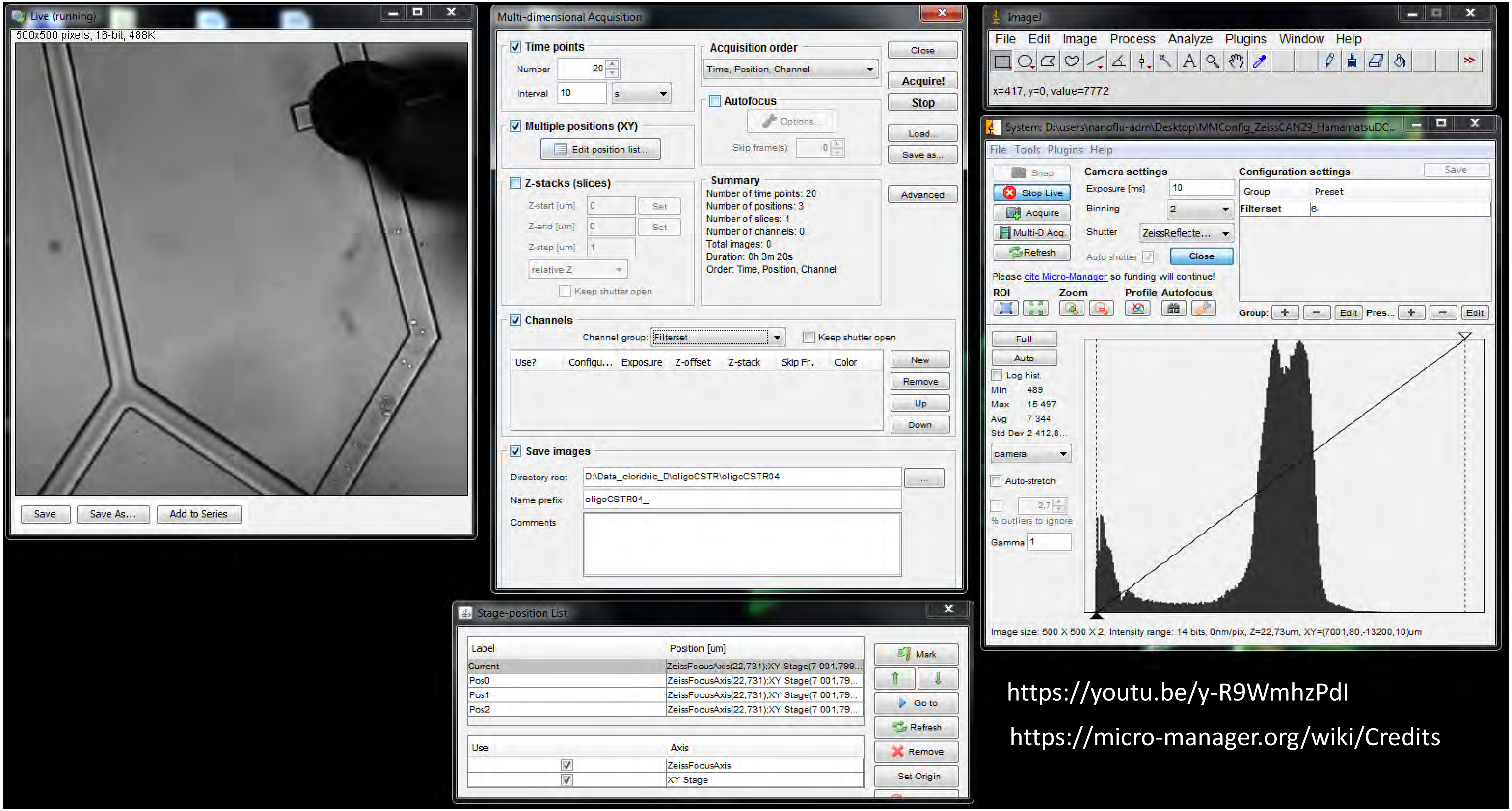
Jan Brocher, Thorsten Wagner

Particle Tracker:



I. F. Sbalzarini and P. Koumoutsakos

Micro-Manager:



https://youtu.be/y-R9WmhzPdI

https://micro-manager.org/wiki/Credits

And much more:

- Autocorrelation
- MRI t2 calculations
- Line Analyzer
- Image Correlator (image correlation)
- Particle Remover
- Circularity
- Modulation Transfer Function
- Specify ROI
- Specify Line Selection
- Comment Writer
- 16-bit Histogram
- Results and Text
- Draw line or point grids
- Moment Calculator
- Batch Statistics
- Cell Counter
- Oval Profile Plot
- Color Comparison
- Radial Profile Plot
- Microscope Scale
- MRI Analysis Calculator
- Sync Measure 3D
- Hough Circles
- Convex Hull, Circularity, Roundness
- Fractal Dimension and Lacunarity
- Measure And Label
- Colocalization
- Granulometry
- Texture Analysis
- Named Measurements
- Cell Outliner
- Grid Cycloid Arc
- RGB Profiler
- Colocalization Finder
- Spectrum Extractor
- Contact Angle
- RG2B Colocalization
- Color Profiler
- Hull and Circle
- MR Urography
- Template Matching
- Extract IMT from ultrasound images
- ITCN (Image-based Tool for Counting Nuclei)
- Multi Cell Outliner
- FRETcalc - FRET by acceptor photobleaching
- JACoP (Just Another Colocalization Plugin)
- FRET and Colocalization Analyzer
- CASA (Computer Assisted Sperm Analyzer)
- Radial Profile Plot Extended
- Concentric Circles (non-destructive overlay)
- Azimuthal Average
- Slanted Edge Modulation Transfer Function
- Calculate 3D Noise
- FWHM (analyze photon detector pinhole images)
- SSIM_index (calculate structural similarity index)
- Image Moments (image moments of n-th rank)
- MS_SSIM_index (multi-scale structural similarity index)
- Colony Counter (count colonies in agar plates)
- Levan (chromosome morphology)
- EXTRAX (electron diffraction intensity extraction)
- Fractal Surface Measurement
- Real Convolver
- FFT
- LoG Filtering
- Background Subtraction and Normalization
- Contrast Enhancer
- Background Correction
- Byte Swapper
- Discrete Cosine Transform (DCT)
- FFT Filter
- FFTJ and DeconvolutionJ
- Unpack 12-bit Images
- De-interlace
- 2D Gaussian Filter
- Kalman Filter
- Dual-Energy Algorithm
- Anisotropic Diffusion (edge-preserving noise reduction)
- Grayscale Morphology Updated
- 2D Hybrid Median Filter
- 3D Hybrid Median Filter
- Spectral Unmixing
- Haar Wavelet Filter and Adaptive Median Filter
- 'A trous' Wavelet Filter
- Kuwahara Filter
- Granulometric Filtering
- Windowed-Sinc Filter (low pass time series filter)
- Anisotropic Diffusion 2D (edge-preserving noise reduction)
- Auto Gamma (gamma correction)
- Linearize Gel Data

If you want to go fast, go alone. If you want to go far, go together.
- African proverb

Getting to know ImageJ

First steps

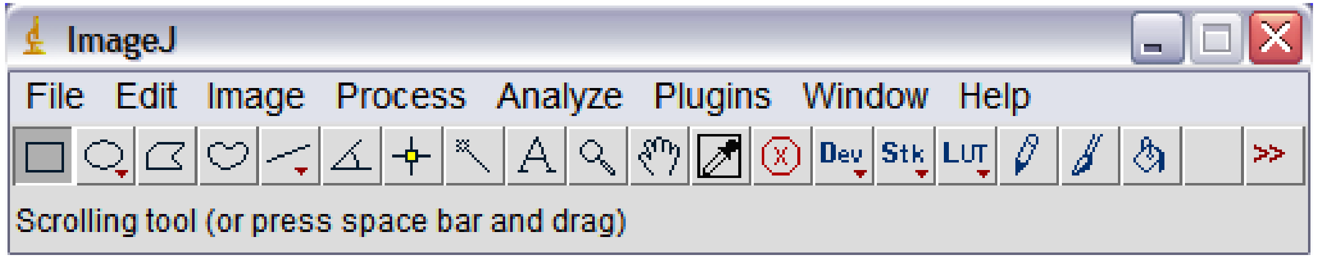
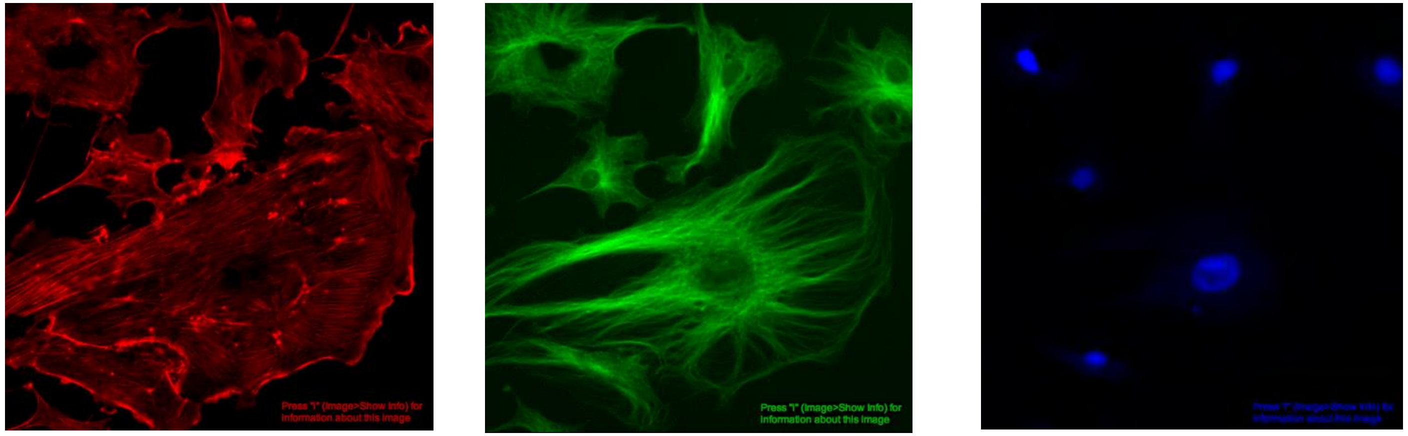


Image Processing Basics



Advanced Tools - Plugins

הטכניון
מכון טכנולוגי לישראל

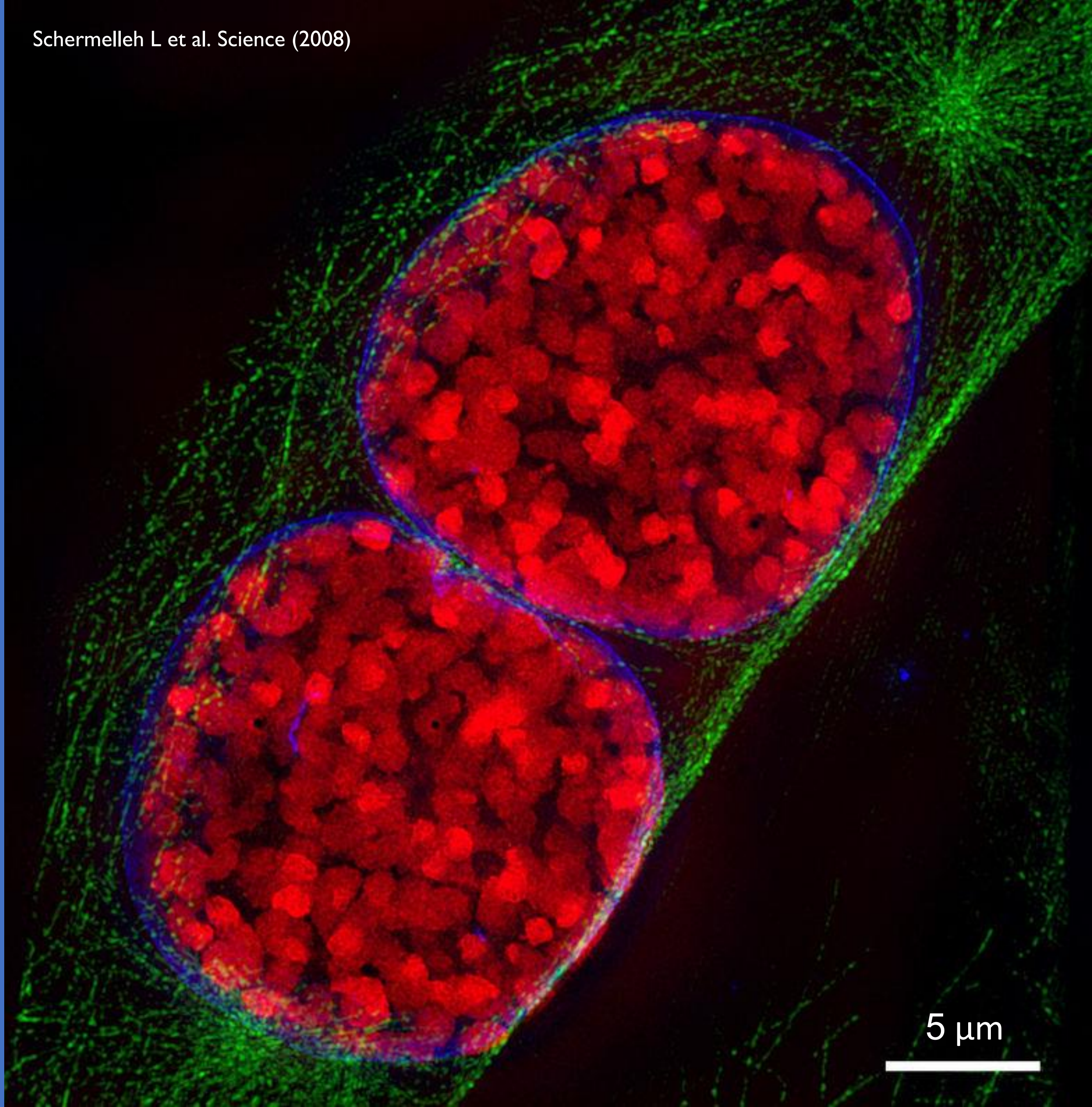
Department of Biomedical Engineering, Technion

Computational optical imaging 336547

Tutorial 2 – Photon detectors

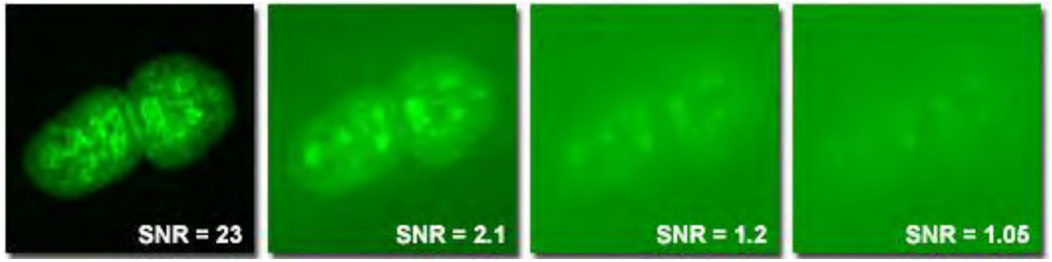
Elias Nehme& Yoav Shechtman

3 November 2020



Photon detectors definition and properties

Devices that **detect events or changes in quantities** (intensities) and provide a **corresponding output** (generally as an electrical signal)

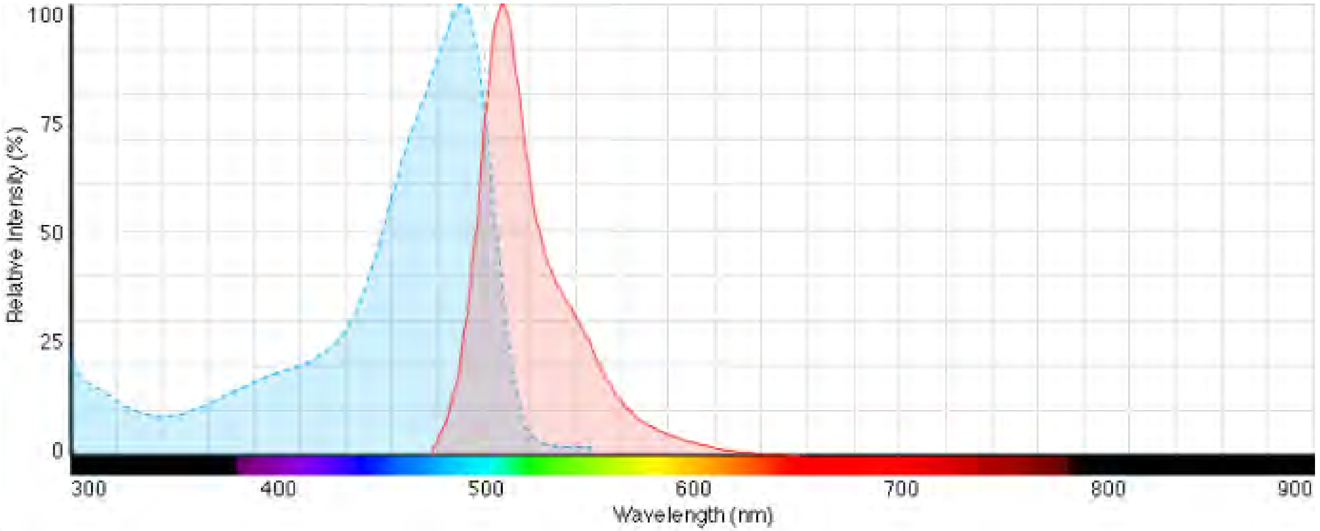


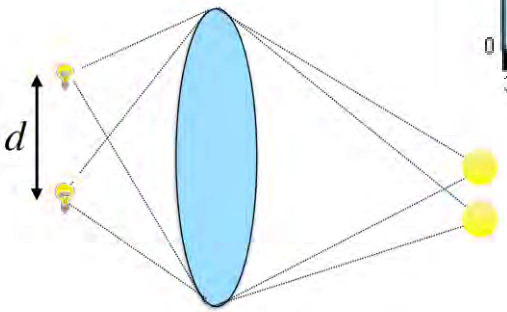
the ‘best’ detector is **sensitive**

Practically there are **many variables to consider**:

Specimen properties

- **Photon flux** emission per unit area
- **Spatial** resolution
- **Temporal** resolution
- Emission **wavelength**
- **Signal-to-noise ratio**
- Microscopy **Technique**

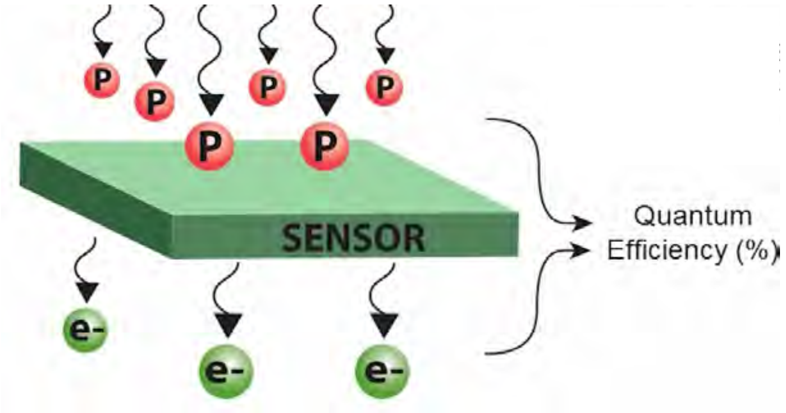




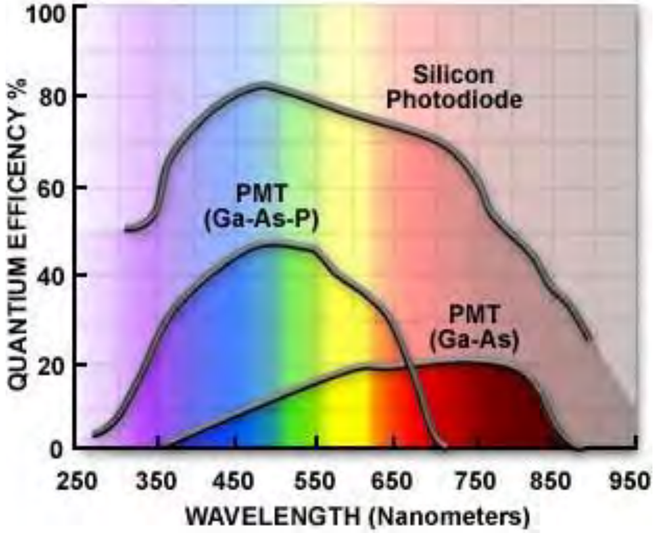
Detector properties

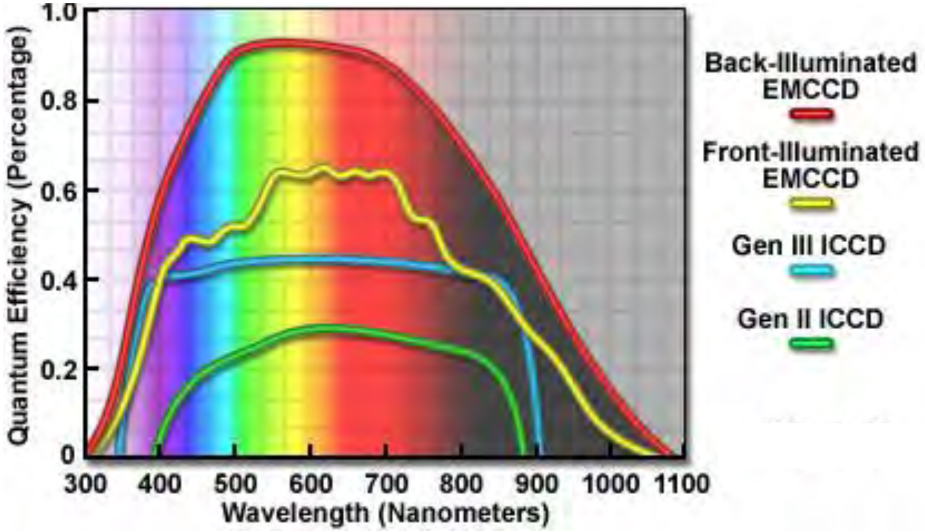
- **Acquisition speed**
- **Quantum efficiency**
- **Noise** levels
- **Pixel size**
- **Dynamic range**

Quantum efficiency – Fraction of photon flux that contributes to the photocurrent in a photodetector or a pixel



EMCCD >= sCMOS > APD > PMT





Detector properties

- **Acquisition speed**
- **Quantum efficiency**
- **Noise** levels
- **Pixel size**
- **Dynamic range**

Array of pixels detectors:

EMCCD

sCMOS

Single pixel detectors:

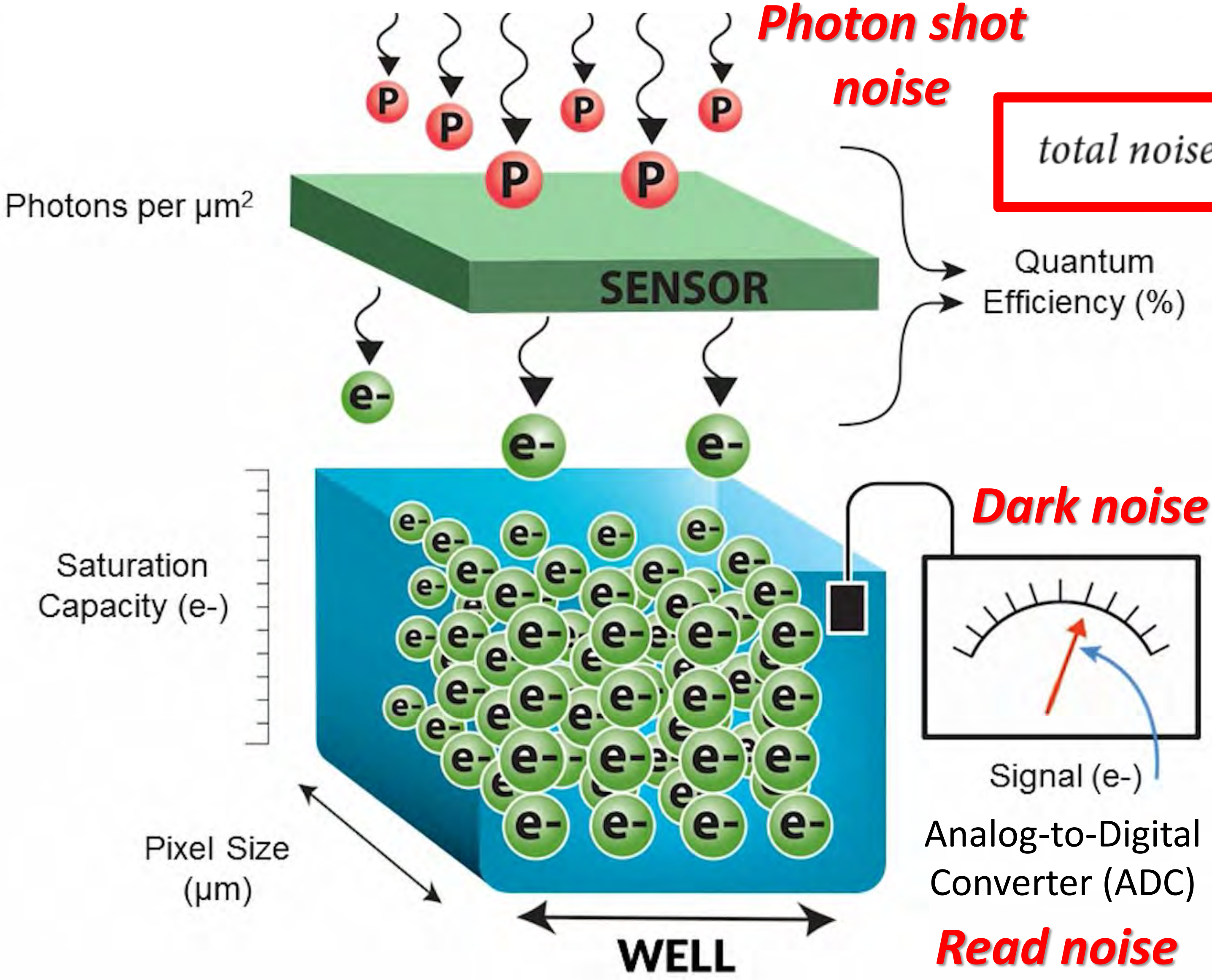
APD

PMT

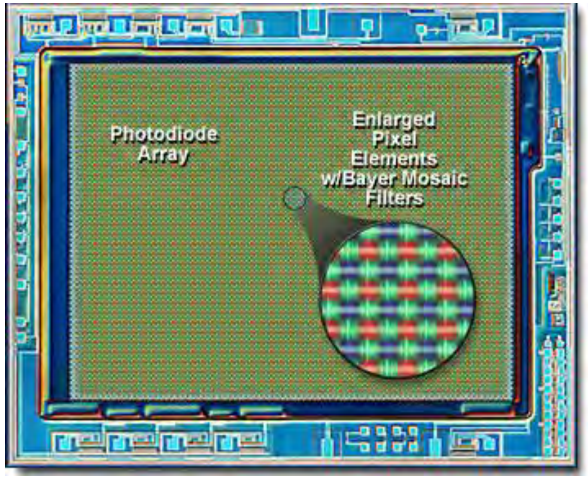
Charged-Coupled Device (CCD) – single pixel

EMCCD

$$\text{total noise} = \sqrt{(\text{read noise})^2 + (\text{dark noise})^2 + (\text{shot noise})^2}$$



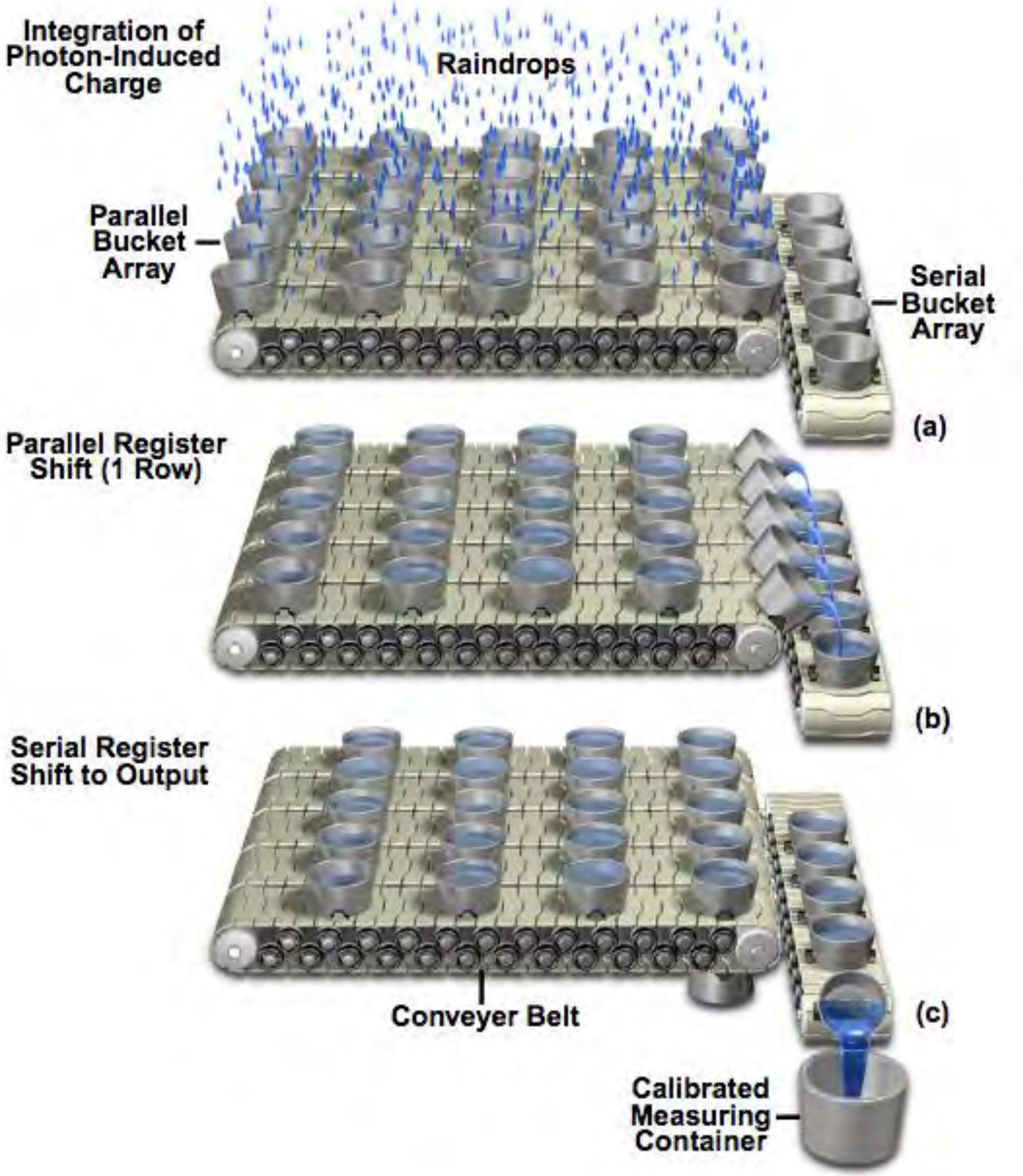
Color detectors Filters out photons:



Grayscale

Monochrome detectors

CCD Readout – Bucket Brigade Analogy



For each exposure time period:

Photons composing the image have been collected by the pixel elements and **converted into electrical potential**

CCD undergoes **readout by shifting rows** of image information in a parallel fashion, one row at a time, **to the serial shift register**

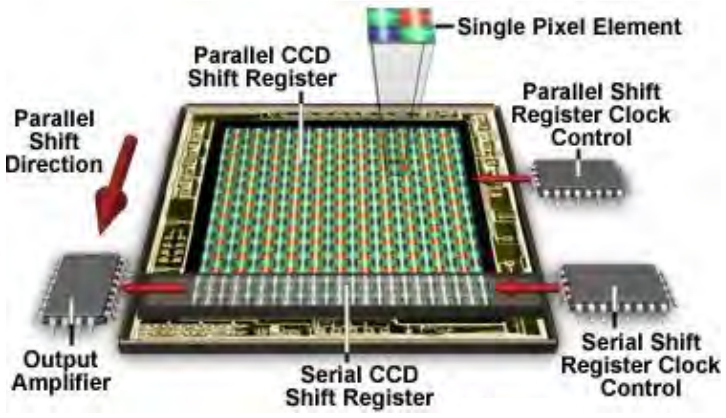
The serial register then **sequentially shifts each row** of image information to an **output amplifier** as a serial data stream

External voltages control the storage and movement of charges

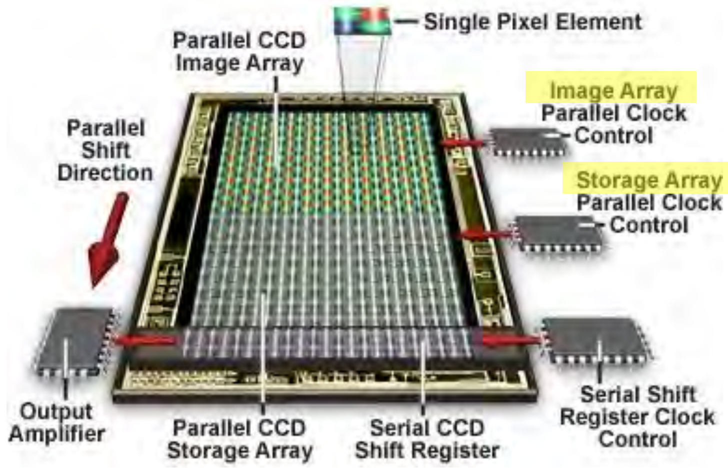
CCD Readout – Full process

1. Camera **shutter is opened** to begin accumulation of photoelectrons
2. End of the integration period = **shutter is closed**
3. Shift of **accumulated charge**
4. An **ADC assigns digital value** for each pixel according to its voltage
5. Each pixel value is **stored in computer** memory or camera frame **buffer**
6. Serial **readout process is repeated** until all pixel rows of the parallel register are emptied
7. CCD is **cleared of residual charge** prior to the next exposure

Full-Frame CCD Architecture

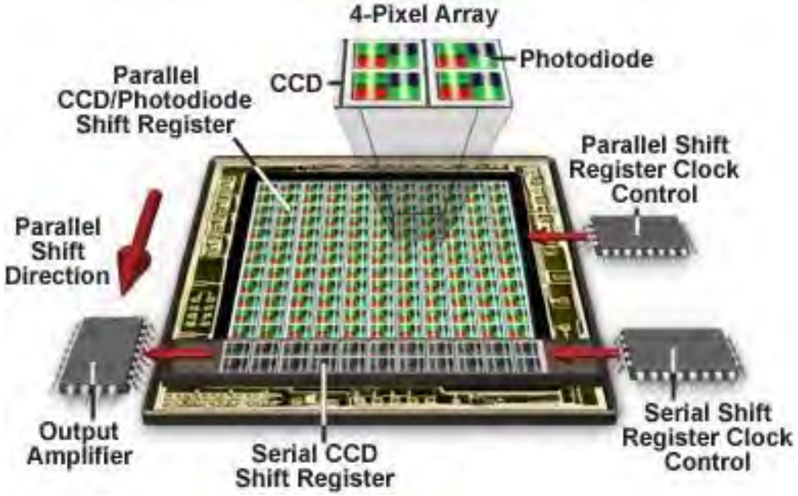


Frame-Transfer CCD Architecture



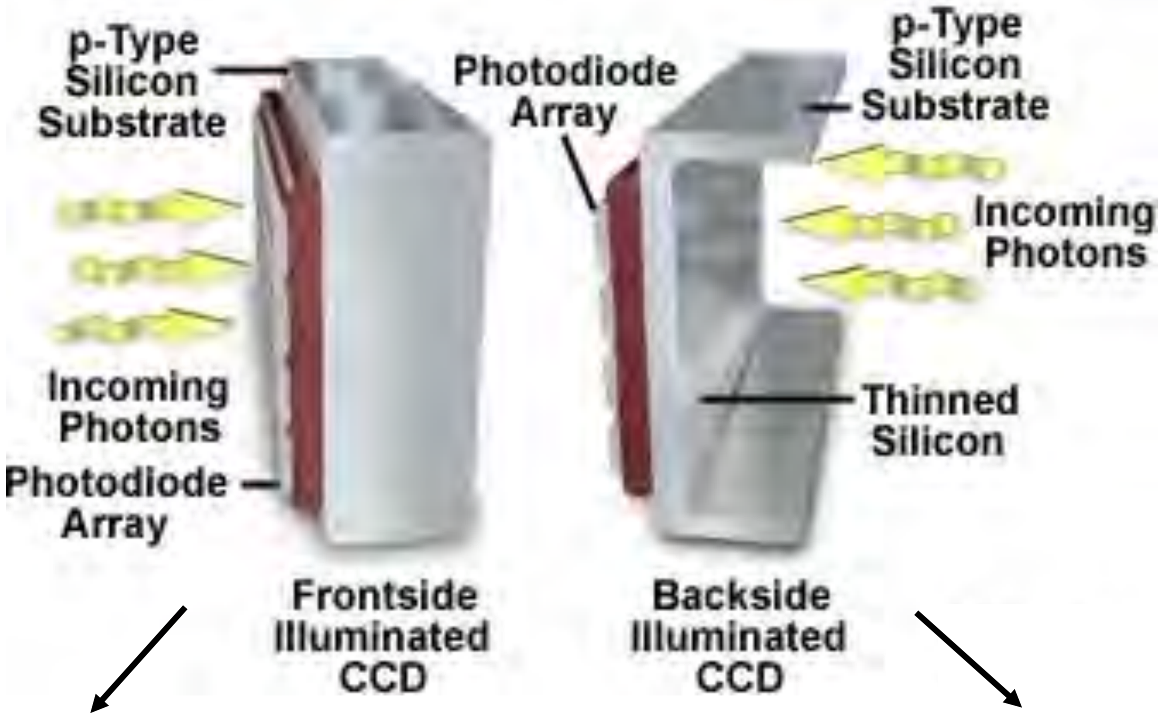
Storage array is being read while the image array is integrating charge for the next image frame

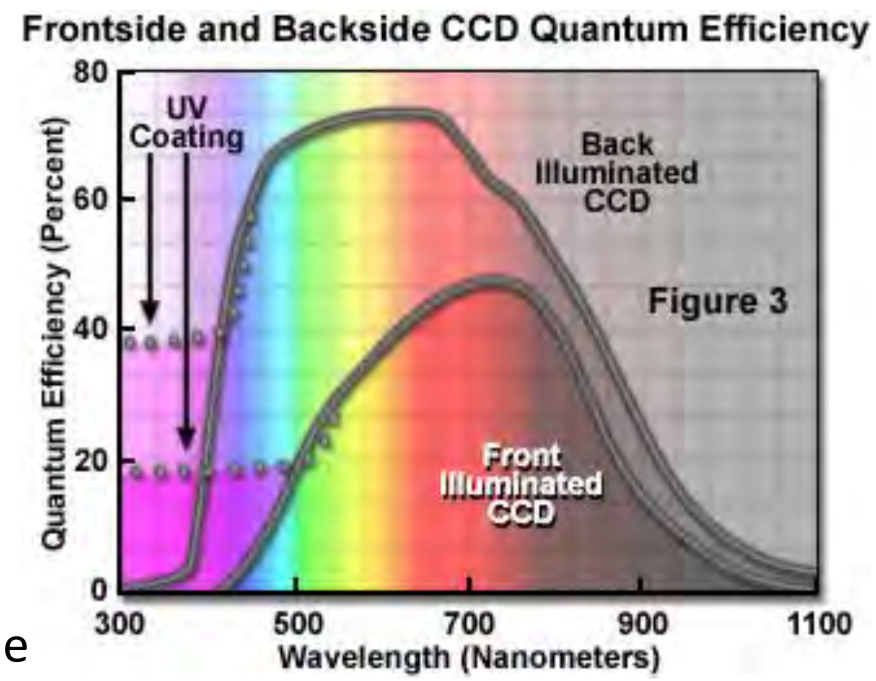
Interline Transfer CCD Architecture



Separate photodiode and parallel readout CCD storage region in each pixel element

Frontside and Backside Illuminated CCDs





Light **passes through structures** used to transfer the charge from the imaging area → reducing the sensitivity (mainly shorter wavelengths)

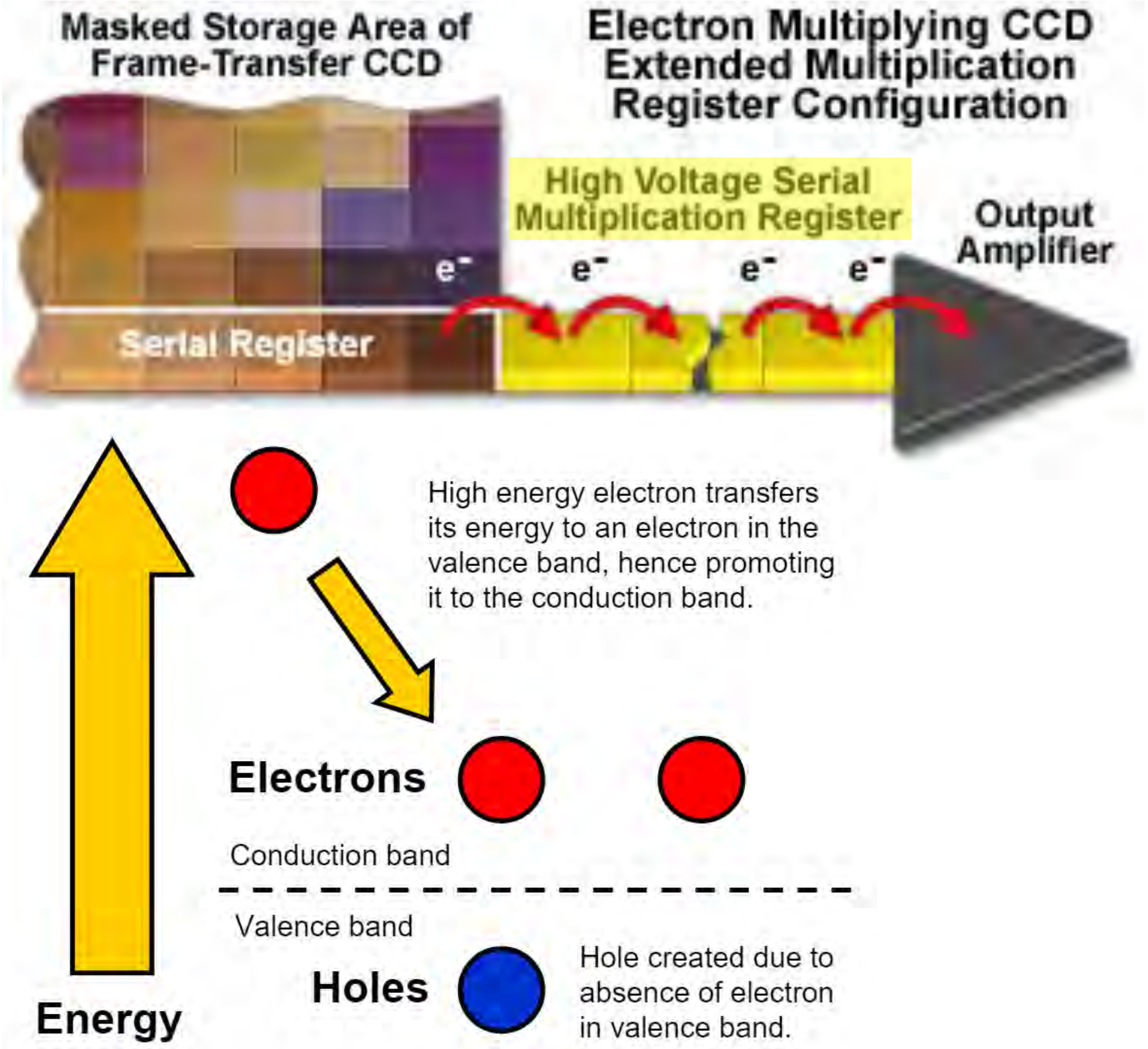
Light falls onto the back of the CCD in a **thinned transparent region** (about 10-15 microns) → high quantum efficiency can be realized

EMCCD – Electron Multiplying CCD

Addition of an **Electron Multiplication register** (‘gain register’ between the usual serial shift register and the output amplifier)

Provide a mechanism to **improve signal-to-noise ratio for signal levels below the CCD read-noise** floor

When charge is transferred by applying a **higher-than-normal voltage**, secondary electrons are generated in the silicon by the process of **impact ionization**



EMCCD – Different effects

High on-chip multiplication gain for single-photon detection: **any level of unsuppressed dark current is significant**

Cooling system

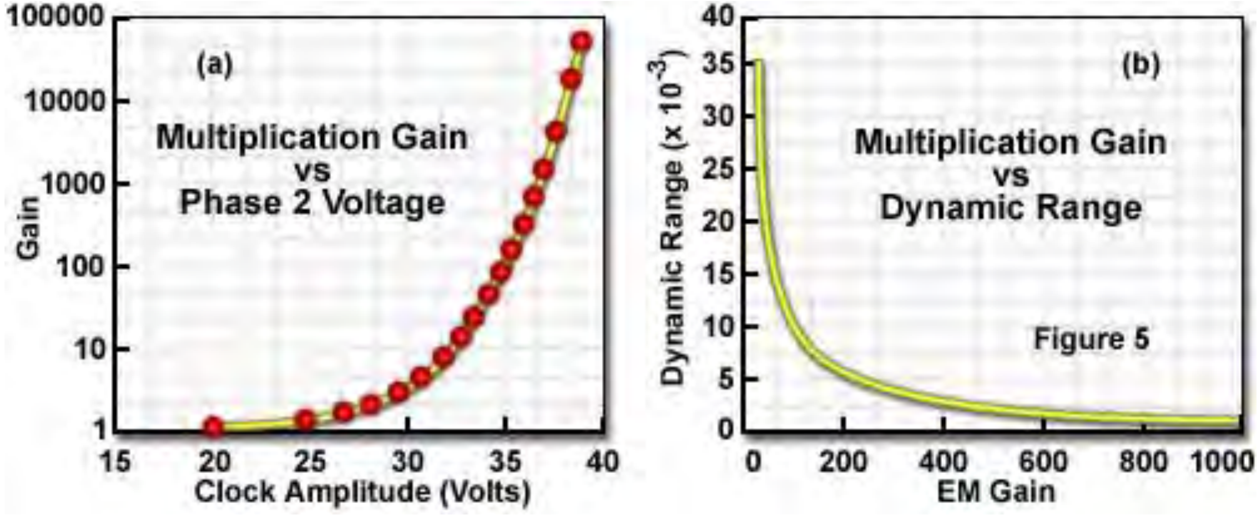
- **Dark noise** arises from thermal fluctuations and is **reduced by cooling the sensor**
- The **probability of secondary electron generation increases as temperature decreases** → higher gain values are achieved
- The variation of multiplication gain with temperature illustrates the **importance of maintaining precise temperature stability**

Example: $\boxed{N_r / M}$

Read noise = 60 electrons (rms) at 10 (MHz)

→ *Sub-electron effective read noise level with gain ≥ 60*

Multiplication gain is independent of readout speed, the **noise performance can be achieved at any speed**



EMCCD – Noise

EMCCD

Due to the **probabilistic nature of the impact ionization process** a statistical variation occurs in the on-chip multiplication gain

The uncertainty in the gain produced introduces an **additional system noise component** which is evaluated quantitatively as the **excess noise factor**

$$\mathbf{SNR = (S \cdot Q_e) / N_{total}}$$

$$\boxed{N_{total} = [\underbrace{(S \cdot Q_e \cdot F^2)}_{\text{Photon shot noise}} + \overbrace{(D \cdot F^2)}^{\text{Dark noise}} + \underbrace{(N_r / M)^2}_{\text{Read noise}}]^{1/2}}$$

S the number of incident photons per pixel
Q(e) the quantum efficiency
$\mathbf{N_{total}}$ the total noise in the system

F the excess noise factor
D the total dark signal
N(r) the camera read noise
M the on-chip multiplication gain

Excess noise factor typically range between 1.0 and 1.4 for multiplication gain factors up to 1000x

Other gain-dependent source of noise:
clocking induced charge (CIC)

Detector properties

- **Acquisition speed**
- **Quantum efficiency**
- **Noise** levels
- **Pixel size**
- **Dynamic range**

Array of pixels detectors:

EMCCD

sCMOS

Single pixel detectors:

APD

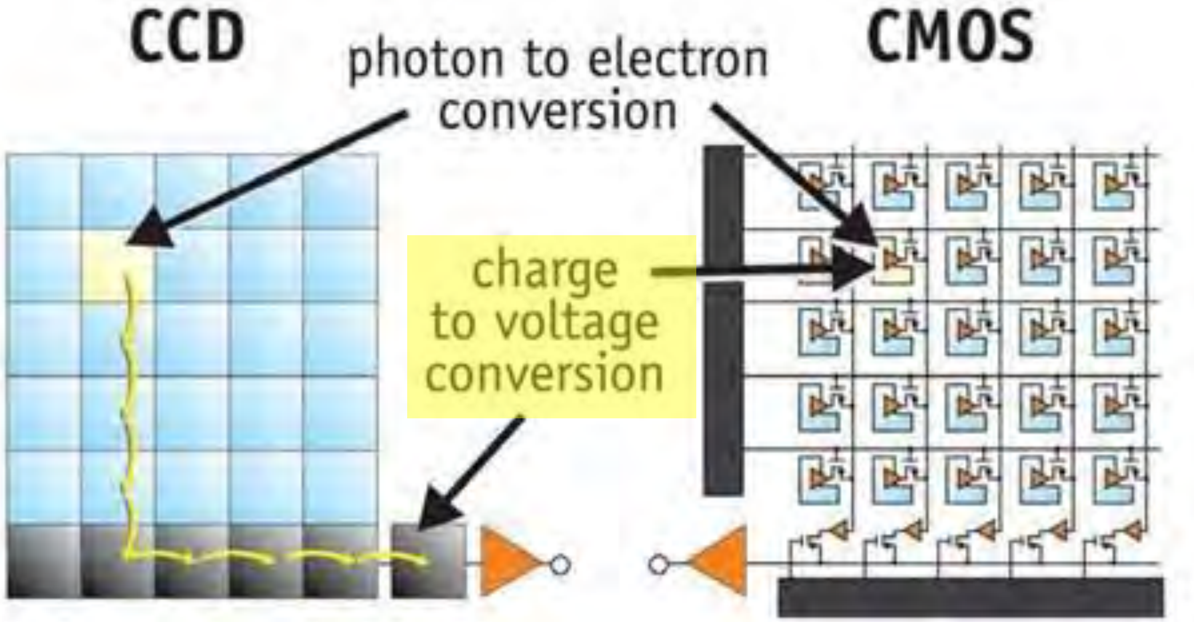
PMT

Complementary Metal Oxide Semiconductor (CMOS)

sCMOS

CMOS convert charge to voltage inside each pixel

CCD move photogenerated charge **from pixel to pixel** and **convert it to voltage at an output node**



__Fill factor__ – the __portion__ of the __entire pixel array that is used to detect__ incoming photons during exposure

CMOS sensors require around **100x less power than CCD** → perfect choice for camera phone sensors

Issues with CMOS:

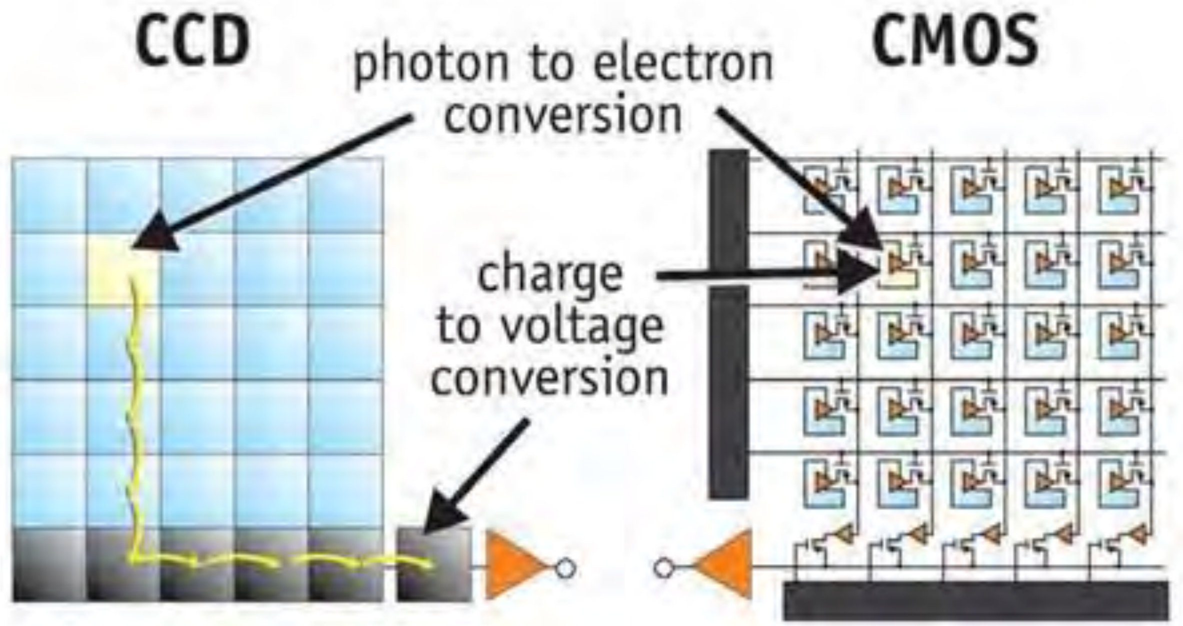
1. **Fill factor** of 30%: **loss in sensitivity and SNR**
2. **Circuitry reflect incident photons**: potential pixel crosstalk, light scattering, and diffraction
3. **Lower quantum efficiency**

→ **Microlens arrays**

CMOS VS CCD

sCMOS

	CCD	CMOS
Fill Factor	High	Low
Image acquisition time	Slow (serial)	Fast (parallel)
Power consumption	High	Low

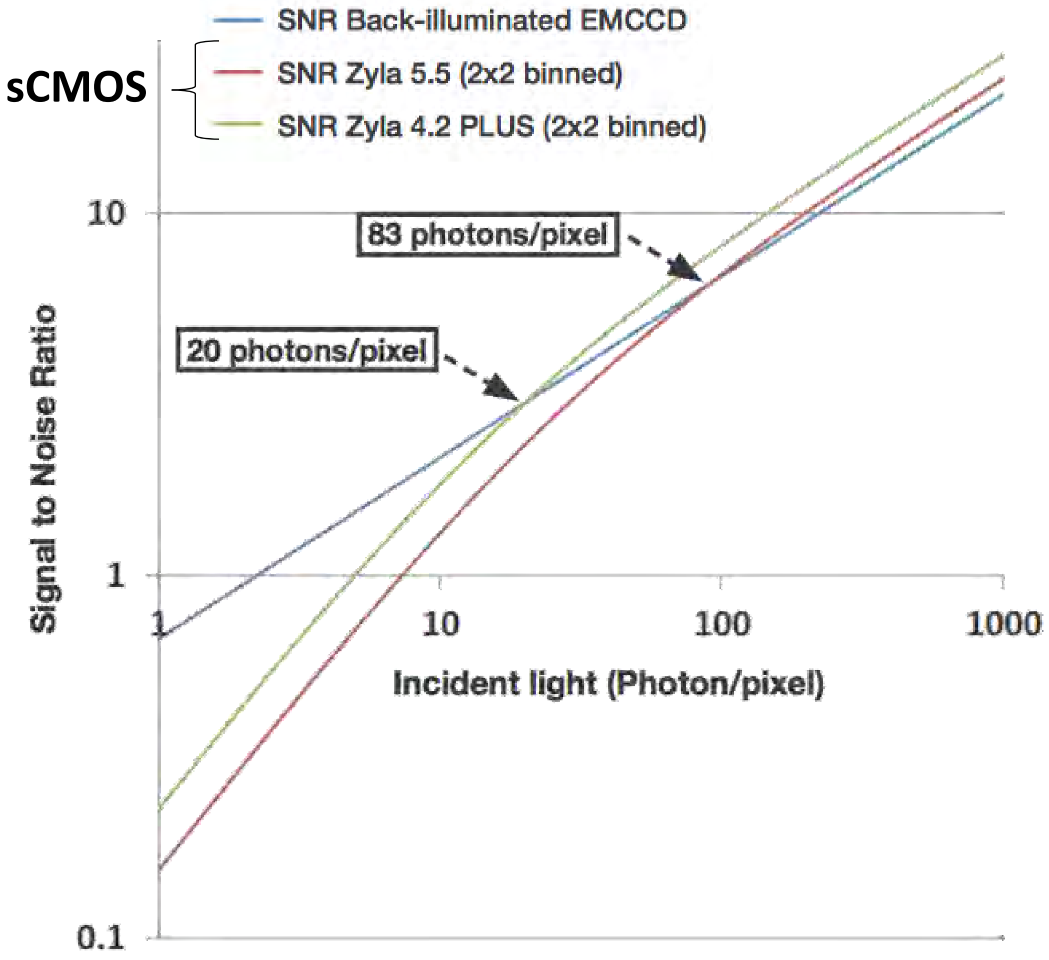


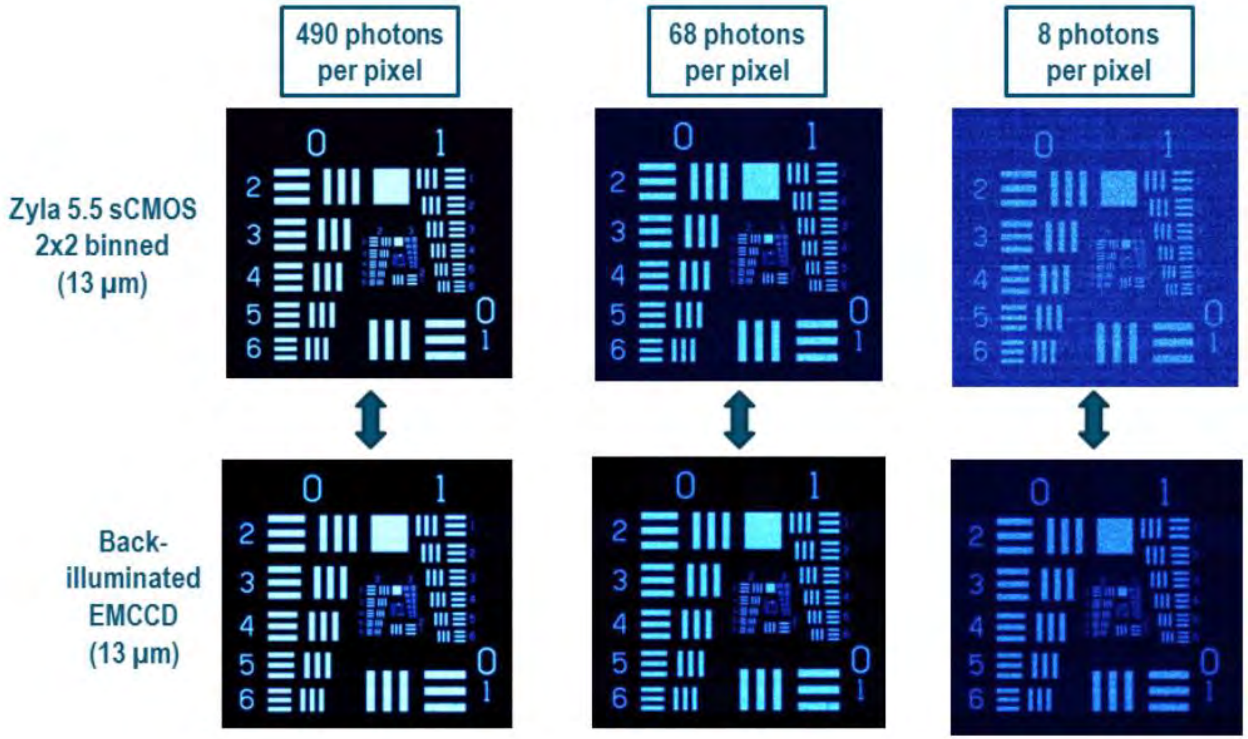
Scientific CMOS (sCMOS) is a breakthrough technology **based on next-generation CMOS image sensor design and fabrication techniques**

A comparison – CCDs, EMCCDs, sCMOS

Parameter	sCMOS (Zyla)	Interline CCD	EMCCD
Sensor Format	5.5 megapixel	1.4 to 4 megapixel	0.25 to 1 megapixel
Pixel Size	6.5 µm	6.45 to 7.4 µm	8 to 16 µm
Read Noise	$1.2e^-$ @ 30 frames/sec $1.45e^-$ @ 100 frames/sec	4 - 10 e^-	< 1 e^- (with EM gain)
Full Frame Rate (max.)	100 frames/sec @ full resolution	3 to 16 frames/sec	~ 30 frames/sec
Quantum Efficiency (max.)	80%	60%	90% 'back-illuminated' 65% virtual phase
Dynamic Range	25,000:1 (@ 30 frames/sec)	~ 3,000:1 (@ 11 frames/sec)	8,500:1 (@ 30 frames/sec with low EM gain)
Multiplicative Noise	None	None	1.41x with EM gain (effectively halves the QE)

Summary – CCDs, EMCCDs, sCMOS





- **CCD**: standard for general microscopy applications, best choice for a variety of fluorescence microscopy applications
- **EMCCD**: best solution when imaging at very low light levels with relatively high speed, such as in single molecule fluorescence
- **sCMOS**: best solution for large field of views, high speed and sensitivity

Detector properties

- **Acquisition speed**
- **Quantum efficiency**
- **Noise** levels
- **Pixel size**
- **Dynamic range**

Array of pixels detectors:

EMCCD

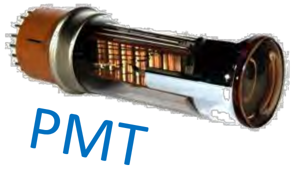
sCMOS

Single pixel detectors:

APD

PMT

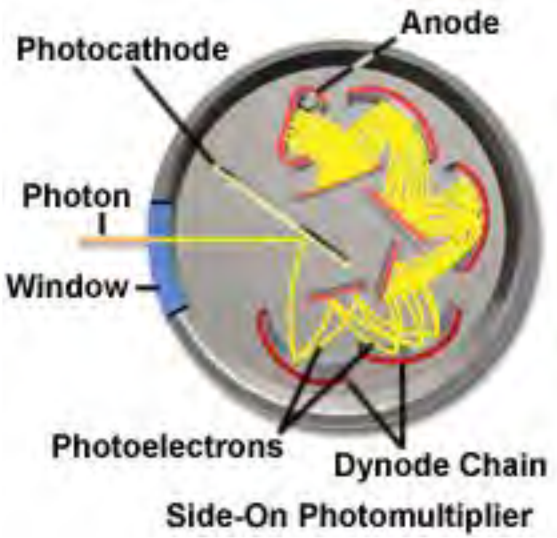
Photomultiplier Tube (PMT)

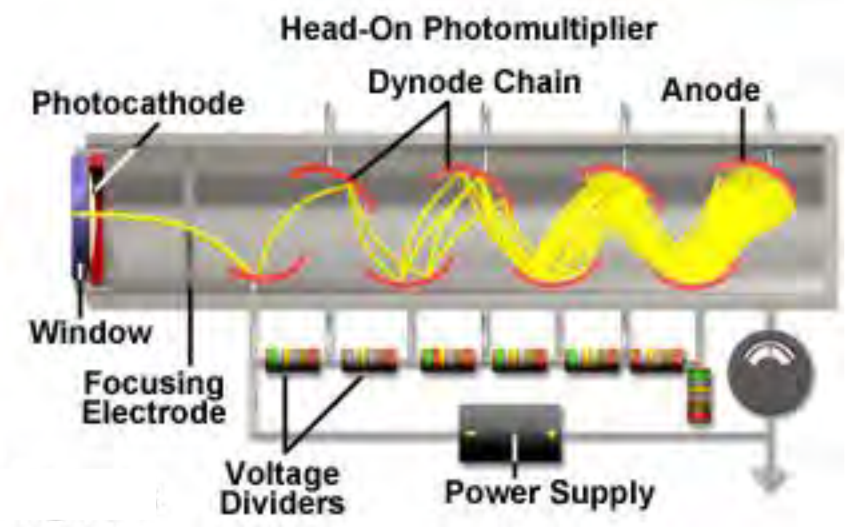


- Low dark current, electron gains of $\mathbf{10^8}$: **Very high signal-to-noise ratio**
- PMTs do not store charge: **nanosecond response to changes to input light fluxes**
- **Photon counting mode**

Reflective

- Faster rise times
- Higher quantum efficiency





Transmission

- Larger and more uniform photosensitive area
- Sensitive photocathode design

S/N (Signal-to-Noise) = $\mathbf{S/(N_s^2 + N_d^2)^{1/2}}$

N(s) Shot noise
N(d) Dark noise fluctuations
S/N Signal-to-noise ratio

Electrons multiplication by impact ionization
Excess Noise Factor < 1.4

Dark current

- Thermal emission of electrons from the photocathode
- Leakage current between dynodes
- Electronic noise
- Stray high-energy radiation

Useful in confocal microscope

Detector properties

- **Acquisition speed**
- **Quantum efficiency**
- **Noise** levels
- **Pixel size**
- **Dynamic range**

Array of pixels detectors:

EMCCD

sCMOS

Single pixel detectors:

APD

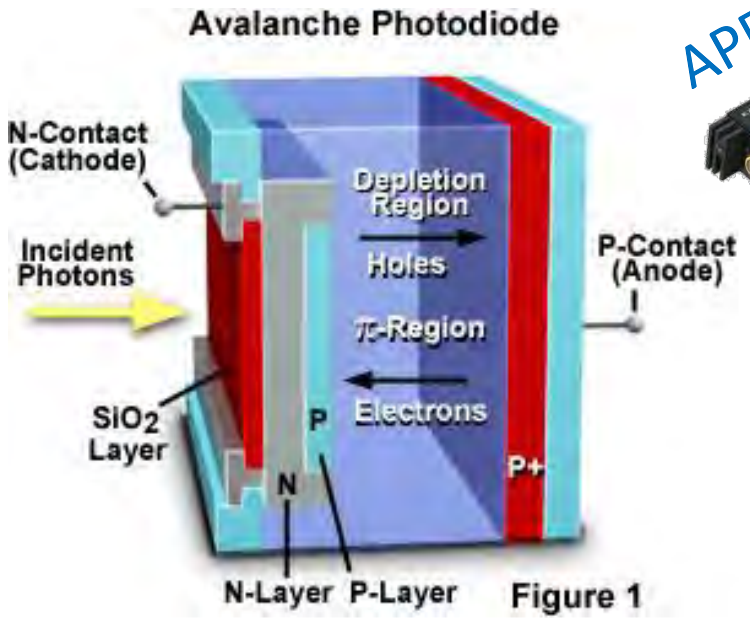
PMT

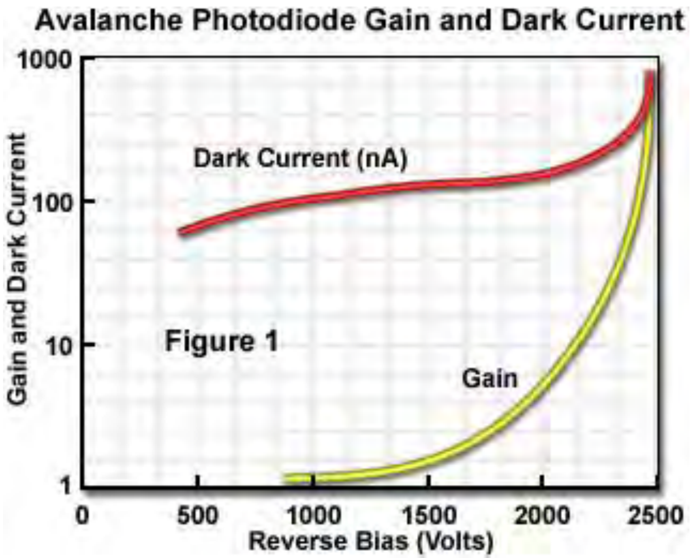
Avalanche Photodiode (APD)

APDs: semiconductor analog of photomultipliers

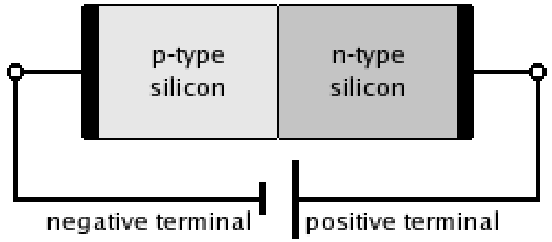
- Modest gain (50-1000)
- High quantum efficiency
- High dark current

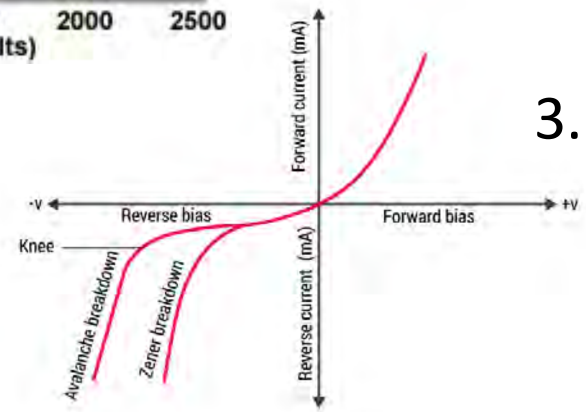
EMCCD >= sCMOS > APD > PMT





1. Absorption of incident photons creates **electron-hole pairs**
2. A high reverse bias voltage creates a strong internal electric field, which **accelerates the electrons** through the silicon crystal lattice
3. This **produces secondary electrons** by **impact ionization**





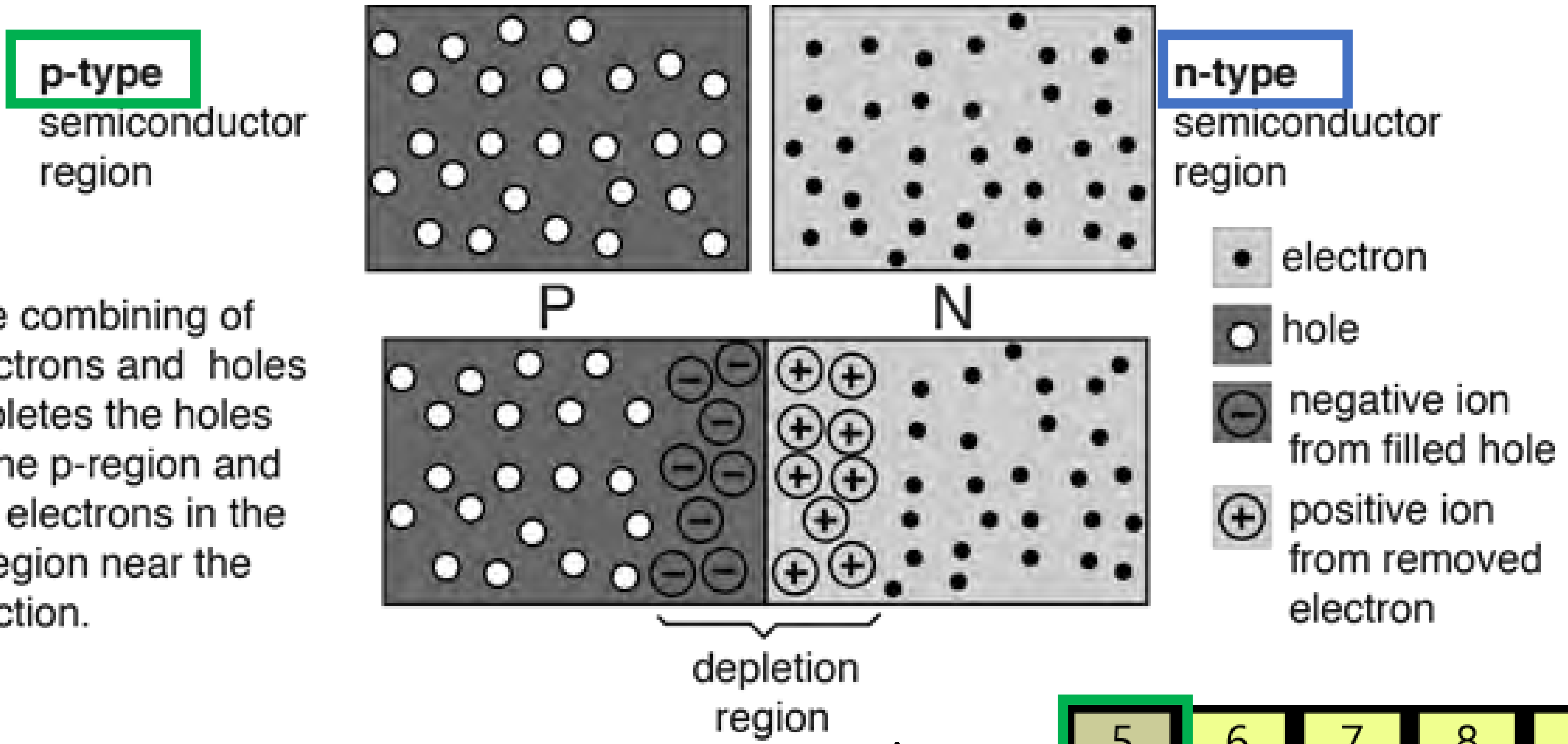
Excess Noise Factor

$$ENF = \kappa M + (2 - 1/M)(1 - \kappa) > 2$$

κ the ionization coefficient ratio

M gain

P-N Junction – reminder



The combining of electrons and holes depletes the holes in the p-region and the electrons in the n-region near the junction.

Acceptor

5 B	6 C	7 N	8 O	9 F	10 Ne
13 Al	14 Si	15 P	16 S	17 Cl	18 Ar

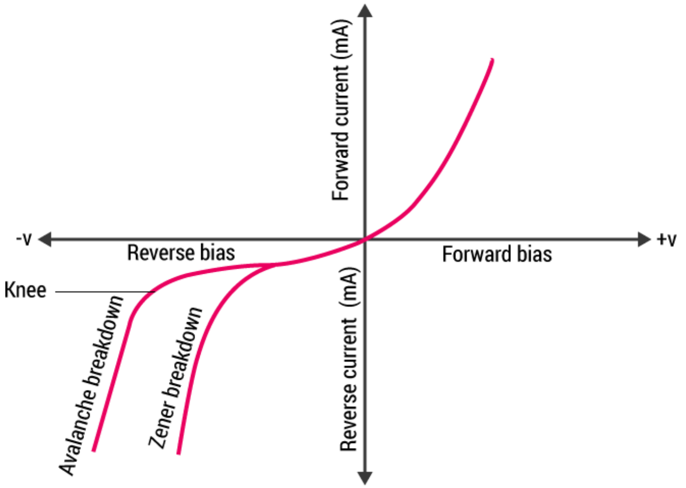
Donor

Single-Photon Avalanche Photodiode (SPAD)

SPADs are APDs reverse-biased at a voltage V_A that **exceeds breakdown voltage** V_B of the junction

At this bias – the electric field is so high that a **single charge carrier injected into the depletion layer can trigger a self-sustaining avalanche** (signal gain > $\mathbf{10^5}$)

1. The **current rises swiftly** (sub-nanosecond rise-time) to a macroscopic steady level in the milliampere range
2. The leading edge of the **avalanche pulse marks** (with picosecond time jitter) **the arrival time of the detected photon**
3. The current continues until the **avalanche is quenched by lowering the bias voltage down to or below** V_B: the lower electric field is no longer able to accelerate carriers to impact-ionize with lattice atoms, therefore current ceases. This stops the breakdown or **resets the APD**
4. In order to be able to detect **another photon, the bias voltage must be raised again above breakdown**



→ **Single photon counting at 10MHz** with **dark count rates well below 1kHz** & **quantum efficiency** reaching **90%**

Applications – PMTs & SPADs

PMT

- Confocal microscopy
- Fluorescence spectroscopy

SPAD

- TCSPC: time-correlated single photon counting
- Single-molecule detection
- STED microscopy
- Fluorescence correlation spectroscopy (FCS)

SPAD arrays

- 100 000 Frames/s 64x32 Single-Photon Detector Array for 2-D Imaging and 3-D Ranging
- Fluorescence lifetime imaging microscopy and correlation spectroscopy

Detector properties

- **Acquisition speed**
- **Quantum efficiency**
- **Noise** levels
- **Pixel size**
- **Dynamic range**

Array of pixels detectors:

EMCCD

sCMOS

Single pixel detectors:

APD

PMT

References

M. Vitali, D. Bronzi, A. J. Krmpot, S. Nikolić , F. Schmitt, C. Junghans, S. Tisa, T. Friedrich, V. Vukojević , L. Terenius, F. Zappa, Senior Fellow, IEEE and R. Rigler "A single-photon avalanche camera for fluorescence lifetime imaging microscopy and correlation spectroscopy", JSTQE, 2014

D. Bronzi, F. A. Villa, S. Tisa, A. Tosi, F. Zappa, D. Durini, S. Weyers, and W. Brockherde, "100 000 Frames/s 64x32 Single-Photon Detector Array for 2-D Imaging and 3-D Ranging", IEEE J. Select. Topics Quantum Electron., vol. 20, no. 6, pp. 354–363, Nov. 2014

http://olympus.magnet.fsu.edu/primer/digitalimaging/index.html
https://www.microscopyu.com/digital-imaging
http://www.hamamatsu.com/jp/en/community/optical_sensors/articles/guide_to_detector_selection/index.html
http://www.andor.com/scientific-cameras
https://www.wikipedia.org

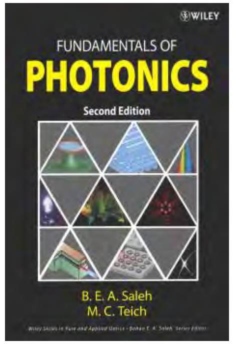
Department of Biomedical Engineering, Technion
Computational optical imaging 336547

Tutorial 3 – Fourier optics, Lenses & FFT

Elias Nehme & Yoav Shechtman

10 November 2020

Fourier Optics

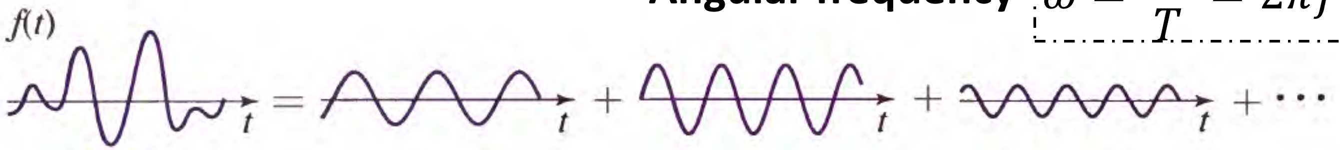


Description of **propagation of light waves** based on **harmonic analysis (FT) and linear systems**

Harmonic analysis (FT)

$$f(t) = \int_{-\infty}^{\infty} F(f)\ \exp(\ j2\pi f t\ )\ df$$

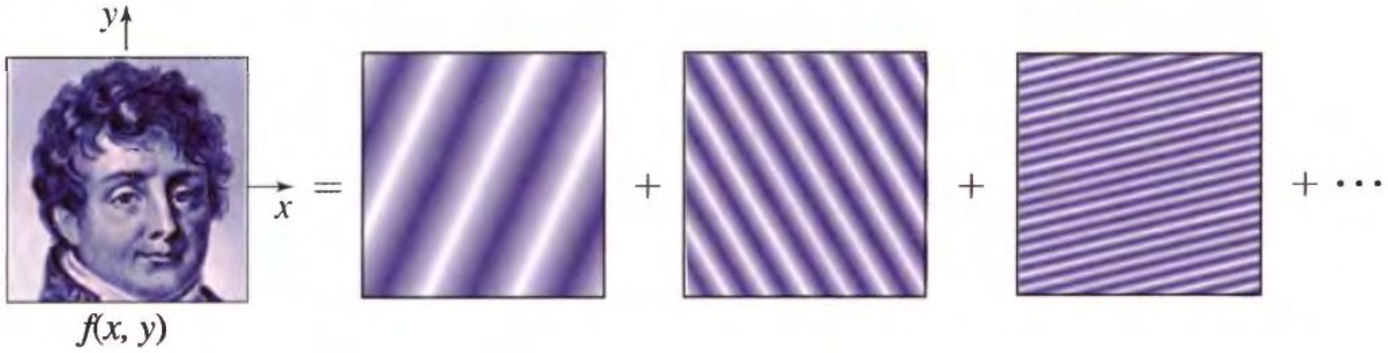
Angular frequency $\omega = \dfrac{2\pi}{T} = 2\pi f$



Match the forward traveling plane wave

$$f(x,y) = \iint_{-\infty}^{\infty} F(\nu_x,\nu_y)\exp\left[-j2\pi(\nu_x x + \nu_y y)\right]\, d\nu_x\, d\nu_y$$

Wave number $k = \dfrac{2\pi}{\lambda} = 2\pi\upsilon$



Linear systems

If the **response** to each **harmonic function** is **known** → the **response** to an **arbitrary input** is **determined**

Plane wave

Wave function $u = u(r,t)$

Angular frequency $\omega = \frac{2\pi}{T} = 2\pi f$

Light speed in a medium $c = \frac{c_0}{n}$

Wave number $k = \frac{2\pi}{\lambda} = 2\pi v$

The optical wave equation:

$$\nabla^2 u - \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} = 0 \qquad \nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2 + \partial^2/\partial z^2$$

Laplacian operator

The monochromatic wave:

$$u(\mathbf{r},t) = a(\mathbf{r})\cos[\,2\pi f t + \varphi(\mathbf{r})]$$

amplitude phase

The complex wavefunction:

$$U(\mathbf{r},t) = a(\mathbf{r})\exp[j\varphi(\mathbf{r})]\exp(\,j2\pi f t) \Longrightarrow u(\mathbf{r},t) = \mathrm{Re}\{U(\mathbf{r},t)\} = \tfrac{1}{2}[U(\mathbf{r},t) + U^*(\mathbf{r},t)]$$

$$\left.\begin{aligned} &U(\mathbf{r},t) = U(\mathbf{r})\exp(\,j2\pi f t) \\ &\nabla^2 U - \frac{1}{c^2}\frac{\partial^2 U}{\partial t^2} = 0 \end{aligned}\right\} \quad \nabla^2 U + k^2 U = 0$$

Helmholtz equation

for the complex amplitude $U(\mathbf{r})$

$k = \frac{2\pi f}{c} = \frac{\omega}{c}$

$\lambda = \frac{c}{f}$

$$U(\mathbf{r}) = A\exp(-j\mathbf{k}\cdot\mathbf{r}) = A\exp[-j(k_x x + k_y y + k_z z)]$$

Plane wave – Simplest solution of the Helmholtz equation

Plane wave

Wave function $u = u(r,t)$ Angular frequency $\omega = \frac{2\pi}{T} = 2\pi f$

Light speed in a medium $c = \frac{c_0}{n}$ Wave number $k = \frac{2\pi}{\lambda} = 2\pi\upsilon$

$\lambda = \frac{c}{f}$ $k = \frac{2\pi f}{c} = \frac{\omega}{c}$

The wavefront (surface with constant phase):

Complex amplitude

$$U(\mathbf{r}) = A\exp(-j\mathbf{k}\cdot\mathbf{r}) = A\exp\left[-j(k_x x + k_y y + k_z z)\right]$$

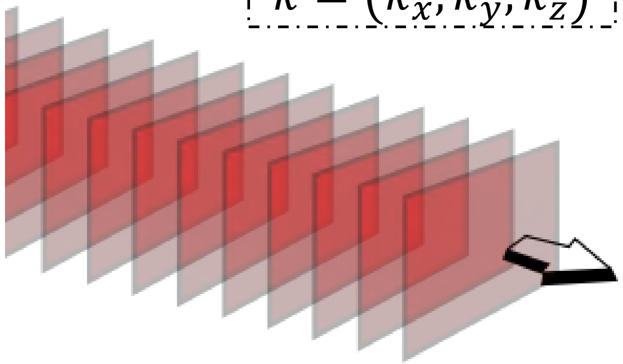
$$\arg\{U(\mathbf{r})\} = \arg\{A\} - \mathbf{k}\cdot\mathbf{r}$$

$$\mathbf{k}\cdot\mathbf{r} = k_x x + k_y y + k_z z = 2\pi q + \arg\{A\}$$

Integer

Parallel planes perpendicular to the wave vector

$\vec{k} = (k_x, k_y, k_z)$



There is one-to-one correspondence between:

$U(x,y,z)$ $\rightarrow$ $f(x,y) = U(x,y,0)$

Plane wave **Harmonic function**

$f(x,y)$ $\rightarrow$ $U(x,y,z) = f(x,y)\exp(-jk_z z)$

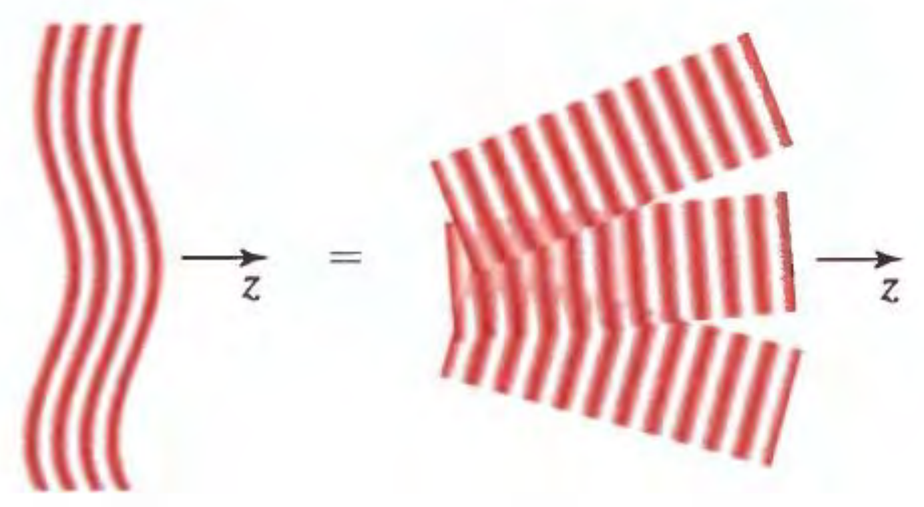
Harmonic function **Plane wave**

$k_x{}^2 + k_y{}^2 + k_z{}^2 = k^2 = \left(\frac{2\pi}{\lambda}\right)^2$ $k_z = \pm\sqrt{k^2 - k_x{}^2 - k_y{}^2}$ *forward traveling wave*

Fourier Optics principles

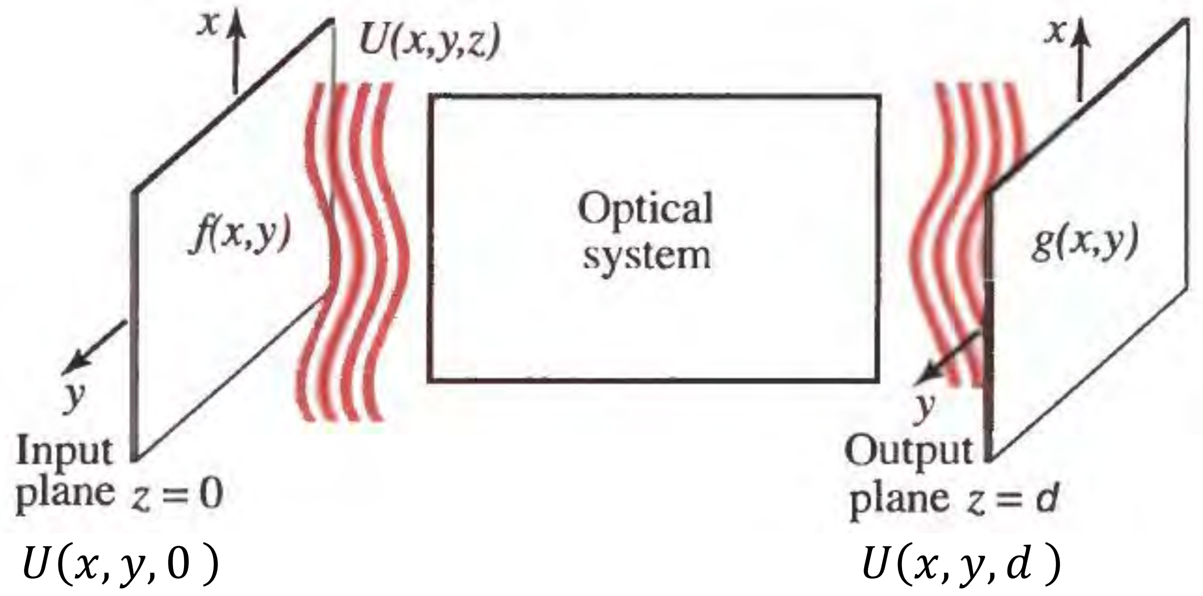
Harmonic analysis (FT)

An **arbitrary function** can be analyzed as a **superposition of harmonic function** → An **arbitrary wave** may be analyzed as a **sum of plane waves**



Linear systems

Describing the **propagation of light** through linear optical component **using linear-system approach**



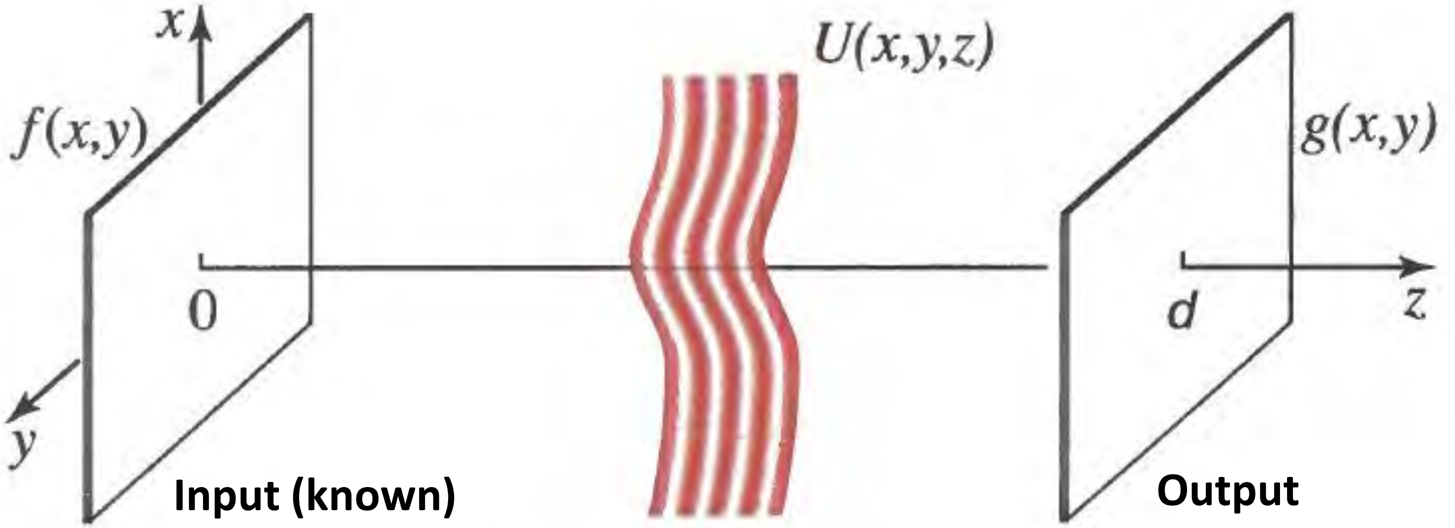
Impulse response function

Transfer function

Transfer function of Free space

$$k_z = \sqrt{k^2 - {k_x}^2 - {k_y}^2}$$

Propagation of monochromatic optical wave in the free space between the planes z=0 and z=d:



$$f(x,y) = U(x,y,0)$$

$$g(x,y) = U(x,y,d)$$

$$A\exp[-j2\pi(\nu_x x + \nu_y y)]$$

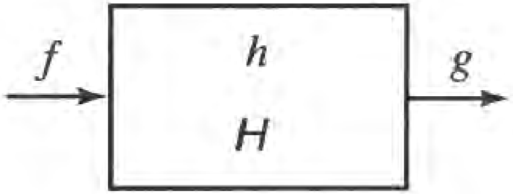
$$g(x,y) = A\exp[-j(k_x x + k_y y + k_z d)]$$

Linear + Shift invariant (LSI) system

$$\nabla^2 U + k^2 U = 0$$

$h(x,y)$ **Impulse response function**

$H(\upsilon_x,\upsilon_y)$ **Transfer function**



$$H(\upsilon_x,\upsilon_y) = \frac{g(x,y)}{f(x,y)} = \exp(-jk_z d) \Rightarrow \boxed{H(\nu_x,\nu_y) = \exp\left(-j2\pi d\sqrt{\lambda^{-2} - \nu_x^2 - \nu_y^2}\right)}$$

Transfer function of Free space

$$\boxed{H(\nu_x,\nu_y)=\exp\left(-j2\pi d\sqrt{\lambda^{-2}-\nu_x^2-\nu_y^2}\right)}$$

$\nu_x^2+\nu_y^2 \le \lambda^{-2}$ $\quad |H(\nu_x,\nu_y)|=1 \quad \arg\{H(\nu_x,\nu_y)\}$

Spatial shift

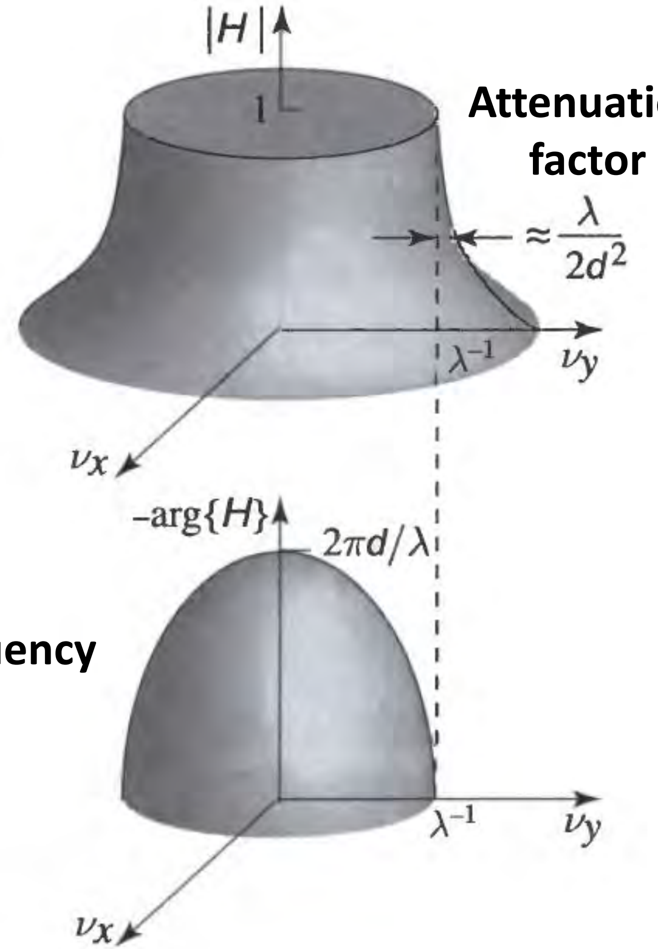
$\nu_x^2+\nu_y^2 > \lambda^{-2}$ $\quad |H(\nu_x,\nu_y)| \quad \arg\{H(\nu_x,\nu_y)\}=0$

Evanescent wave

LPF

Cut off frequency

$\sim\lambda^{-1}$



Attenuation factor

Fresnel approximation:

$$\nu_x^2+\nu_y^2 \ll \lambda^{-2}$$

$$\boxed{H(\nu_x,\nu_y)\approx H_0\exp\left[j\pi\lambda d\left(\nu_x^2+\nu_y^2\right)\right]}$$

$$H_0=\exp(-jkd)$$

$$N_F\frac{\theta_m^2}{4}\ll 1$$

$$N_F=\frac{a^2}{\lambda d}$$

$$\theta_x=\sin^{-1}\lambda\nu_x,\quad \theta_y=\sin^{-1}\lambda\nu_y.$$

$$\theta_x\approx\lambda\nu_x,\quad \theta_y\approx\lambda\nu_y$$

Paraxial approximation

a largest radial distance in the **output plane**

Impulse response of Free space

Fresnel approximation:

$$H(\nu_x, \nu_y) \approx H_0 \exp\left[j\pi\lambda d\left(\nu_x^2 + \nu_y^2\right)\right] \xrightarrow{F^{-1}} \boxed{h(x,y) \approx h_0 \exp\left[-jk\frac{x^2+y^2}{2d}\right]}$$

$$h_0 = \frac{j}{\lambda d}\exp(-jkd)$$

Fraunhofer approximation:

$$\boxed{g(x,y) \approx h_0\, F\left(\frac{x}{\lambda d}, \frac{y}{\lambda d}\right)} \qquad N_F \ll 1 \quad \text{and} \quad N'_F \ll 1$$

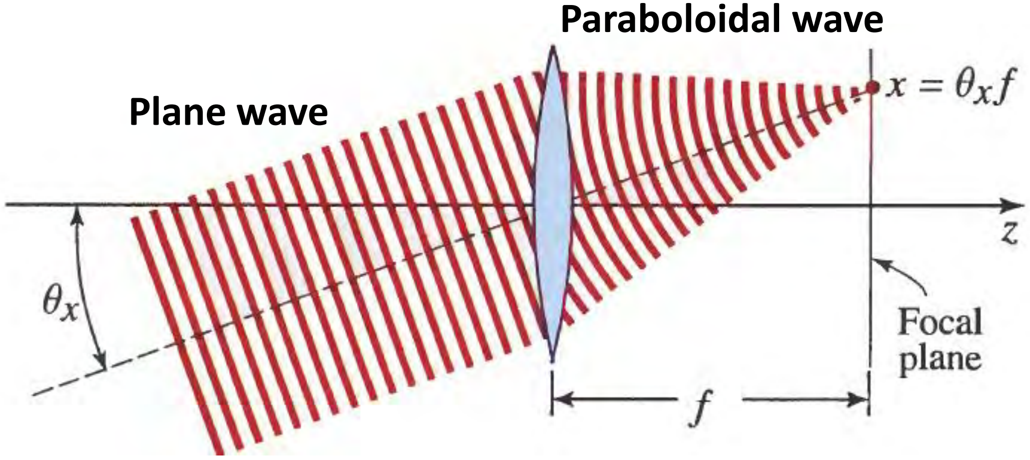
$N'_F = b^2/\lambda d$ **Input plane** confined to a circle of **radius b**

$N_F = \frac{a^2}{\lambda d}$ **Output plane** confined to a circle of **radius a**

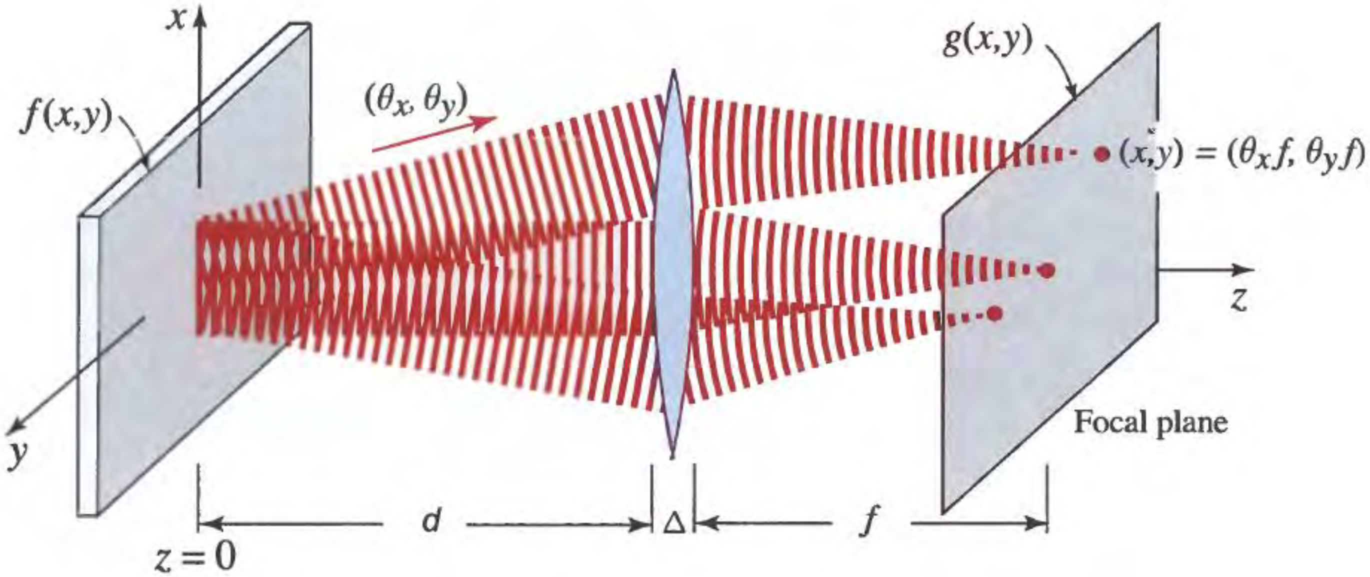
If the **propagation distance d is sufficiently long**, the only plane wave that contributes to the complex amplitude at a point (x, y) in the output plane, is the **wave with direction making angles $\boldsymbol{\theta_x = \frac{x}{d}}$ and $\boldsymbol{\theta_y = \frac{y}{d}}$ with the optical axis**

Paraxial approximation: $\theta_x \approx \lambda\nu_x, \quad \theta_y \approx \lambda\nu_y \Rightarrow \nu_x = \frac{x}{\lambda d} \quad \nu_y = \frac{y}{\lambda d}$

Fourier-Transform Property of a Lens



Lens maps each direction (θ_x, θ_y) into a single point $(\theta_x f, \theta_y f)$ In the focal plane



Regardless of d: $$I(x,y) = \frac{1}{(\lambda f)^2}\left|F\left(\frac{x}{\lambda f}, \frac{y}{\lambda f}\right)\right|^2$$

Assuming paraxial waves and using Fresnel approximation:

$$g(x,y) = h_l \underbrace{\exp\left[j\pi\frac{(x^2+y^2)(d-f)}{\lambda f^2}\right]}_{\text{Phase factor quadratic function}} F\left(\frac{x}{\lambda f}, \frac{y}{\lambda f}\right) \xrightarrow{d=f}$$

$$h_l = H_0 h_0 = (j/\lambda f)\exp[-jk(d+f)]$$

2f system

$$\boxed{g(x,y) = h_l F\left(\frac{x}{\lambda f}, \frac{y}{\lambda f}\right)}$$

$$h_l = (j/\lambda f)\exp(-j2kf)$$

FT, DTFT & DFT

Continuous Fourier Transform - FT

$$\omega = \frac{2\pi}{T} = 2\pi f$$

$$X(f) = \mathcal{F}\{x\}(f) = \int_{-\infty}^{\infty} x(t)e^{-i2\pi ft}dt$$

Sampling in time domain
Periodic in frequency domain

Discrete Time Fourier Transform - DTFT

$$X_{1/T}(f) = \mathcal{F}\left\{\sum_{n=-\infty}^{\infty} x[n]\delta(t-nT)\right\}(f) = \sum_{n=-\infty}^{\infty} x[n]e^{-i2\pi fnT}$$

Poisson Formula:

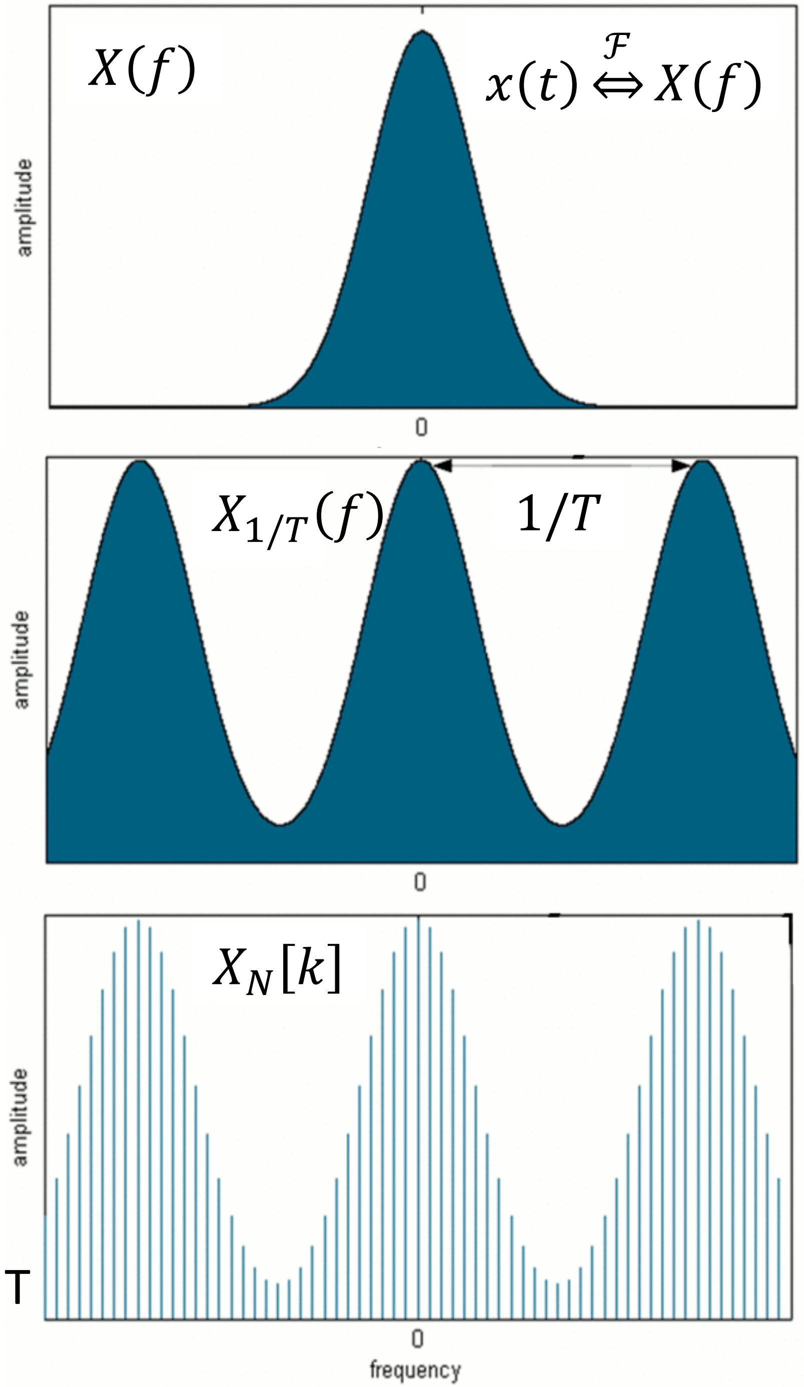
FT and DTFT relation

$$\sum_{n=-\infty}^{\infty} x[n]e^{-i2\pi fnT} = \sum_{k=-\infty}^{\infty} X(f-k/T)$$

Discrete Fourier Transform - DFT

$$X_N[k] = X_{1/T}\left({}^{k}/_{NT}\right) = \sum_{n=0}^{N-1} x[n]e^{-i2\pi kn/N}$$

Sampling in both time and frequency domain



N samples per cycle T

DFT properties

Invertible linear transformation

$$X_N[k] = \sum_{n=0}^{N-1} x[n] e^{-i2\pi kn/N} \quad \& \quad x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X_N[k] e^{i2\pi kn/N}$$

$$\mathbf{F}^{-1} = \frac{1}{N}\mathbf{F}^* \qquad \omega_N = e^{-2\pi i/N}$$

$$\mathbf{F} = \begin{bmatrix} \omega_N^{0\cdot 0} & \omega_N^{0\cdot 1} & \cdots & \omega_N^{0\cdot (N-1)} \\ \omega_N^{1\cdot 0} & \omega_N^{1\cdot 1} & \cdots & \omega_N^{1\cdot (N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_N^{(N-1)\cdot 0} & \omega_N^{(N-1)\cdot 1} & \cdots & \omega_N^{(N-1)\cdot (N-1)} \end{bmatrix}$$

Orthogonal basis

$$u_k = \left[e^{-i2\pi kn/N} \mid n = 0,1,\dots,N-1\right]^T$$

N-periodic

$$X_N[k] = X_N[k+N] \quad \& \quad \boxed{x[n] = x[n+N]}$$

The signal must be periodic – if not it is concatenated

$$F \cdot F^* = I$$

Orthonormal basis – Unitary DFT matrix

$$u_k = \left[\frac{1}{\sqrt{N}} e^{-i2\pi kn/N} \mid n = 0,1,\dots,N\right]^T$$

Signal energy unchanged

Parseval's Theorem – Energy conservation

$$\sum_{n=0}^{N-1} |x[n]|^2 = \sum_{k=0}^{N-1} |X[k]|^2$$

Translation

$$x[n] \to x[n-m] \overset{\mathcal{F}}{\Leftrightarrow} X[k] \to X[k]e^{-i2\pi km/N}$$

$$X[k] \to X[k-p] \overset{\mathcal{F}}{\Leftrightarrow} x[n] \to x[n]e^{i2\pi pn/N}$$

Convolution

$$x[n] * y[n] \overset{\mathcal{F}}{\Leftrightarrow} X[k]Y[k]$$

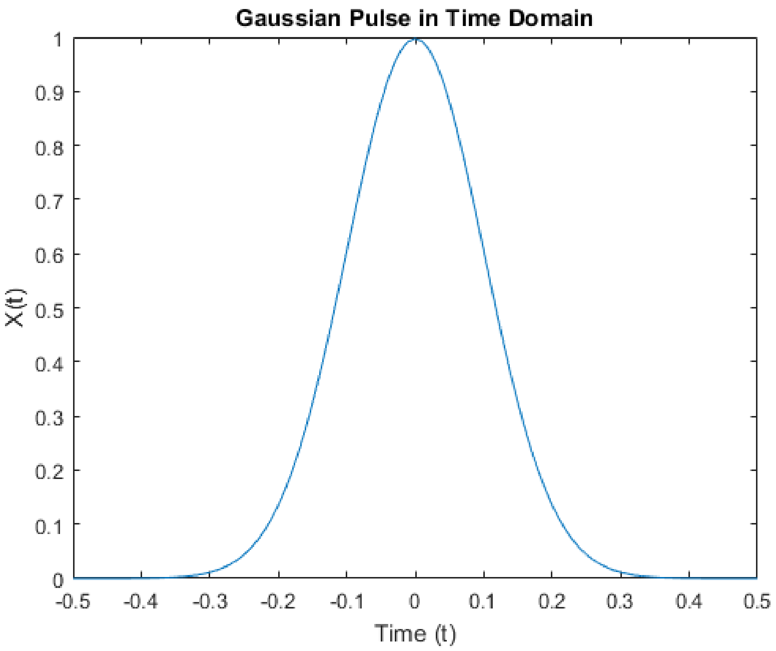
$$X[k] * Y[k] \overset{\mathcal{F}}{\Leftrightarrow} N \cdot x[n] \cdot y[n]$$

Fast Fourier Transform (FFT)

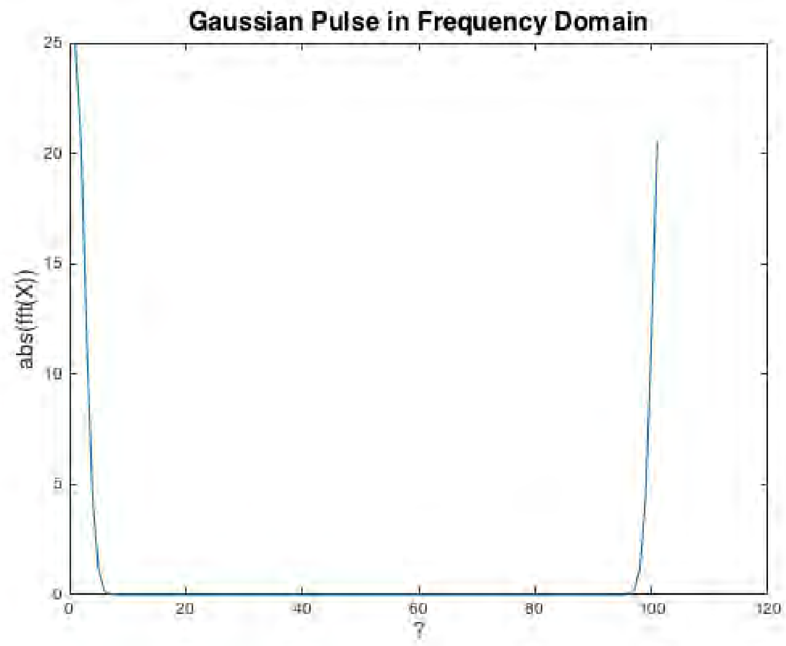
Algorithm that computes the DFT of a sequence

$$X_N[k] = \sum_{\boxed{n=1}}^{N} x[n]e^{-2\pi i(n-1)(k-1)/N}$$

Spectral shift:



FFT



x-axis in MATLAB:

0 1 2 ⋯ ⋯ ⋯ N/2-2 N/2-1 N/2 | -N/2 -(N/2-1) -(N/2-2) ⋯ ⋯ ⋯ -2 -1

Time to Frequency conversion:

n – Number of samples

$dt = 1/f_s$ & $T = n/f_s$ → $df = f_s/n$ & $F = f_s$

$dt = 1/f_s$	&	$T = n/f_s$		$df = f_s/n$	& $F = f_s$
Space between time samples	Sampling frequency	Total time $T = n \cdot dt$		Space between frequency samples	Max frequency

FFT in MATLAB

$dt = 1/f_s$ & $T = n/f_s$

Space between time samples | Sampling frequency | Total time $T = n \cdot dt$

n – Number of samples

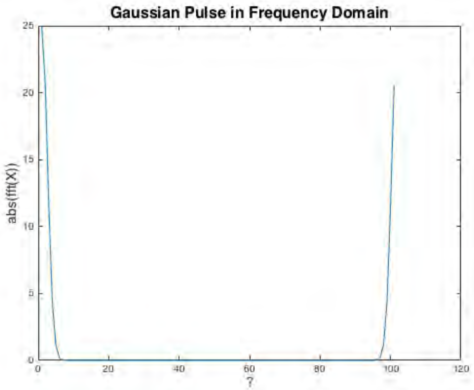
$df = f_s/n$ & $F = f_s$

Space between frequency samples | Max frequency

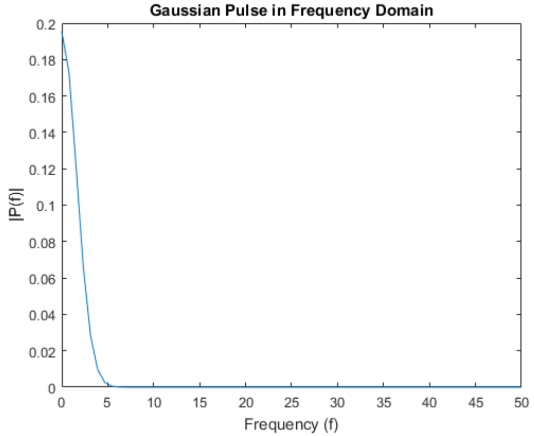
Quantity	Description
`x`	Sampled data
`n = length(x)`	Window length (number of samples)
`fs`	Samples/unit time
`dt = 1/fs`	Time increment per sample
`t = (0:m-1)/fs`	Time range for data
`y = fft(x,n)`	Discrete Fourier transform (DFT)
`abs(y)`	Amplitude of the DFT
`(abs(y).^2)/n`	Power of the DFT
`fs/n`	Frequency increment
`f = (0:n-1)*(fs/n)`	Frequency range
`fs/2`	Nyquist frequency

```
y0 = fftshift(y); % for visualizing the Fourier transform with the zero-frequency component in the middle of the spectrum.
```

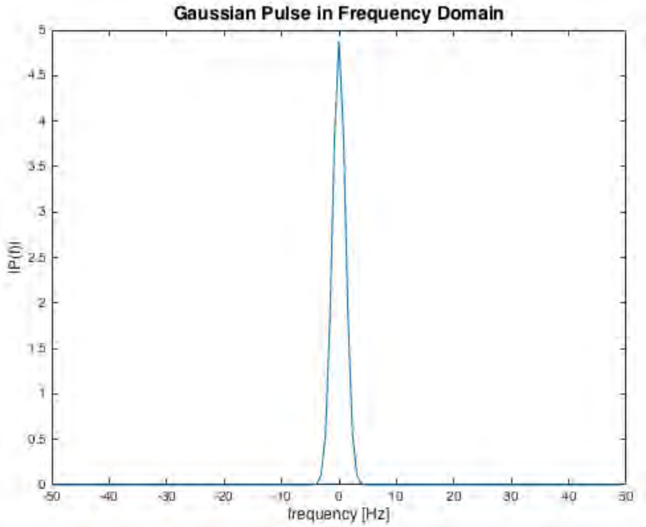
```
f0 = (-n/2:n/2-1)*(fs/n); % 0-centered frequency range
power0 = y0.*conj(y0)/n; % 0-centered power
```



Taking half the range



FFT shift



A Lens in MATLAB

1D DFT:

$$X_N[k] = \sum_{n=0}^{N-1} x[n] e^{-i2\pi kn/N}$$

2D DFT & spatial:

$$g[k,p] = F[k,p] = \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} f[n,m] e^{-2\pi i[nk+mp]/N}$$

$$f_{max,n} = \frac{1}{dn} \to df_n = \frac{f_{max,n}}{N}$$

$$\Rightarrow k \in df_n \cdot [0, \dots, N-1]$$

$f_{max,n}$: equivalent sampling frequency. A single measurement is performed *per* physical (pixel) size dn

Time:

$dt = 1/f_s$ & $T = n/f_s$

Space between time samples | Sampling frequency | Total time $T = n \cdot dt$

n – Number of samples

$df = f_s/n$ & $F = f_s$

Space between frequency samples | Max frequency

Lens FT (d=f)

$$g(x,y) = h_l F\left(\frac{x}{\lambda f}, \frac{y}{\lambda f}\right) = \iint_{-\infty}^{\infty} f(u,v) e^{-i2\pi\left(\frac{x}{\lambda f}u + \frac{y}{\lambda f}v\right)} dudv$$

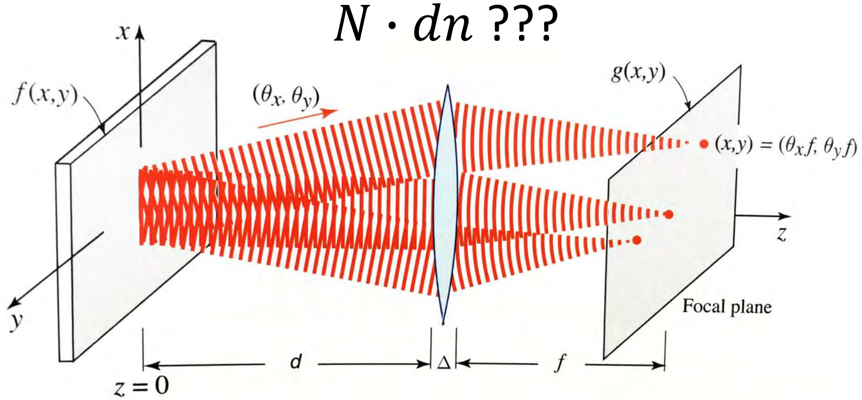
$$\Rightarrow k = \frac{n}{\lambda f} \to \frac{n}{\lambda f} \in df_n \cdot [0, \dots, N-1]$$

$$\boxed{n = \frac{\lambda f}{N \cdot dn} \cdot [0, \dots, N-1]}$$

Pixel size

That follows Nyquist

$N \cdot dn$???

