Washington University in St. Louis

Plug-In SGD: Image Reconstruction in the Age of Machine Learning

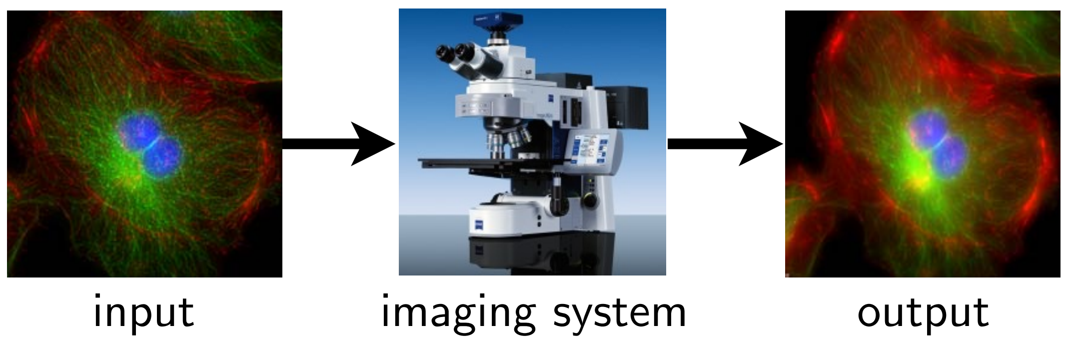
Ulugbek S. Kamilov
Computational Imaging Group (CIG)
Washington University, St. Louis, USA
cigroup.wustl.edu • @wustlcig • kamilov@wustl.edu

Imaging technology is going through a paradigm shift with computation at its core

Washington University in St. Louis

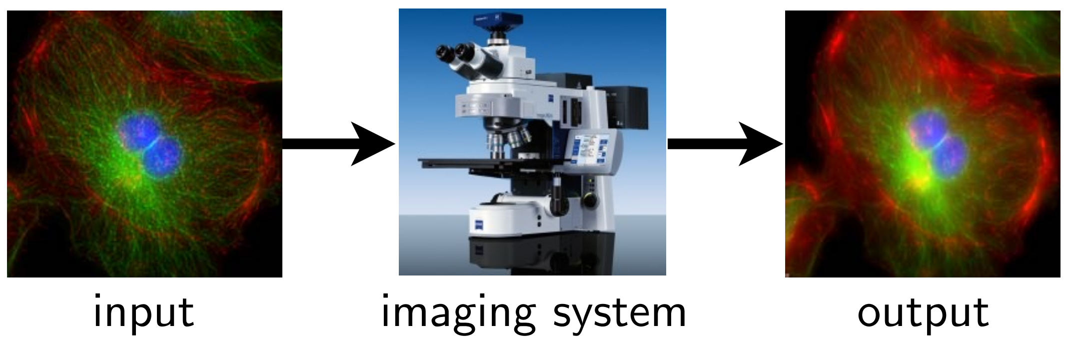
Imaging technology is going through a paradigm shift with computation at its core

Past: Can I see?