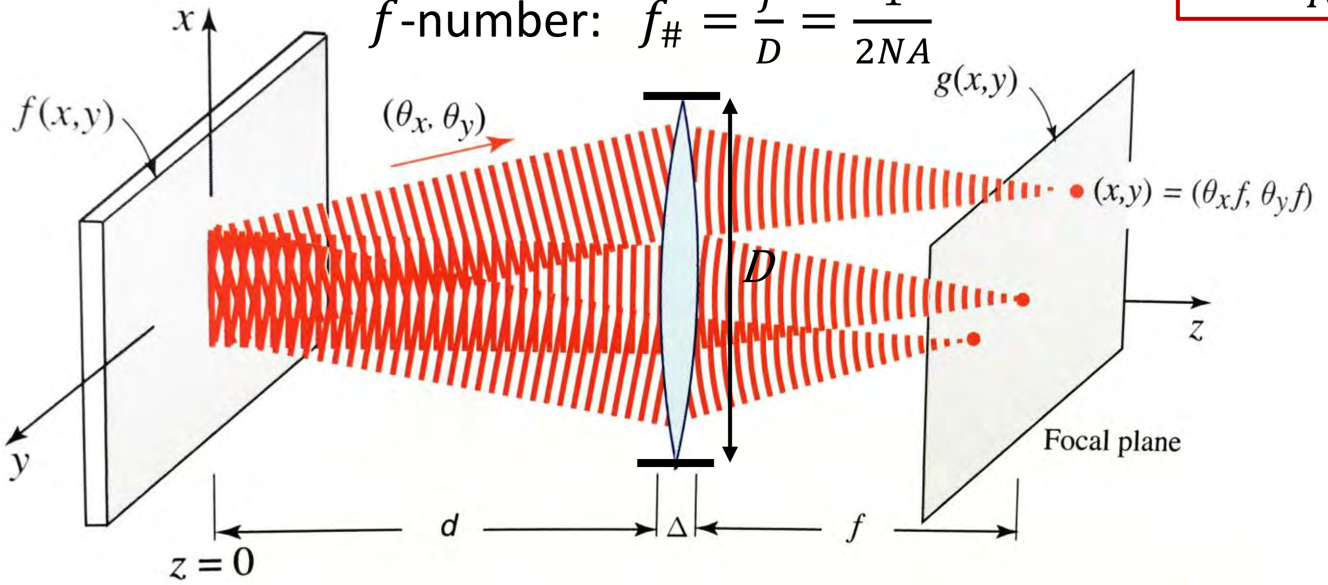


A Lens in MATLAB

$$n = \frac{\lambda f}{N \cdot dn} \cdot [0, \dots, N-1]$$

$N \cdot dn$???

f-number: $f_{\#} = \frac{f}{D} = \frac{1}{2NA}$



Diffraction Limit – Rayleigh Criterion: $\Delta d = \left(0.61 \frac{\lambda}{NA}\right)$

Nyquist Sampling: $dx = \left(0.61 \frac{\lambda}{NA}\right) \frac{1}{2} \implies dn = \lfloor dx \rfloor$

***N*: # of Fourier samples**

$$f_{max,n} = \frac{1}{dn} \rightarrow df_n = \frac{f_{max,n}}{N}$$

***N* determines the sampling resolution in the Fourier domain**. Technically, N is entirely determined by the lowest frequency component. In an $K \times K$ image, the "largest" feature is the size of the image itself. Therefore, $N = K$

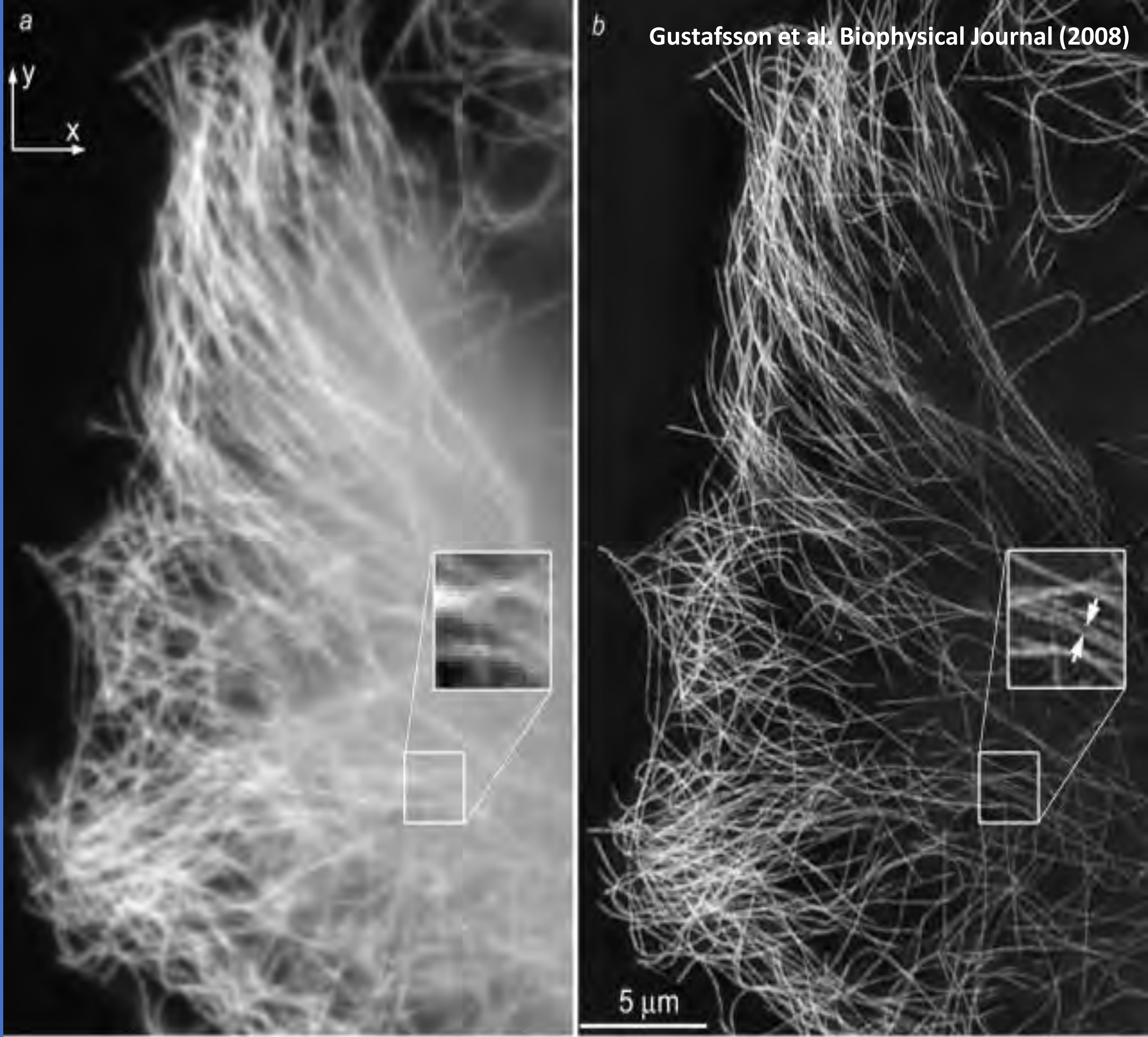
Should one desire to improve the Fourier domain sampling resolution, he may do so by **padding the image with additional elements** (thereby increasing the "numerical" image size)

Department of Biomedical Engineering,
Technion
Computational optical imaging 336547

Tutorial 4 – Structured Illumination Microscopy

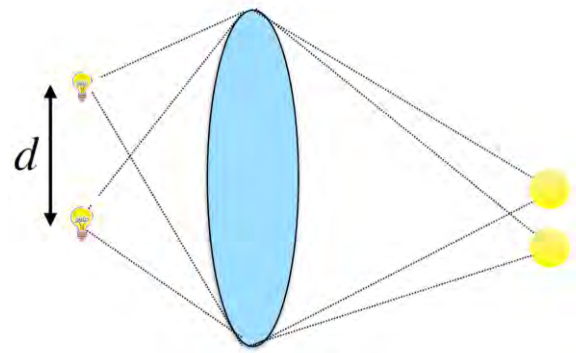
Elias Nehme & Yoav Shechtman

17 November 2020



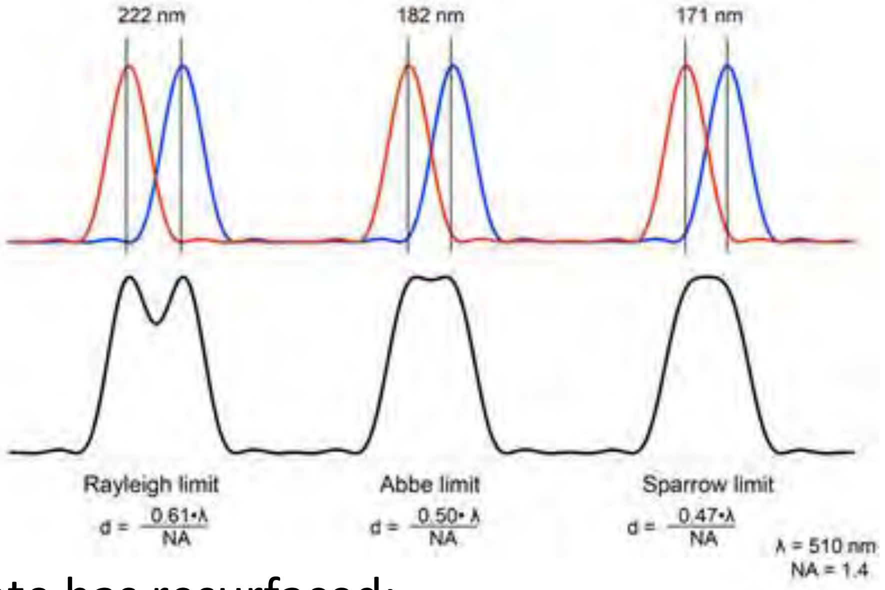
Spatial resolution

Subject of passionate scientific debate for decades:



The minimum **resolvable** distance between two point-sources emitting at the same time

Classical resolution definitions:



With the development of **fluorescence nanoscopy techniques** this debate has resurfaced:

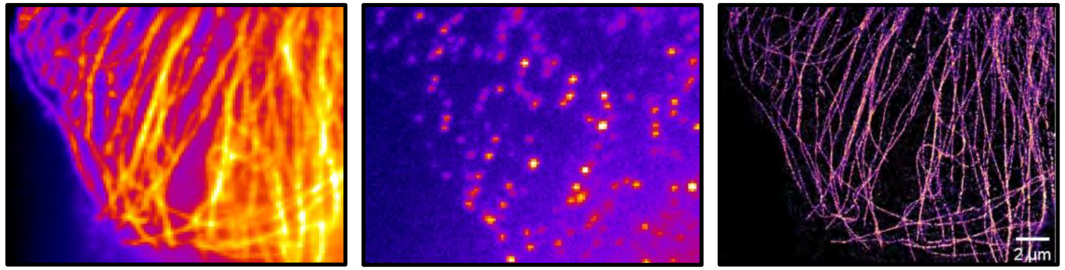
Response to Comment on "Extended-resolution structured illumination imaging of endocytic and cytoskeletal dynamics"

Li, Dong, and Eric Betzig, Science (2016)

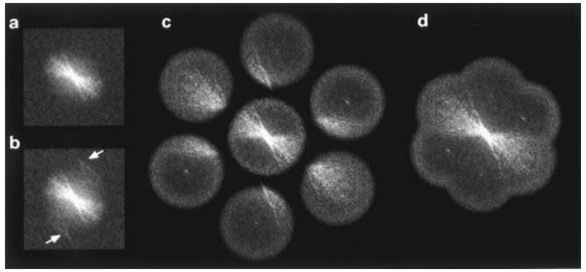
Sahl *et al*. in their Comment raise criticisms of our work that fall into three classes: image artifacts, resolution criteria, and comparative performance on live cells. We explore each of these in turn.

STED

STORM/PALM



SIM



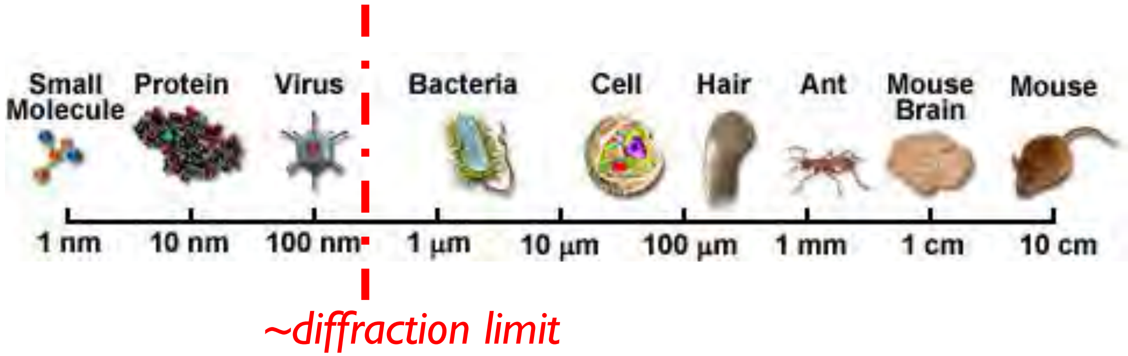
Spatial resolution – PSF and OTF

$$\text{Resolution}_{x,y} = \lambda/2[\eta \cdot \sin(\alpha)]$$

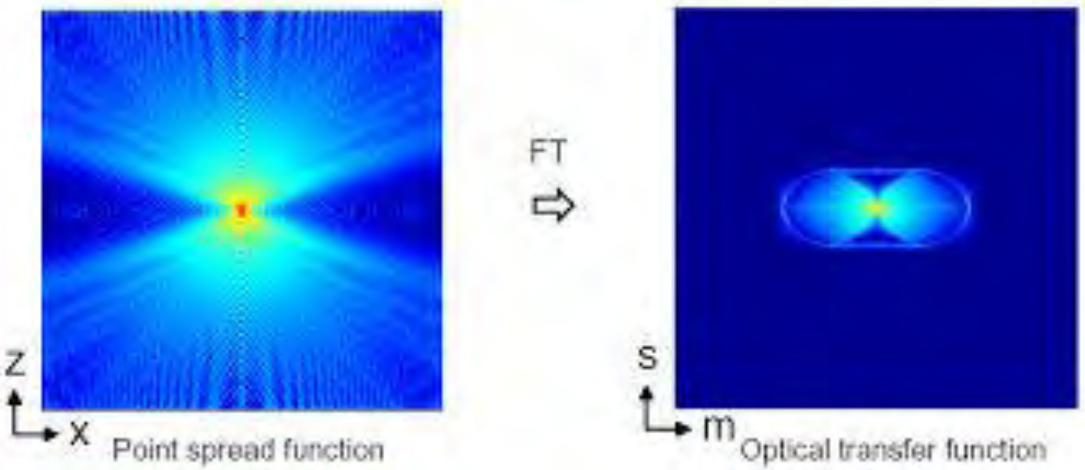
For visible light:
Lateral resolution~**200nm**

$$\text{Resolution}_{z} = 2\lambda/[\eta \cdot \sin(\alpha)]^2$$

Axial resolution~**500nm**



Convolution with the PSF acts as a **low-pass filter**



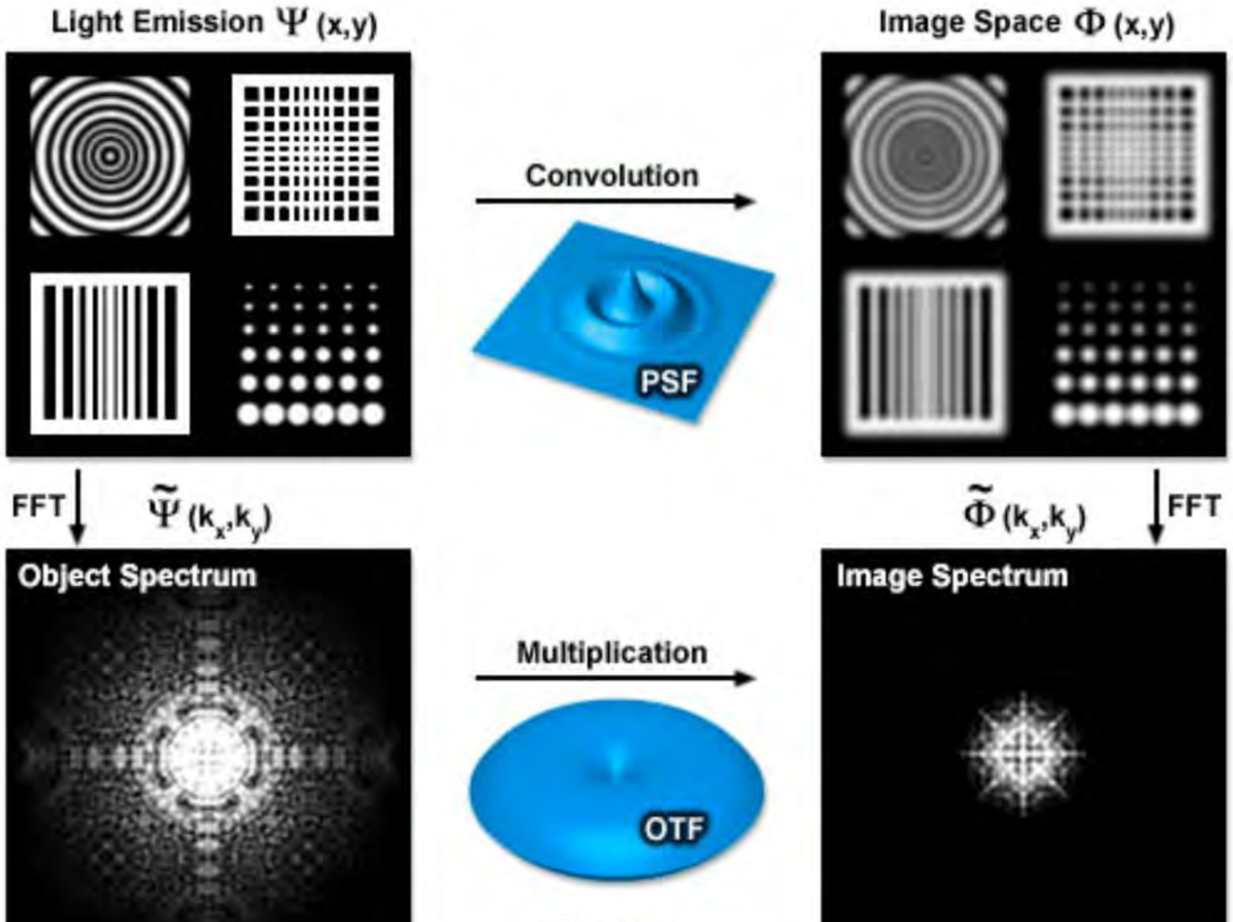


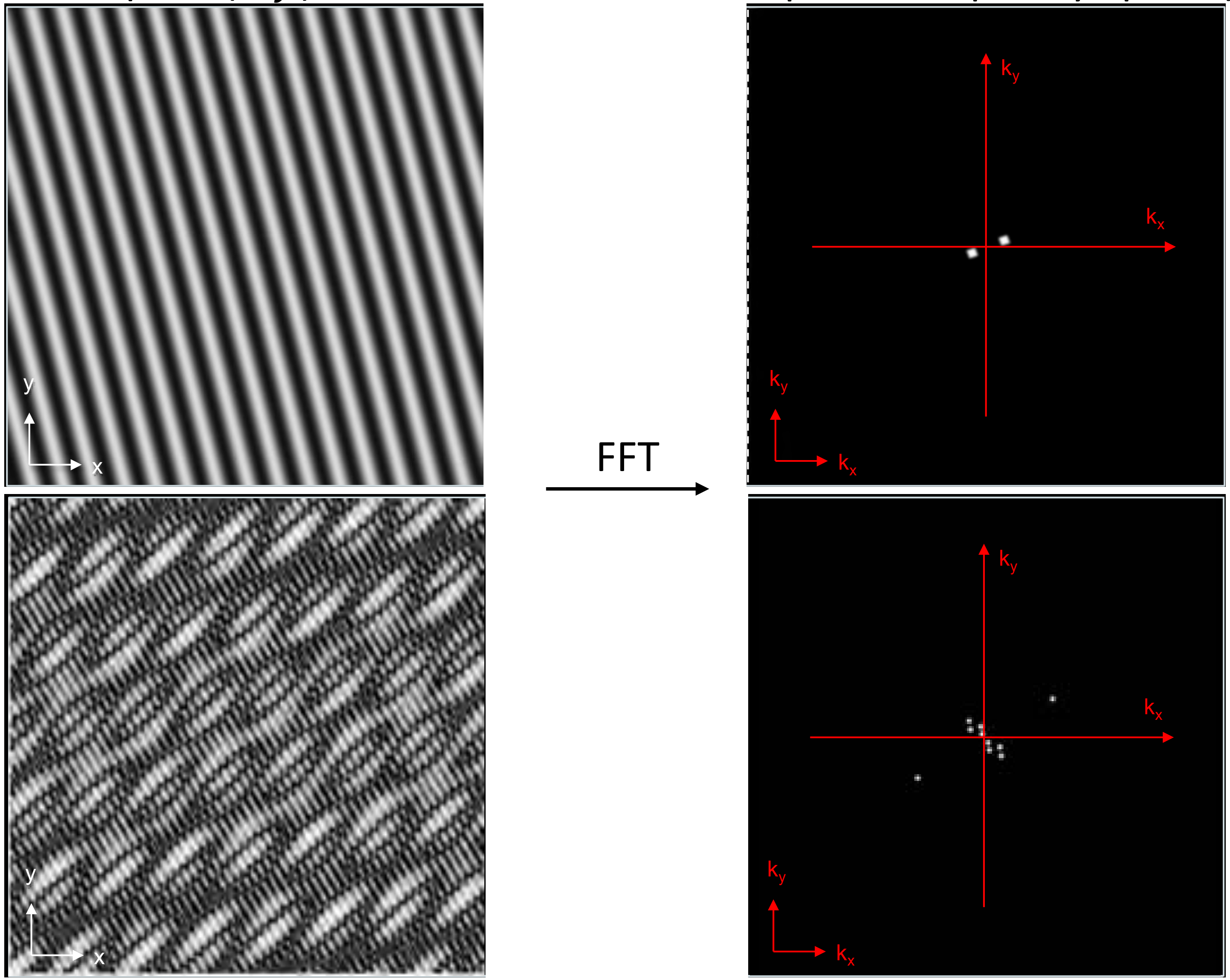
Figure 4

OTF is the normalized **Fourier transform of the PSF** of the optical system

http://zeiss-campus.magnet.fsu.edu/articles/superresolution/supersim.html

Harmonic Functions & Complex Amplitudes

Real space (x, y) Spatial Frequency Space (k_x, k_y)

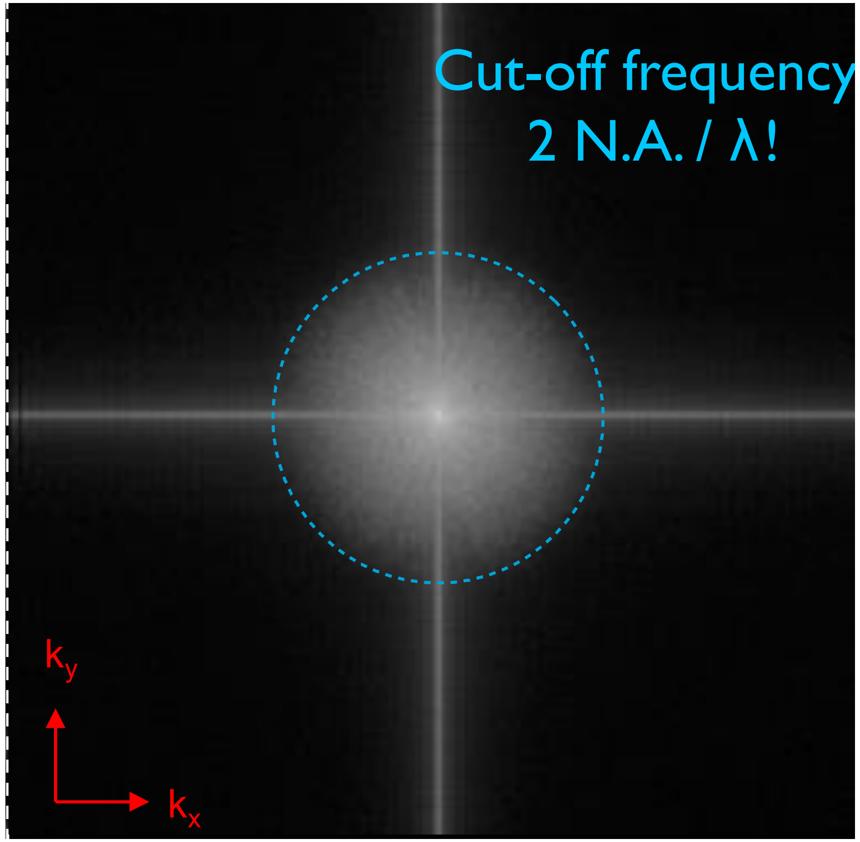


Harmonic Functions & Complex Amplitudes

Real space (x, y)

high N.A.

Spatial Frequency Space (k_x, k_y)

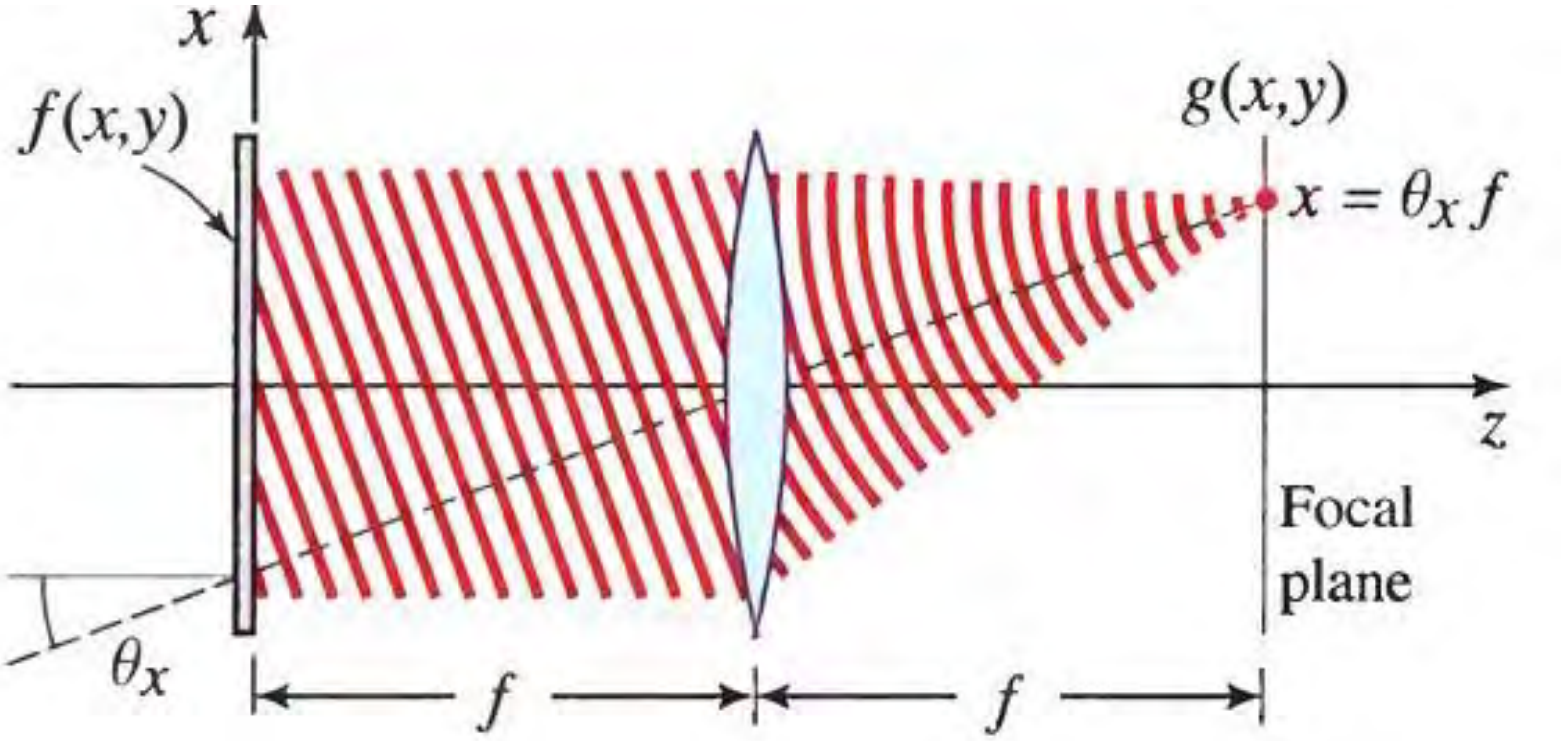


The **classical limit of resolution in the microscope translates into frequency space**, defining a maximum observable spatial frequency:

$k_0 = 2NA / \lambda_{em}$

→ An image may be analyzed as a **sum of harmonic functions of different spatial frequencies and complex amplitudes**

Fourier-Transform Lens Property – A Reminder



A "**spatial frequency**" → **frequency of a harmonic function** with which the image is analyzed (previous slides)

Each harmonic function in the image correspondingly **acts as a local diffraction grating**, thus producing plane waves traveling at **an angle with the optical axis**

$$f(x,y) \xrightarrow{\text{Fourier Transform}} F\left(k_x = \frac{x}{\lambda f}, k_y = \frac{y}{\lambda f}\right)$$

The lens subsequently performs a Fourier transform; consequently, **harmonic functions (complex exponentials) are transformed into spots**

In short: **the "finer" the image features**, the **higher the spatial frequency**, the "finer" the effective grating, the **larger the diffraction angle** and the **farther from the optical axis** is the focused spot on the focal (Fourier) plane

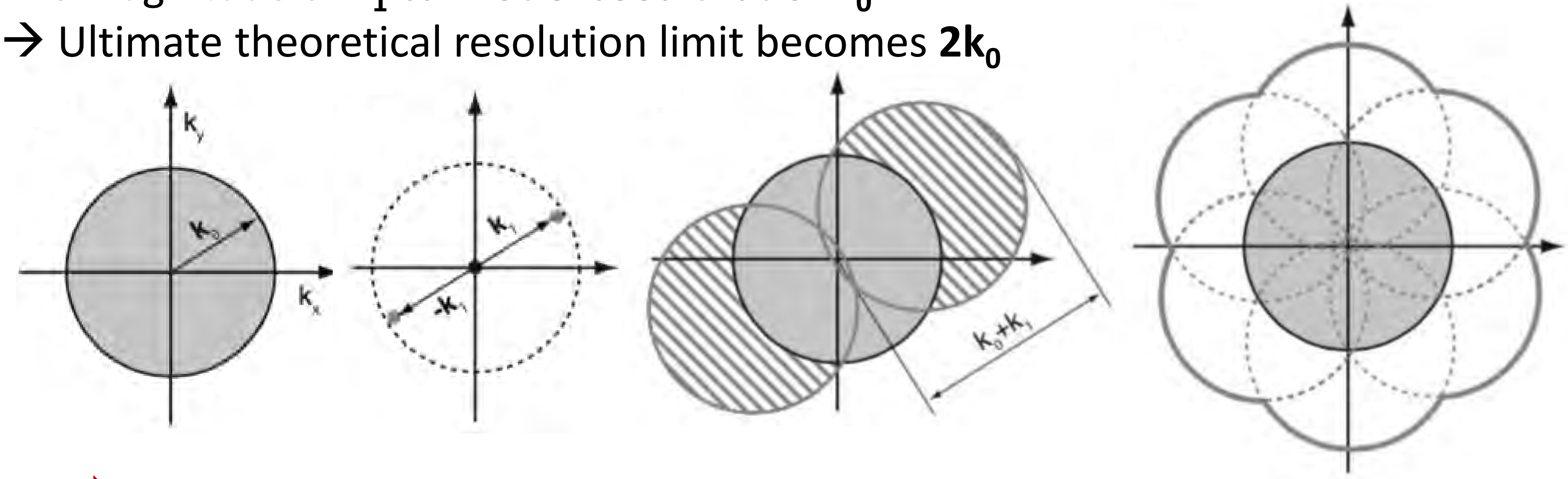
Structured Illumination Microscopy (SIM) – Concept

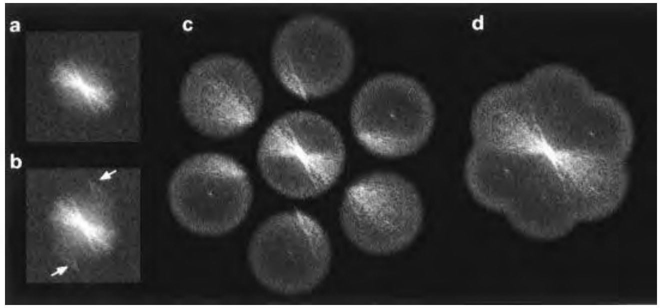
Artificially move unobservable high-frequency information into the observable region through frequency mixing with a known illumination structure

__Moiré fringes__ – (a) and (b) are two examples of fine patterns. When one is superimposed onto the other, a coarser beat pattern—Moiré fringes—appears (c)

→ **Encoding high frequency information in the form of lower (observable) frequency components**

Resolution is stretched from k_0 to $k_1 + k_0$
The magnitude of **k_1 cannot exceed that of k_0**
→ Ultimate theoretical resolution limit becomes **$2k_0$**





Rotated through steps of 120 degrees

2x lateral resolution improvement over diffraction-limited imaging

SIM – Formulation

Sinusoidal Illumination Intensity

$$I_{\theta,\phi}(\mathbf{r}) = I_o\left[1 - \frac{m}{2}\cos\left(2\pi\mathbf{p}_\theta \cdot \mathbf{r} + \phi\right)\right] \qquad (1)$$

Observed Fluorescence Emission

$$D_{\theta,\phi}(\mathbf{r}) = [S(\mathbf{r})I_{\theta,\phi}(\mathbf{r})] \otimes H(\mathbf{r}) + N(\mathbf{r}) \qquad (2)$$

Observed Fluorescence Emission in Frequency Domain

$$\begin{aligned}\tilde{D}_{\theta,\phi}(\mathbf{k}) &= \left[\tilde{I}_{\theta,\phi}(\mathbf{k}) \otimes \tilde{S}(\mathbf{k})\right] \cdot \tilde{H}(\mathbf{k}) + \tilde{N}(\mathbf{k}) \\ &= \frac{I_o}{2}\left[\tilde{S}(\mathbf{k}) - \frac{m}{2}\tilde{S}(\mathbf{k} - \mathbf{p}_\theta)e^{-i\phi}\right. \\ &\qquad \left.-\frac{m}{2}\tilde{S}(\mathbf{k} + \mathbf{p}_\theta)e^{i\phi}\right] \cdot \tilde{H}(\mathbf{k}) + \tilde{N}(\mathbf{k}) \qquad (3)\end{aligned}$$

Obtaining 3 elements in k space – the original one and 2 more which are shifted versions of the original

$I_{\theta,\phi}(\boldsymbol{r})$: illuminating sinusoidal intensity pattern
$\boldsymbol{r} = (x, y)$: spatial position vector
I_0 : peak illumination
$\boldsymbol{p}_\theta = (p \cdot \cos\theta \;, p \cdot \sin\theta)$: sinusoidal illumination frequency vector in reciprogal space
θ: orientation of sinusoidal illumination pattern
ϕ: phase of sinusoidal illumination pattern
m: modulation factor

$S(\boldsymbol{r})$: Fluorophore density distribution within specimen
$D_{\theta,\phi}(\boldsymbol{r})$: observed fluorescence emission through the optical system
$H(\boldsymbol{r})$: optical system's PSF
$N(\boldsymbol{r})$: additive Gaussian (white) noise

$\tilde{H}(\boldsymbol{k})$: Optical system's OTF

SIM – Formulation

Three SIM images – $D_{\theta,\phi_1}(\boldsymbol{r}), D_{\theta,\phi_2}(\boldsymbol{r})$ and $D_{\theta,\phi_3}(\boldsymbol{r})$– of the specimen are acquired, corresponding to **three different illumination phases**; typically $\phi_1 = 0°$, $\phi_2 = 120°$ and $\phi_3 = 240°$

$$\begin{aligned}\tilde{D}_{\theta,\phi}(\mathbf{k}) &= \left[\tilde{I}_{\theta,\phi}(\mathbf{k}) \otimes \tilde{S}(\mathbf{k})\right] \cdot \tilde{H}(\mathbf{k}) + \tilde{N}(\mathbf{k}) \\ &= \frac{I_o}{2}\left[\tilde{S}(\mathbf{k}) - \frac{m}{2}\tilde{S}(\mathbf{k}-\mathbf{p}_\theta)e^{-i\phi} \right. \\ &\qquad \left. -\frac{m}{2}\tilde{S}(\mathbf{k}+\mathbf{p}_\theta)e^{i\phi}\right] \cdot \tilde{H}(\mathbf{k}) + \tilde{N}(\mathbf{k})\end{aligned}$$

$$\begin{bmatrix} \tilde{D}_{\theta,\phi_1}(\mathbf{k}) \\ \tilde{D}_{\theta,\phi_2}(\mathbf{k}) \\ \tilde{D}_{\theta,\phi_3}(\mathbf{k}) \end{bmatrix} = \frac{I_o}{2}\mathbf{M} \begin{bmatrix} \tilde{S}(\mathbf{k})\tilde{H}(\mathbf{k}) \\ \tilde{S}(\mathbf{k}-\mathbf{p}_\theta)\tilde{H}(\mathbf{k}) \\ \tilde{S}(\mathbf{k}+\mathbf{p}_\theta)\tilde{H}(\mathbf{k}) \end{bmatrix} + \begin{bmatrix} \tilde{N}_{\theta,\phi_1}(\mathbf{k}) \\ \tilde{N}_{\theta,\phi_2}(\mathbf{k}) \\ \tilde{N}_{\theta,\phi_3}(\mathbf{k}) \end{bmatrix}$$

$$\text{where} \quad \mathbf{M} = \begin{bmatrix} 1 & -\frac{m}{2}e^{-i\phi_1} & -\frac{m}{2}e^{+i\phi_1} \\ 1 & -\frac{m}{2}e^{-i\phi_2} & -\frac{m}{2}e^{+i\phi_2} \\ 1 & -\frac{m}{2}e^{-i\phi_3} & -\frac{m}{2}e^{+i\phi_3} \end{bmatrix} \qquad (4)$$

$$\begin{matrix}\text{noisy} \\ \text{estimate} \\ \text{of}\end{matrix} \begin{bmatrix} \tilde{S}(\mathbf{k})\tilde{H}(\mathbf{k}) \\ \tilde{S}(\mathbf{k}-\mathbf{p}_\theta)\tilde{H}(\mathbf{k}) \\ \tilde{S}(\mathbf{k}+\mathbf{p}_\theta)\tilde{H}(\mathbf{k}) \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} \tilde{D}_{\theta,\phi_1}(\mathbf{k}) \\ \tilde{D}_{\theta,\phi_2}(\mathbf{k}) \\ \tilde{D}_{\theta,\phi_3}(\mathbf{k}) \end{bmatrix} \qquad (5)$$

$$\mathbf{M}^{-1} = \frac{1}{\Delta} \times \begin{bmatrix} e^{i(\phi_2-\phi_3)} - e^{i(\phi_3-\phi_2)} & e^{i(\phi_3-\phi_1)} - e^{i(\phi_1-\phi_3)} & e^{i(\phi_2-\phi_1)} - e^{i(\phi_1-\phi_2)} \\ \frac{2}{m}\left(e^{i\phi_3} - e^{i\phi_2}\right) & \frac{2}{m}\left(e^{i\phi_1} - e^{i\phi_3}\right) & \frac{2}{m}\left(e^{i\phi_2} - e^{i\phi_1}\right) \\ \frac{2}{m}\left(e^{-i\phi_2} - e^{-i\phi_3}\right) & \frac{2}{m}\left(e^{-i\phi_3} - e^{-i\phi_1}\right) & \frac{2}{m}\left(e^{-i\phi_1} - e^{-i\phi_2}\right) \end{bmatrix}$$

$$\text{where} \quad \Delta = \left[e^{i(\phi_2-\phi_1)} - e^{i(\phi_1-\phi_2)} - e^{i(\phi_3-\phi_1)} + e^{i(\phi_1-\phi_3)} + e^{i(\phi_3-\phi_2)} - e^{i(\phi_2-\phi_3)}\right]$$

Subsequently, the ungraded approximations of $\tilde{S}(\boldsymbol{k})$, $\tilde{S}(\boldsymbol{k} - \boldsymbol{p}_\theta)$ and $\tilde{S}(\boldsymbol{k} + \boldsymbol{p}_\theta)$ are obtained by **Wiener Filtering** of their corresponding noisy estimates obtained by Eq. 5

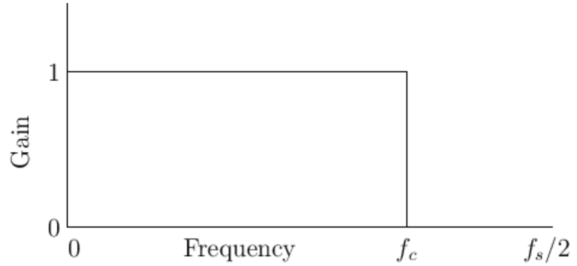
Weiner filter – A Reminder

Consider a convolution system:

$$x \rightarrow \boxed{h} \rightarrow y$$

$$y = h * x \longrightarrow y_f = h_f \cdot x_f \longrightarrow \text{What about } \hat{x}_f = y_f / h_f\text{?}$$

We can't just divide in frequency domain because there are zeros in h_f



Real world:

$$y_f = h_f \cdot x_f + n$$

Wiener filter: "regularize" the problem $\hat{x}_f = \dfrac{h_f^*}{|h_f|^2 + \frac{1}{SNR_f}} \cdot y_f = \dfrac{1}{h_f}\left[\dfrac{|h_f|^2}{|h_f|^2 + \frac{1}{SNR_f}}\right] \cdot y_f$

This **suppresses frequencies where the SNR is low** (high noise).

And **acts as an inverse filter where the noise is negligible**.

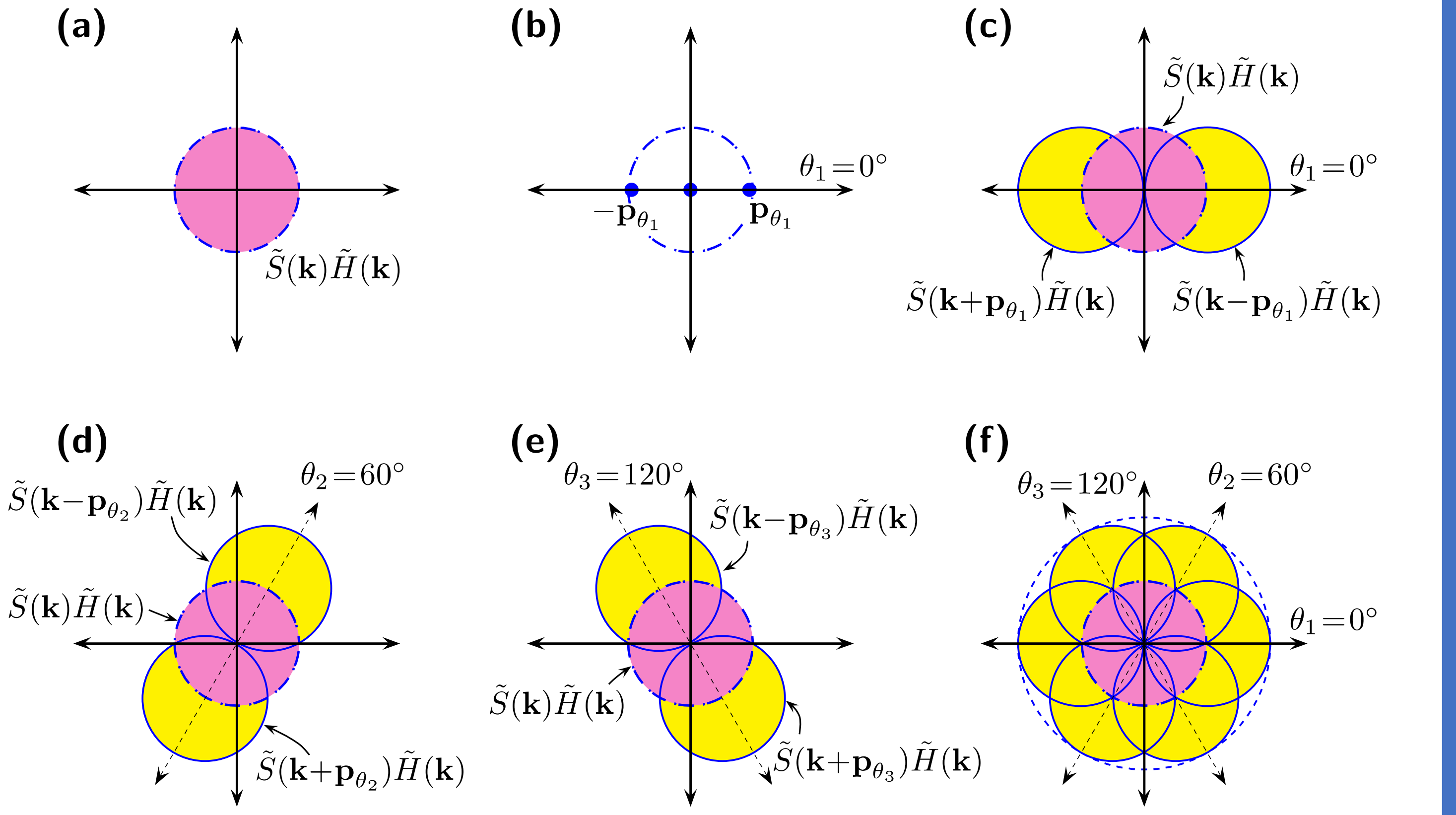
SIM – Formulation

$$\begin{array}{r} \text{noisy} \\ \text{estimate} \\ \text{of} \end{array} \begin{bmatrix} \tilde{S}(\mathbf{k})\tilde{H}(\mathbf{k}) \\ \tilde{S}(\mathbf{k}-\mathbf{p}_\theta)\tilde{H}(\mathbf{k}) \\ \tilde{S}(\mathbf{k}+\mathbf{p}_\theta)\tilde{H}(\mathbf{k}) \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} \tilde{D}_{\theta,\phi_1}(\mathbf{k}) \\ \tilde{D}_{\theta,\phi_2}(\mathbf{k}) \\ \tilde{D}_{\theta,\phi_3}(\mathbf{k}) \end{bmatrix}$$

Finally, the centers of the frequency components $\tilde{S}(\boldsymbol{k}-\boldsymbol{p}_\theta)$ and $\tilde{S}(\boldsymbol{k}+\boldsymbol{p}_\theta)$ are **sub-pixelly shifted** by $+\boldsymbol{p}_\theta$ and $-\boldsymbol{p}_\theta$, respectively, in the reciprocal (Freq.) space (Fig. (c))

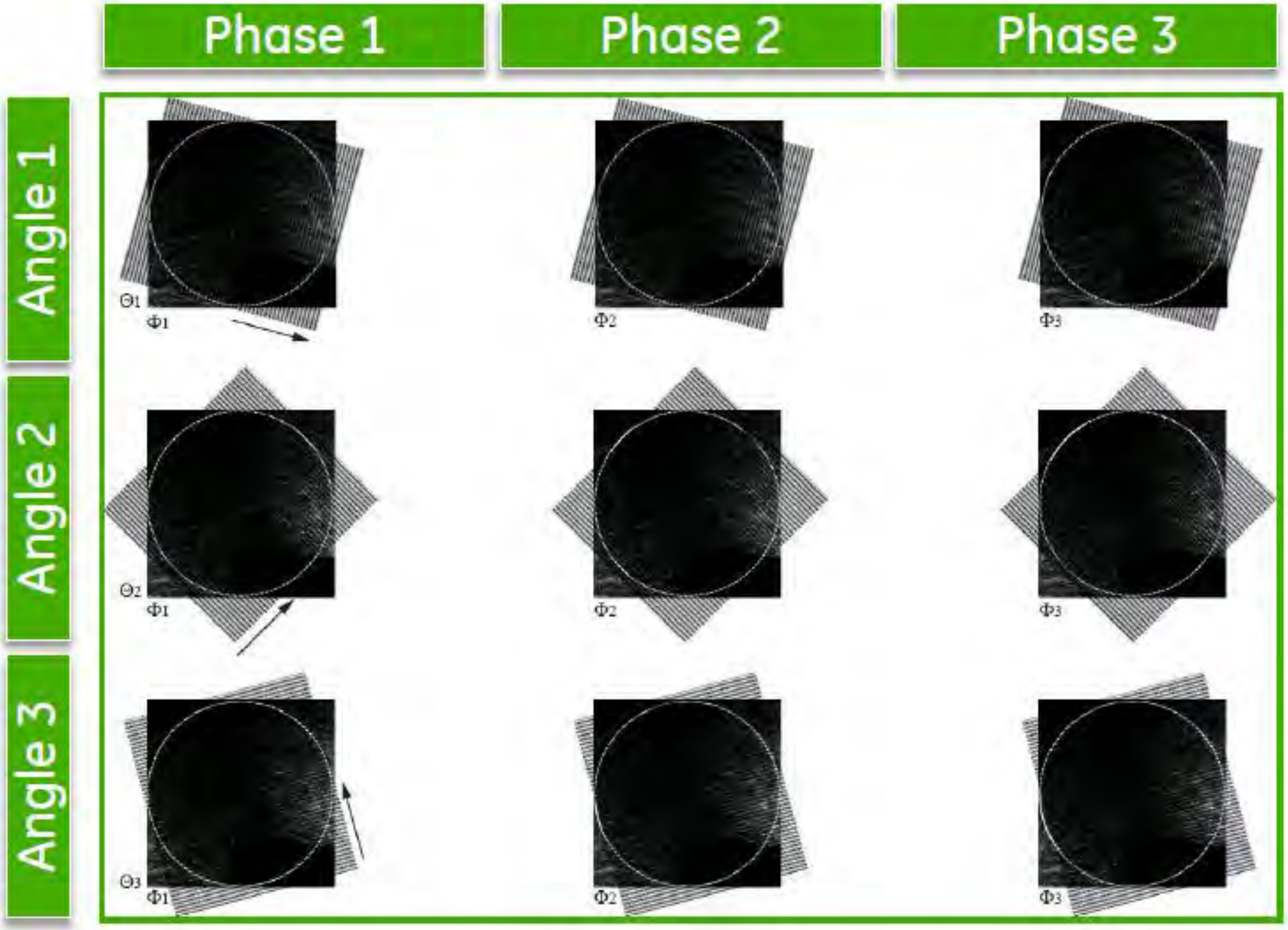
By **changing the angular orientation θ of the illuminating sinusoidal pattern** ($\theta_1 = 0°$, $\theta_2 = 60°$ and $\theta_3 = 120°$ suffices), and by **repeating the above procedure**, (almost) all frequency content of specimen lying within a circular region of radius twice of that governed by the OTF of optical system may be computed (Fig. (f))

→ Enabling **spatial reconstruction of specimen with twice the resolution** than that which is directly obtainable using the same optical system

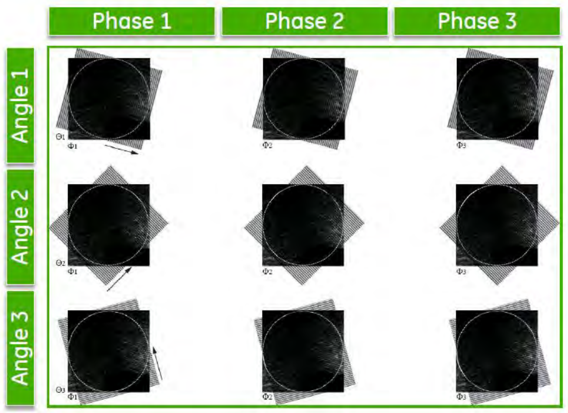


Raw SIM images

Shift pattern through 3 phases at 3 angles (total 9)



SIM – Experimental Procedure



1. **Acquisition of raw SIM images (3X3)**

2. **System OTF determination**

   Intensity distribution of **hundreds of 100nm fluorescent microspheres** super-imposed and averaged to obtain an **approximation of the system PSF** → **Fourier Transform** of this PSF provides an estimate of **system OTF**

$$\begin{bmatrix} \tilde{S}(\mathbf{k})\tilde{H}(\mathbf{k}) \\ \tilde{S}(\mathbf{k}-\mathbf{p}_\theta)\tilde{H}(\mathbf{k}) \\ \tilde{S}(\mathbf{k}+\mathbf{p}_\theta)\tilde{H}(\mathbf{k}) \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} \tilde{D}_{\theta,\phi_1}(\mathbf{k}) \\ \tilde{D}_{\theta,\phi_2}(\mathbf{k}) \\ \tilde{D}_{\theta,\phi_3}(\mathbf{k}) \end{bmatrix}$$

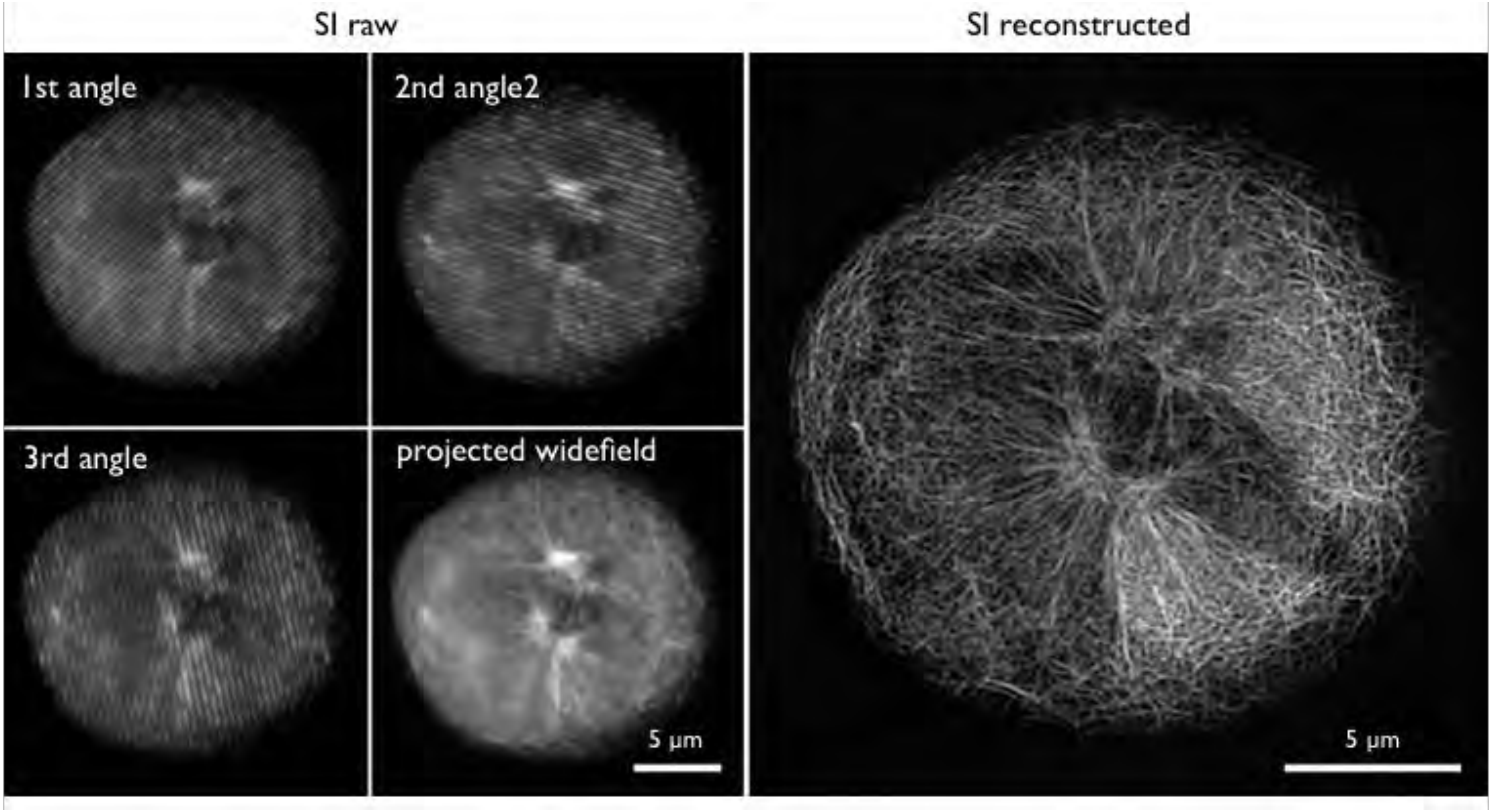
$\tilde{H}(\boldsymbol{k})$: Optical system's OTF

3. **Preprocessing of raw SIM images**

   a. **Intensity normalization**: Raw SIM images re-scaled to have identical global mean and standard deviation (bleaching, intensity fluctuations, differences in intensity between illumination pattern angles, total intensity and motion variation)
   b. **Background removal**: morphological operators, scaled substraction
   c. **Image Processing & Filtering**

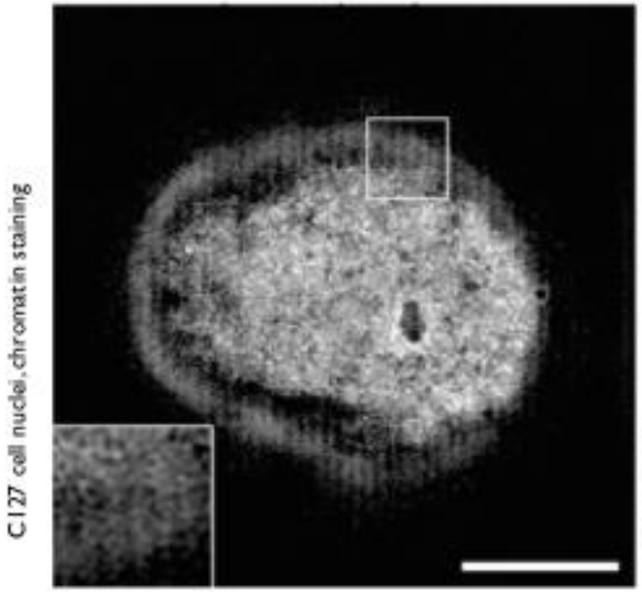
4. **Reconstruction of high resolution image using SIM-RA**

SIM – Example


SI raw
SI reconstructed
1st angle
2nd angle2
3rd angle
projected widefield
5 µm
5 µm

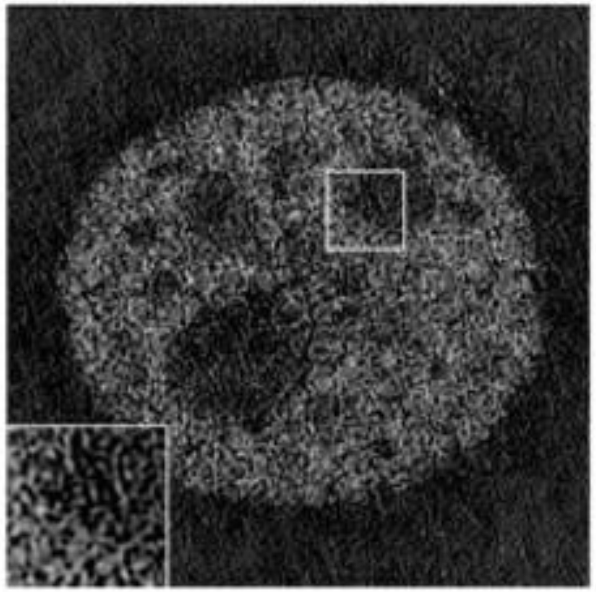
SIM – Reconstruction Artifacts

Stripes



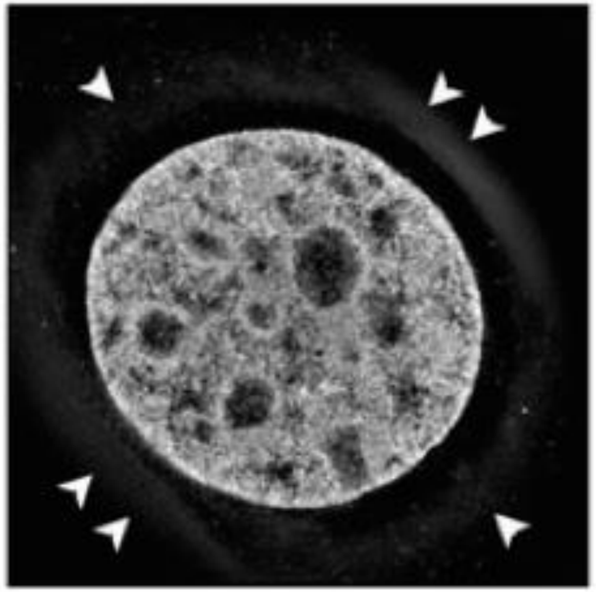
Bleaching, Drift or vibrations,
Moving particles

High frequency noise

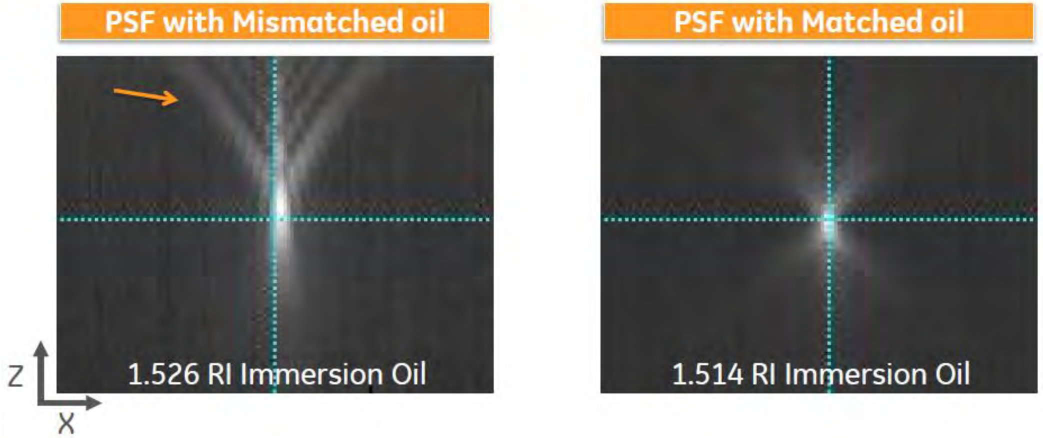


Low contrast-to-noise,
Low modulation
contrast

Halo / Doubling



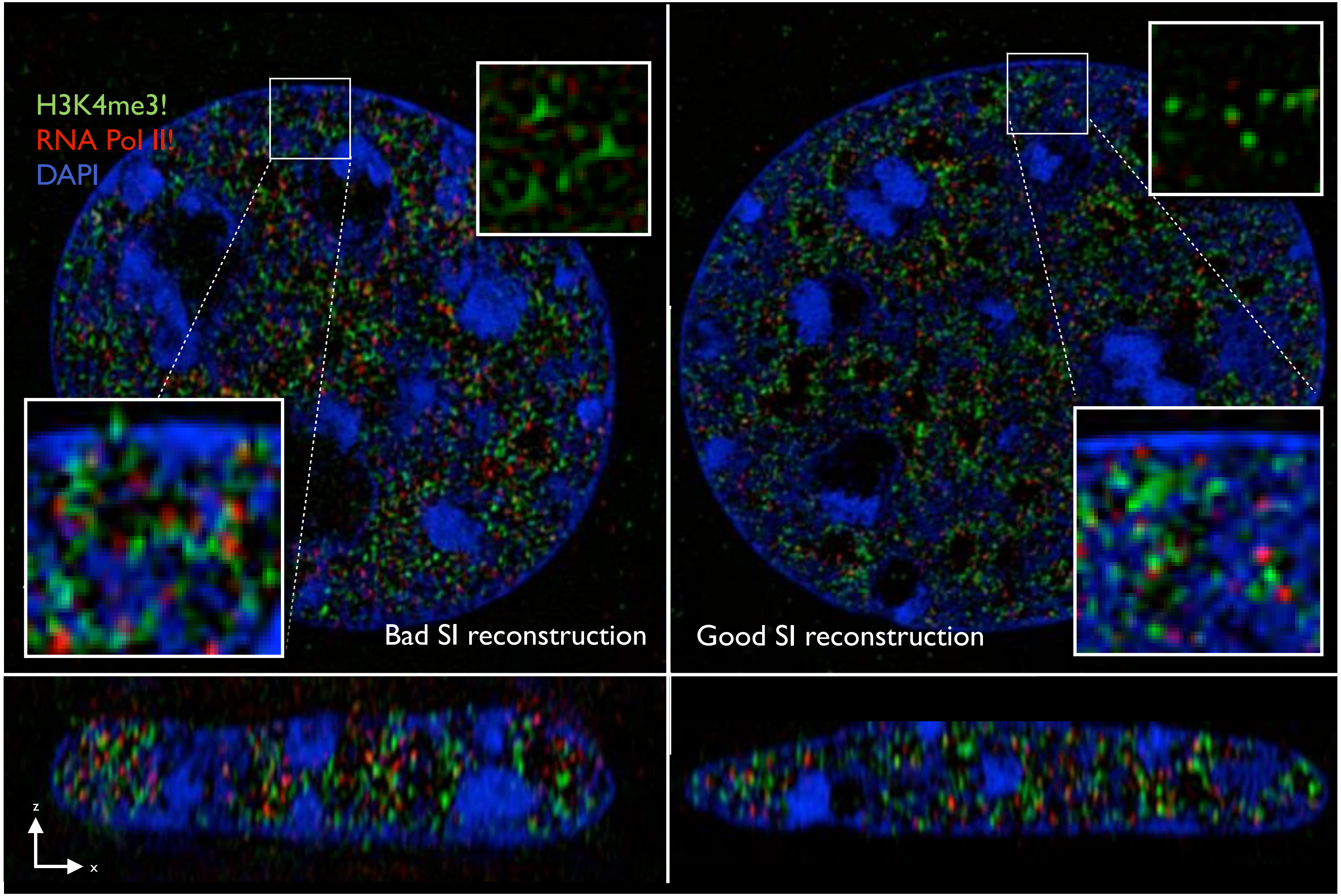
Spherical aberration caused by
Refractive index mismatch



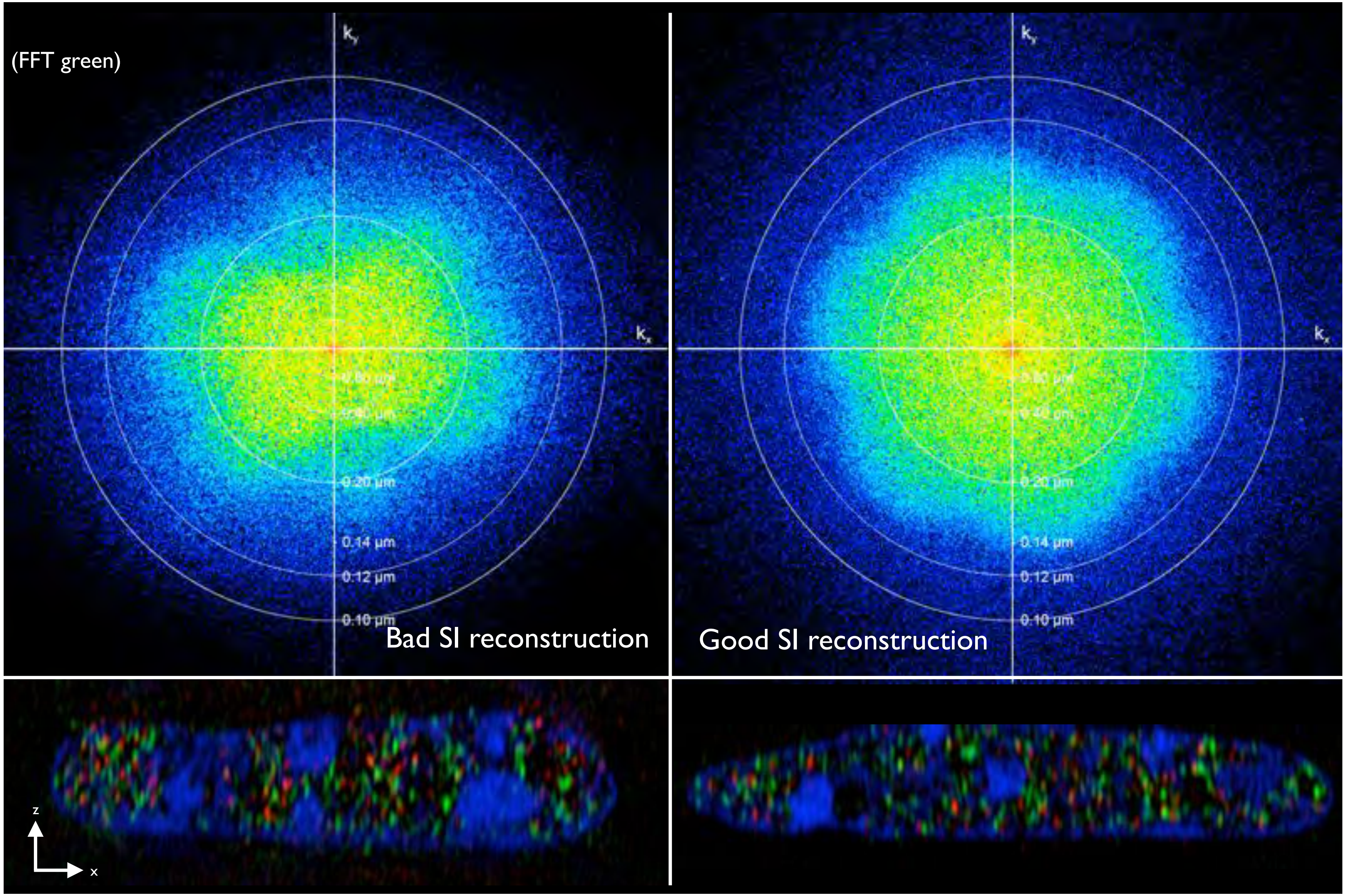
Balance between contrast and bleaching

SIM algorithm assumes a perfectly matched PSF!
When it detects out of focus light from mismatched PSF,
it assumes this is real signal & reconstructs it

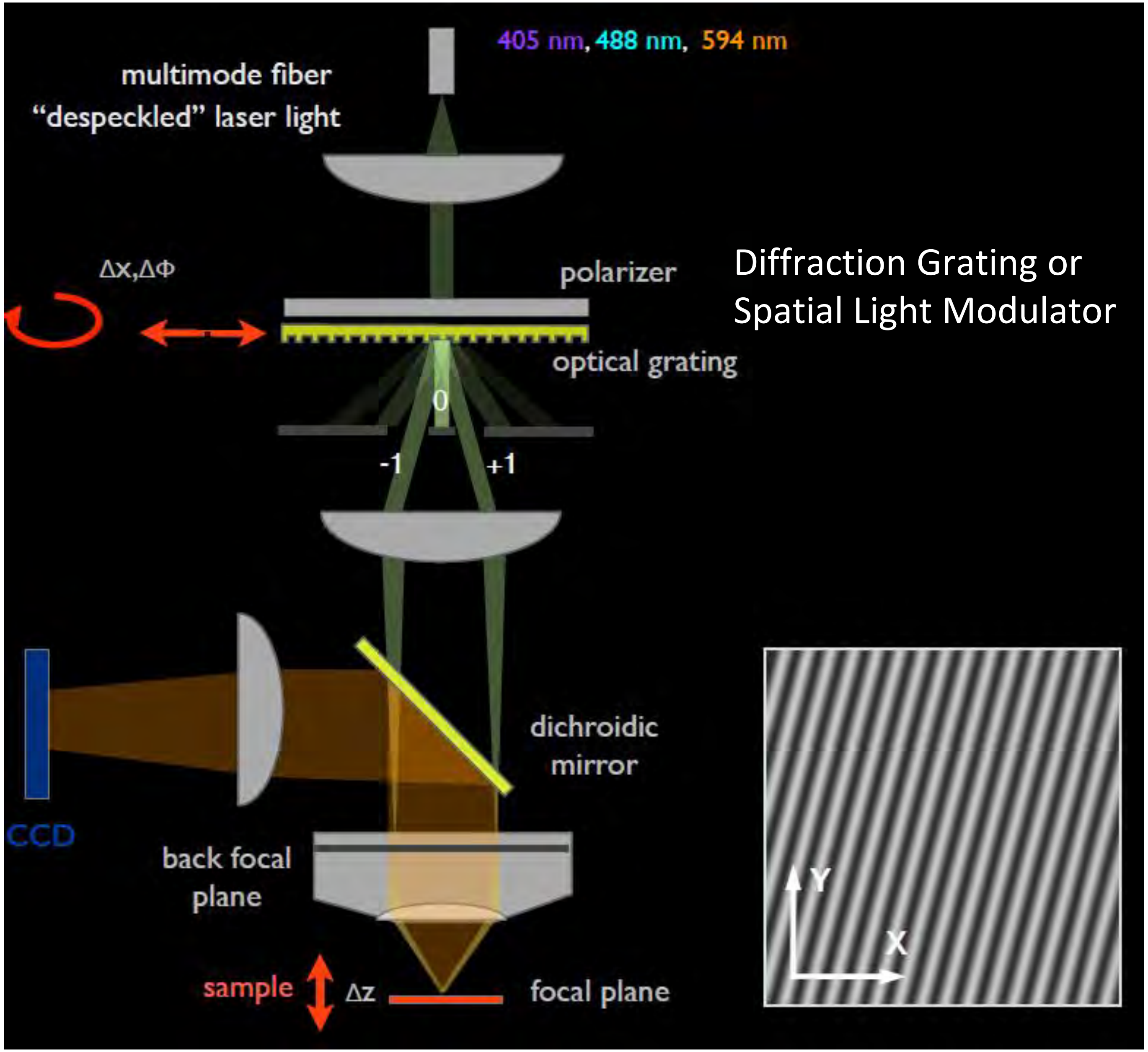
SIM – Reconstruction Artifacts → Quality Control



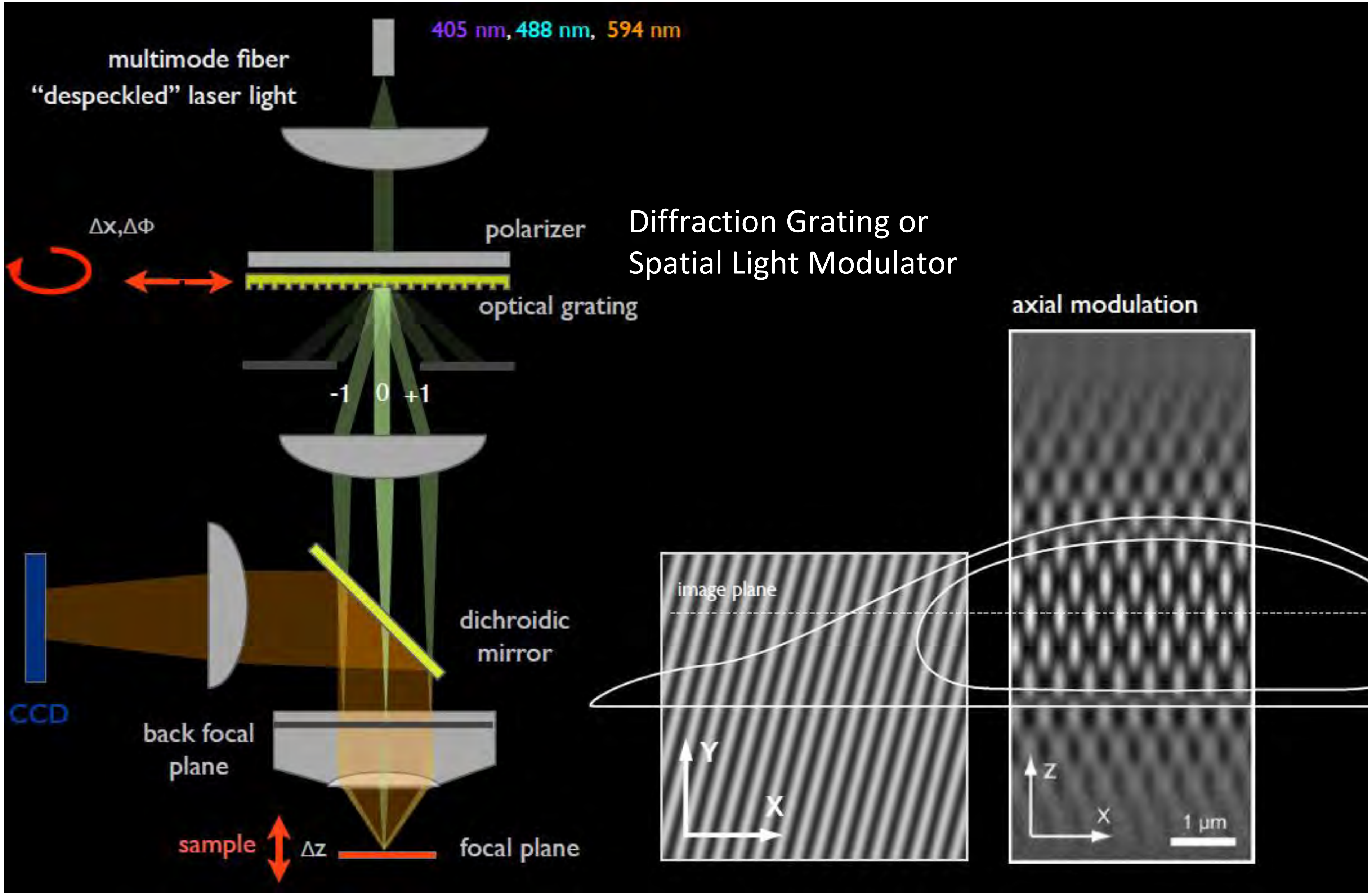
SIM – Reconstruction Artifacts → Quality Control



2D SIM – Optical system

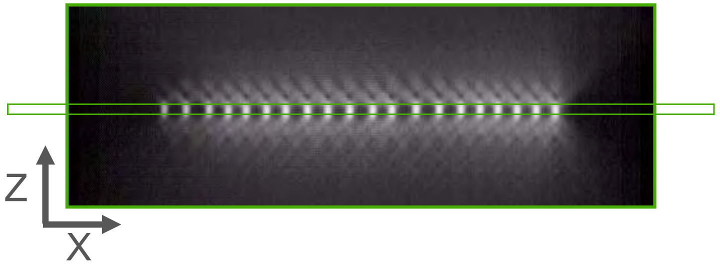


3D SIM – Optical system

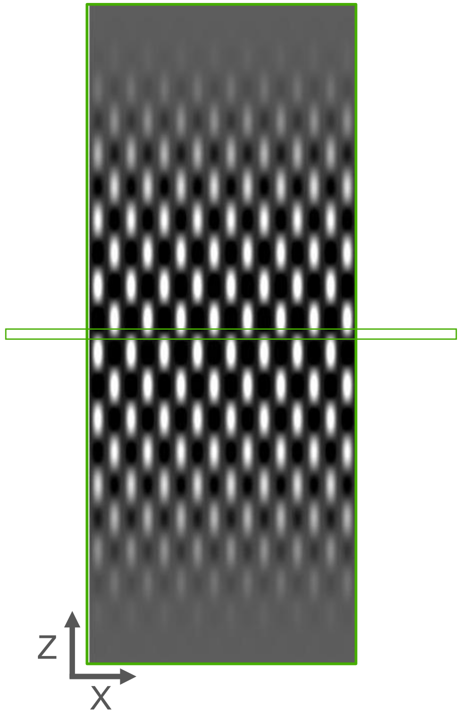


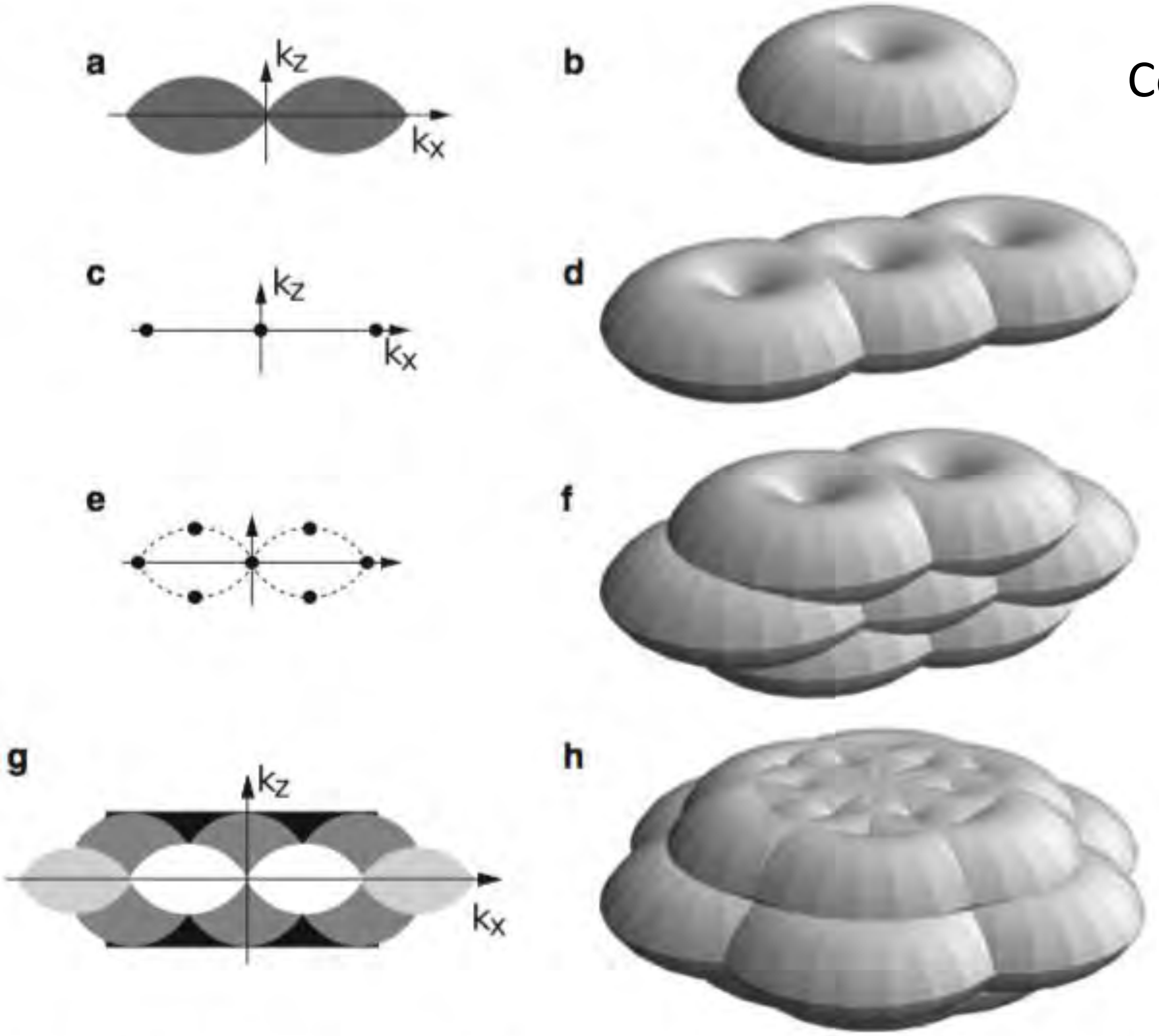
3D SIM – Concept

2D SIM



3D SIM





Conventional OTF

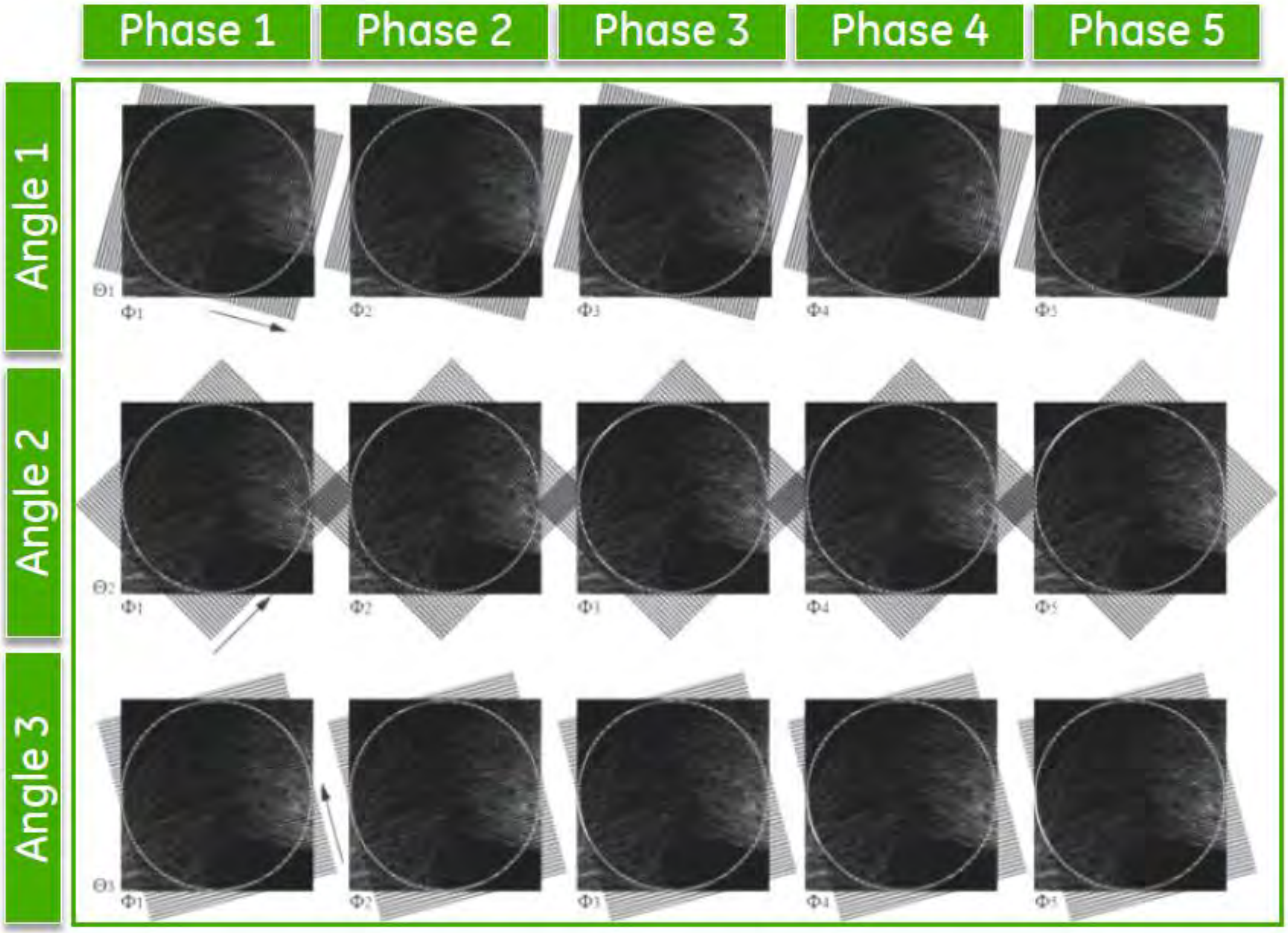
2 beams

3 beams

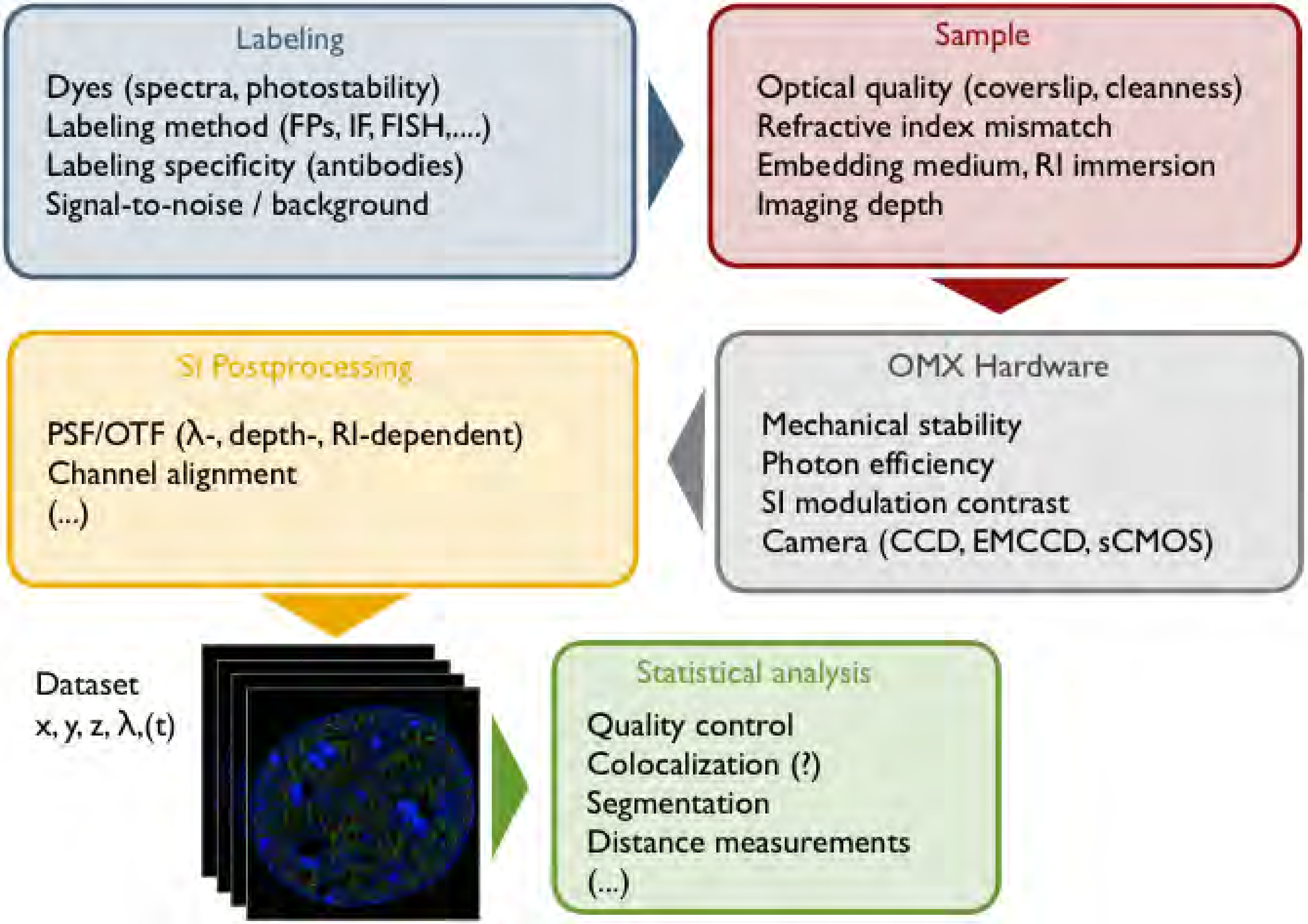
Fig. 3 Principle of 3D SIM. Observable regions for the conventional microscope (**a** and **b**), for structured illumination microscopy using two illumination beams (**d**), and three illumination beams in one (**f**) or three (**g**, **h**) sequential orientations. (**a**) and (**g**) are the k_x–k_z cross section of the 3D observable regions shown in (**b**) and (**h**), respectively. The spatial-frequency components of the structured illumination intensity for the two-beam (**c**) and three-beam (**e**) case. The *dotted outline* in panel (**e**) indicates the set of the highest spatial frequencies that are possible to generate by illumination through the objective lens; compare with the observable region in panel (**a**)

Raw 3D SIM images

Shift pattern through 5 phases at 3 angles (total 15)



SIM – How to get the best image?



SIM – Pros & Cons

+ Multicolor, **standard dyes**

+ 3D with **2x resolution** in XY and Z

+ Massive **contrast enhancement** / High dynamic range

+ **Optical sectioning** over large volumes

+ **Sensitive** (EMCCD and sCMOS) and **fast** (SLM)

+ Fast imaging over a **large field of view**

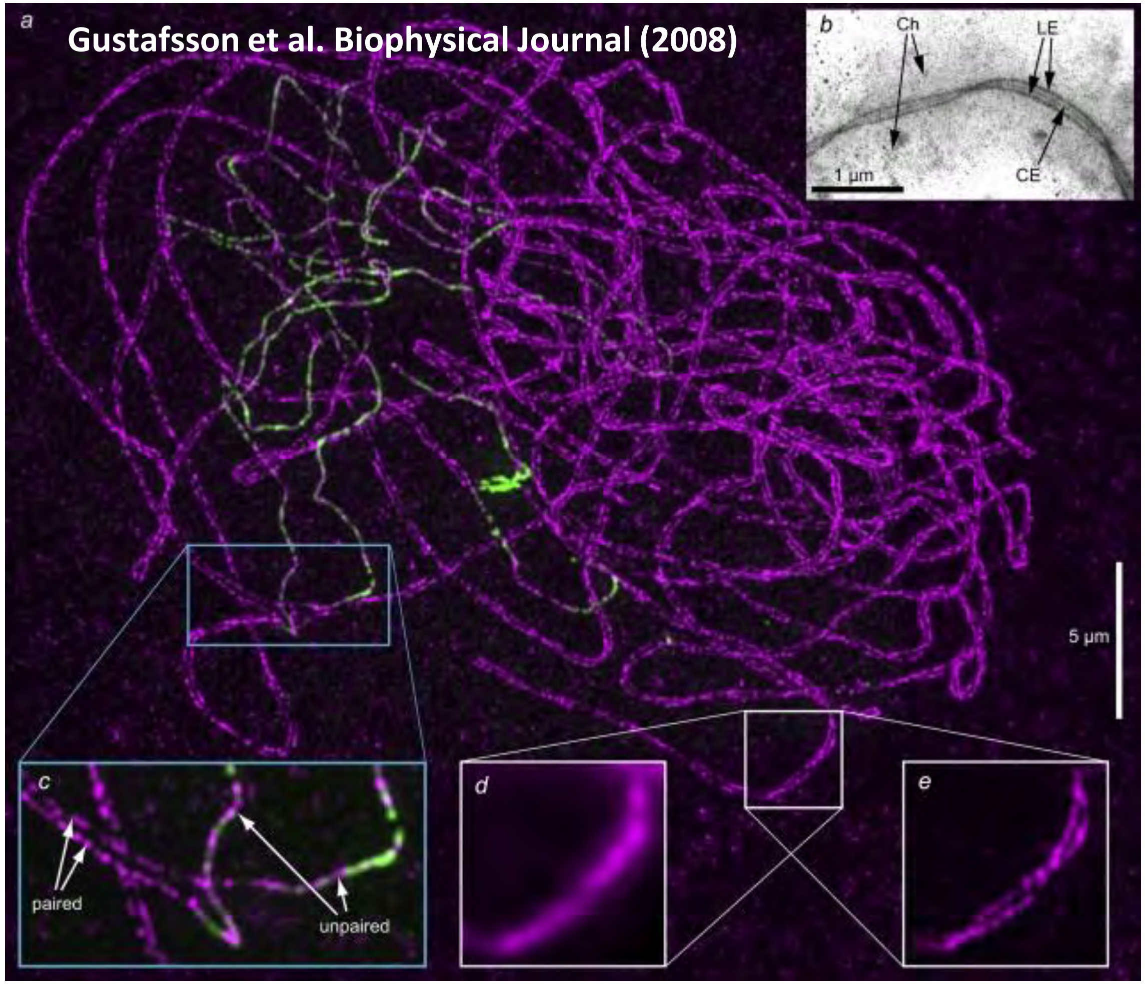
— **Moderate lateral resolution improvement**

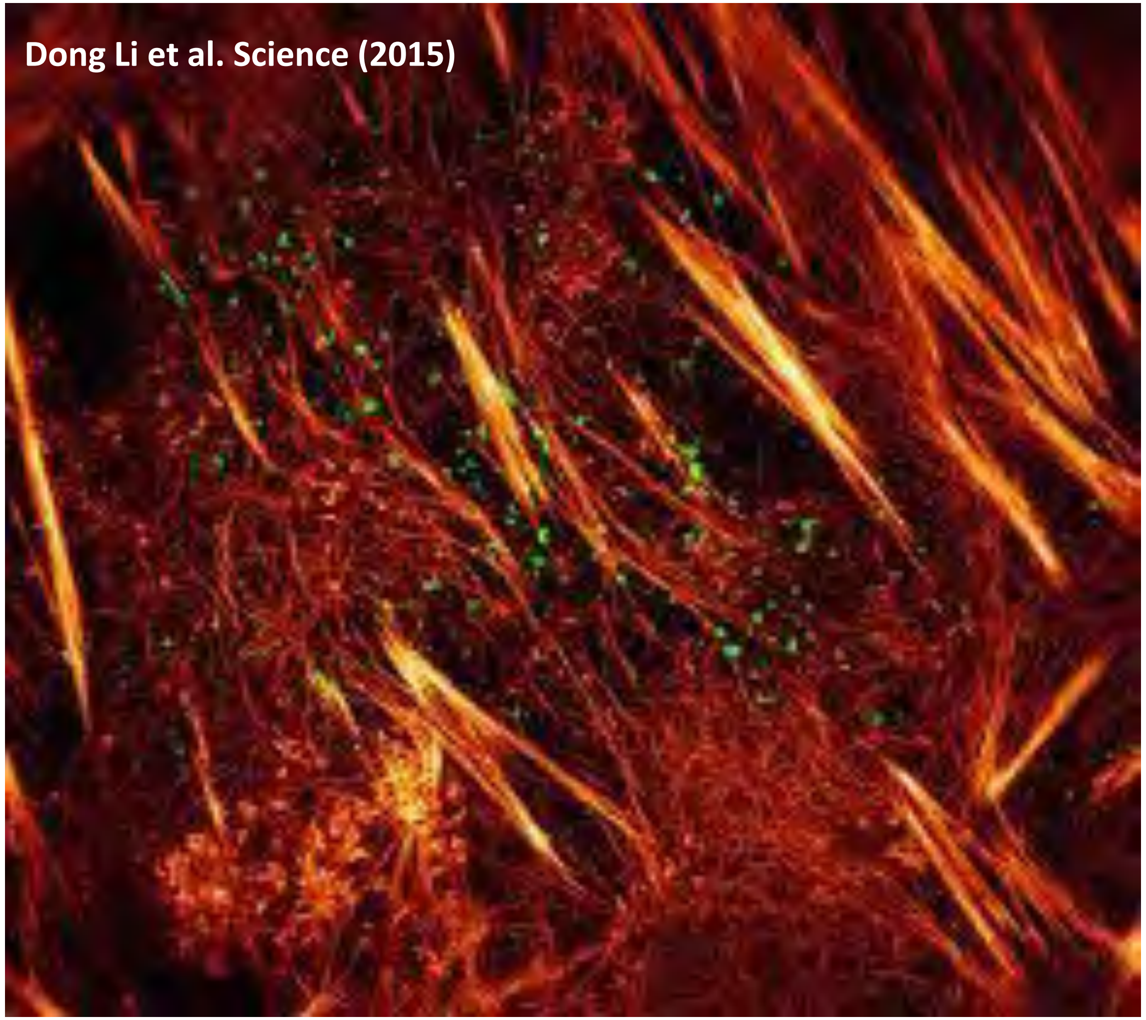
— Mathematical reconstruction which may lead to **artifacts**

— **High requirements** on sample quality and system calibration

By projecting a sinusoidal fringe pattern onto the specimen, **SIM images the fringe efficiently only on the parts of the specimen that are in focus**. The out-of-focus background can be removed

SIM – Experimental results





References

Gustafsson, M. G. (2000). Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy. *Journal of Microscopy*, *198*(Pt 2), 82–87. http://doi.org/10.1046/j.1365-2818.2000.00710.x

Gustafsson, M. G. L., Shao, L., Carlton, P. M., Wang, C. J. R., Golubovskaya, I. N., Cande, W. Z., et al. (2008). **Three-dimensional** resolution doubling in wide-field fluorescence microscopy by structured illumination. *Biophysical Journal*, *94*(12), 4957–4970. http://doi.org/10.1529/biophysj.107.120345

Schermelleh, L., Carlton, P. M., Haase, S., Shao, L., Winoto, L., Kner, P., et al. (2008). Subdiffraction Multicolor Imaging of the Nuclear Periphery with 3D Structured Illumination Microscopy. *Science*, *320*(5881), 1332–1336. http://doi.org/10.1126/science.1156947

Rego, E. H., & Shao, L. (2014). Practical Structured Illumination Microscopy. In *Cell Imaging Techniques* (Vol. 1251, pp. 175–192). New York, NY: Springer New York. http://doi.org/10.1007/978-1-4939-2080-8_10

Li, D., Shao, L., Chen, B.-C., Zhang, X., Zhang, M., Moses, B., et al. (2015). Extended-resolution structured illumination imaging of endocytic and cytoskeletal dynamics. *Science*, *349*(6251), aab3500–aab3500. http://doi.org/10.1126/science.aab3500

Lal, A., Shan, C., & Xi, P. (2016, February 19). Structured illumination microscopy image **reconstruction algorithm**. *arXiv.org*. http://doi.org/10.1109/JSTQE.2016.2521542

Demmerle, J., Innocent, C., North, A. J., Ball, G., Müller, M., Miron, E., et al. (2017). Strategic and practical guidelines for successful structured illumination microscopy. *Nature Protocols*, *12*(5), 988–1010. http://doi.org/10.1038/nprot.2017.019

הטכניון
מכון טכנולוגי לישראל

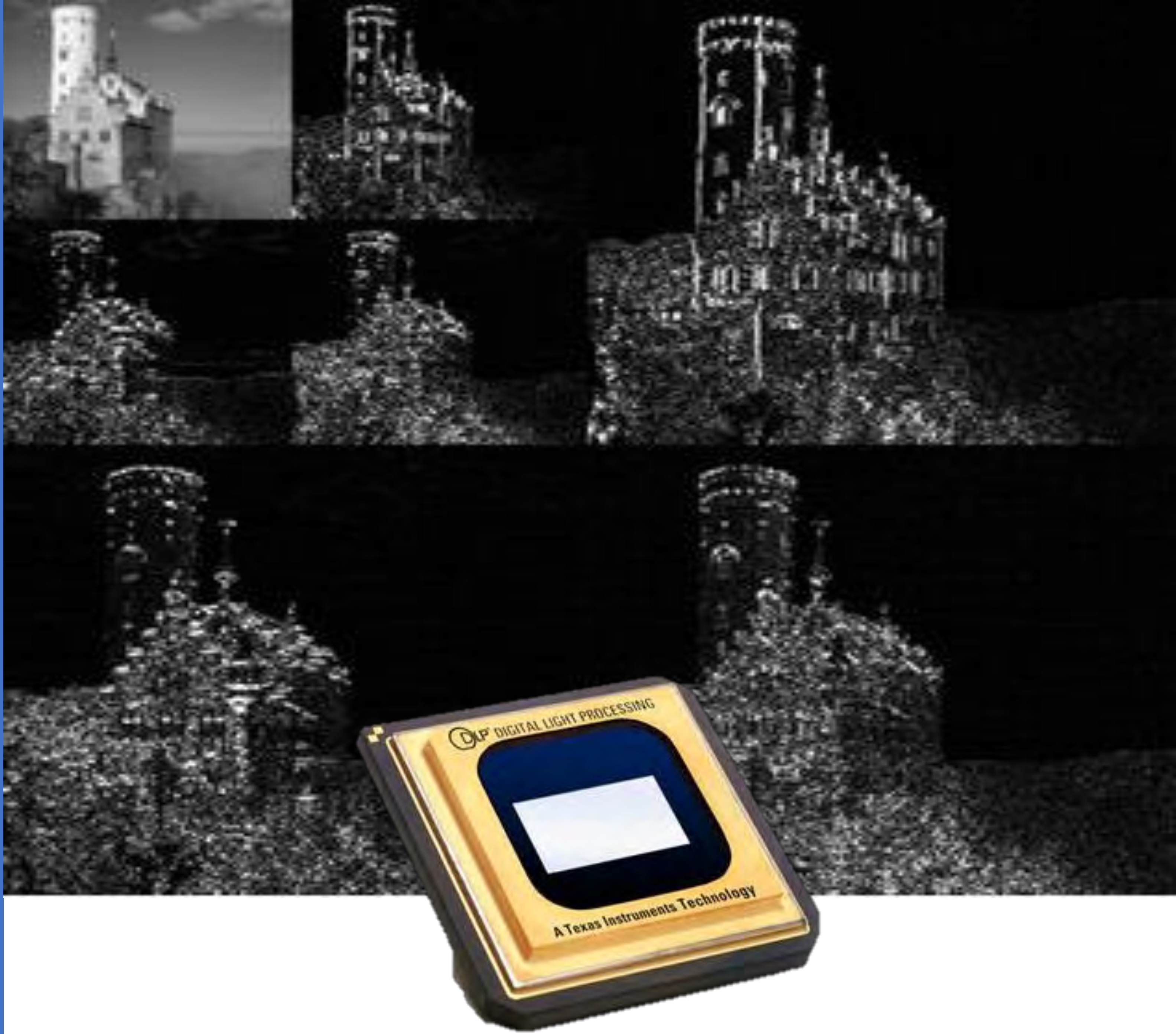
Department of Biomedical Engineering, Technion

Computational optical imaging 336547

Tutorial 5+6 – Compressed Sensing

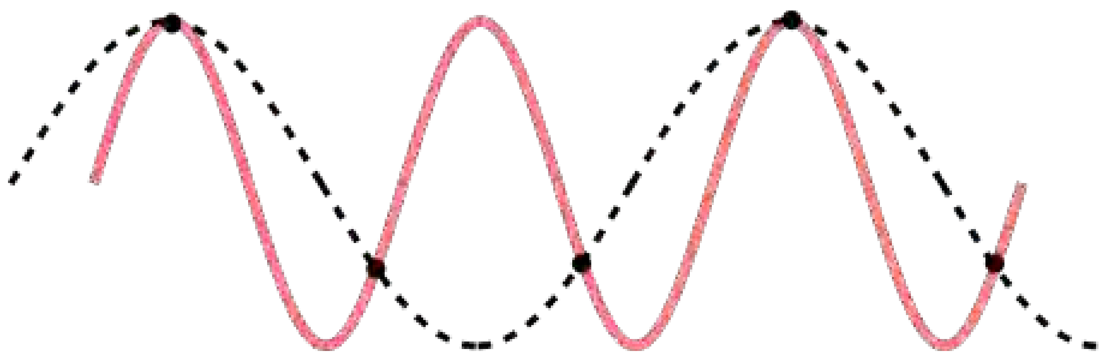
Elias Nehme & Yoav Shechtman

24 November 2020



Nyquist Sampling Theorem

Traditional sampling method:



1D case:

If a function x(t) contains no frequencies higher than **B hertz**, it is completely determined by giving its ordinates at a series of points spaced **1/(2B) seconds** apart:

$$f_{sampling} > 2f_{max}$$

2D case:

Pixel size is small for acquiring high frequencies, hence for **large field of view the number of pixels is large**

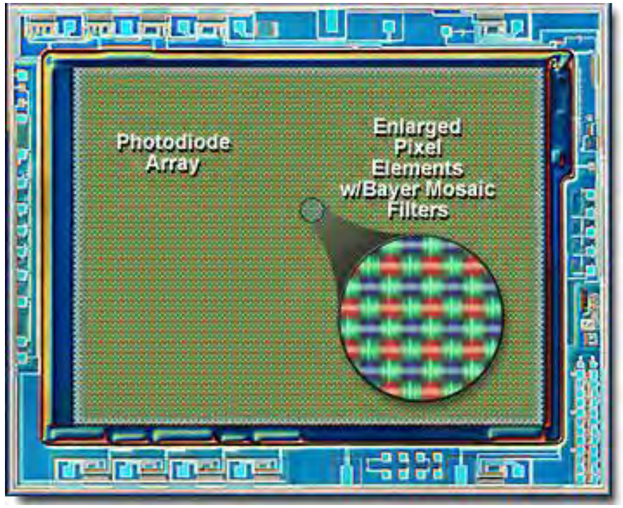
→ Digital cameras in the **megapixel range**

Using **silicon** which converts photons to electrons in the **visual wavelengths**

Image Compression

In a digital camera, the samples are obtained by a 2-D array of N pixel sensors on a **CCD or CMOS imaging chip**

We represent these samples using the vector x with elements x[n], n = 1, 2, . . . ,N → $N \sim 10^6$



Raw image data x is often **compressed**:

$$x = \sum_{i=1}^{N} \alpha_i \psi_i$$

$\{\psi_i\}_{i=1}^{N}$ NX1 orthonormal basis vectors

α_i N coefficients

$$\psi = [\psi_1 | \psi_2 | \dots | \psi_N] \qquad \alpha = \begin{bmatrix} \alpha_1 \\ \cdot \\ \cdot \\ \cdot \\ \alpha_N \end{bmatrix}$$

Matrix form:

$$\boxed{x = \psi \alpha}$$

NX1 NXN NX1

The aim is to **find a basis ψ where the coefficient vector α is sparse**
→ where only **K<<N coefficients are nonzero**

Basis vectors for **natural images**:
- Discrete cosine transform (**DCT**)
- **Wavelet**

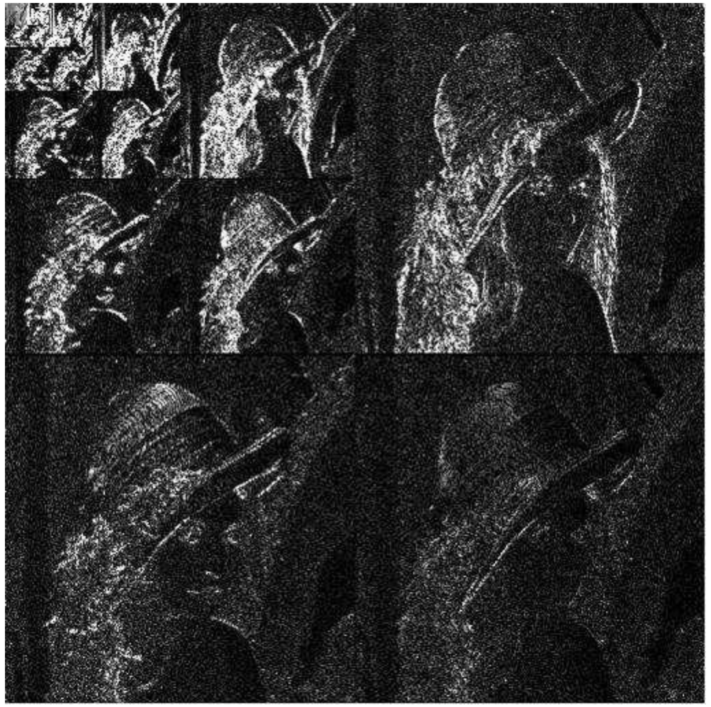
→ On which the **JPEG and JPEG-2000** compression standards are based

Only the <u>values and locations</u> of the K significant coefficients are encoded

Image compression – Sparse representation

Decompose the signal into a **sparse linear expansion**

Low freq.



Wavelet basis

High freq.

$$x = \sum_i \alpha_i \psi_i$$ such that $$\|\alpha\|_0^0 = \mathrm{K}$$

sparse

lp norm = $(\sum |x_i|^p)^{\frac{1}{p}}$

Transform the physical signal into a **sparse dataset** and register a **fraction of the strongest coefficients**

Sparsity - Reminder

How are unique **sparse representations determined from signals**?

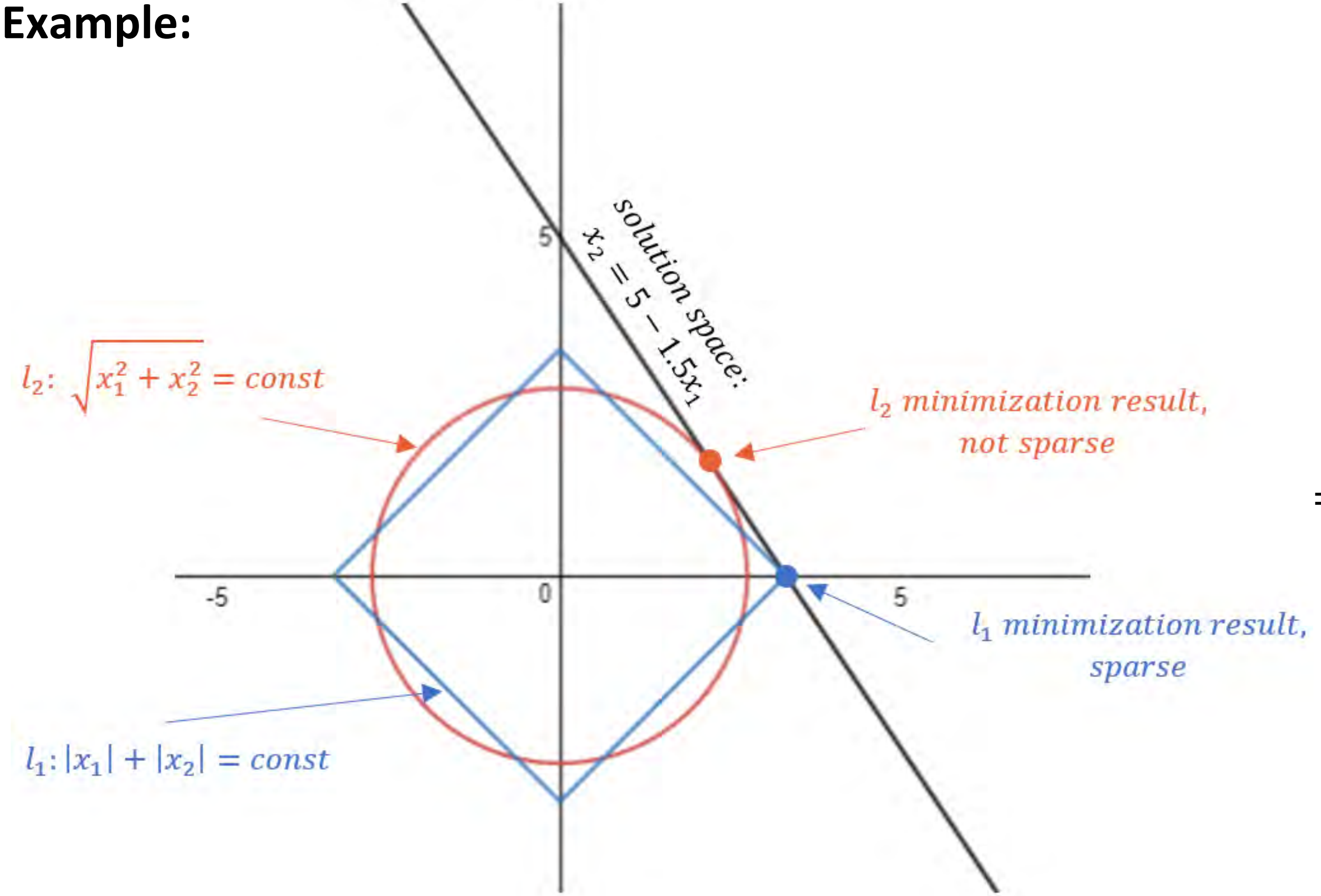
- **Iterative Greedy Algorithm (MP)**: l_0–norm minimization, non-convex
- **Relaxation (BP)**: l_1–norm minimization, convex → **optimization problem promoting sparsity**

l_0 norm $\|x\|_0^0$ = #non-zero elements

l_1 norm $\|x\|_1 = \sum |x_i|$

$l_2$norm $\|x\|_2 = (\sum |x_i|^2)^{\frac{1}{2}}$

Example:



$$\begin{cases} A = (1.5\ 1) \\ y = 5 \end{cases}$$

$$\Rightarrow \underbrace{(1.5\ 1)}_{A} \underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{x} = \underbrace{5}_{y} \xRightarrow{solution\ space} x_2 = 5 - 1.5x_1$$

$$||x||_0 = \begin{cases} 0, & x_1 = x_2 = 0 \\ 1, & x_1 = 0\ or\ x_2 = 0 \\ 2, & else \end{cases}$$

Sparsity - Reminder

l_1 norm $\|x\|_1 = \sum |x_i|$

Relaxation (BP): l_1–norm minimization, convex → optimization problem promoting sparsity

$$x = \sum_i \alpha_i \psi_i \quad \text{such that} \quad \underbrace{\|\alpha\|_0^0 = \mathrm{K}}_{\textit{sparse}}$$

→

$$x = \sum_i \alpha_i \psi_i \quad \text{such that} \quad \underbrace{\|\alpha\|_1 = \mathrm{K}}_{\textit{sparse}}$$

→ **Sparsity argument:** minimizes the "number of non-zero coefficients"

$$\overbrace{\arg\min_{\alpha} \|\alpha\|_1} \quad \text{such that} \quad \underbrace{x = \psi\alpha}$$

Signal constraint: Ensures that the signal x can be recovered from the sparse coefficients α

α **Sparse coefficients** of x in ψ

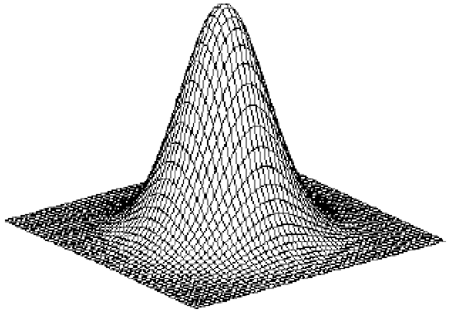
ψ **Sparsifying basis**

x **Signal**

→ l_1-solvers recover **the best sufficiently sparse approximation** of a signal by penalizing the l_1-norm of the coefficients

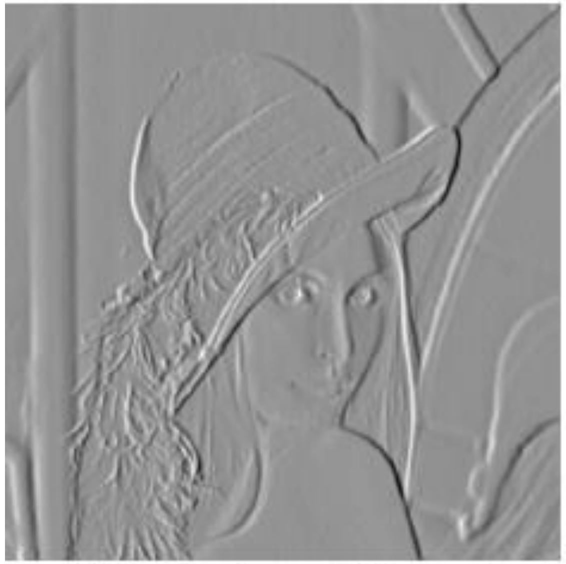
Another example of Sparsity – Total Variation Minimization

Gradient Magnitude



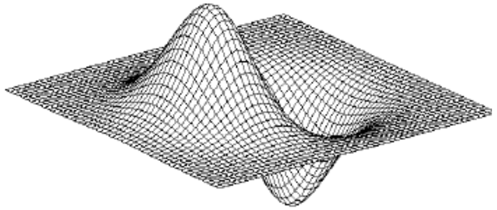
Gaussian

$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



X-Derivative of Gaussian

Y-Derivative of Gaussian



derivative of Gaussian

$$\frac{\partial}{\partial x} h_\sigma(u,v)$$

Another example of Sparsity – Total Variation Minimization

If $u \in X = \mathbb{R}^{N\times N}$, the linear gradient operator ∇u is a vector in $Y = X \times X$ given by:

$$(\nabla u)_{i,j} = \left((\nabla u)_{i,j}{}^{x}, (\nabla u)_{i,j}{}^{y}\right)$$

with

$$(\nabla u)_{i,j}{}^{x} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases}$$

$$(\nabla u)_{i,j}{}^{y} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases}$$

Recall the l_1–norm

$$\|x\|_1 = \sum_{i=1}^{N} |x_i|$$

The total variation of u is defined by $J(u) = \sum_{1\le i,j\le N} \left|(\nabla u)_{i,j}\right|$

Total Variation Minimization Problem

$$\operatorname{argmin}_x \|x\|_{TV} \quad \text{such that} \quad y = Ax$$

$$\|x\|_{TV} = J(x) = \sum_{1\le i,j\le N} \left|(\nabla x)_{i,j}\right| = \sum \sqrt{\left|D_1 x(t_1,t_2)\right|^2 + \left|D_2 x(t_1,t_2)\right|^2}$$

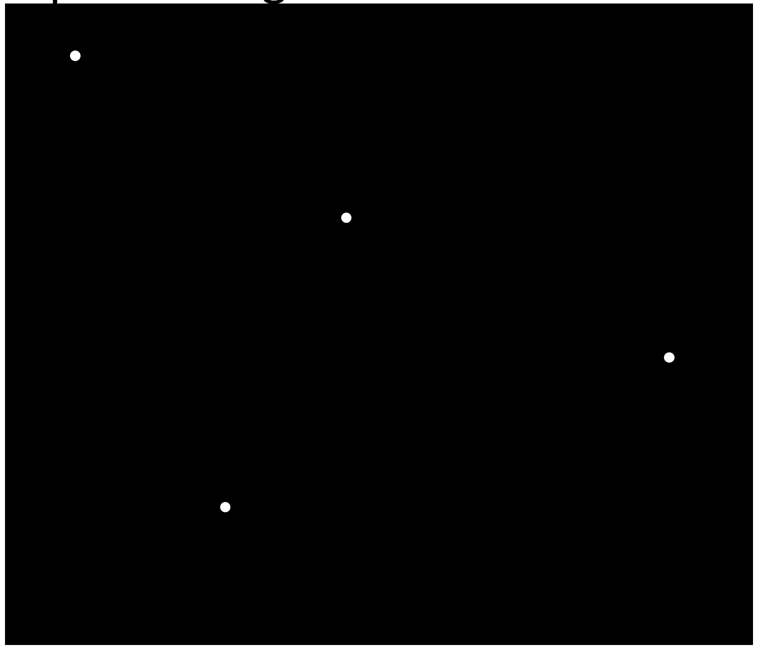
$$\|x\|_{TV} = J(x) = \sum_{1\le i,j\le N} \left|(\nabla x)_{i,j}\right| = \|\nabla x\|_1$$

→ It is possible to **use the sparsity assumption on the gradient of the signal** and to perform l_1 minimization

Image compression – Sparse representation

Sample-then-compress

1000x1000 pixels, 10^6 sample points



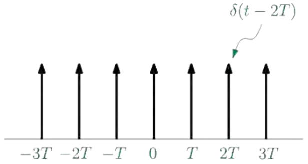
1000x1000 pixels, 10^6 sample points

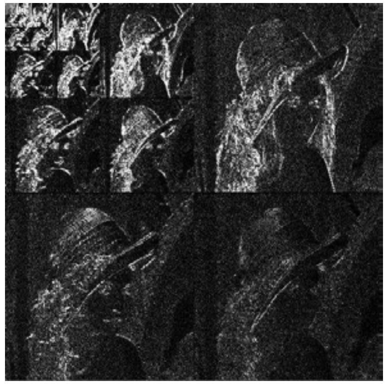
- Huge information is acquired by sampling, although **most of it is a waste**
- Does the image of **4 points over a black background require to sample 10^6 points**?

An alternative → Compressive sampling

Compressed Sensing (CS)

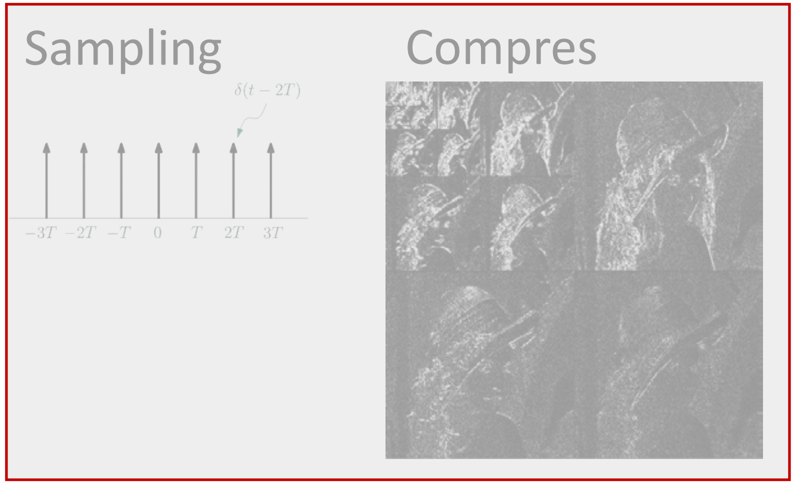
Physical Signal → Sampling → Compres → Decode





Compressive Sampling

Physical Signal →



→ Decode

CS bypasses the sampling process → directly acquires a condensed representation

CS – Principles

Acquire directly condensed representation by using *M* < *N* linear measurements y between *x* and a collection of M test functions: $\{\phi_m\}_{m=1}^{M}$

To get: $y[m] = \langle x, \phi_m \rangle$

Rather than measuring pixel samples of the scene
→ measure inner products between the scene and a set of test functions

$$\phi = \begin{bmatrix} \phi_1 \\ - \\ . \\ . \\ - \\ \phi_M \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ . \\ . \\ . \\ y_M \end{bmatrix}$$

Matrix form:

$$y = \phi x$$

MX1 MXN NX1

$$x = \psi\alpha$$

Random matrix Sparsitfying basis

$$\boxed{y = \phi\psi\alpha}$$

MXN NXN NX1

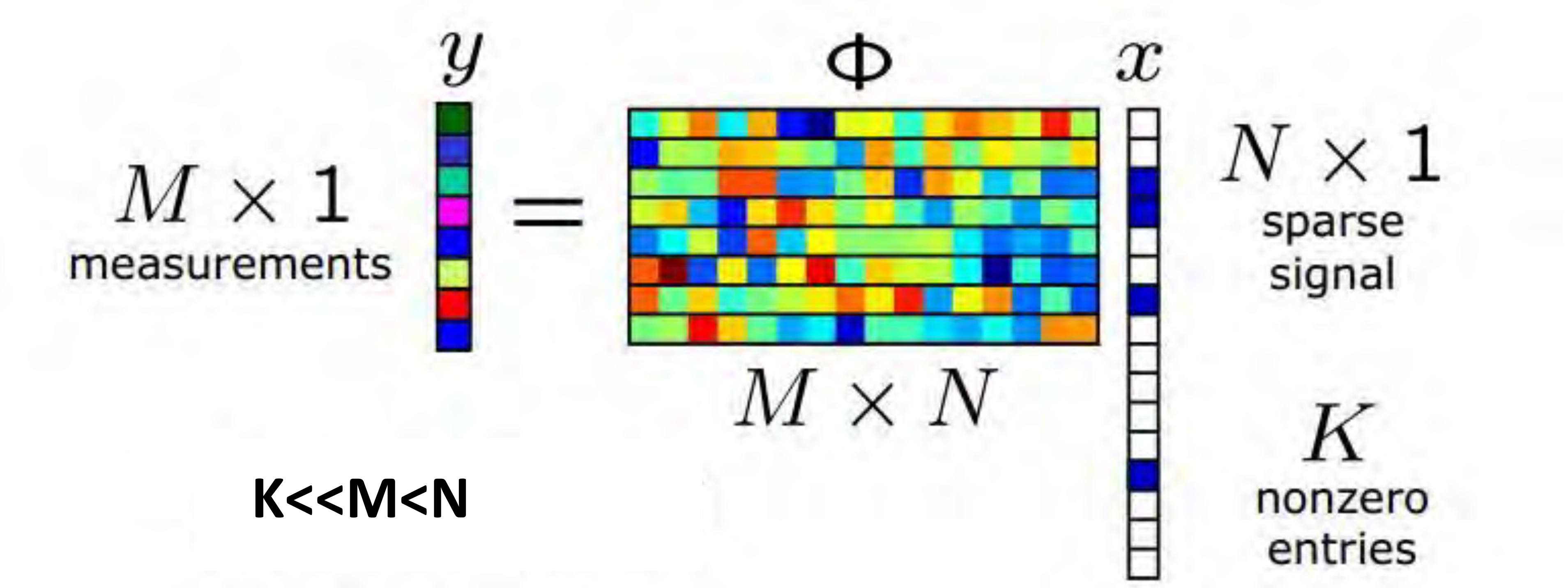
K<<M<N

Each measurement is a random sum of pixel values taken across the entire image

Since M < N there are **infinitely many x such that** $\phi x = y$
→ The magic of CS is that ϕ can be designed such that sparse/compressible x can be recovered from the measurements y

CS – Principles

- **Assume the physical signal x is sparse**
- Records M different linear combinations of all values of x



K<<M<N

Recover x from y

- Nyquist Theorem: $M = N$ and $\Phi = I$ is trivial
- Compressed Sensing Theory: $M < N$ if x is sparse (K nonzero entries). How?

$$\arg\min_x \|x\|_1 \qquad \text{such that} \qquad \Phi x = y$$

CS Theorem

- Candes, Romberg and Tao showed that one could **almost always recover the *K*-sparse signal x exactly by solving the convex problem**:

$$\arg\min_x \|x\|_1 \quad \text{such that} \quad \Phi x = y$$

$$\boxed{\|x\|_0^0 \leq \frac{\sigma_{spark}}{2}} \quad \boxed{\|x\|_0^0 \leq \frac{1}{2}\left(1 + \frac{1}{\mu}\right)}$$

Under the condition that ***Φ* obeys the "restricted isometry hypothesis".** ***Spark, Mutual coherence***

- Alternatively, If the *K*-sparse signal is $\alpha = \Psi^T x$:

$$\arg\min_\alpha \|\alpha\|_1 \quad \text{such that} \quad \Phi\Psi\alpha = y$$

When the measurement basis Φ cannot sparsely represent the elements of the sparsifying basis Ψ (x is sparse in a known orthorgonal system Ψ) – a condition known as ***incoherence*** of the two bases – and the number of measurements ***M* is large enough**, then it is possible to recover the signal x from the measurements y

$$M \geq \text{Const} \cdot \mu^2 \cdot K \cdot \log(N)$$

Coherence
$$\mu = \max_{i,j} \left|\langle \Phi_i, \psi_j \rangle\right|$$

Candès, E.J. *IEEE Trans. Inform. Theory*, 2004

Candes,E.J.,Romberg,J.,Tao,T. *IEEE Trans. Inform. Theory* **52** (2006), 489–509.

Incoherent Bases

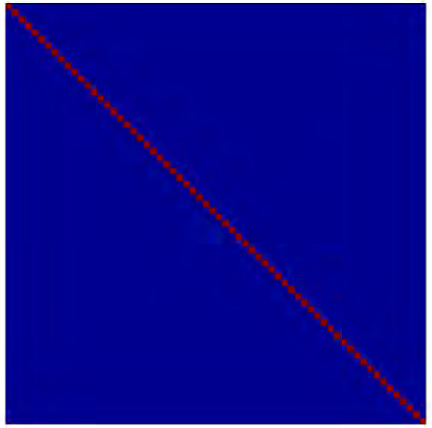
$$\boxed{y = \phi\psi\alpha}$$

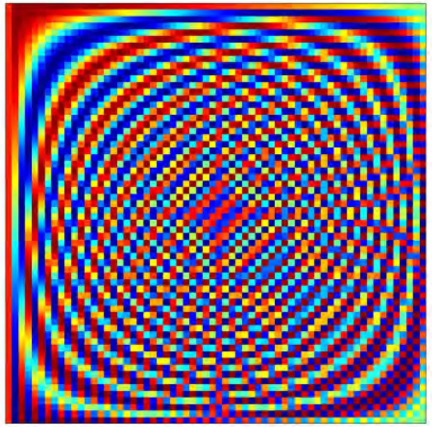
MXN NXN NX1

K<<M<N

- Spikes and sines (Fourier)

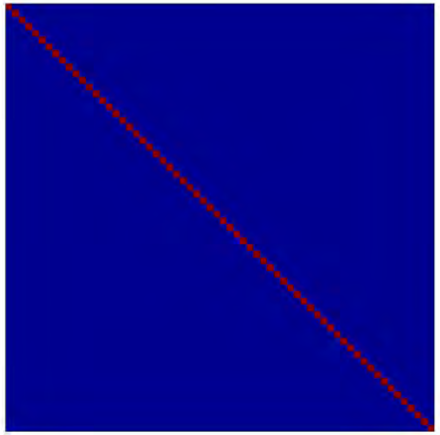
$\Psi = I$ $\Phi = \mathrm{idct}(I)$

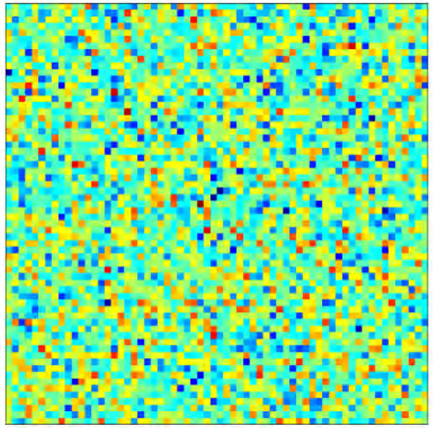




- Spikes and "random basis"

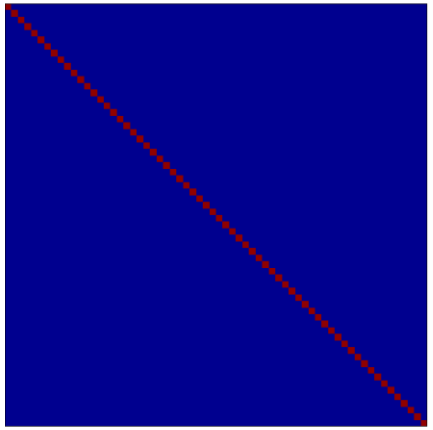
$\Psi = I$ $\Phi = \mathrm{randn}$

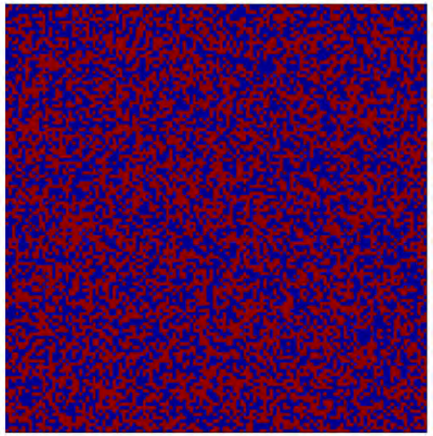




- Spikes and "random sequences"

$\Psi = I$ Φ





Incoherent Bases – Random matrices

$$\boxed{y = \phi\psi\alpha}$$

MXN NXN NX1

K<<M<N

- Spikes and sines (Fourier)

$\Psi = I$ $\Phi = \mathrm{idct}(I)$

Fourier measurements. Φ is a *partial Fourier matrix* obtained by selecting M rows uniformly at random and renormalizing the columns (unit-normed). Then Candès and Tao showed that Φ obeys the restricted isometry property with overwhelming probability if:
K ≤ C · M/(log N)^6

- Spikes and "random basis"

$\Psi = I$ $\Phi = \mathrm{randn}$

Gaussian measurements. The entries of matrix Φ are independently sampled from *N(0,1/M)*
Then if:
K ≤ C · M/ log(N/M)
Φ obeys the restricted isometry property

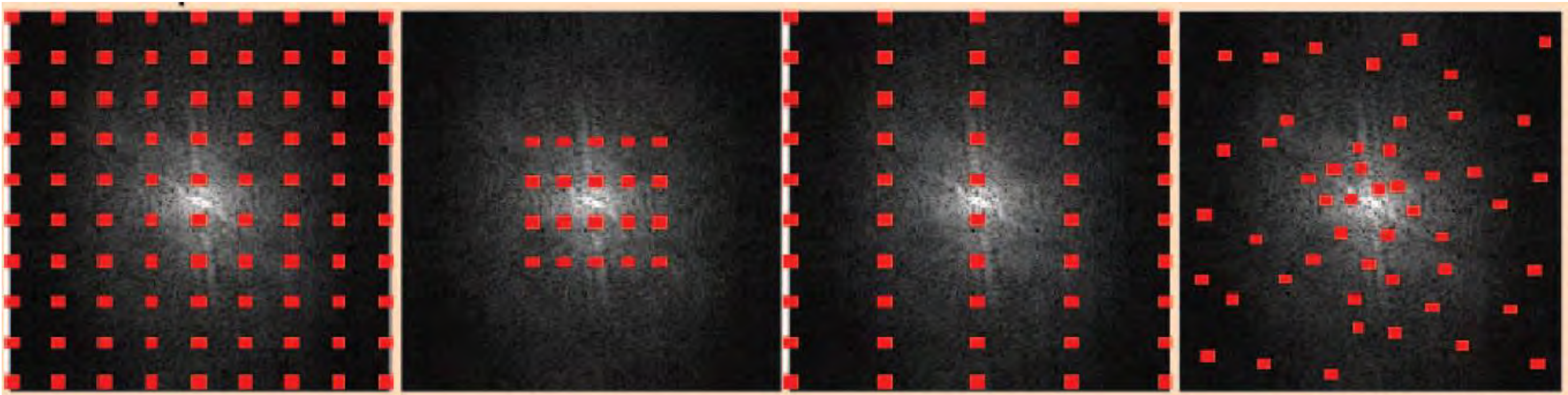
- Spikes and "random sequences"

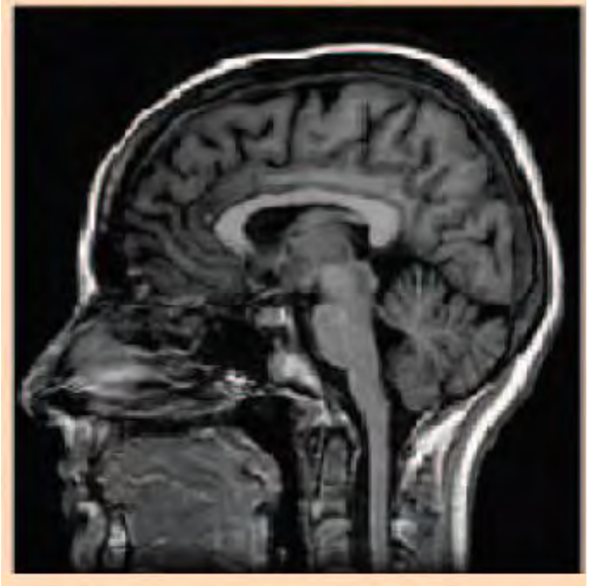
$\Psi = I$ Φ

Binary measurements. The entries of matrix Φ are independently sampled from the symmetric *Bernoulli distribution* P (Φ_{ki} =±1/ M) = 1/2. Then if:
K ≤ C ·M/ log(N/M)
Φ obeys the restricted isometry property

*CS **places most of its computational complexity in the recovery system** → Often has **more substantial computational resources than the measurement system***

Incoherent Sampling – Partial Fourier





Lustig, M. *IEEE Sig. Proc. Magazine*, 2008

CS Application – Single Pixel Camera

$\boxed{y = \phi\psi\alpha}$

MXN NXN NX1

K<<M<N

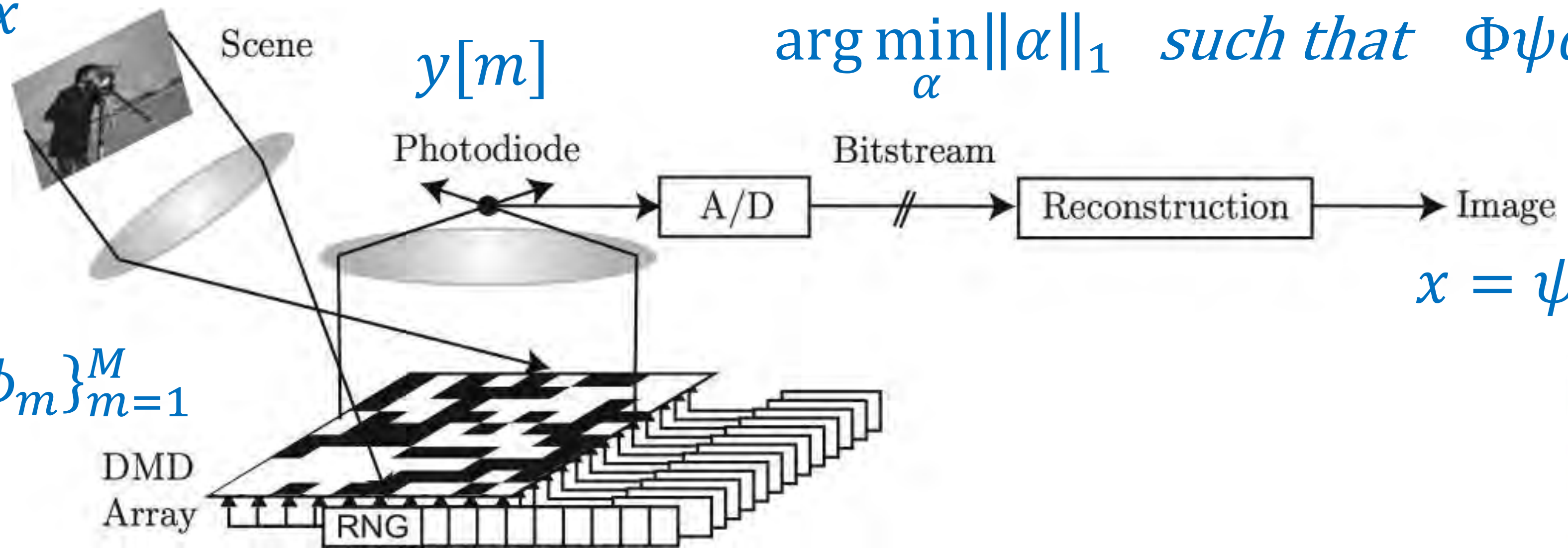
x

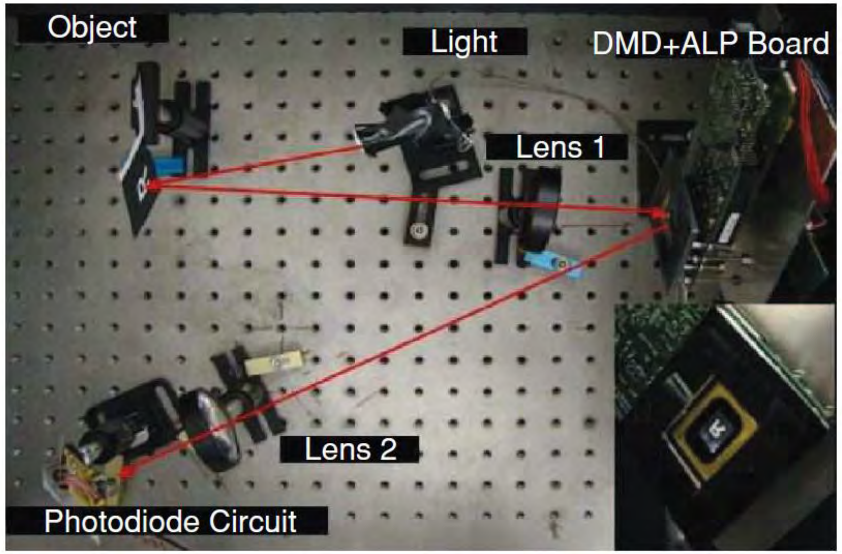
$y[m]$

$\arg\min_{\alpha} \|\alpha\|_1 \quad \textit{such that} \quad \Phi\psi\alpha = y$

$x = \psi\alpha$

$\{\phi_m\}_{m=1}^{M}$





Computes random linear measurements of the scene under view

The camera design **reduces the required size, complexity, and cost of the photon detector array down to a single unit →** Enables the use of **exotic detectors** that would be impossible in a conventional digital camera

Photomultiplier tube or an avalanche photodiode for low-light (photon-limited) imaging

PMT

APD

Digital micromirror device (DMD)

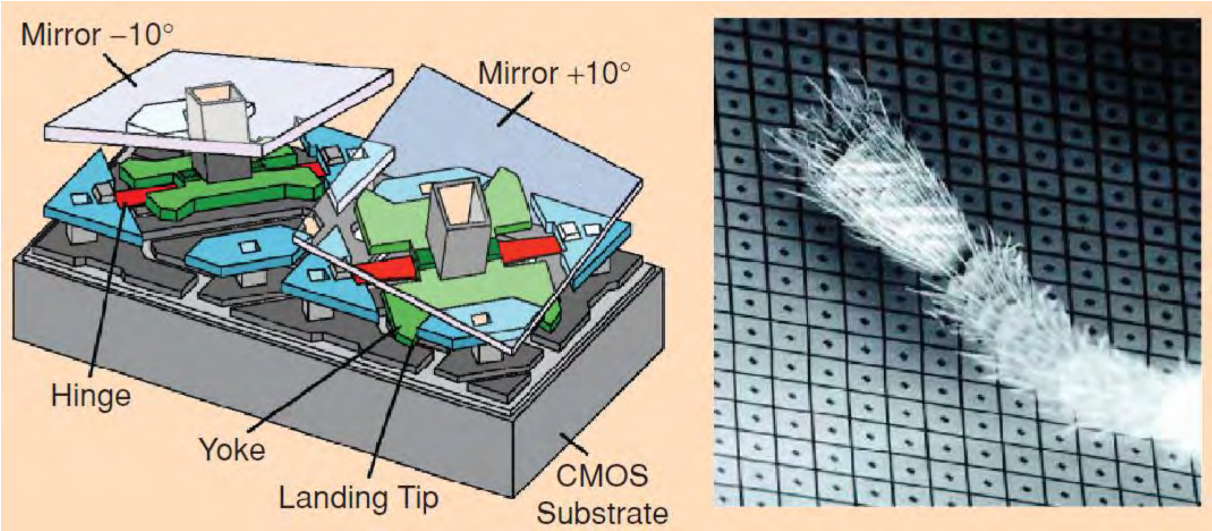
Spatial light modulator (**SLM**) modulates the intensity (or phase) of a light beam according to a control signal

DMD – Reflective SLM that selectively redirects parts of the light beam

The DMD consists of an **array of bacterium-sized, electrostatically actuated micromirrors**

Each mirror rotates about a hinge and can be **positioned in one of two states** (+10° and −10° from horizontal) according to which bit is loaded

Light falling on the DMD can be **reflected in two directions depending on the orientation of the mirrors** (to get “on” and “off” states)



Duarte et al. *IEEE signal processing magazine* (2008).

CS Application – Single Pixel Camera

$$\boxed{y = \phi\psi\alpha}$$

MXN NXN NX1

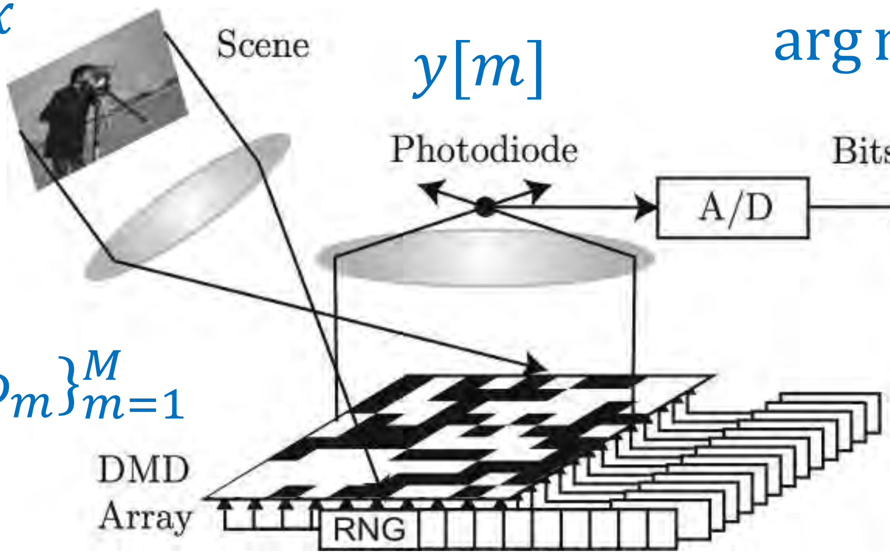
K<<M<N

$$\arg\min_{\alpha} \|\alpha\|_1 \quad \textit{such that} \quad \Phi\psi\alpha = y$$

x

$y[m]$

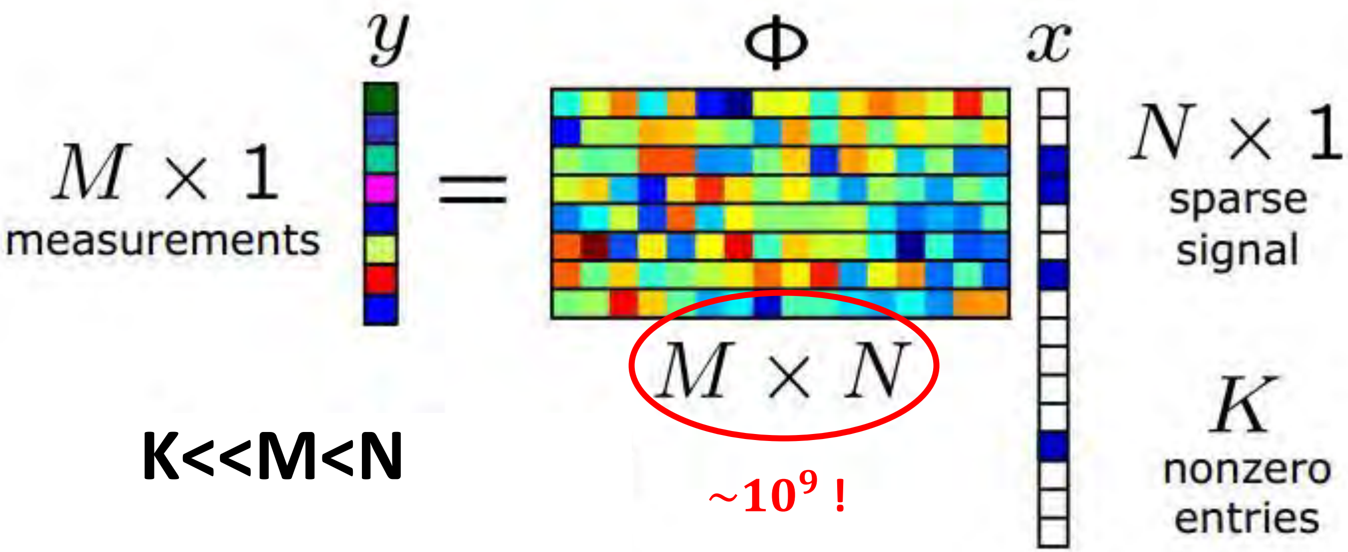
$\{\phi_m\}_{m=1}^{M}$



$x = \psi\alpha$

256 × 256
conventional image

Single-pixel camera
M = 1,300

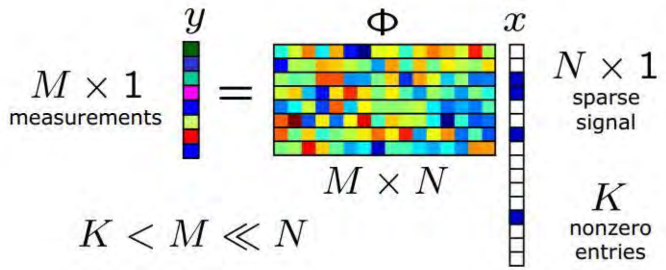


K<<M<N

$\sim \mathbf{10^9}$ **!**

The implementation of matrix Φ on the DMD requires **a large amount of RAM memory**

Single Pixel Camera – MATLAB simulation



Total Variation Minimization

$\arg\min_x \|x\|_{TV}$ such that $\Phi x = y$

We need a **fast and reversible** transformation which **does not require to construct a matrix Φ**
Random Gaussian ensemble do not exhibit such a property although "**randomness**" is highly desirable for achieving maximum **incoherence** with the sparsifying matrix (equivalently satisfying for relatively large K-sparse signals, the restricted isometry property)

→ *Fast and reversible transformation:* ***FFT***
Randomness: ***scrambling*** *operator* → ***real-valued scrambled Fourier ensemble***

1. Randomly permute the samples of x
2. FFT
3. Sample randomly $\frac{M}{2} \ll N$ fourier coefficients
4. Separate $\frac{M}{2}$ real and $\frac{M}{2}$ imaginary part (to have real values)

For randomness to the FFT (incoherence)

Actual DMD only real values are implemented (sine pattern and then cosine pattern)

Note: Pay attention to **normalization**!

MATLAB functions

Minimization Algorithms
l1eq_pd: solves the Basis Pursuit problem (P_1)
l1qc_logbarrier: solves quadratically constrained l_1 minimization (P_2)

tveq_logbarrier: solves equality constraint TV minimization (TV_1)
tvqc_logbarrier: solves quadratically constrainted TV minimization (TV_2)

Least Squares

$\arg\min_x \|y - Ax\|_2$

$\rightarrow x = (A^T A)^{-1} A^T y$

`x = A\y` , only rank(A) non zero coefficients

`x = pinv(A)*y` , $\min\|x\|_2$

(TV_1) $\arg\min_x TV(x)$ such that $\mathrm{A}x = y$

(TV_2) $\arg\min_x TV(x)$ such that $\|\mathrm{A}x - y\|_2 \leq \epsilon$

(P_1) $\arg\min_x \|x\|_1$ such that $\mathrm{A}x = y$

(P_2) $\arg\min_x \|x\|_1$ such that $\|\mathrm{A}x - y\|_2 \leq \epsilon$

When the measurements y are corrupted by noise

Randomized constructions
rand: pseudorandom values drawn from the standard uniform distribution on (0,1)
randn: pseudorandom values drawn from the standard normal distribution
randperm: random permutation of integers

Signal Processing
fft: applies the one dimensional *fast fourier transform*

And your most valued friend
help *name*: displays the help for the functionality specified by *name*, such as a function, operator, symbol, method, class, or toolbox
doc *name*: displays the reference page for *name* in the Help browser.

Dictionary Learning

Question: What ψ is the best to represent our signal $x = \psi\alpha$?

$$\arg\min_{\alpha} ||\alpha||_1 \quad such\ that \quad \Phi\psi\alpha = y$$

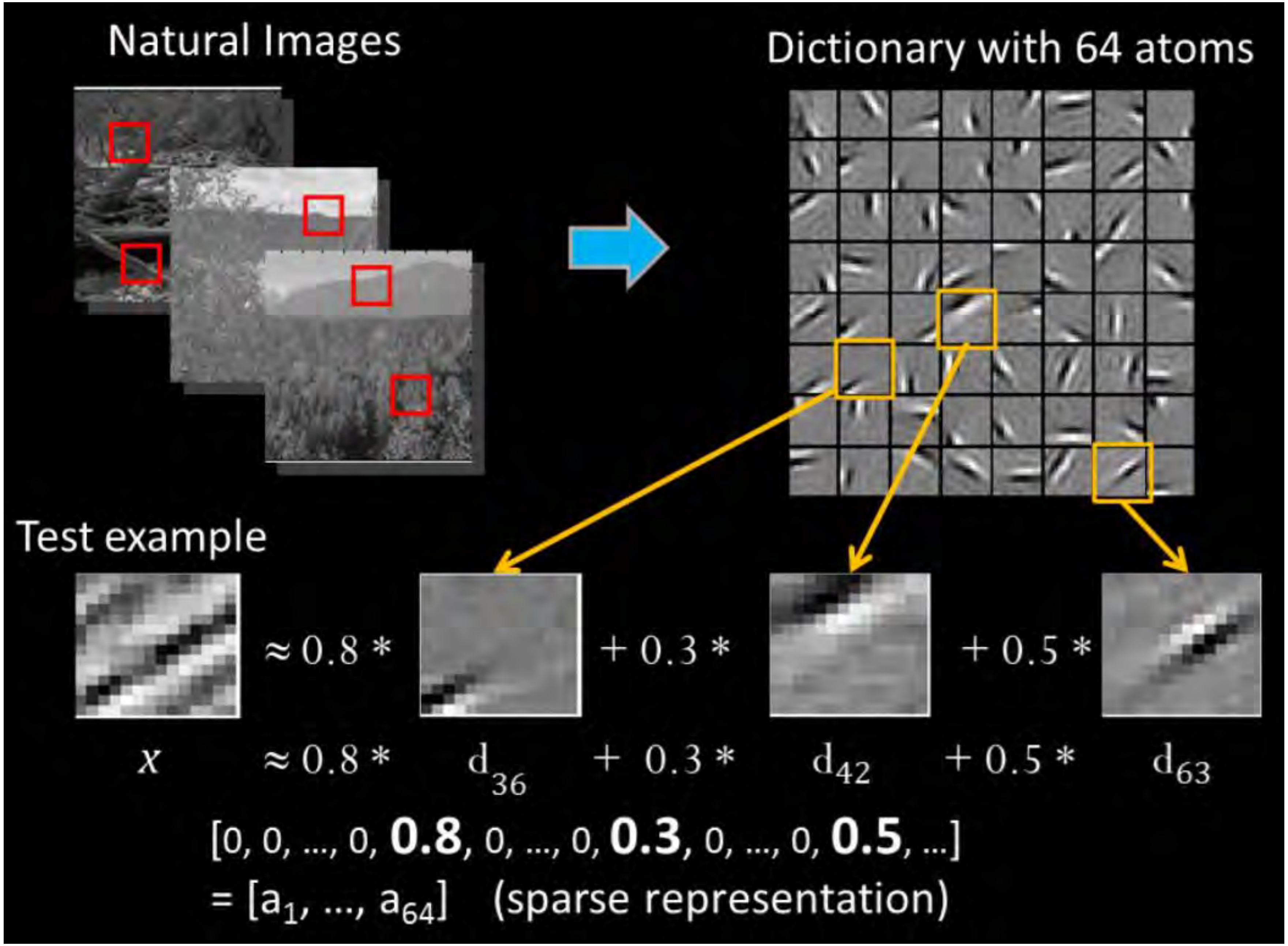
Answer: Optimize ψ and α jointly from the provided data $y \triangleq$ Learn the dictionary ψ

$$\arg\min_{\psi,\alpha} ||\alpha||_1 \quad such\ that \quad ||y - \Phi\psi\alpha||_2^2 \leq \epsilon$$

Numerous algorithms, with the most prominent one being “K-SVD” invented by Michael Elad, Freddy Bruckstein and their student Michal Aharon (CS department).
Main Idea: alternate between 2 steps:

- Sparse Coding (MP or BP)
- Dictionary Update

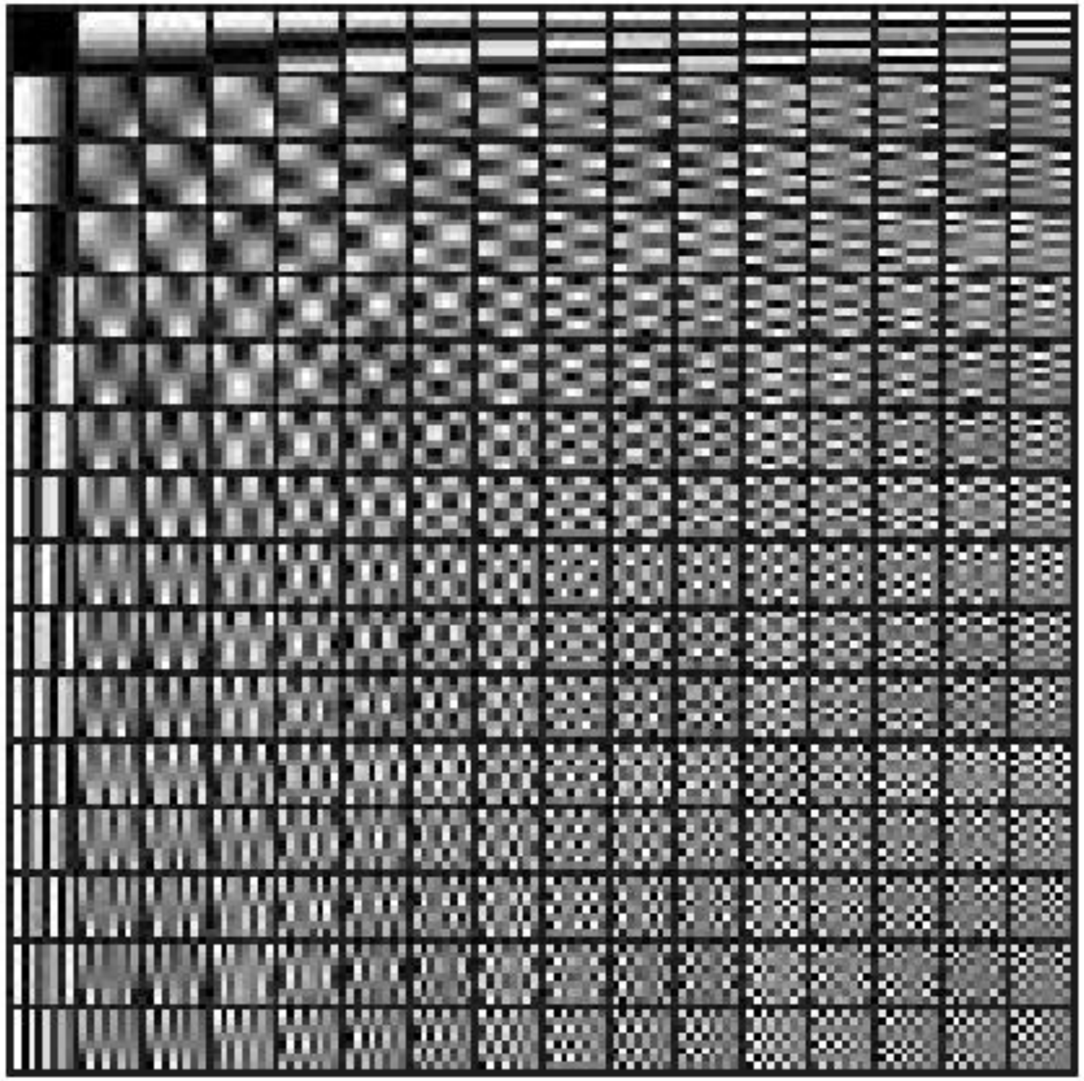
Dictionary Learning



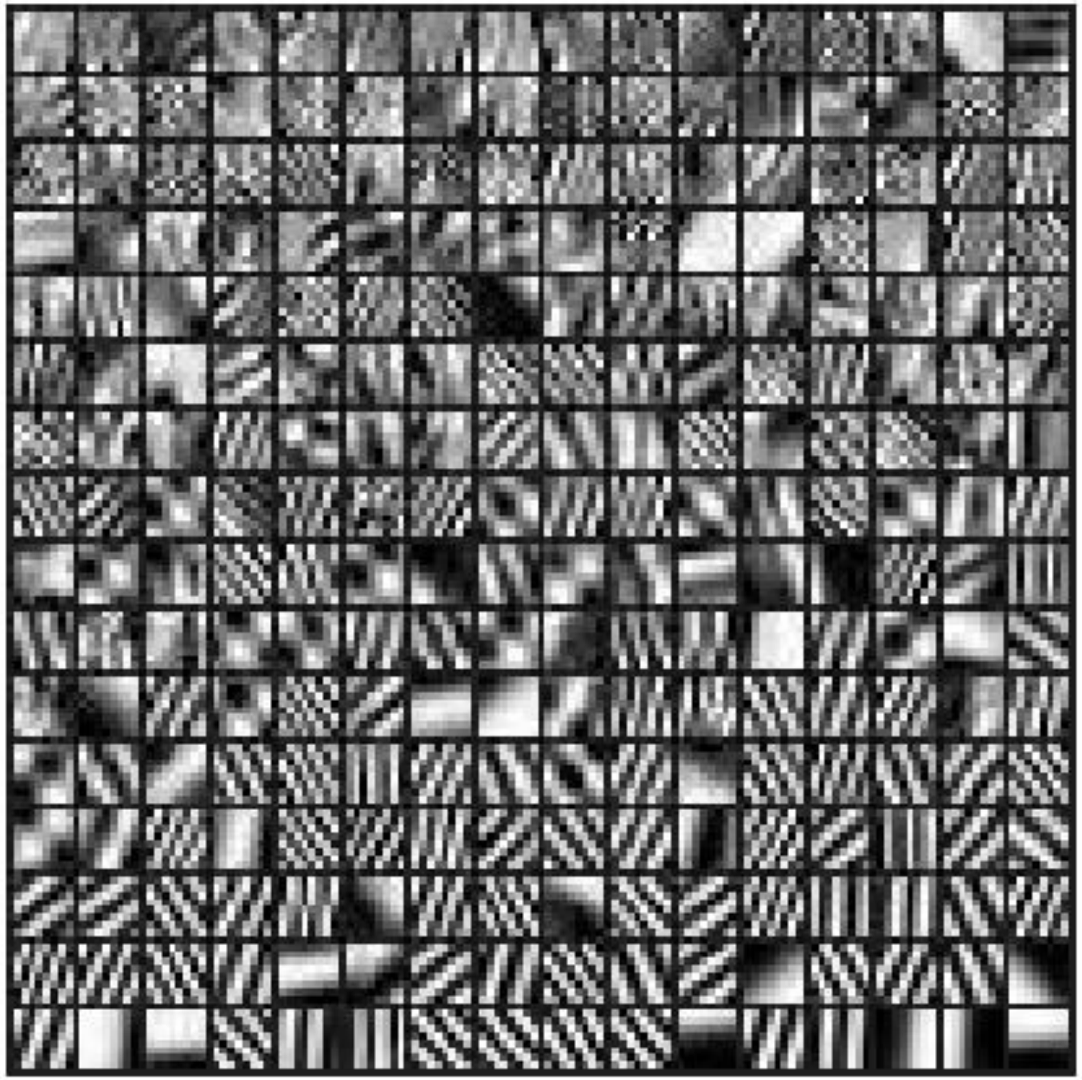
Andrew Ng, ECCV 2010 Tutorial

Dictionary Learning: Image “Barbara”

DCT



Learned Dictionary



Aharon M., *IEEE Trans. Image Proc.*, 2006

הטכניון
מכון טכנולוגי לישראל

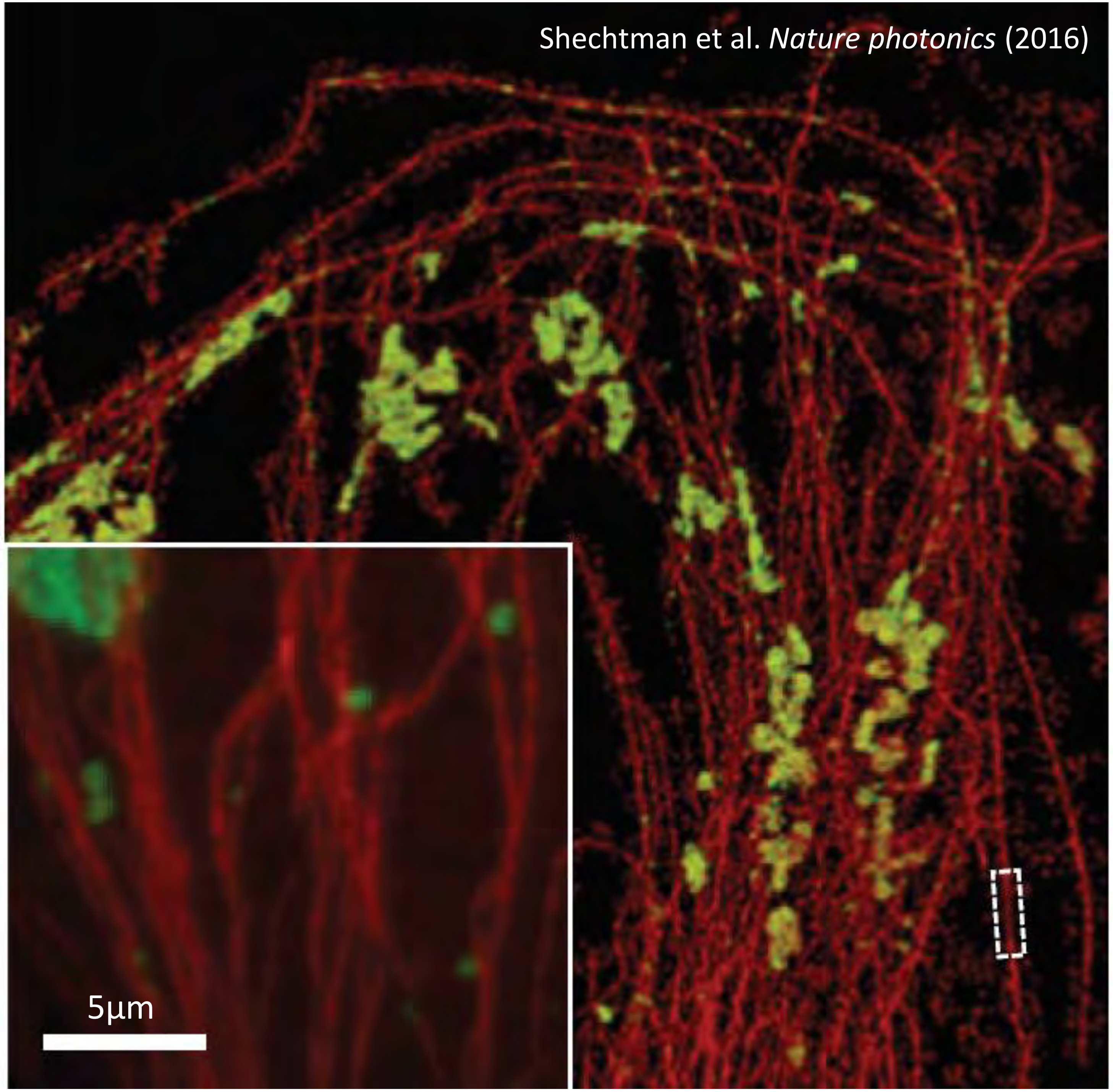
Department of Biomedical Engineering, Technion

Computational optical imaging 336547

Tutorial 7 – Localization Microscopy

Elias Nehme & Yoav Shechtman

8 December 2020



Super-Resolution Localization Microscopy Concept

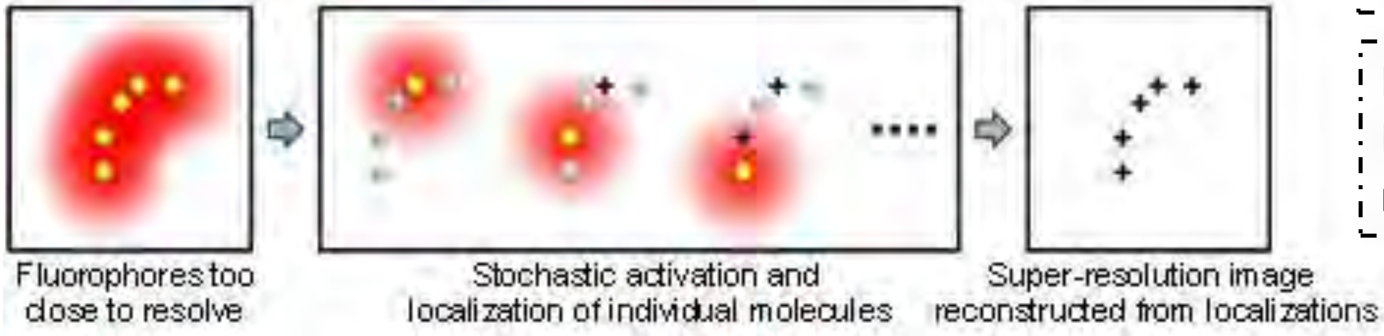
$$d = \frac{\lambda}{2NA} = \frac{\lambda}{2n\sin\theta}$$

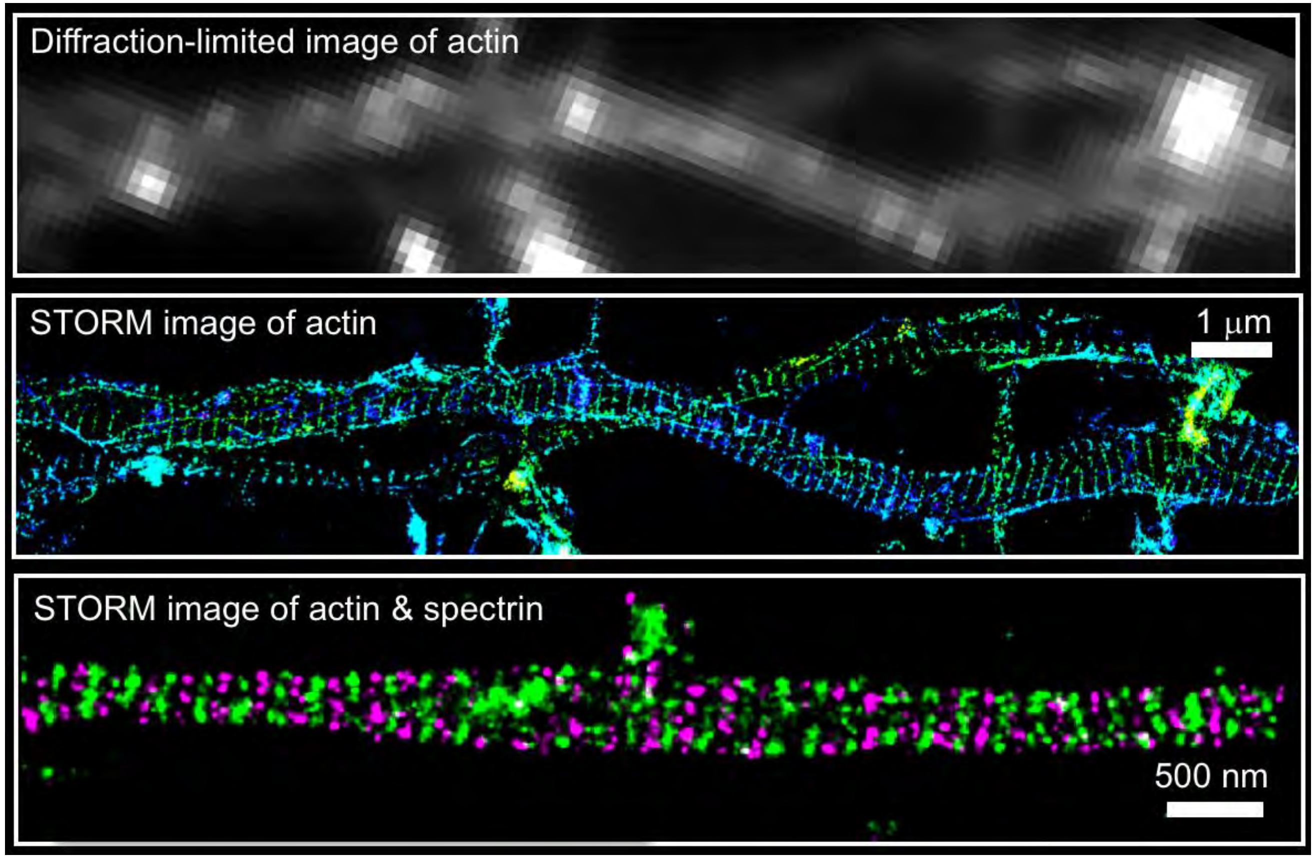
For visible light: Lateral resolution~**200nm**

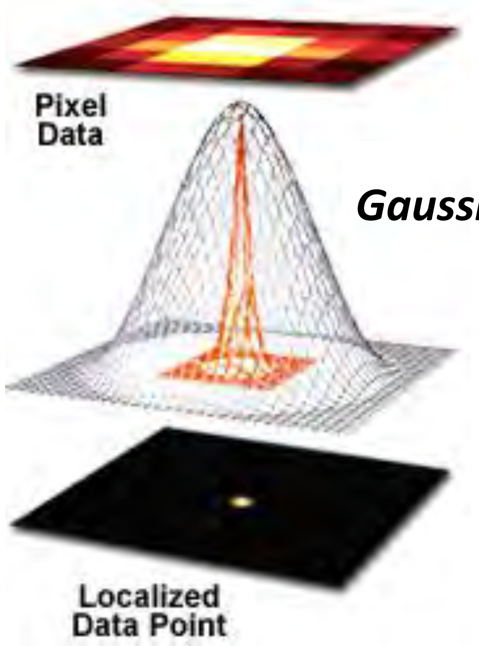
Resolving close fluorophores:

1. Switchable fluorophores

2. Powerful localization algorithms







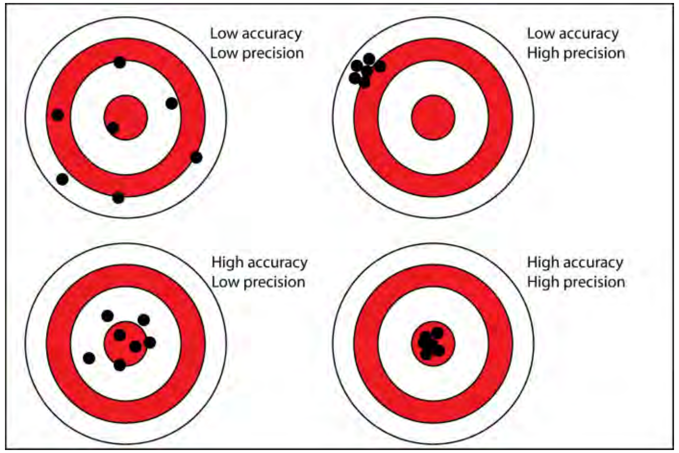
Gaussian fit

What determines the dimensions of the localized data point i.e. precision?

What determines the accuracy with which the data point is localized?

Probing biology at the nm scale via fluorescence

Localization Precision and Accuracy



Localization Precision: *The spread of the estimates around its mean value*

$$\sigma_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(x_{p,i} - \bar{x}_p\right)^2} \qquad \mathrm{FWHM}_x = 2\sigma_x\sqrt{2\ln 2}$$

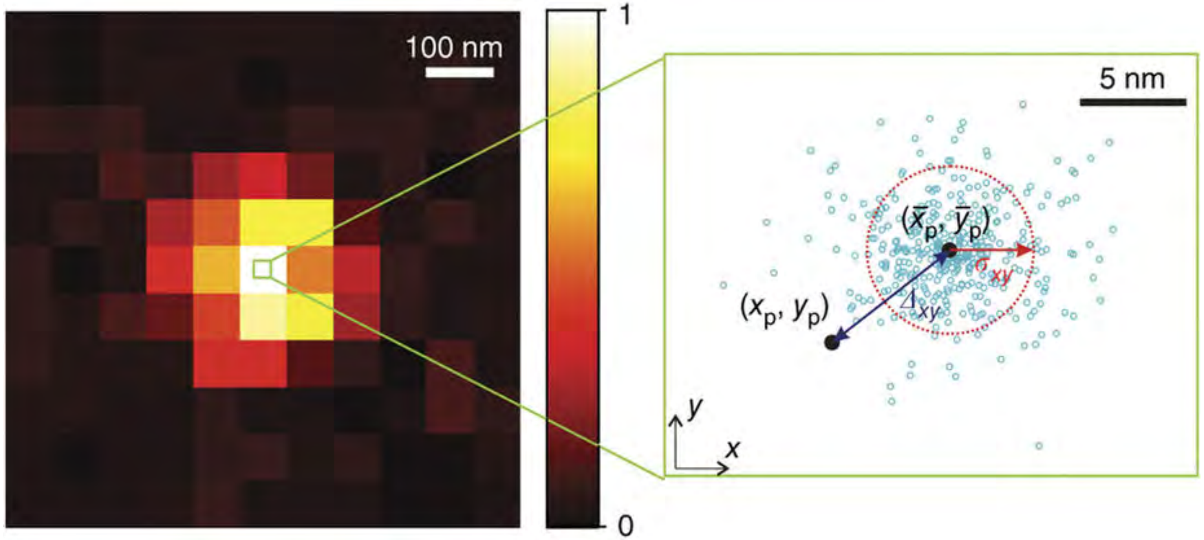
x_p: true position of a particle
$x_{p,i}$: estimate i of x_p
n: number of estimates
$\bar{x}_p$: mean of all estimates $x_{p,i}$

Localization Accuracy: *The deviation of the mean measured position coordinates from the true position coordinate*

$$\Delta_x = \bar{x}_p - x_p$$

Δ_x=0 for **an unbiased estimation** → Accurate
Calculated only when the true position is known

Experimentally recorded image of a single emitter:



Blue circles → experimentally determined position estimates from different images of the same emitter

(x_p, y_p) – real particle position

$(\bar{x}_p, \bar{y}_p)$ – average of the estimated positions

$\sigma_{xy} = 0.5 \times \sqrt{\sigma_x^2 + \sigma_y^2}$ – lateral localization precision

$\Delta_{xy} = \sqrt{\Delta_x^2 + \Delta_y^2}$ – lateral localization accuracy

Localization Precision and Accuracy

Localization Accuracy

- The algorithm estimating x_p must be **unbiased**

$$\boxed{\Delta_x = \bar{x}_p - x_p}$$

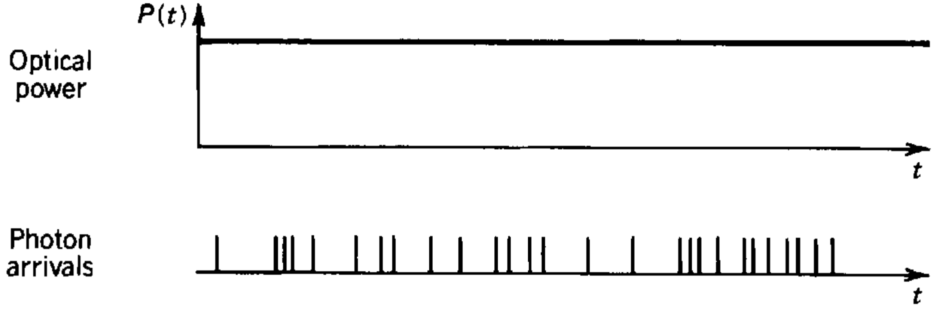
- **Insensitive to shot noise** (does not involve individual measurements $x_{p,i}$), sensitive to background, spatial photon distribution, detector and sample properties

- **No fundamental limit on the achievable localization accuracy**

Localization Precision

- Coherent light has a constant optical power

$$\boxed{\sigma_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(x_{p,i} - \bar{x}_p\right)^2}}$$



Average on the random photon arrivals is constant

Localization Precision and Accuracy

Random arrival of photons in a light beam of power P within intervals of duration T. **Although the optical power is constant the number *n* of photons arriving within each interval is random**.

Photon registration: Poisson distribution

$$p(n) = \frac{\bar{n}^n e^{-\bar{n}}}{n!}, n = 0,1,2, \ldots \quad \begin{cases} \text{mean: } \bar{n} \\ \text{variance: } \sigma_n^2 = \bar{n} \end{cases}$$

The **number of photons** arriving in a certain time interval follows a **Poisson distribution**, the standard deviation of which is known as **shot noise**

e.g. the presence of $\bar{n} = 100$ photons is accompanied by an inaccuracy of $\pm\sigma_n = 10$ photons

Localization Precision

Due to shot noise ***each image will have a slightly different center*** → the estimated fluorophore's position will give ***different results for each image***

Precision is affected by <u>number of photons</u>, <u>emission profile</u> of particles (fixed dipole, translation movement, diffraction of the light microscope), <u>detector and sample properties</u>

Minimum Variance Unbiased Estimation (MVUE)

Cramer-Rao Lower Bound

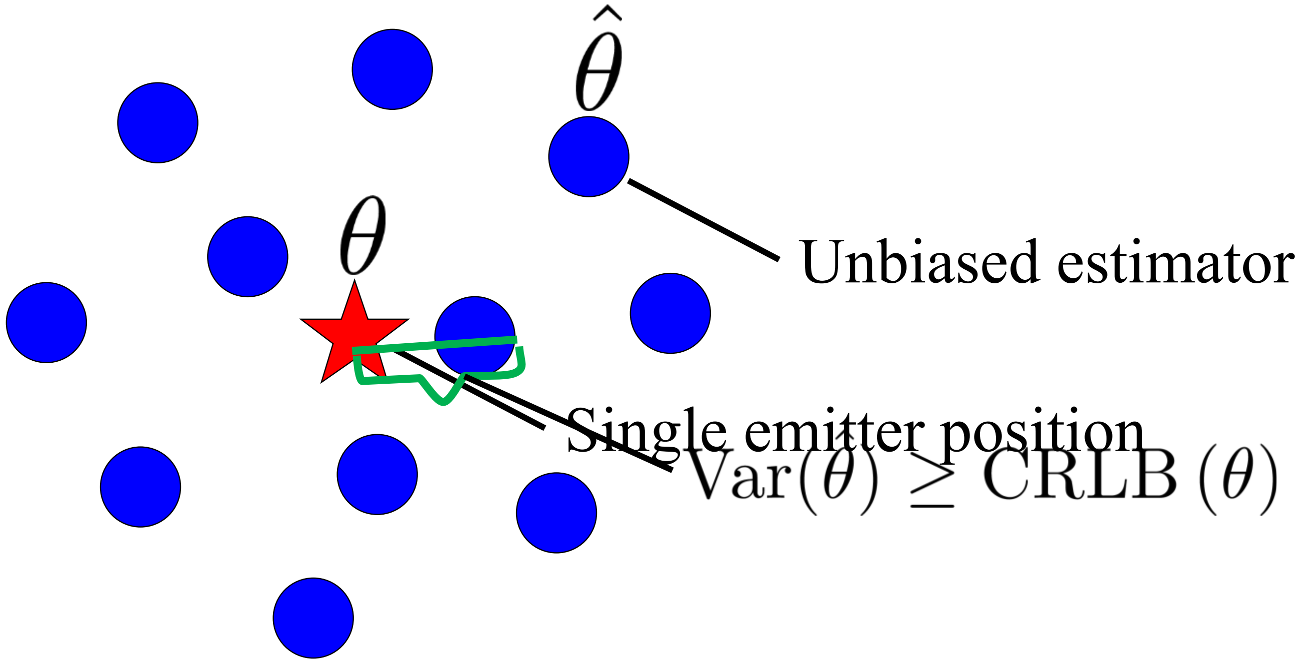
The best localization precision theoretically achievable is given by the square root of the Cramér-Rao lower bound (CRLB), which is defined as the **smallest possible variance any unbiased estimation algorithm can have**

Assumed measurement model

$$Y \sim P\left(y;\theta\right)$$

Likelihood function

$$\ell\left(\theta\right) = \prod_k P\left(y_k;\theta\right)$$



Fisher information matrix

$$\left[\mathcal{F}\left(\theta\right)\right]_{i,j} = \mathbb{E}\left[\left(\frac{\partial}{\partial\theta_i}\ell\left(\theta\right)\right)\cdot\left(\frac{\partial}{\partial\theta_j}\ell\left(\theta\right)\right)\middle|\theta\right] \rightarrow \mathrm{CRLB}\left(\theta_i\right) = \left[\mathcal{F}\left(\theta\right)^{-1}\right]_{i,i}$$

Localization Precision – CRLB

Cramer-Rao Lower Bound

The best localization precision theoretically achievable is given by the square root of the Cramér-Rao lower bound (CRLB), which is defined as the **smallest possible variance any unbiased estimation algorithm can have**

Spatial distribution of photon positions that is dictated by the **emission profile of the particle** in combination with the light diffraction in the microscope

$$f\left(x, y; \theta = [x_0, y_0]\right)$$

Assuming shot noise, each pixel measurement will be Poisson distributed:

$$Y_{r,k} \sim \mathcal{P}\left(\lambda = f\left(x_r, y_k; \theta = [x_0, y_0]\right)\right)$$

Resulting Fisher information matrix elements

$$[\mathcal{F}(\theta)]_{i,j} = \sum_r \sum_k \frac{1}{f(x_r, y_k; \theta = [x_0, y_0])} \frac{\partial f}{\partial \theta_i}\Big|_{(x_r, y_k)} \cdot \frac{\partial f}{\partial \theta_j}\Big|_{(x_r, y_k)}, \quad 1 \leq i, j \leq 2$$

Localization Precision – CRLB

Cramer-Rao Lower Bound

The best localization precision theoretically achievable is given by the square root of the Cramér-Rao lower bound (CRLB), which is defined as the **smallest possible variance any unbiased estimation algorithm can have**

Isotropic emitter in or **close to the focal plane**, the PSF is approximately Gaussian:

$$f\left(x,y;\theta=[x_0,y_0]\right)\approx\frac{N}{2\pi\sigma^2}e^{-\frac{1}{2\sigma^2}\left[(x-x_0)^2+(y-y_0)^2\right]}+b$$

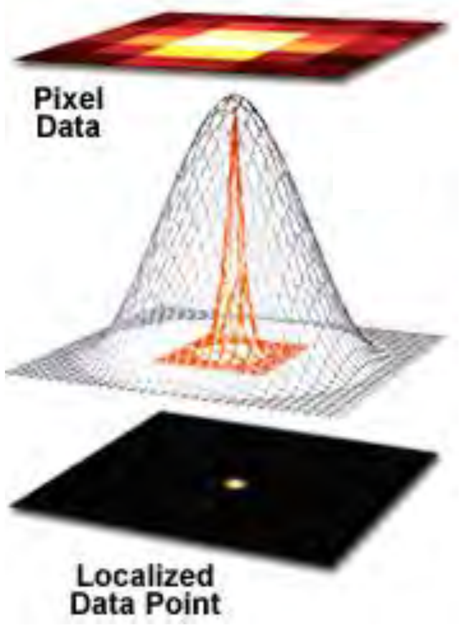
Considering **only shot noise ($\boldsymbol{b=0}$)**, the precision limit becomes simple

$$\sigma_{\hat{x}_0}\geq\frac{\sigma}{\sqrt{N}}$$

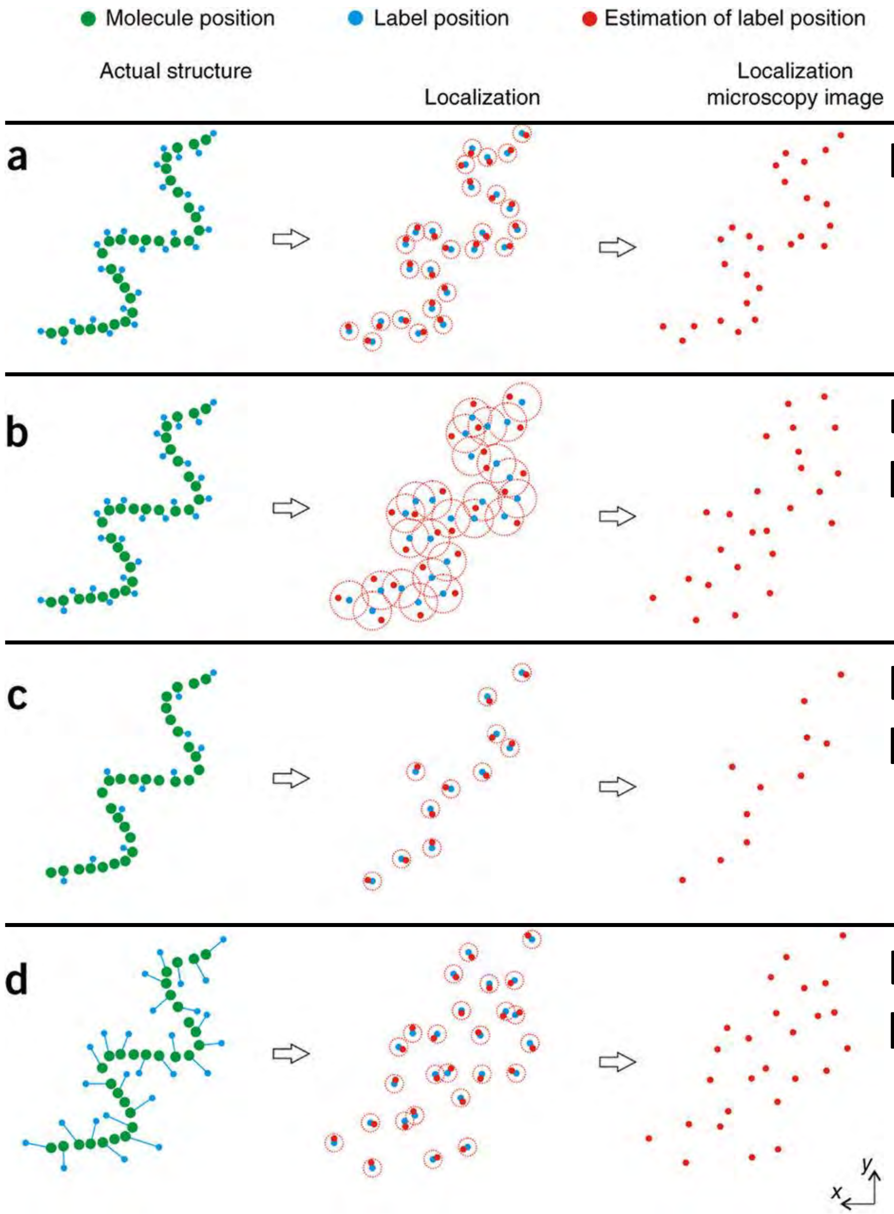
Considering also **background and pixelization**:

a^2 pixel area

$$\sigma_{\hat{x}_0}\geq\sqrt{\frac{\sigma_a^2}{N}\left(\frac{9}{16}+\frac{8\pi\sigma_a^2b^2}{Na^2}\right)},\quad\sigma_a^2=\sigma^2+\frac{\boxed{a^2}}{12}$$



Spatial Resolution vs Localization Precision and Accuracy



a High spatial resolution

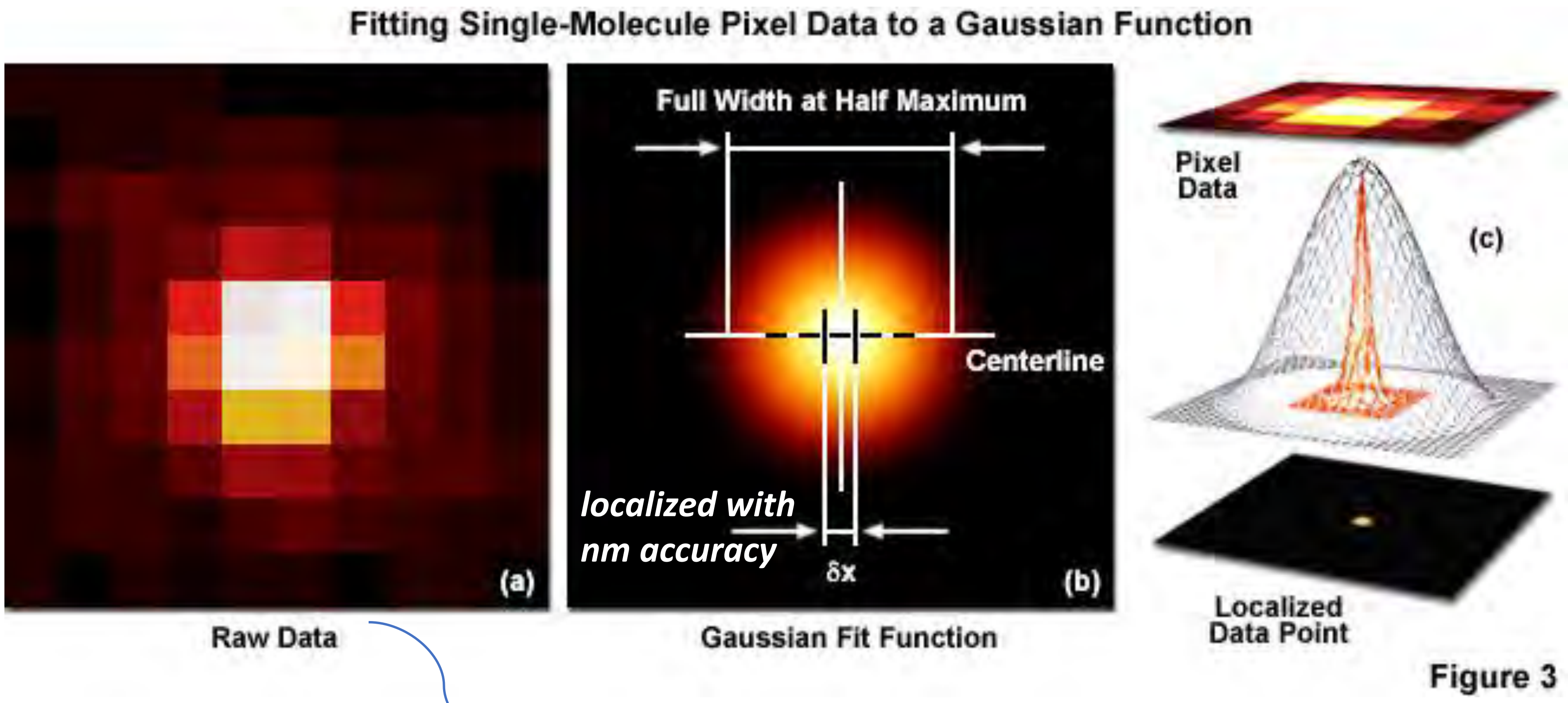
b Lower spatial resolution due to lower localization precision

c Lower spatial resolution due to lower label density (Nyquist criterion)

d Lower spatial resolution due to higher label displacement

Computing Localization precision and accuracy of an algorithm is **not the same** as determining the resolution of an image produced by a localization algorithm

The art of localizing emitters



Mathematically treated to fit a two-dimensional Gaussian function and localized with nanometer accuracy

Sub-pixel 2D localization of molecules:

- ❖ Center of Gravity (CoG)
- ❖ Least Squares (LS)
- ❖ Weighted Least Squares (WLS)
- ❖ Maximum Likelihood Estimation (MLE)

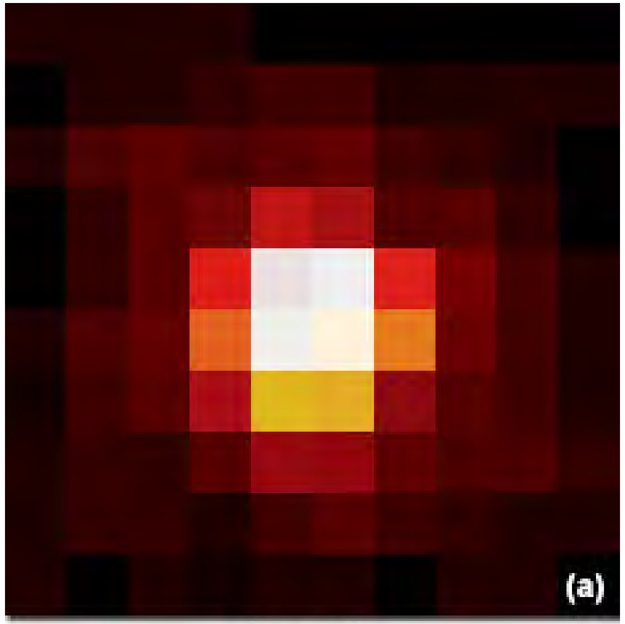
Center-of-Gravity (CoG)

Mean pixel positions weighted by the intensity of the image data:

$$\hat{x}_0 = \frac{\sum_{x,y \in D} x\tilde{I}(x,y)}{\sum_{x,y \in D} \tilde{I}(x,y)}, \quad \hat{y}_0 = \frac{\sum_{x,y \in D} y\tilde{I}(x,y)}{\sum_{x,y \in D} \tilde{I}(x,y)}$$

- Does not require **any prior knowledge**
- **Very fast** (non-iterative algorithm)
- Does **not estimate the intensity or imaged size of molecules**
- **Sensitive to noise**
- **Biased estimator in the presence of background** (towards center for uniform background)

➡ If the **image profile is Gaussian,** the **center of gravity estimator is a maximum likelihood estimator**



Raw Data

Fitting point-spread-function (PSF) models

A single molecule emitter is treated as an **incoherent point source and is described by the PSF**

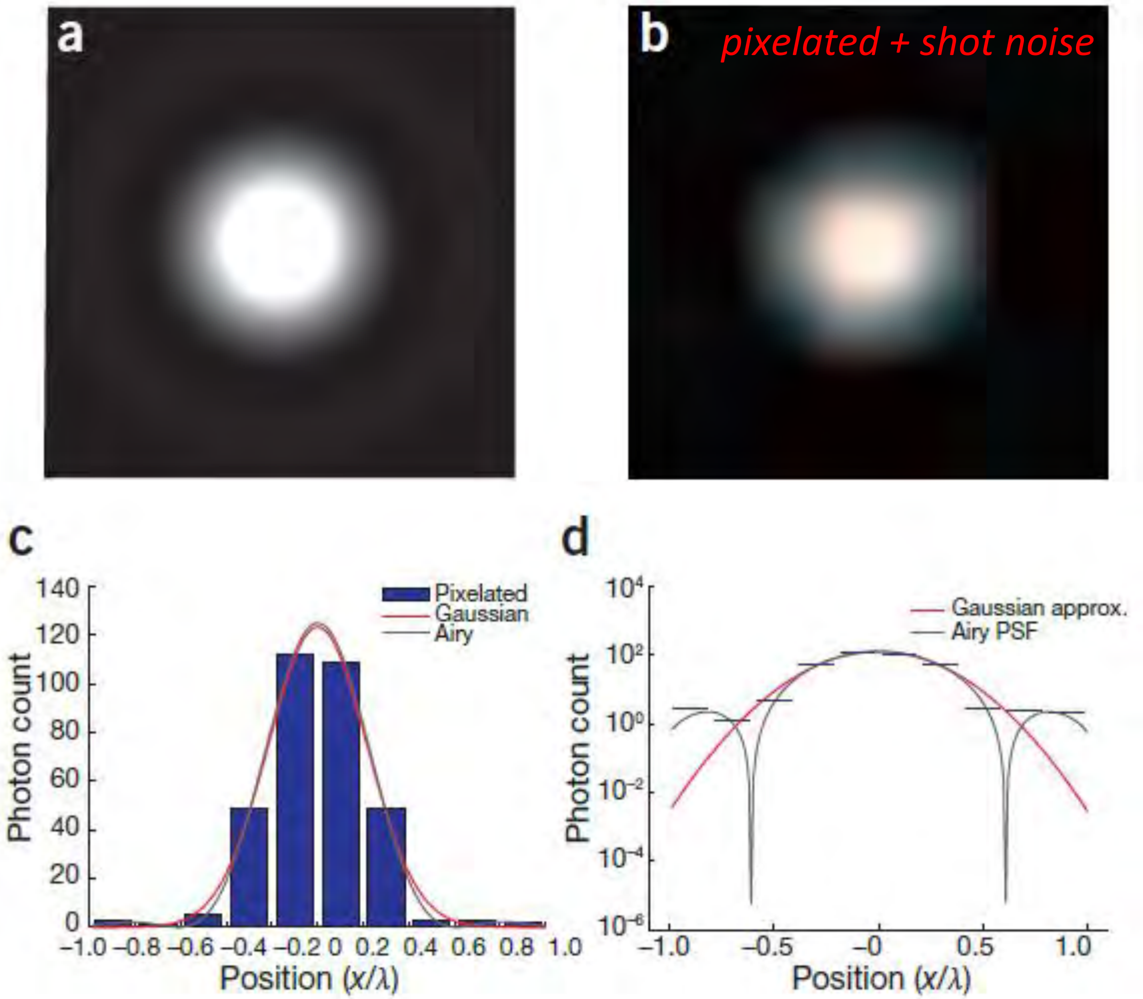
PSF - proportional to **the average number of photons** at a given position relative to the source

Airy PSF is tedious for many practical calculations → PSF of an isotropic source is often approximated as a **Gaussian function**

Gaussian approximation gives useful and reasonably accurate results for **focused images of fluorophores**

In the tails the approximation can break down as a Gaussian decays more rapidly than many PSFs → Poses issues in minimizing discrepancies between the model and the data in the edges of the image.

Solution: Using a small ROI (tradeoff it discards useful information)



Fitting point-spread function (PSF) models

Gaussian function ≈ real PSF of a microscope (due to pixelation and noise)

- Simplicity
- Robustness
- Computation efficiency

$$\boldsymbol{\theta} = [\theta_x, \theta_y, \theta_\sigma, \theta_N, \theta_b]$$

θ_x sub-pixel molecular x-coordinate
θ_y sub-pixel molecular y-coordinate
θ_σ imaged size of the molecule
θ_N total number of photons emitted by the molecule
θ_b background offset

Symmetric 2D Gaussian function

$$\mathrm{PSF}_G(x, y|\boldsymbol{\theta}) = \frac{\theta_N}{2\pi\theta_\sigma^2} e^{-\frac{(x-\theta_x^2)^2 + (y-\theta_y^2)^2}{2\theta_\sigma^2}} + \theta_b$$

Expected photon count at the integer pixel position (x, y) for the parameters $\boldsymbol{\theta} = [\theta_x, \theta_y, \theta_\sigma, \theta_N, \theta_b]$

Integrated form of a symmetric 2D Gaussian function

$$\mathrm{PSF}_G(x, y|\boldsymbol{\theta}) = \theta_N E_x E_y + \theta_b$$

$$E_x = \frac{1}{2}\mathrm{erf}\left(\frac{x - \theta_x + \frac{1}{2}}{\sqrt{2}\theta_\sigma}\right) - \frac{1}{2}\mathrm{erf}\left(\frac{x - \theta_x - \frac{1}{2}}{\sqrt{2}\theta_\sigma}\right)$$

$$E_y = \frac{1}{2}\mathrm{erf}\left(\frac{y - \theta_y + \frac{1}{2}}{\sqrt{2}\theta_\sigma}\right) - \frac{1}{2}\mathrm{erf}\left(\frac{y - \theta_y - \frac{1}{2}}{\sqrt{2}\theta_\sigma}\right)$$

Considers the **discrete nature of pixels present in digital cameras**; assuming a uniform distribution of pixels with unit size

The parameters are varied to find the values that give the best 'fit' to the data

Least-squares methods

Optimization problem typically solved by the Levenberg-Marquadt algorithm

expected photon count (model)

$$\hat{\boldsymbol{\theta}} = \arg\min_{\theta} \chi^2(\boldsymbol{\theta}|D) = \arg\min_{\theta} \sum_{x,y\in D} w(\tilde{I}(x,y) - \mathrm{PSF}(x,y|\boldsymbol{\theta}))^2$$

observed photon count

$$\boldsymbol{\theta} = [\theta_x, \theta_y, \theta_\sigma, \theta_N, \theta_b]$$

θ_x sub-pixel molecular x-coordinate
θ_y sub-pixel molecular y-coordinate
θ_σ imaged size of the molecule
θ_N total number of photons emitted by the molecule
θ_b background offset
w one over expected **variance** of the signal per pixel

<u>least-squares</u>: $w = 1$
All measurements are equally significant

<u>weighted least-squares</u>: $w = 1/PSF(x,y|\theta)$
Considers the uncertainty in the number of detected photons

- **No detailed knowledge** (weighted) **or none** ($w = 1$) **required on noise**

- Weighting **gives extra importance to the tails of the PSF** – criteria often used to choose between LS and WLS in low background conditions (misspecifying the tail is <u>less of an issue when there is substantial background</u>)

- Weighting should be done with respect to the **expected variance** (i.e. ***the model prediction***)

➡ If the **noise can be approximated as Gaussian**, the **weighted least squares algorithm is a maximum likelihood estimator**

For high-background fluorescence (>10 photons/pixel) OR high photon count

Maximum-Likelihood Estimator

Likelihood of the parameters $\boldsymbol{\theta}$

photons are usually independent of each other

$$L(\boldsymbol{\theta}|D) = \prod_{x,y \in D} \frac{\mathrm{PSF}(x,y|\boldsymbol{\theta})^{\tilde{I}(x,y)} e^{-\mathrm{PSF}(x,y|\boldsymbol{\theta})}}{\tilde{I}(x,y)!}$$

Photon registration: Poisson distribution

$\mathrm{PSF}(x,y|\boldsymbol{\theta})$ expected photon count

$\tilde{I}(x,y)$ observed photon count

Log Likelihood – Optimization problem typically solved by the Nelder-Mead method

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\theta} \sum_{x,y \in D} [\tilde{I}(x,y) \ln(\mathrm{PSF}(x,y|\boldsymbol{\theta})) - \mathrm{PSF}(x,y|\boldsymbol{\theta})]$$

- **Requires a model of noise** (**shot noise**, or shot noise plus Gaussian read noise)
- Requires a good PSF model but can use an approximate PSF width (**PSF width can be a fit parameter**)
- **Known to be unbiased, and consistent!**
- **In low background, center and crop the ROI to avoid PSF tail misspecifications**
- **For high-background fluorescence**, the noise can be approximated as **constant-variance Gaussian model**

➡ MLE estimates the positions with (often) the highest possible precision (**approaches CRLB**)

LS vs MLE

Favor MLE when adequate information is available on PSF shape and camera performance

Table 1 | Comparison of the MLE and LS criteria for localization of single isotropic point sources

Maximum-likelihood estimation	Least-squares criterion
• Can, in principle, achieve theoretical limit of precision	• Often has lower precision but close to MLE precision for high photon counts and background
• Works best with a good model of camera noise	• Requires no information about noise; equivalent to MLE for Gaussian noise
• Requires a good PSF model for optimal performance but can use approximate PSF shape; PSF width can be a fit parameter	• Robust against misspecification of PSF shape but requires well-specified PSF width, or PSF width can be a fit parameter
• Takes more time to converge if PSF width is a fit parameter	
• Typically implemented with analytical PSF (i.e., a formula) but has been implemented with measured PSFs for 3D imaging[41,48]	
• Potential small-denominator problem when background is low and PSF tail is misspecified; this is solvable by proper centering and sizing of the ROI	
• Suitable for GPU implementation; fast algorithm available that fits *x* and *y* independently[56]	• Suitable for GPU implementation; fast algorithm available that fits only *x* and *y* and ignores other fit parameters[16]

Localization algorithms

	Fitting approach and PSF	Common implementations	Noise model	Notes for use
Single-fluorophore fits	MLE with isotropic PSF	Ober lab[a], Lidke lab (GPU implementation[47]), rapidSTORM[52], M2LE[56]	All assume shot noise; Ober's software also allows Gaussian camera read noise	Good for fluorophores with freely rotating dipole moments. Usually use a Gaussian PSF. Defocus can be accounted for via variable PSF width. M2LE includes an ellipticity test for rejection of multiple-fluorophore images
	MLE with elliptical PSF	Lidke lab, rapidSTORM	Shot noise	Most useful for astigmatism-based 3D imaging if the model assumes an ellipse oriented along one of the detector axes. Useful for rejection of two-molecule overlaps when the ellipse is arbitrarily oriented
	MLE with isotropic PSF, EMCCD excess noise and read noise	UAIM by Ober lab[46]	Combination of Poisson noise, electron-multiplication noise of EMCCD and Gaussian read noise	Optimized for use with very high magnification, but the noise model is applicable to almost any single-molecule experiment with an EMCCD
	LS with experimental PSF	Bewersdorf lab[41]	No detailed assumptions, but performance approaches theoretical limit if noise is a Gaussian; background correction is possible	Developed with particular attention to defocused fluorophores for 3D biplane imaging
	Fast LS with circular Gaussian PSF	Gaussian mask[16]		Practical when the PSF is not known in detail or when computational time is crucial
	Center of mass	Virtual window center of mass (VWCM)[63]	No detailed assumptions	Appropriate for diffusing fluorophores[64]. Good first-pass estimate to seed an iterative fitting routine. Designed for background correction
	fluoroBancroft[19]	LivePALM[66]	No detailed assumptions	Assumes a Gaussian PSF and requires single iteration
	Radial symmetry	Parthasarathy lab[32], Ma lab[67,68]	No detailed assumptions	PSF is only assumed to be radially symmetric. Performance is good even for nonradial PSFs[32]

Practical considerations in Localization Microscopy

1. Image filtering and **feature enhancement**
   a. Averaging filter
   b. Gaussian filter
   c. Lowered Gaussian filter
   d. Difference-of-Gaussian filter
   e. Wavelet filter
   f. Median filter
   g. No filter

2. Finding **approximate positions** of molecules
   a. Detection of local intensity maxima
   b. Non-maximum suppression
   c. Centroid of connected components
   d. Threshold selection

3. Sub-pixel **2D localization** of molecules
   a. Today's awesome tutorial

4. Sub-pixel **3D localization** of molecules
   a. PSF model
   b. Defocusing models
   c. Calibration of the imaging system
   d. Localization uncertainty

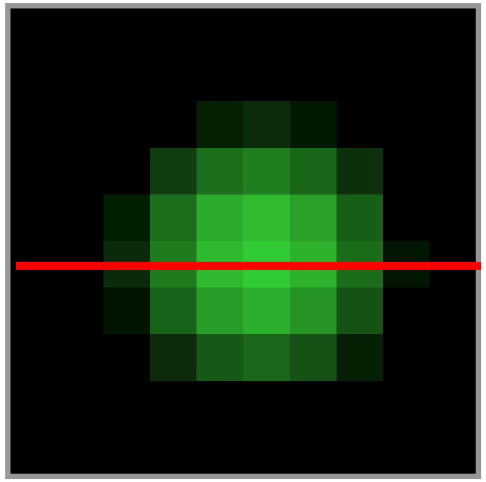
5. The **Crowded-field problem**
   a. Multiple-emitter fitting analysis
   b. Model selection

6. **Post-processing analysis**
   a. Removing molecules with poor localization
   b. Local density filter
   c. Merging of reappearing molecules
   d. Lateral drift correction (cross-correlation)
   e. Z-stage scanning

7. **Visualization methods**
   a. Scatter plot
   b. Histogram
   c. Averaged shifted histograms
   d. Gaussian rendering

Localizing sparse emitters

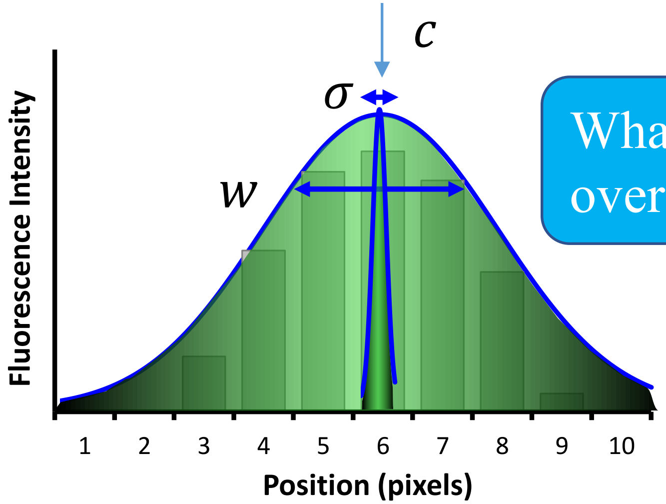
Diffraction-limited spot
recorded on camera



Localization precision

$$\sigma \propto \frac{1}{\sqrt{N}}$$

500 photons ↔ ~25 nm



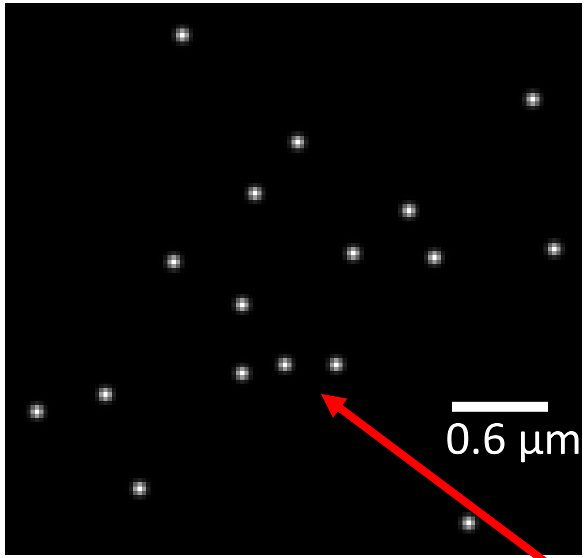
What about overlap?

Localize: fit to a Gaussian

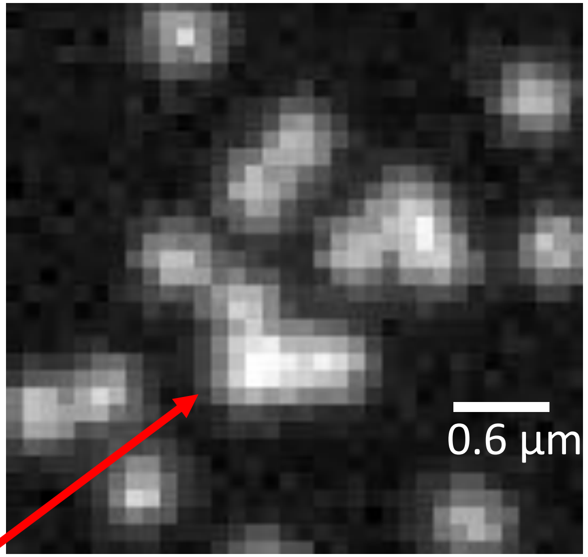
$$I = \hat{A}\exp\left(\frac{-(x-\hat{c})^2}{2\hat{w}^2}\right) + \hat{B}$$

High density fitting is challenging

Ground Truth

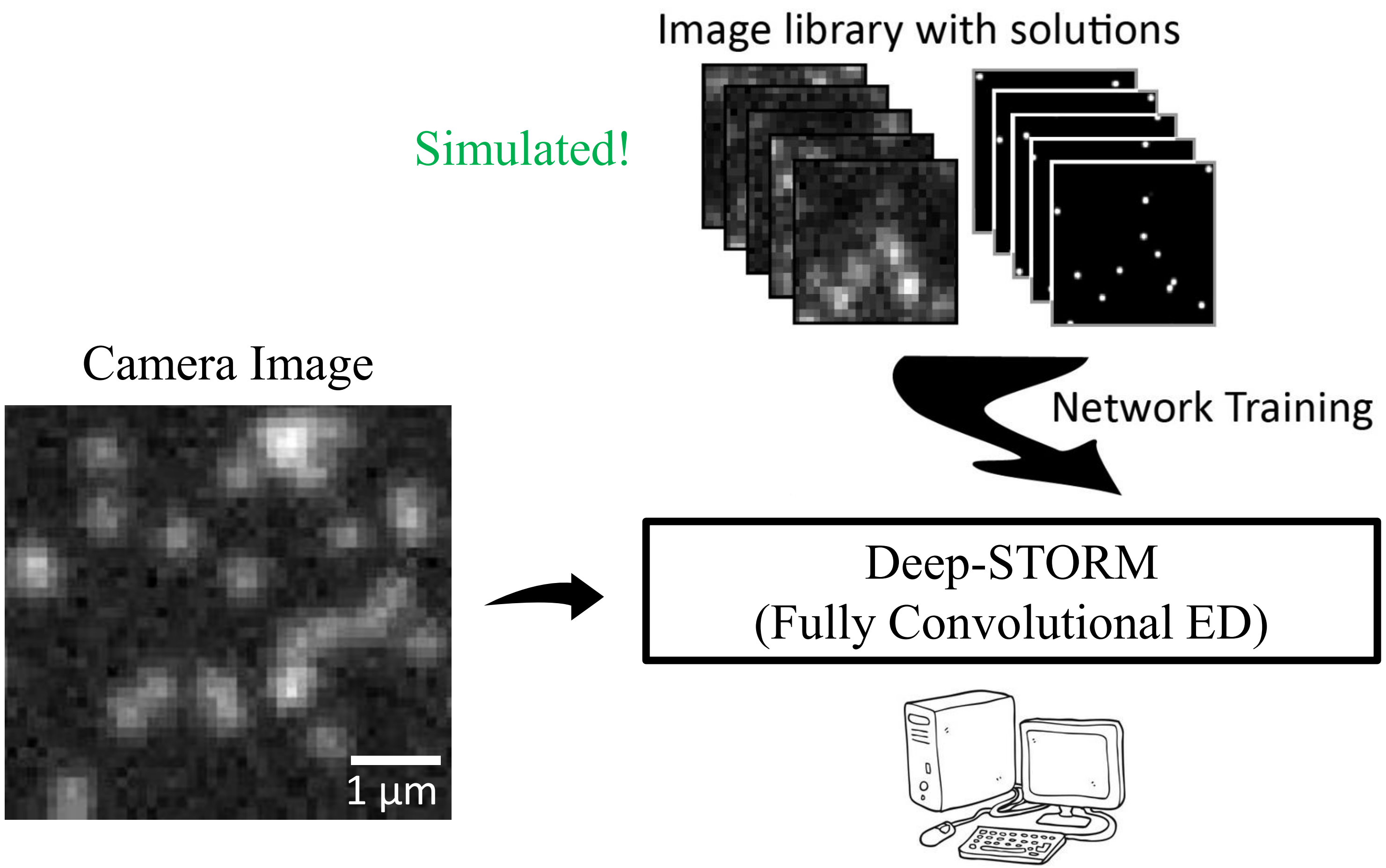


Camera Image

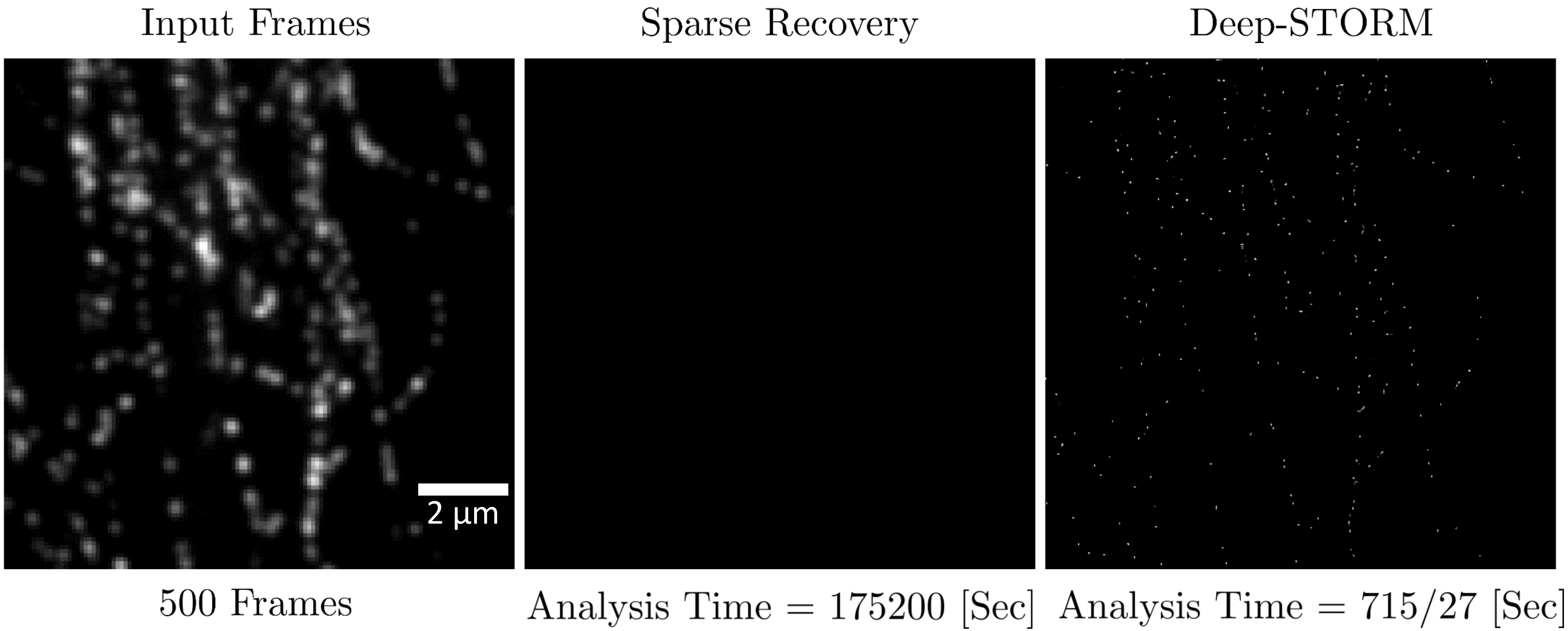


Multi-emitter Gaussian
fitting will perform poorly!

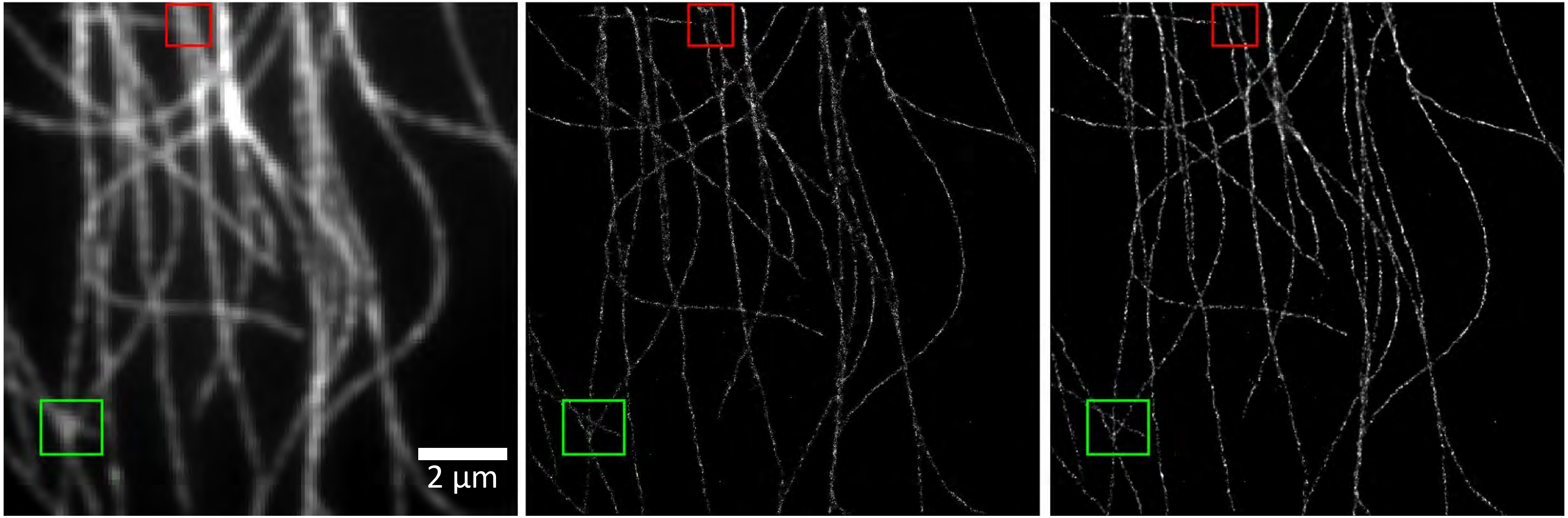
Deep-STORM general idea



Real microtubules experiment



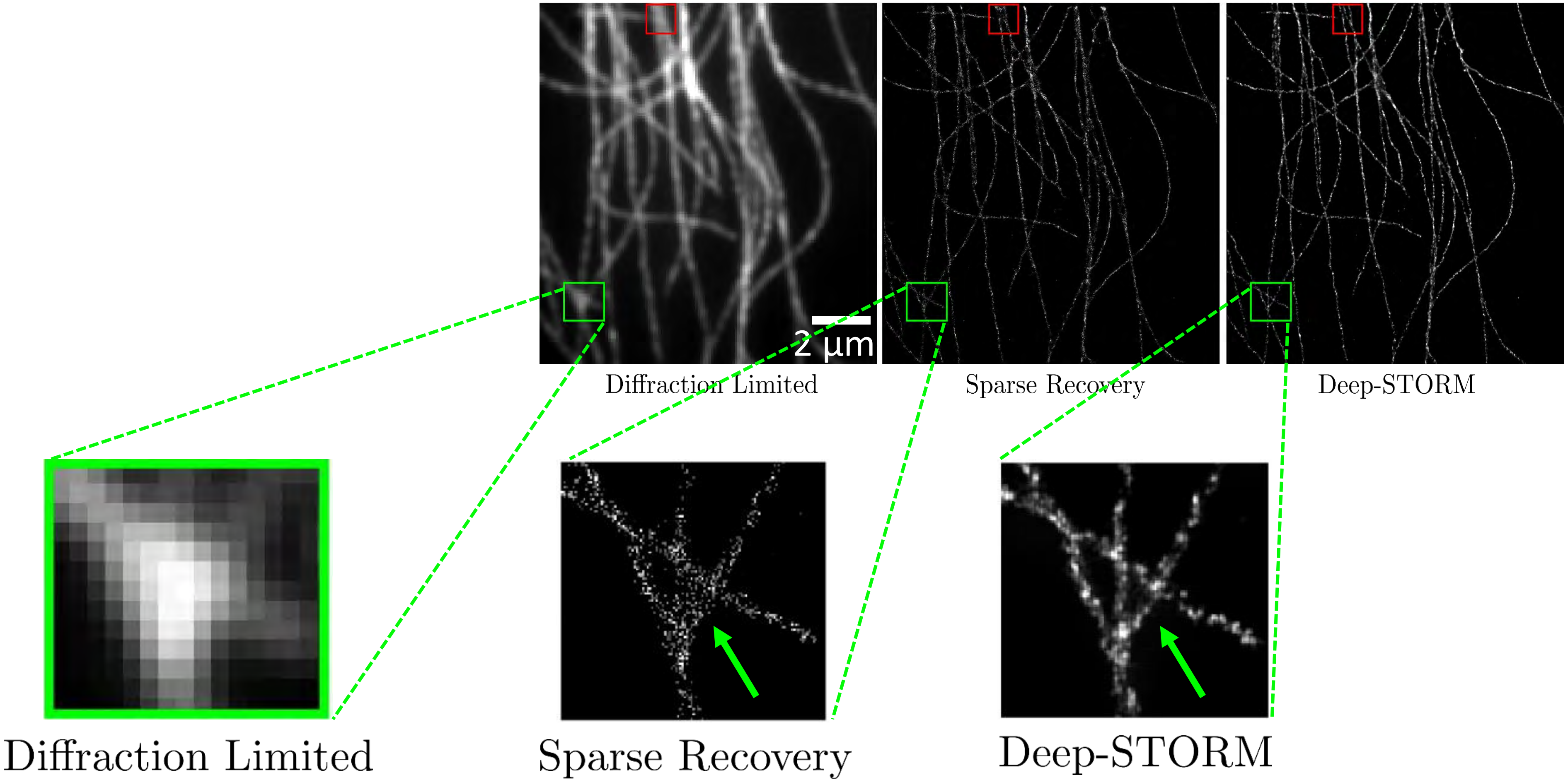
Real microtubules experiment



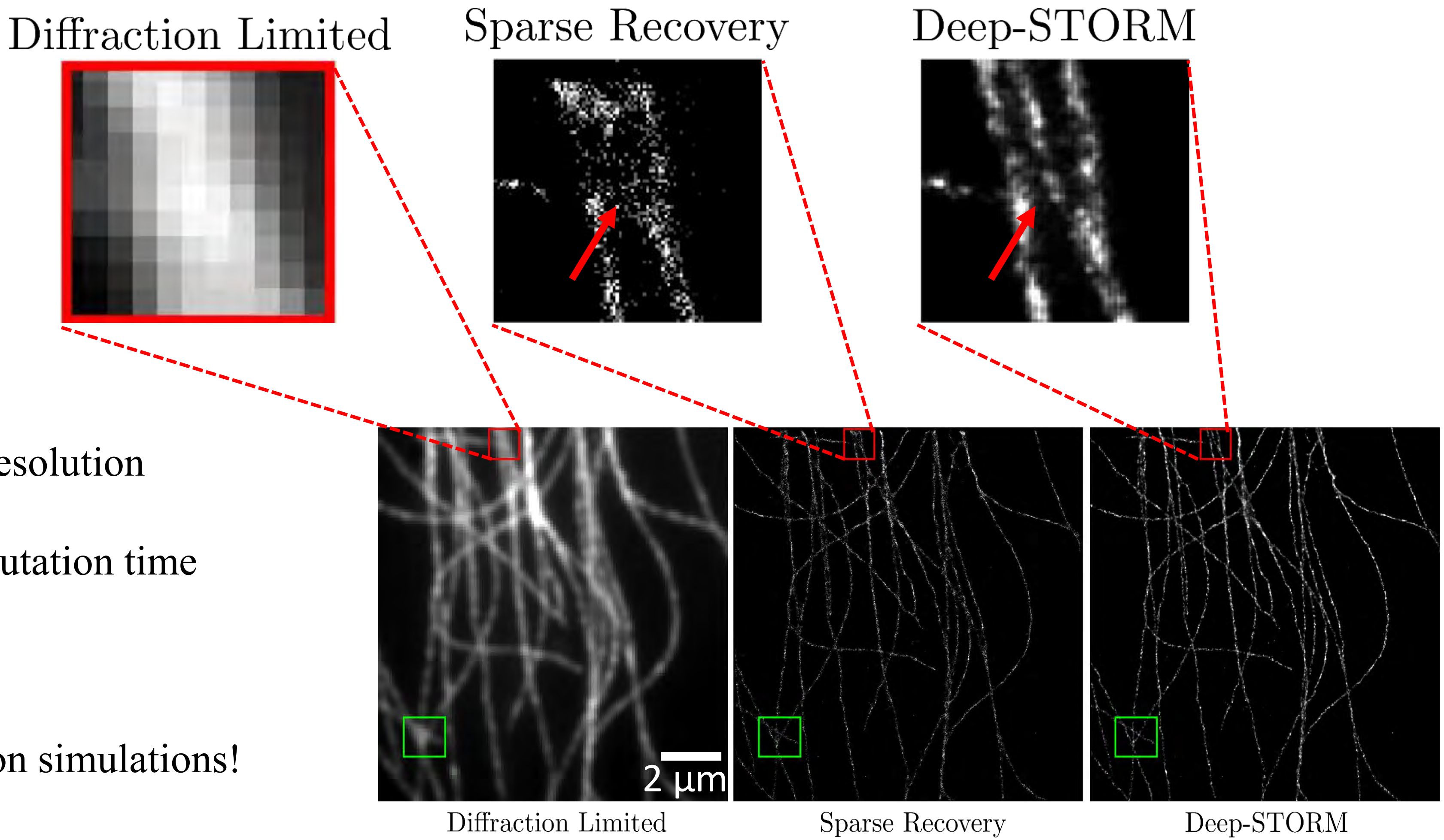
Diffraction Limited Sparse Recovery Deep-STORM

*Experimental data - ground truth is not available.

Qualitative assessment of the results



Qualitative assessment of the results



- ✓Similar or better resolution
- ✓Much faster computation time
- ✓Parameter free
- ✓Training entirely on simulations!

For more details

Research Article Vol. 5, No. 4 / April 2018 / *Optica* 458

optica

Deep-STORM: super-resolution single-molecule microscopy by deep learning

ELIAS NEHME,[1,2] LUCIEN E. WEISS,[2] TOMER MICHAELI,[1] AND YOAV SHECHTMAN[2,*]

[1]*Electrical Engineering Department, Technion, 32000 Haifa, Israel*
[2]*Biomedical Engineering Department, Technion, 32000 Haifa, Israel*
**Corresponding author: yoavsh@bm.technion.ac.il*

Received 13 February 2018; accepted 13 March 2018 (Doc. ID 323156); published 12 April 2018

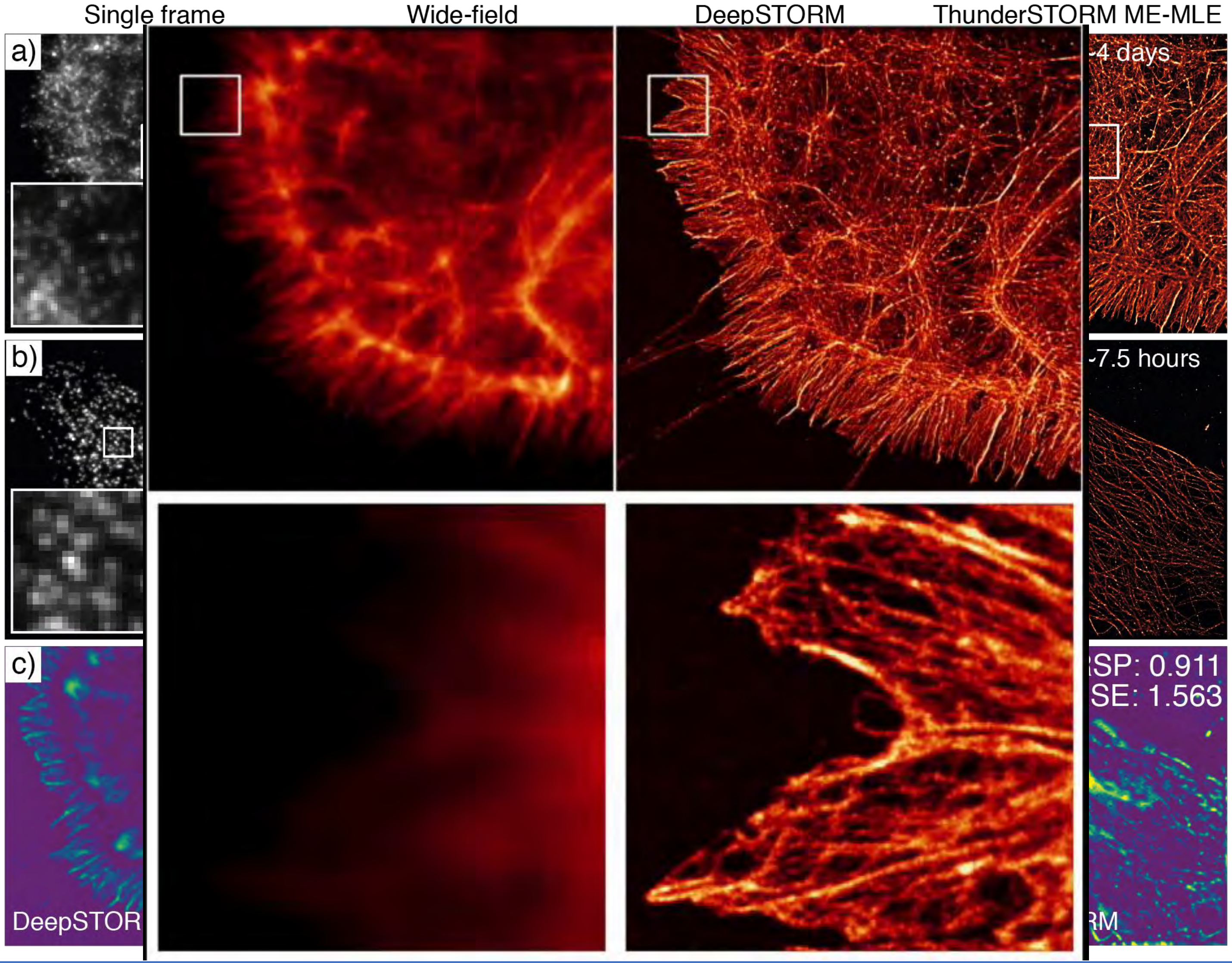
We present an ultrafast, precise, parameter-free method, which we term Deep-STORM, for obtaining super-resolution images from stochastically blinking emitters, such as fluorescent molecules used for localization microscopy. Deep-STORM uses a deep convolutional neural network that can be trained on simulated data or experimental measurements, both of which are demonstrated. The method achieves state-of-the-art resolution under challenging signal-to-noise conditions and high emitter densities and is significantly faster than existing approaches. Additionally, no prior information on the shape of the underlying structure is required, making the method applicable to any blinking dataset. We validate our approach by super-resolution image reconstruction of simulated and experimentally obtained data.

OCIS codes: (100.6640) Superresolution; (180.2520) Fluorescence microscopy; (150.1135) Algorithms; (100.0100) Image processing.

https://doi.org/10.1364/OPTICA.5.000458

Code: https://github.com/EliasNehme/Deep-STORM

Impact on biology research



References

Ober, R. J., Ram, S., & Ward, E. S. (2004). Localization Accuracy in Single-Molecule Microscopy. *Biophysical Journal*, *86*(2), 1185–1200. http://doi.org/10.1016/S0006-3495(04)74193-4

Abraham, A. V., Ram, S., Chao, J., Ward, E. S., & Ober, R. J. (2009). Quantitative study of single molecule location estimation techniques. *Optics Express*, *17*(26), 23352–22. http://doi.org/10.1364/OE.17.023352

K. Xu, H.P. Babcock & X. Zhuang (2012). Dual-objective STORM reveals three-dimensional filament organization in the actin cytoskeleton. *Nature Methods* **9**, 185–188 (2012) doi:10.1038/nmeth.1841

Small, A., & Stahlheber, S. (2014). **Fluorophore localization algorithms for super-resolution microscopy**. *Nature Methods*, *11*(3), 267–279. http://doi.org/10.1038/nmeth.2844

Deschout, H., Zanacchi, F. C., Mlodzianoski, M., Diaspro, A., Bewersdorf, J., Hess, S. T., & Braeckmans, K. (2014). **Precisely and accurately localizing single emitters in fluorescence microscopy**. *Nature Methods*, *11*(3), 253–266. http://doi.org/10.1038/nmeth.2843

Ovesny, M., Ovesný, M., K i ek, P., Křížek, P., Borkovec, J., Borkovec, J., et al. (2014). ThunderSTORM: a comprehensive ImageJ plug-in for PALM and STORM data analysis and super-resolution imaging. *Bioinformatics*, *30*(16), 2389–2390. http://doi.org/10.1093/bioinformatics/btu202

http://zeiss-campus.magnet.fsu.edu/articles/superresolution/palm/practicalaspects.html

הטכניון
מכון טכנולוגי לישראל

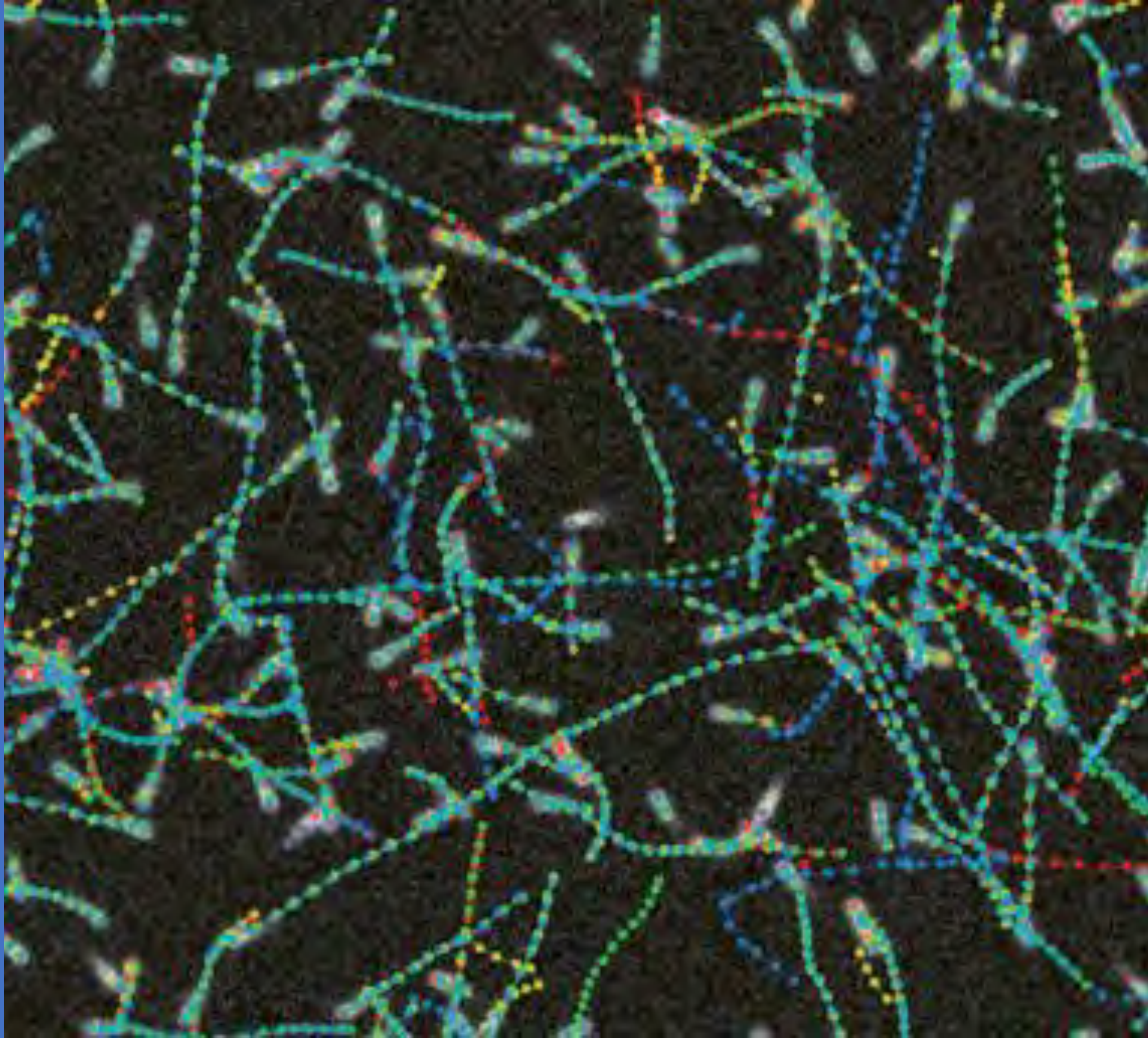
Department of Biomedical Engineering, Technion

Computational optical imaging 336547

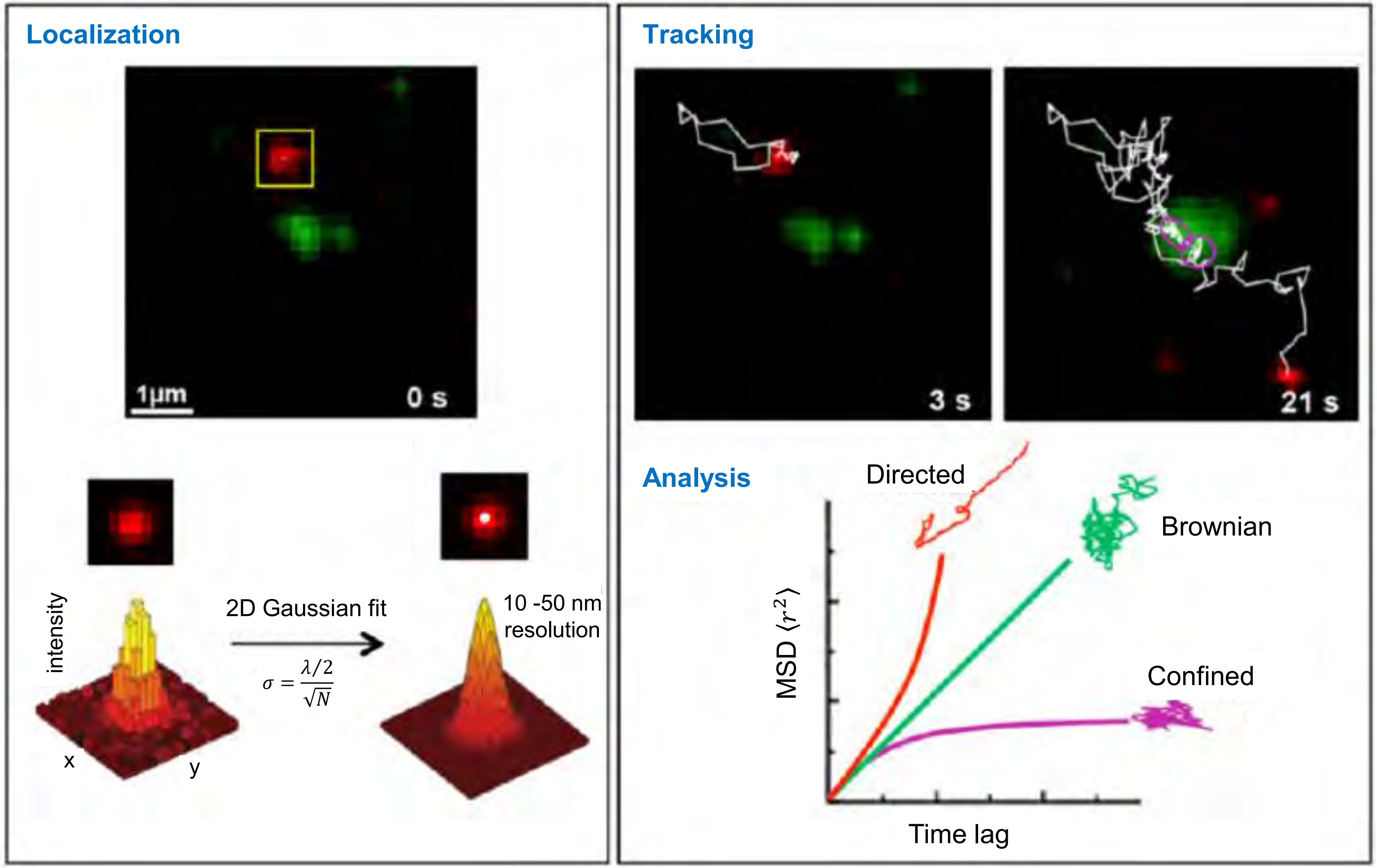
Tutorial 8 – Single Particle Tracking

Elias Nehme & Yoav Shechtman

22 December 2020



Tracking Particles



$$\langle r^2 \rangle = 4Dt \quad \text{normal diffusion}$$

$$\langle r^2 \rangle = 4Dt^{\alpha} \quad \text{anomalous diffusion}$$

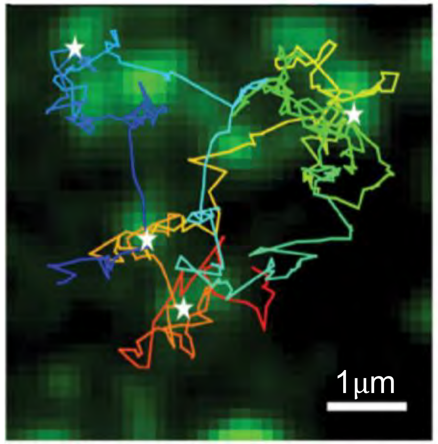
Brownian

α>1 Super-diffusion (directed)

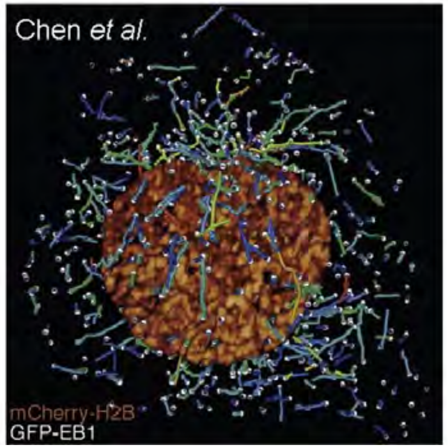
α<1 Sub-diffusion (confined)

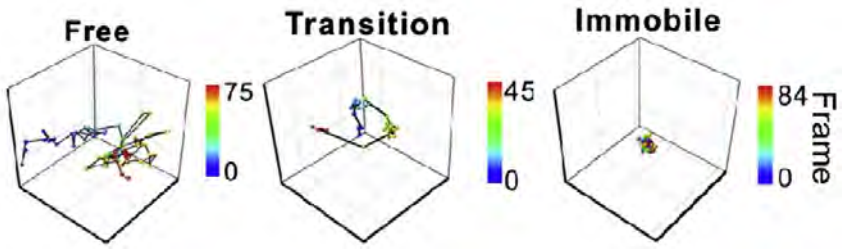
Single Particle Tracking in Biology

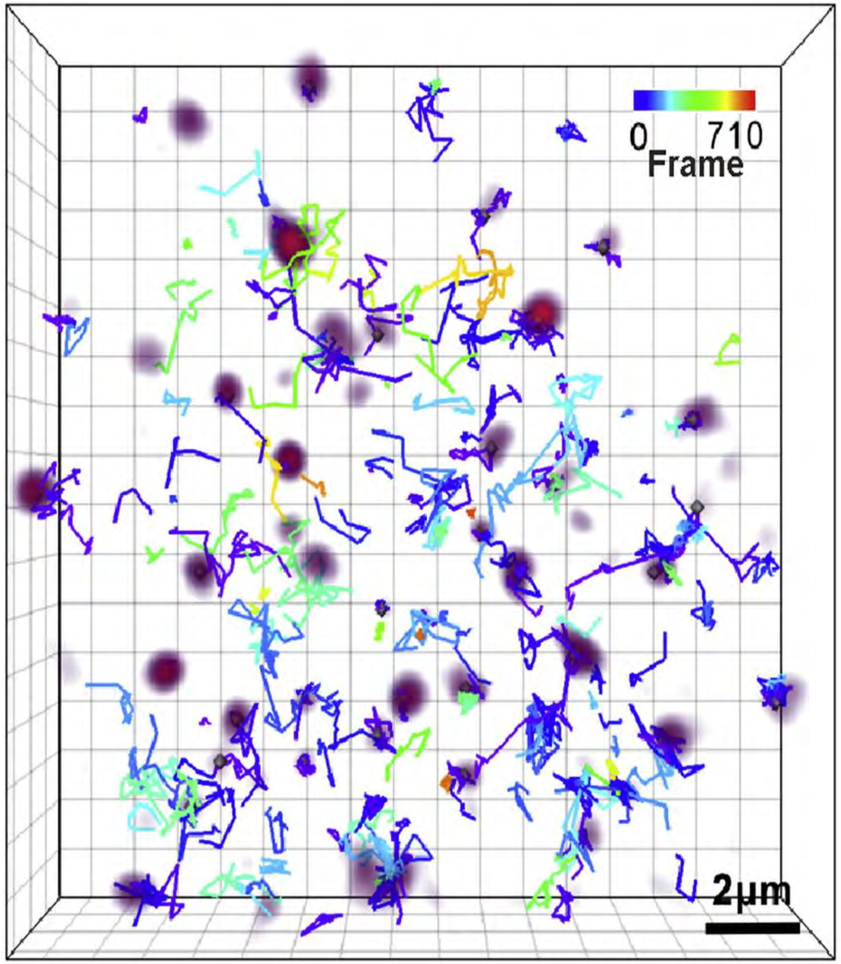
Potassium channels tracking overlaid on CCP



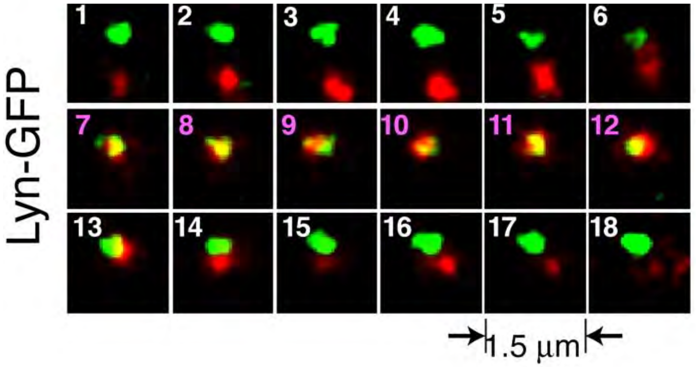
3D Tracking of EB1 Dynamics in a Live Cell

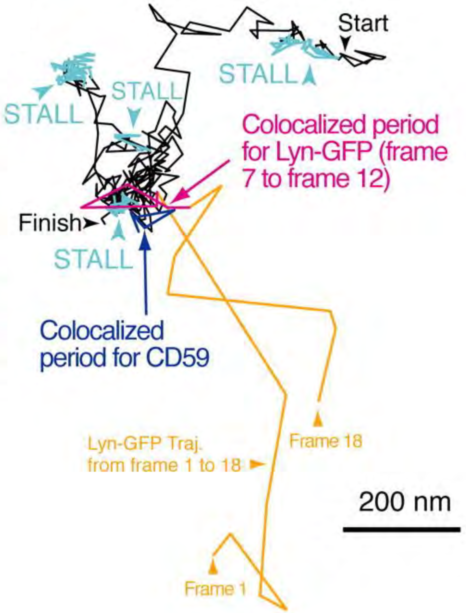






Recruitment of Lyn to CD59 clusters

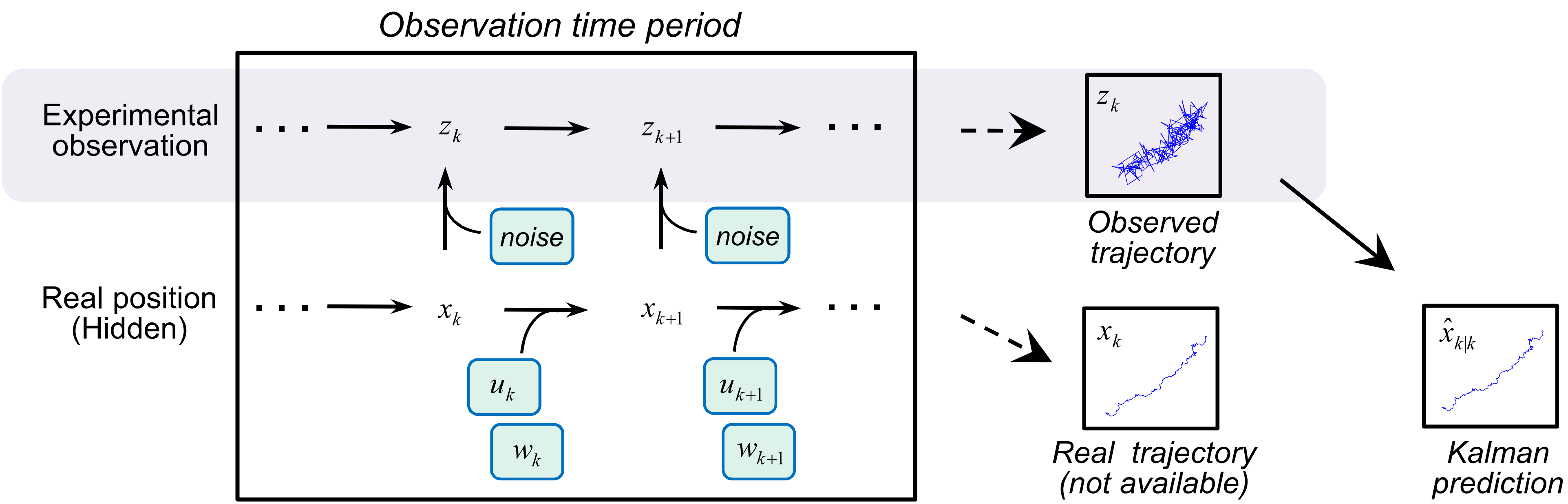




Single Particle Tracking in Biology

- In a particle tracking experiment, the **sensor noise** in the image acquisition system is transformed into a **positioning error → computed particle trajectory is a noisy version of the true particle trajectory**

- The **Kalman filter** finds the optimal state estimate for **linear** dynamic systems from sensor measurements in the presence of **Gaussian noise**



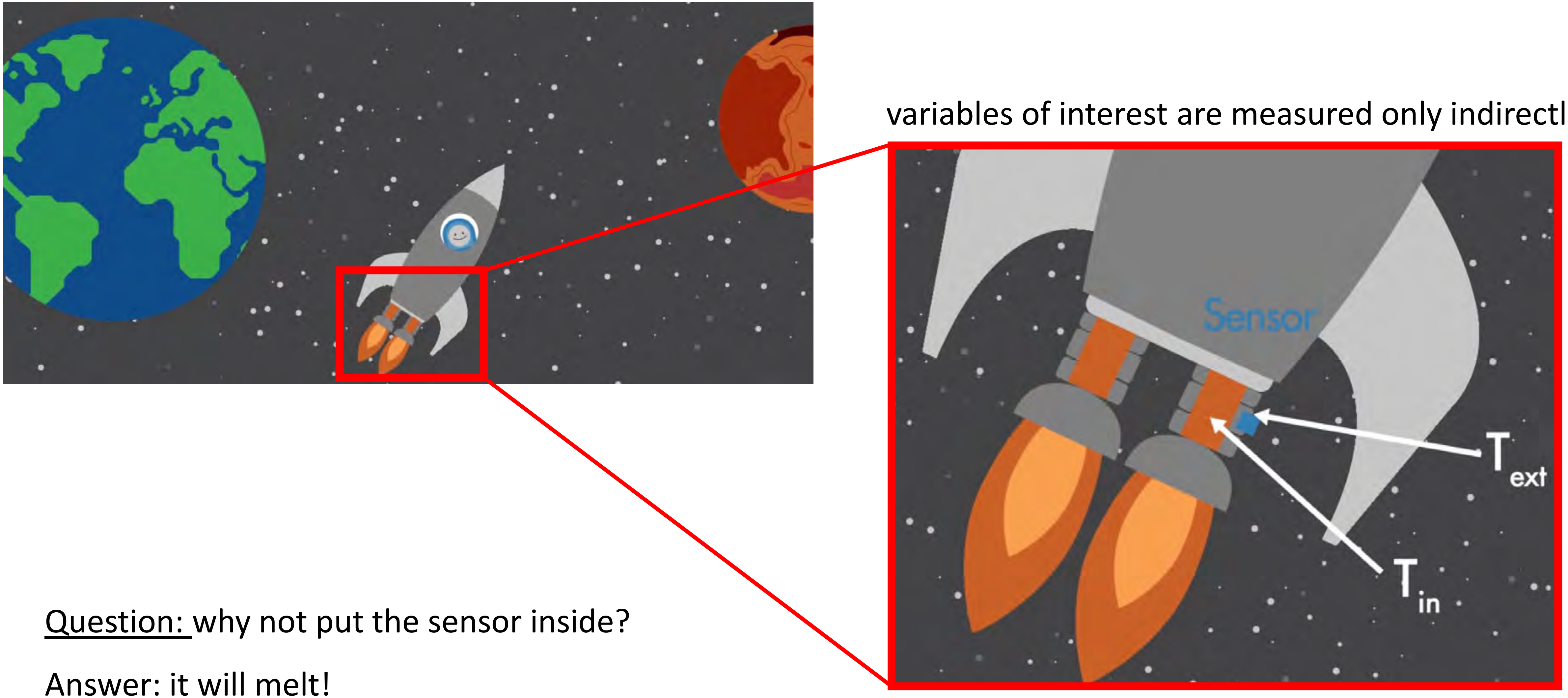
Rudolf E. Kalman

A Kalman filter is an optimal estimation algorithm used to estimate states of a system from indirect and uncertain measurements

State observers

State observers are used to **estimate the internal states of a system**:

variables of interest are measured only indirectly



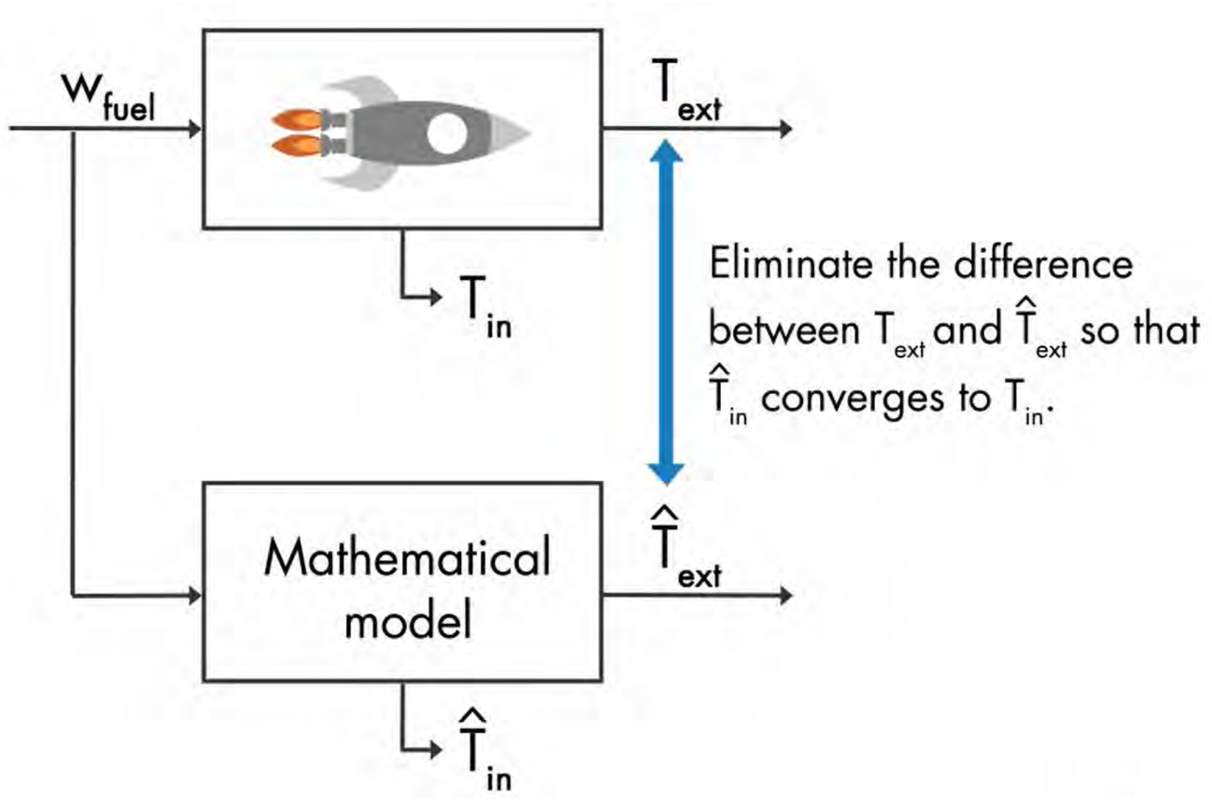
Question: why not put the sensor inside?

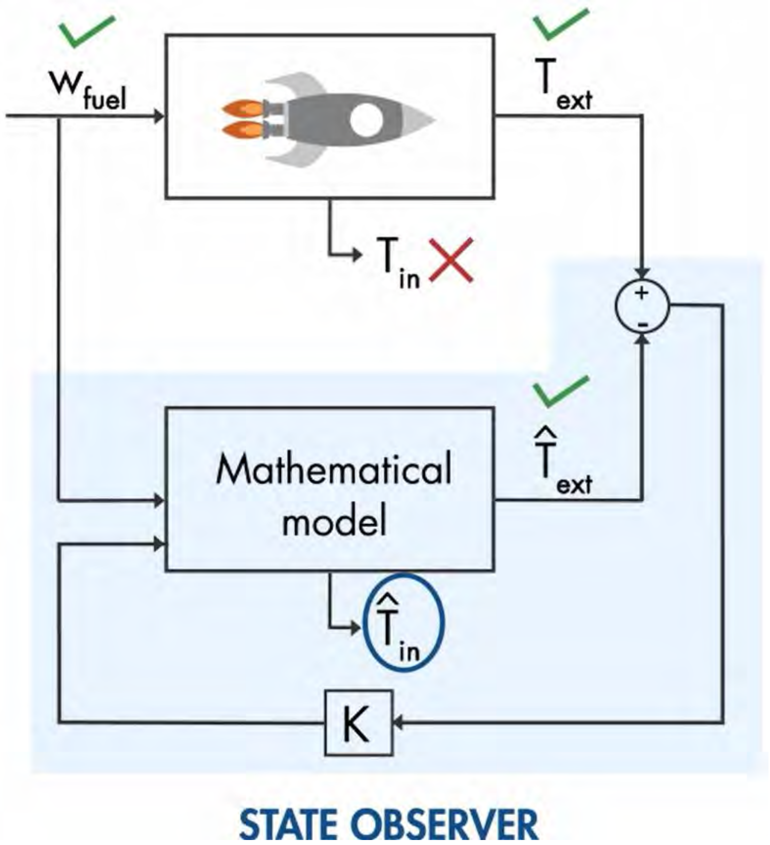
Answer: it will melt!

https://www.mathworks.com/videos/series/understanding-kalman-filters.html

State observer and Kalman filter

State observers are used to **estimate the internal states of a system**:



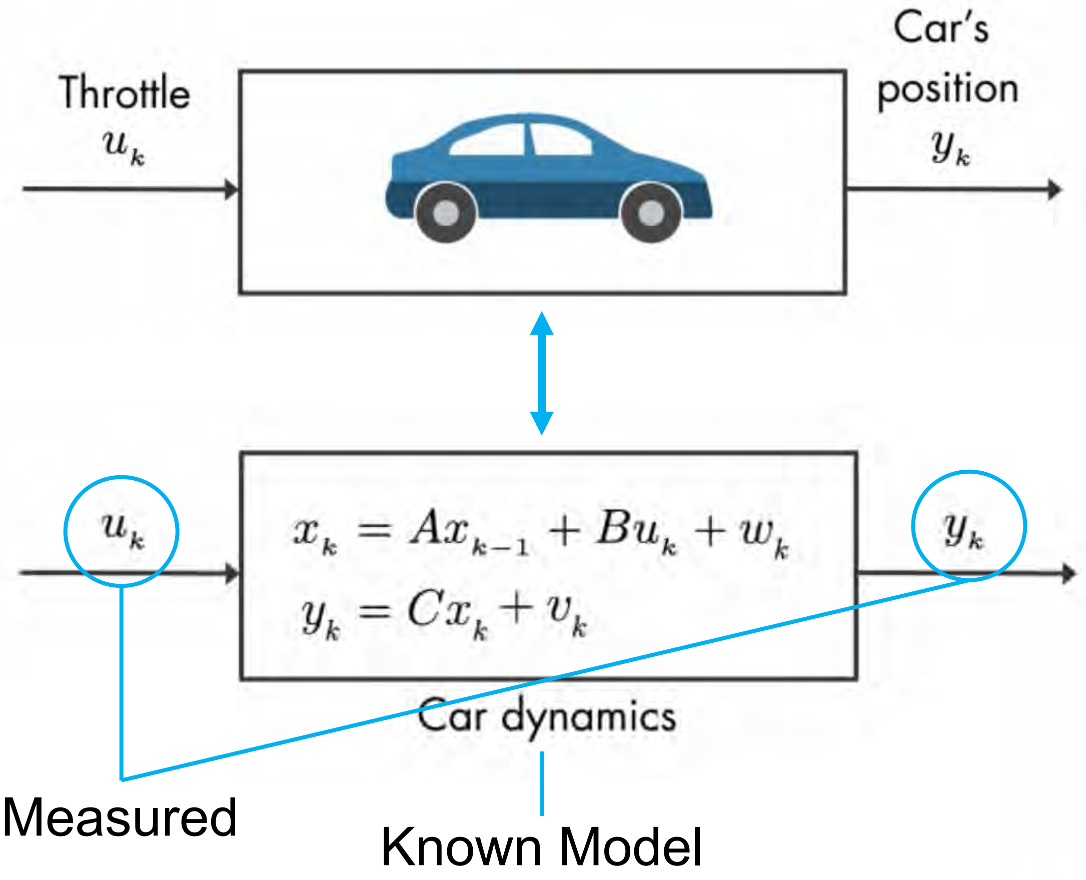


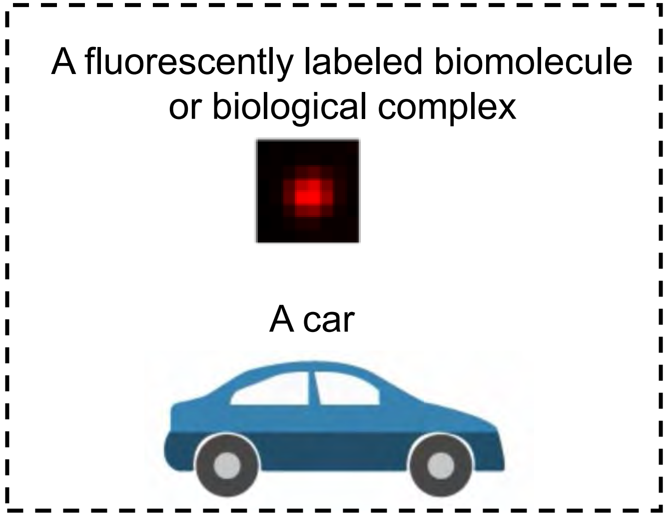
- State observer **utilizes feedback control to drive the estimated states to the true states**
- **Kalman filtering provides an optimal way of choosing the gain** of this feedback controller

Kalman filter – Tracking the Position of a Vehicle

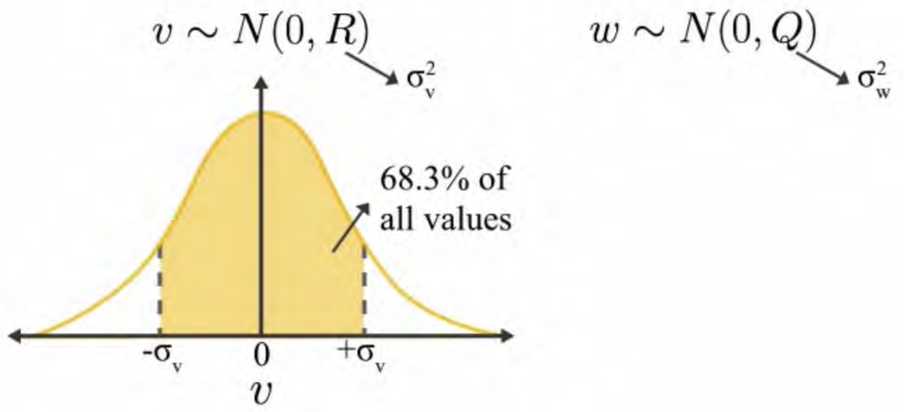
Kalman filters **combine** two sources of information: **predicted states and noisy measurements**

→ **To produce optimal and unbiased estimates of system states**



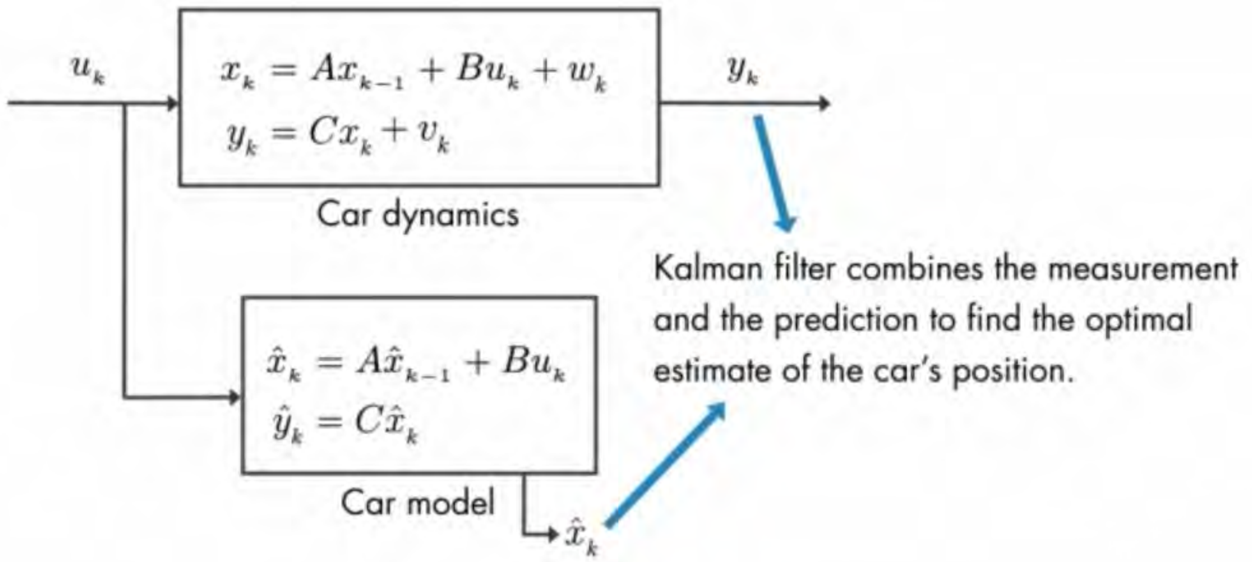


legitimate approximation

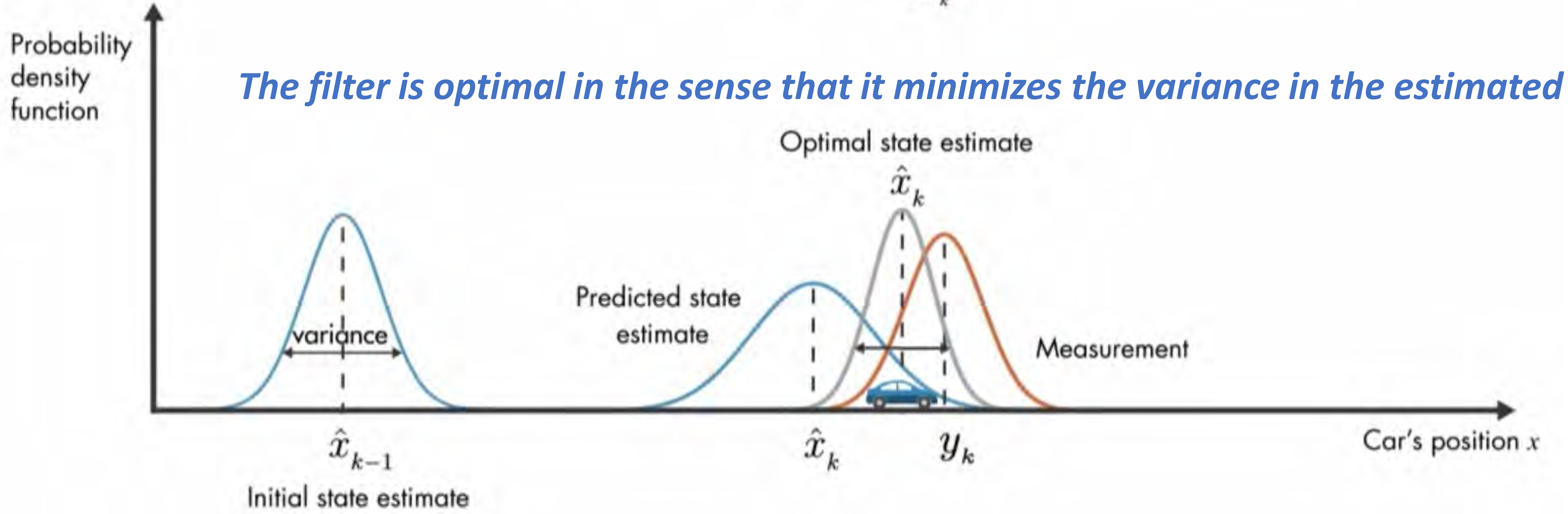


Q: How to improve $\hat{x}_k$?

Kalman filter – Tracking the Position of a Vehicle



The filter is optimal in the sense that it minimizes the variance in the estimated states



Kalman filter – Estimation

$$\boxed{\hat{x}_k = A\hat{x}_{k-1} + Bu_k + K_k(y_k - C(A\hat{x}_{k-1} + Bu_k))}$$

Stochastic system

Prediction

A Priori estimate → $\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$

A Priori covariance → $P_k^- = AP_{k-1}A^T + Q$

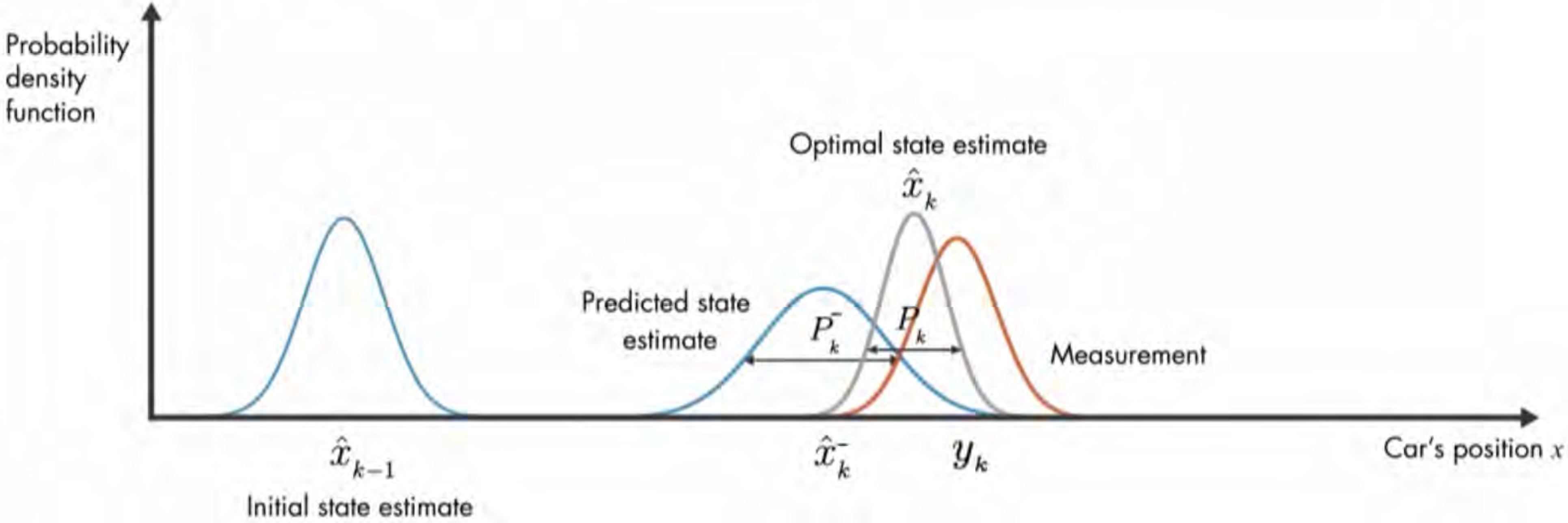
Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

Update

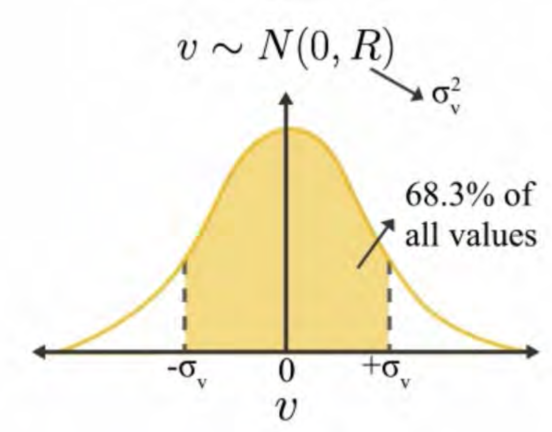
$$K_k = \frac{P_k^- C^T}{CP_k^- C^T + R}$$

$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-)$ ← A Posteriori estimate

$P_k = (I - K_k C)P_k^-$ ← A Posteriori covariance



Kalman filter – 1D Particle Tracking - MATLAB



$w \sim N(0, Q)$ → σ_w^2

$$\boxed{\begin{aligned} R_k &\sim R = \sigma_v^2 \\ Q_k &\sim Q = \sigma_w^2 = 2 \times D \times \tau \end{aligned}}$$

- One-dimensional particle **position** x_k at timestep k
- **Constant directed movement** u_0
- **Process noise**, w_k : thermal fluctuations at timestep k
- **Measurement noise**, v_k: zero-mean white noise at timestep k

$$\underline{x}_k = A\underline{x}_{k-1} + w_k$$
$$\bar{y}_k = C\underline{x}_k + v_k$$

↓

$$\boxed{\begin{aligned} x_k &= x_{k-1} + u_0 \Delta t + w_k \\ y_k &= x_k + v_k \end{aligned}}$$

$\underline{x}_k = \begin{pmatrix} x_k \\ u_0 \end{pmatrix}$ state estimate

$A = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$ state transition model

$C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ observation model

Assuming u_0 is constant and noise free

$Q_{MAT} = \begin{pmatrix} \sigma_w^2 & 0 \\ 0 & \mathbf{0} \end{pmatrix}$ process noise covariance matrix

$R_{MAT} = \begin{pmatrix} \sigma_v^2 & 0 \\ 0 & \mathbf{0} \end{pmatrix} + \varepsilon$ measurement noise covariance matrix

Correcting division by zero

Estimation of σ_w^2

$S = \boldsymbol{Q} + 2 \times R \longrightarrow \boldsymbol{Q} = S - 2 \times R$

- S: variance of the measured displacement ≡ `var(dy)`
- R: a priori knowledge on the measurement noise

Kalman filter – 1D Particle Tracking - MATLAB

$$\underline{x}_k = \begin{pmatrix} x_k \\ u_0 \end{pmatrix}$$ state estimate

$$A = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$$ state transition model

$$C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$ observation model

$$Q_{MAT} = \begin{pmatrix} \sigma_w^2 & 0 \\ 0 & \mathbf{0} \end{pmatrix}$$ process noise covariance matrix

$$R_{MAT} = \begin{pmatrix} \sigma_v^2 & 0 \\ 0 & \mathbf{0} \end{pmatrix} + \varepsilon$$ measurement noise covariance matrix

for k=1:N

$$\underline{x}_{k-1} = \begin{pmatrix} x_{k-1} \\ u_0 \end{pmatrix}$$

$$\underline{\hat{x}}_{k-1} = \begin{pmatrix} \hat{x}_{k-1} \\ u_0 \end{pmatrix}$$

Prediction

$$\hat{x}_k^- = A\hat{x}_{k-1} + \cancel{Bu_k}$$

state estimate

$$\cancel{P_k^- = AP_{k-1}A^T + Q}$$

Initial estimates for $\hat{x}_{k-1}$ $\cancel{\text{and } P_{k-1}}$

Update

$$\cancel{K_k = \frac{P_k^- C^T}{CP_k^- C^T + R}}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-)$$

$$\cancel{P_k = (I - K_k C)P_k^-}$$

$$K_k = K = \frac{(Q/R) + \sqrt{(Q/R)^2 + 4(Q/R)}}{2 + (Q/R) + \sqrt{(Q/R)^2 + 4(Q/R)}}$$

Optimal value of K under assumptions of **linear dynamics with Gaussian process and measurement noise**

Simulating realistic particle tracks - MATLAB

While $M < \#$ of particles

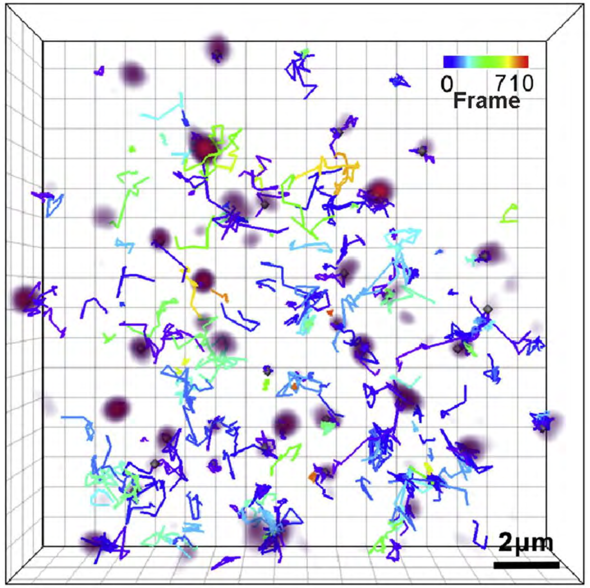
1. Assign an initial number N of sub-particles (drawn from a uniform random distribution $[1,4]$)

2. Initialize the 3D particle position $(z_0 = 0)$ randomly within the field of view (FOV)

3. Construct $\mathrm{x}, \mathrm{y}, z$ −trajectory (75% probability of undergoing xy linear motion). The z trajectory always consists of Brownian motion.

4. Assign the number $N_S$of splitting events (drawn from a uniform random distribution $[1, N]$) and the corresponding time points t_S at which these occur (drawn from a uniform random distribution).

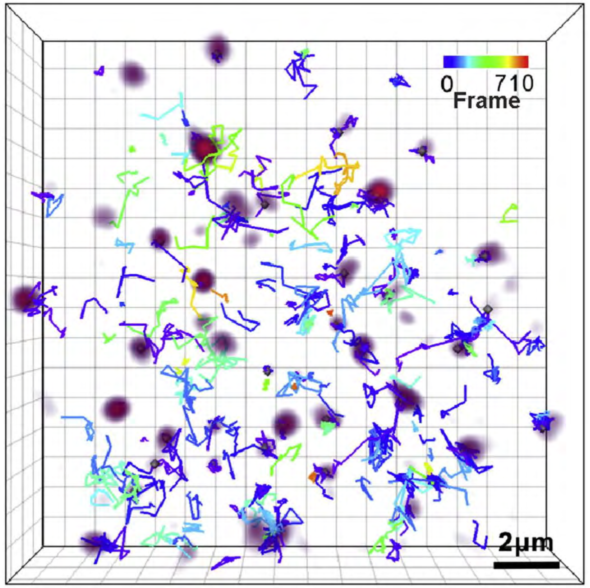
5. Create an additional trajectory (linear motion at a 2D random orientation) which starts at the splitting time $t = t_S(1)$, ends at $\mathrm{t} = T$ and whose initial point correspond to the position of p_i at $t_S(1)$. This trajectory now corresponds to the one of a new particle p_{i+1}. Update the number of particles left in p_i.



Guidelines

- 50 particle tracks
- Brownian motion
- 75% of particles undergoing linear motion
- particles undergoing splitting and merging events (50/50)

Simulating realistic particle tracks - MATLAB



6. Repeat 5. $N_S - 1$ times going through t_S. At this point you should obtain N_S trajectories, all ending at t $= T$ and starting at t ∈ $[0, t_S(1), \ldots, t_S(N_S - 1)]$

7. Split the trajectory of the initial particle p_i into N_S segments/particles $p_{i+N-1+k}$ according to t_S $(k = 1{:}\,N_S)$. Update the number of particles in each $p_{i+N-1+k}$. E.g. an initial particle splitting two times produces five particles.

8. With a probability of 50%, flip the trajectories (x,y,z,t,# of sub-particles) to convert splitting into merging events (fliplr).

9. Update the total number of particles M

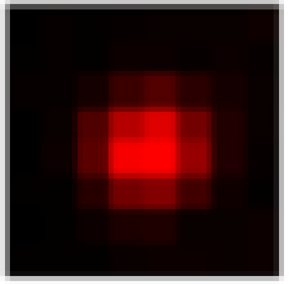
end

Guidelines
- 50 particle tracks
- Brownian motion
- 75% of particles undergoing linear motion
- particles undergoing splitting and merging events (50/50)

The purpose of this simulated tracks is to assess the performance of any particle tracking algorithm for a particular application

Simulating realistic particle tracks – MATLAB – Optical image

- **Acquisition Model & Pixelation** ∀particles & time points



$$\mathrm{PSF}_G(x, y|\boldsymbol{\theta}) = \theta_N E_x E_y E_z + \theta_b$$

`meshgrid, erf`

Alternative: 3D Gaussian centered on $[\theta_x, \theta_y, \theta_z]$

$$E_x = \frac{1}{2}\mathrm{erf}\left(\frac{x - \theta_x + \frac{1}{2}}{\sqrt{2}\theta_\sigma}\right) - \frac{1}{2}\mathrm{erf}\left(\frac{x - \theta_x - \frac{1}{2}}{\sqrt{2}\theta_\sigma}\right)$$

$$E_y = \frac{1}{2}\mathrm{erf}\left(\frac{y - \theta_y + \frac{1}{2}}{\sqrt{2}\theta_\sigma}\right) - \frac{1}{2}\mathrm{erf}\left(\frac{y - \theta_y - \frac{1}{2}}{\sqrt{2}\theta_\sigma}\right)$$

$$E_z = \frac{1}{2}\mathrm{erf}\left(\frac{z - \theta_z + \frac{1}{2}}{2\sqrt{2}\theta_\sigma}\right) - \frac{1}{2}\mathrm{erf}\left(\frac{z - \theta_z - \frac{1}{2}}{2\sqrt{2}\theta_\sigma}\right)$$

- **Quantum Efficiency & Poisson Noise** ∀frames

  `poissrnd`

- **Readout Noise & Dark Current** ∀frames

  `normrnd`

- **Discretization** ∀pixels

$$g = \left[2^n \left({}^{g}/_{fw}\right)\right]$$

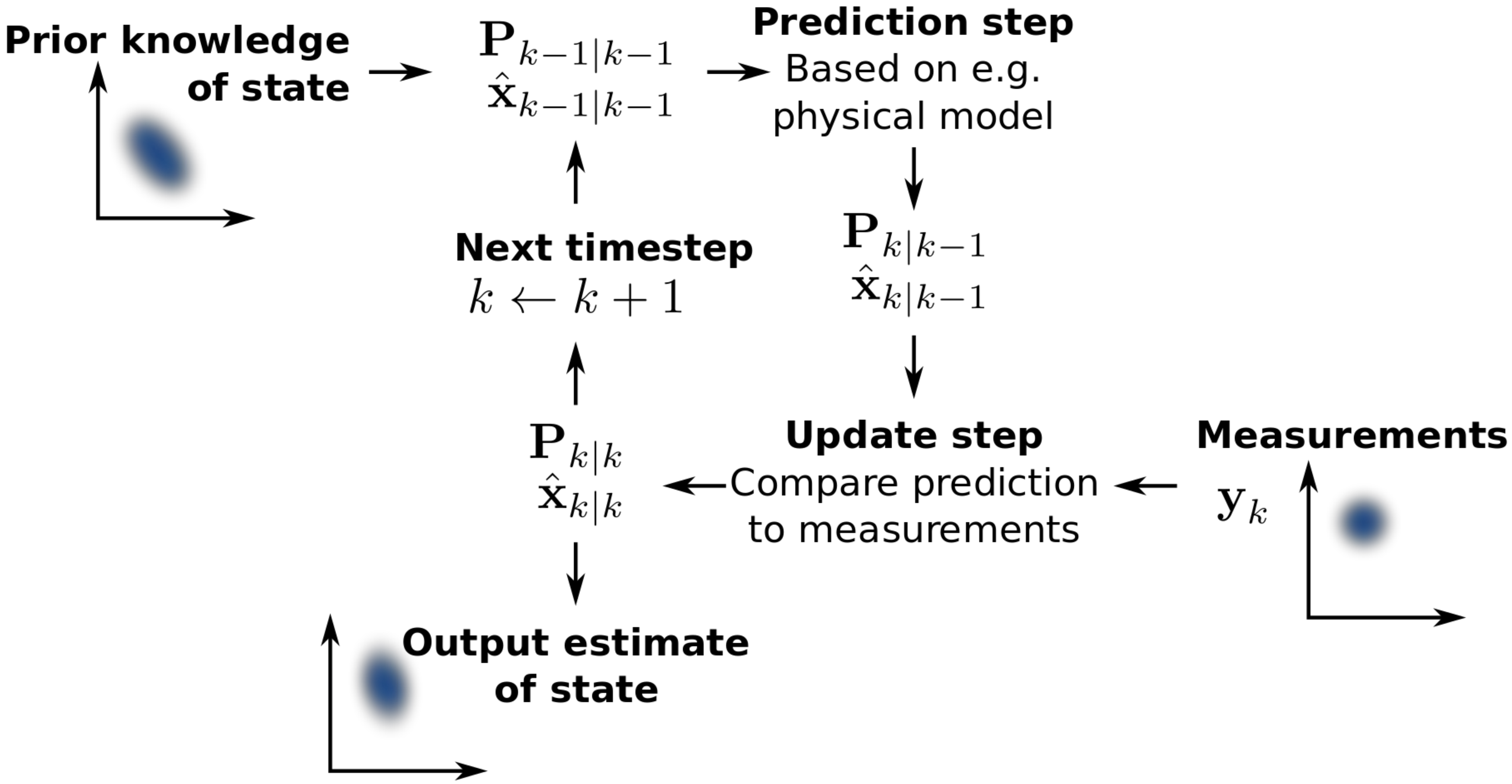
`uint16`

n: number of bits
g: image stack of particles
fw: full well capacity

$$\boldsymbol{\theta} = [\theta_x, \theta_y, \theta_\sigma, \theta_N, \theta_b]$$

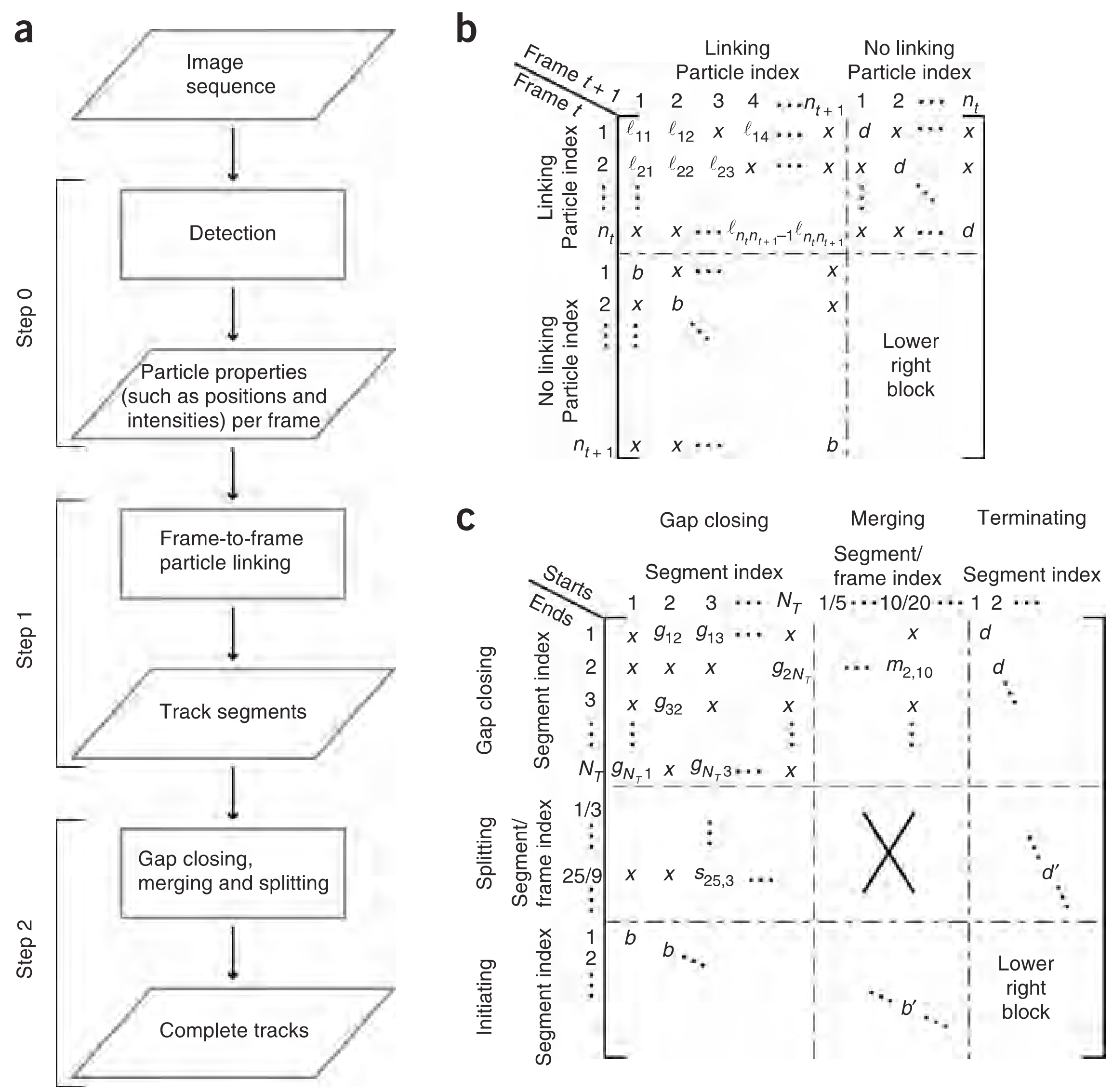
θ_x sub-pixel molecular x-coordinate
θ_y sub-pixel molecular y-coordinate
θ_z sub-pixel molecular z-coordinate
θ_σ imaged size of the molecule
θ_N total number of photons emitted by the molecule
θ_b background offset

Illustration of Kalman filtering for position estimation in 2D



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

Single Particle Tracking (SPT) using μ-track



$$\hat{A}_{\arg\min} = \sum_{i=1}^{\text{Number of rows}} \sum_{j=1}^{\text{Number of columns}} A_{ij} C_{ij}$$

`scriptTrackGeneral`:
determines the final tracks based on a target motion model

`scriptDetectGeneral`:
detection of diffraction-limited objects such as single molecules and small molecular aggregates

`plotTracks2D`:
statically plots the tracks generated by scriptTrackGeneral

https://downloads.openmicroscopy.org/u-track/2.1.1/artifacts/u-track-2.1.1.pdf

A Java package for running ImageJ and Fiji within Matlab

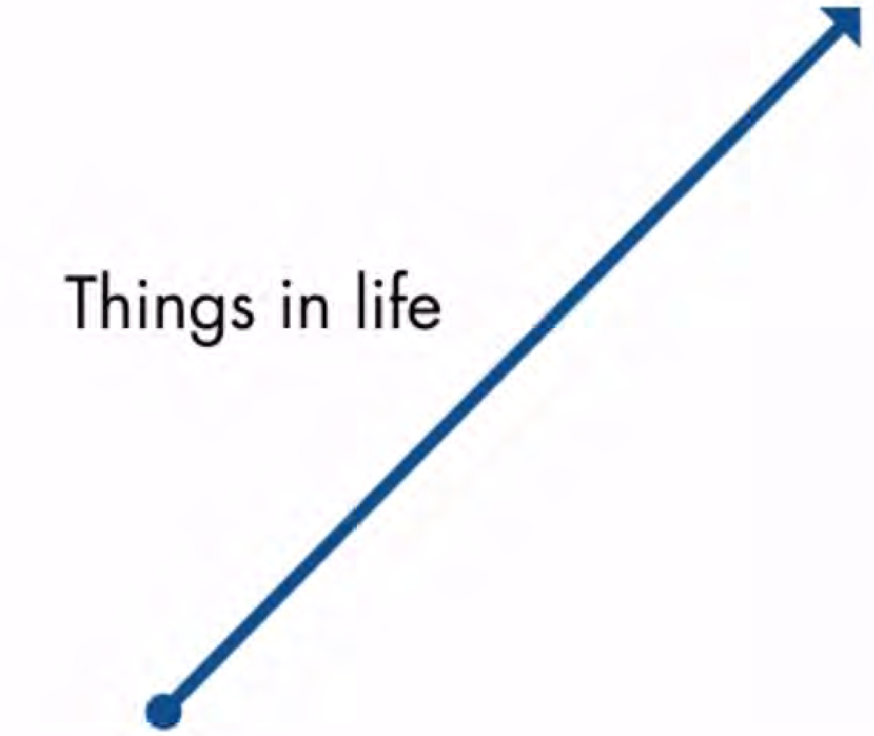
http://bigwww.epfl.ch/sage/soft/mij/

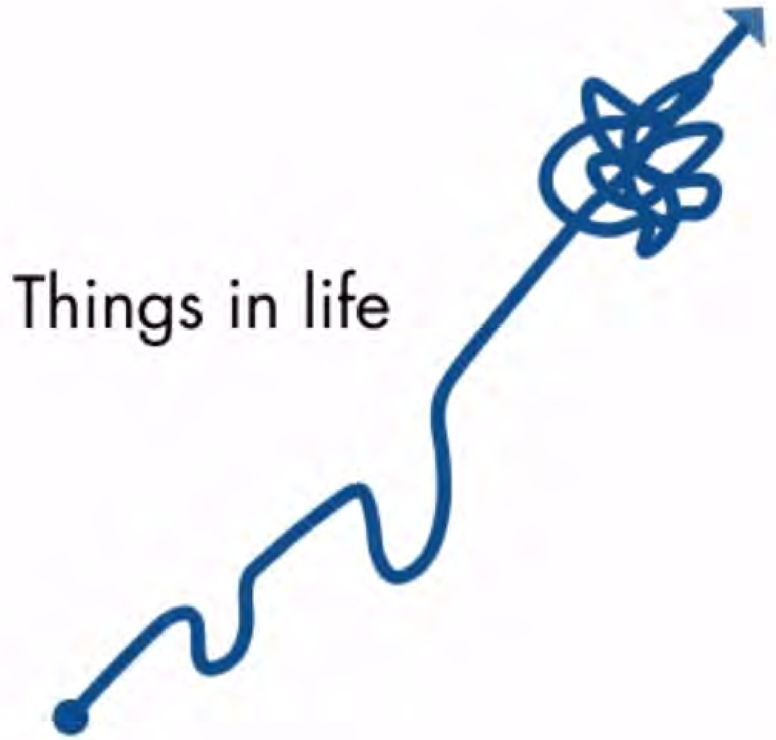
SPT methods

		Detection			Linking				
Method	Authors	Prefilter	Approaches	Remarks	Principle	Approaches	Remarks	Dim.	Refs.
1	I.F. Sbalzarini Y. Gong J. Cardinale	–	M, C	Iterative intensity-weighted centroid calculation	Combinatorial optimization	MF, MT, GC	Greedy hill-climbing optimization with topological constraints	2D & 3D	32
2	C. Carthel S. Coraluppi	Disk	M, T	Adaptive local-maxima selection	Multiple hypothesis tracking	MF, MT, MM	Motion models are user specified (near-constant position and/or velocity)	2D & 3D	33,34
3	N. Chenouard F. de Chaumont J.-C. Olivo-Marin	Wavelets	M, T	Maxima after thresholding two-scale wavelet products	Multiple hypothesis tracking	MF, MT, MM, GC	Motion models are user specified (near-constant position and/or velocity)	2D & 3D	35–37
4	M. Winter A.R. Cohen	Gaussian, median and morphology	M, T, C	Adaptive Otsu thresholding	Multitemporal association tracking	MF, MT, GC	Post-tracking refinement of detections	2D & 3D	38,39
5	W.J. Godinez K. Rohr	Laplacian of Gaussian or Gaussian fitting	M, T, F, C	Either thresholding + centroid or maxima + Gaussian fitting	Kalman filtering + probabilistic data association	MF, MM	Interacting multiple models using motion models as specified	2D & 3D	29,40
6	Y. Kalaidzidis	Windowed floating mean background subtraction	T, F	Lorentzian function fitting to structures above noise level	Dynamic programming	MF, GC	Track assignment by the weighted sum of multiple features	2D	41
7	L. Liang J. Duncan H. Shen Y. Xu	Laplacian of Gaussian	M, T, F	Gaussian mixture model fitting	Multiple hypothesis tracking	MF, MM	Interacting multiple models with forward and backward linking	2D	42
8	K.E.G. Magnusson J. Jaldén H.M. Blau	Deconvolution	M, T, F	Watershed-based clump splitting and parabola fitting	Viterbi algorithm on state-space representation	MF, MT	Brownian motion is assumed in all cases	2D & 3D	43,44
9	P. Paul-Gilloteaux	Laplacian of Gaussian or Gaussian filtering	M, T, F	Either maxima with pixel precision (2D) or thresholding + Gaussian fitting (3D)	Nearest neighbor + global optimization	MF, MT, GC	Global optimization of associations using simulated annealing	2D & 3D	45,46
10	P. Roudot C. Kervrann F. Waharte	Structure tensor	T, F	Histogram-based thresholding and Gaussian fitting	Gaussian template matching	–	Only local and per-trajectory particle linking	2D	47–49
11	I. Smal E. Meijering	Wavelets	M, F, C	Gaussian fitting (round particles) or centroid calculation (elongated particles)	Sequential multiframe assignment	MF, MT, MM, GC	Global linking cost minimization	2D	35,50,51
12	J.-Y. Tinevez S.L. Shorte	Difference of Gaussian	M, T, F	Parabolic fitting to localized maxima	Linear assignment problem	MT, GC	Two-step approach (frame-to-frame and segment linking)	2D & 3D	52,53
13	J. Willemse K. Celler G.P. van Wezel	Gaussian and top hat	T, C	Watershed-based clump splitting	Nearest neighbor	MM, GC	Allows merging and splitting of particles and uses a linear motion model	2D & 3D	54,55
14	H.-W. Dan Y.-S. Tsai	Gaussian, Wiener and top hat	T, C	Morphological opening–based clump splitting	Nearest neighbor + Kalman filtering	MM	Essentially a 2D method keeping track of maximum intensity in z	2D & 3D	56,57

See **Supplementary Note 1** for further details on methods 1–14. Dim, dimensionality. Detection approaches: M, maxima detection; T, thresholding; F, fitting; C, centroid estimation. Linking approaches: MF, multiframe; MT, multitrack; MM, motion models; GC, gap closing.

Nonlinear Systems: Extended KF, Unscented KF, and Particle Filter





$$x_k = Ax_{k-1} + Bu_k + w_k$$
$$y_k = Cx_k + Du_k + v_k$$

$$x_k = f(x_{k-1}, u_k, w_k)$$
$$y_k = g(x_k, u_k, v_k)$$

Nonlinear Systems: Extended KF, Unscented KF, and Particle Filter

State Estimator	Model	Assumed distribution	Computational cost
Kalman filter (KF)	Linear	Gaussian	Low
Extended Kalman filter (EKF)	Locally linear	Gaussian	Low (if the Jacobians need to be computed analytically) Medium (if the Jacobians can be computed numerically)
Unscented Kalman filter (UKF)	Nonlinear	Gaussian	Medium
Particle filter (PF)	Nonlinear	Non-Gaussian	High

References

Liu, Z., Lavis, L. D., & Betzig, E. (2015). Imaging Live-Cell Dynamics and Structure at the Single-Molecule Level. *Molecular Cell*, *58*(4), 644–659. http://doi.org/10.1016/j.molcel.2015.02.033

Chenouard, N., Smal, I., de Chaumont, F., Maška, M., Sbalzarini, I. F., Gong, Y., et al. (2014). Objective comparison of particle tracking methods. *Nature Methods*, *11*(3), 281–289. http://doi.org/10.1038/nmeth.2808

Suzuki, K.G.N, Fujiwara, T.K., Sanematsu, F., Iino, R., Edidin, M. & Kusumi, A. (2007). GPI-anchored receptor clusters transiently recruit Lyn and Gα for temporary cluster immobilization and Lyn activation: single-molecule tracking study 1. *Journal of Cell Biology*, *177*(4), 717–730. http://doi.org/10.1083/jcb.200609174

Wieser, S., Moertelmaier, M., Fuertbauer, E., Stockinger, H., & Schutz, G.J. (2007). (Un)Confined Diffusion of CD59 in the Plasma Membrane Determined by High-Resolution Single Molecule Microscopy. *Biophysical Journal*, *92* (10), 3719–3728. https://doi.org/10.1529/biophysj.106.095398

Torreno-Pina, A.T., Manzo, C., & Garcia-Parajo, M.F. (2016). Uncovering homo-and hetero-interactions on the cell membrane using single particle tracking approaches. *Journal of Physics D: Applied Physics*, *49*(10). https://doi.org/10.1088/0022-3727/49/10/104002

Cheezum, M. K., Walker, W. F., & Guilford, W. H. (2001). Quantitative Comparison of Algorithms for Tracking Single Fluorescent Particles. *Biophysical Journal*, *81*(4), 2378–2388. http://doi.org/10.1016/S0006-3495(01)75884-5

Saxton, M. J., & Jacobson, K. (1997). Single-Particle Tracking: Applications to Membrane Dynamics. *Annual Review of Biophysics and Biomolecular Structure*, *26*(1), 373–399. http://doi.org/10.1146/annurev.biophys.26.1.373

Wu, P.-H., Agarwal, A., Hess, H., Khargonekar, P. P., & Tseng, Y. (2010). Analysis of Video-Based Microscopic Particle Trajectories Using Kalman Filtering. *Biophysical Journal*, *98*(12), 2822–2830. http://doi.org/10.1016/j.bpj.2010.03.020

Jaqaman, K., Loerke, D., Mettlen, M., Kuwata, H., Grinstein, S., Schmid, S. L., & Danuser, G. (2008). Robust single-particle tracking in live-cell time-lapse sequences. *Nature Methods*, *5*(8), 695–702. http://doi.org/10.1038/nmeth.1237

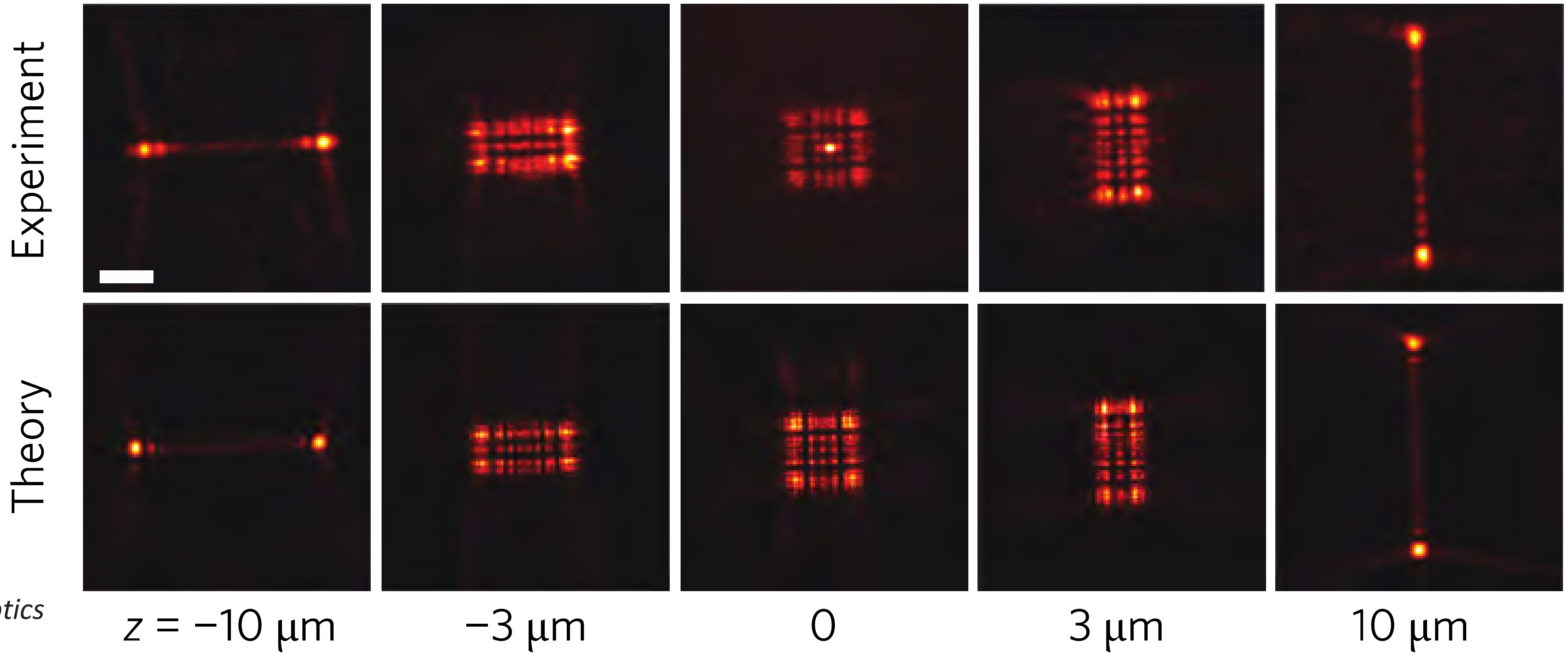
הטכניון
מכון טכנולוגי לישראל

Department of Biomedical Engineering, Technion
Computational optical imaging 336547

Tutorial 9 – Phase retrieval

Elias Nehme & Yoav Shechtman

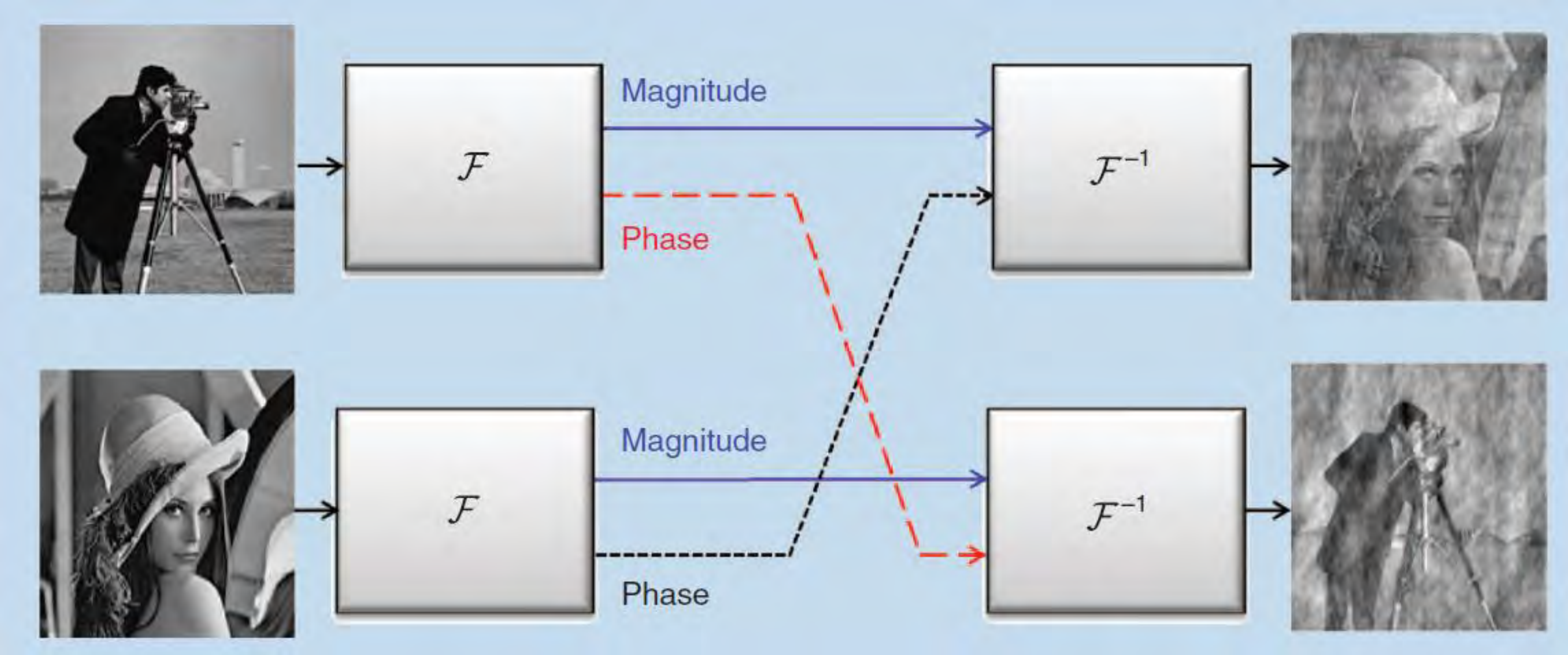
29 December 2020



N. Petrov, P. et al. *Optics express* (2017)

The significance of knowing the Fourier phase

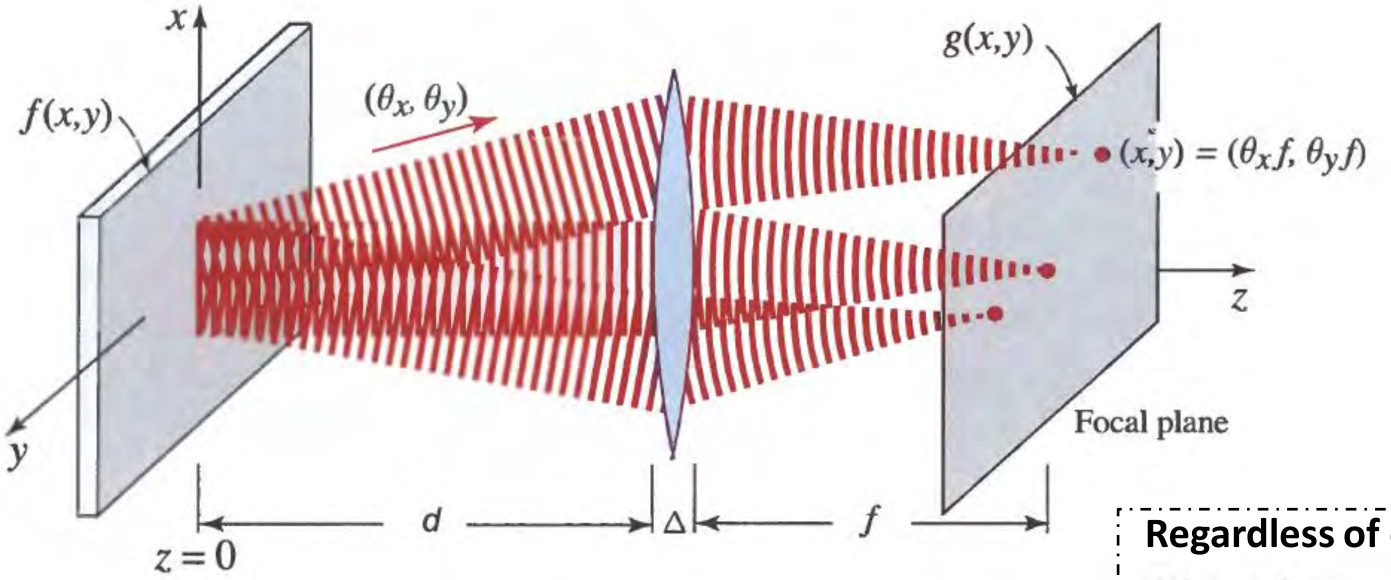
Fourier phase contains more information than the Fourier magnitude:



Shechtman, Y., et al. IEEE signal processing magazine (2015)

Phase retrieval problem and reminder

The recovery of a function **given the magnitude of its Fourier transform**:



Regardless of d:

$$I(x,y) = \frac{1}{(\lambda f)^2}\left|F\left(\frac{x}{\lambda f}, \frac{y}{\lambda f}\right)\right|^2$$

Assuming paraxial waves and using Fresnel approximation:

$$g(x,y) = h_l \underbrace{\exp\left[j\pi\frac{(x^2+y^2)(d-f)}{\lambda f^2}\right]}_{\text{Phase factor quadratic function}} F\left(\frac{x}{\lambda f}, \frac{y}{\lambda f}\right) \xrightarrow{d = f}$$

$$h_l = H_0 h_0 = (j/\lambda f)\exp[-jk(d+f)]$$

2f system

$$\boxed{g(x,y) = h_l F\left(\frac{x}{\lambda f}, \frac{y}{\lambda f}\right)}$$

$$h_l = (j/\lambda f)\exp(-j2kf)$$

Phase retrieval problem

Phase retrieval in optics:
The electromagnetic field oscillates at rates of ~$\mathbf{10^{15}}$ **Hz**

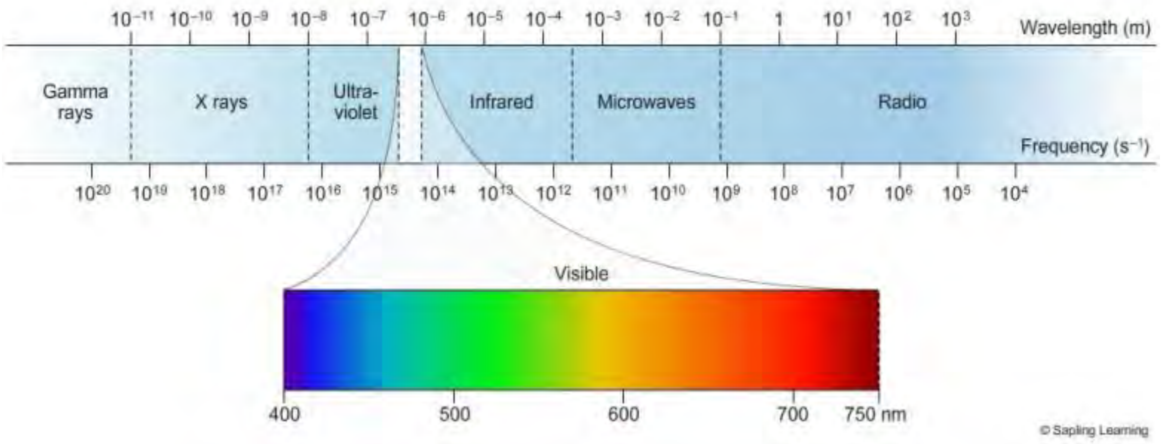
→ No electronic measurement device can follow

Measuring the phase of optical waves involves **additional complexity**, typically by requiring interference with another known field:

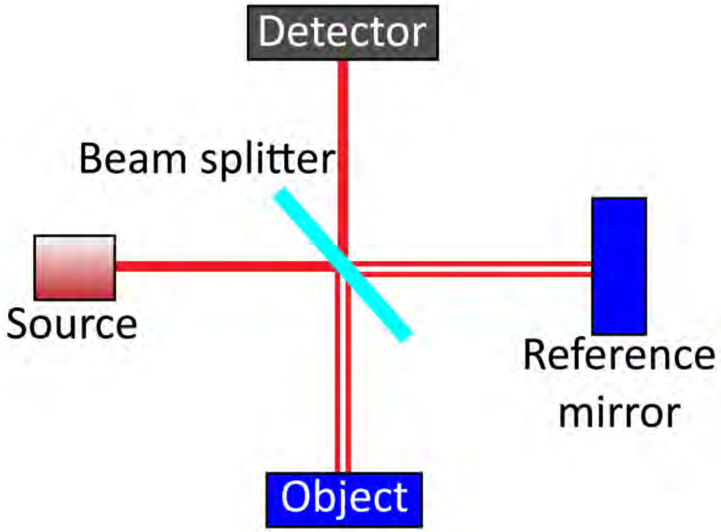
Phase is **measured using the interference pattern** of a beam which is split through two paths: a reference mirror and the sample

Impractical to implement for an existing microscope

The alternative is to recover the phase of the pupil function based on **measurements of the intensity PSF** and a **phase retrieval algorithm**

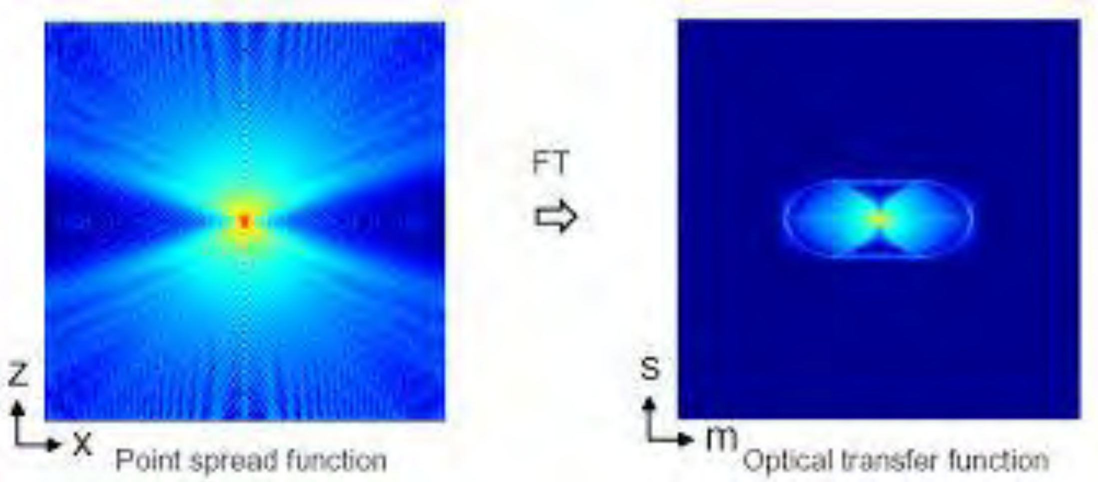


Light speed in a medium $\boxed{c = \frac{c_0}{n}}$ $\boxed{\lambda = \frac{c}{f}}$

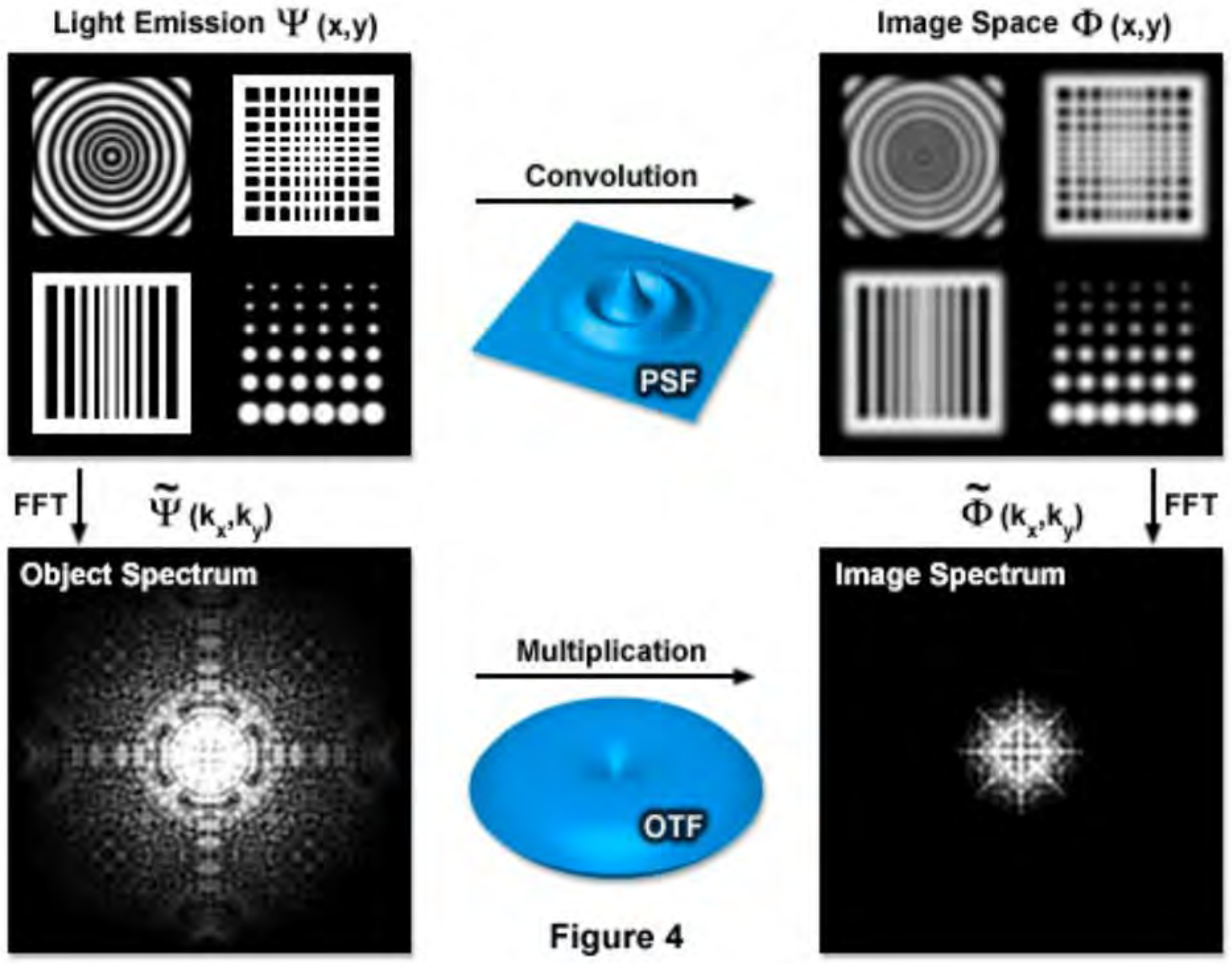


Optical Transfer Function & Point Spread Function - reminder

Convolution with the PSF acts as a **low-pass filter**

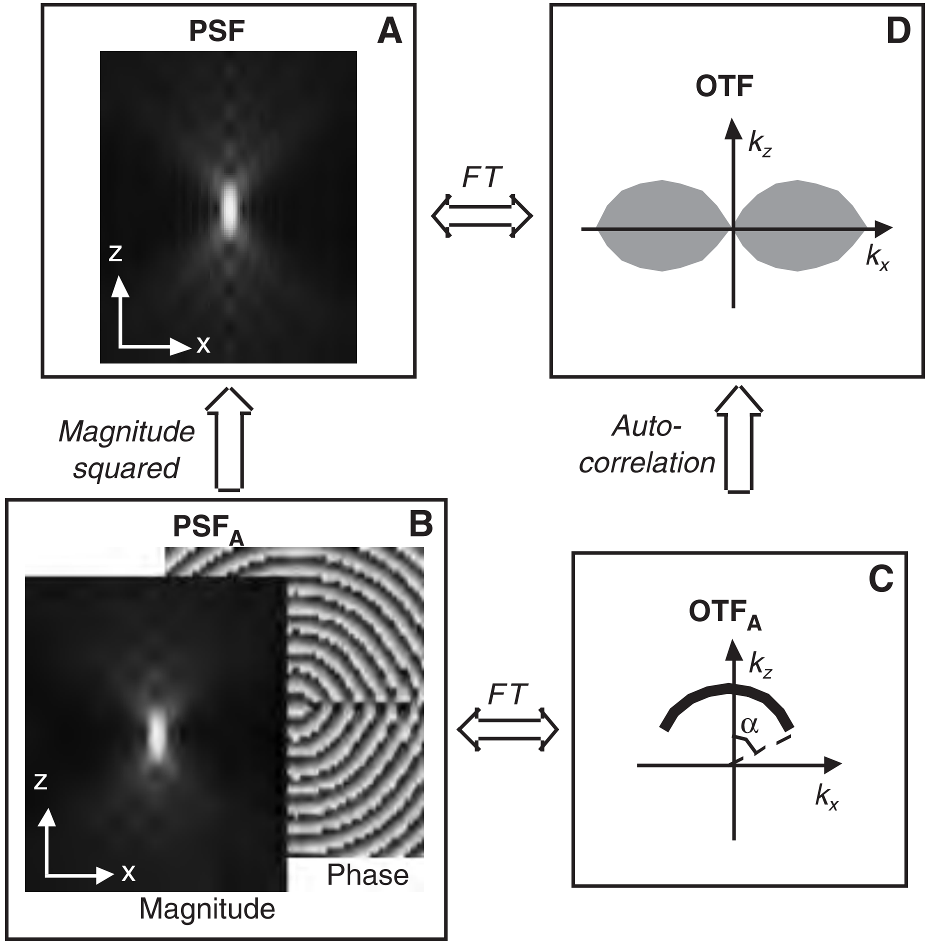


OTF is the normalized **Fourier transform of the PSF** of the optical system



http://zeiss-campus.magnet.fsu.edu/articles/superresolution/supersim.html

Pupil Function, Optical Transfer Function & Point Spread Function



Wave vector

$\vec{k} = (k_x, k_y, k_z)$

$k_x{}^2 + k_y{}^2 + k_z{}^2 = k^2 = \left(\frac{2\pi}{\lambda}\right)^2$

$k = \frac{2\pi}{\lambda} = 2\pi\upsilon$

$k_z = \sqrt{k^2 - k_x{}^2 - k_y{}^2}$

The pupil function is the **projection of the $\mathbf{OTF_A}$ onto the lateral $(\boldsymbol{k_x, k_y})$plane:**

$$PSF_A(x,y,z) = \iiint OTF_A(k_x,k_y,k_z) \cdot e^{i(k_x x + k_y y + k_z z)} dk_x dk_y dk_z =$$

$$= \iint \underbrace{\int OTF_A(k_x,k_y,k_z) dk_z}_{\triangleq P(k_x,k_y)} e^{i(k_x x + k_y y)} e^{i k_z (k_x,k_y) z} dk_x dk_y =$$

$$= \iint P(k_x,k_y) e^{i(k_x x + k_y y)} e^{i k_z (k_x,k_y) z} dk_x dk_y$$

Defocus → Spherical phase

Lateral shift → Linear phase

Pupil function

$$k_z(k_x, k_y) = \sqrt{\left(\frac{2\pi n}{\lambda}\right)^2 - (k_x^2 + k_y^2)} = 2\pi \frac{n}{\lambda} \sqrt{1 - \rho^2 \left(\frac{NA}{n}\right)^2}$$

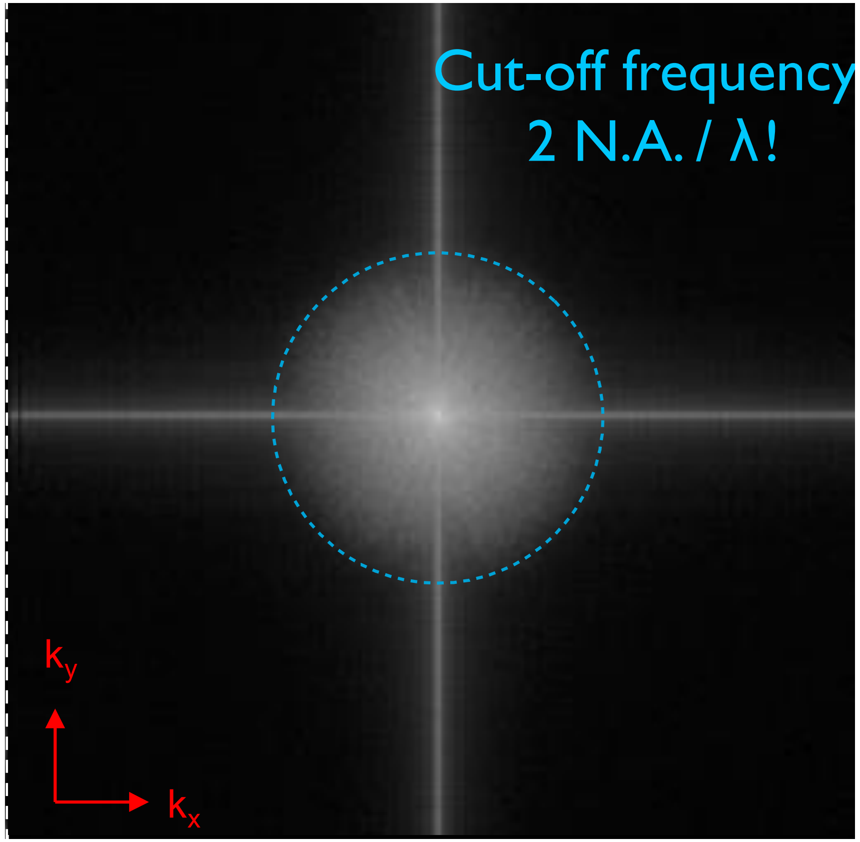
$$k_x^2 + k_y^2 \leq (2\pi NA/\lambda)^2, \rho^2 = \left(\frac{\lambda}{2\pi NA}\right)^2 (k_x^2 + k_y^2)$$

Normalized such that its value is unity at the **radius** of the limiting aperture

Maximum observable spatial frequency - reminder

Real space (x,y)

Spatial Frequency Space (k_x, k_y)

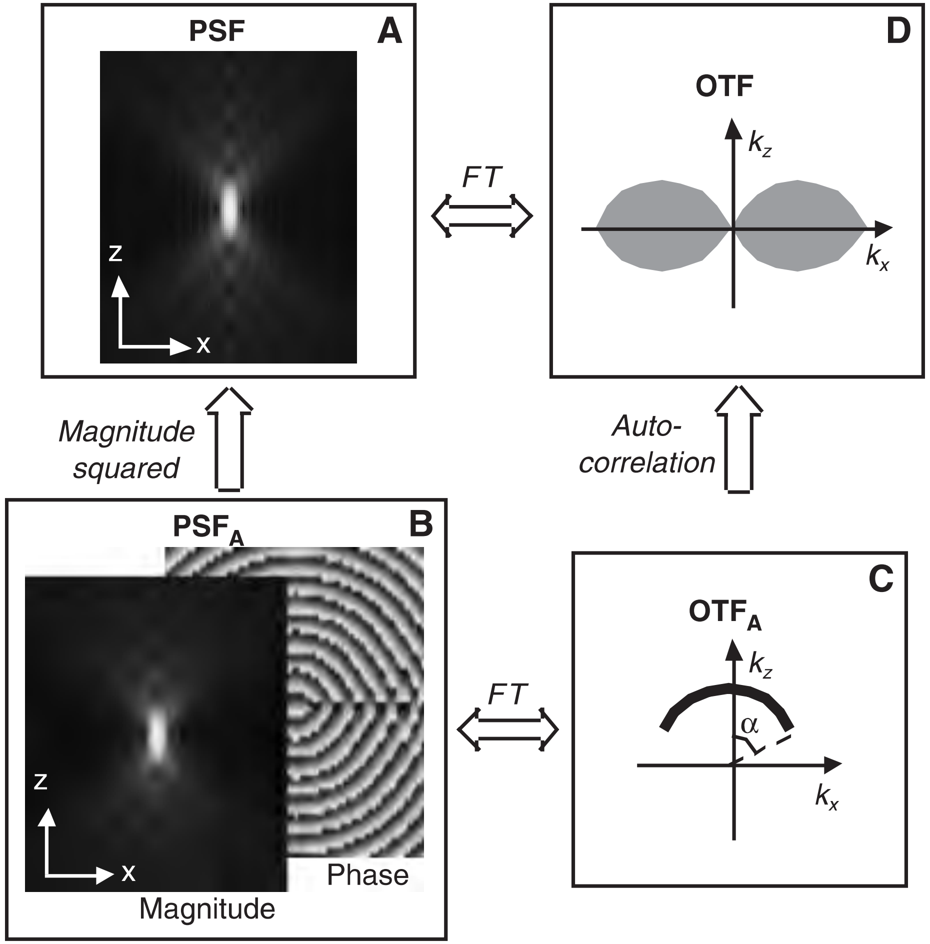


high N.A.

The **classical limit of resolution in the microscope translates into frequency space**, defining a maximum observable spatial frequency:

$k_0 = 2NA / \lambda_{em}$

Pupil Function, Optical Transfer Function & Point Spread Function



Wave vector

The pupil function is the **projection of the OTF$_A$ onto the lateral $(\boldsymbol{k_x}, \boldsymbol{k_y})$plane:**

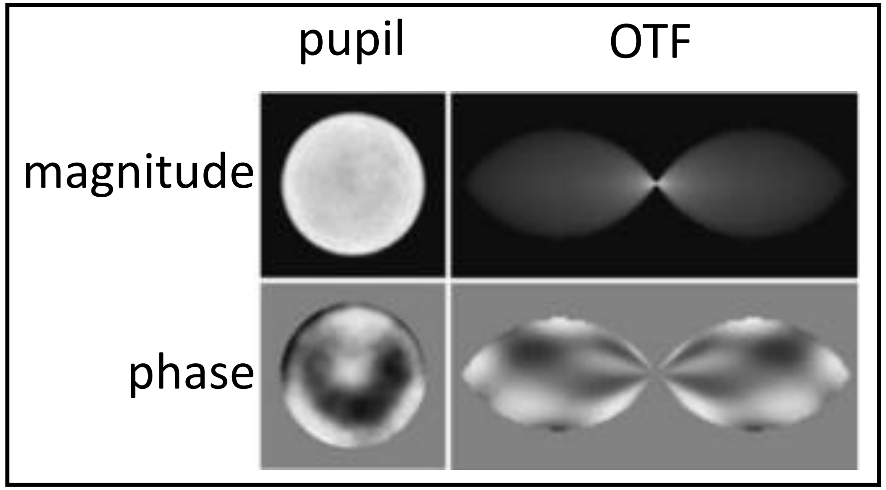
$$PSF_A(x,y,z)=\iiint OTF_A(k_x,k_y,k_z)\cdot e^{i(k_x x+k_y y+k_z z)}dk_x dk_y dk_z =$$

$$=\iint \underbrace{\int OTF_A(k_x,k_y,k_z)dk_z}_{\triangleq P(k_x,k_y)} e^{i(k_x x+k_y y)} e^{ik_z(k_x,k_y)z} dk_x dk_y =$$

$$=\iint P(k_x,k_y) e^{i(k_x x+k_y y)} e^{ik_z(k_x,k_y)z} dk_x dk_y$$

Pupil function:

- Measured by interferometric methods
- Inferred by phase retrieval algorithms

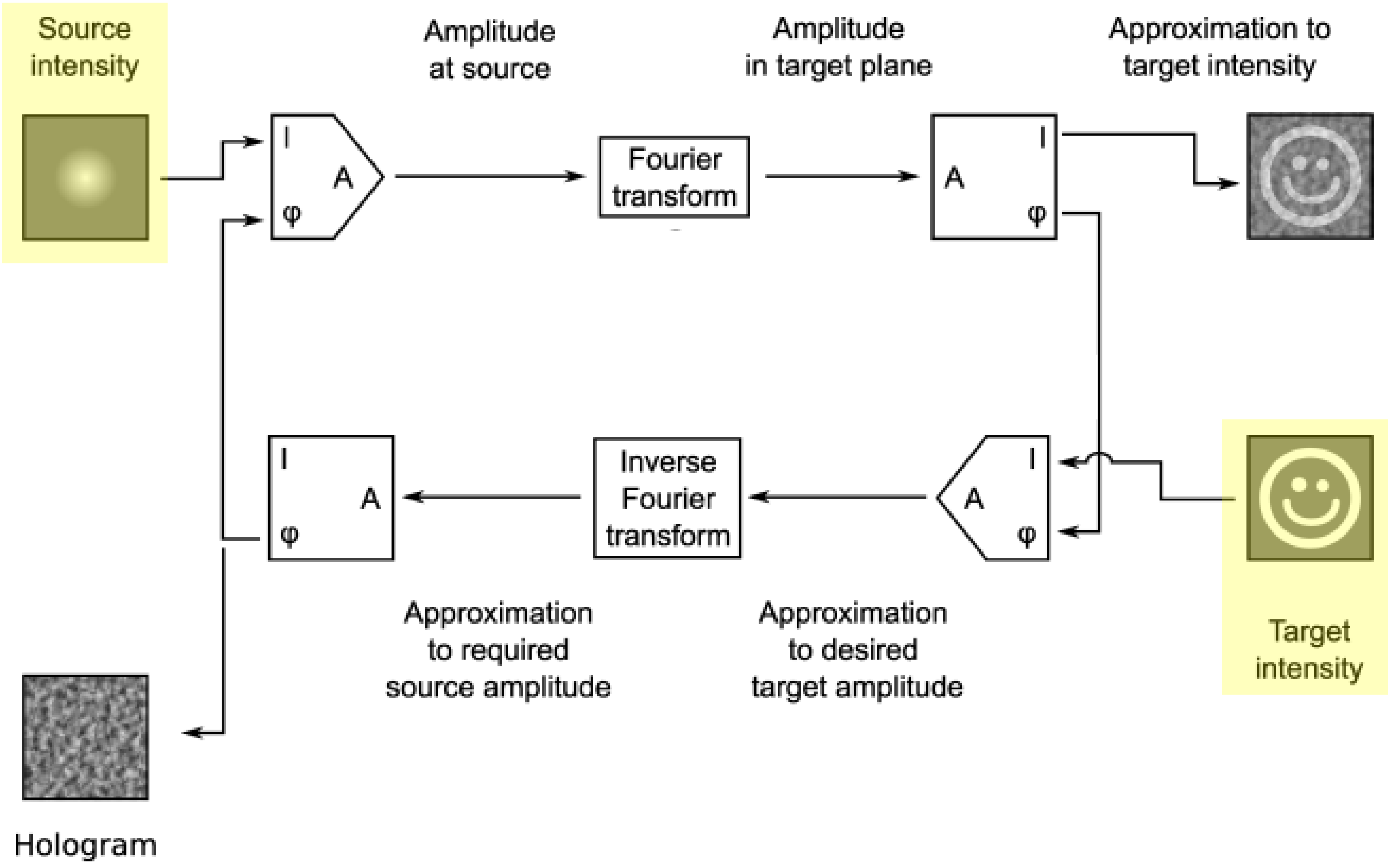


Intensity PSF vs Pupil function:

- 2D, less prone to artefacts and noise
- Compact and modifiable description of a 3D widefield fluorescence microscope

Gerchberg-Saxton Algorithm

- The most popular class of phase-retrieval methods
- Recovering a complex image from **magnitude measurements at two different planes** – **imaging plane and Fourier plane**



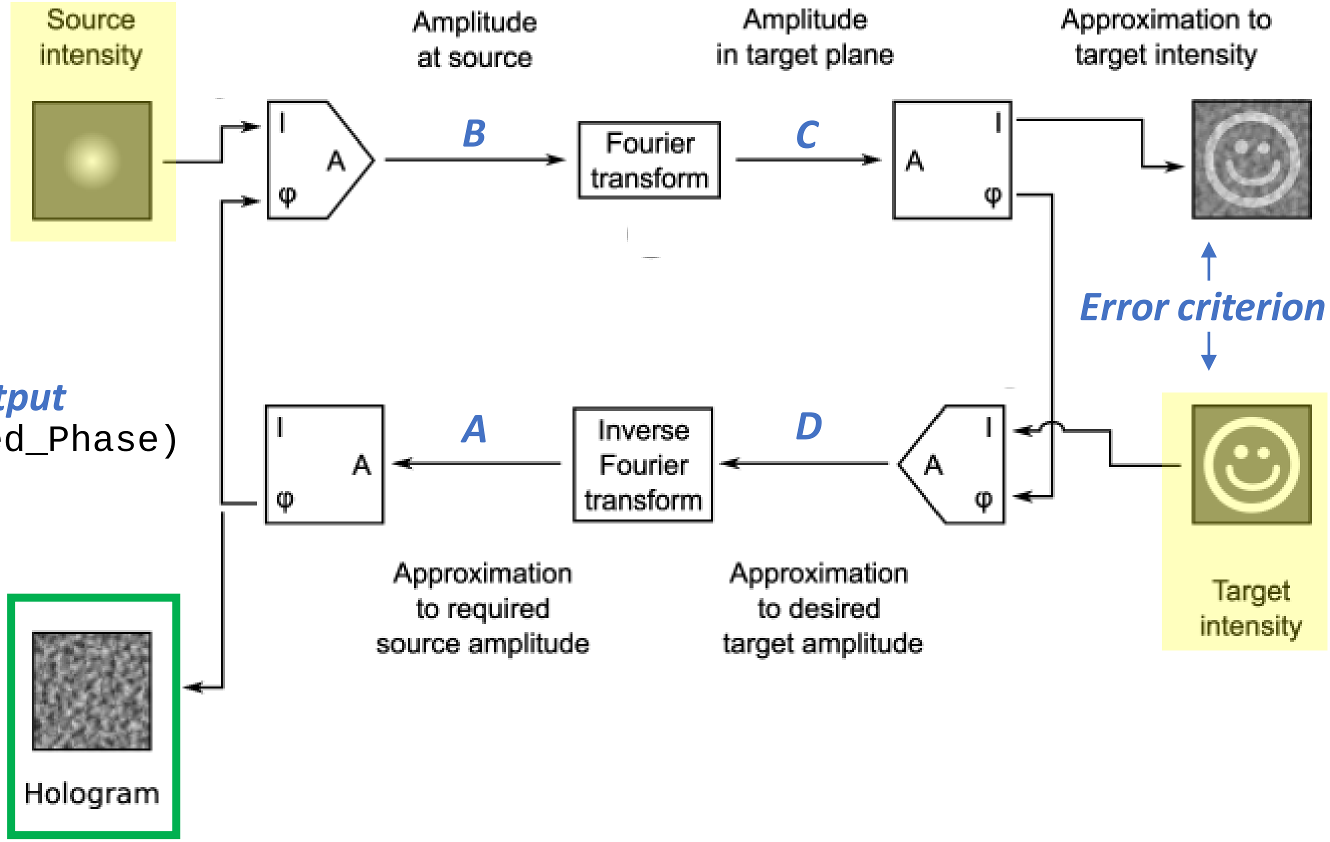
Gerchberg-Saxton Algorithm

Let:

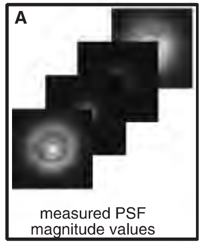
- A, B, C & D: complex planes with the same dimension as Target and Source Amplitude
- Amplitude-extracting function: e.g. for complex $z = x + iy$, amplitude(z) = sqrt($x \cdot x + y \cdot y$) for real x, amplitude(x) = $|x|$
- Phase-extracting function: e.g. phase(z) = arctan(y/x)

Inputs *Output*

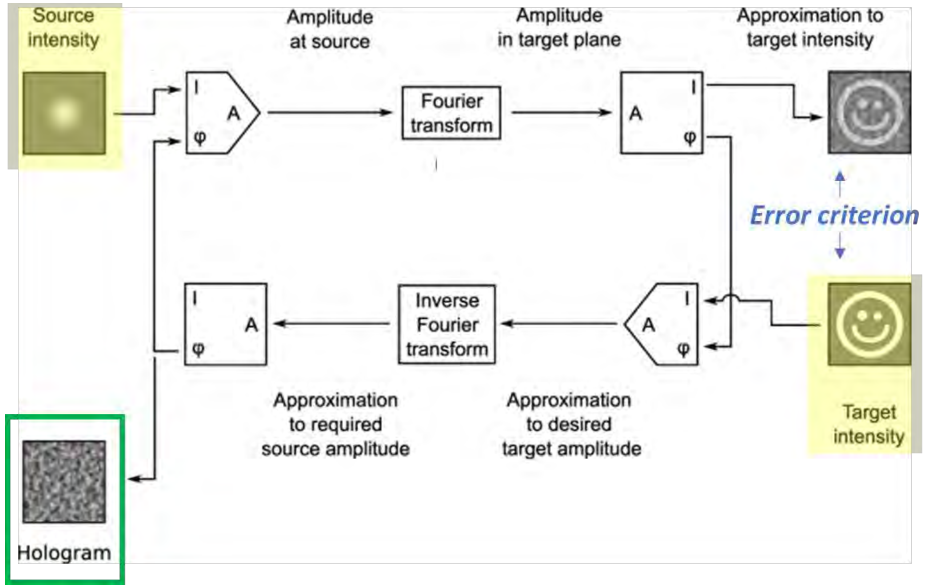
```
Gerchberg–Saxton Algorithm(Source, Target, Retrieved_Phase)
A = IFT(Target)
while error criterion is not satisfied
  B = Amplitude(Source) * exp(i*Phase(A))
  C = FT(B)
  D = Amplitude(Target) * exp(i*Phase(C))
  A = IFT(D)
end
Retrieved_Phase = Phase(A)
```



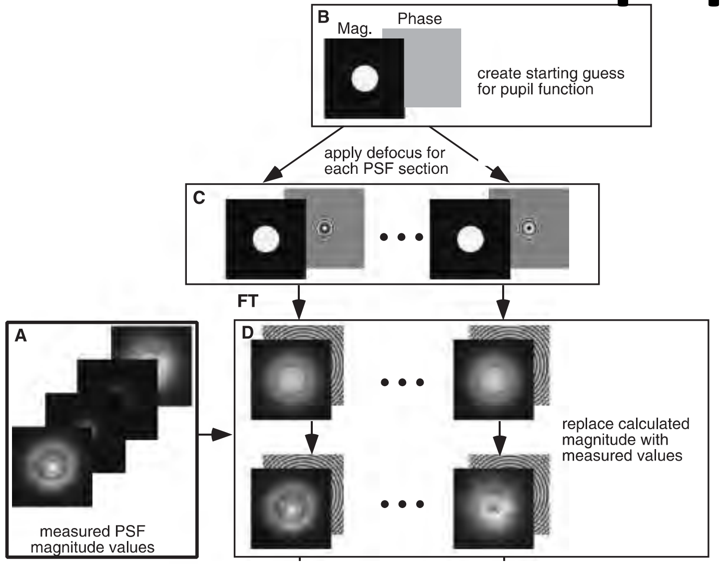
Phase-retrieved pupil functions in widefield fluorescence microscopy



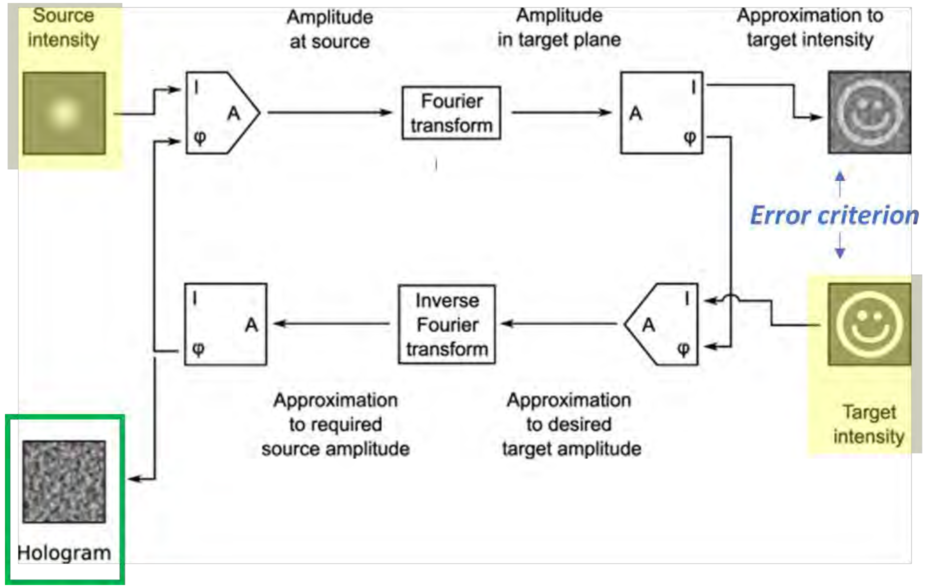
A. Collect a series of defocus images (sections) of a sub-resolution point source
B. Start a guess of the pupil function; the intensity is simply set to unity over the support defined by the objective lens NA and to zero elsewhere.
C. Apply defocus to the pupil function to create each PSF section by multiplying it with $e^{+ik_z(k_x,k_y)z}$ (i.e. the spherical phase)
D. Fourier transform the defocused-adjusted pupils to produce sections of the complex amplitude 3D PSF. The magnitudes of these calculated PSF sections are then replaced by the square root of the corresponding sections of the measured intensity data, while their phase values are left unchanged.



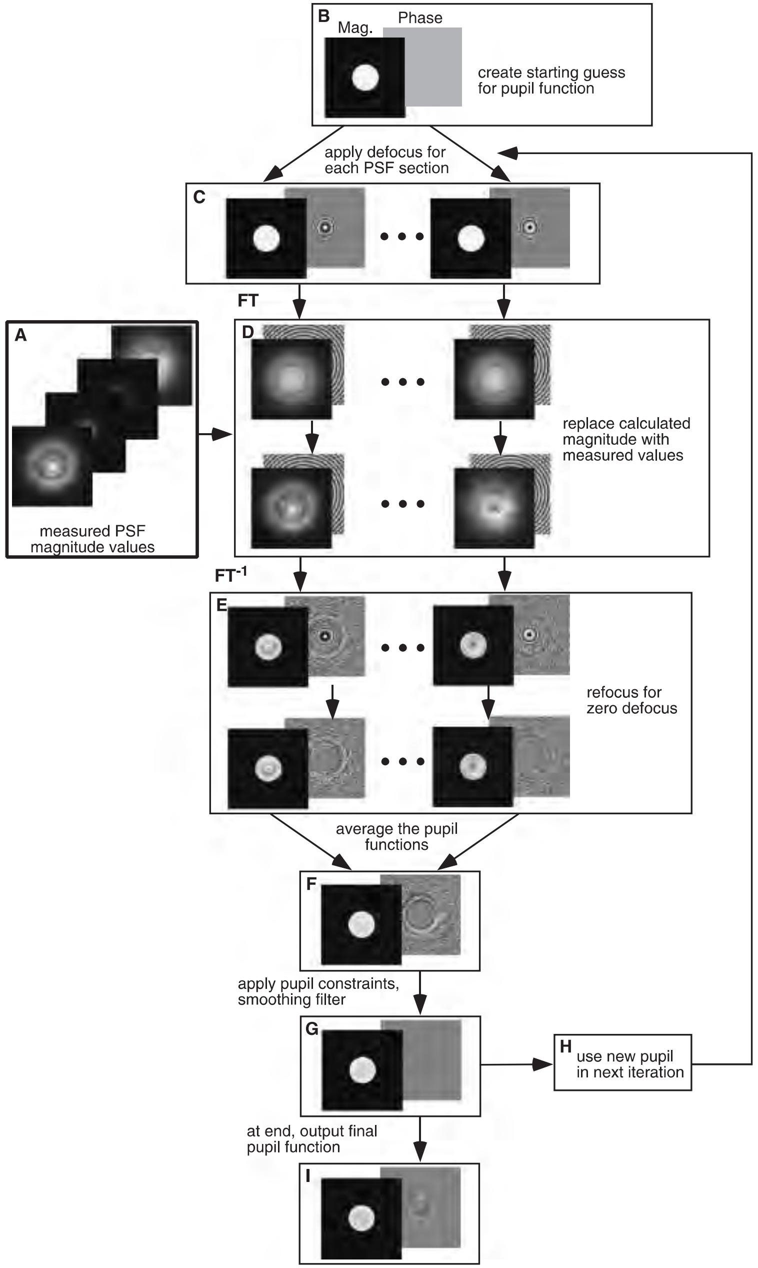
Phase-retrieved pupil functions in widefield fluorescence microscopy



E. These magnitude-corrected PSF sections are Fourier transformed back, and the defocus of each is readjusted back to zero by multiplying by the inverse defocus function $e^{-ik_z(k_x,k_y)z}$
F. These modified pupil functions are averaged to produce a single pupil function estimate
G. The NA limit constraint is then imposed to remove spatial frequency values outside of the pupil limit. A smoothing filter to suppress noise may optionally be applied.
H. This new pupil function estimate forms the starting pupil for the next iteration.
I. After a stopping criterion has been reached, the final pupil function estimate is output.



Phase-retrieved pupil functions in widefield fluorescence microscopy



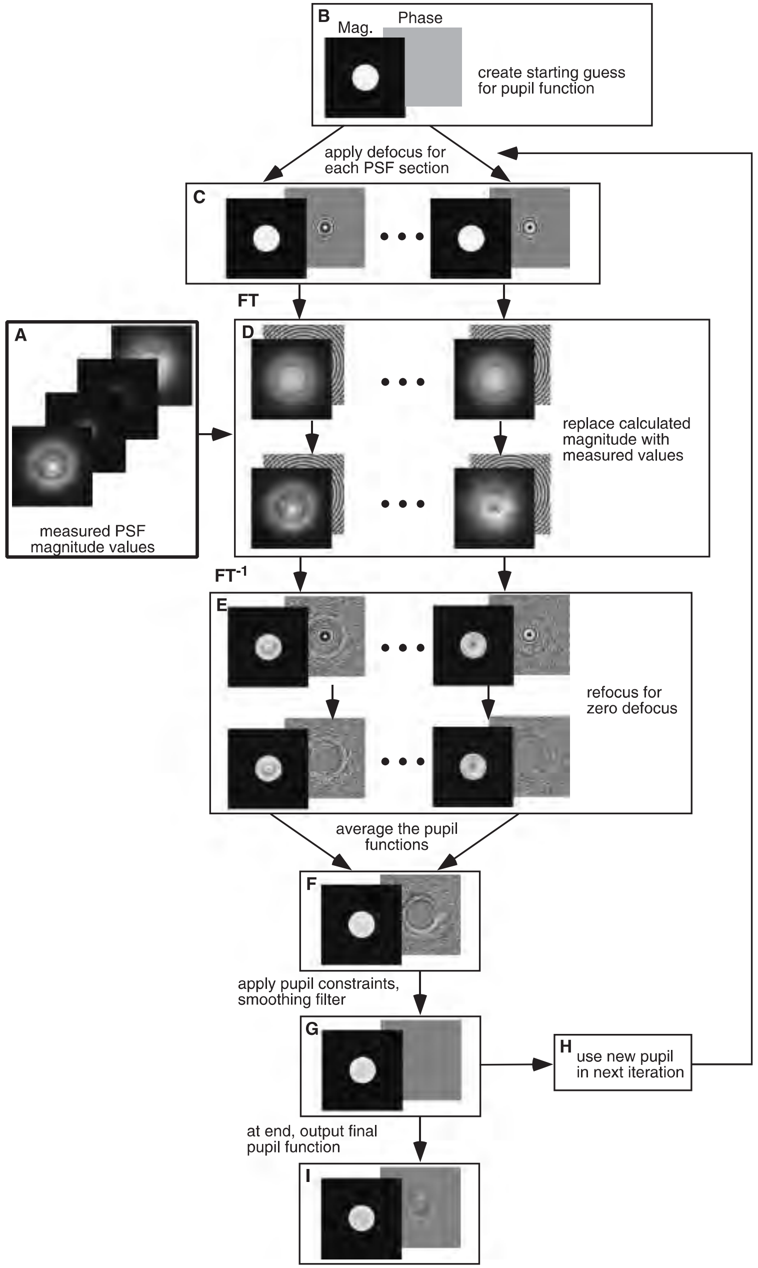
Practical considerations:

- It is possible to estimate the unknown phase information because of the redundancy provided by the multiple focus levels in the measured PSF and because of a priori knowledge of wavelength and NA, which place geometric constraints on the pupil function.

- Unlike in the original Gerchberg–Saxton algorithm, usually we allow the pupil function's magnitude to vary over the aperture, as one cannot generally assume that the pupil function's magnitude is constant over the pupil for high-NA systems.

- Alternatively, the pupil magnitude can be measured and used as an initial guess or fixed. Finally, the pupil magnitude may be fixed to unity (as in the GS algorithm) or modeled.

Phase-retrieved pupil functions in widefield fluorescence microscopy

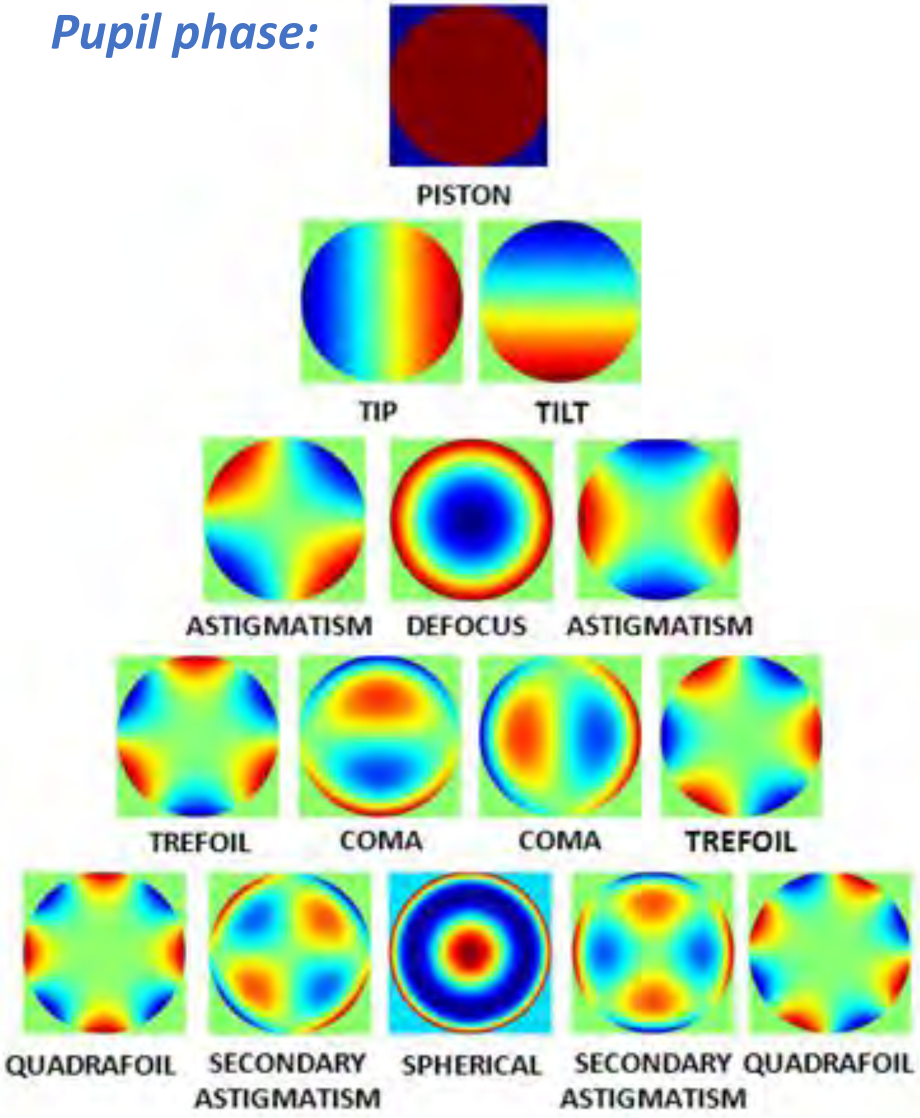


Practical considerations:

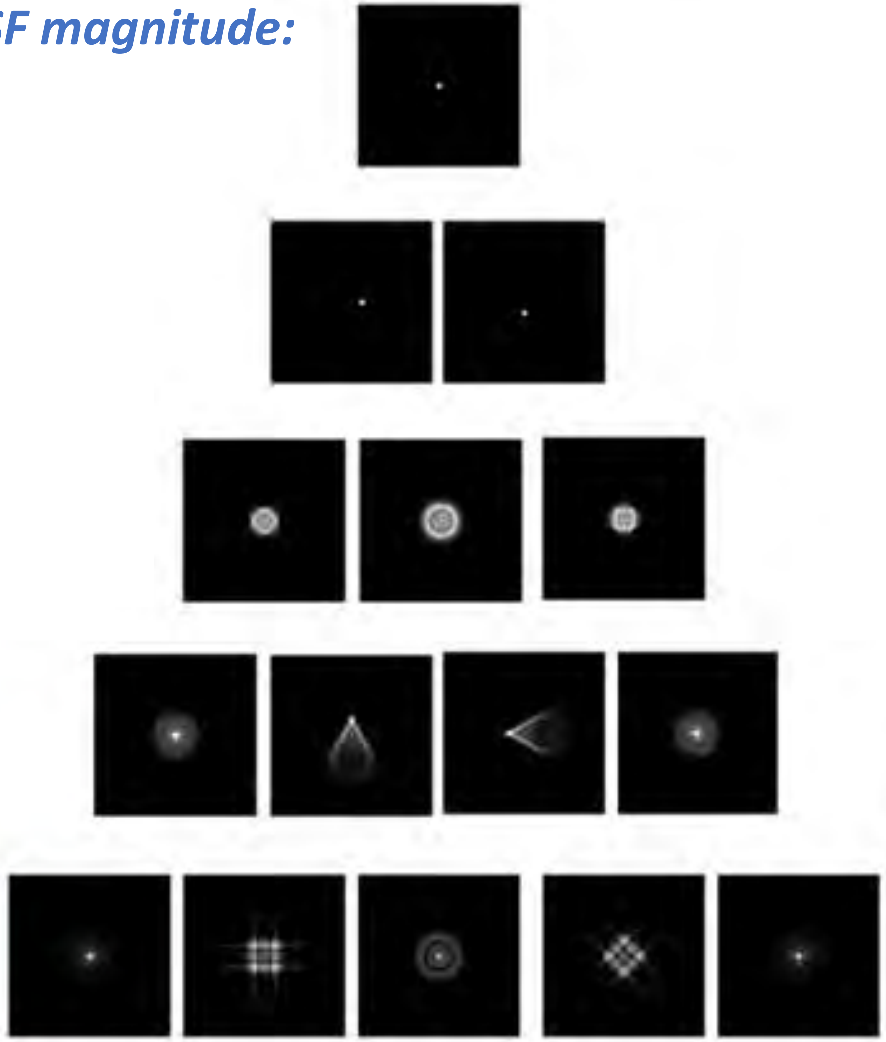
- <u>Smoothing constraints may be applied on the phase and/or the magnitude</u> of the pupil function. Robustness against 'dust-related' features.
- <u>Conservation of energy between the intensity PSF and the pupil plane magnitude</u> (***remember the normalization of FFT***)
- <u>Assumptions made</u>: bead size, vectorial nature of light, index mismatch, wavefront compression. See Hanser et al. Journal of microscopy (2004)

Zernike Polynomials and Optical Aberrations

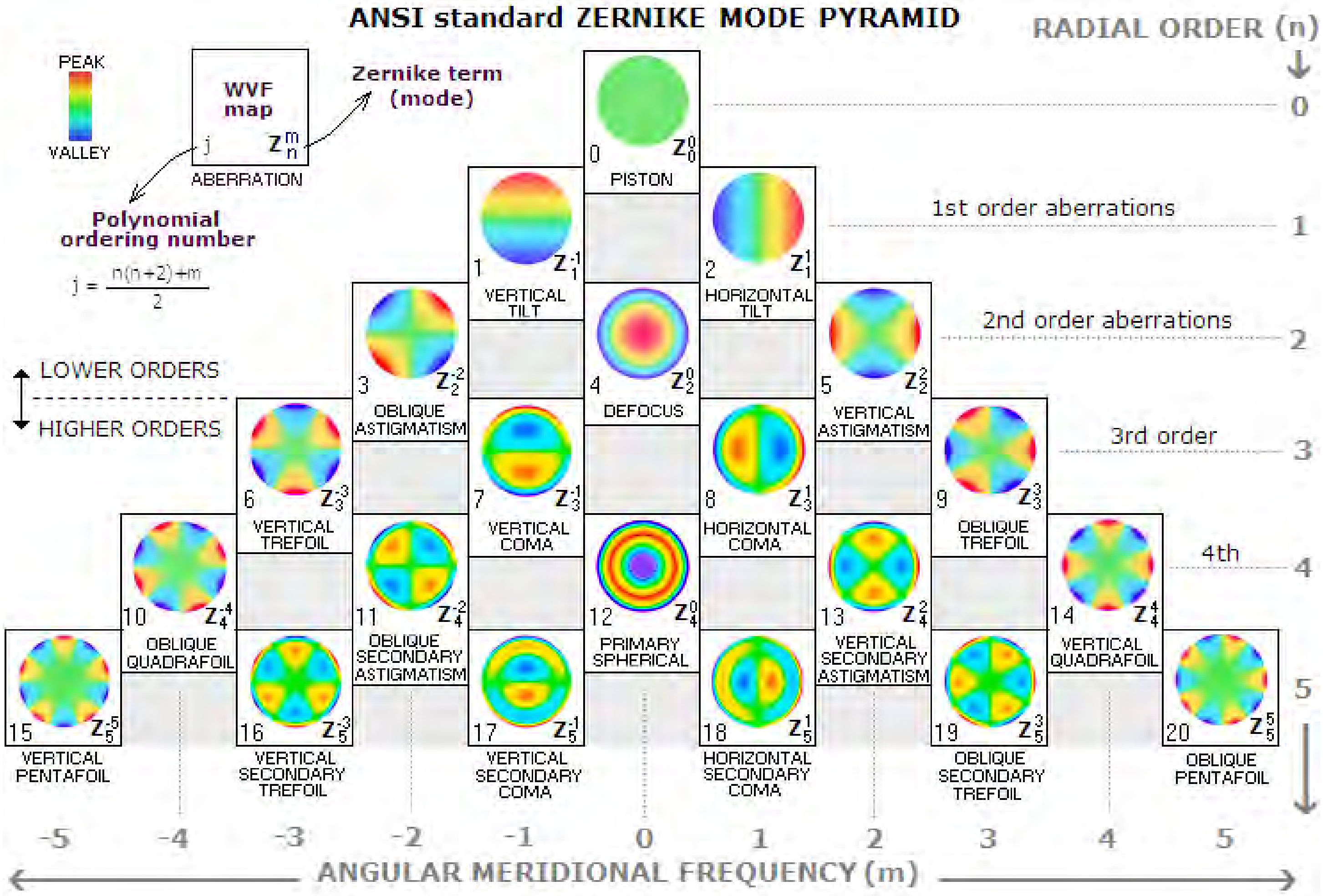
Pupil phase:



PSF magnitude:



Zernike Polynomials and Optical Aberrations



- reduces the optical aberration function to a **few coefficients**
- removes fine-scale noise
- **provides meaningful information about the optical system**

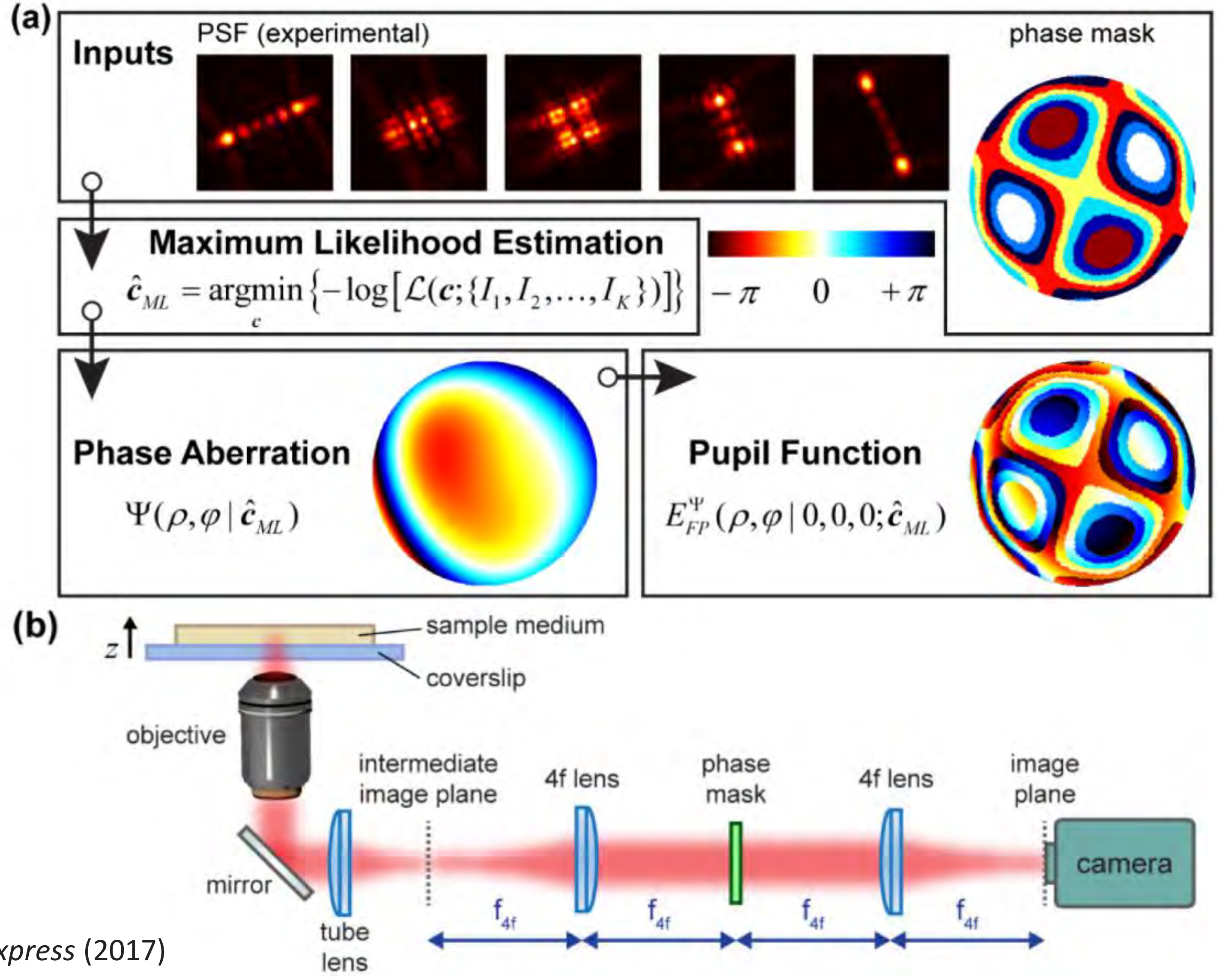
Zernike Polynomials and Optical Aberrations

Z_n^m	Radial degree (n)	Azimuthal degree (m)	Z_j	Classical name
Z_0^0	0	0	1	Piston
Z_1^{-1}	1	-1	$2\rho\sin\theta$	Tilt (vertical tilt)
Z_1^1	1	1	$2\rho\cos\theta$	Tip (horizontal tilt)
Z_2^{-2}	2	-2	$\sqrt{6}\rho^2\sin 2\theta$	Oblique astigmatism
Z_2^0	2	0	$\sqrt{3}(2\rho^2-1)$	Defocus
Z_2^2	2	2	$\sqrt{6}\rho^2\cos 2\theta$	Vertical astigmatism
Z_3^{-3}	3	-3	$\sqrt{8}\rho^3\sin 3\theta$	Vertical trefoil
Z_3^{-1}	3	-1	$\sqrt{8}(3\rho^3-2\rho)\sin\theta$	Vertical coma
Z_3^1	3	1	$\sqrt{8}(3\rho^3-2\rho)\cos\theta$	Horizontal coma
Z_3^3	3	3	$\sqrt{8}\rho^3\cos 3\theta$	Oblique trefoil
Z_4^{-4}	4	-4	$\sqrt{10}\rho^4\sin 4\theta$	Oblique quadrafoil
Z_4^{-2}	4	-2	$\sqrt{10}(4\rho^4-3\rho^2)\sin 2\theta$	Oblique secondary astigmatism
Z_4^0	4	0	$\sqrt{5}(6\rho^4-6\rho^2+1)$	Primary spherical
Z_4^2	4	2	$\sqrt{10}(4\rho^4-3\rho^2)\sin 2\theta$	Vertical secondary astigmatism
Z_4^4	4	4	$\sqrt{10}\rho^4\cos 4\theta$	Vertical quadrafoil

Normalized such that:

$$\int_0^{2\pi}\int_0^1 Z_j^2\,\rho\,d\rho\,d\theta = \pi$$

Practical application of phase retrieval on PSF engineering and aberration correction



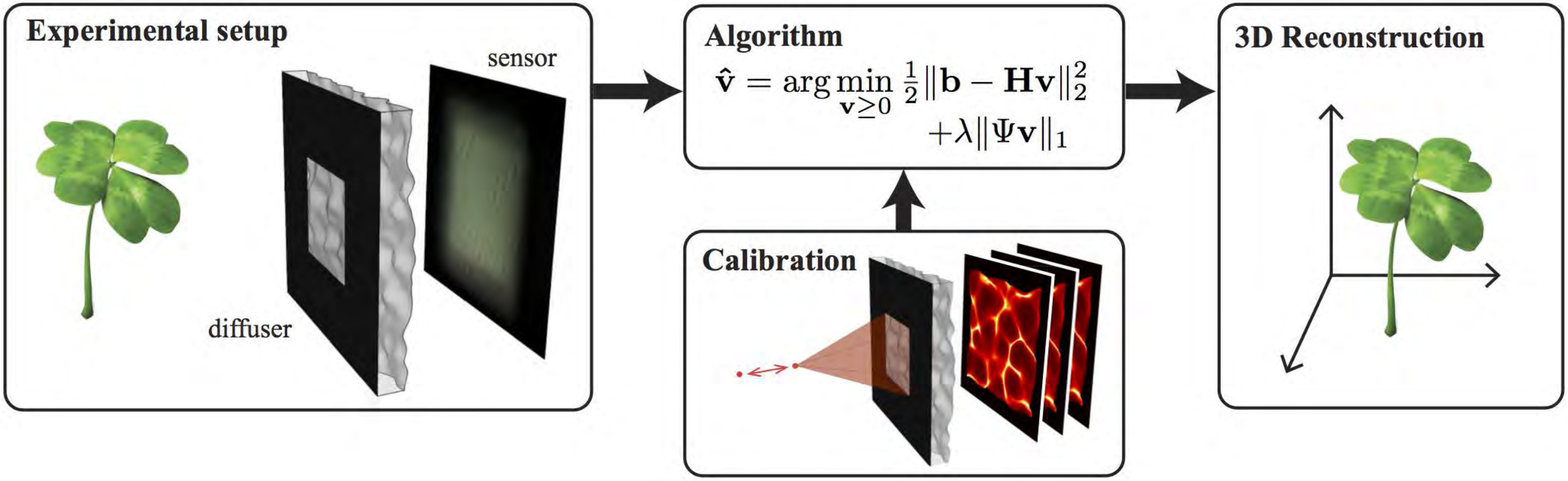
N. Petrov, P. et al. *Optics express* (2017)

Department of Biomedical Engineering,
Technion
Computational optical imaging 336547

Tutorial 10 – Lensless 3D imaging

Elias Nehme & Yoav Shechtman

05 January 2021



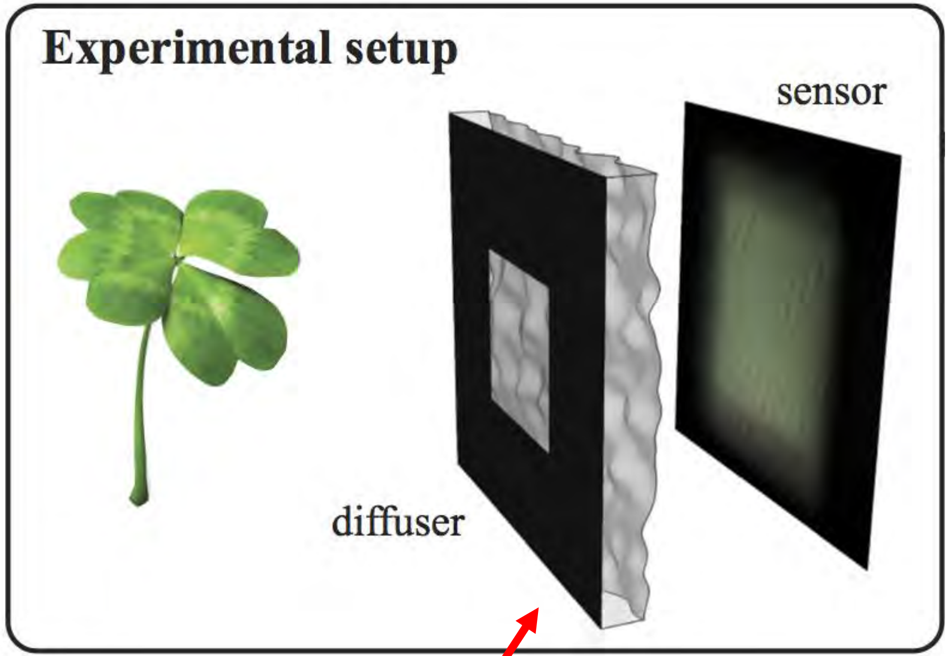
Antipa et al. Optica 5, 1-9 (2018)

"DiffuserCam" setup

What? How?!

1.3M pixel

100M voxels



Depth encoding PSF

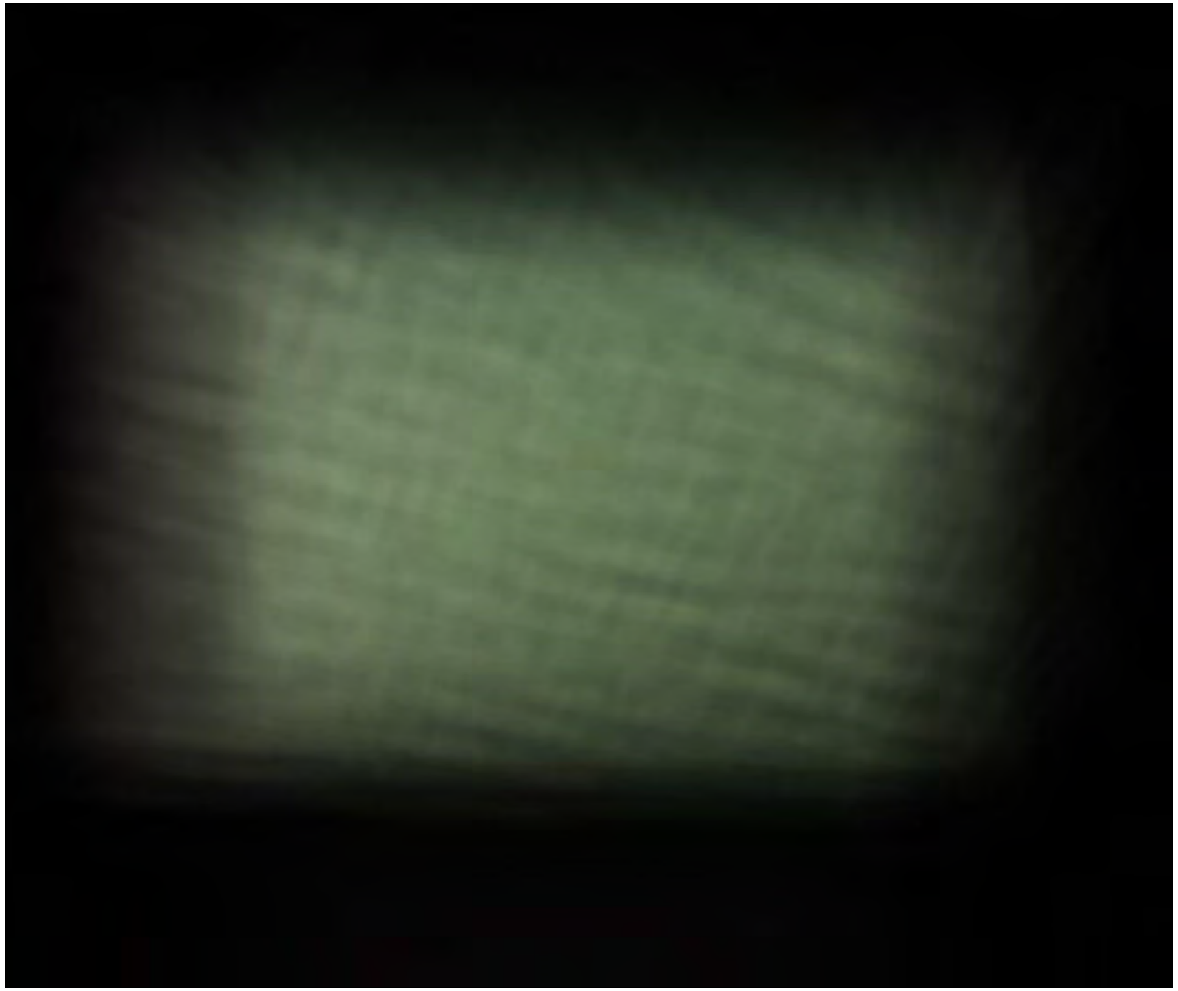
Single-shot 3D

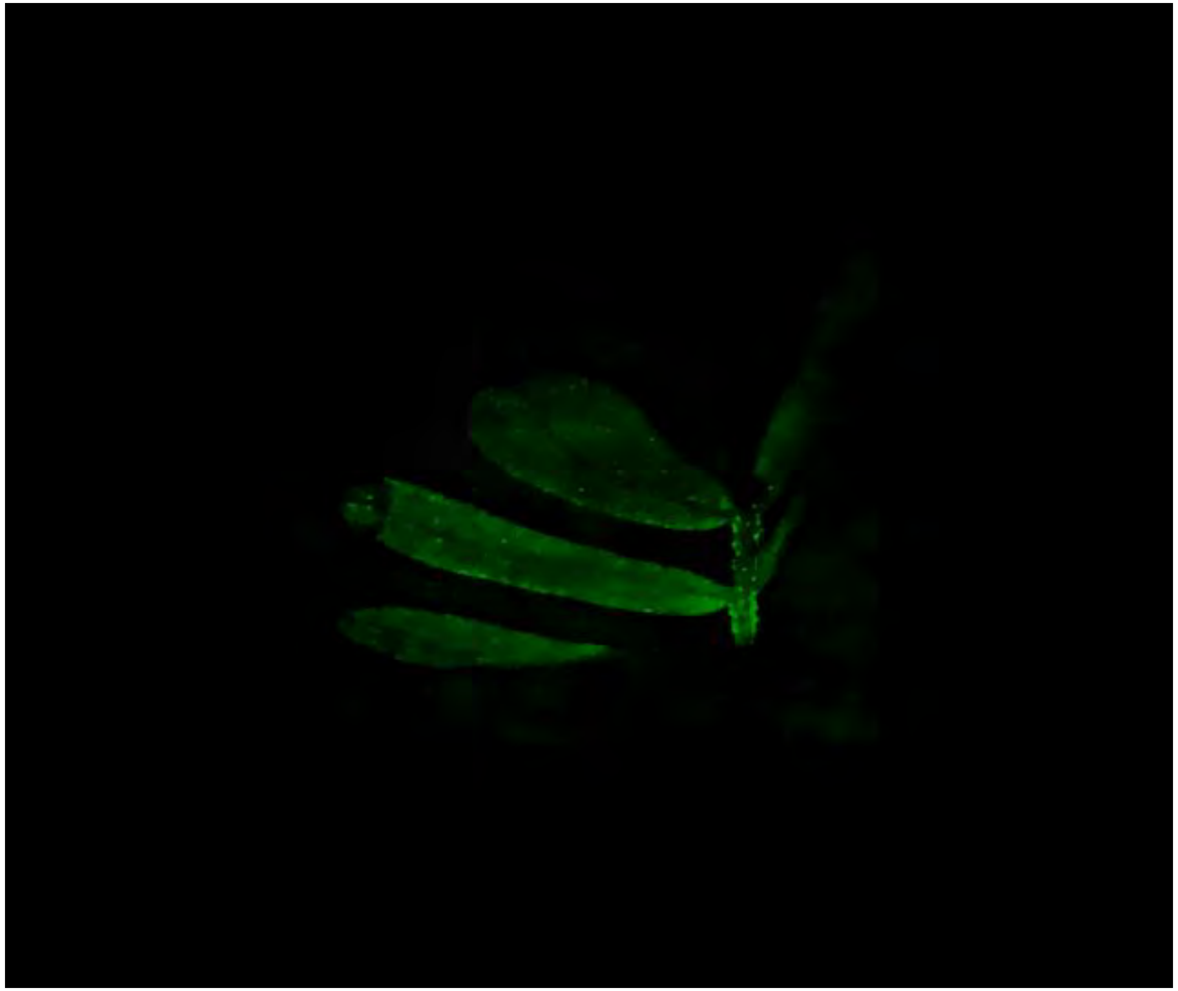
Nonlinear CS recon.

Antipa et al. Optica 5, 1-9 (2018)

Experimental 3D reconstruction from a snapshot

480x320x128 voxels reconstructed in ~3 mins





Antipa et al. Optica 5, 1-9 (2018)

Outline

► Depth-encoding PSF

► Nonlinear CS rec.

► Resolution analysis

► Experimental results and extensions

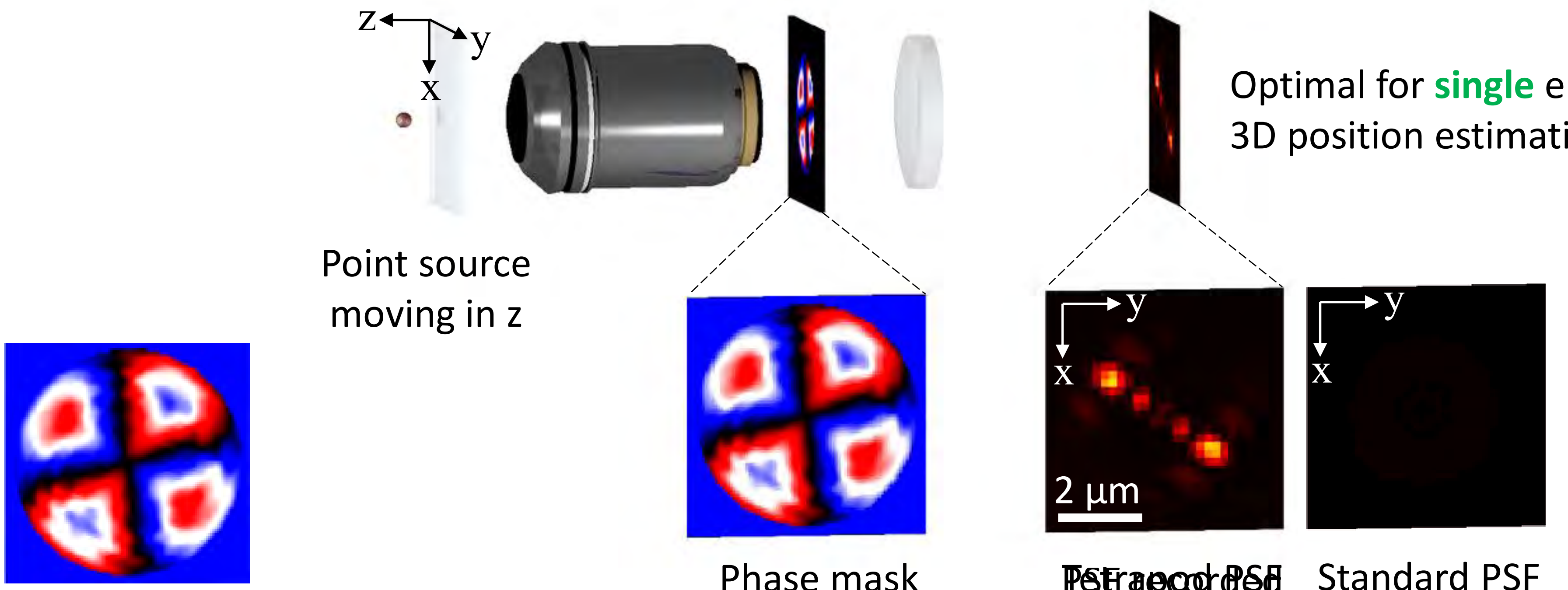
Outline

- ☛ Depth-encoding PSF
- ► Nonlinear CS rec.
- ► Resolution analysis
- ► Experimental results and extensions

Extending a microscope to 3D: reminder

- Standard microscope PSF have depth ambiguity and signal is lost
- Main idea is to encode depth into the PSF shape efficiently

Fourier plane



Optimal for **single** emitter 3D position estimation.

Point source moving in z

Phase mask

Phase mask in Fourier plane

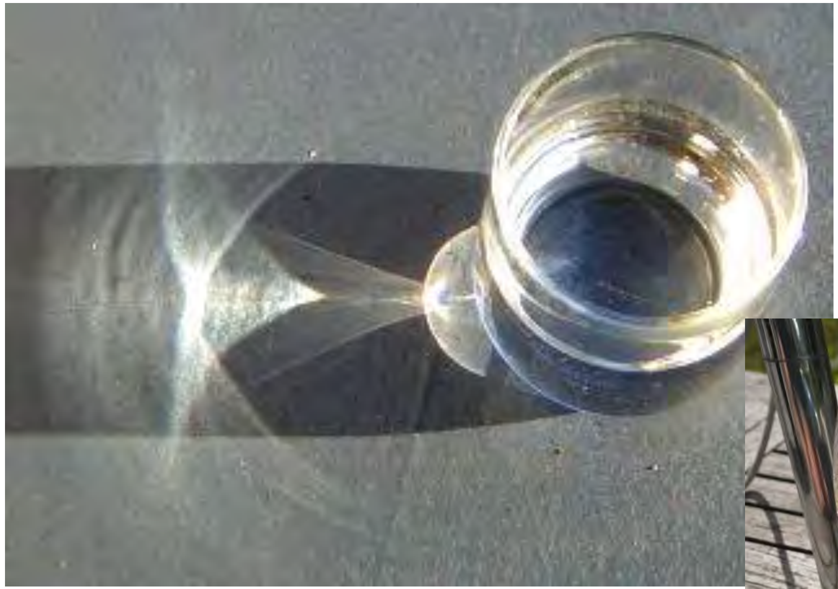
Tetrapod PSF

PSF recorded in image plane

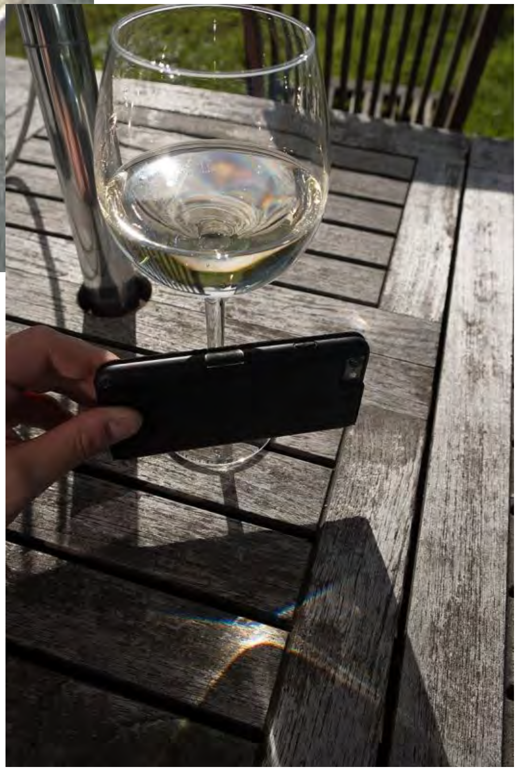
Standard PSF

Shechtman et al., Phys. Rev. Letters (2014)

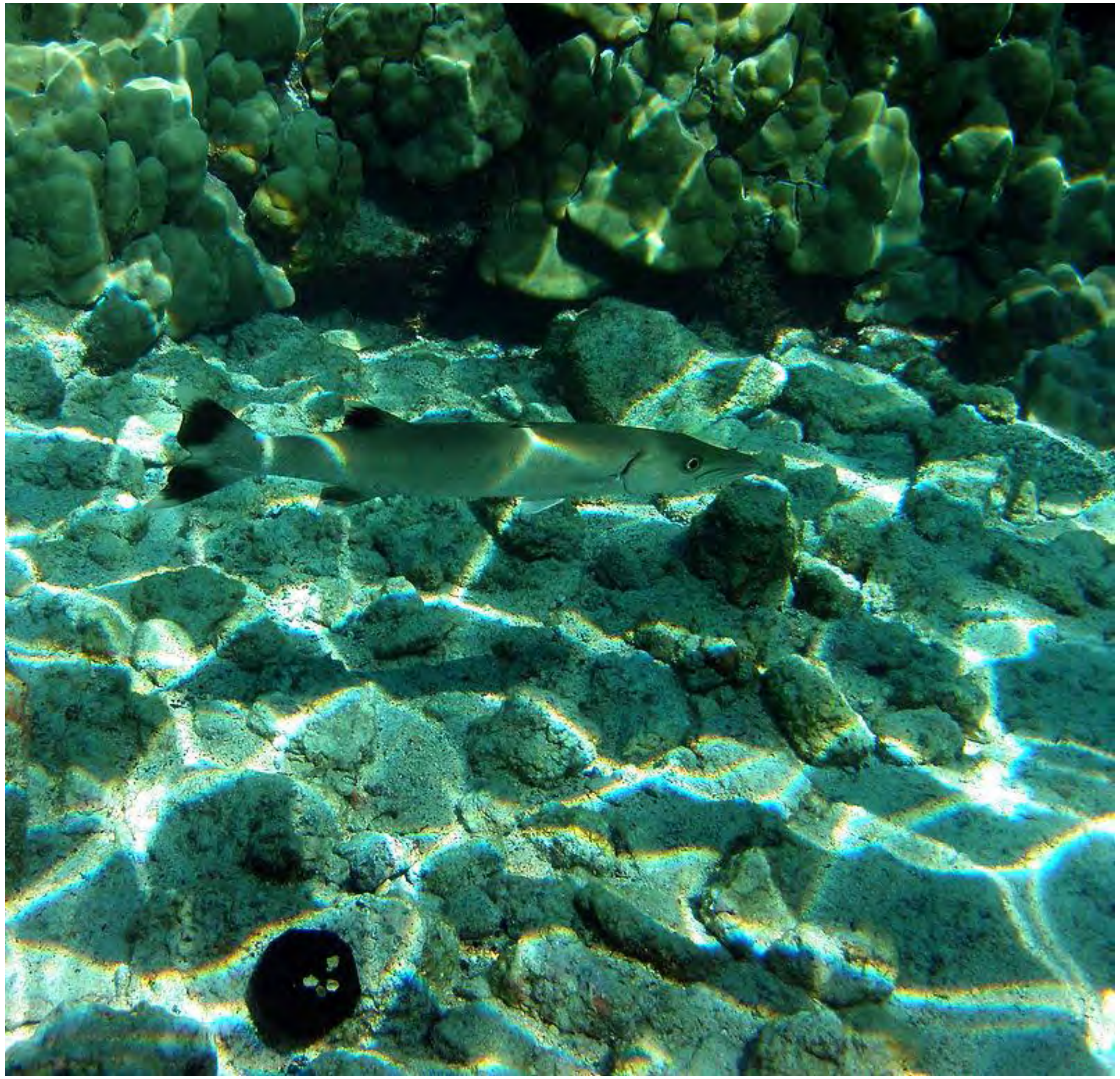
Diffuser-induced caustics PSF



Glass of water



Glass of wine



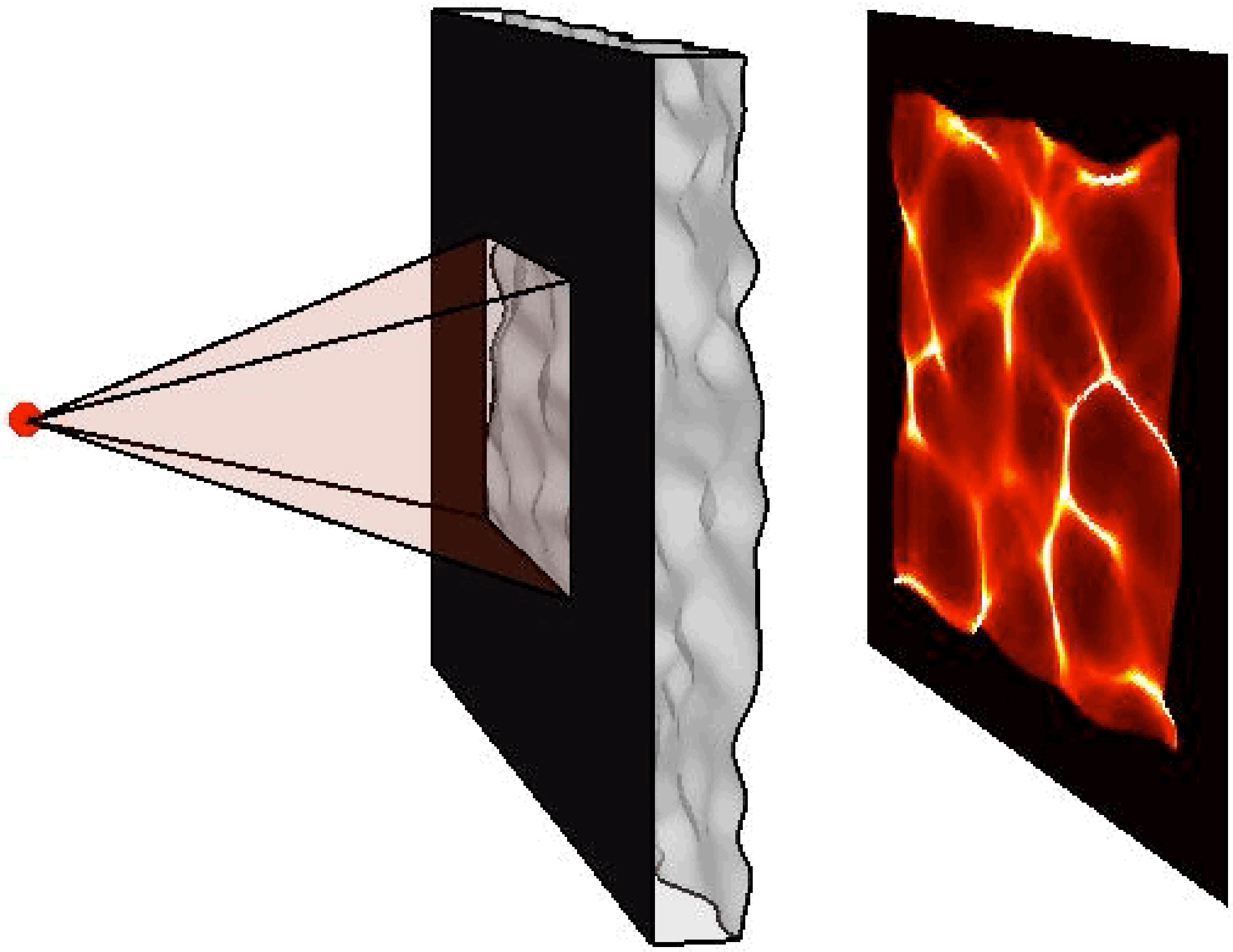
Caustics produced by the surface of water

https://en.wikipedia.org/wiki/Caustic_(optics)

Diffuser-induced caustics PSF

- Main idea is to encode depth into the PSF shape

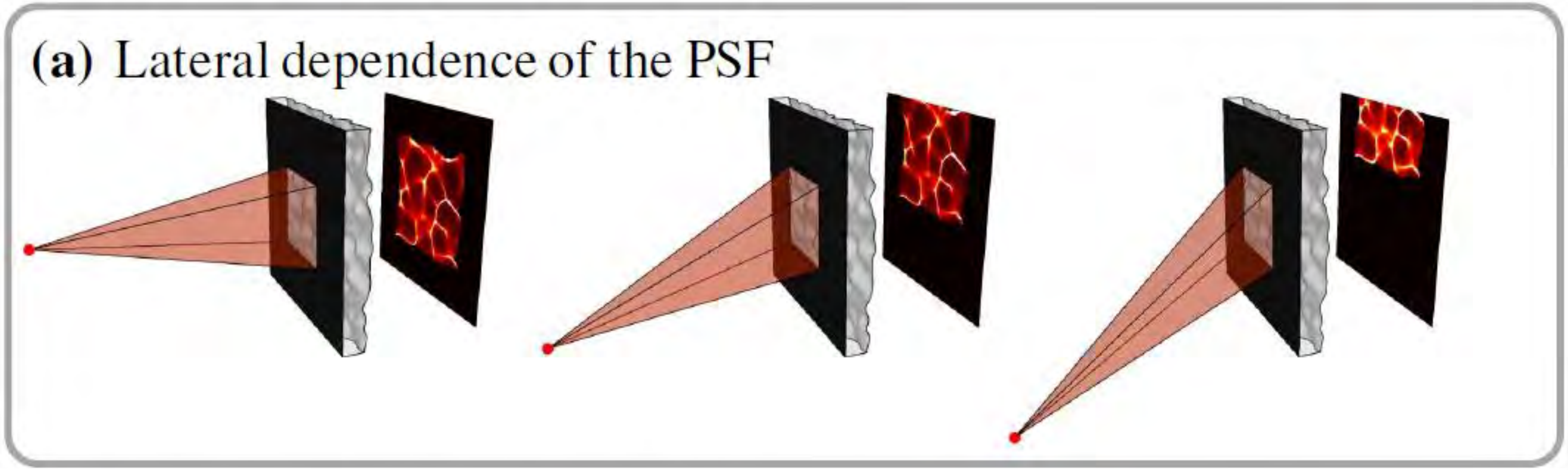
- Question: What kind of assumptions do we need on the PSF to make the inverse problem manageable?
- Answer: we really like convolution models..



Antipa et al. Optica 5, 1-9 (2018)

Diffuser-induced caustics PSF

Assumption 1: Shift-Invariant in xy.



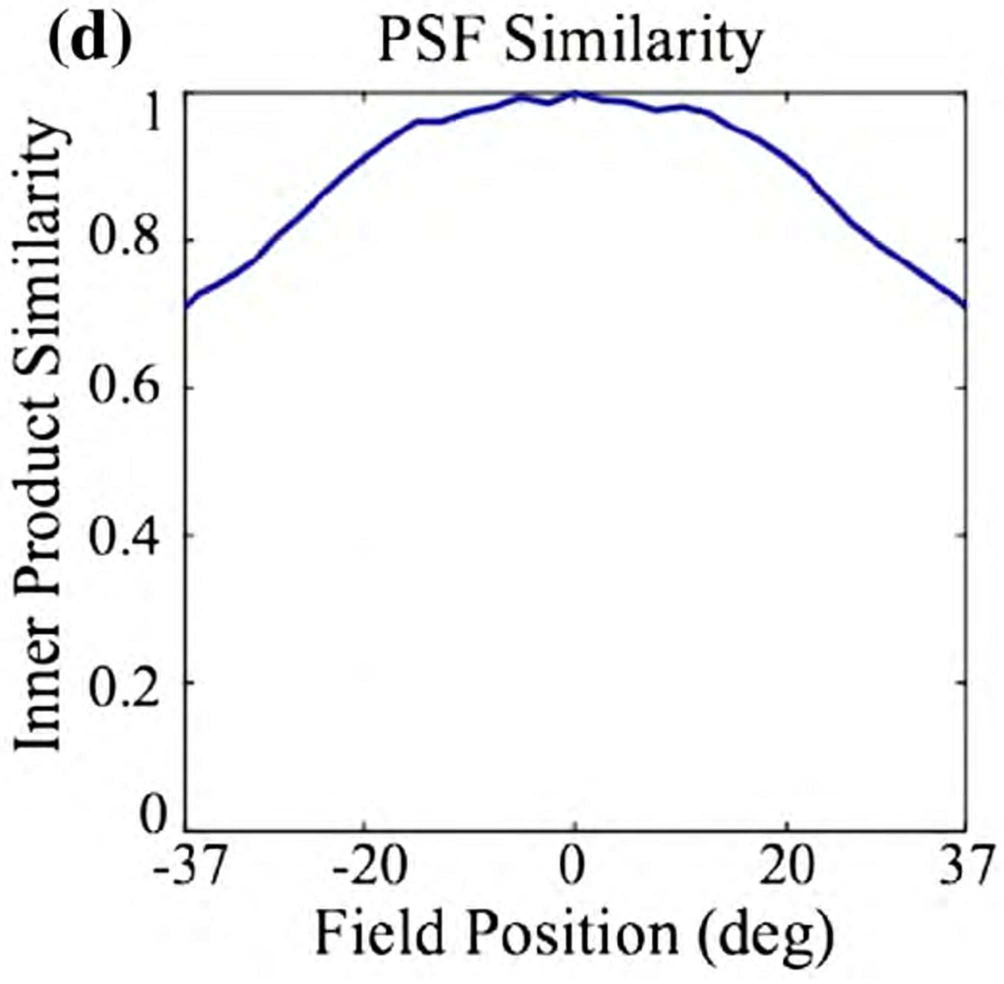
Assumption 2: Only Scaling in z.

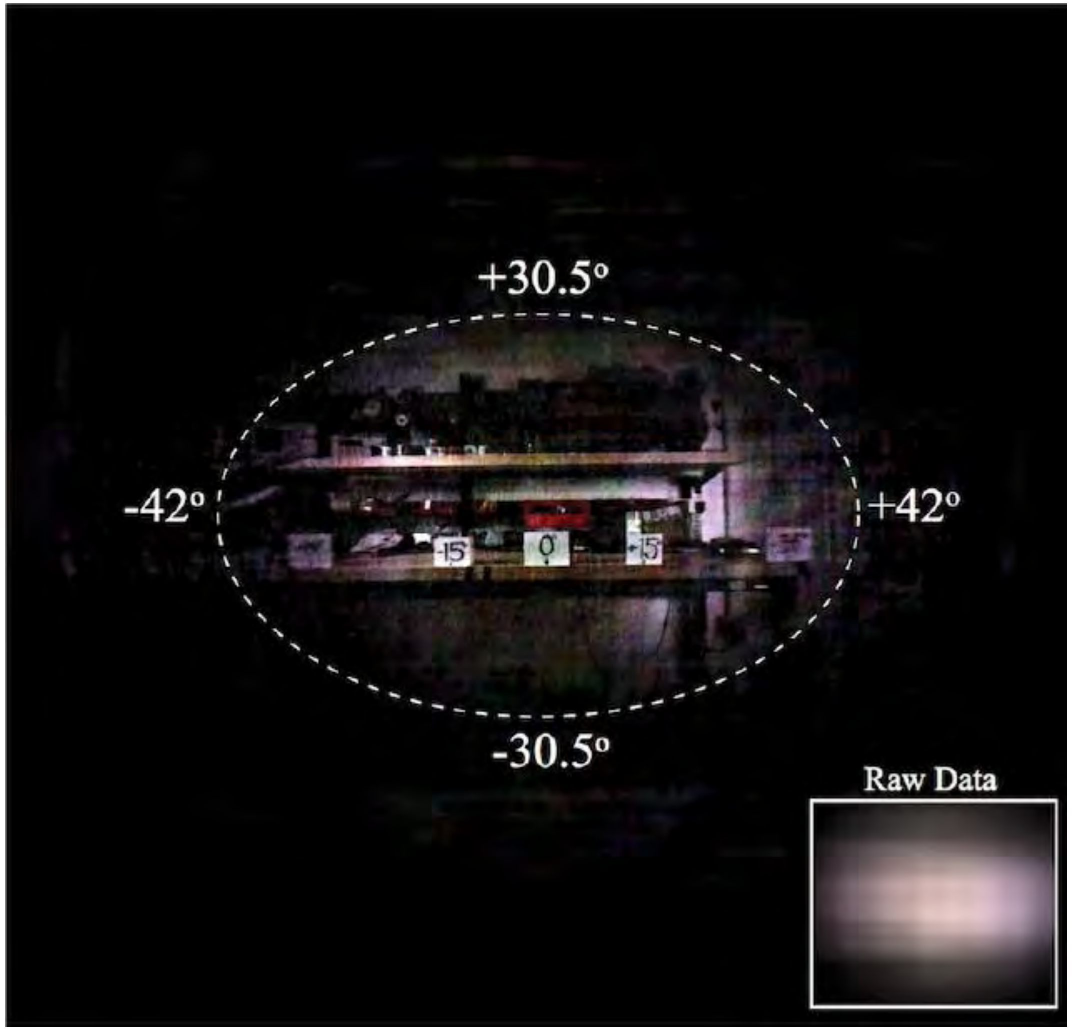
Hyperfocal plane (35.3 mm)

Minimal allowable dist. (9.9 mm)

Validity of the assumptions: Shift-invariance

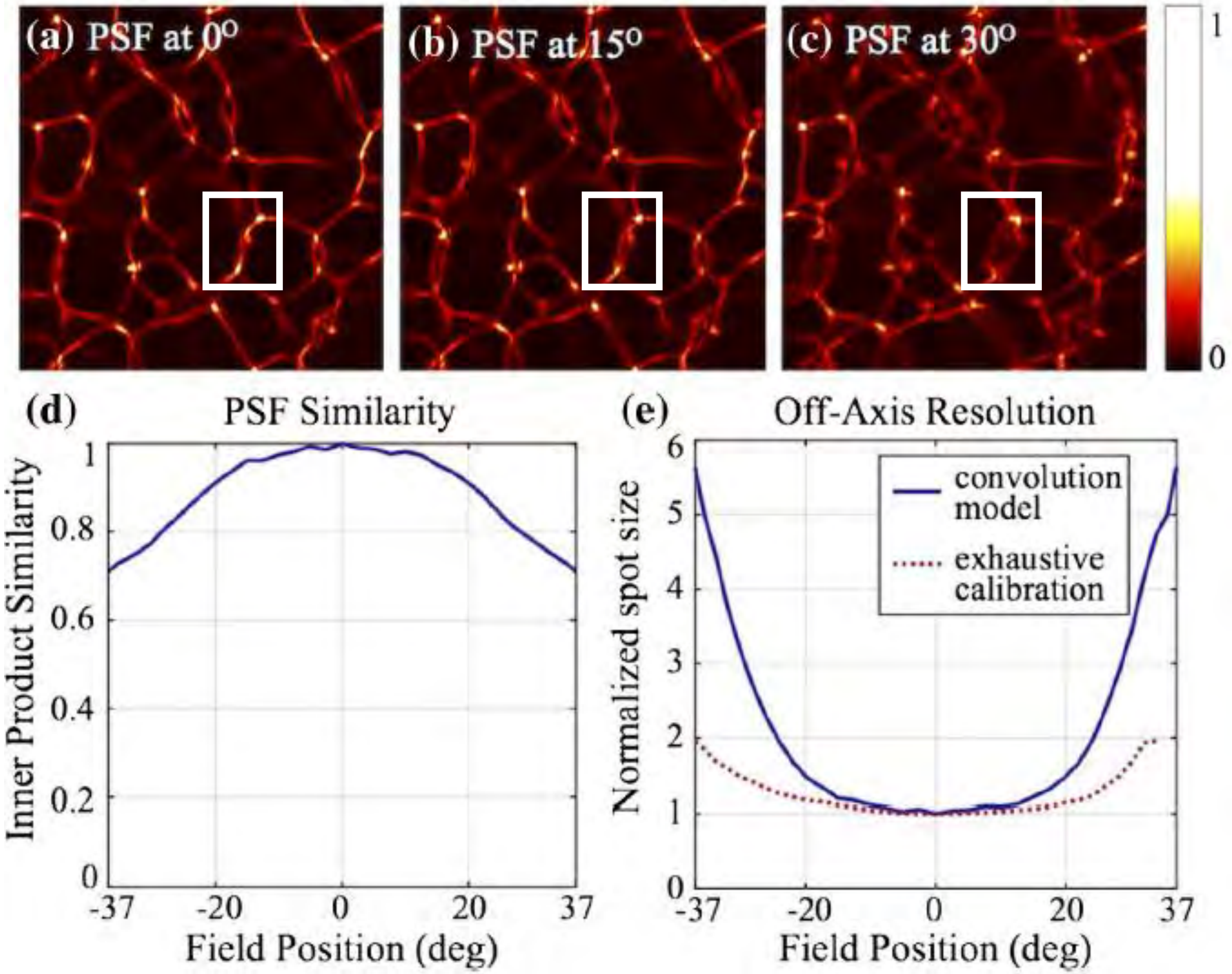
Shift-invariance in xy
Holds for reasonable FOV



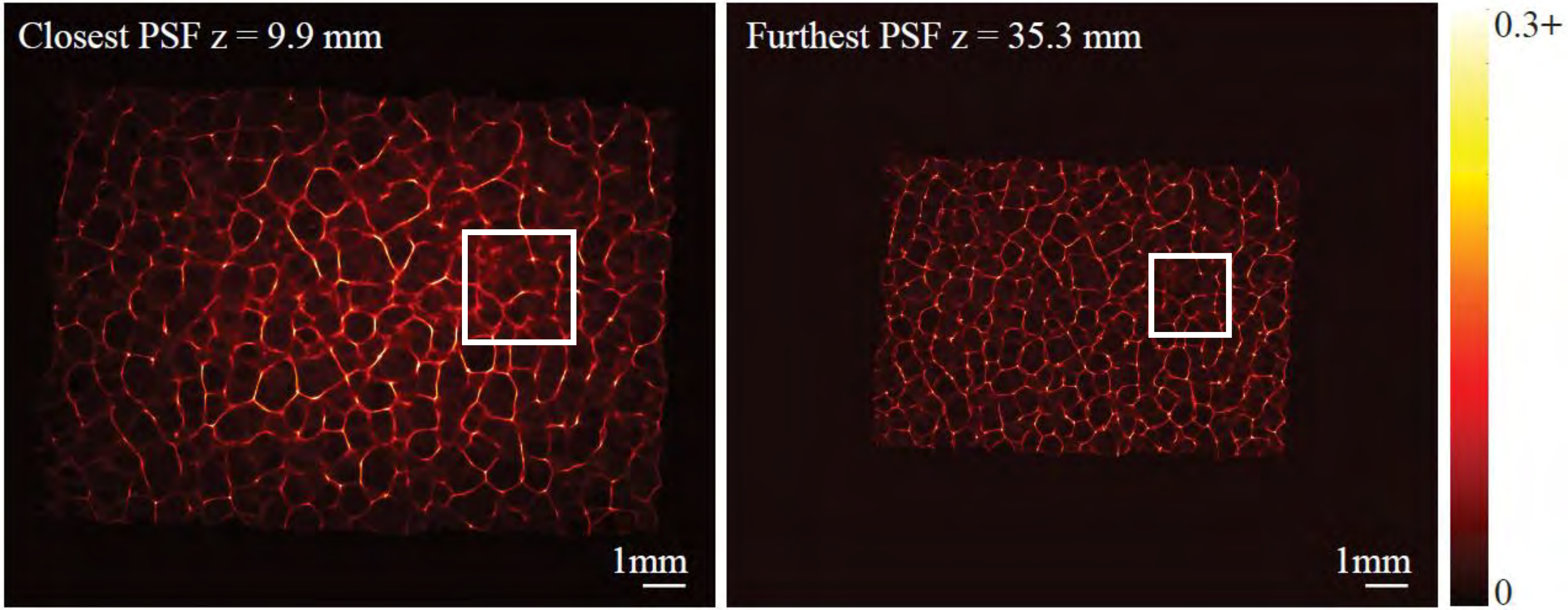


Validity of the assumptions: Shift-invariance

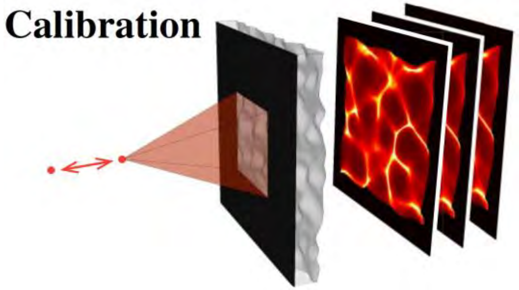
- PSFs at 0 and 15 degress are approximately shifted versions of the same pattern.
- PSFs at 30 has subtle differences.
- Inner products between the on-axis PSF and registered off-axis PSF can quantify the assumption.



Validity of the assumptions: Scaling with depth



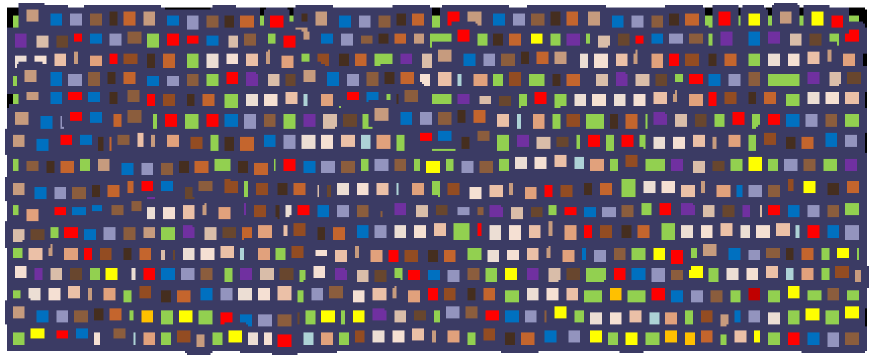
- Almost but not exactly ☹..How can we fix it?
- <u>Answer:</u> On-axis calibration!



Outline

- 👍 Depth-encoding PSF
- 👉 Nonlinear CS rec.
- ▶ Resolution analysis
- ▶ Experimental results and extensions

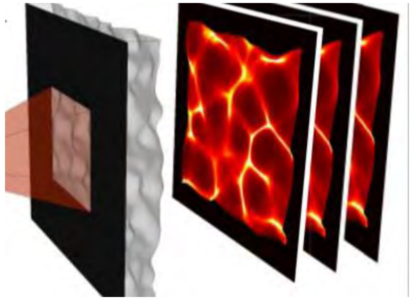
Compressed sensing: nonlinear recovery

$b \in \mathbb{R}^m$ =  $v \in \mathbb{R}^d$ + $e \in \mathbb{R}^m$

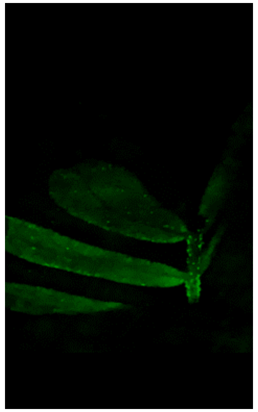
$H \in \mathbb{R}^{m \times d}$

Measured 2D image

Optical measurement operator (forward model)

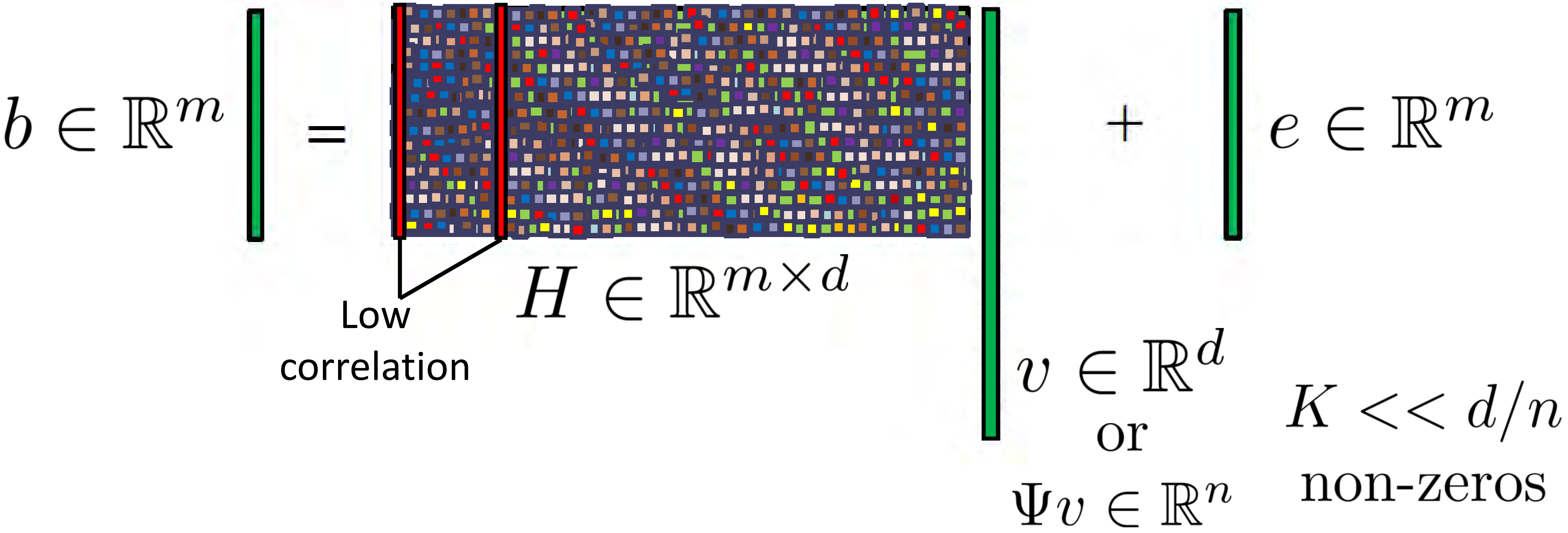


Unknown 3D volume



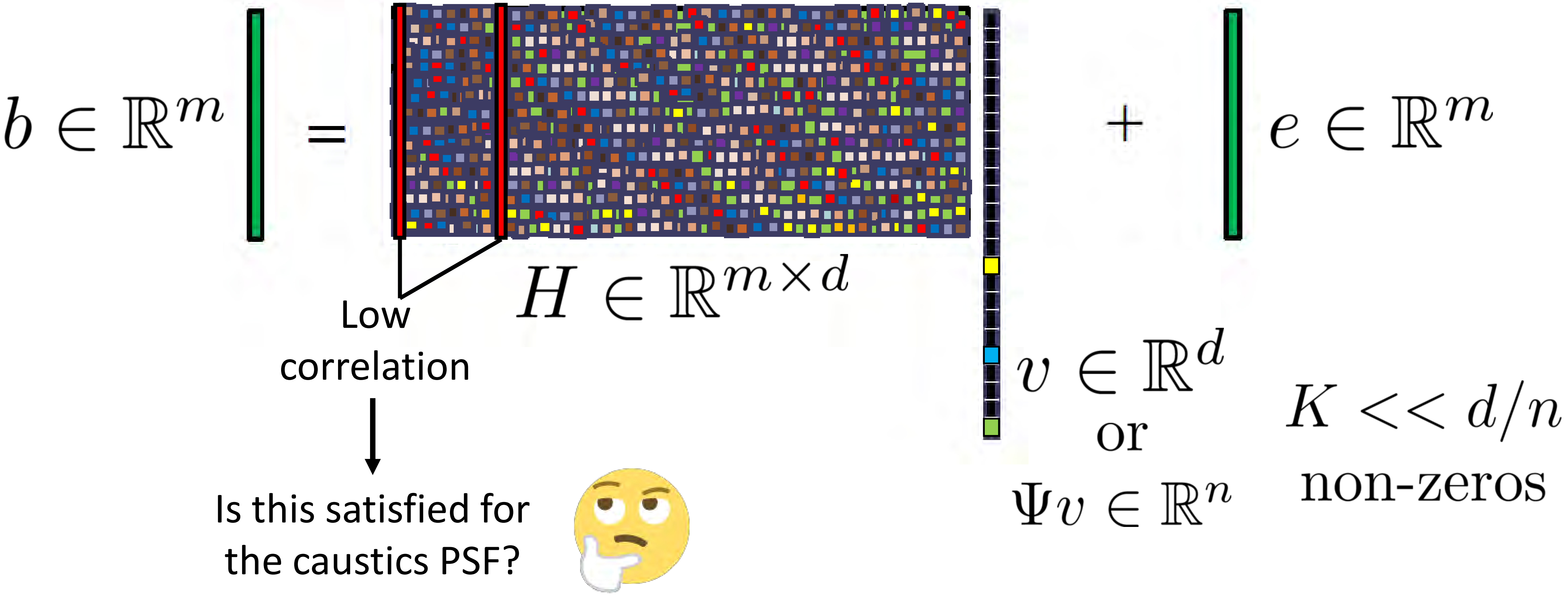
Noise

Compressed sensing: nonlinear recovery

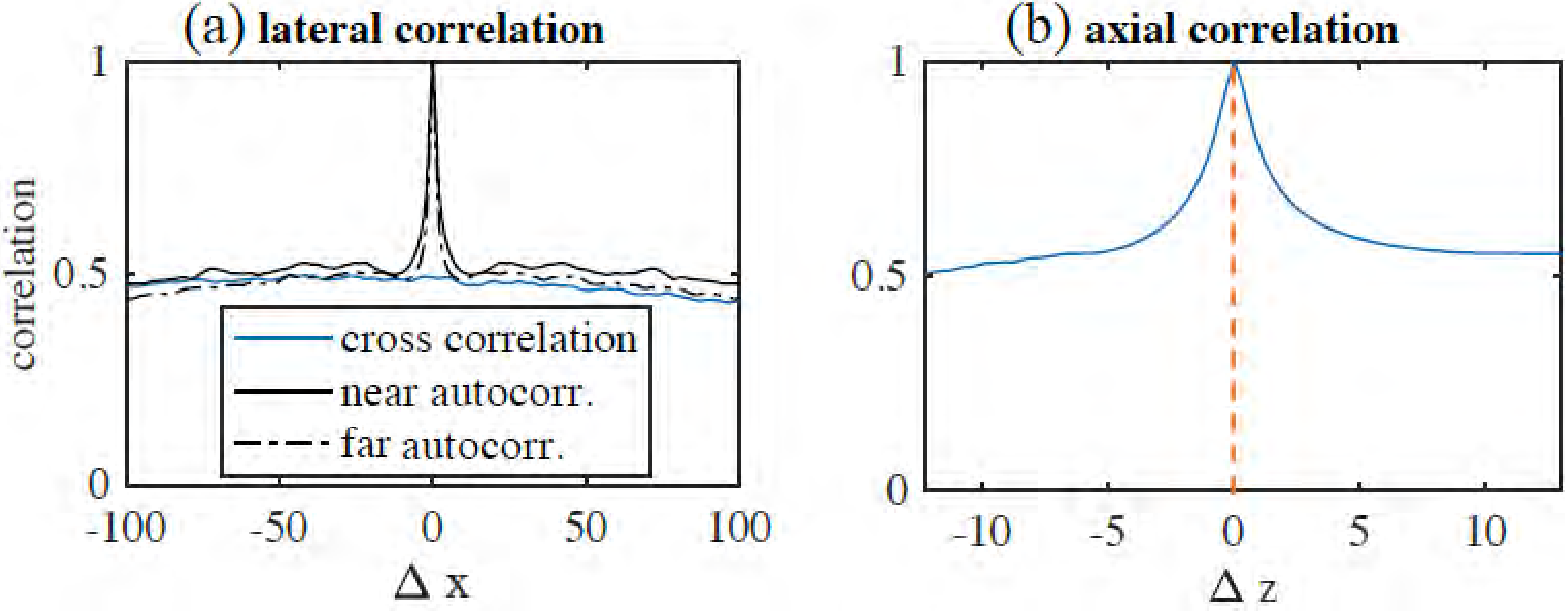


$$\hat{v}_{TV} = argmin_{v \geq 0} \left\{ \frac{1}{2} \|b - Hv\|^2 + \lambda \|\Psi v\|_1 \right\}$$

Compressed sensing: nonlinear recovery



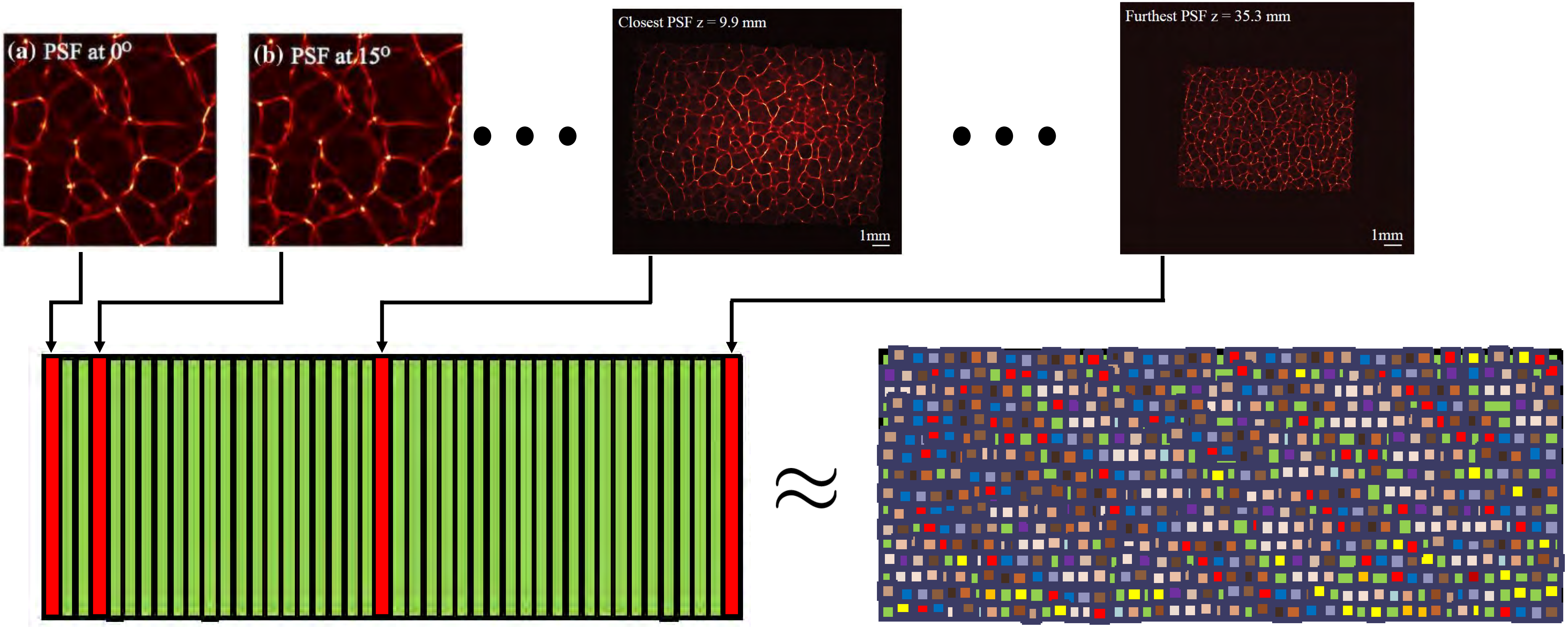
Caustics are approximately not correlated laterally/axially



Caustics at a given depth are unique over shifting

Caustics from two different depths are not similar, even under translation

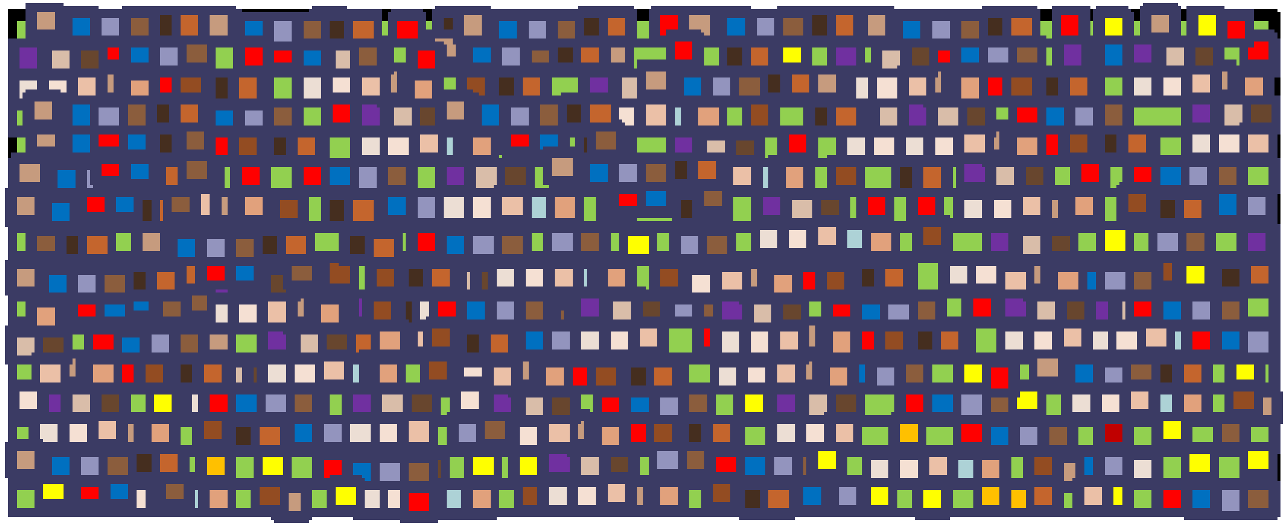
Caustics forward operator



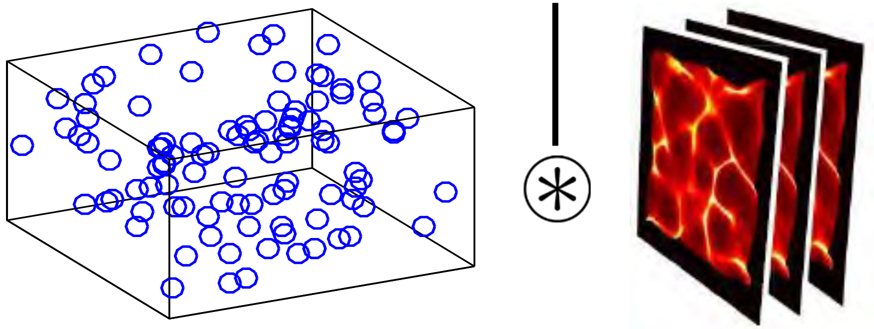
$$H \in \mathbb{R}^{m \times d}$$

<u>Question:</u> How can we handle this computationally for large volumes?

Convolutional implementation



↓ Slice-wise Correlation



$$\Sigma_z \left\{ \mathbf{v}(x, y, z) \circledast \mathcal{P}_z(x, y) \right\}$$

$$\mathbf{b}(x, y) =$$

$$\Sigma_z \left[\mathbf{v}(-x, -y, z) \overset{(x,y)}{*} \mathcal{P}(x, y; z) \right] =$$

$$\left[\mathbf{v}(-x, -y, z) \overset{(x,y,z)}{*} \mathcal{P}(x, y; -z) \right]_{z=0}$$

<u>Question:</u> How can we implement this efficiently?

<u>Answer:</u> Use 3D FFT!

(All of this is true up to some cropping operator)

Optimization trick

- Variable splitting:

$$\hat{v}_{MAP} = argmin_v \left\{ \frac{1}{2\sigma_n^2} \|b - Hv\|^2 + \lambda p(v) \right\}$$

$$(\hat{v}, \hat{z}) = argmin_{v,z} \left\{ \frac{1}{2\sigma_n^2} \|b - Hv\|^2 + \lambda p(z) \; subject \; to \; v = z \right\}$$

- Solve with Half Quadratic Splitting (or Alternating Direction Method of Multipliers)

$$(\hat{v}, \hat{z}) = argmin_{v,z} \left\{ \frac{1}{2\sigma_n^2} \|b - Hv\|^2 + \lambda p(z) + \frac{\mu}{2} \|v - z\|^2 \right\}$$

- Update v: $v^{k+1} = argmin_v \left\{ \frac{1}{2\sigma_n^2} \|b - Hv\|^2 + \frac{\mu}{2} \|v - z^k\|^2 \right\}$
- Update z: $z^{k+1} = argmin_z \left\{ \lambda p(z) + \frac{\mu}{2} \|v^{k+1} - z\|^2 \right\}$
- Update μ: $\mu_{k+1} = \gamma \mu_k$

In the paper:

$$p(v) = \|\Psi v\|_1 = TV_{3D}(v)$$

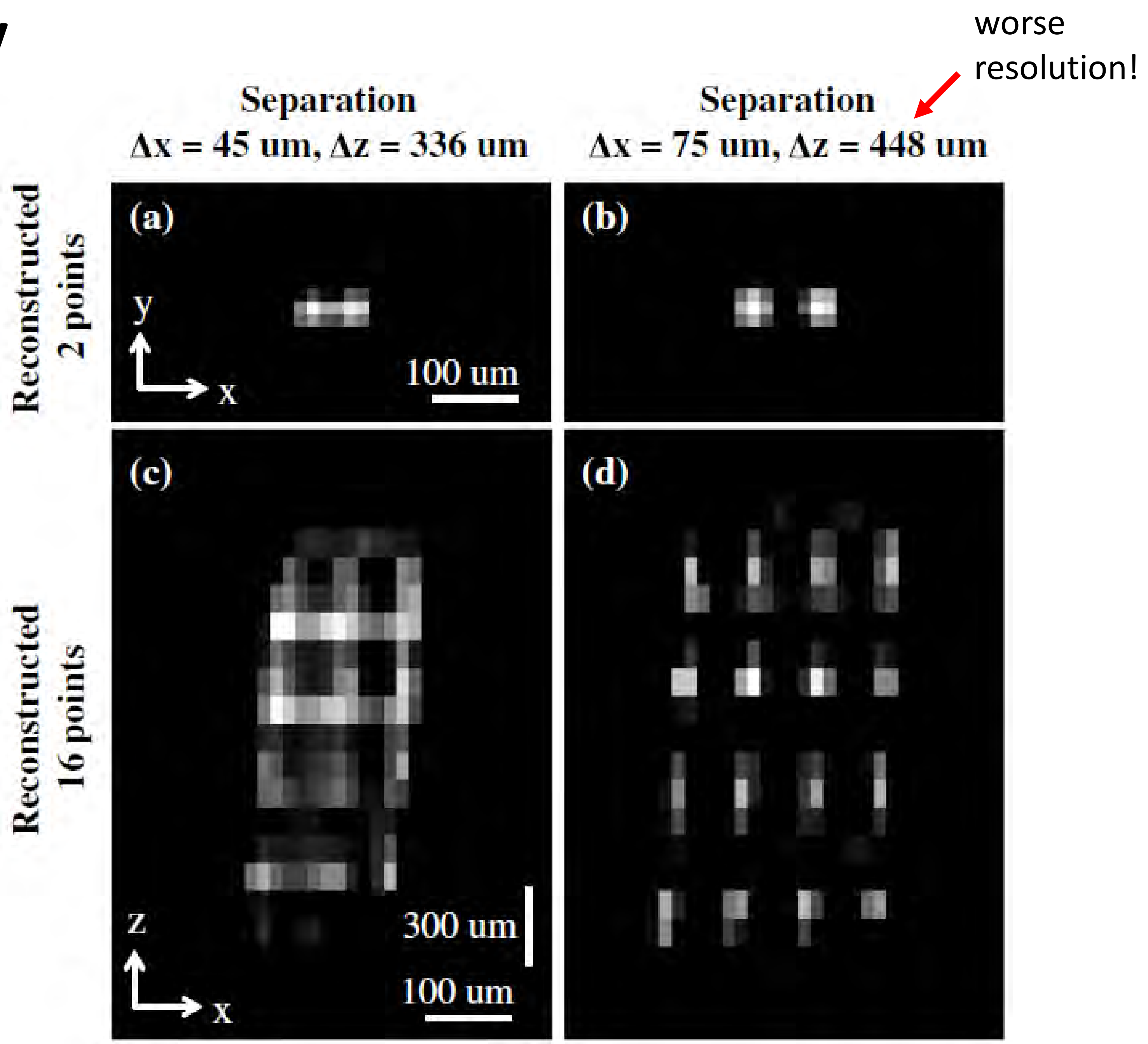
Bouman et. Al [2013]

Outline

- Depth-encoding PSF
- Nonlinear CS rec.
- Resolution analysis
- Experimental results and extensions

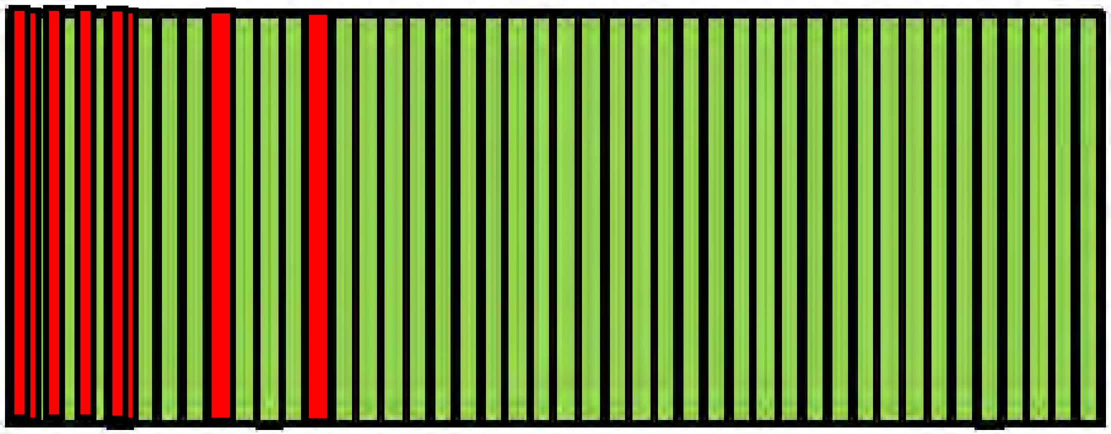
Two-point distinguishability

- Unlike typical cameras, in computational cameras performance depend on scene complexity.
- Two-point distinguishability is not a good metric.
- Non-isotropic resolution



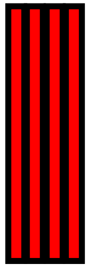
Local condition number

- Main idea: Define resolution through invertibility of the forward model H



$$H \in \mathbb{R}^{m \times d}$$

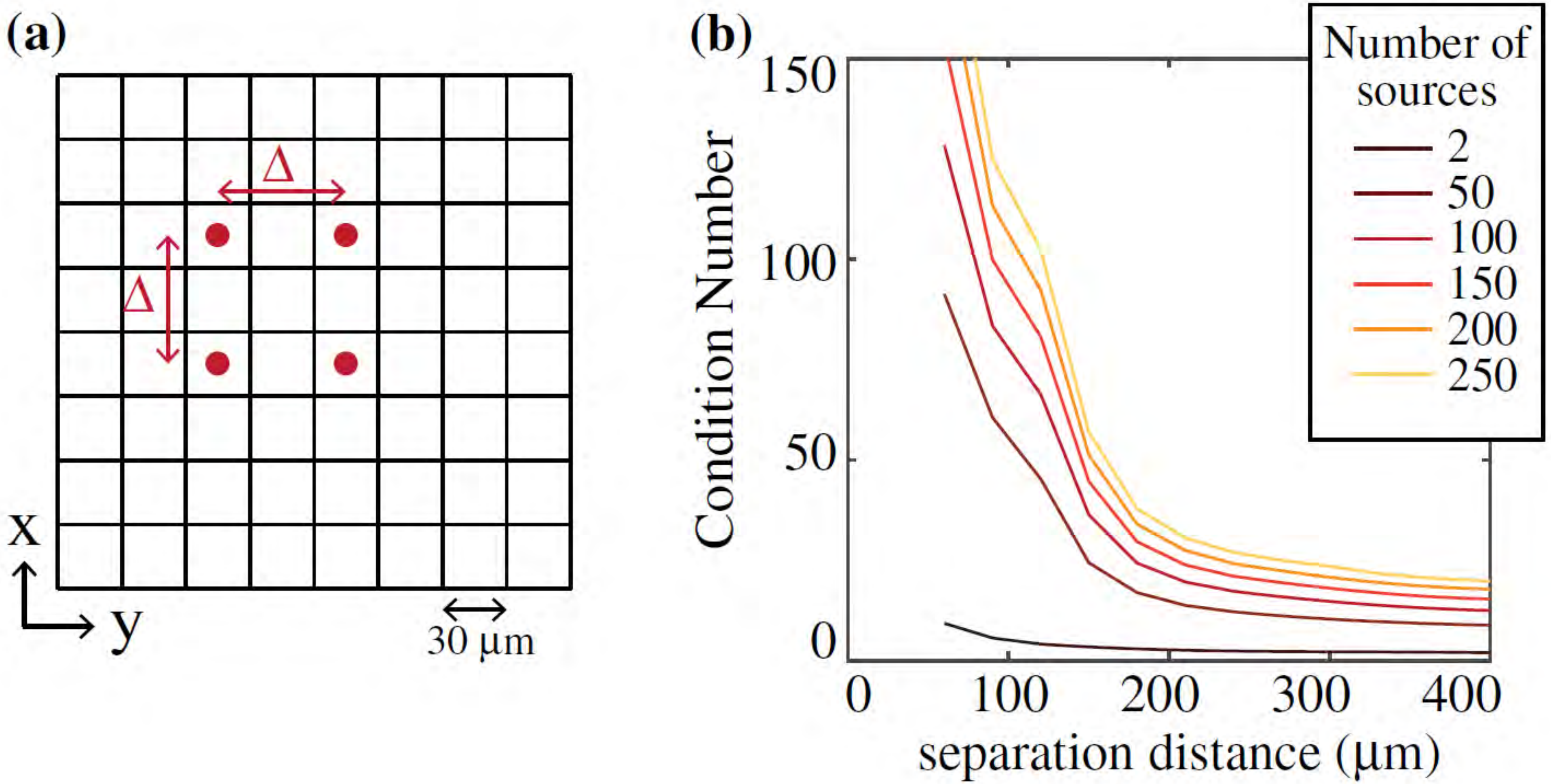
- Oracle support assumption:



$$\rightarrow \kappa\left(H_s\right) = \frac{\sigma_{max}(H_s)}{\sigma_{min}(H_s)}$$

$$H_s \in \mathbb{R}^{m \times s}$$

Local condition number



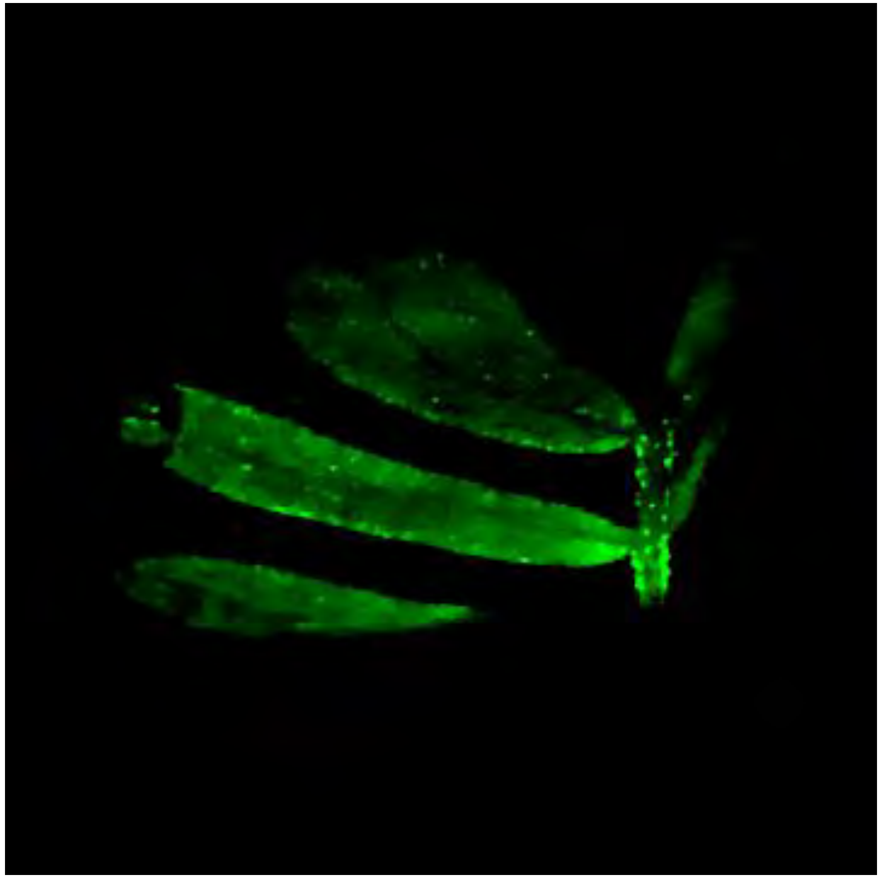
- Higher condition number = lower resolution

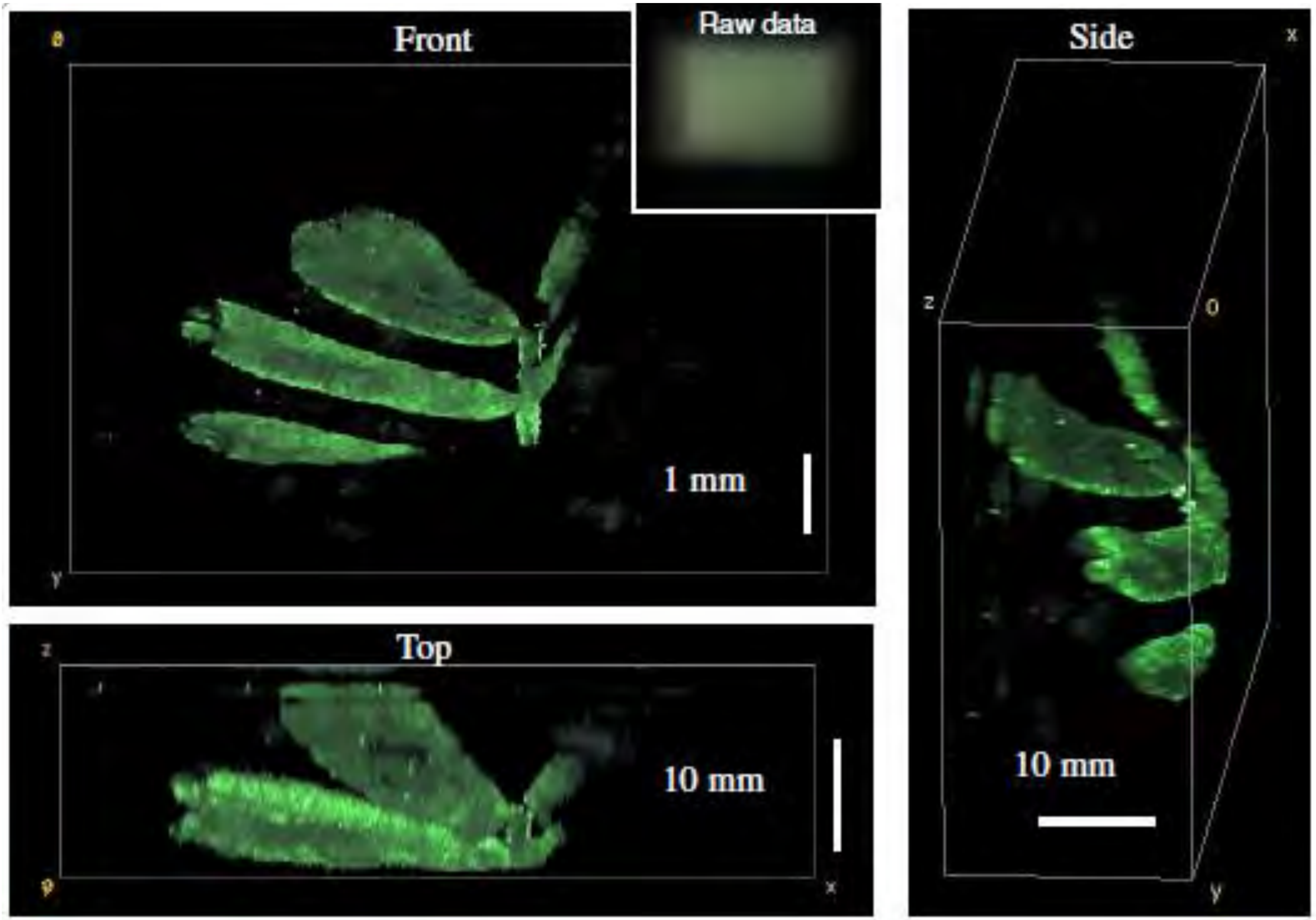
Outline

- 👍 Depth-encoding PSF
- 👍 Nonlinear CS rec.
- 👍 Resolution analysis
- 👉 Experimental results and extensions

Experimental 3D reconstruction from a snapshot

480x320x128 voxels reconstructed in ~3 mins

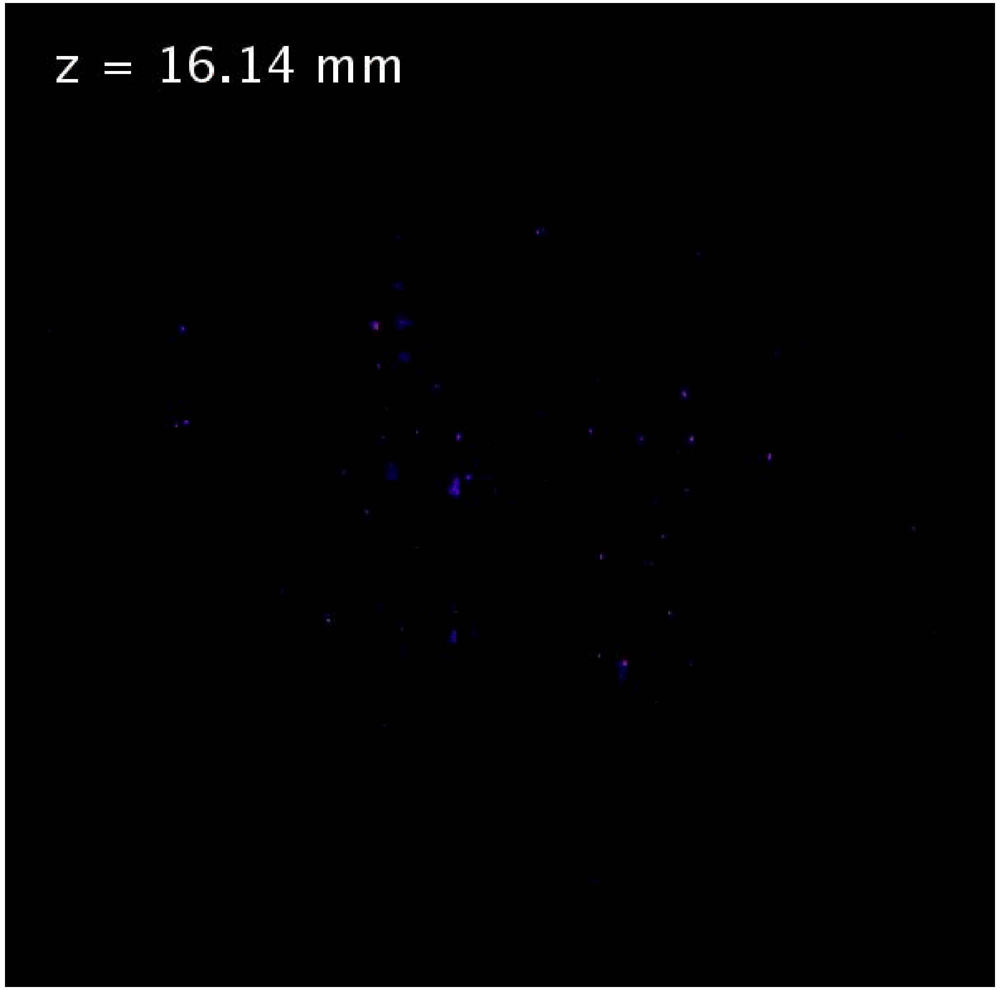


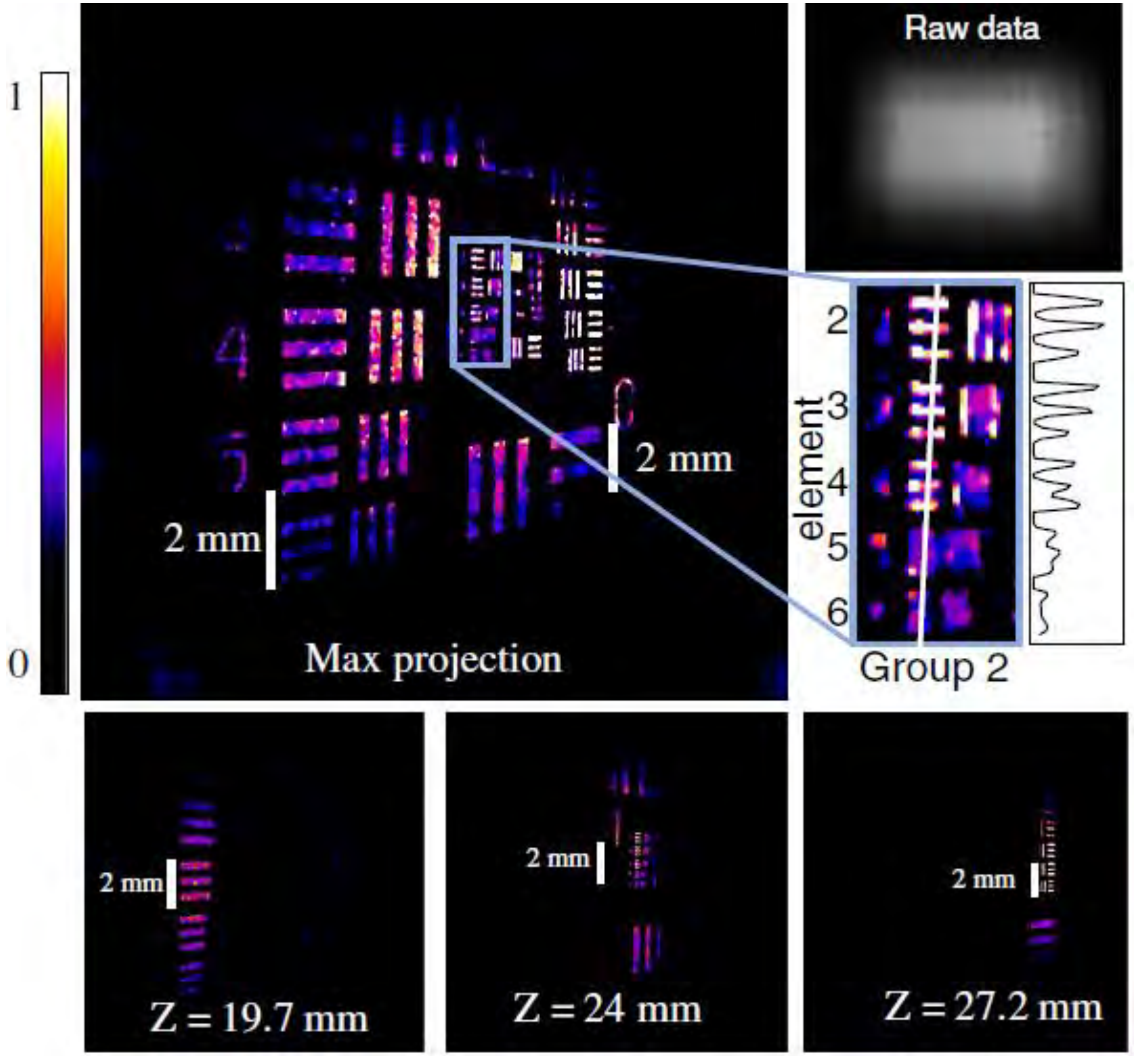


Antipa et al. Optica 5, 1-9 (2018)

Experimental 3D reconstruction from a snapshot

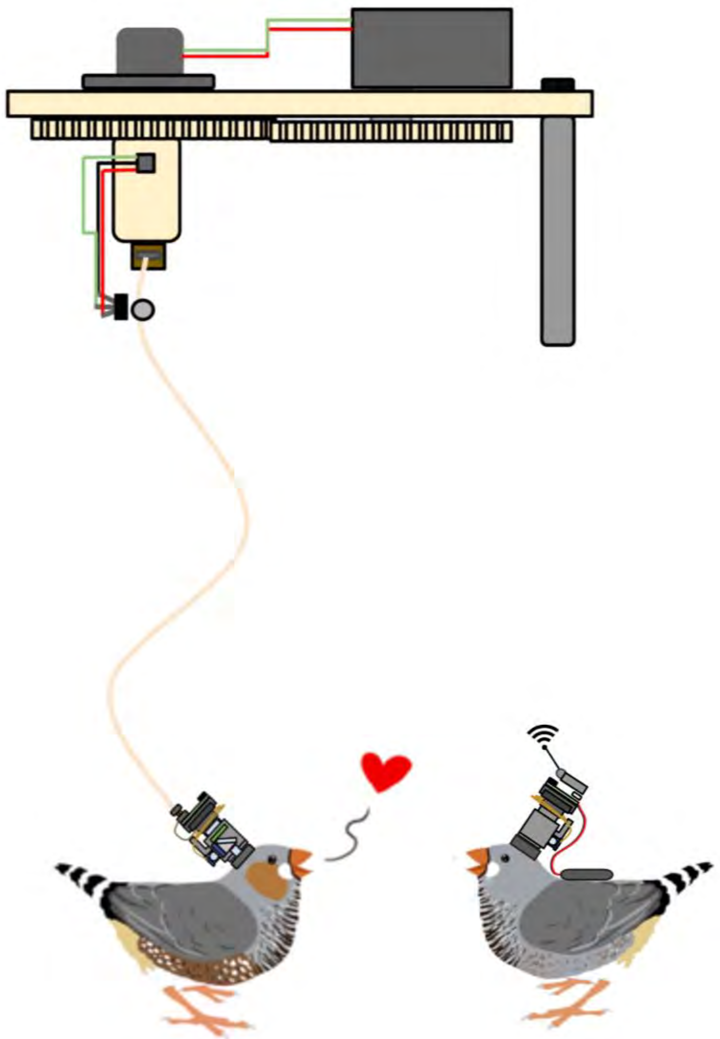
640x640x50 voxels reconstructed in ~3 mins



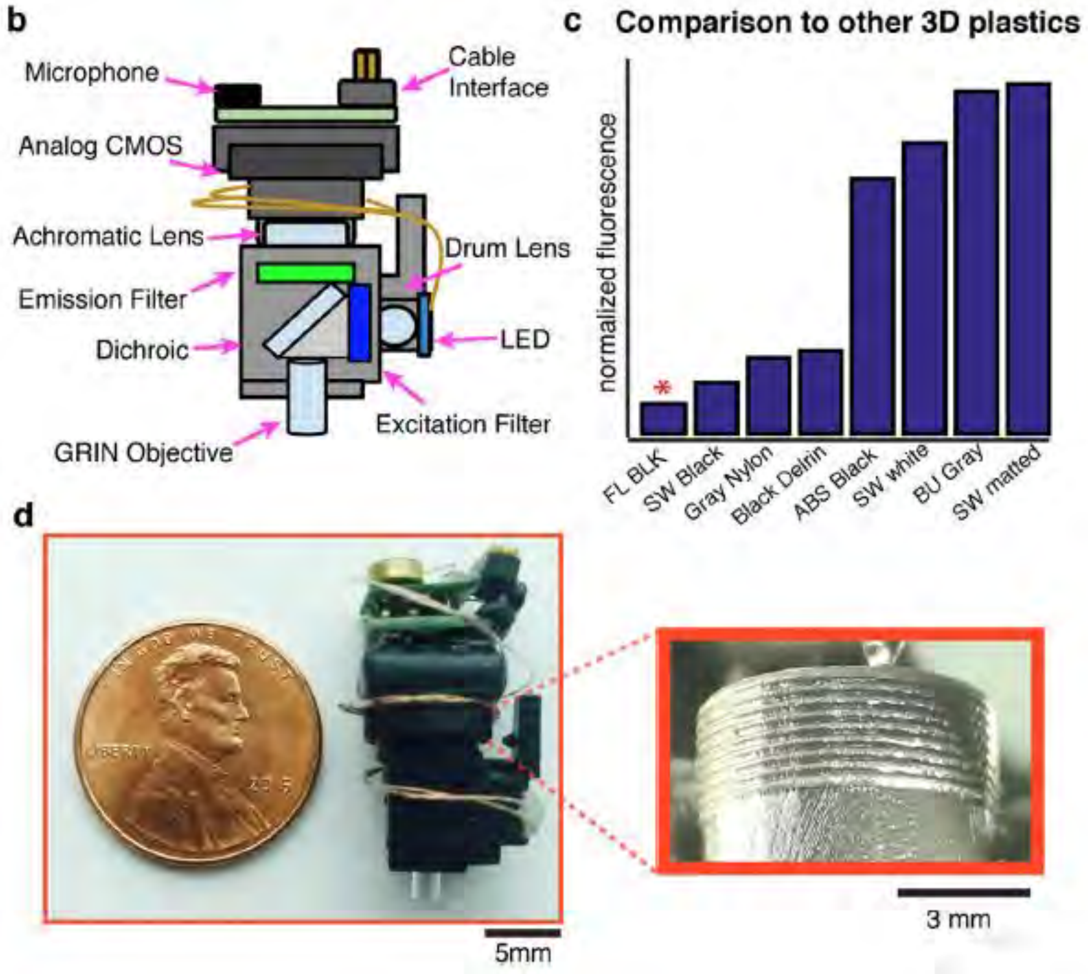


Antipa et al. Optica 5, 1-9 (2018)

What is a Miniscope? http://miniscope.org/

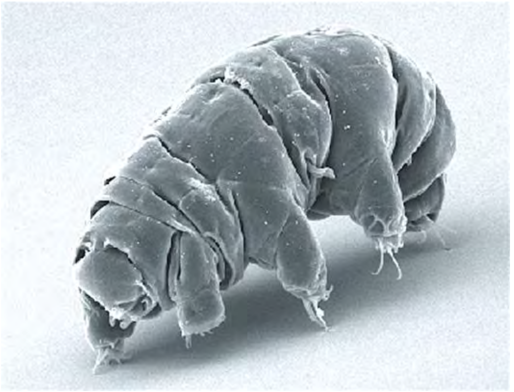


https://github.com/gardner-lab/FinchScope

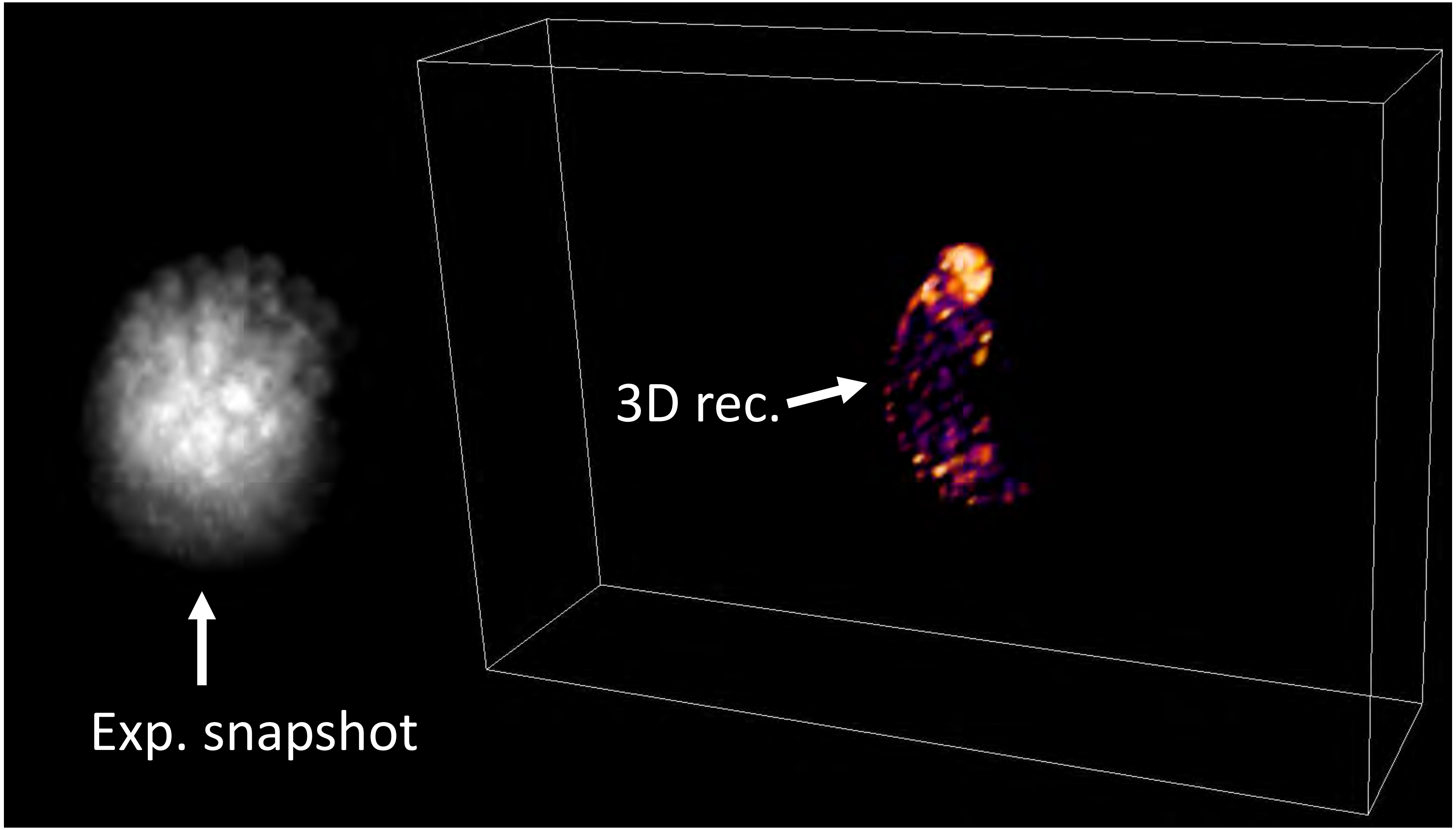


Liberti III et al., J. Neural Eng. 14 (2017)

Freely moving tardigrades



SEM image



Yanny et al. Light: Science & Applications (2020) 9:171

References

Nick Antipa, Sylvia Necula, Ren Ng, and Laura Waller. "Single-shot diffuser-encoded light field imaging." In 2016 IEEE International Conference on Computational Photography (ICCP), pp. 1-11. IEEE, 2016.

Nick Antipa, Grace Kuo, Reinhard Heckel, Ben Mildenhall, Emrah Bostan, Ren Ng, and Laura Waller, "DiffuserCam: lensless single-exposure 3D imaging." ***Optica*** **5, 1-9 (2018).**

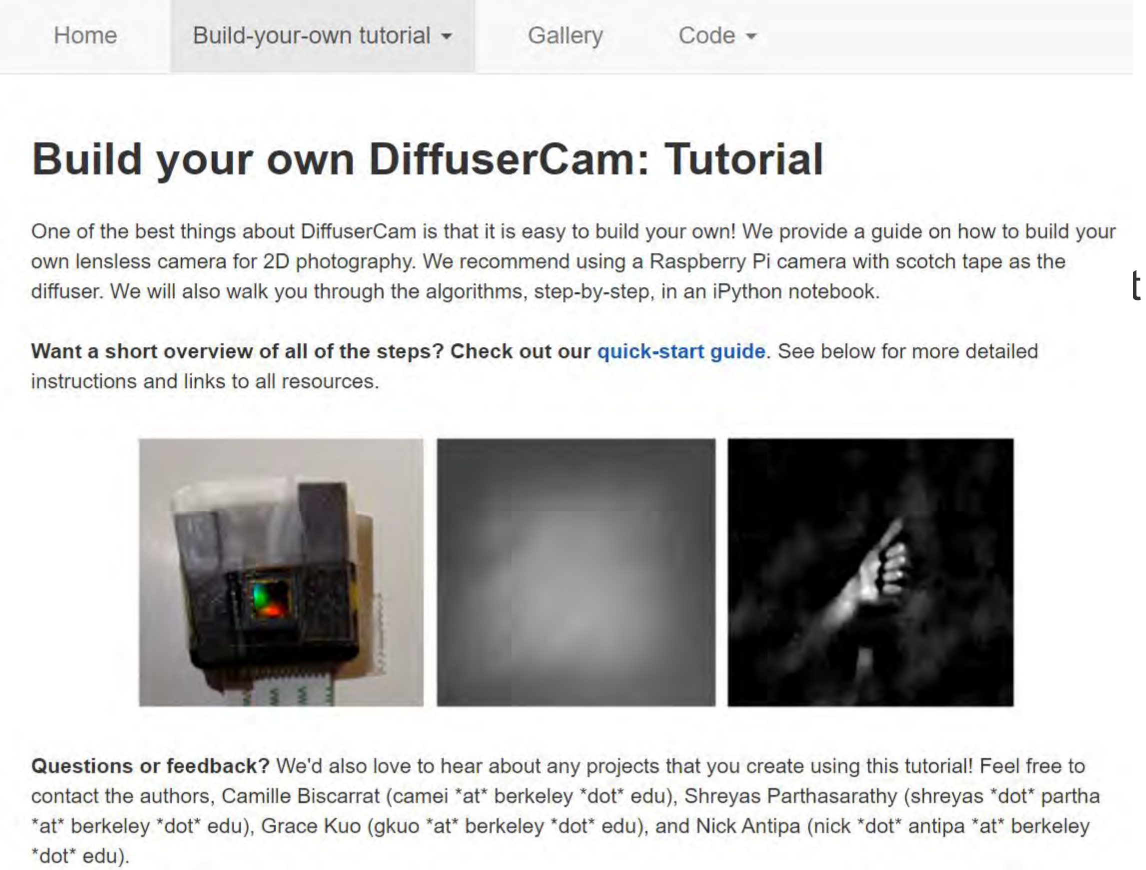
Nick Antipa, Patrick Oare, Emrah Bostan, Ren Ng, and Laura Waller. "Video from stills: Lensless imaging with rolling shutter." In 2019 IEEE International Conference on Computational Photography (ICCP), pp. 1-8. IEEE, (2019)

Kyrollos Yanny*, Nick Antipa*, William Liberti, Sam Dehaeck,Kristina
Laura Waller. Miniscope3D: optimized single-shot miniature 3D fluor

Grace Kuo, Fanglin Linda Liu, Irene Grossrubatscher, Ren Ng, and l
random microlens diffuser," Opt. Express 28, 8384-8399 (2020)

Fanglin Linda Liu, Grace Kuo, Nick Antipa, Kyrollos Yanny, and Laur
field microscopy with a diffuser," Opt. Express 28, 28969-28986 (202

Kristina Monakhova*, Kyrollos Yanny*, Neerja Aggarwal, and Laura \
hyperspectral imaging with a spectral filter array," Optica 7, 1298-13



הטכניון
מכון טכנולוגי לישראל

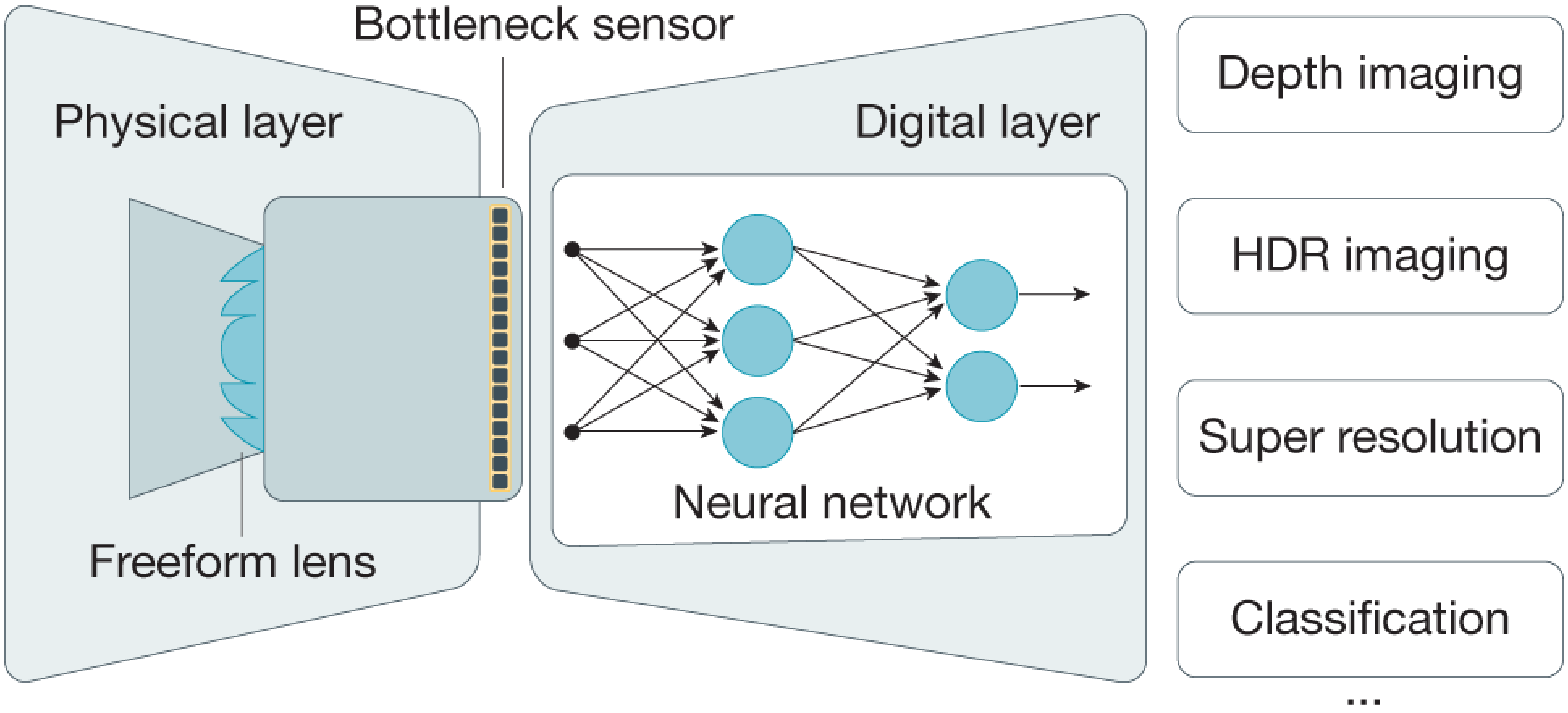
Department of Biomedical Engineering, Technion

Computational optical imaging 336547

Tutorial 11+12 – Deep Optics

Elias Nehme & Yoav Shechtman

12 January 2021



Wetzstein et al. Nature **588**, p. 39–47 (2020)

Computational Imaging = Co-design of acquisition + computation

Peng et al., ACM SIGGRAPH (2020)

Computational Imaging = Co-design of acquisition + computation

HDR Imaging

[Mann,Devebec,Nayar,...]

Super-resolution

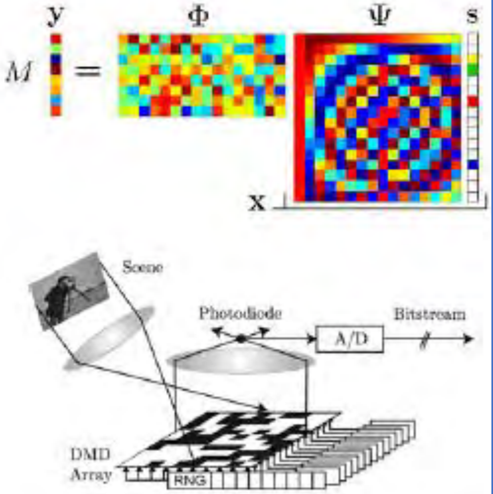
[Baker,Ben-Ezra,...]

EDOF

[Dowski,Nayar,...]

Light Fields

[Levoy,...]



Compressive Imaging

[Baraniuk,...]

optics

+

sensing

+

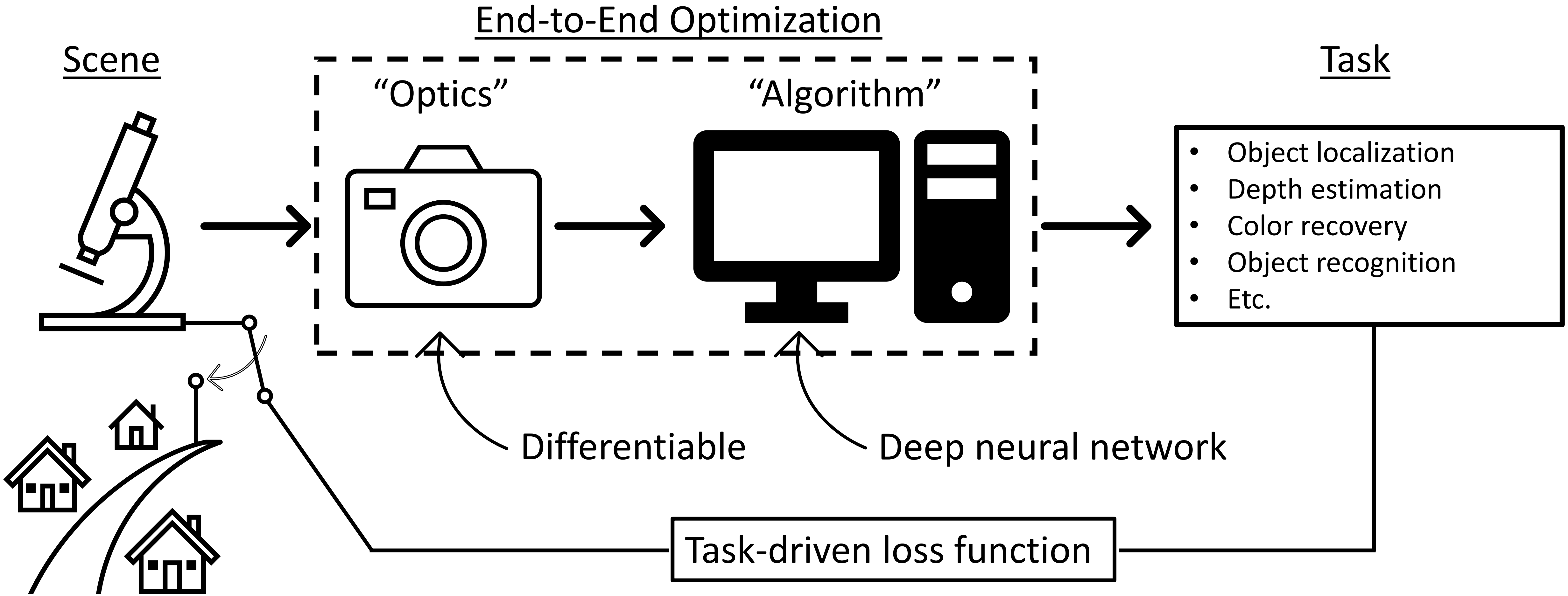
computation

Computational Imaging

Tutorial 5+6, and HW1

Peng et al., ACM SIGGRAPH (2020)

Deep Computational Cameras = “Deep Optics” = “Neural Sensors”



Outline

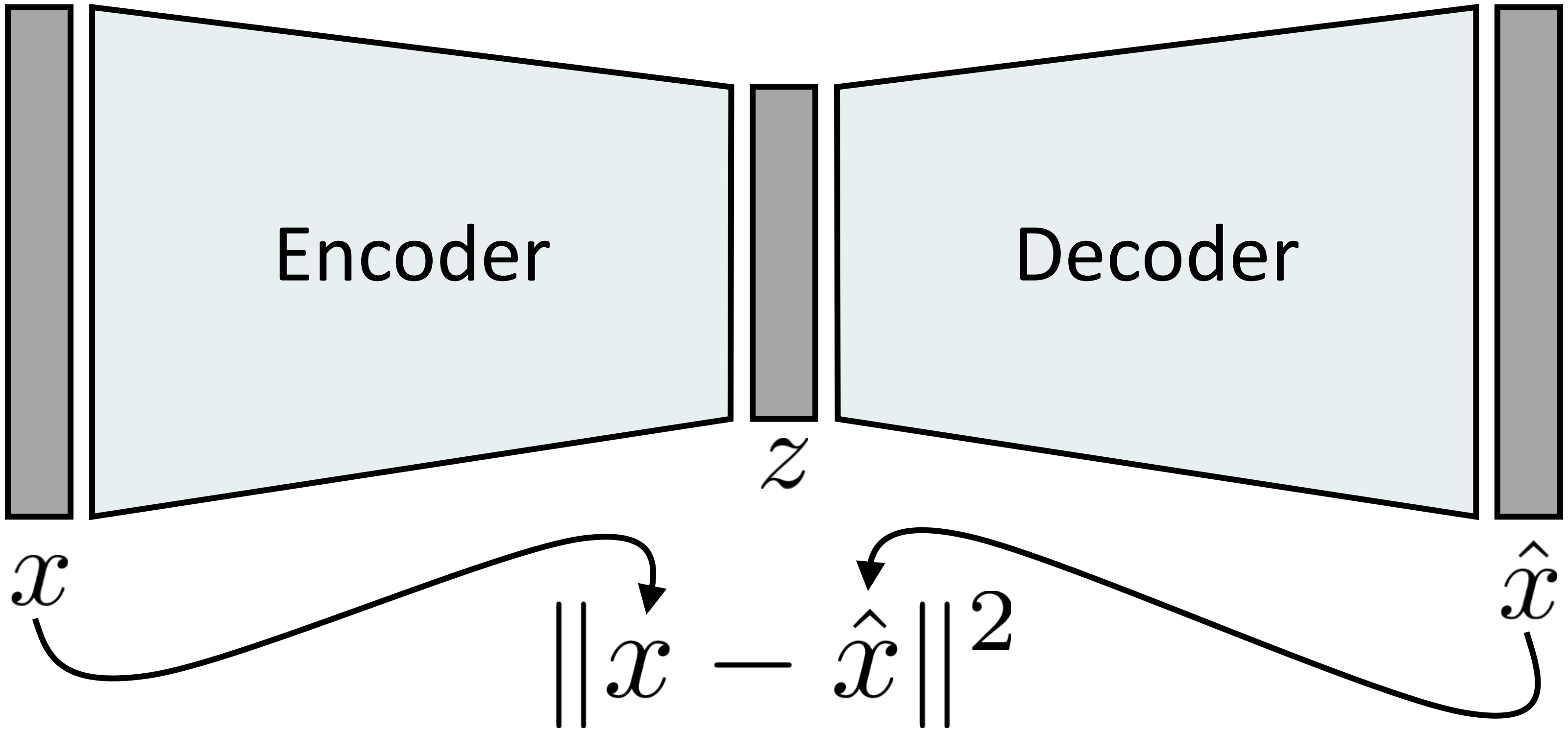
- ►Autoencoder interpretation
- ►Learning dense 3D imaging
- ►Generality to higher level tasks
- ►Multi-measurement systems
- ►Beyond microscopy

Outline

- ☛ Autoencoder interpretation
- ▶ Learning dense 3D imaging
- ▶ Generality to higher level tasks
- ▶ Multi-measurement systems
- ▶ Beyond microscopy

Autoencoders: Background

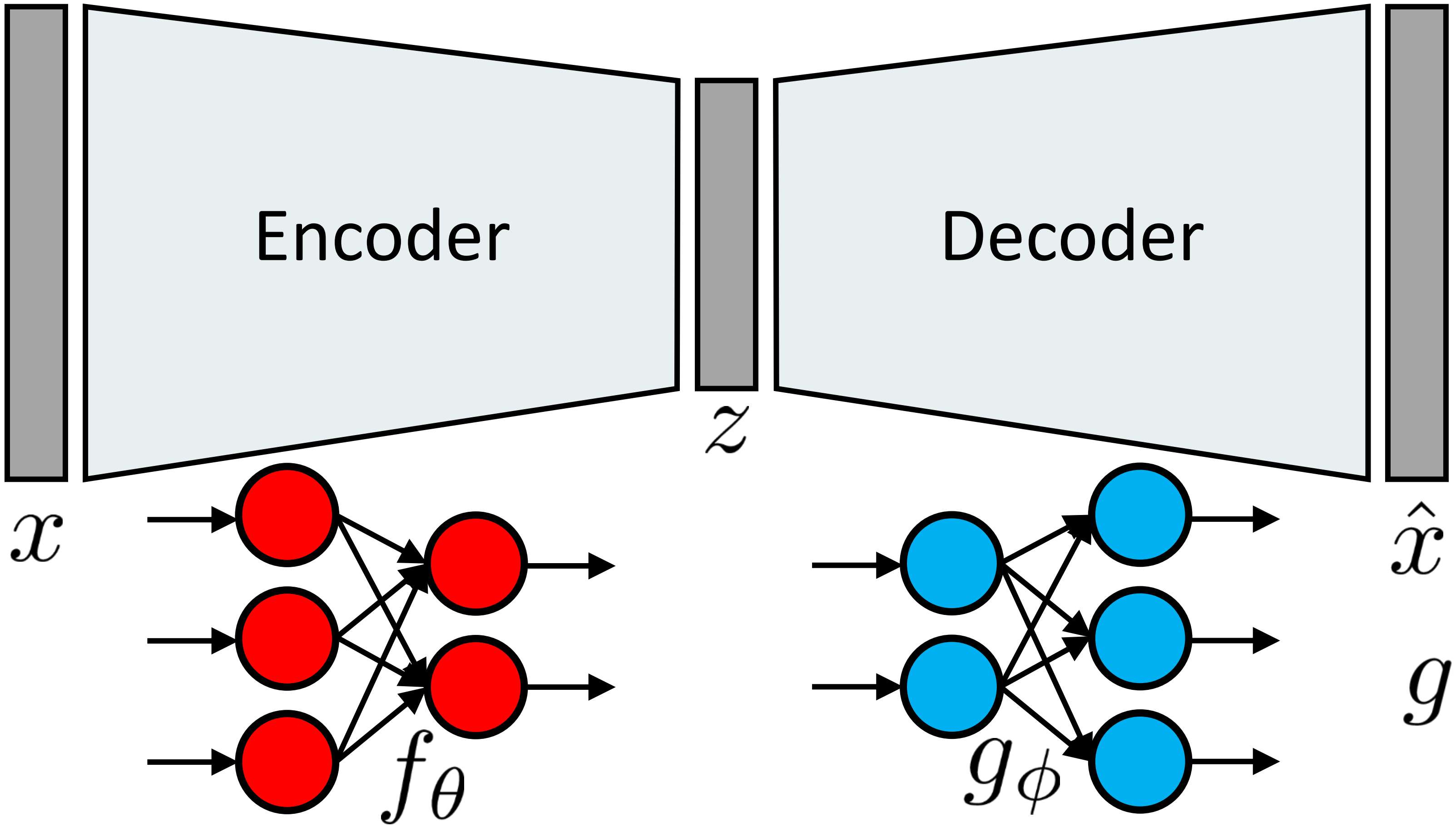
- Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data



- Train such that features can be used to reconstruct original data

Autoencoders: Background

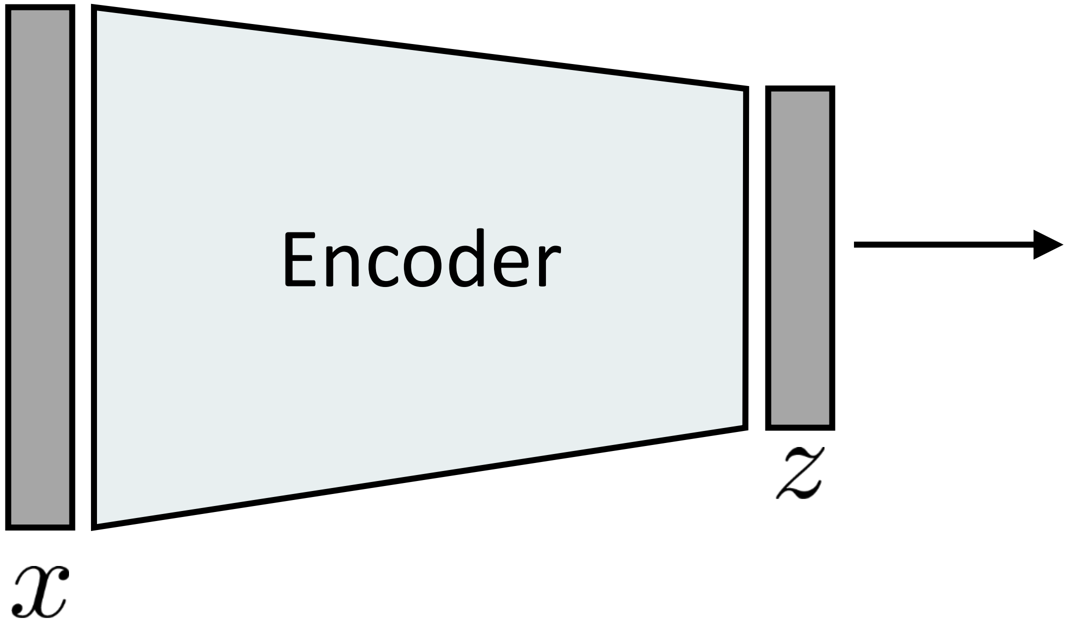
- Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data



$$g_\phi \approx f_\theta^{-1}$$

Autoencoders: Background

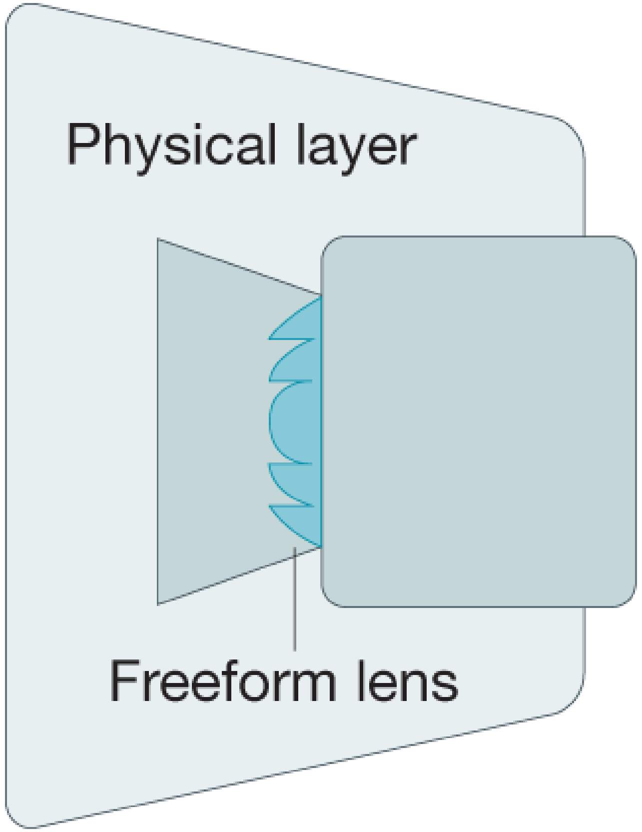
- Learned lower-dimensional representation can be used for classification



Use to train a classifier with limited training data:
Cat, Dog, Truck, Plane, etc.

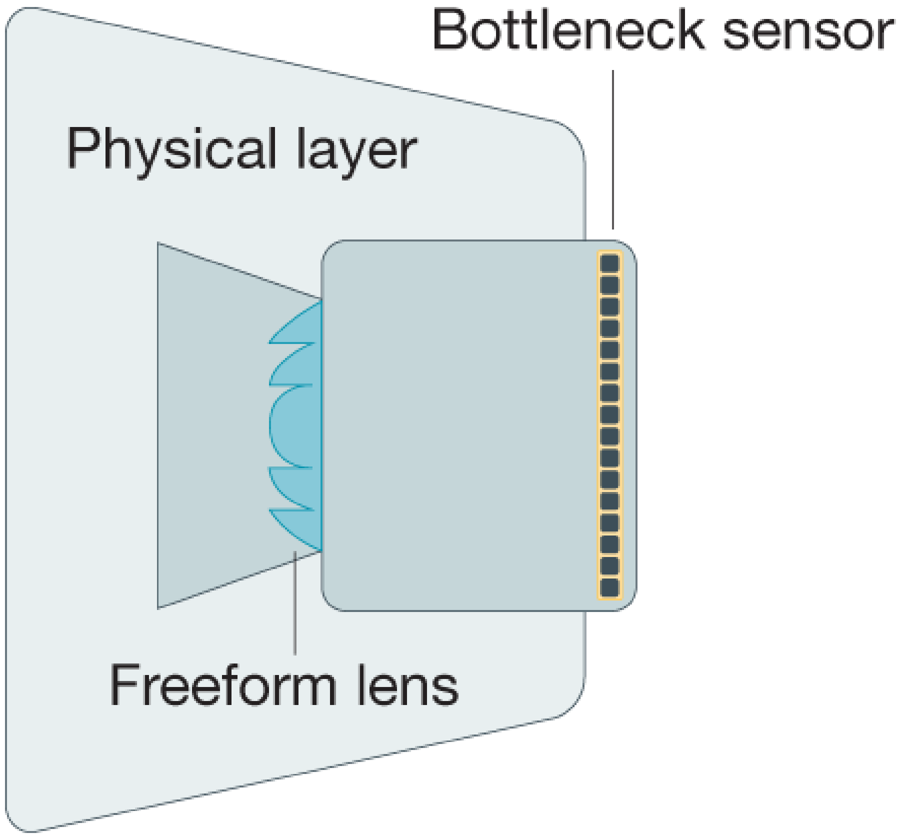
Example from Stanford CS231n

Autoencoder interpretation: Physical encoder – Electronic decoder



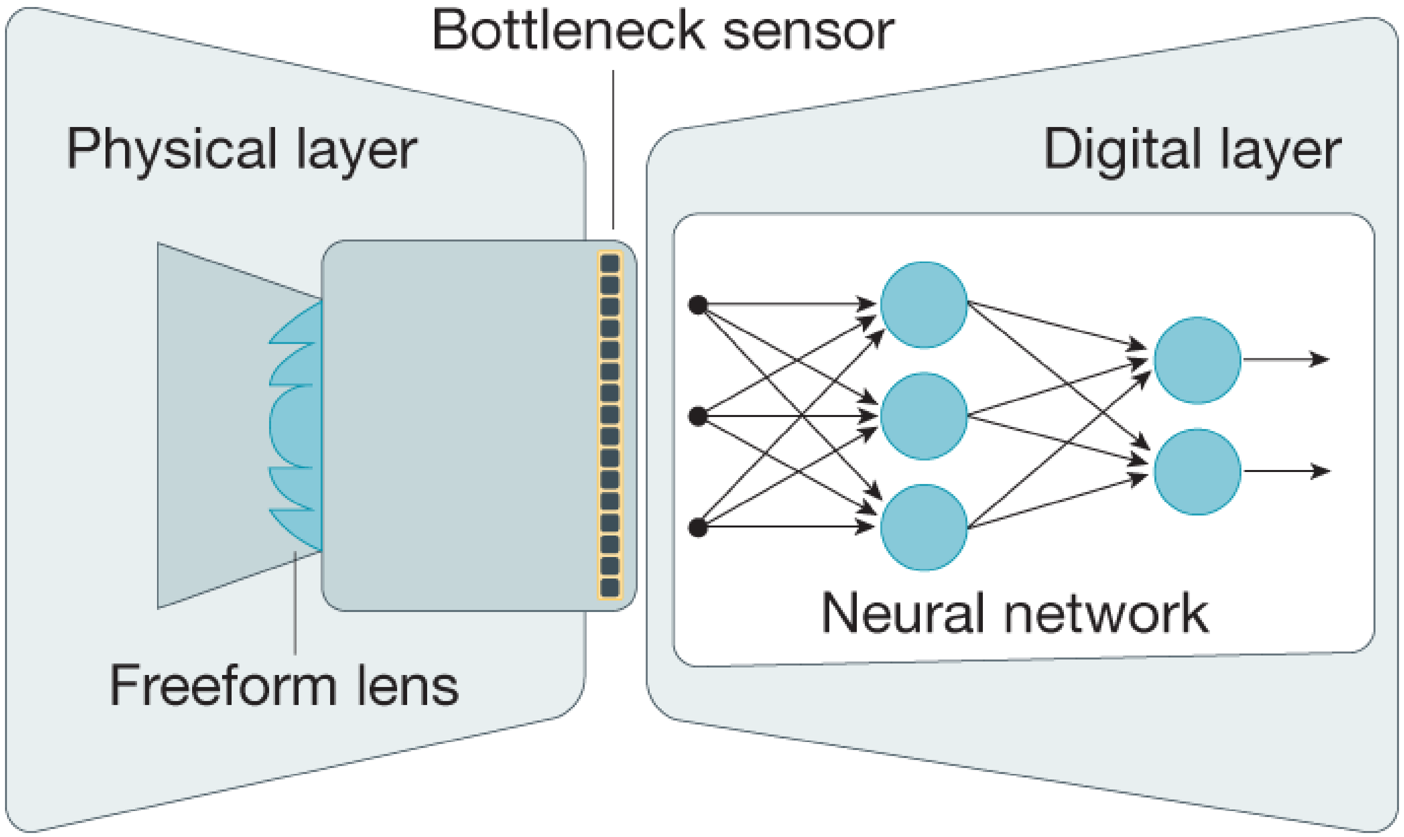
Wetzstein et al. Nature **588**, p. 39–47 (2020)

Autoencoder interpretation: Physical encoder – Electronic decoder



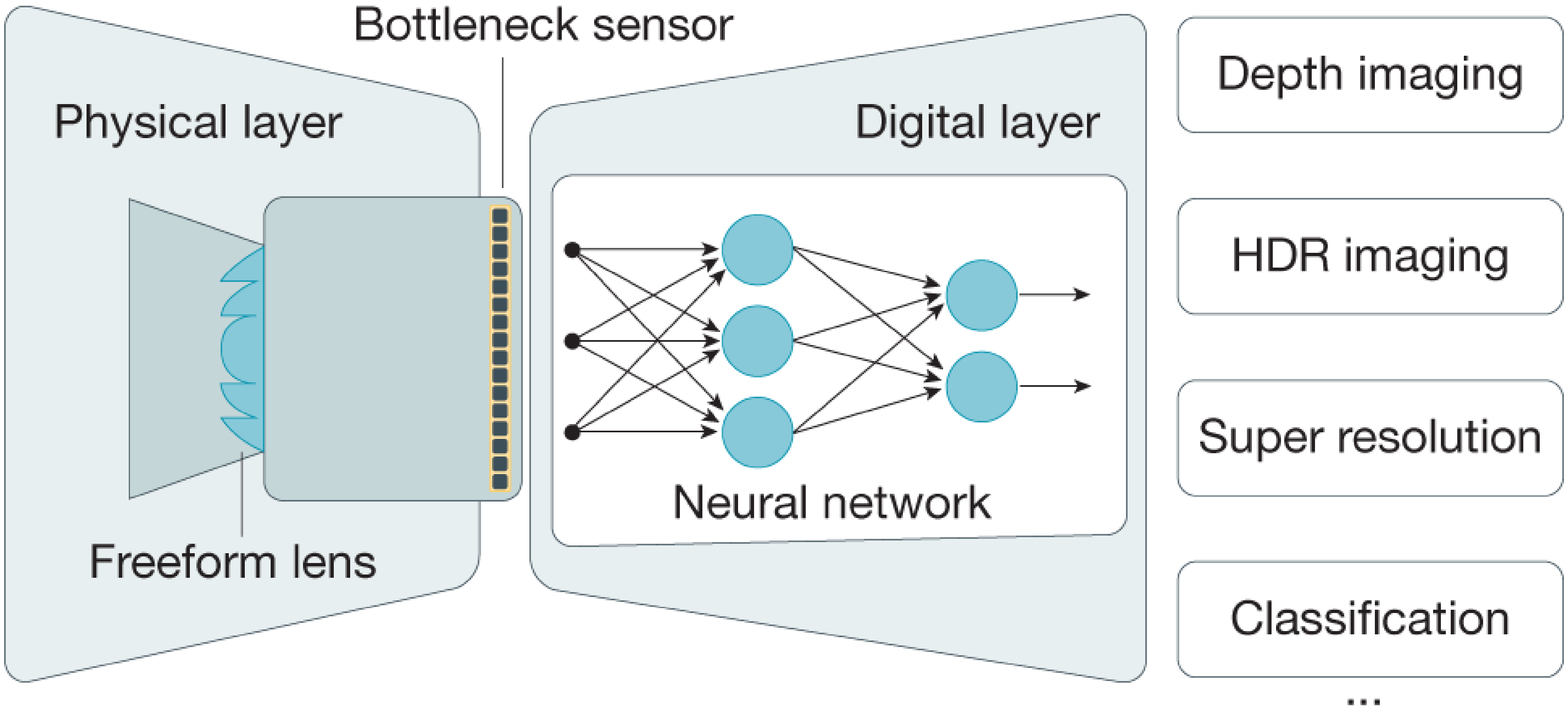
Wetzstein et al. Nature **588**, p. 39–47 (2020)

Autoencoder interpretation: Physical encoder – Electronic decoder



Wetzstein et al. Nature **588**, p. 39–47 (2020)

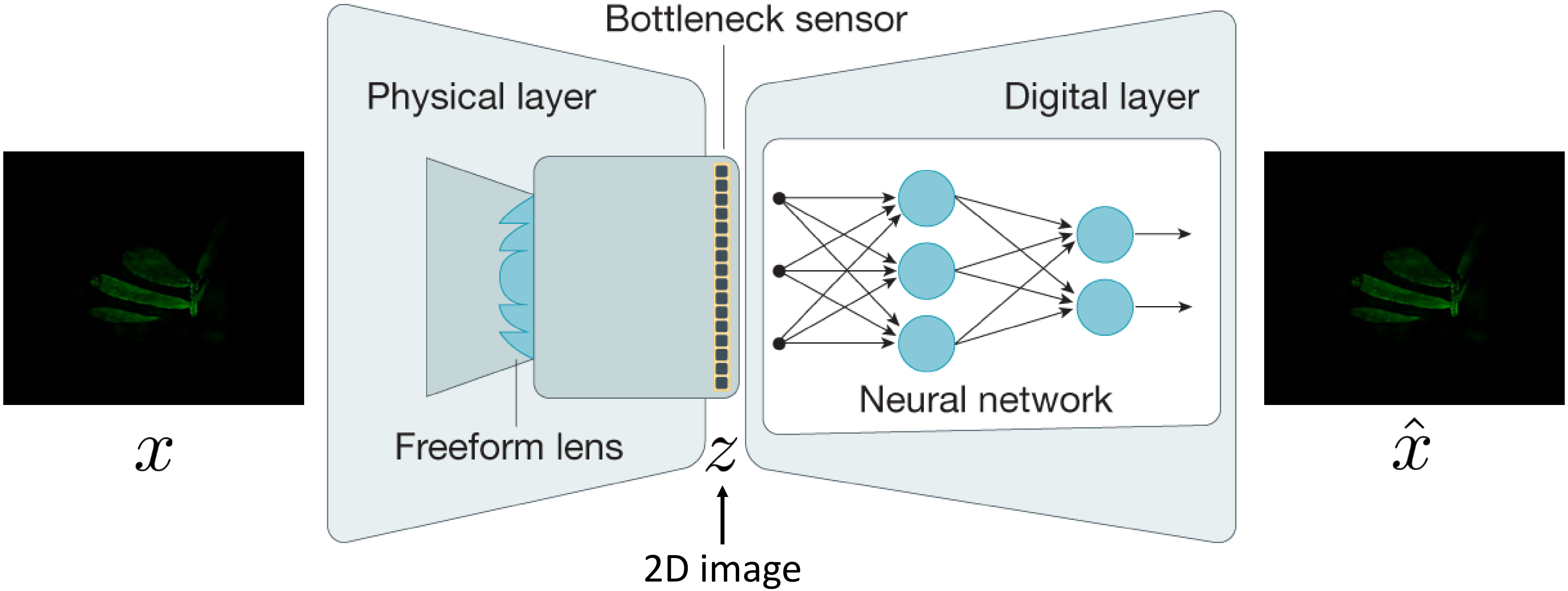
Autoencoder interpretation: Physical encoder – Electronic decoder



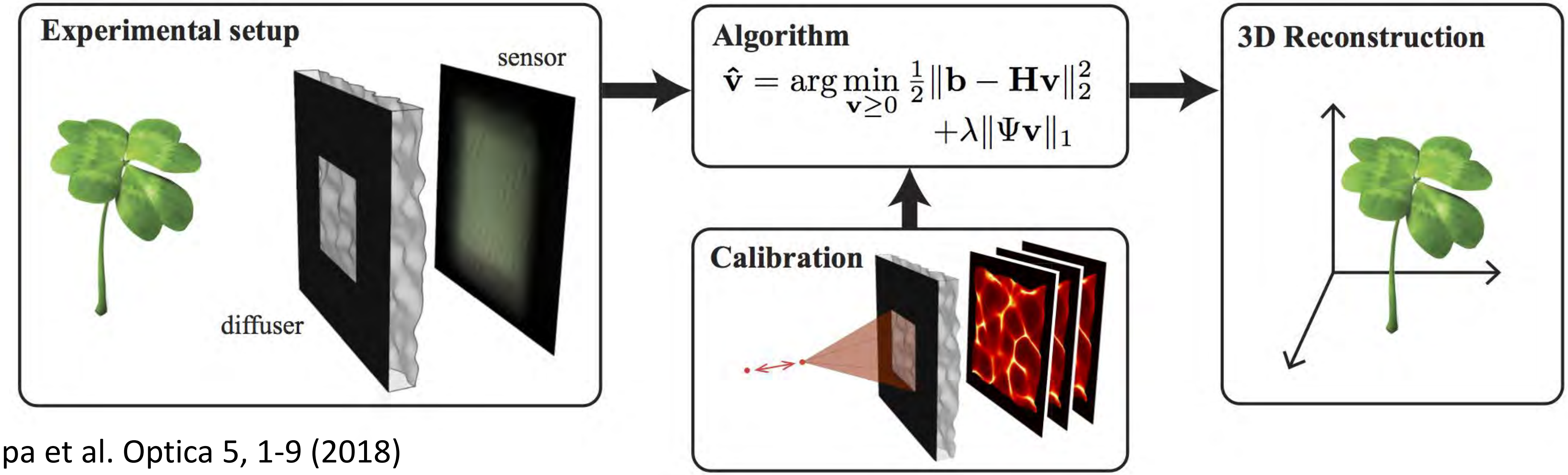
Wetzstein et al. Nature **588**, p. 39–47 (2020)

Autoencoder interpretation: Physical encoder – Electronic decoder

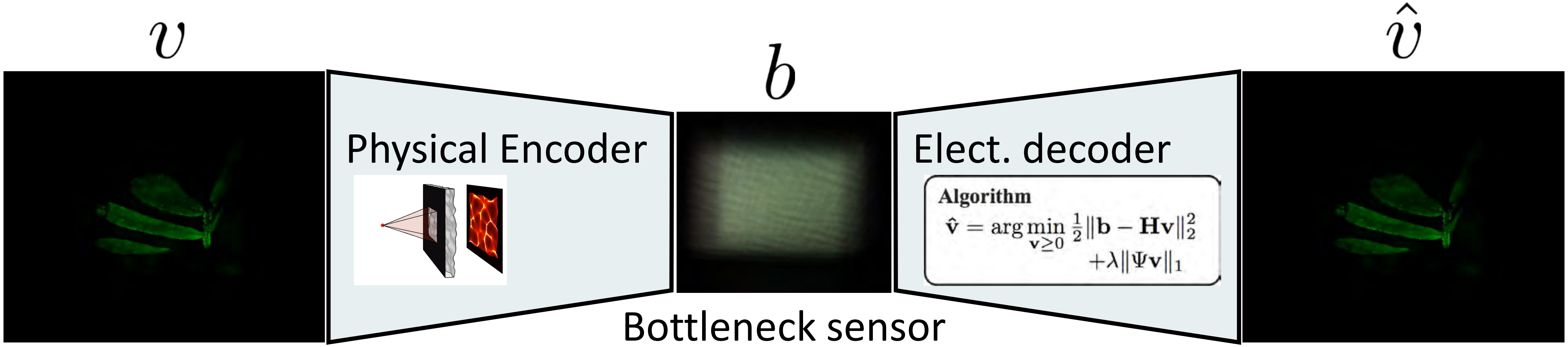
- For example, we can optimize the “Freeform” lens for depth imaging



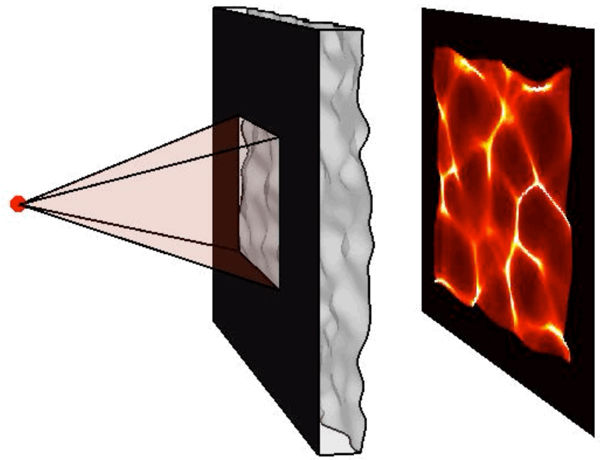
Analogy to DiffuserCam



Antipa et al. Optica 5, 1-9 (2018)

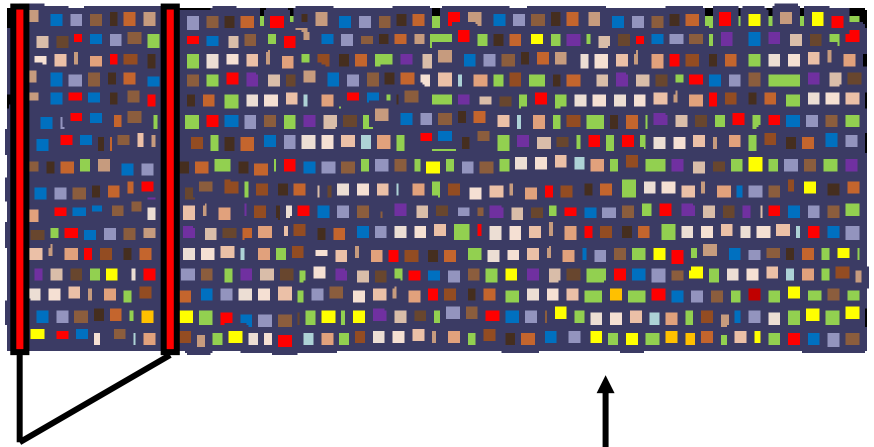


Main difference compared to DiffuserCam



Antipa et al. Optica 5, 1-9 (2018)

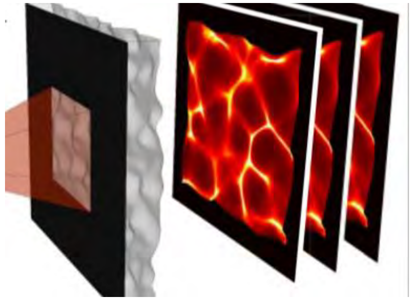
Chosen such that →



Low correlation

Optical measurement operator (forward model)

<u>Question 1:</u> How should we design the physical element if the algorithm is not based on compressed sensing theory?

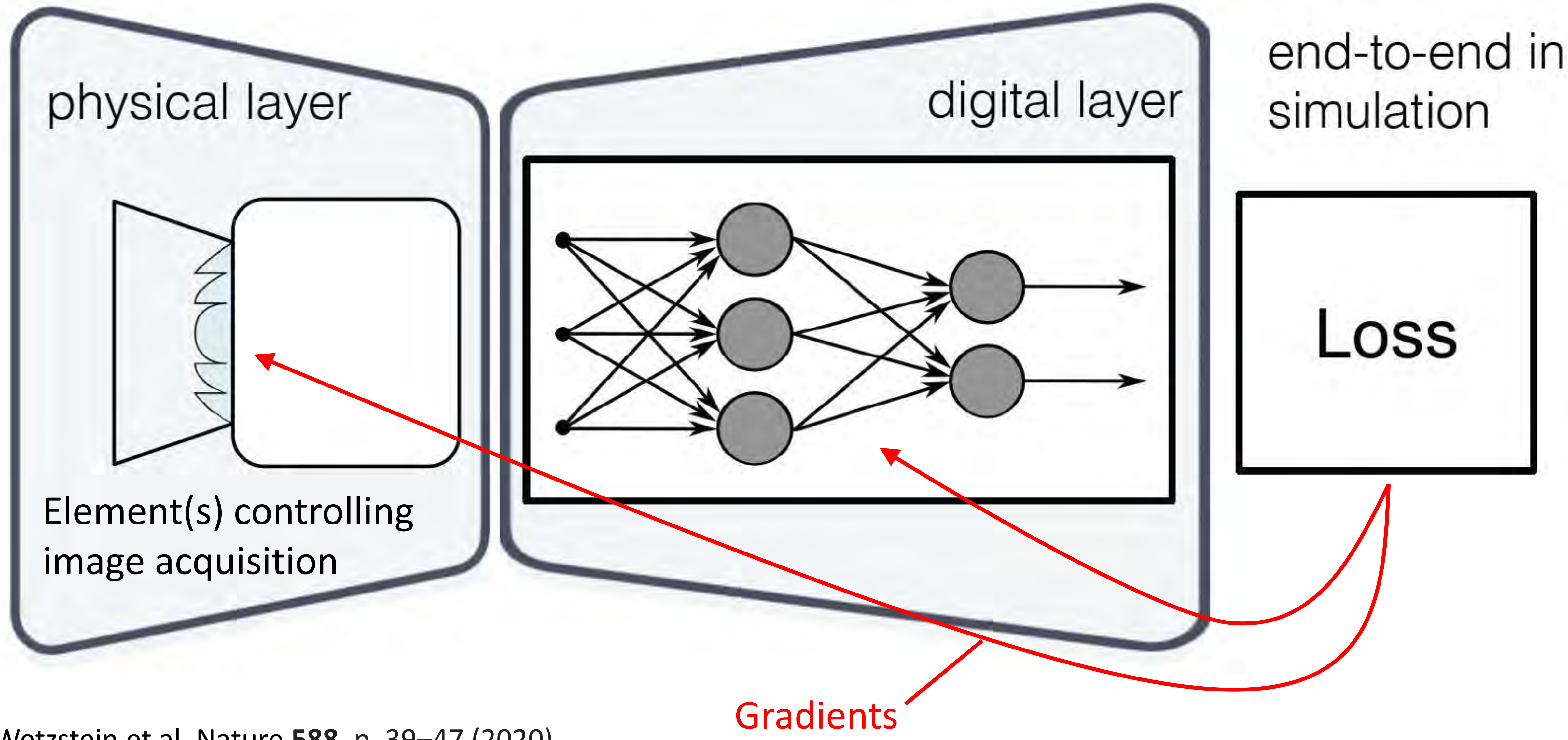


<u>Question 2:</u> How can we incorporate complex prior knowledge on the data?

<u>Question 3:</u> Can this concept be extended to higher level tasks like classification?

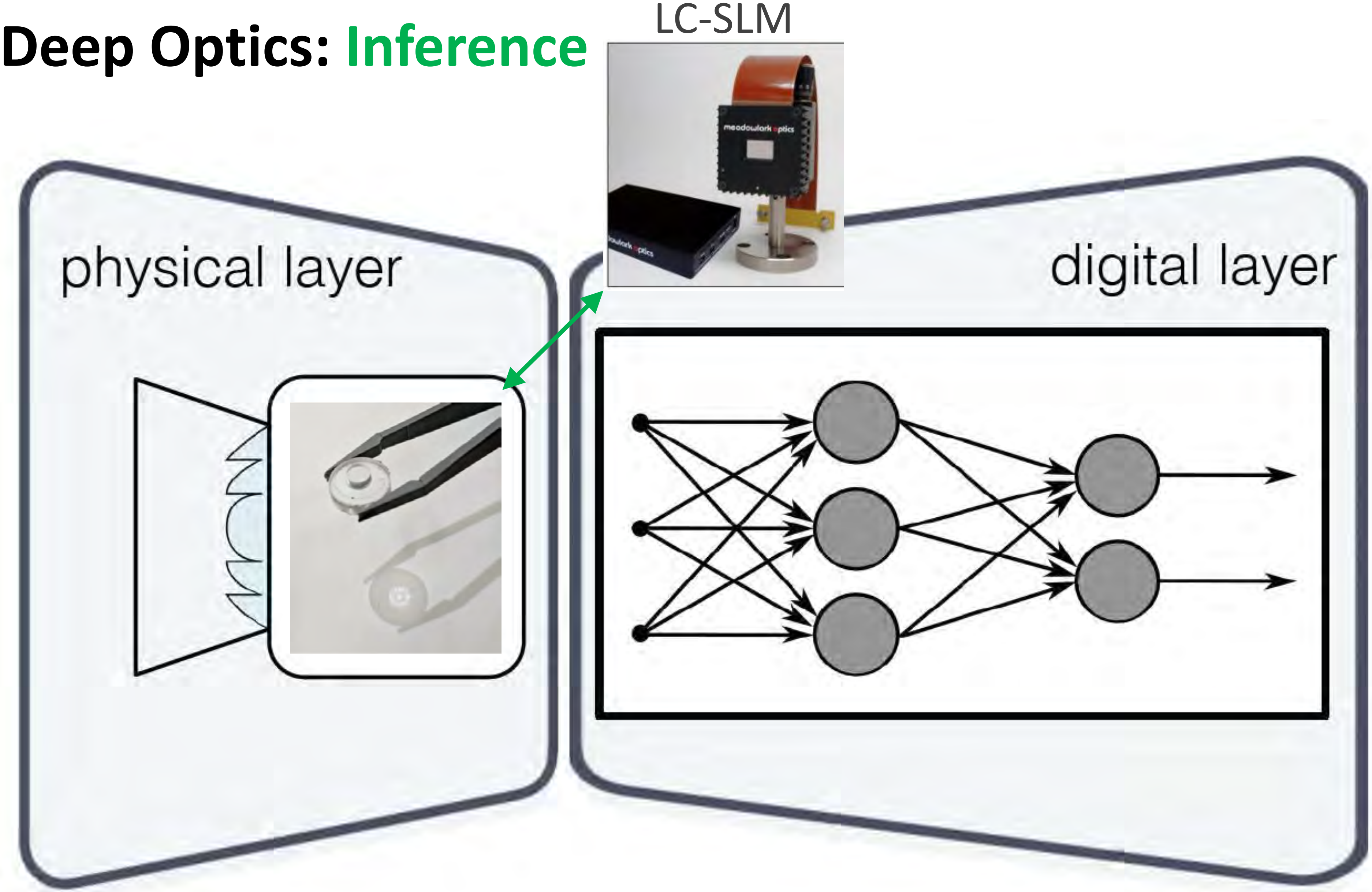
Answer: Deep Optics!

Deep Optics: Training



Wetzstein et al. Nature **588**, p. 39–47 (2020)

Deep Optics: Inference



Fabricate or implement lens or other physical components, and run network on measurements

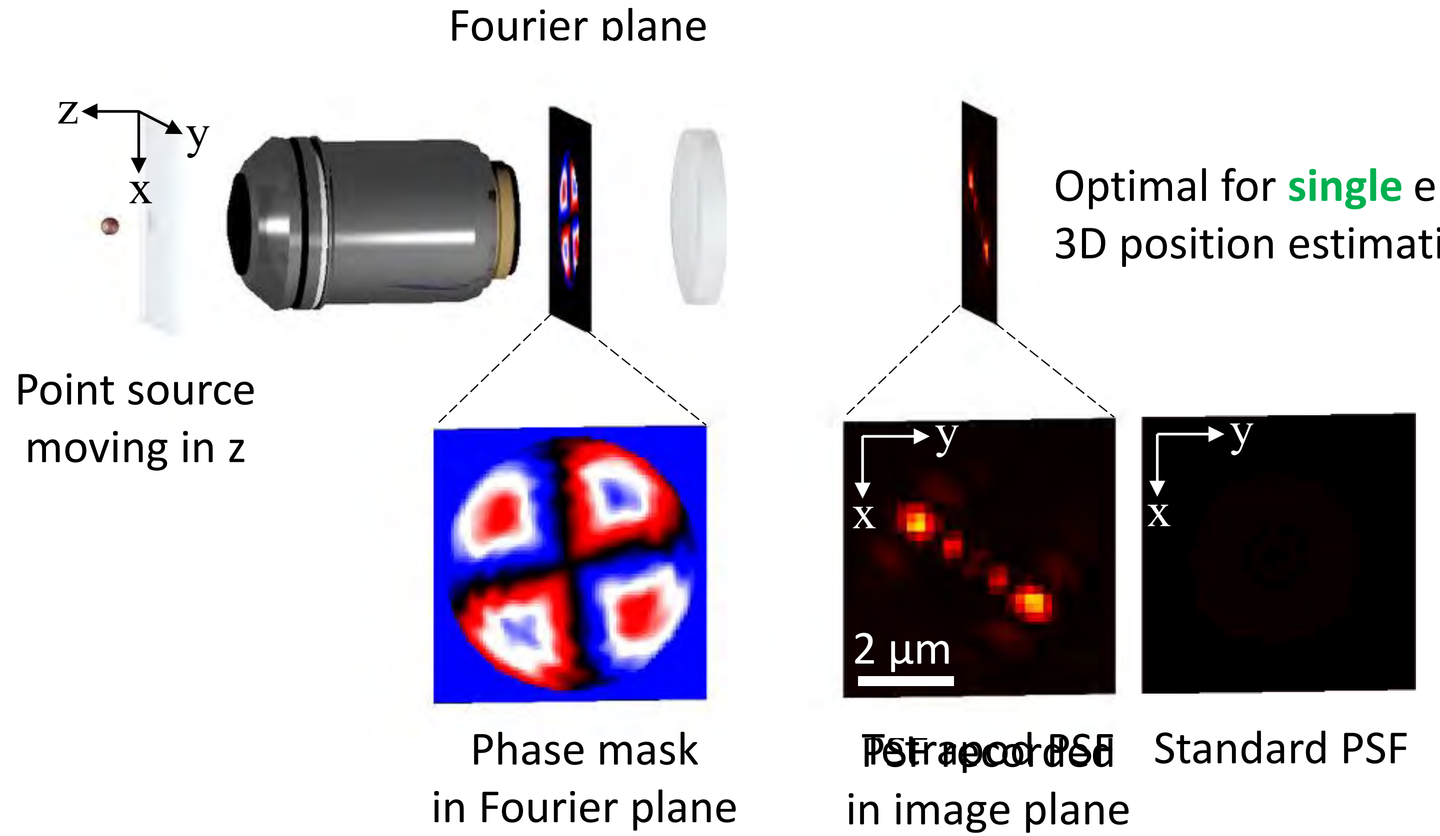
Wetzstein et al. Nature **588**, p. 39–47 (2020)

Outline

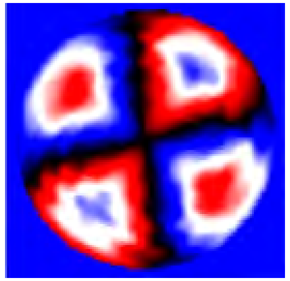
- 👍 Autoencoder interpretation
- 👉 Learning dense 3D imaging
- ▶ Generality to higher level tasks
- ▶ Multi-measurement systems
- ▶ Beyond microscopy

Extending a microscope to 3D: reminder

- Standard microscope PSF have depth ambiguity and signal is lost
- Main idea is to encode depth into the PSF shape efficiently



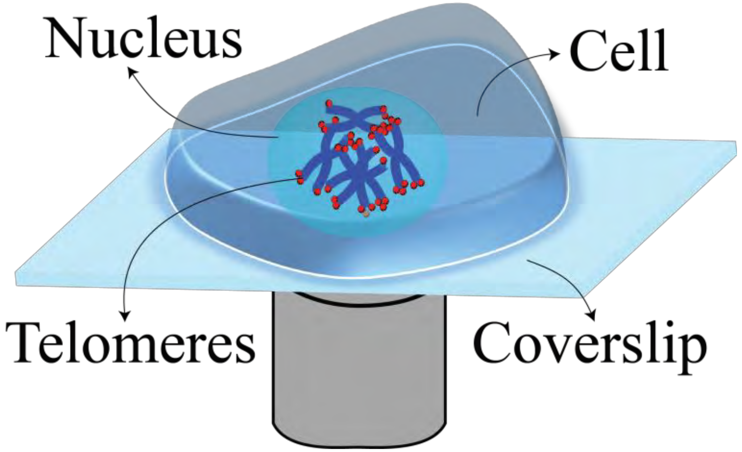
Optimal for **single** emitter 3D position estimation.

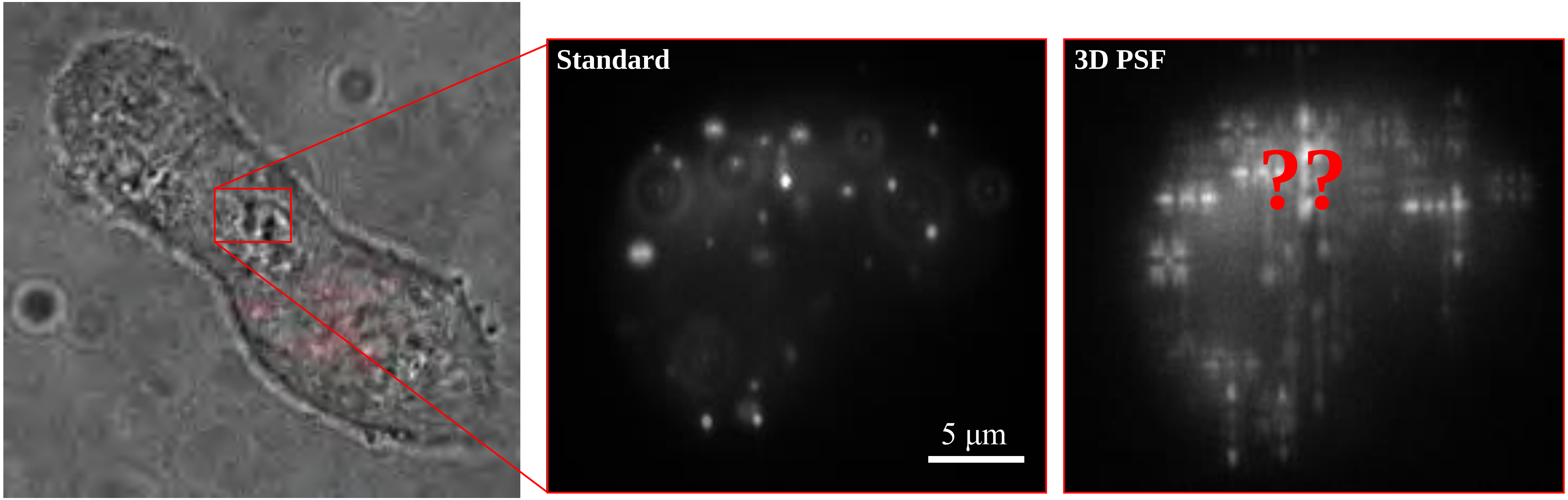


Phase mask

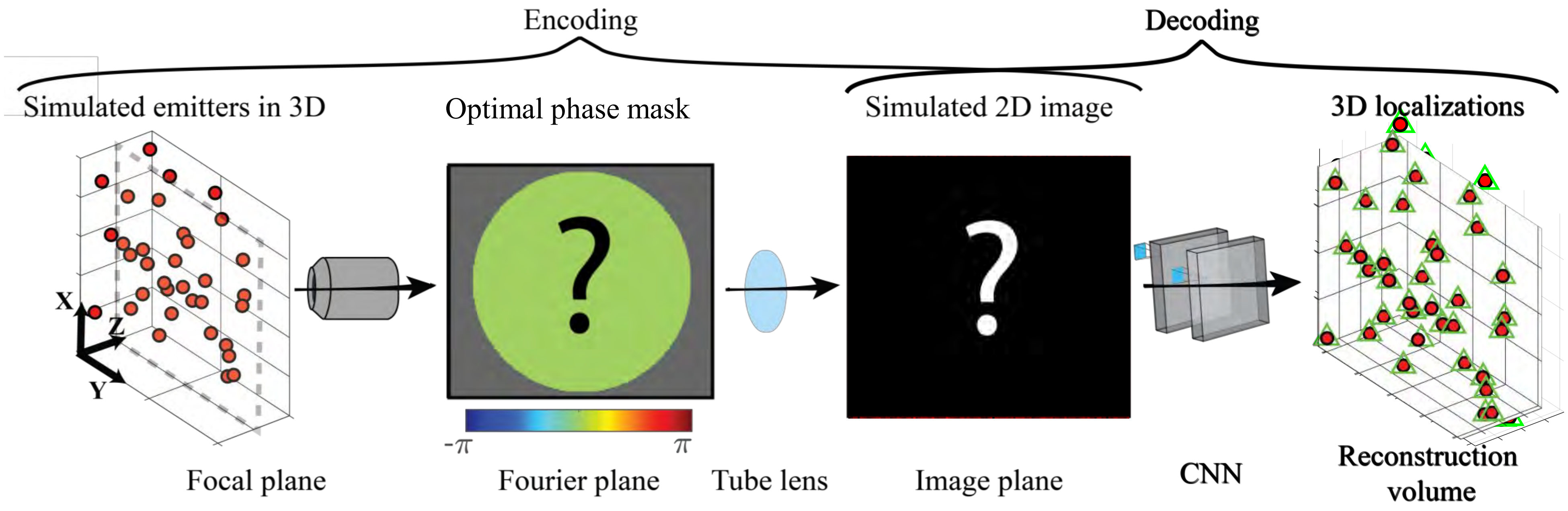
Shechtman et al., Phys. Rev. Letters (2014)

Sleep and cancer research require 3D tracking of telomeres in the nucleus of live cells





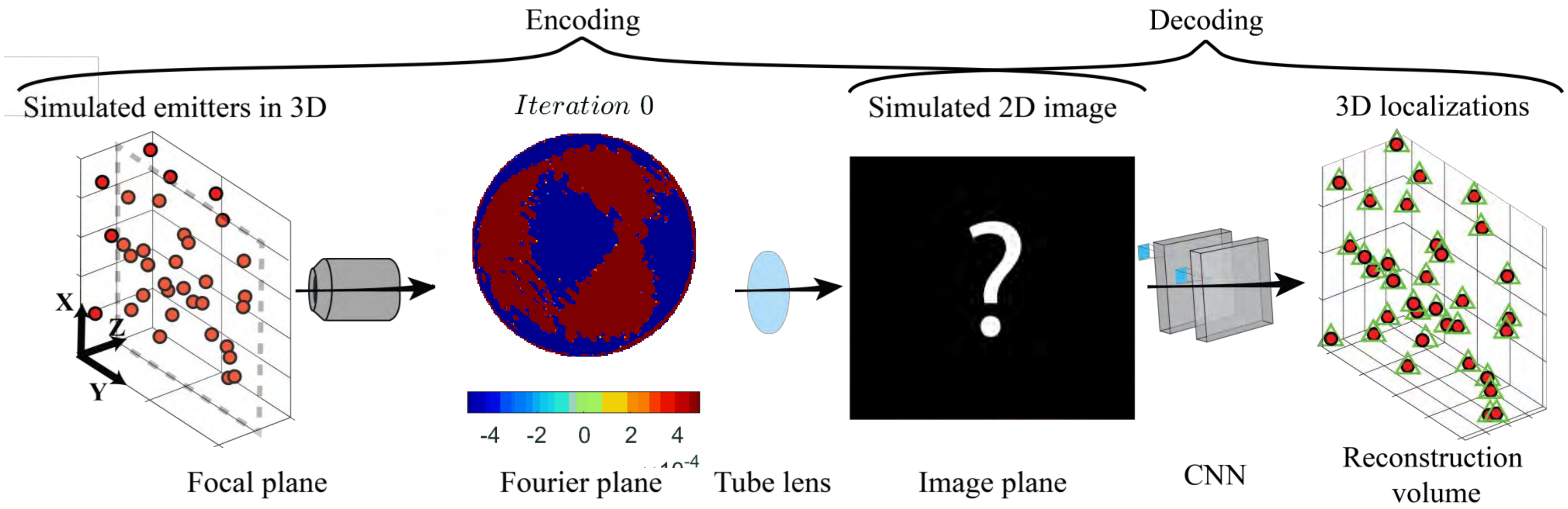
What is the optimal PSF for high density imaging?

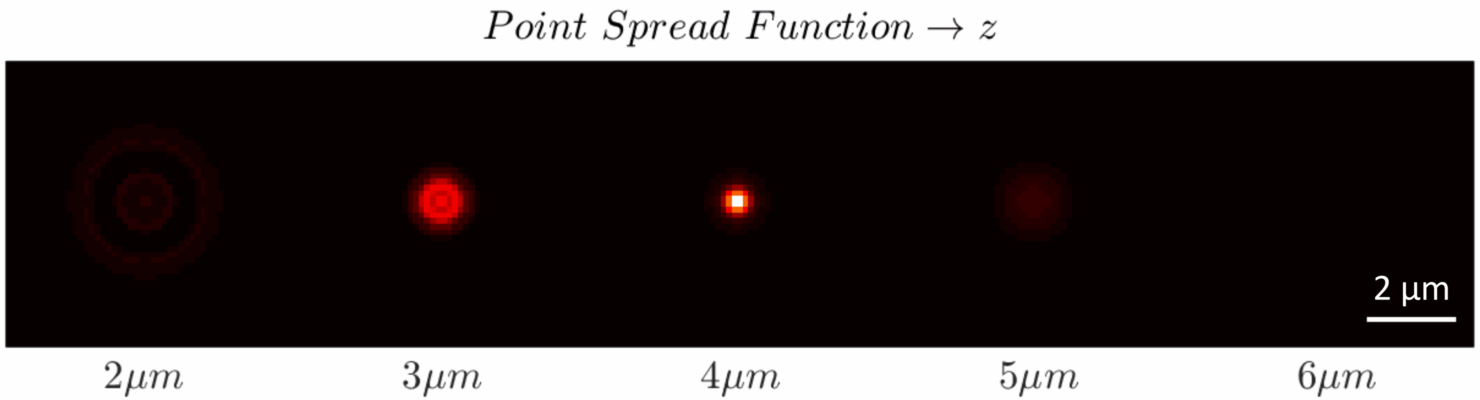


Answer: Let the net design it via backpropagation!

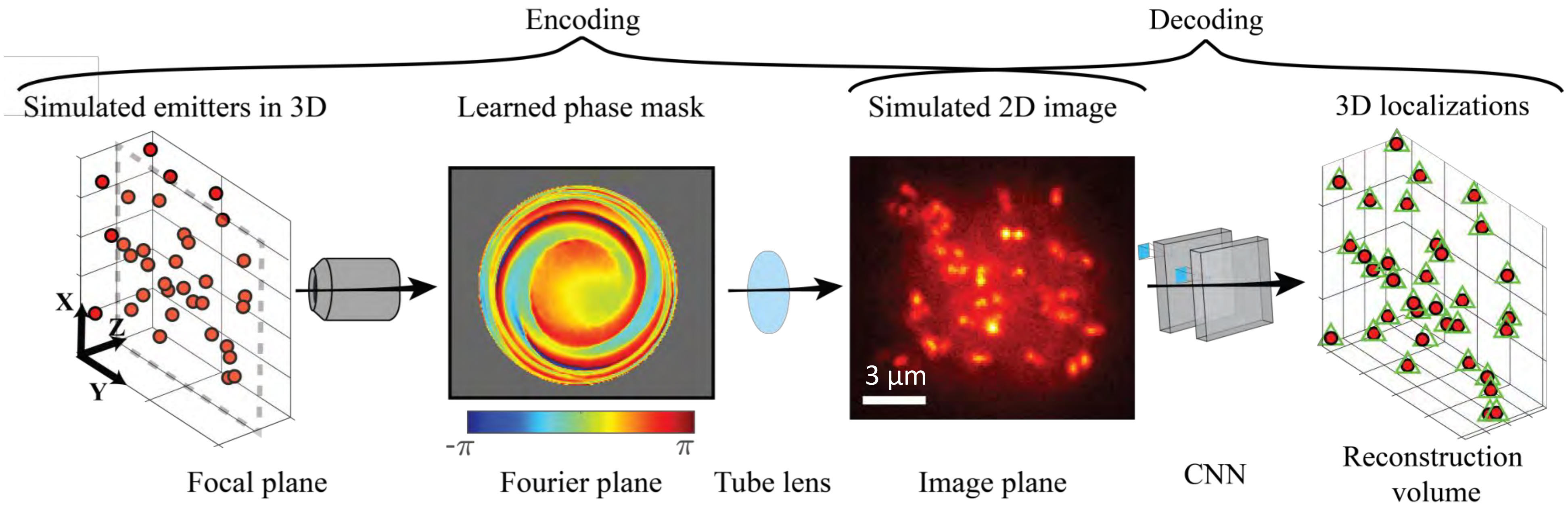
Nehme et al., Nature Methods (2020)

What is the optimal PSF for high density imaging?





What is the optimal PSF for high density imaging?

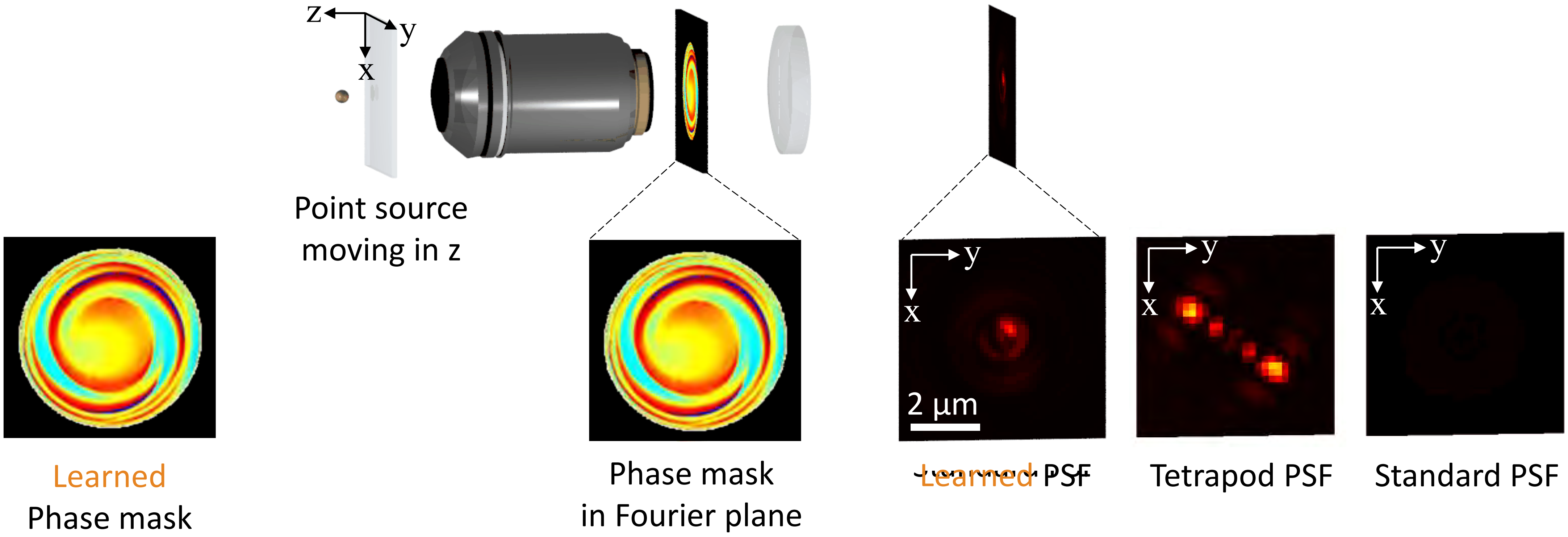


Put simply: The net designs its favorite PSF in order to perform best at decoding high density of emitters, thereby jointly optimizing the optics (encoding) and the localization algorithm (decoding)!

Nehme et al., Nature Methods (2020)

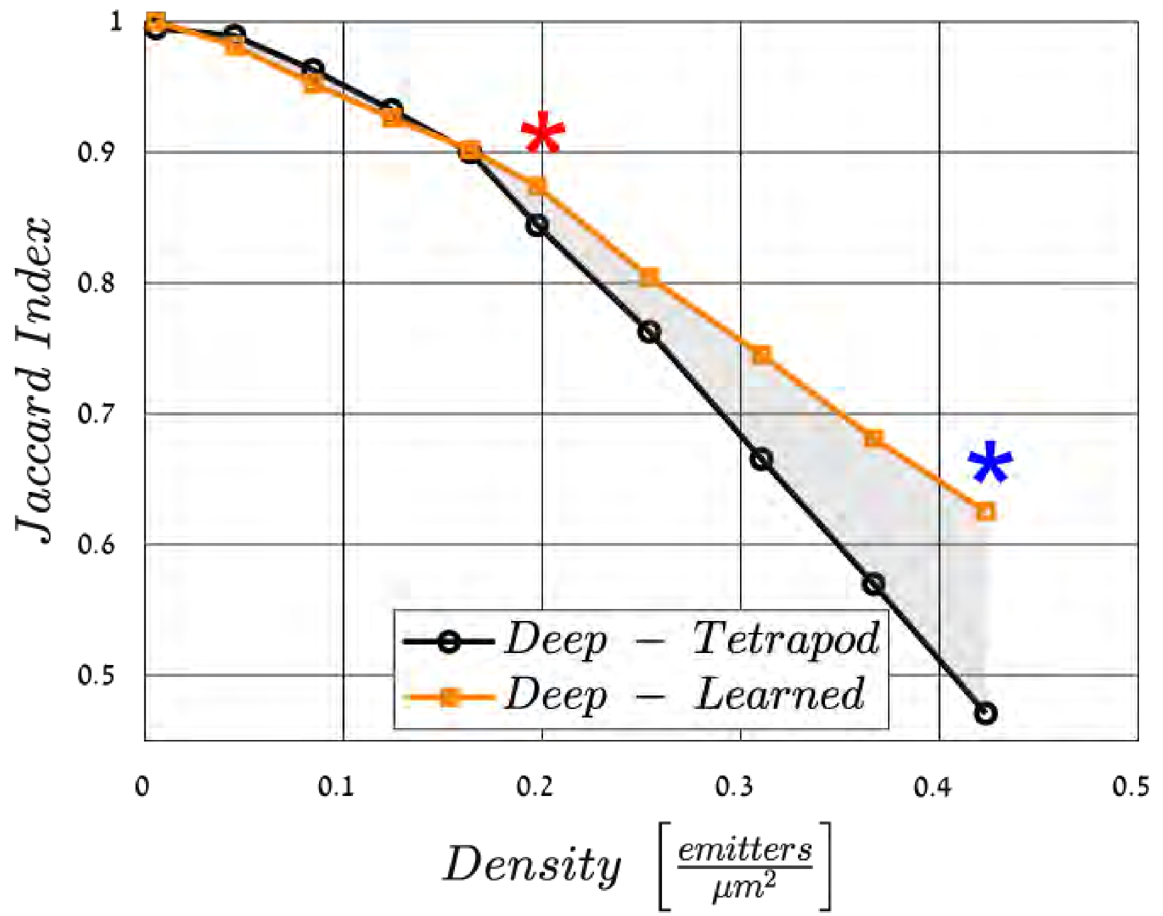
Learned phase mask and PSF

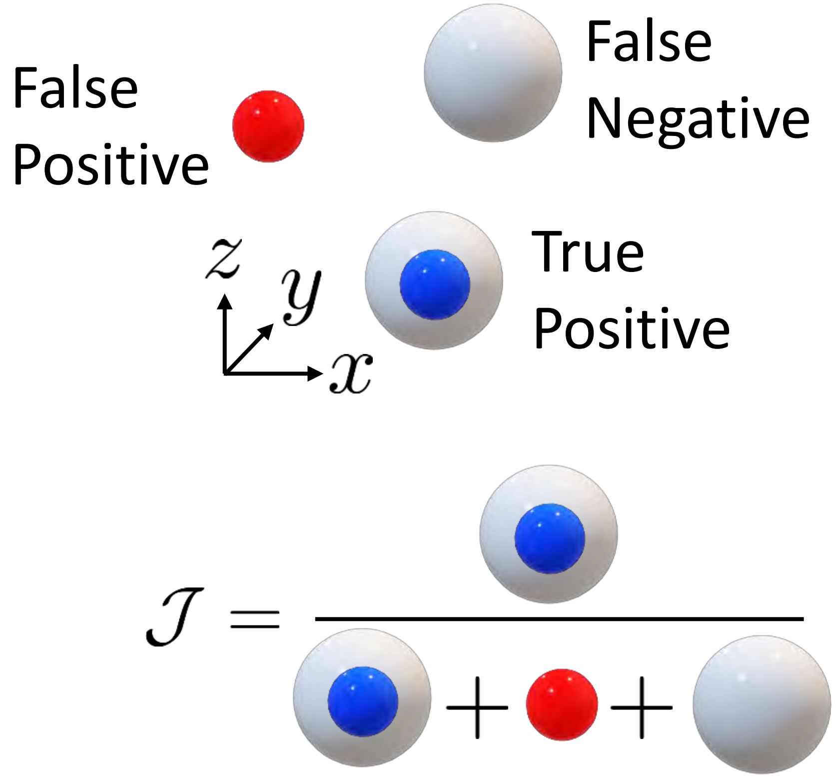
- Resembles familiar PSFs at different axial ranges



Nehme et al., Nature Methods (2020)

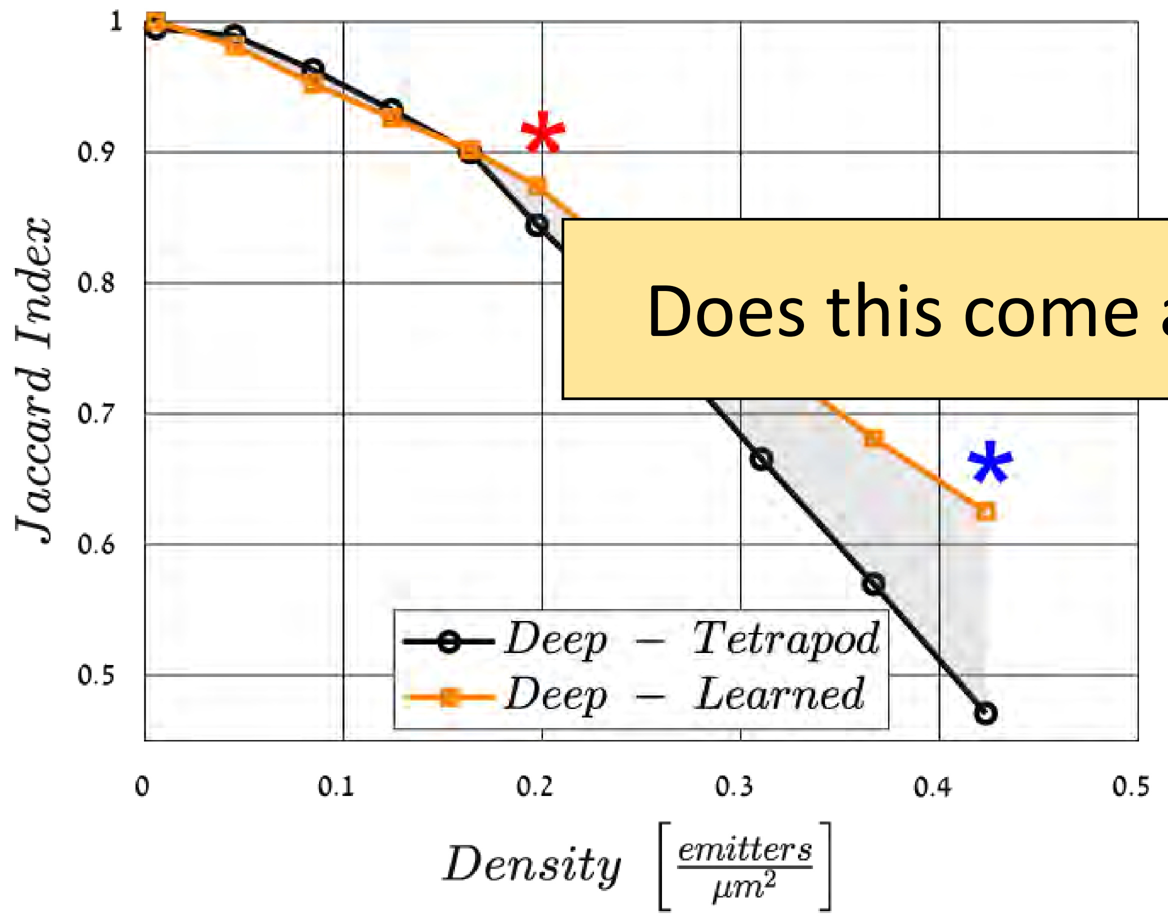
Tetrapod vs. Learned PSF





Nehme et al., Nature Methods (2020)

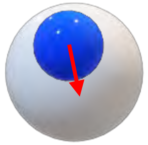
Tetrapod vs. Learned PSF

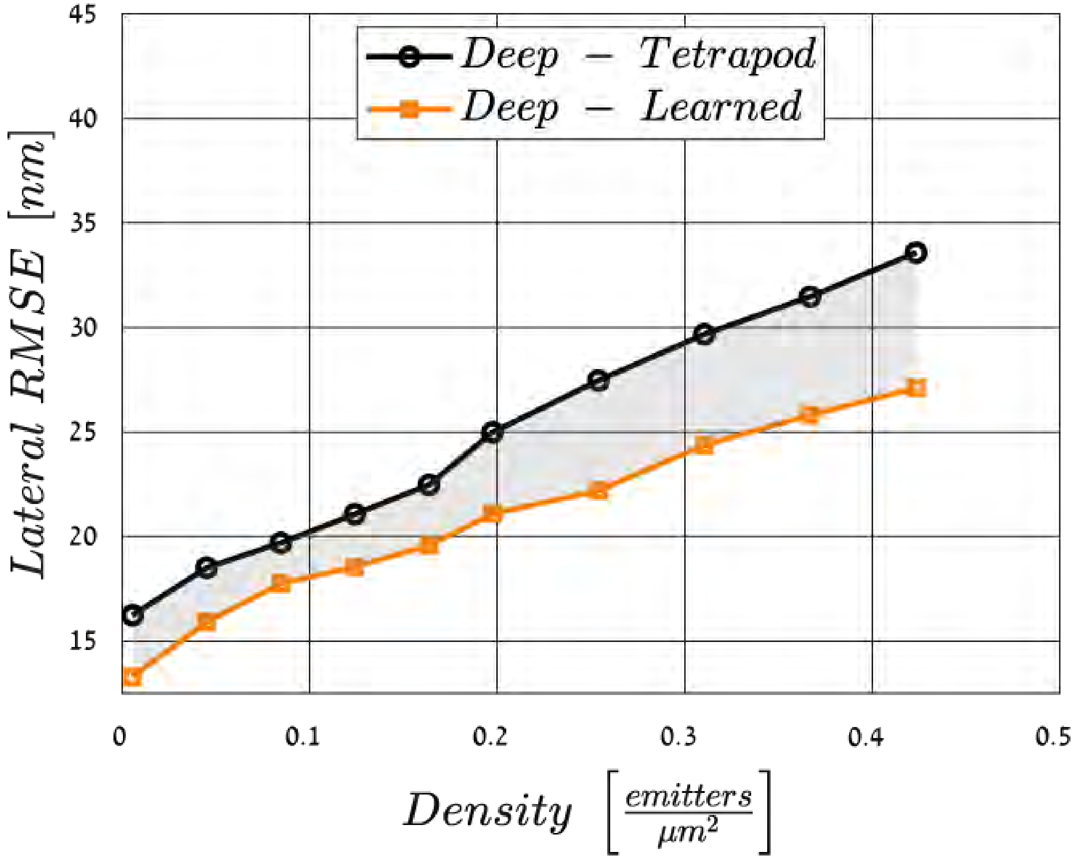


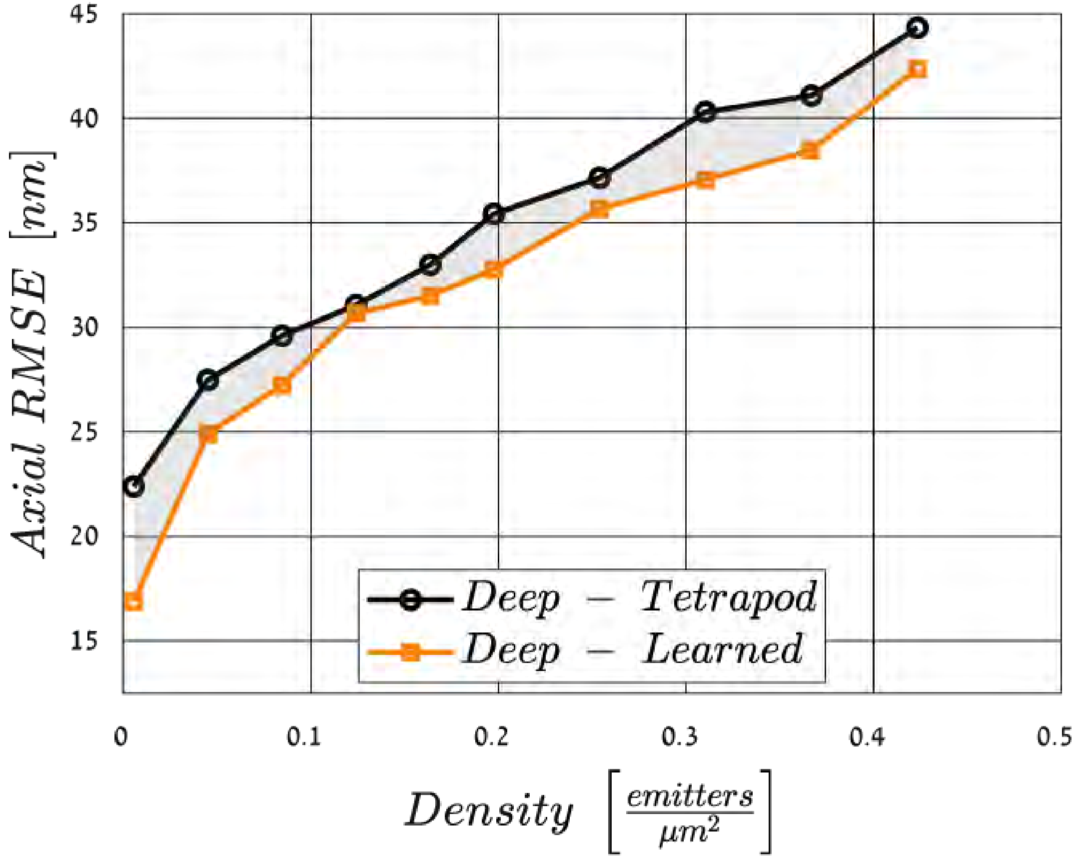
Does this come at a cost of precision?

Nehme et al., Nature Methods (2020)

Tetrapod vs. Learned PSF

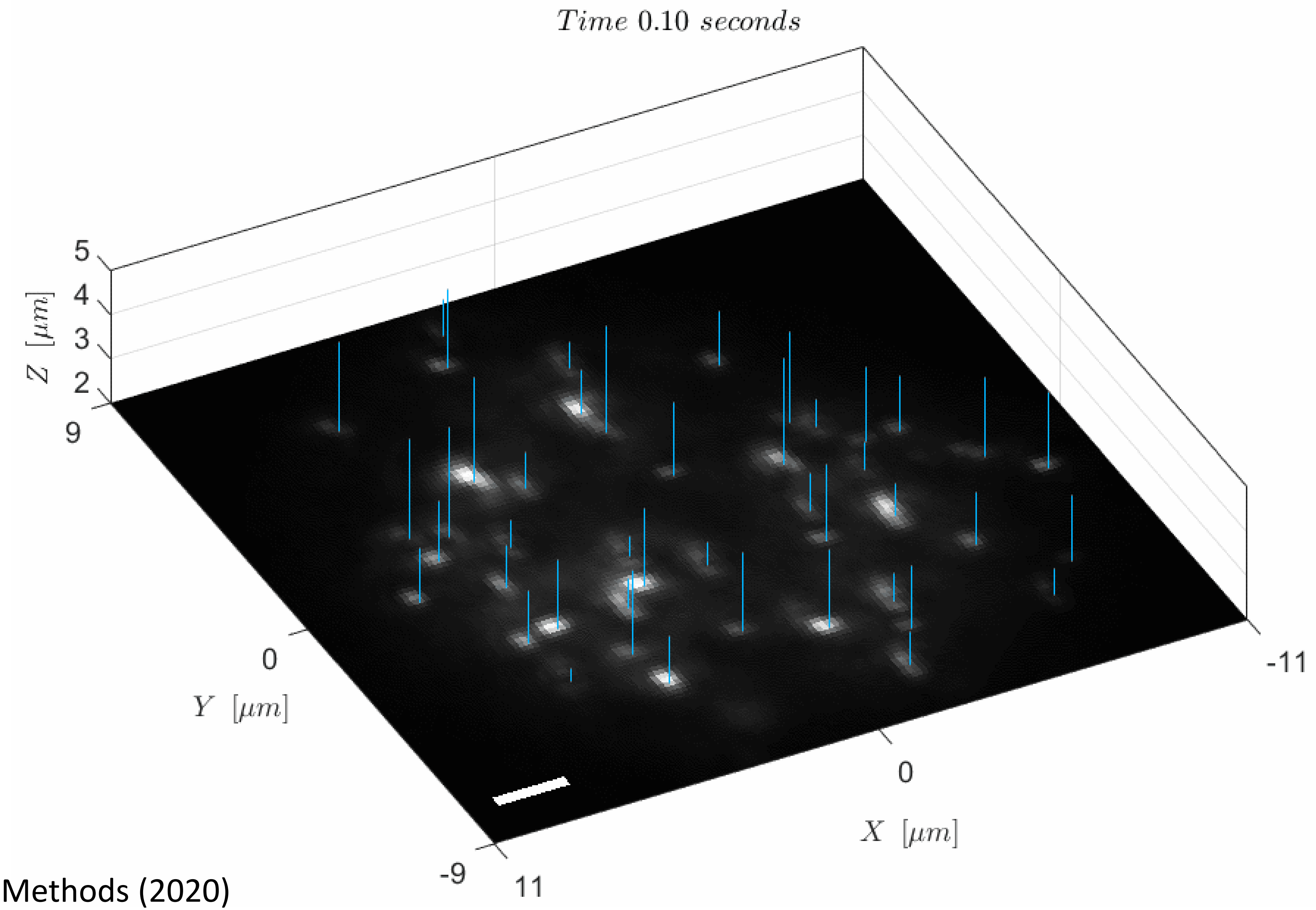






Nehme et al., Nature Methods (2020)

Live cell 3D tracking with the Learned PSF



Nehme et al., Nature Methods (2020)

Outline

- 👍 Autoencoder interpretation
- 👍 Learning dense 3D imaging
- 👉 Generality to higher level tasks
- ▶ Multi-measurement systems
- ▶ Beyond microscopy

How do computer vision pipelines work?

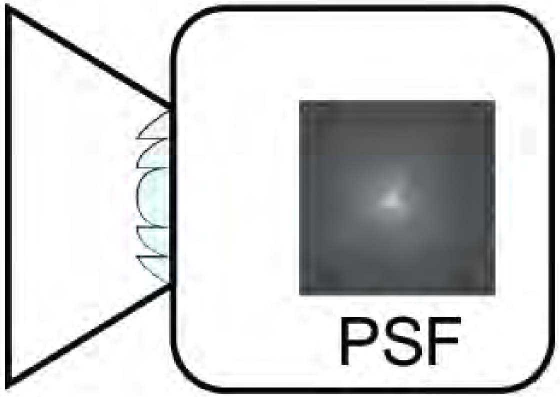
Objective: Solve CV problem on scene

What is this?

Peng et al., ACM SIGGRAPH (2020)

How do computer vision pipelines work?

Step 1: Build camera

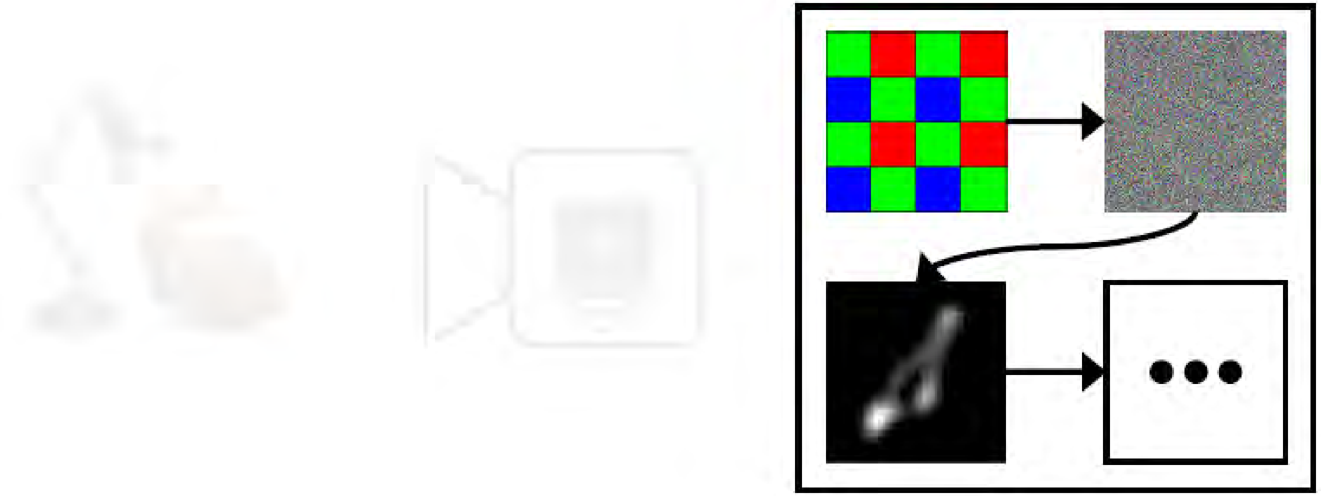


Optimize optics to minimize aberrations:
Blur/spot size, chromatic aberrations, distortions, …

Peng et al., ACM SIGGRAPH (2020)

How do computer vision pipelines work?

Step 2: Image Signal Processing (ISP)

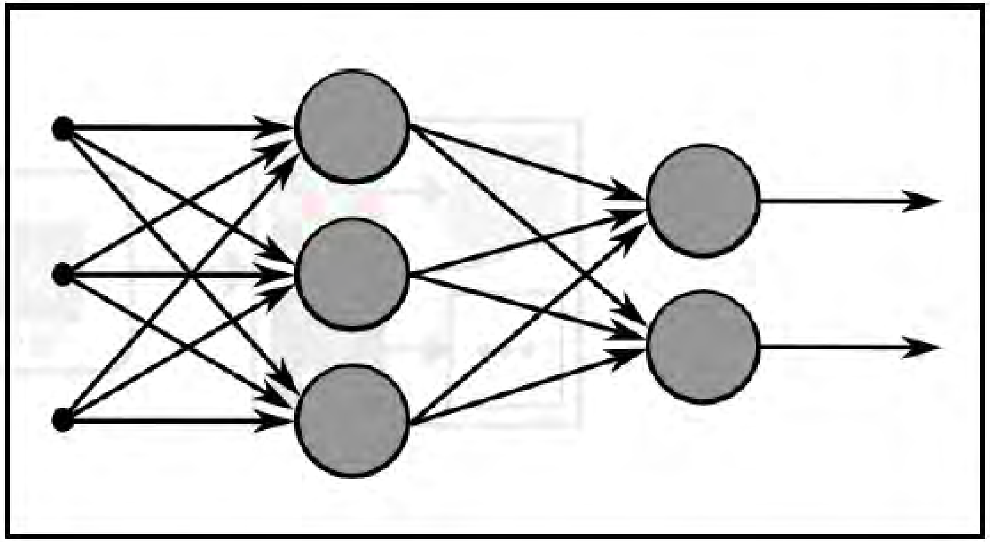


Maximize PSNR:
Demosaicking, Denoising, Deblurring, …

Peng et al., ACM SIGGRAPH (2020)

How do computer vision pipelines work?

Step 3: CNN for Semantic task

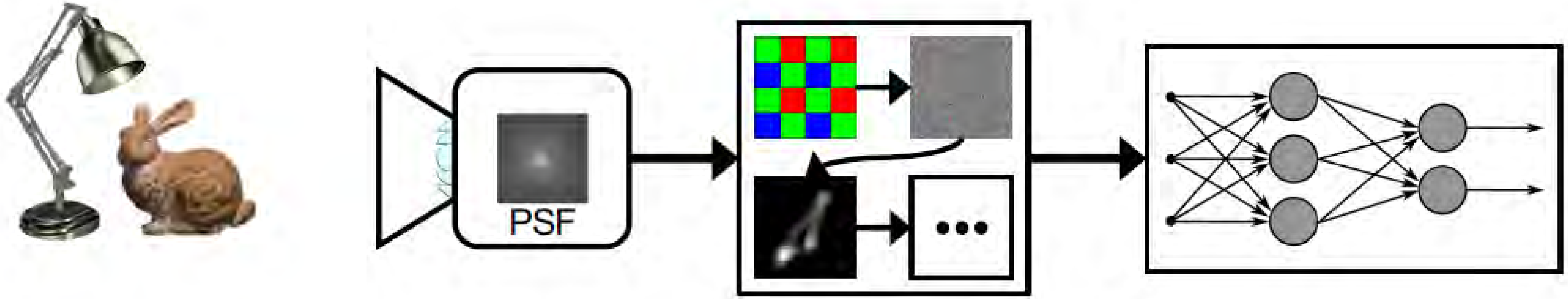


Minimize semantic loss:
classification error, segmentation error, …

Peng et al., ACM SIGGRAPH (2020)

How do computer vision pipelines work?

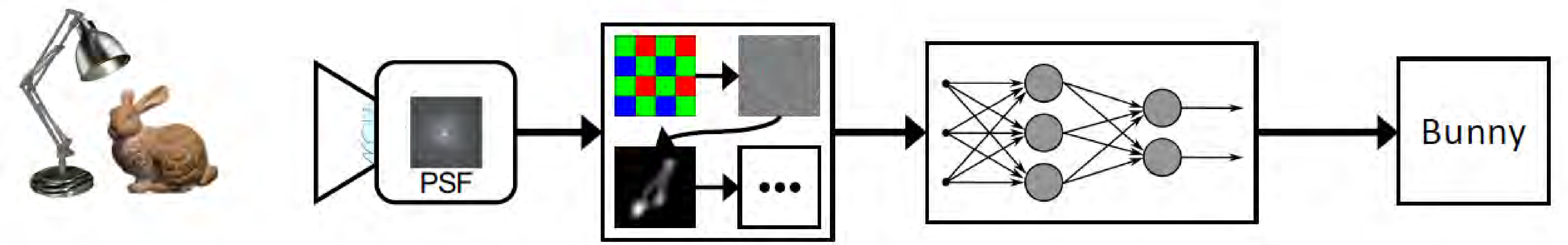
. . .



Peng et al., ACM SIGGRAPH (2020)

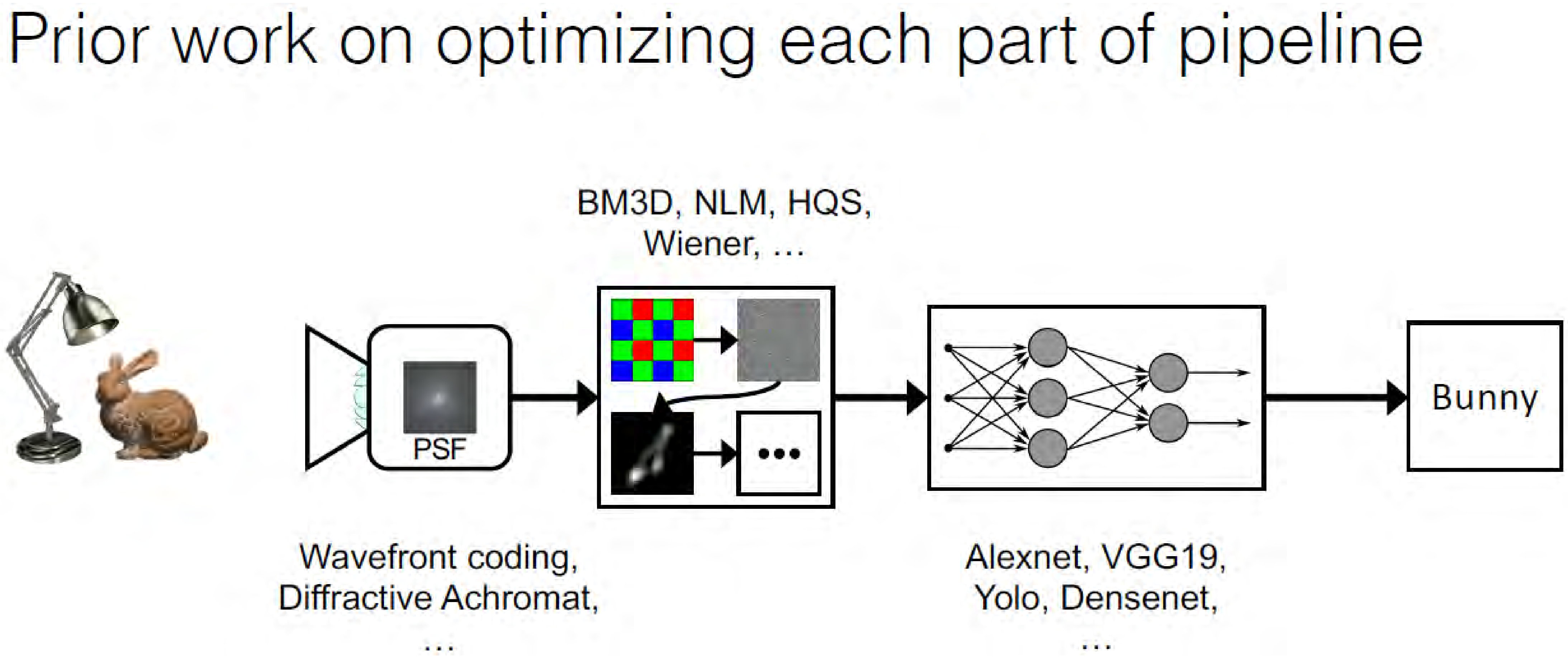
How do computer vision pipelines work?

It works!



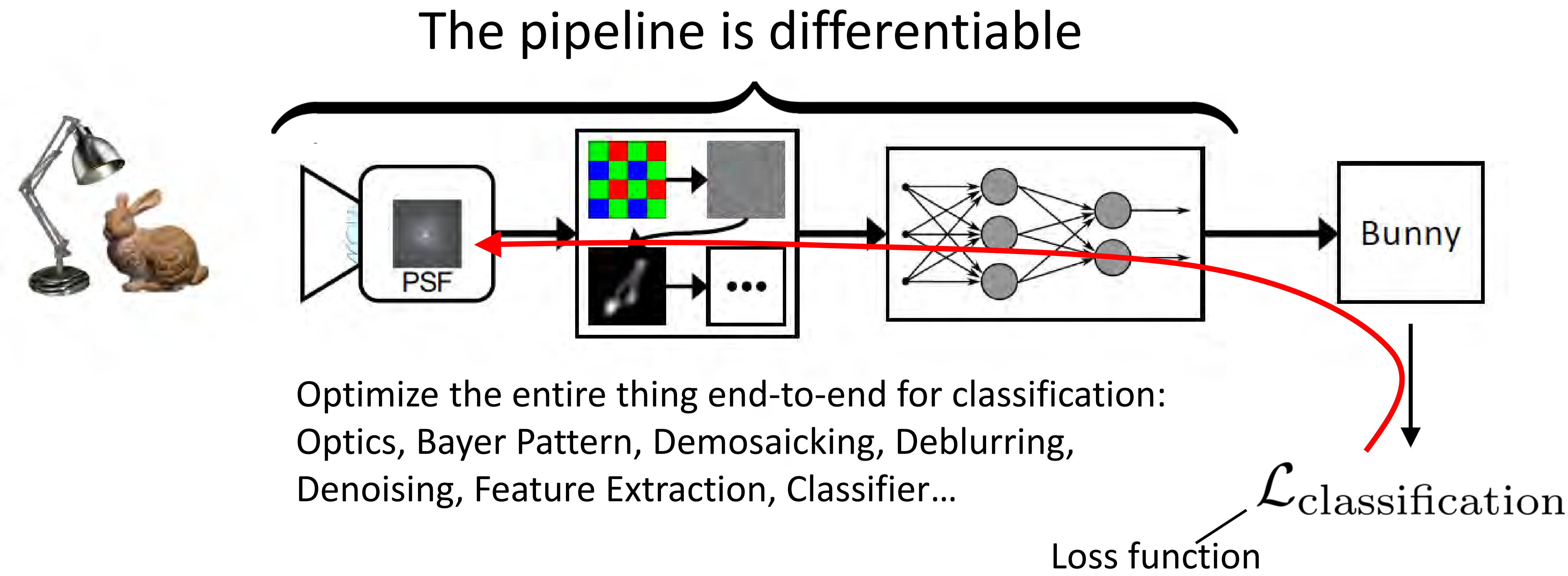
Peng et al., ACM SIGGRAPH (2020)

How do computer vision pipelines work?



Peng et al., ACM SIGGRAPH (2020)

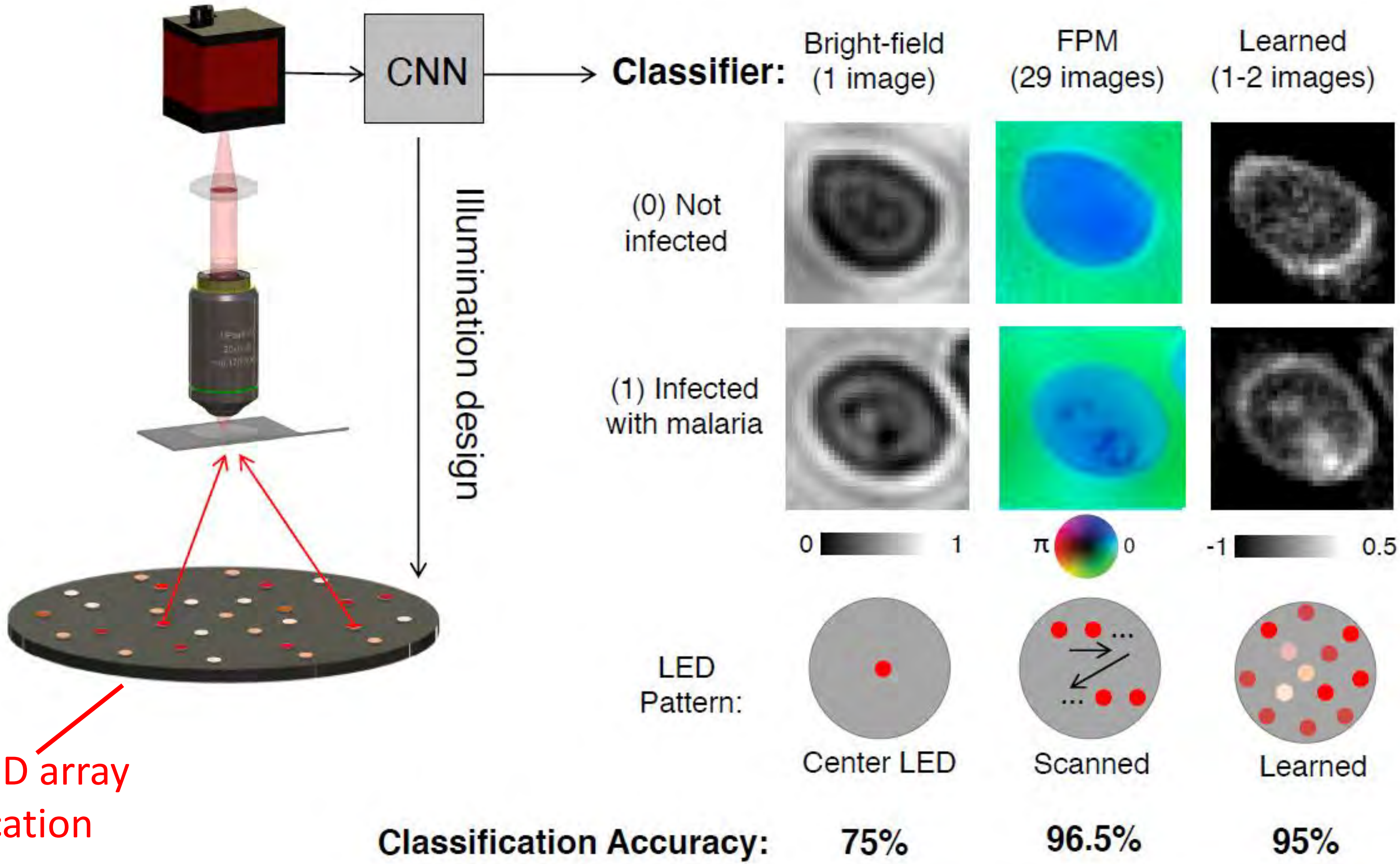
Why not optimize sensors for classification?



Optimize the entire thing end-to-end for classification:
Optics, Bayer Pattern, Demosaicking, Deblurring, Denoising, Feature Extraction, Classifier…

Peng et al., ACM SIGGRAPH (2020)

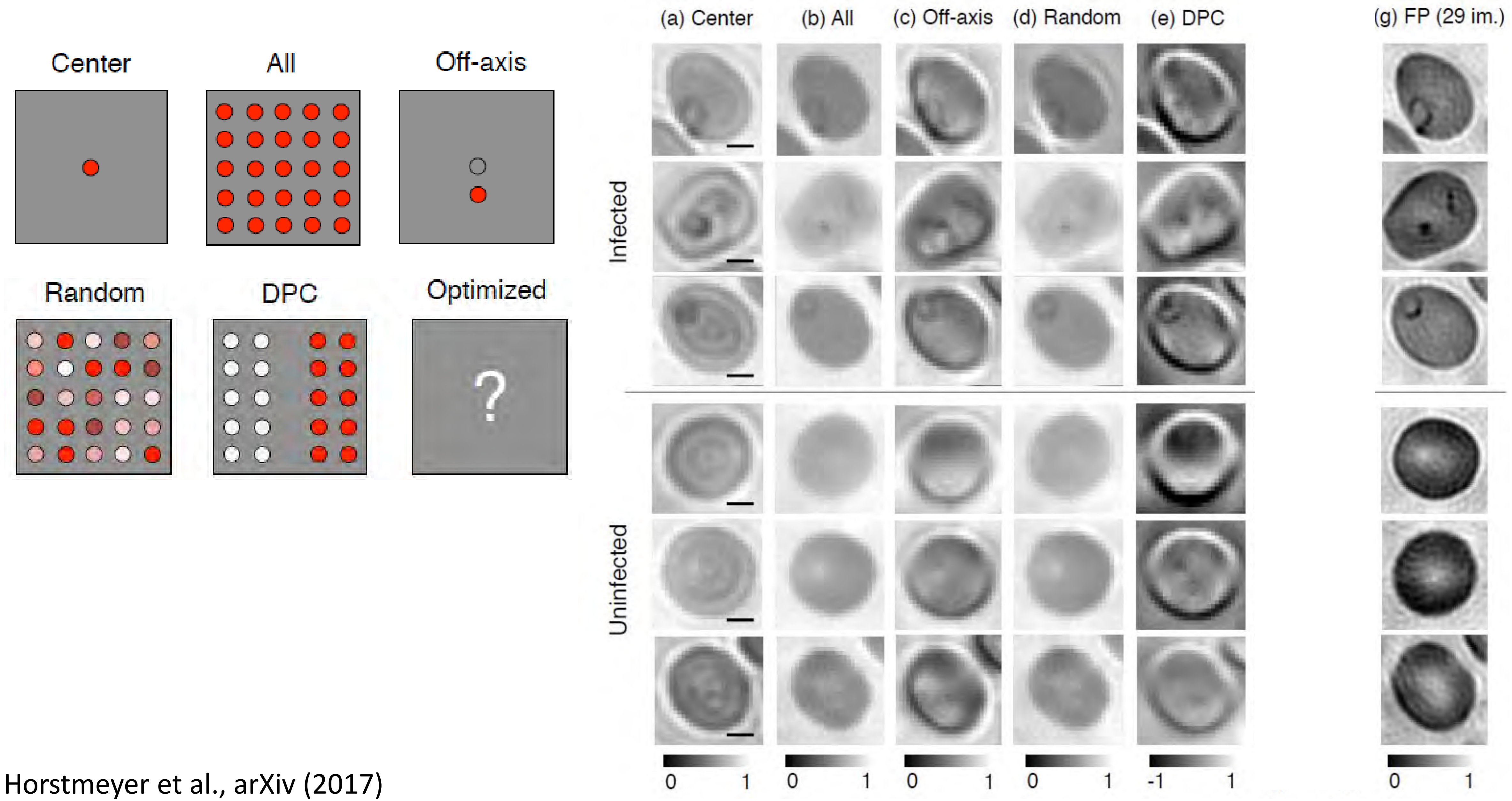
Optimizing microscopes for malaria classification



Optimized LED array for classification

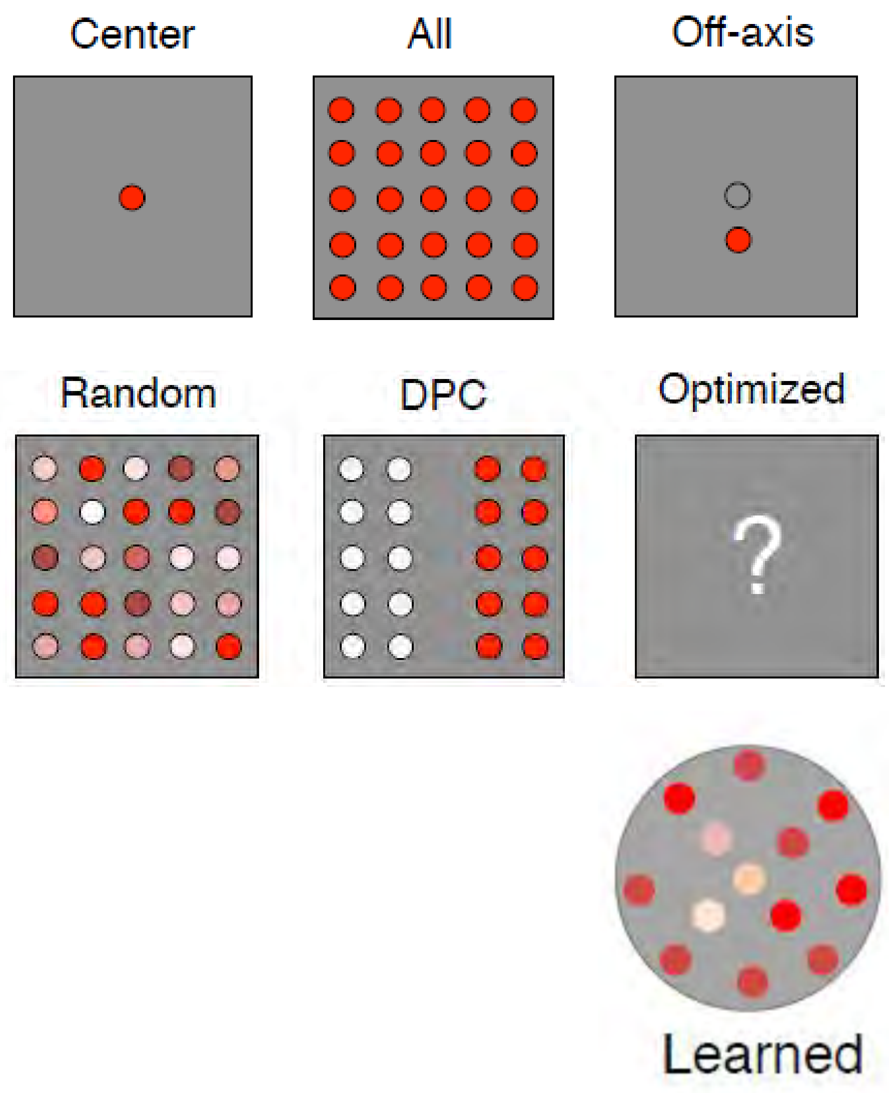
Horstmeyer et al., arXiv (2017)

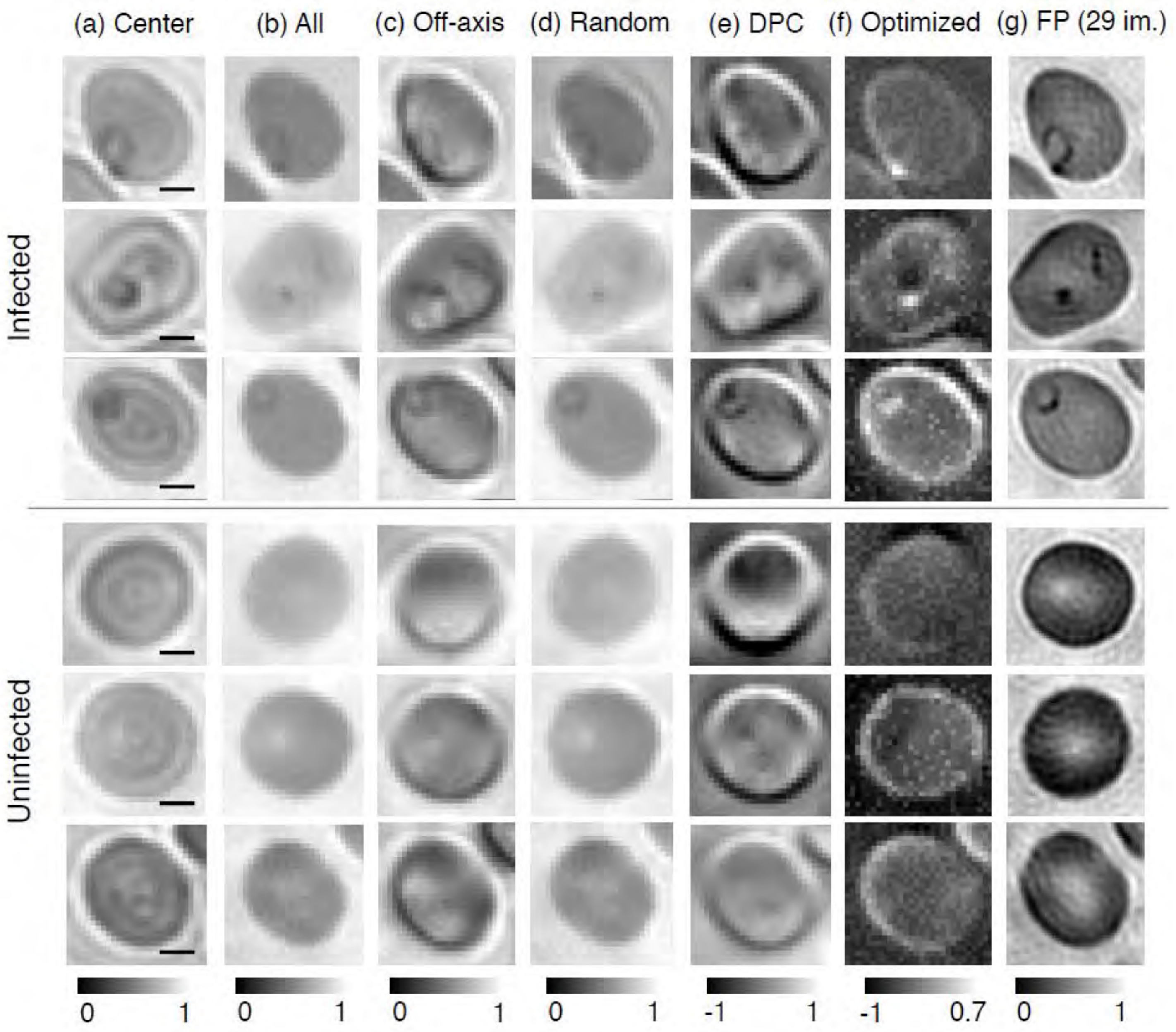
Optimizing microscopes for malaria classification



R. Horstmeyer et al., arXiv (2017)

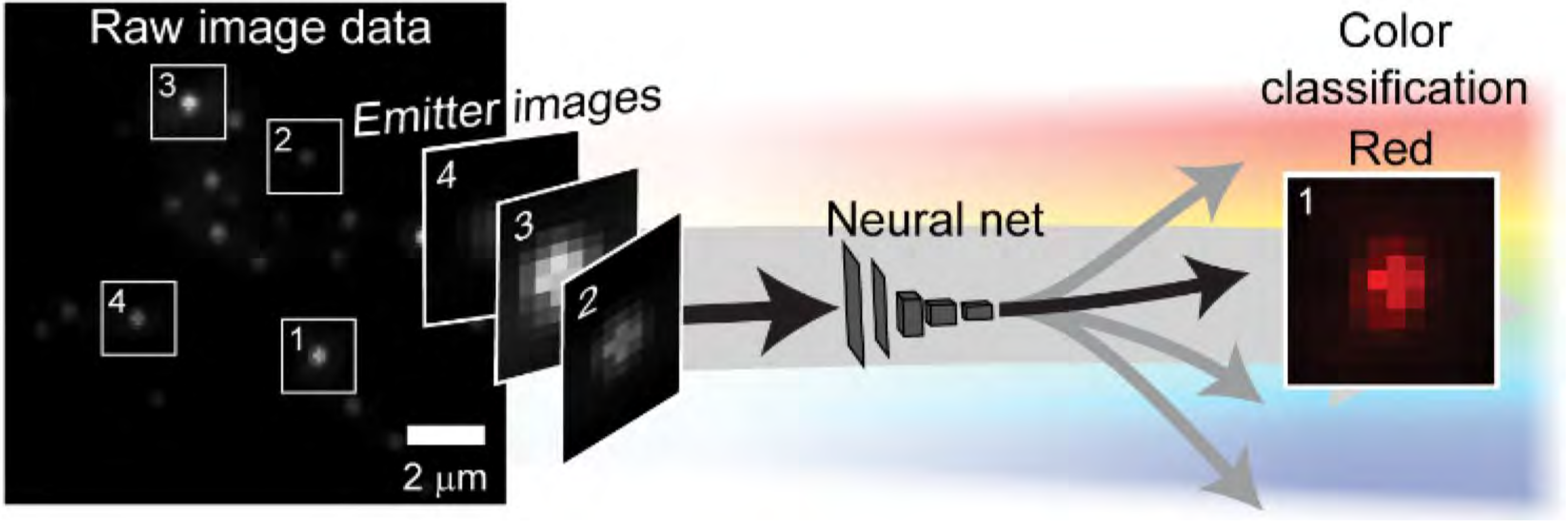
Optimizing microscopes for malaria classification





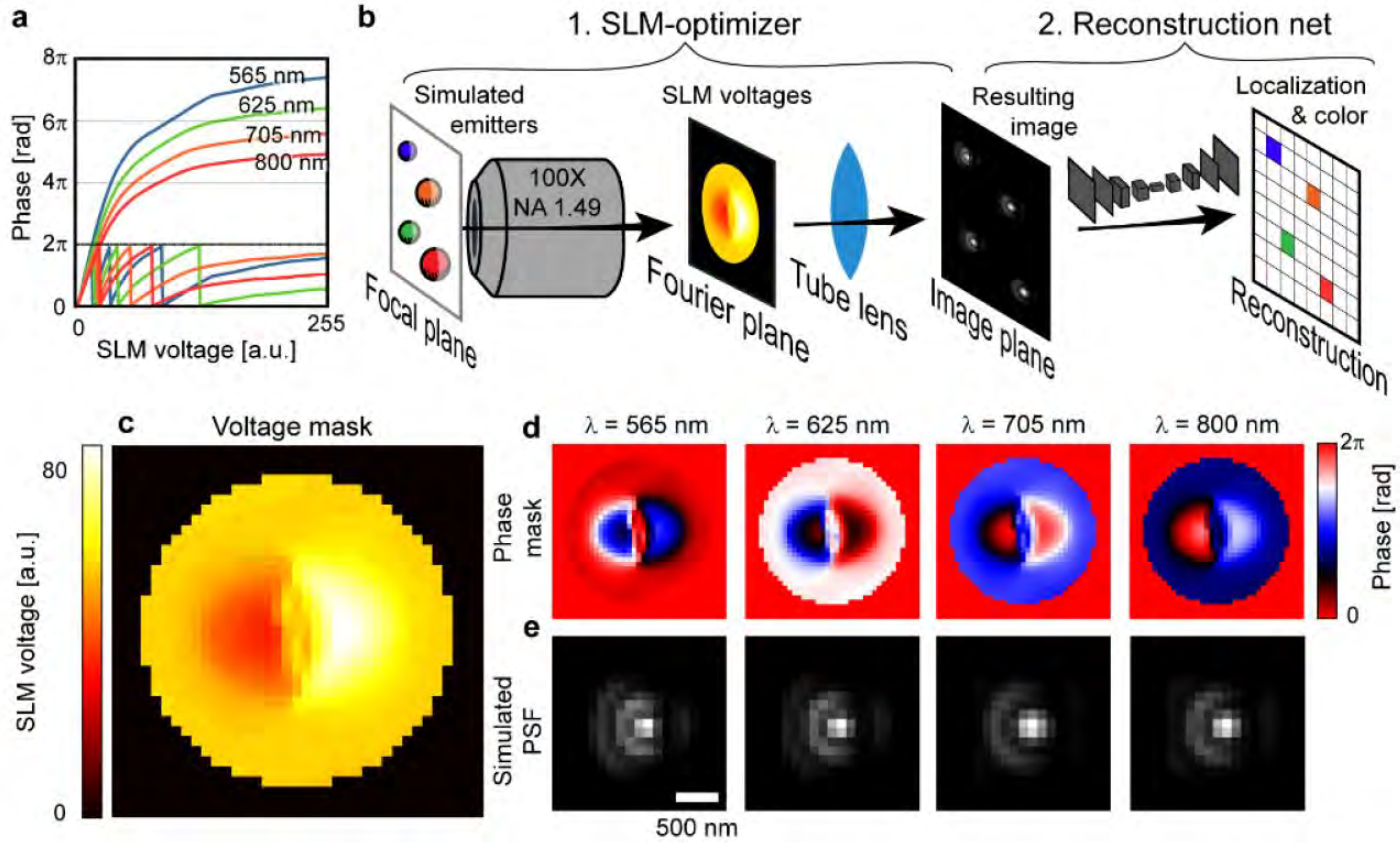
R. Horstmeyer et al., arXiv (2017)

Similar ideas can be used for single-emitter color classification



Hershko et al., Optics Express (2019)

Similar ideas can be used for single-emitter color classification



Hershko et al., Optics Express (2019)

Outline

- 👍 Autoencoder interpretation
- 👍 Learning dense 3D imaging
- 👍 Generality to higher level tasks
- 👉 Multi-measurement systems
- ▶ Beyond microscopy

Cameras are everywhere!

iPhone 11

Huawei P20 Pro

Galaxy A9

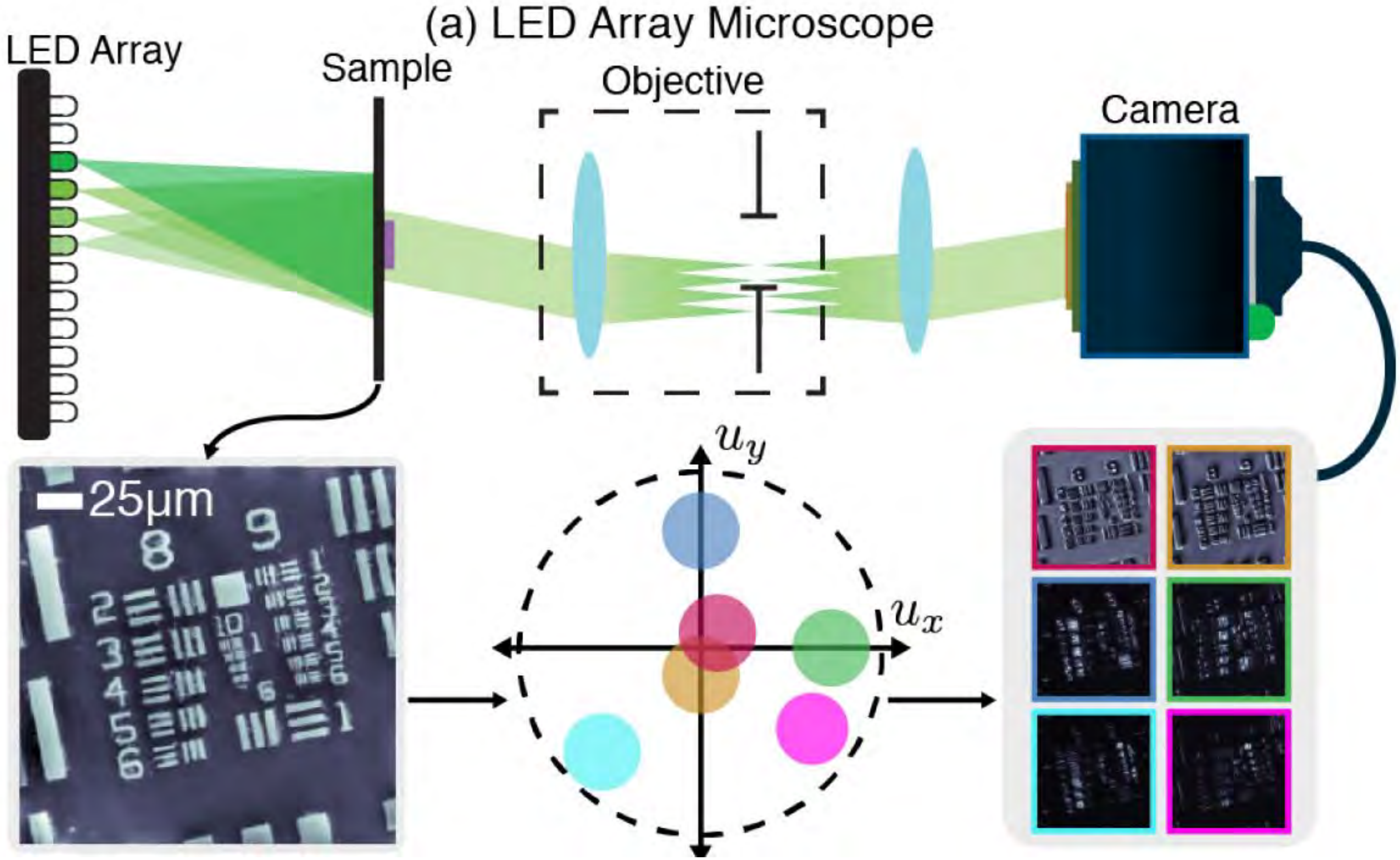
Nokia 9

<u>Question 1:</u> Can we design cameras with multiple acquisitions to produce the final result? Yes!

<u>Question 2:</u> Can we design multiple cameras simultaneously? Yes!

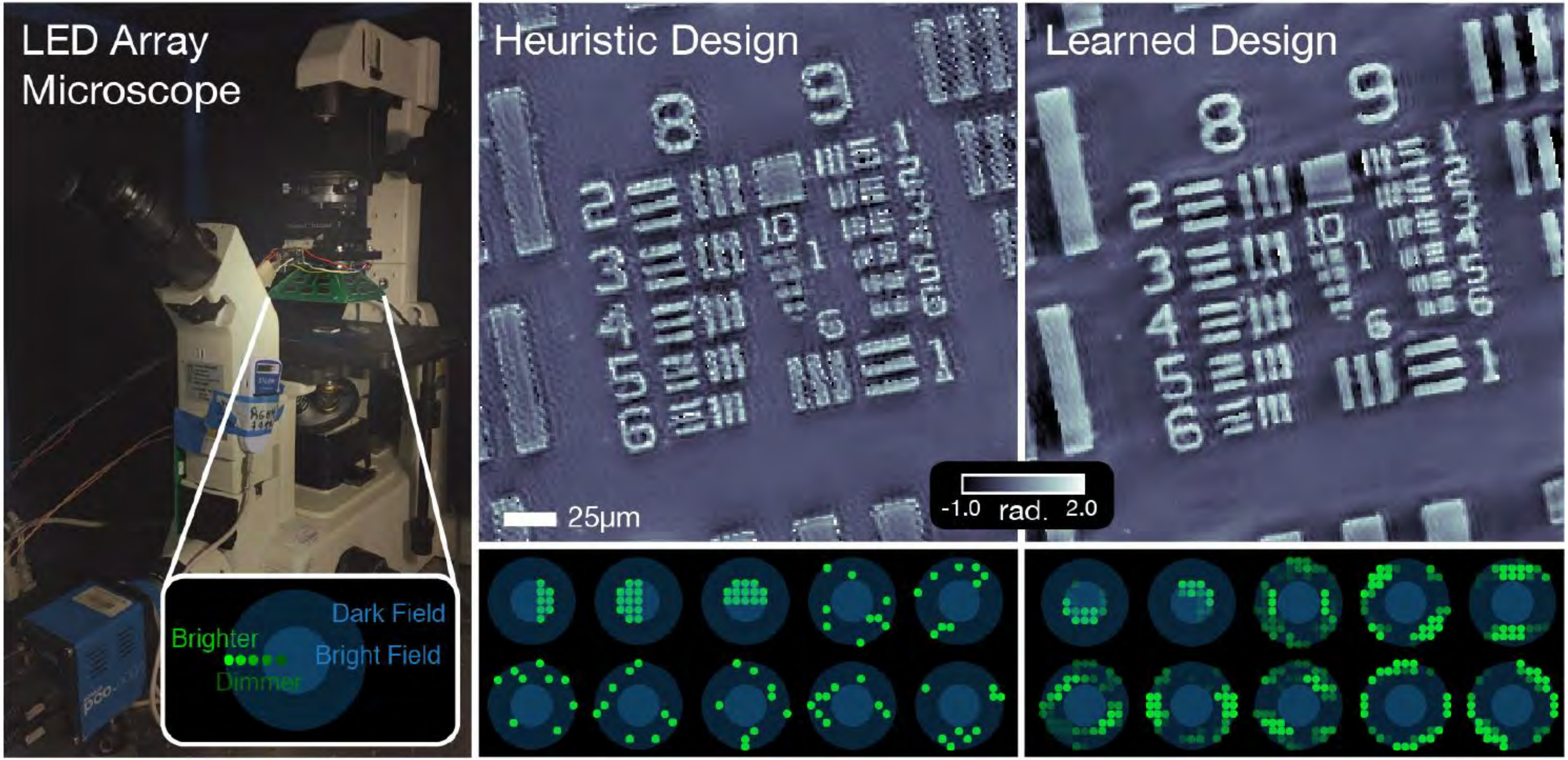
<u>Question 3:</u> How far can we push the concept of end-to-end design before the optimization landscape becomes prohibitive? We don’t know!

Learning multiple LED patterns for Fourier Ptychography



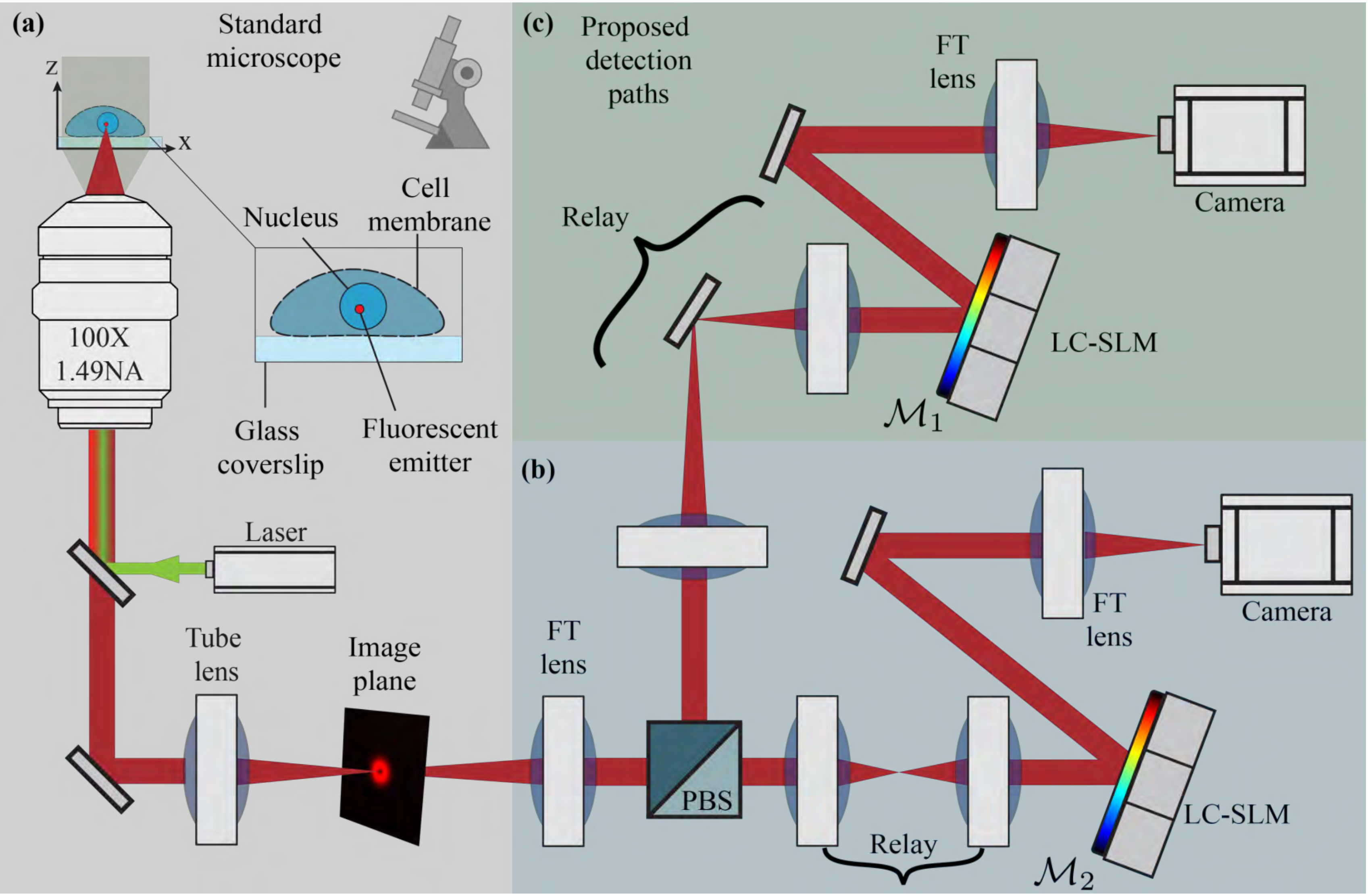
Kellman et al., *IEEE Trans. on Comp. Imaging* (2019)

Learning multiple LED patterns for Fourier Ptychography



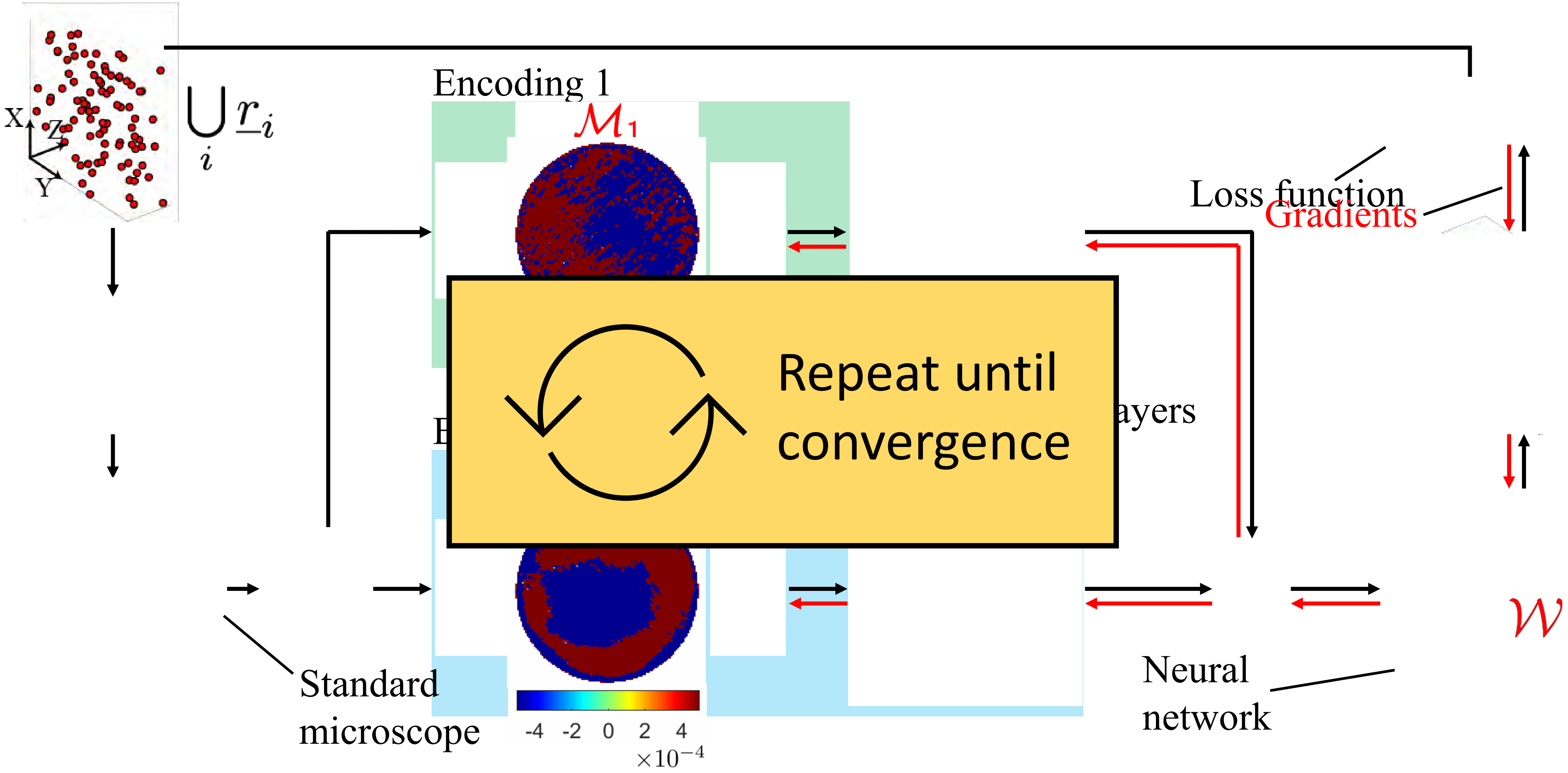
Kellman et al., *IEEE Trans. on Comp. Imaging* (2019)

Can we go beyond a single camera?



Nehme et al., arXiv (2020)

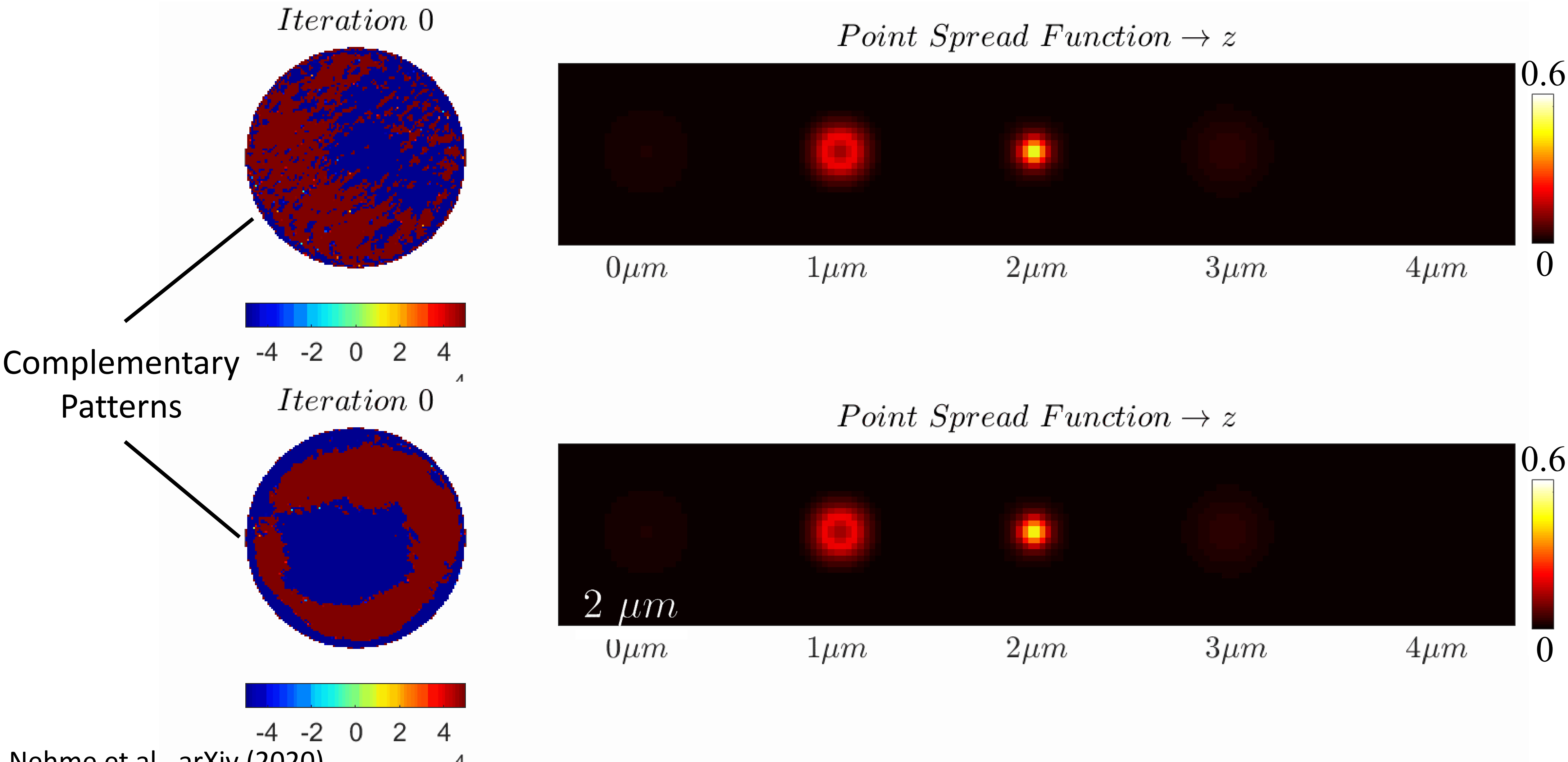
Design both cameras via backpropagation through physics



Nehme et al., arXiv (2020)

Fourier plane

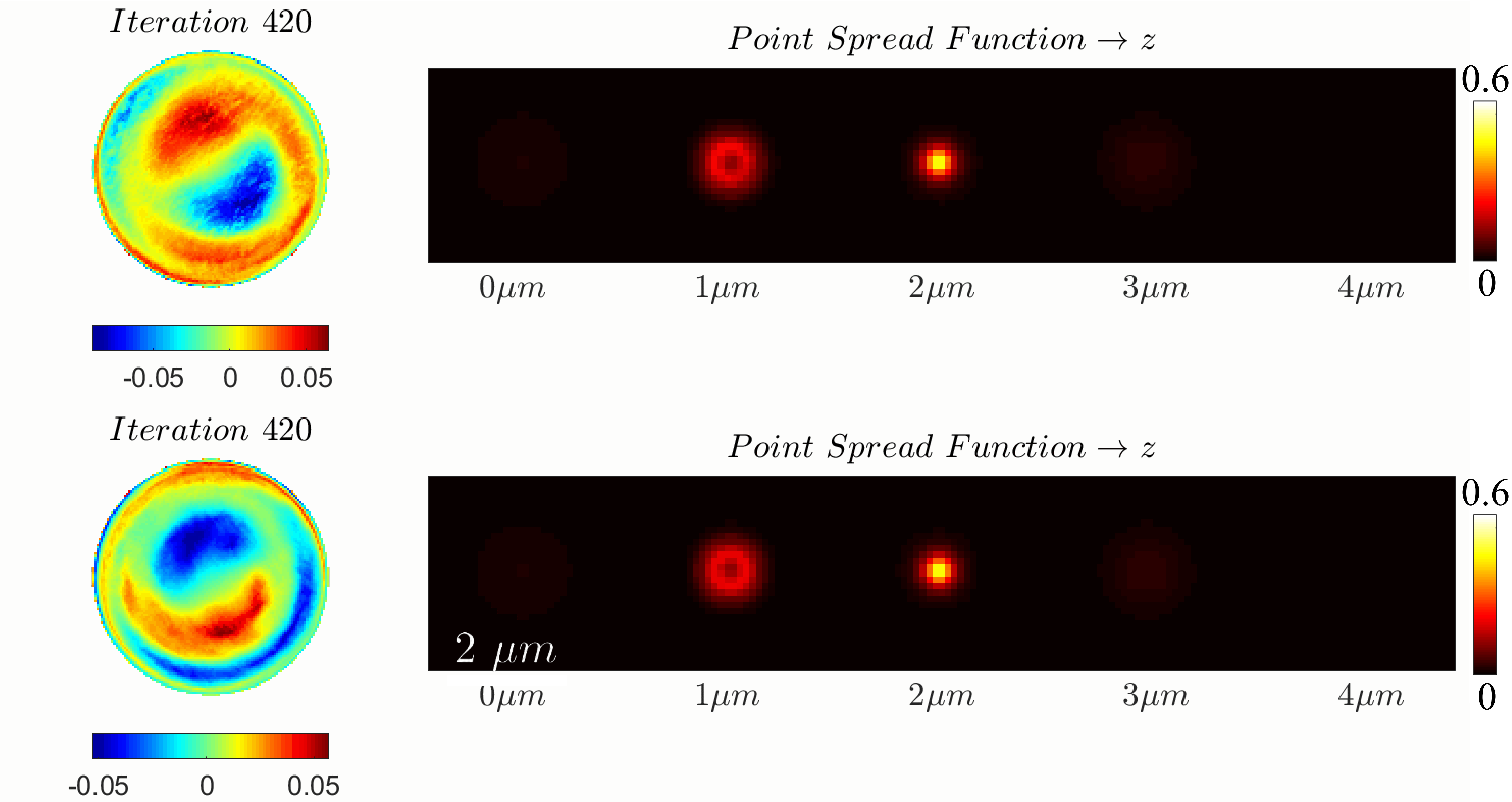
Image plane



Nehme et al., arXiv (2020)

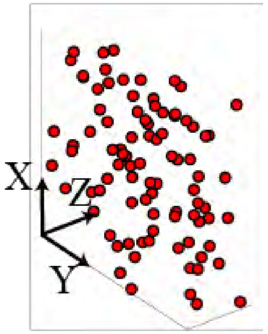
Fourier plane

Image plane



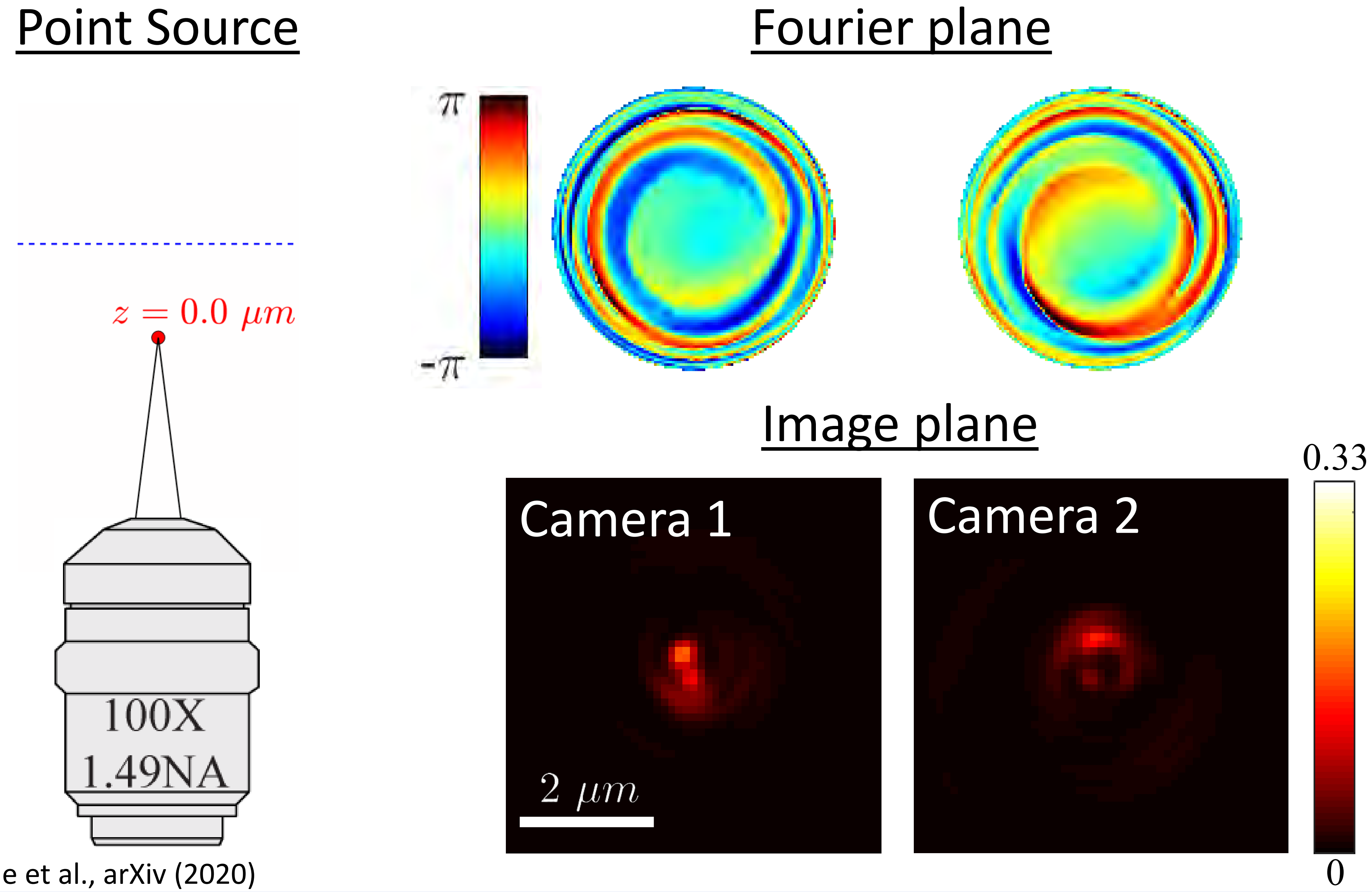
Nehme et al., arXiv (2020)

Jointly optimized "encoder-decoder"



$$\bigcup_i \underline{r}_i$$

Nehme et al., arXiv (2020)



Nehme et al., arXiv (2020)

Fourier plane

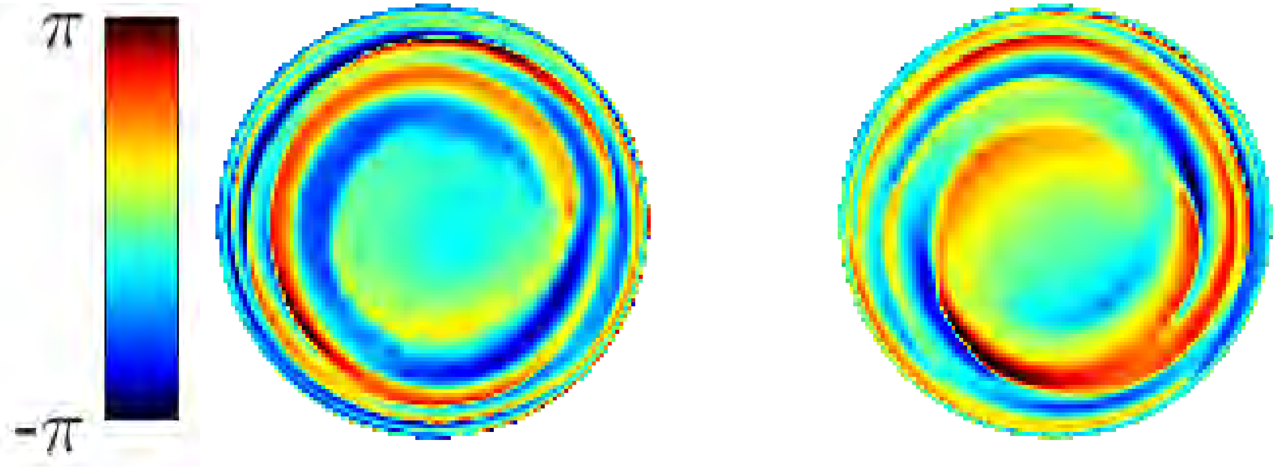
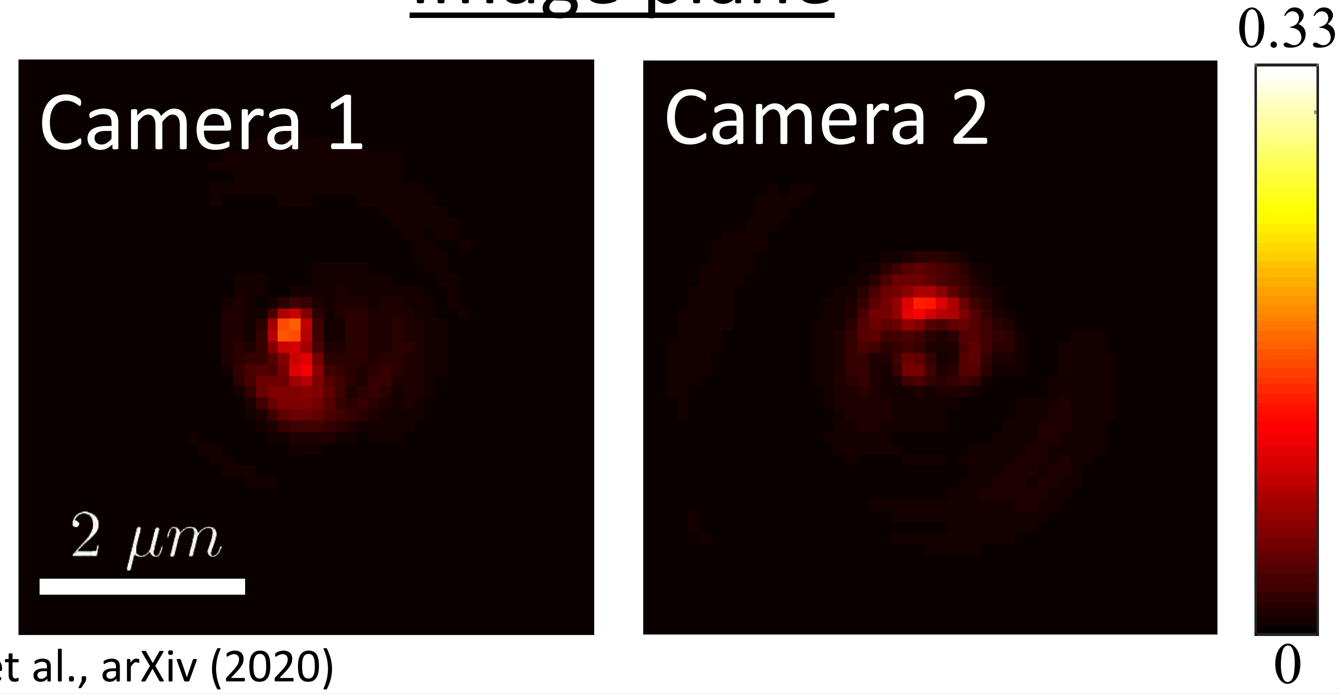
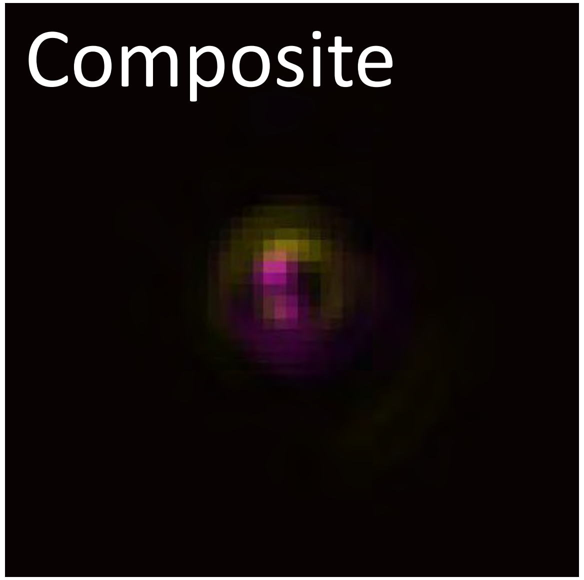


Image plane

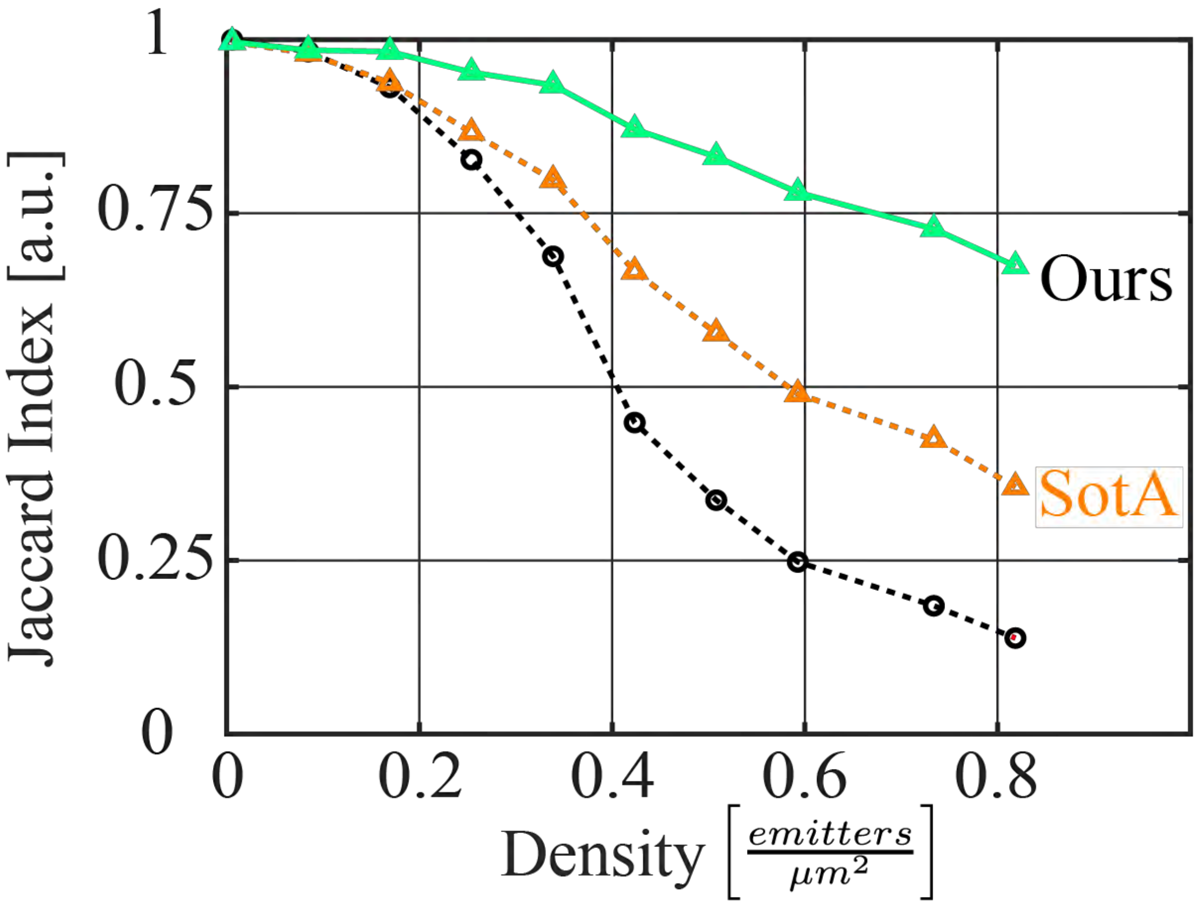


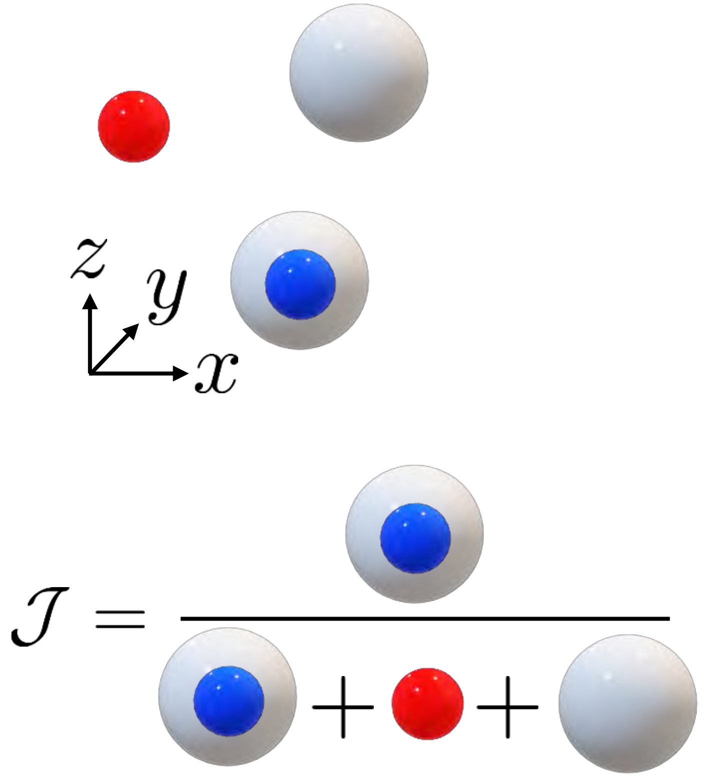
Complementary lobes



Nehme et al., arXiv (2020)

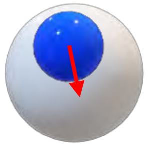
State-of-the-art results in detection

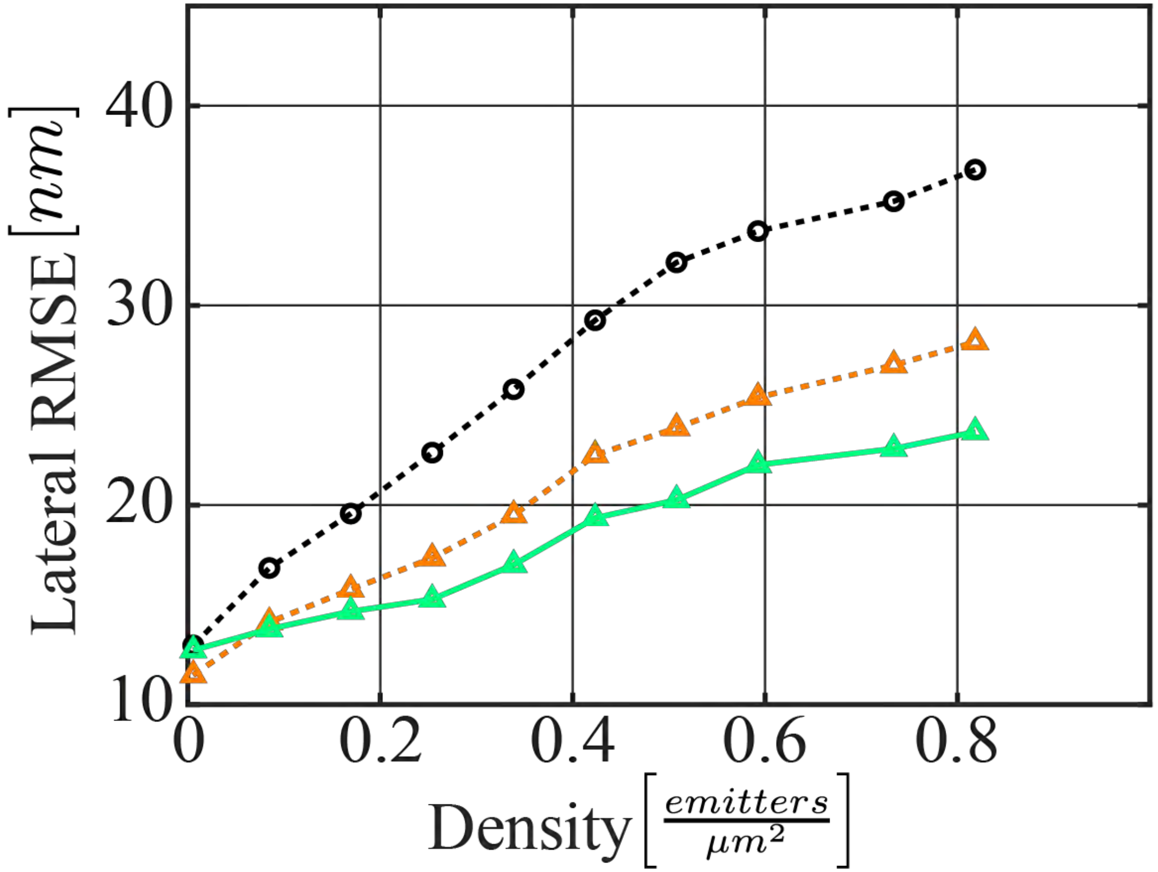


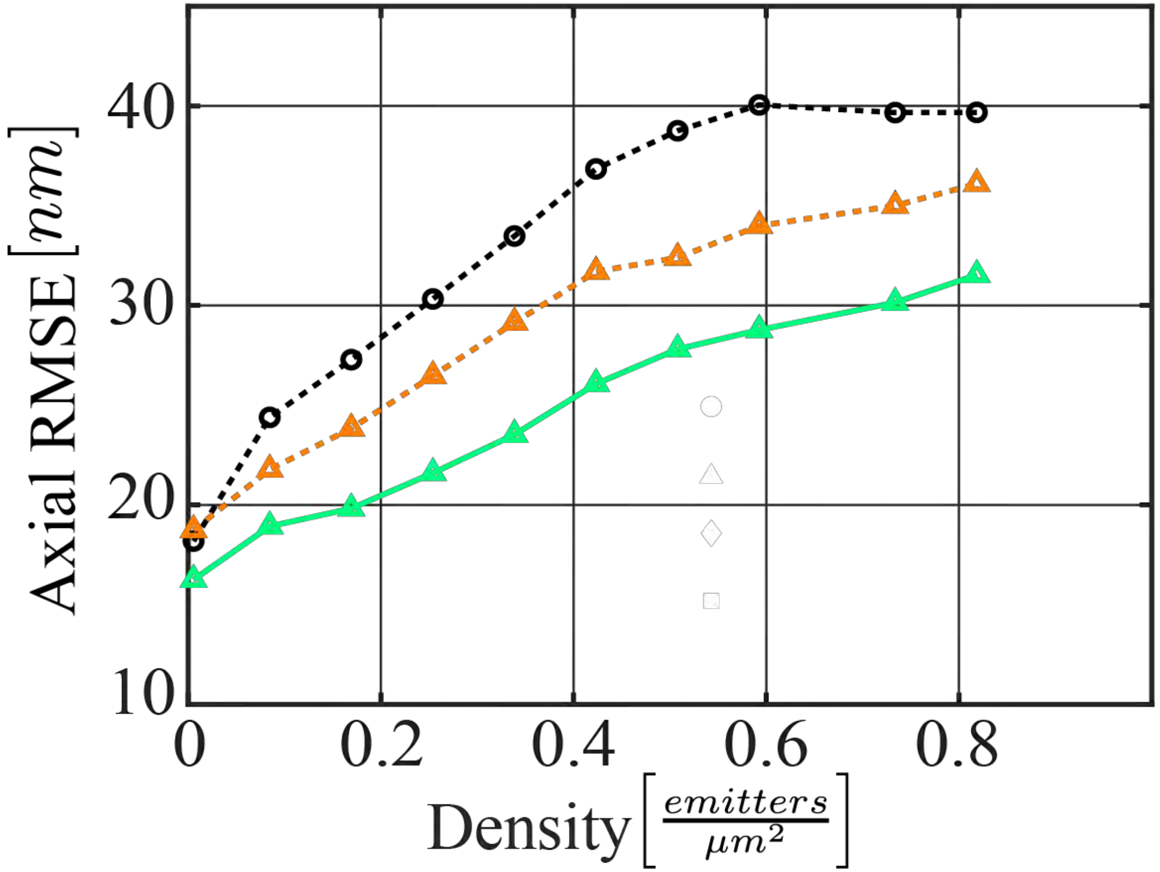


Nehme et al., arXiv (2020)

State-of-the-art results in precision

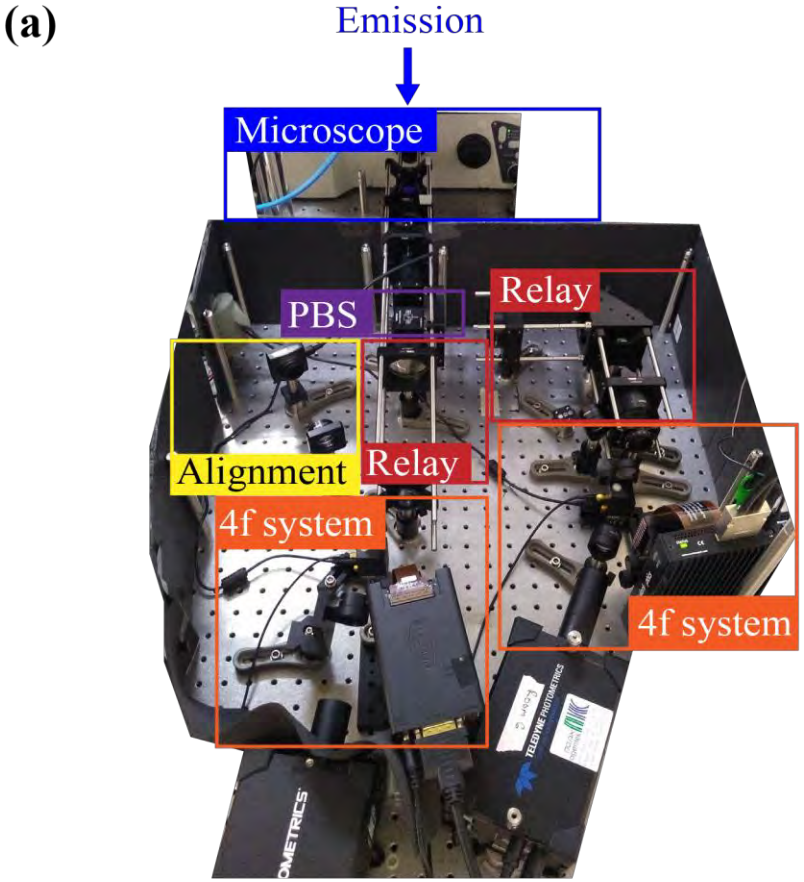


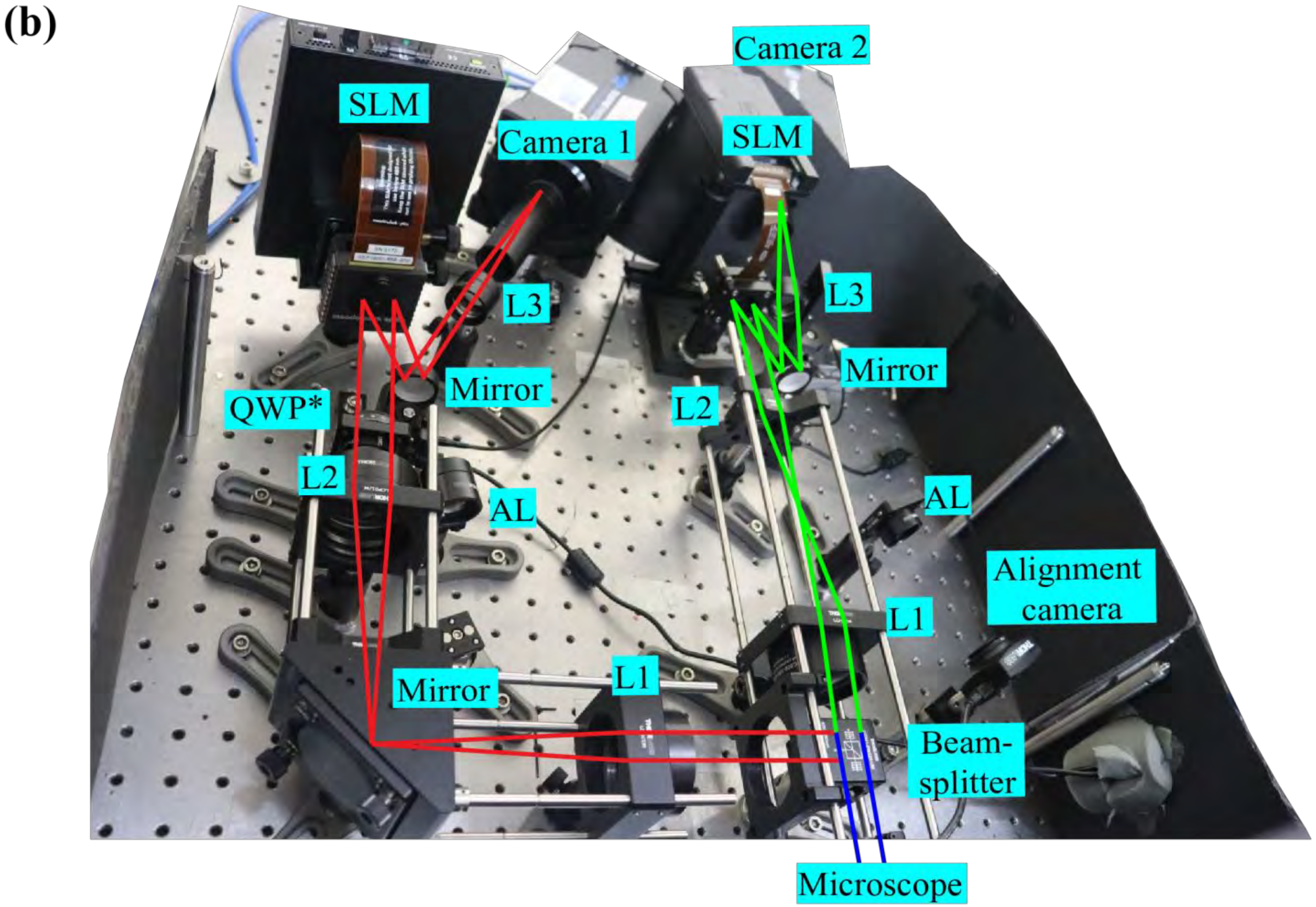




Nehme et al., arXiv (2020)

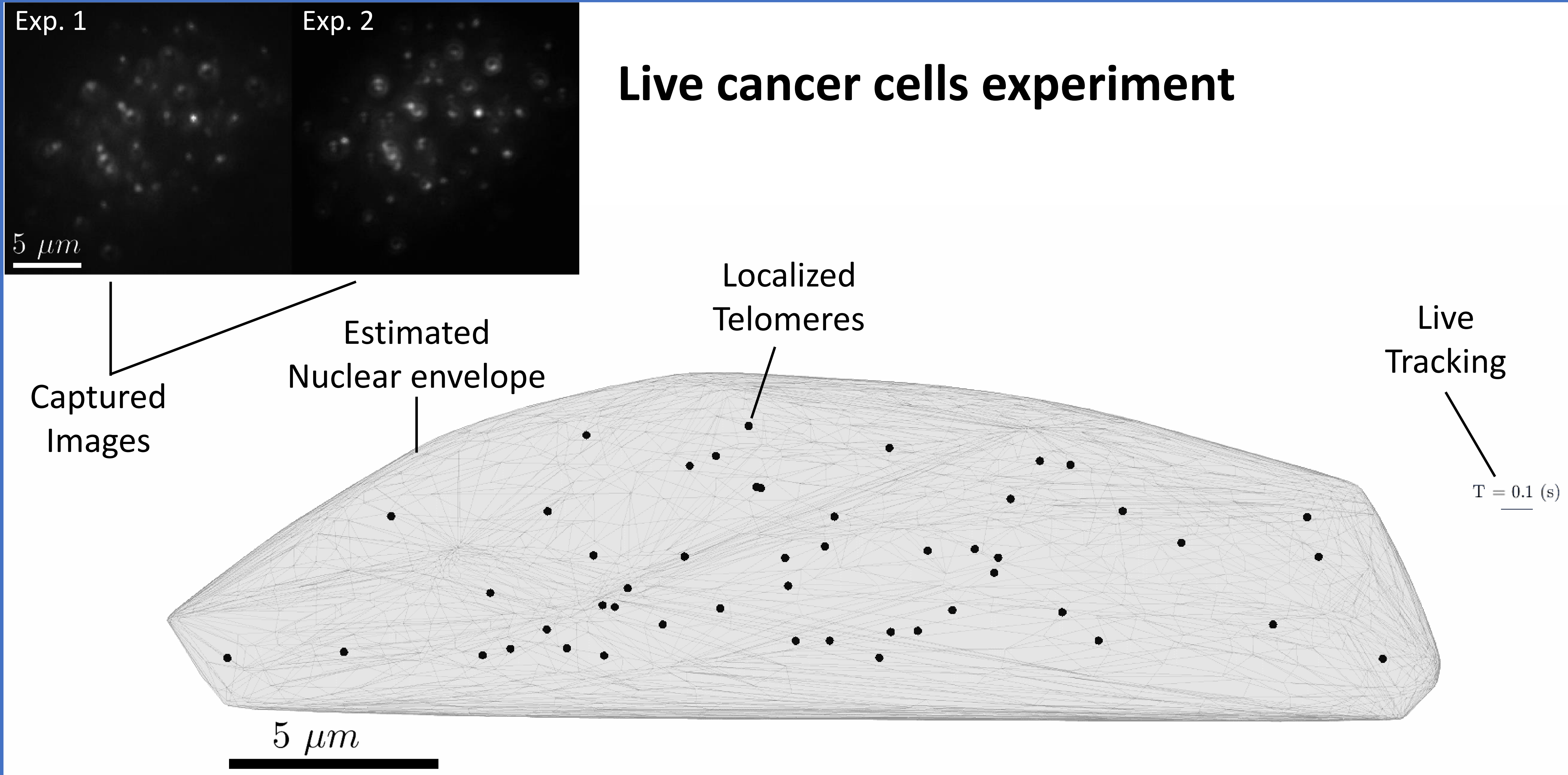
Experimental implementation with 2 LC-SLMs





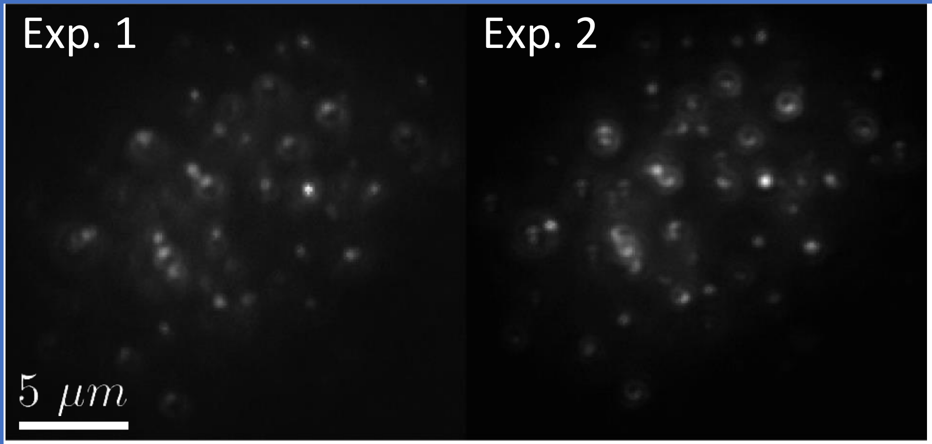
Nehme et al., arXiv (2020)

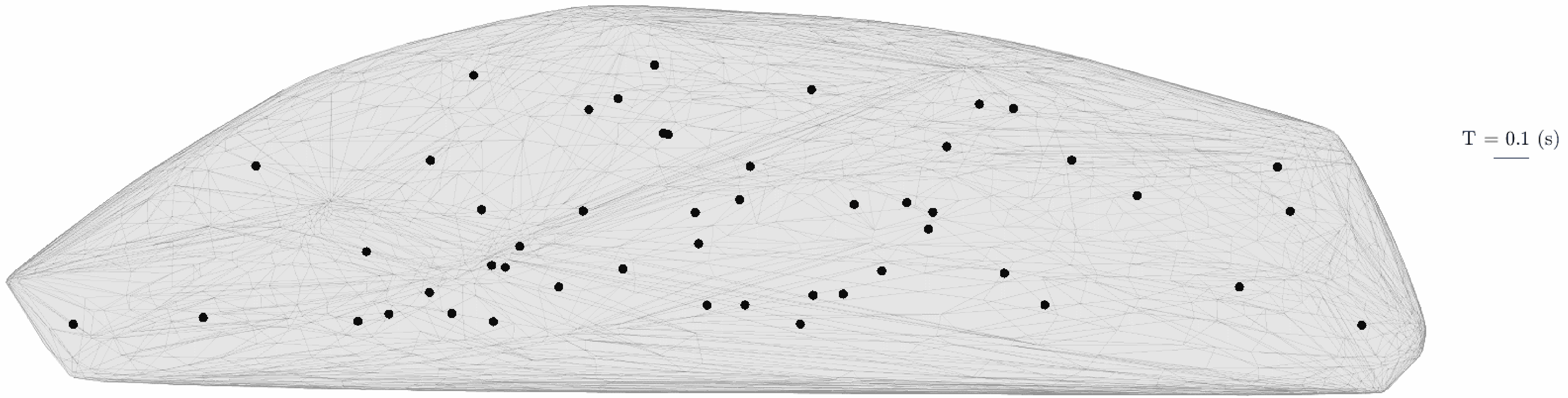
Live cancer cells experiment



Nehme et al., arXiv (2020)

Live 3D tracking of telomeres diffusing in a single cell nucleus



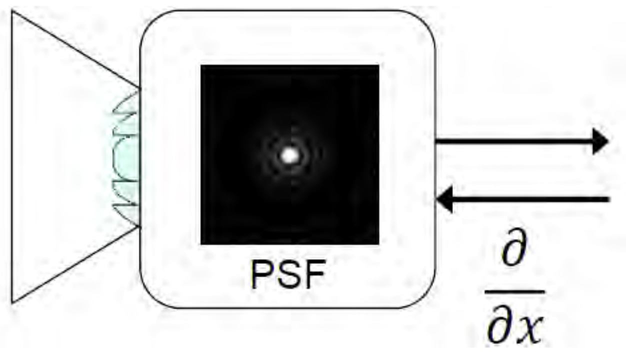


Nehme et al., arXiv (2020)

Outline

- 👍 Autoencoder interpretation
- 👍 Learning dense 3D imaging
- 👍 Generality to higher level tasks
- 👍 Multi-measurement systems
- 👉 Beyond microscopy

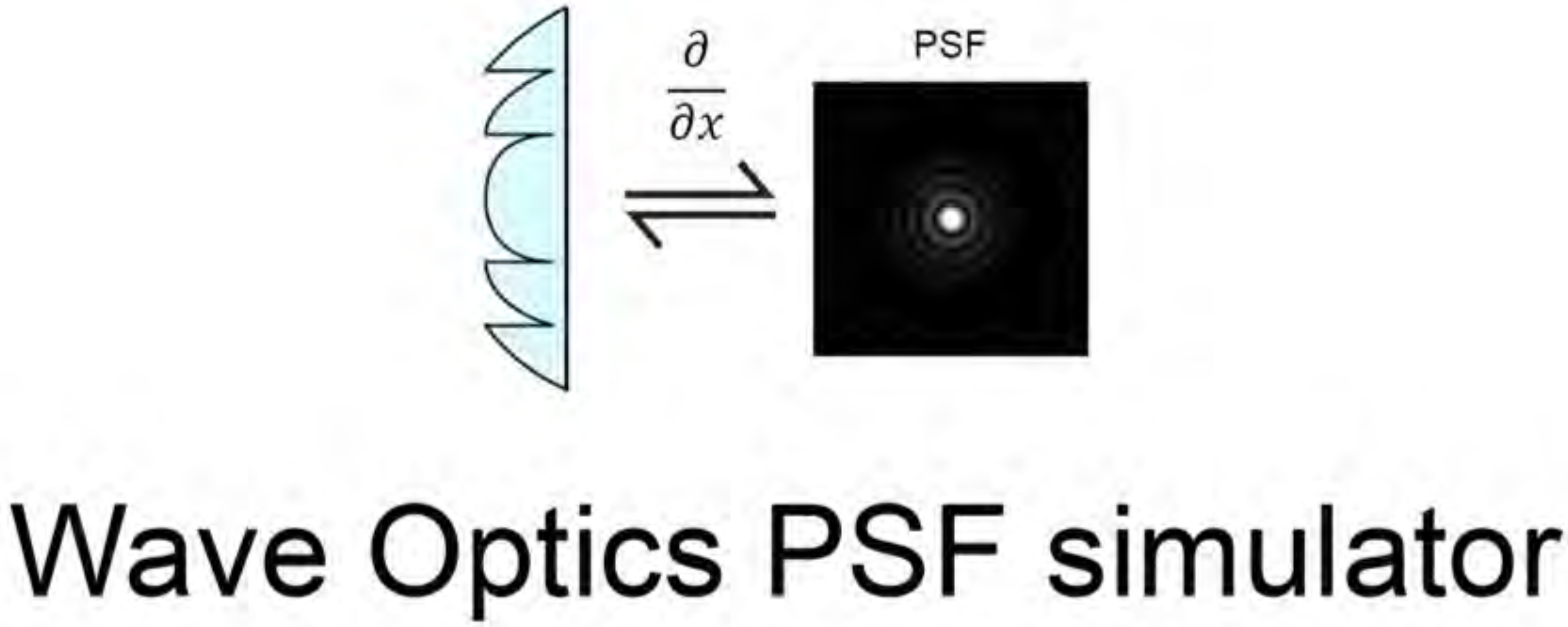
Optimizing phase masks for domain-specific cameras



A differentiable optics model

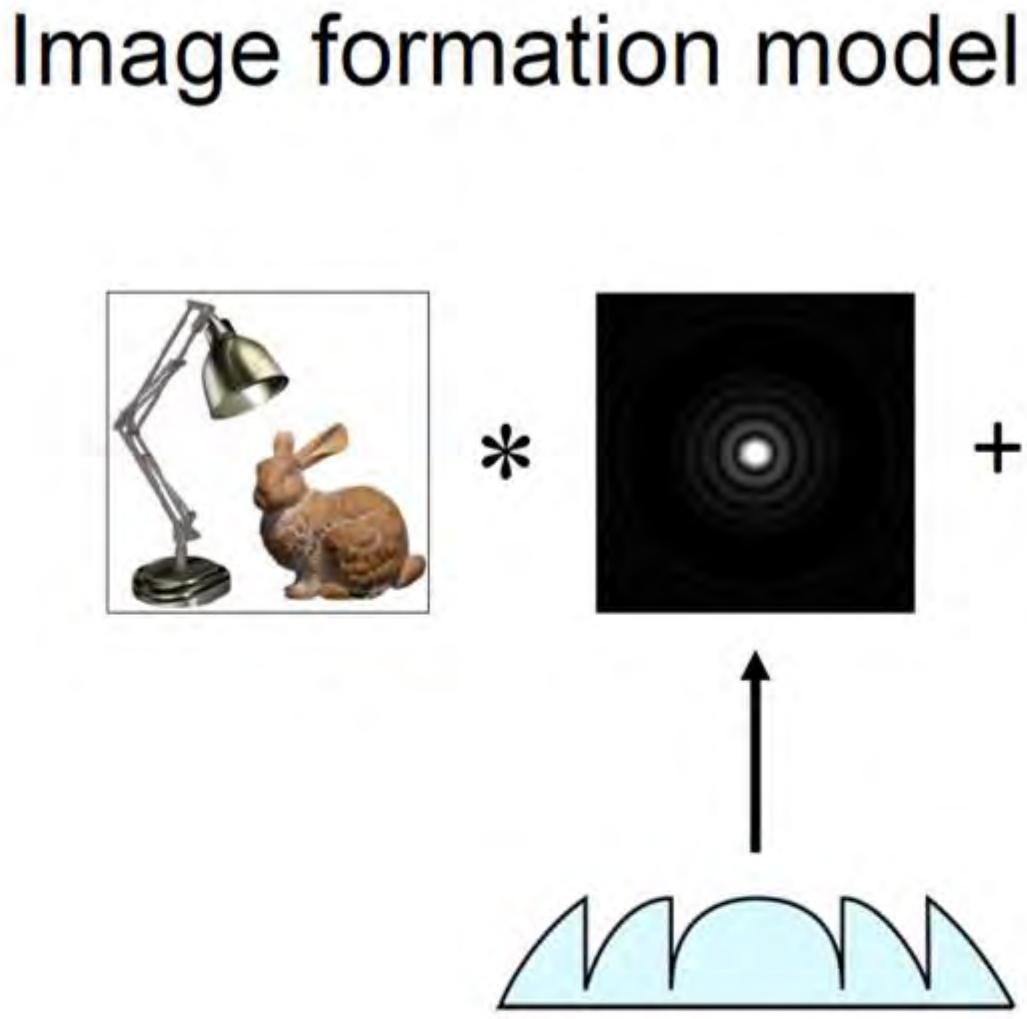
Sitzmann et al., SIGGRAPH (2018)

Optimizing phase masks for domain-specific cameras



Sitzmann et al., SIGGRAPH (2018)

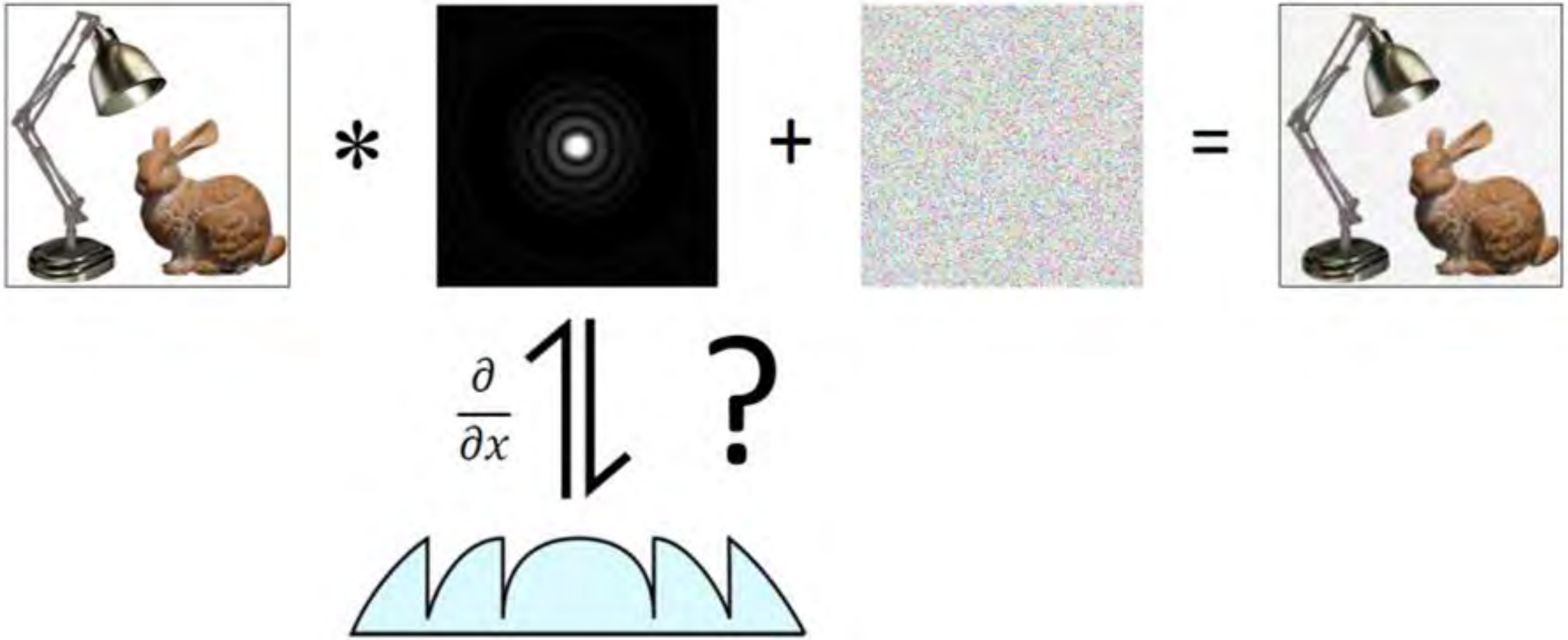
Optimizing phase masks for domain-specific cameras



Sitzmann et al., SIGGRAPH (2018)

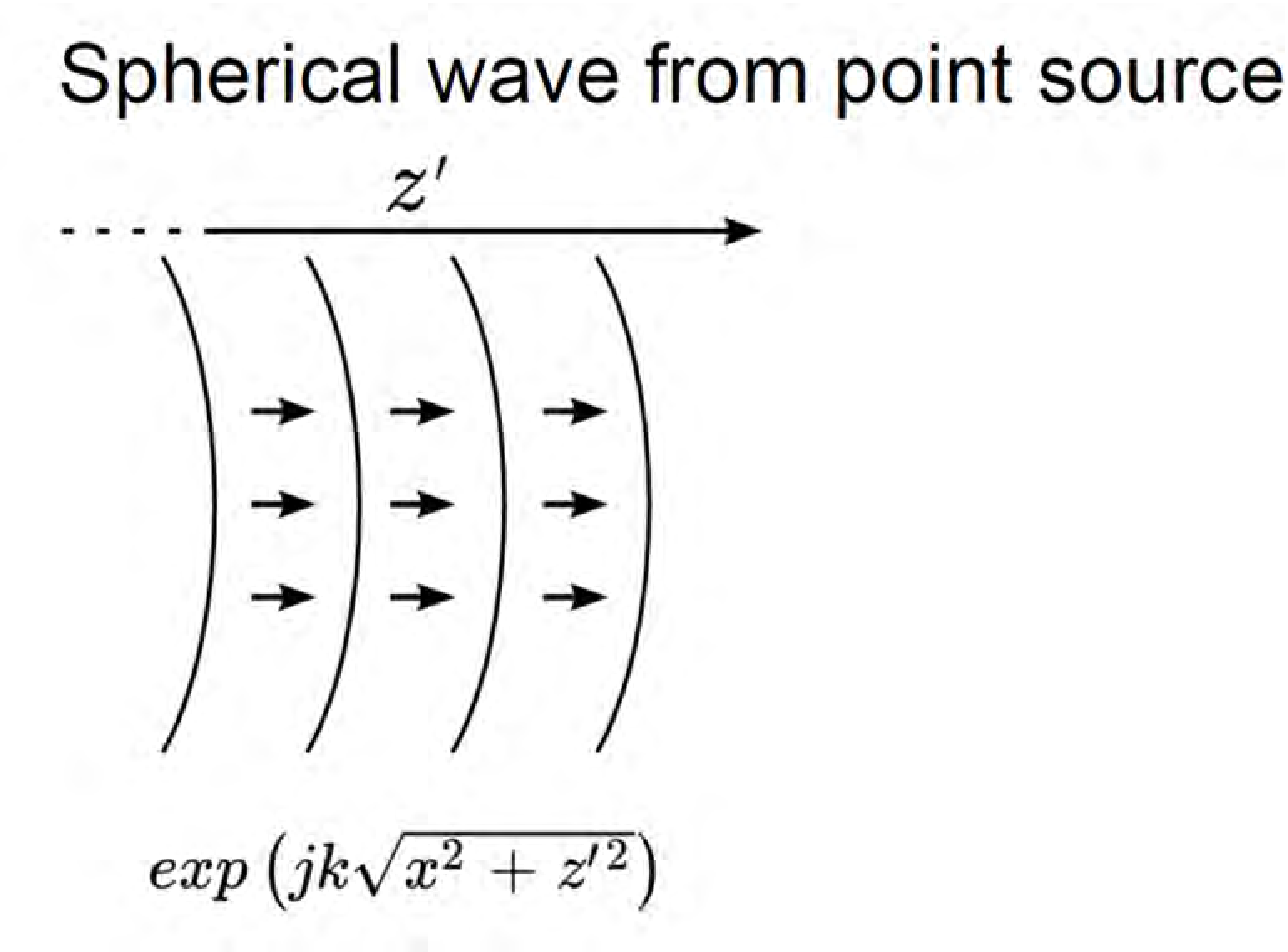
Optimizing phase masks for domain-specific cameras

How does the optical element map to the PSF?



Sitzmann et al., SIGGRAPH (2018)

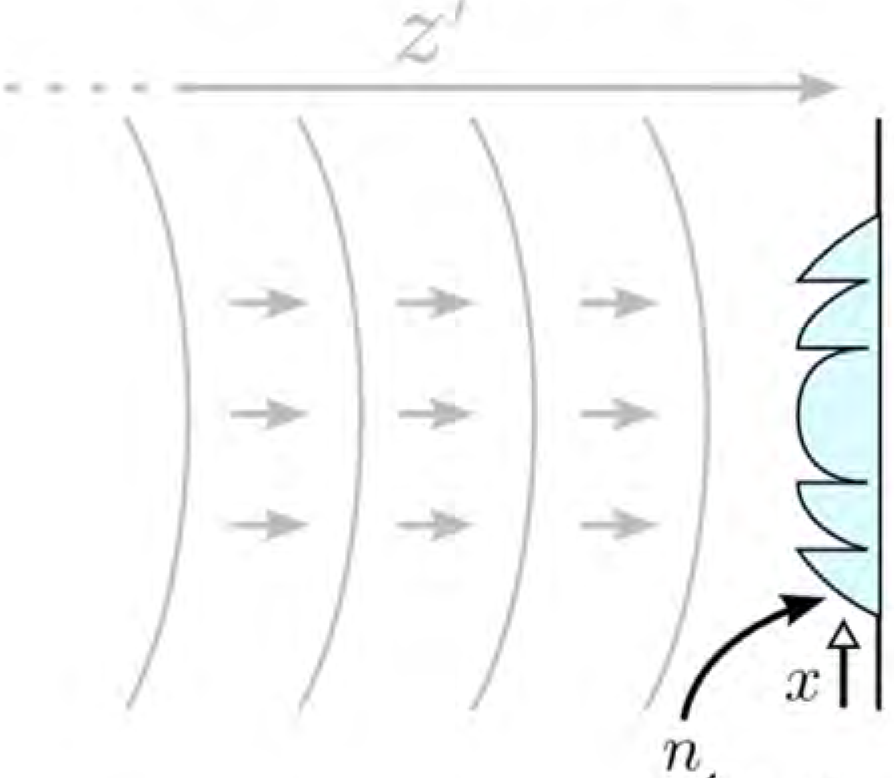
Optimizing phase masks for domain-specific cameras



Sitzmann et al., SIGGRAPH (2018)

Optimizing phase masks for domain-specific cameras

Phaseshift by optical element

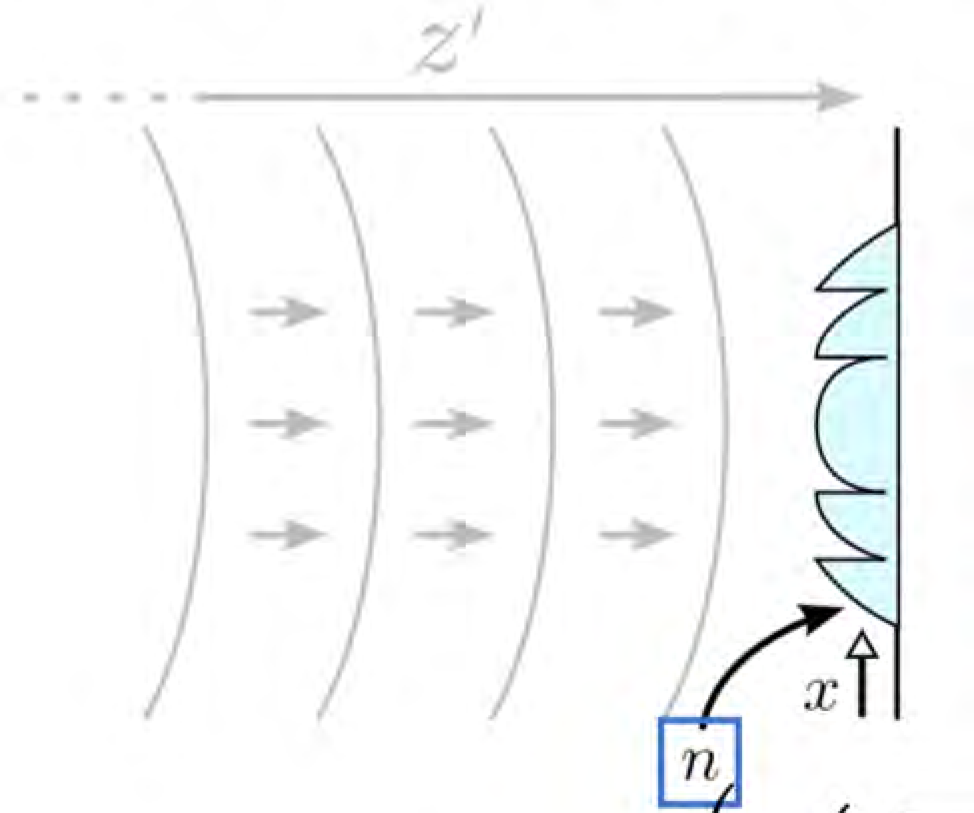


$$U(x) = \exp\left(jk\left(\sqrt{x^2 + z'^2} + (n-1)\Phi(\mathrm{x})\right)\right)$$

Sitzmann et al., SIGGRAPH (2018)

Optimizing phase masks for domain-specific cameras

Phaseshift by optical element

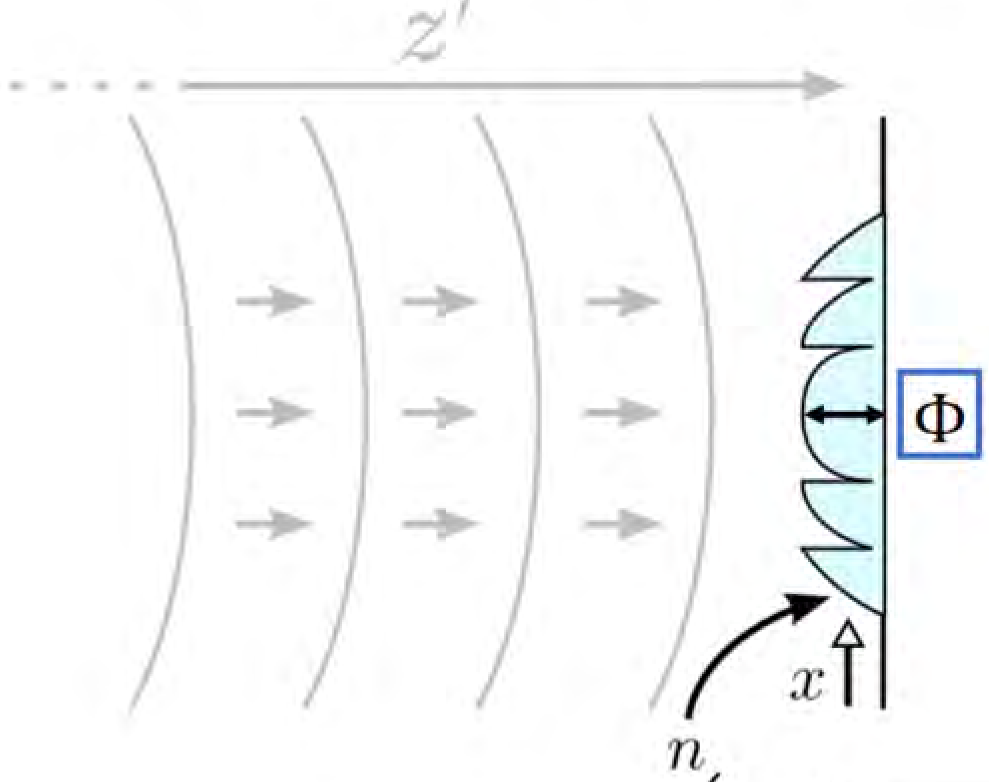


$$U(x) = \exp\left(jk\left(\sqrt{x^2 + z'^2} + (\boxed{n} - 1)\Phi(\mathrm{x})\right)\right)$$

Sitzmann et al., SIGGRAPH (2018)

Optimizing phase masks for domain-specific cameras

Phaseshift by optical element

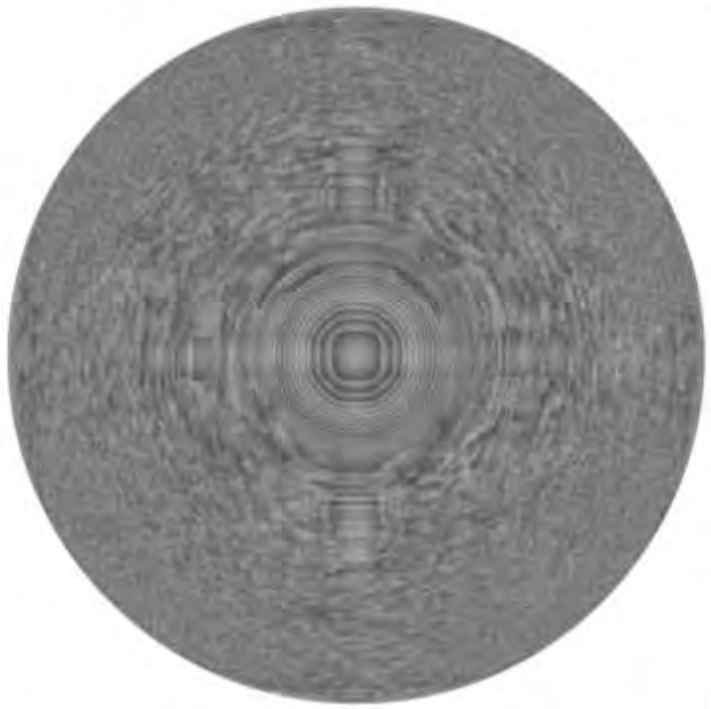


$$U(x) = \exp\left(jk\left(\sqrt{x^2 + z'^2} + (n-1)\boxed{\Phi}(\mathrm{x})\right)\right)$$

Sitzmann et al., SIGGRAPH (2018)

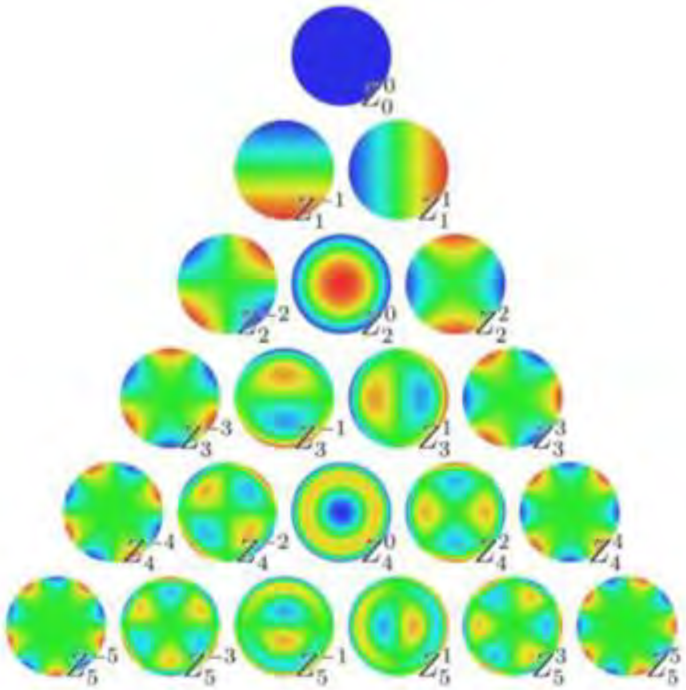
Optimizing phase masks for domain-specific cameras

Height Map parameterization (diffractive)



$$\Phi[x] = [[a_{11}, a_{12}, \ldots], \ldots]$$

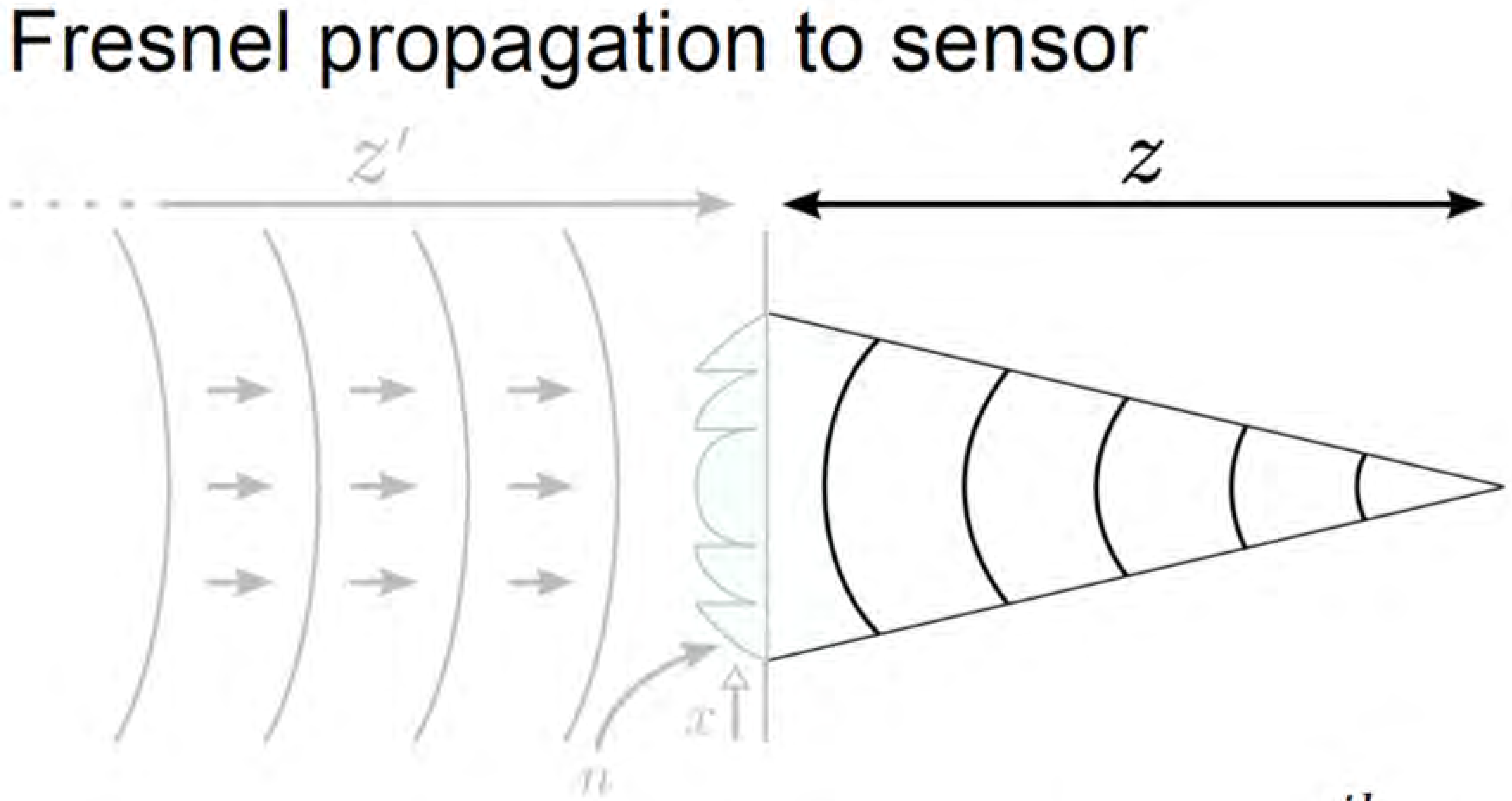
Zernike basis parameterization (refractive)



$$\Phi[x] = \sum Z_i^j[x] \cdot a_{ij}$$

Sitzmann et al., SIGGRAPH (2018)

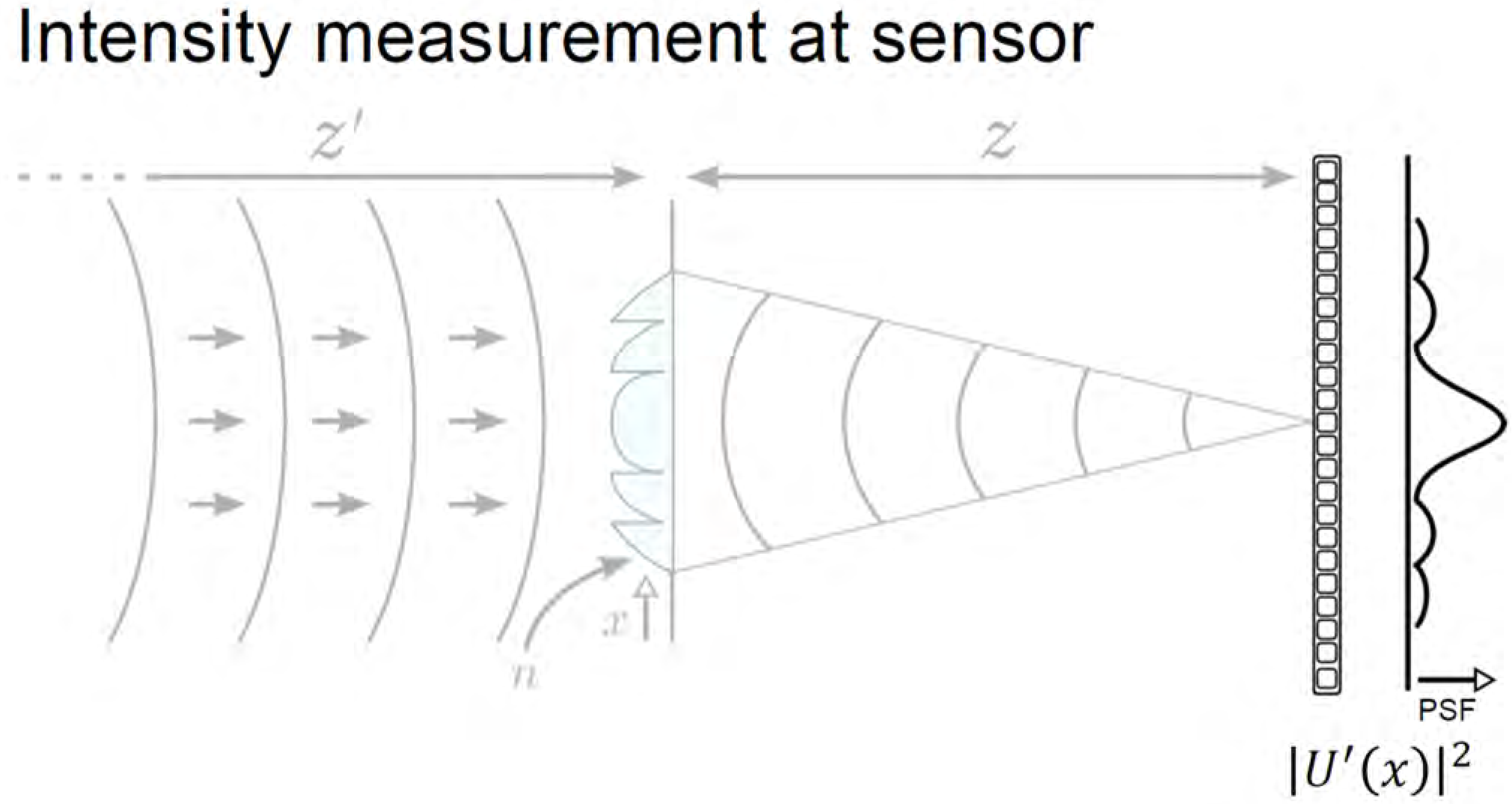
Optimizing phase masks for domain-specific cameras



$$U'(x) = U(x) * \exp(\frac{jk}{2z} x^2)$$

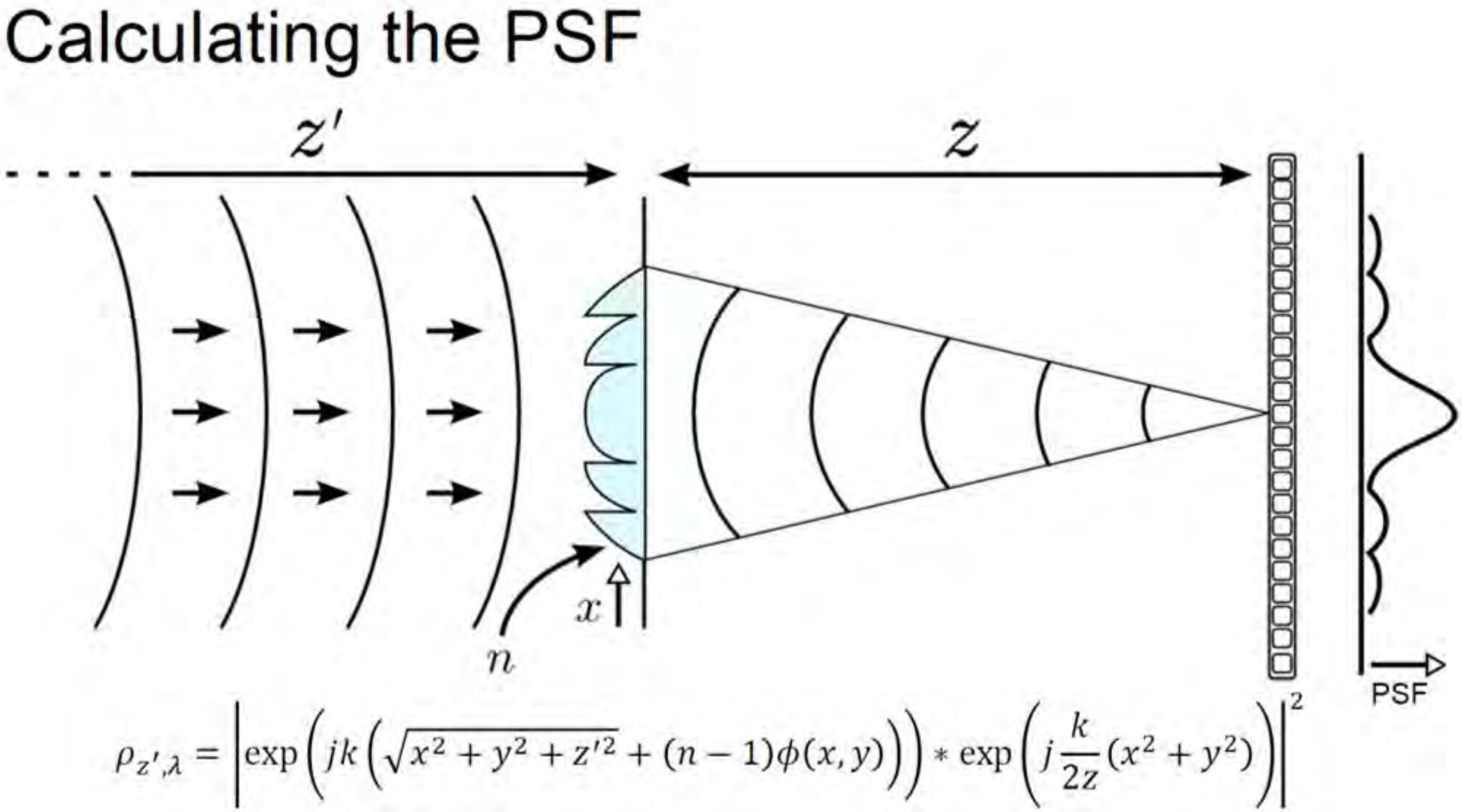
Sitzmann et al., SIGGRAPH (2018)

Optimizing phase masks for domain-specific cameras



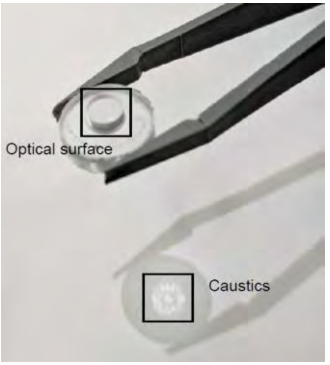
Sitzmann et al., SIGGRAPH (2018)

Optimizing phase masks for domain-specific cameras



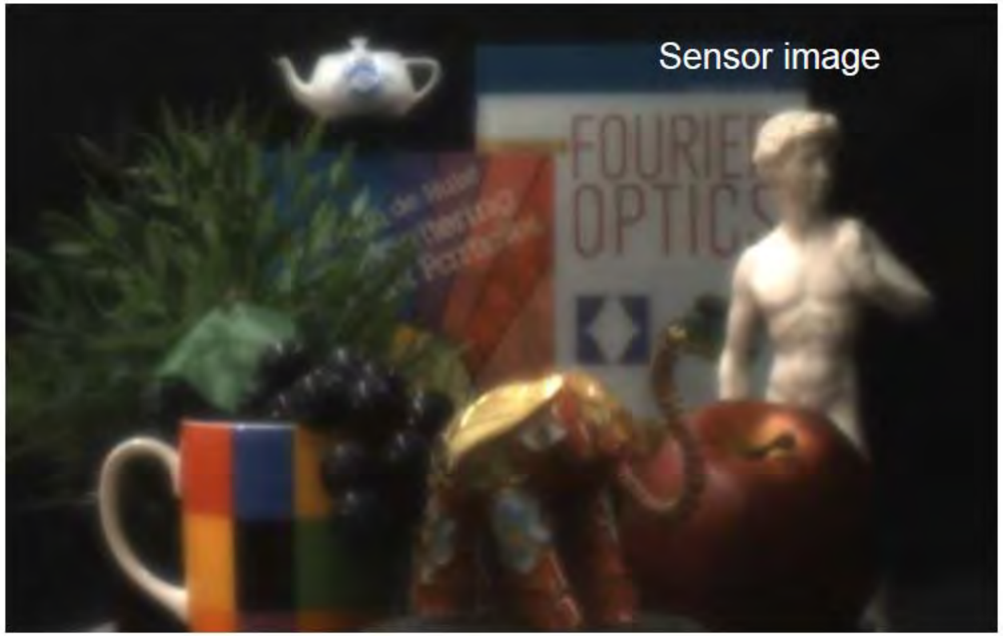
Sitzmann et al., SIGGRAPH (2018)

Extended Depth of Field (EDOF) imaging



Test scene

Regular bi-convex lens

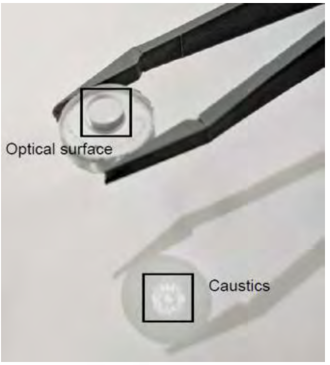


Optimized lens

Elephant (0.5m) ⋯⋯⋯▸ Book (2.0m)

Sitzmann et al., SIGGRAPH (2018)

Extended Depth of Field (EDOF) imaging



Test scene

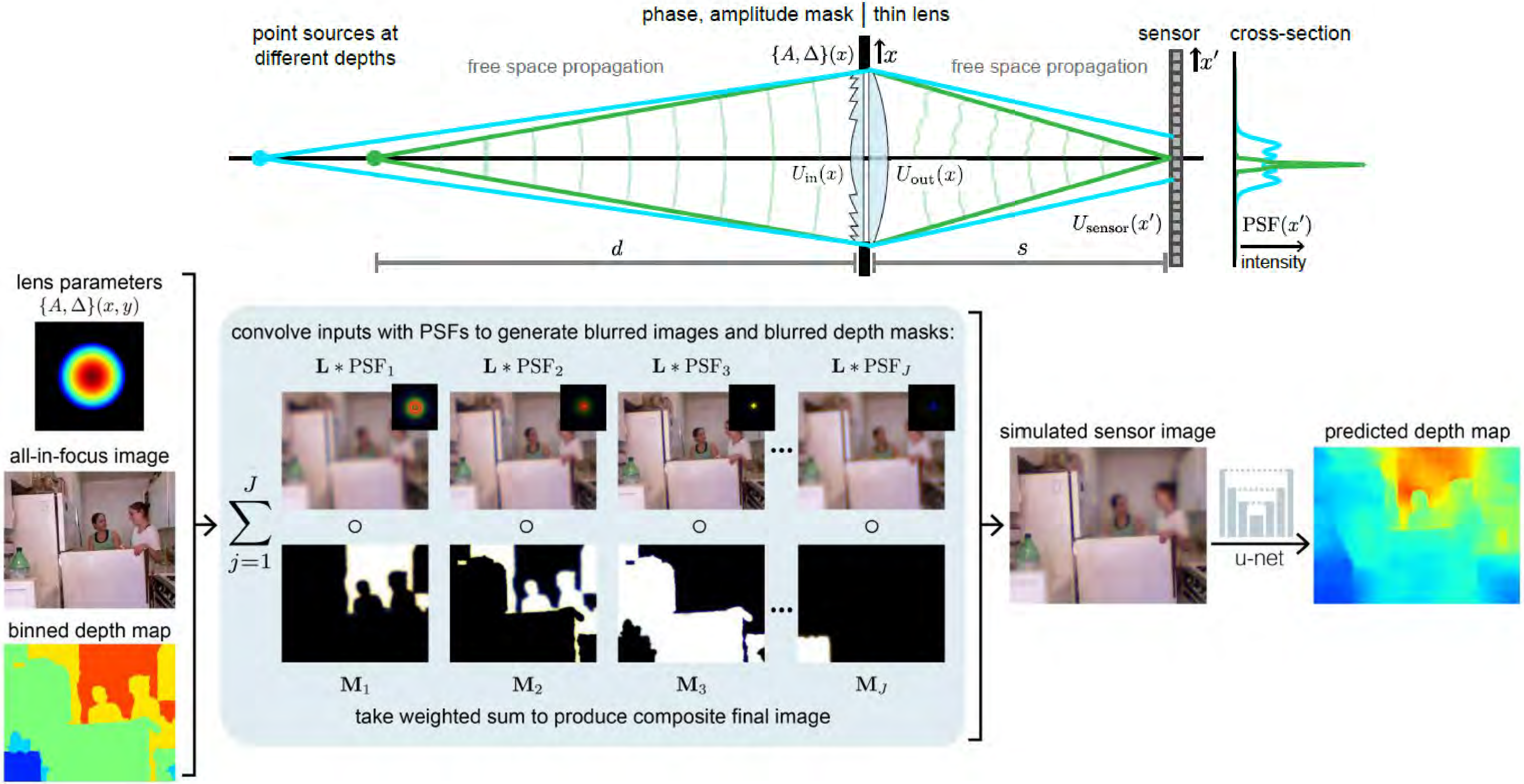
Regular bi-convex lens

Optimized lens

Elephant (0.5m) → Book (2.0m)

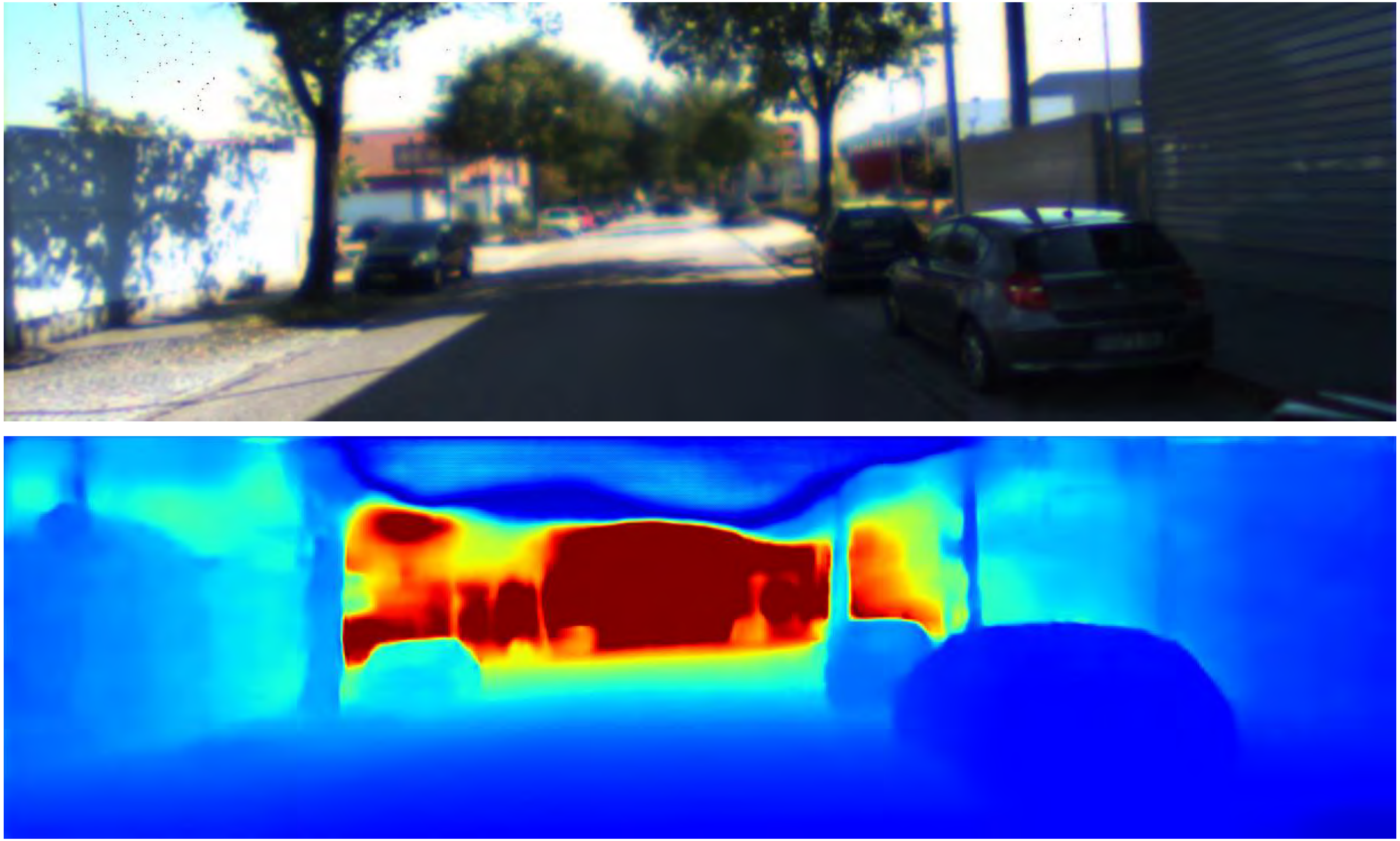
Sitzmann et al., SIGGRAPH (2018)

Monocular depth estimation and 3D object detection



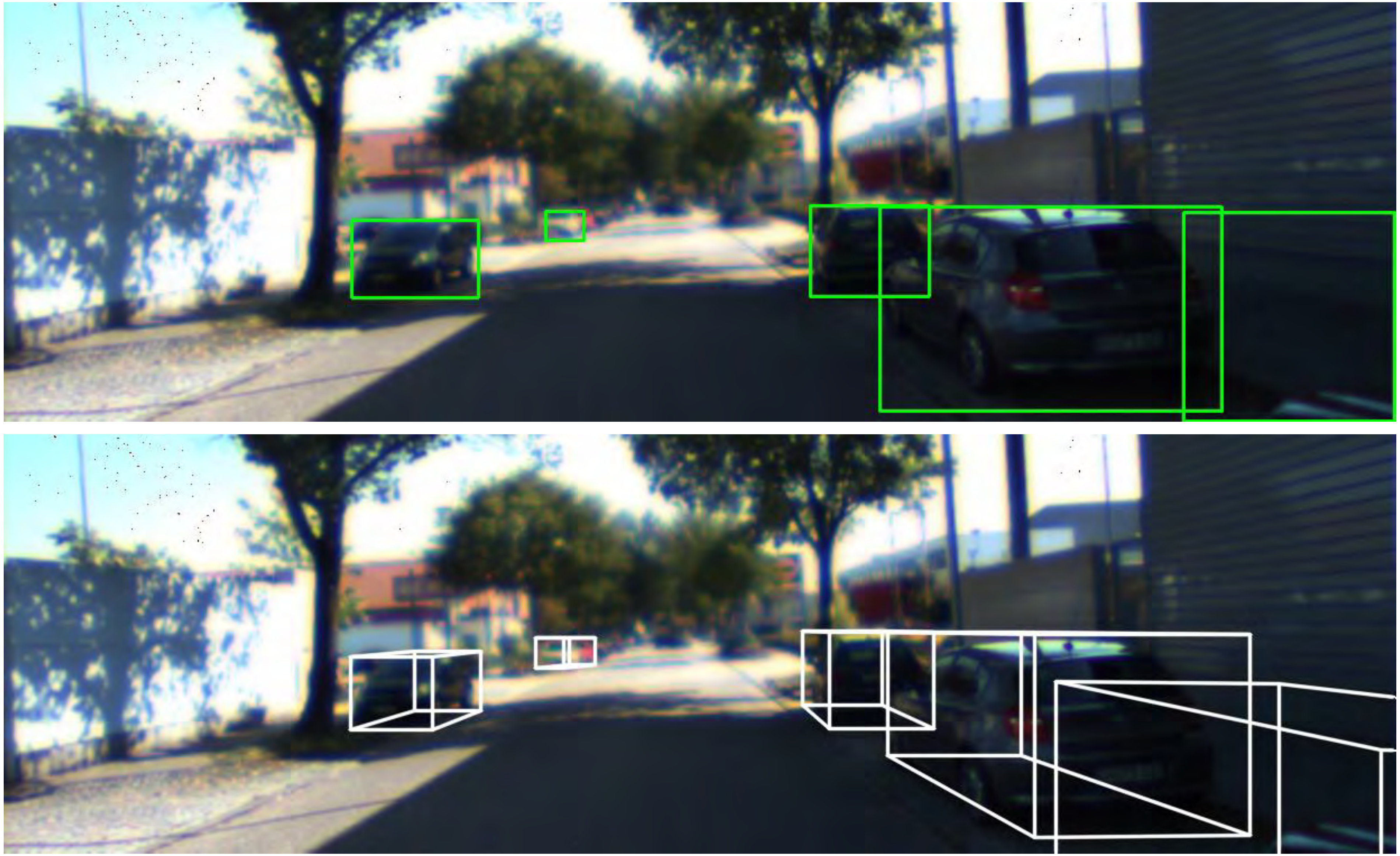
Chang et al., ICCV (2019)

Monocular depth estimation and 3D object detection



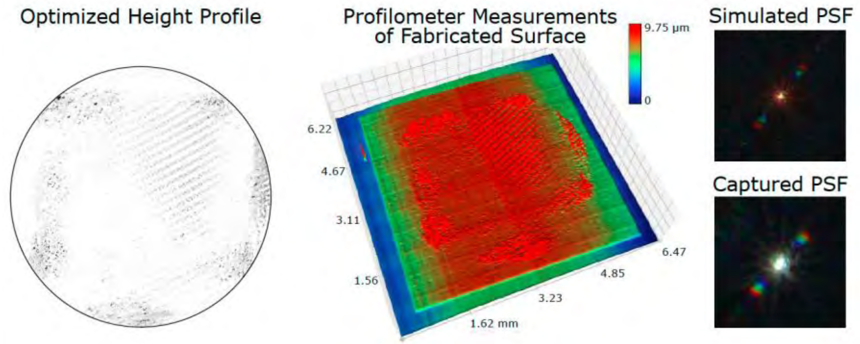
Chang et al., ICCV (2019)

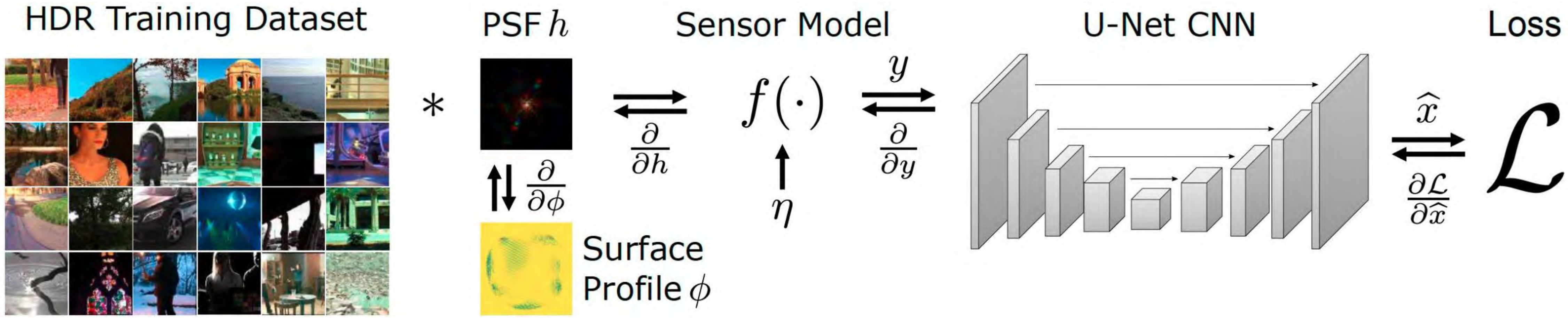
Monocular depth estimation and 3D object detection

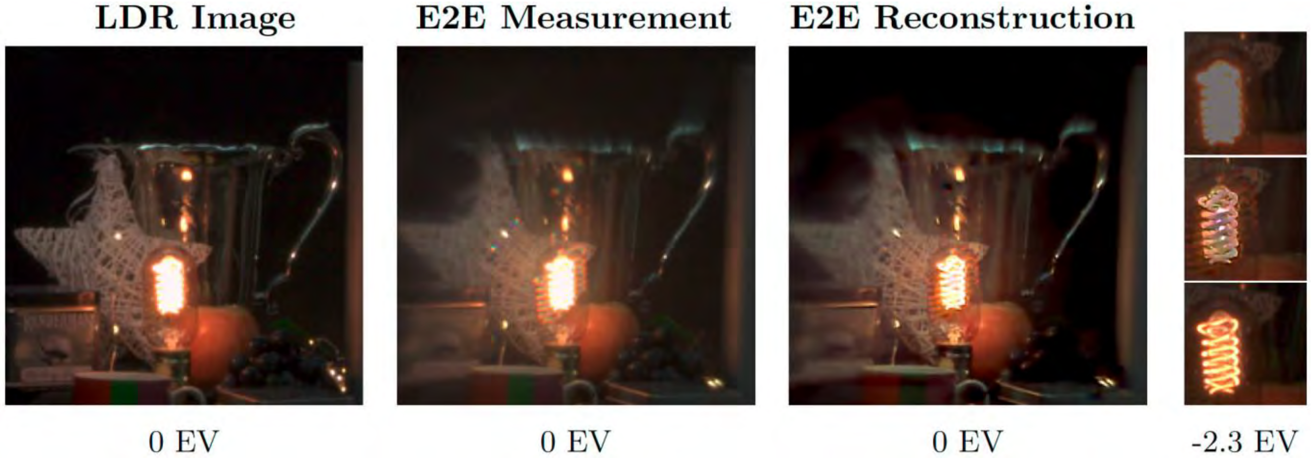


Chang et al., ICCV (2019)

High-dynamic range imaging

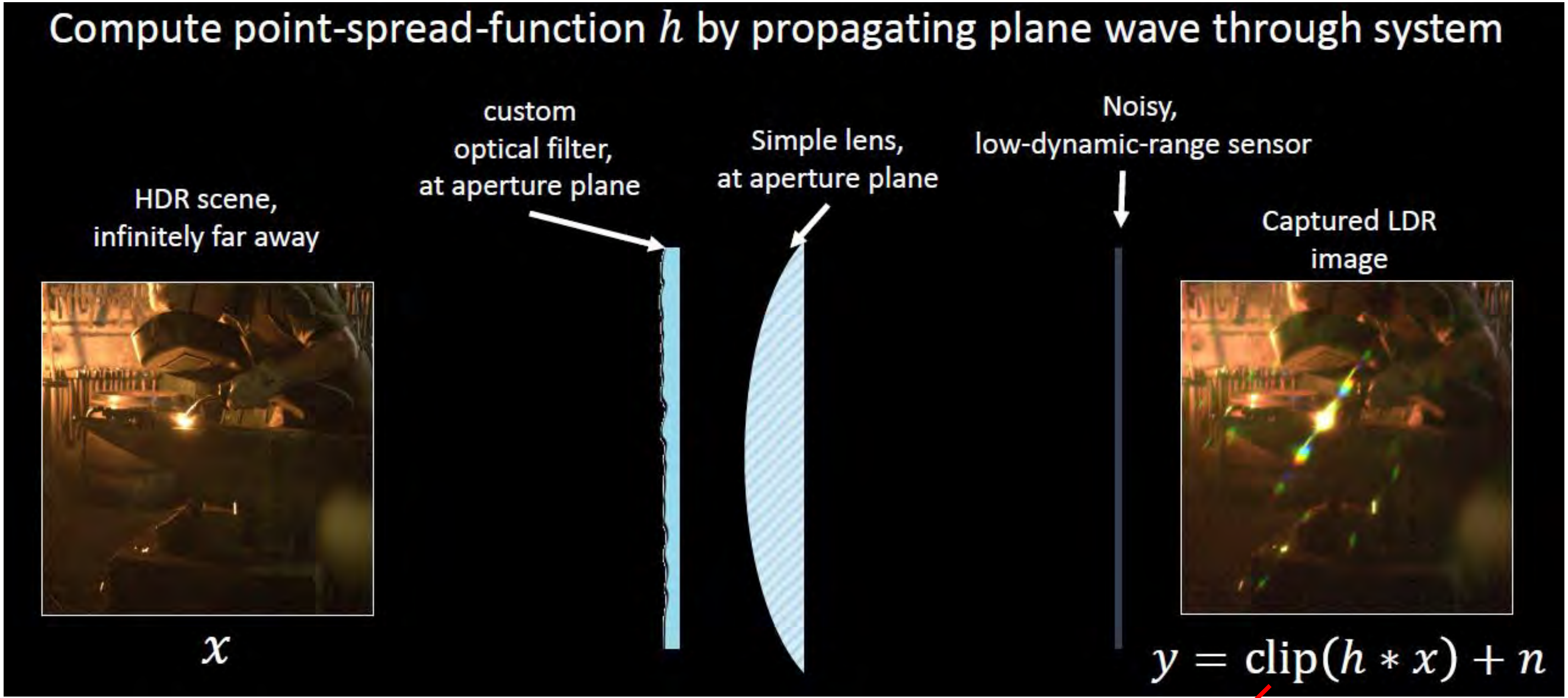






Meltzer et al., CVPR (2020)

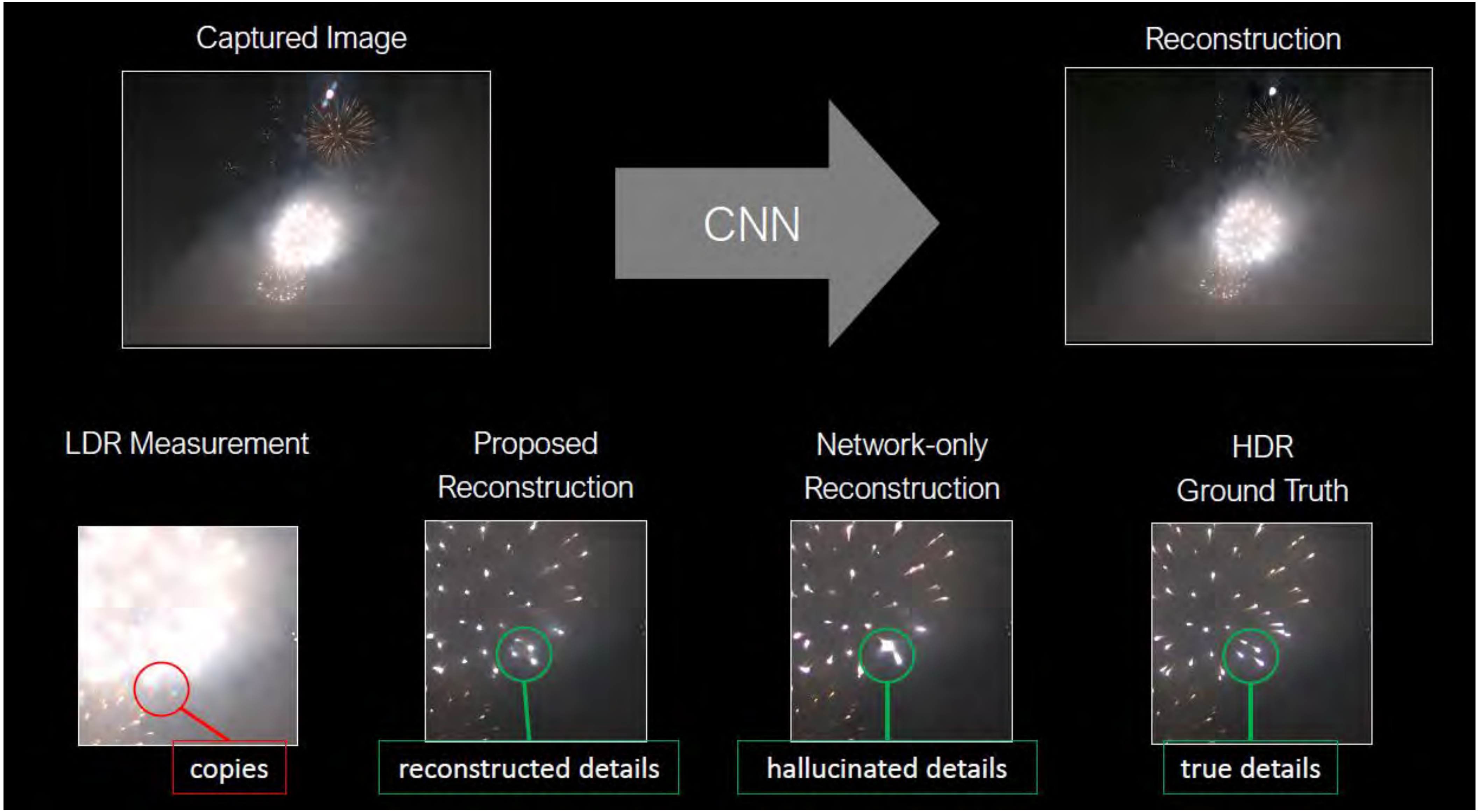
High-dynamic range imaging



Sensor model
includes clipping

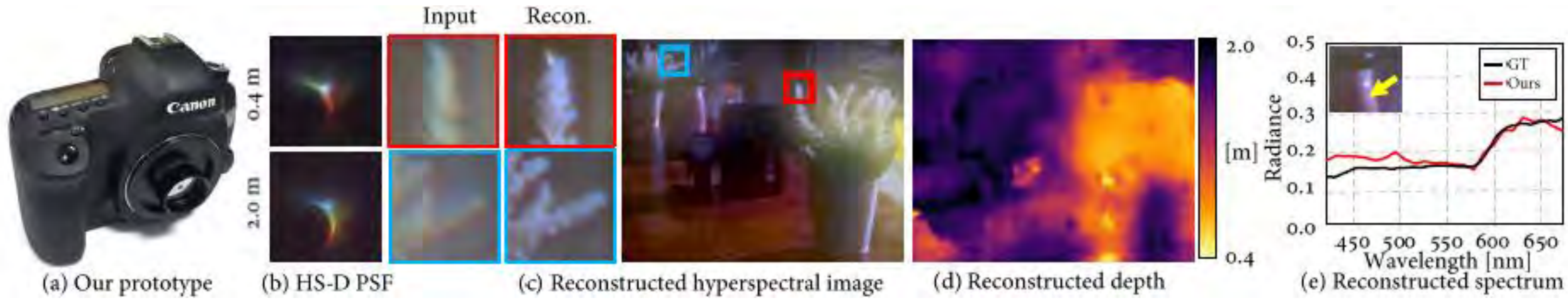
Meltzer et al., CVPR (2020)

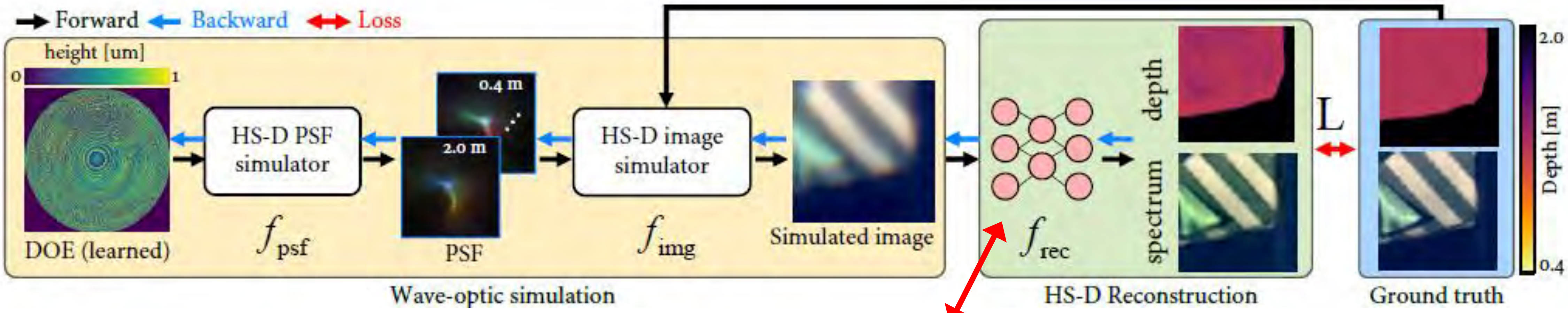
High-dynamic range imaging

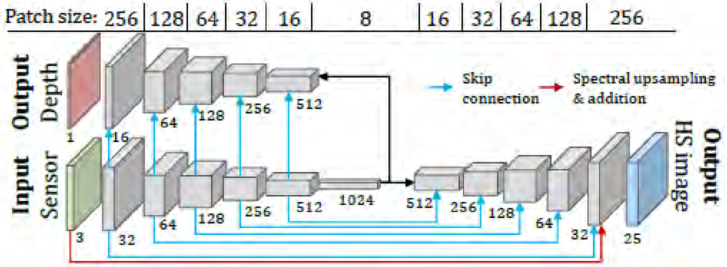


Meltzer et al., CVPR (2020)

Hyper-spectral and depth imaging



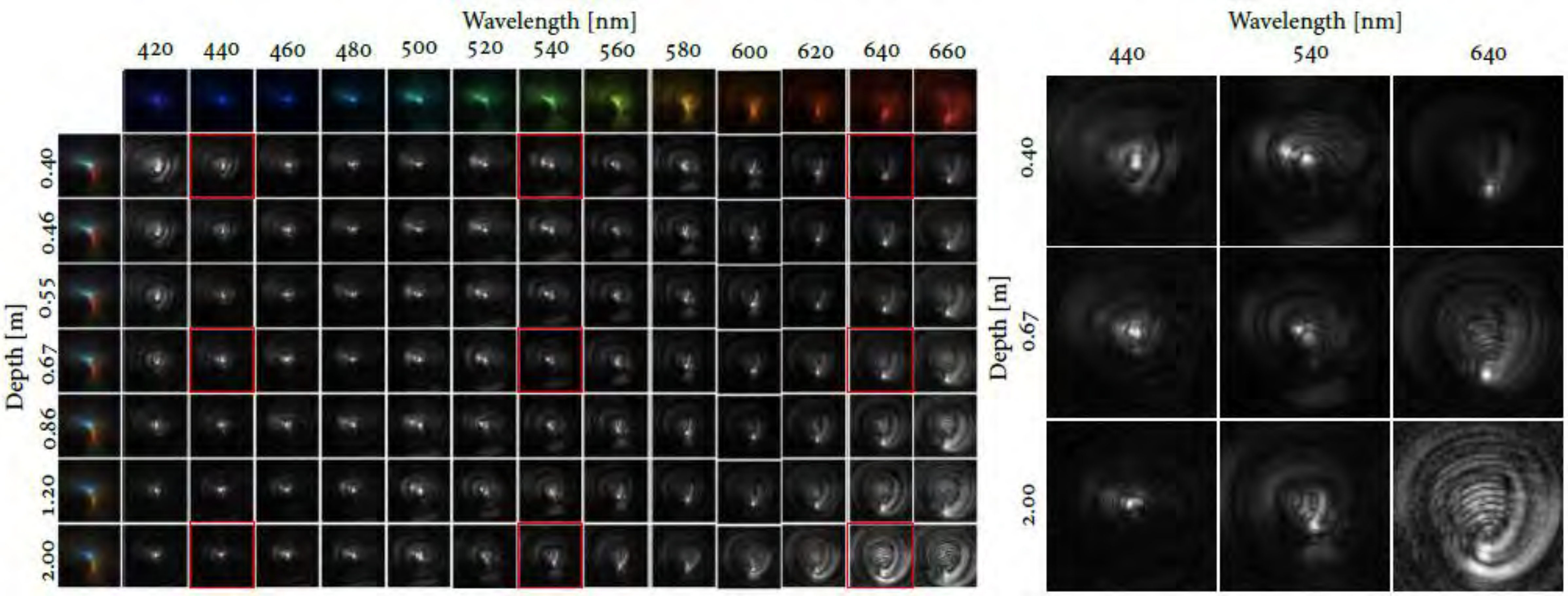




Baek et al., SIGGRAPH (2020)

Hyper-spectral and depth imaging

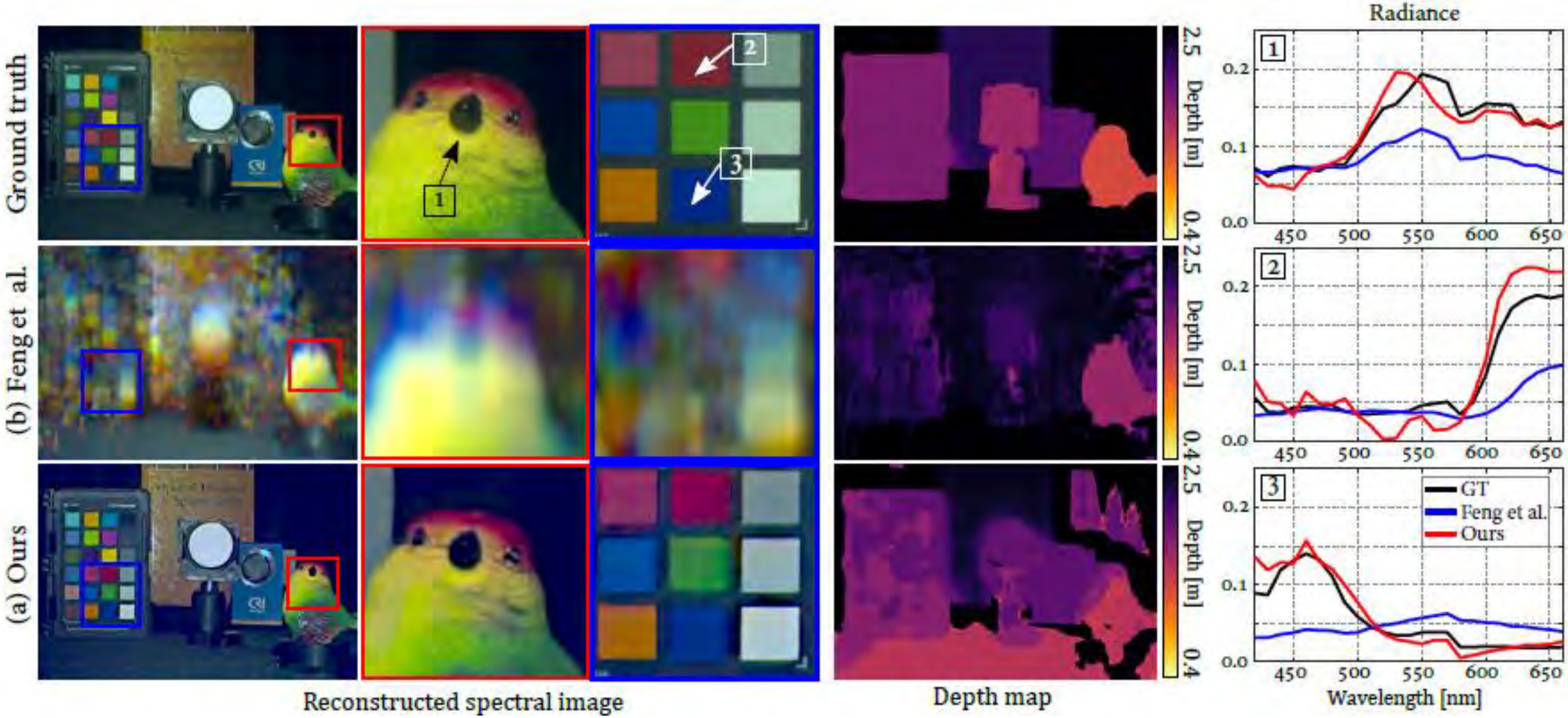
Encoding more than a single physical quantity: Wavelength + Depth.



Baek et al., SIGGRAPH (2020)

Hyper-spectral and depth imaging

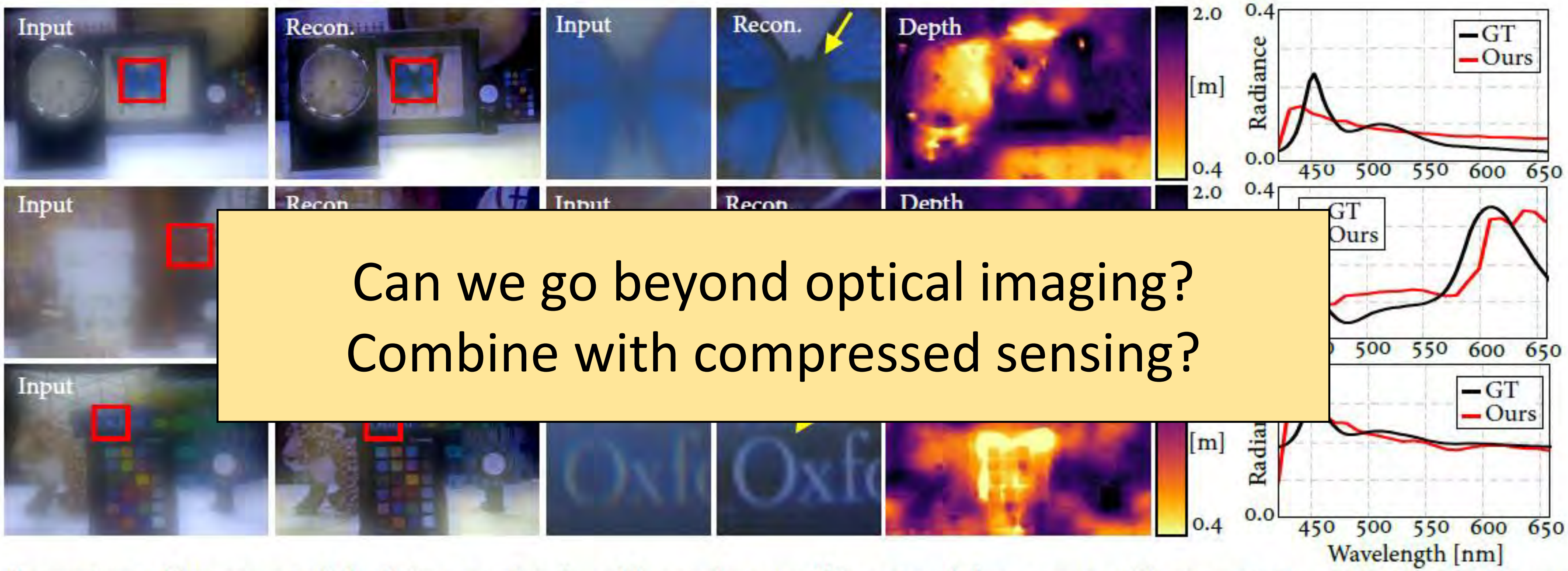
GT from simulation..



Baek et al., SIGGRAPH (2020)

Hyper-spectral and depth imaging

Real captures didn't work out as great..

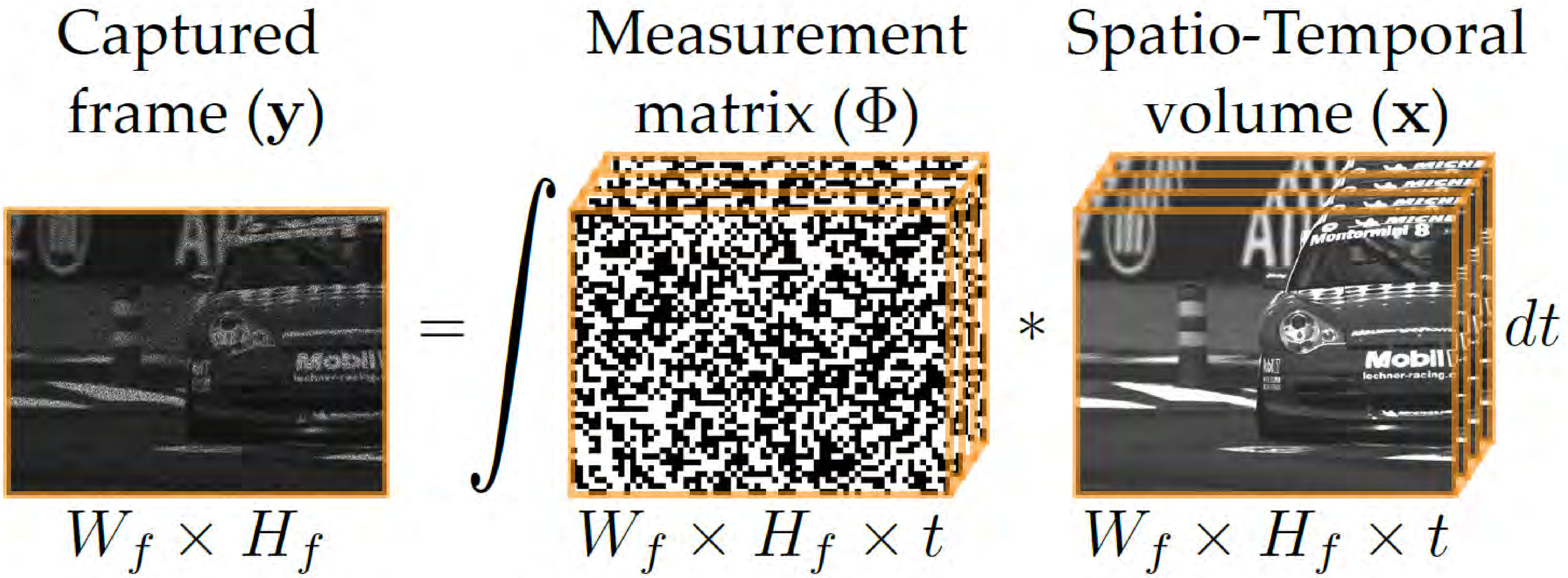


Can we go beyond optical imaging?
Combine with compressed sensing?

Fig. 15. Reconstructed hyperspectral-depth images of real-world, casual scenes. We captured these scenes with our prototype and compare the normalized radiance of resulting HS-D data with the ground truth measured by a spectroradiometer at points indicated by yellow arrows.

Baek et al., SIGGRAPH (2020)

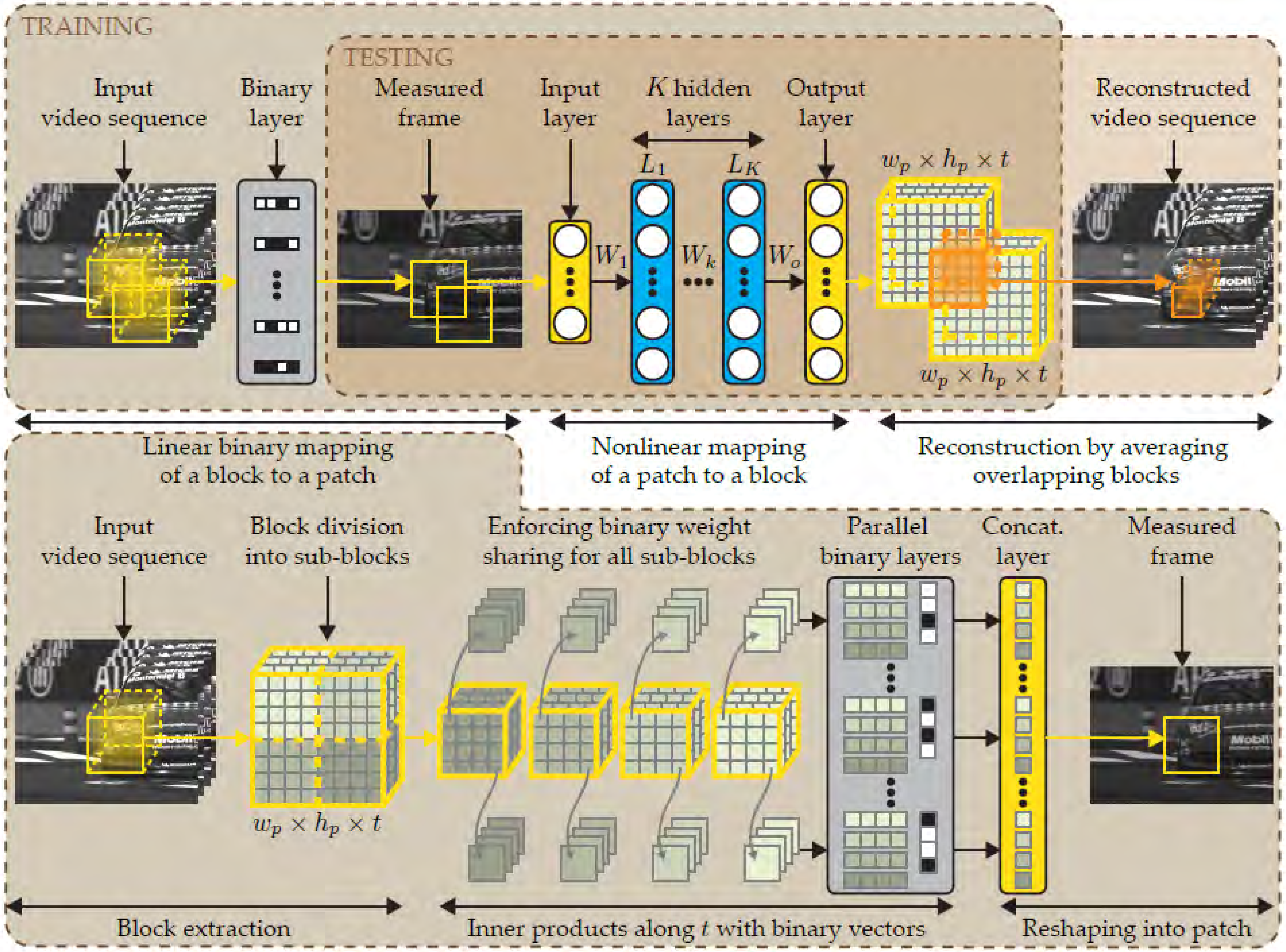
Learning video compressive sensing



Optimized Parameters!

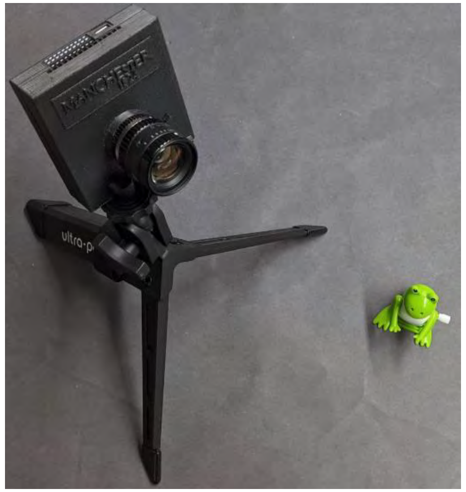
Iliadis et al., *Digital Signal Processing* (2020)

Learning video compressive sensing

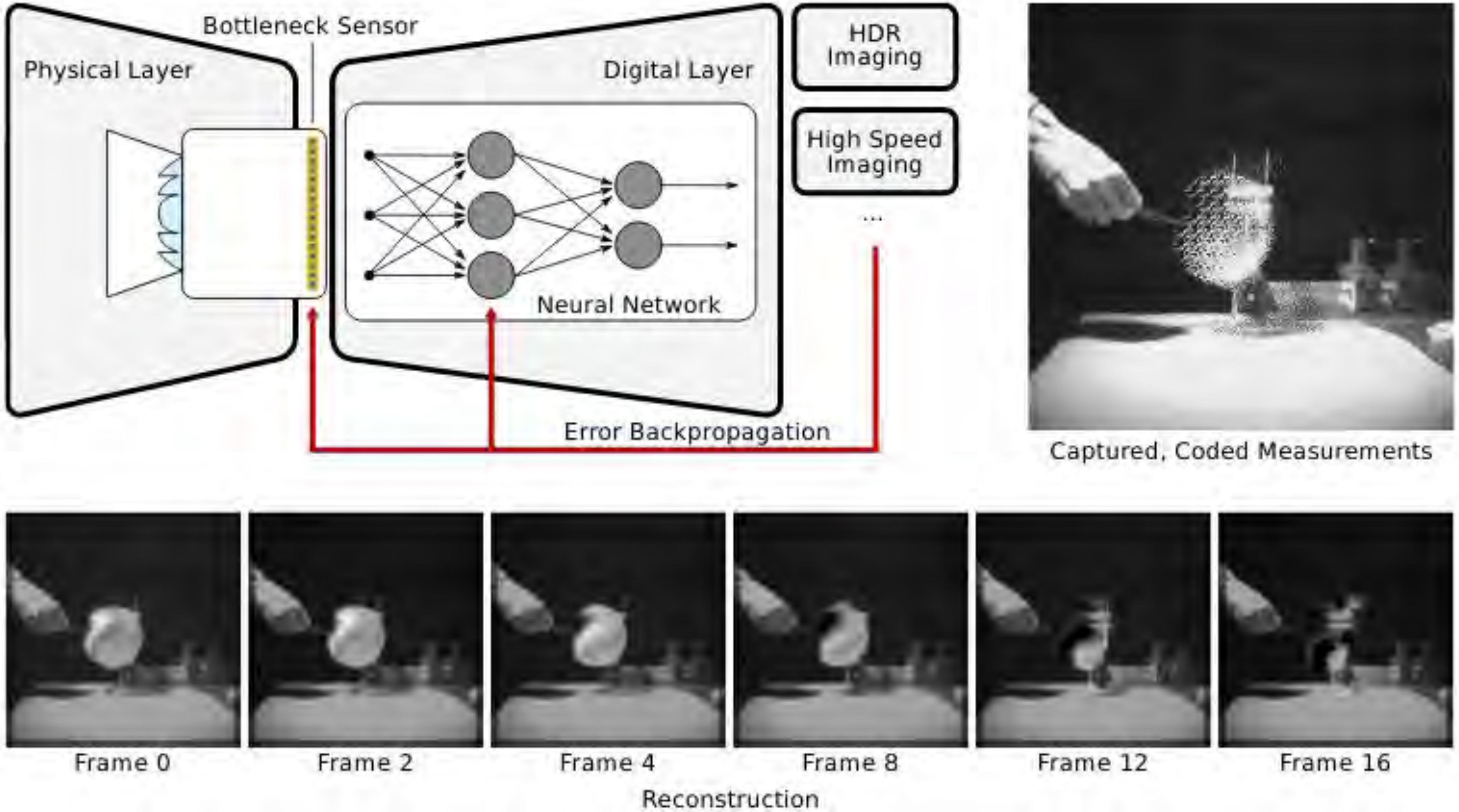


Iliadis et al., *Digital Signal Processing* (2020)

Learning video compressive sensing



Similar idea only not binary and combines dynamic range considerations



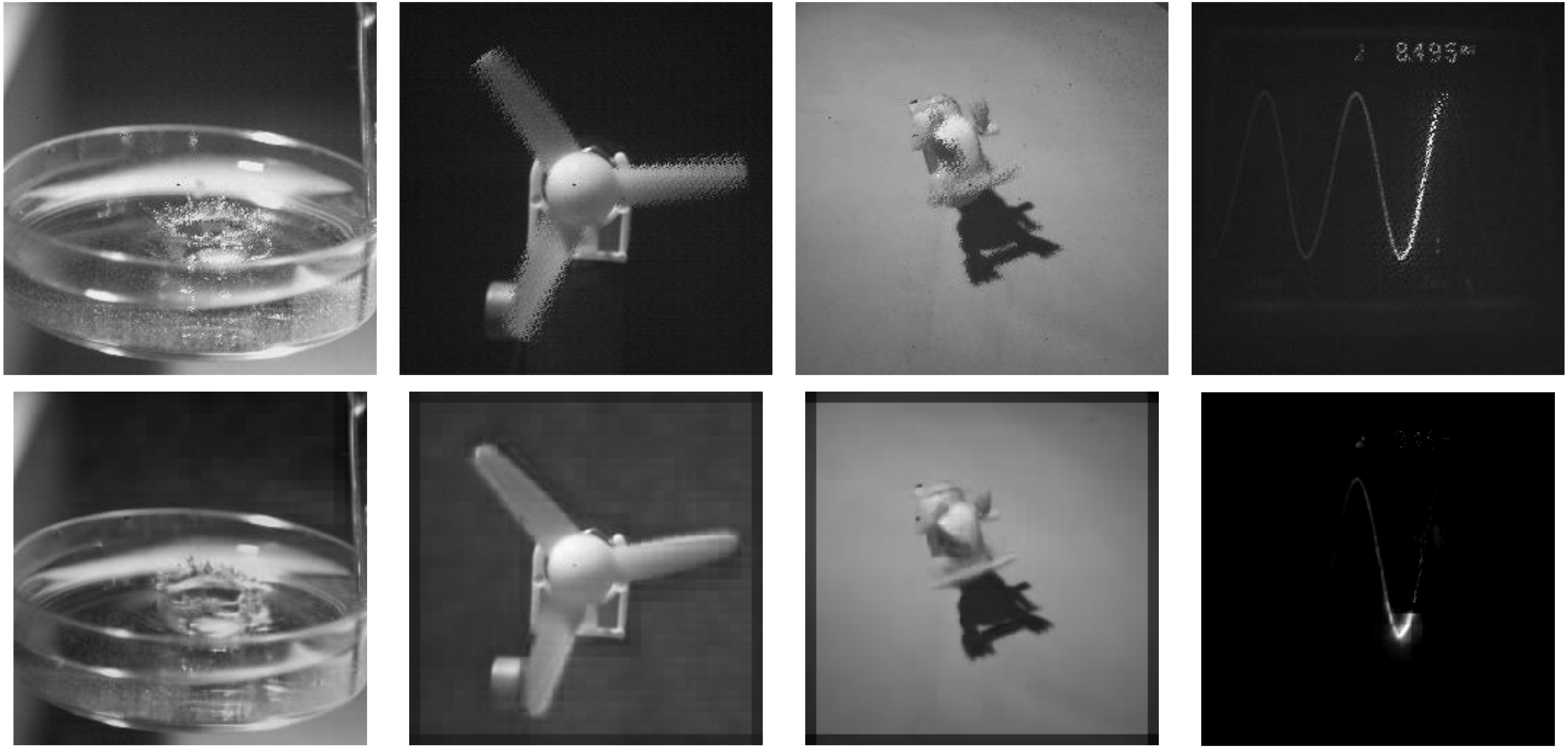
Martel et al., ICCP (2020)

Learning video compressive sensing

4 measure. | 4 measure. | 4 measure. | 1 measure.

Coded Measurements

Output video



64 frames | 64 frames | 64 frames | 16 frames

Martel et al., ICCP (2020)

If you are interested in such techniques

ICCP 2021 is taking place at the Technion! (Assuming COVID allows it…Otherwise Zoom)

References

Horstmeyer, Roarke, Richard Y. Chen, Barbara Kappes, and Benjamin Judkewitz. "Convolutional neural networks that teach microscopes how to image." *arXiv preprint arXiv:1709.07223* (2017).

Muthumbi, Alex, Amey Chaware, Kanghyun Kim, Kevin C. Zhou, Pavan Chandra Konda, Richard Chen, Benjamin Judkewitz, Andreas Erdmann, Barbara Kappes, and Roarke Horstmeyer. "Learned sensing: jointly optimized microscope hardware for accurate image classification." *Biomedical Optics Express* 10, no. 12 (2019): 6351-6369.

Hershko, Eran, Lucien E. Weiss, Tomer Michaeli, and Yoav Shechtman. "Multicolor localization microscopy and point-spread-function engineering by deep learning." *Optics express* 27, no. 5 (2019): 6158-6183.

Kellman, Michael R., Emrah Bostan, Nicole A. Repina, and Laura Waller. "Physics-based learned design: Optimized coded-illumination for quantitative phase imaging." *IEEE Transactions on Computational Imaging* 5, no. 3 (2019): 344-353.

Nehme, Elias, Daniel Freedman, Racheli Gordon, Boris Ferdman, Lucien E. Weiss, Onit Alalouf, Tal Naor, Reut Orange, Tomer Michaeli, and Yoav Shechtman. "DeepSTORM3D: dense 3D localization microscopy and PSF design by deep learning." *Nature Methods* 17, no. 7 (2020): 734-740.

Nehme, Elias, Boris Ferdman, Lucien E. Weiss, Tal Naor, Daniel Freedman, Tomer Michaeli, and Yoav Shechtman. "Learning an optimal PSF-pair for ultra-dense 3D localization microscopy." *arXiv preprint arXiv:2009.14303* (2020).

Peng, Yifan, Ashok Veeraraghavan, Wolfgang Heidrich, and Gordon Wetzstein. "Deep optics: joint design of optics and image recovery algorithms for domain specific cameras." In *ACM SIGGRAPH 2020 Courses*, pp. 1-133. 2020.

Wetzstein, Gordon, Aydogan Ozcan, Sylvain Gigan, Shanhui Fan, Dirk Englund, Marin Soljačić, Cornelia Denz, David AB Miller, and Demetri Psaltis. "Inference in artificial intelligence with deep optics and photonics." *Nature* 588, no. 7836 (2020): 39-47.

And much more….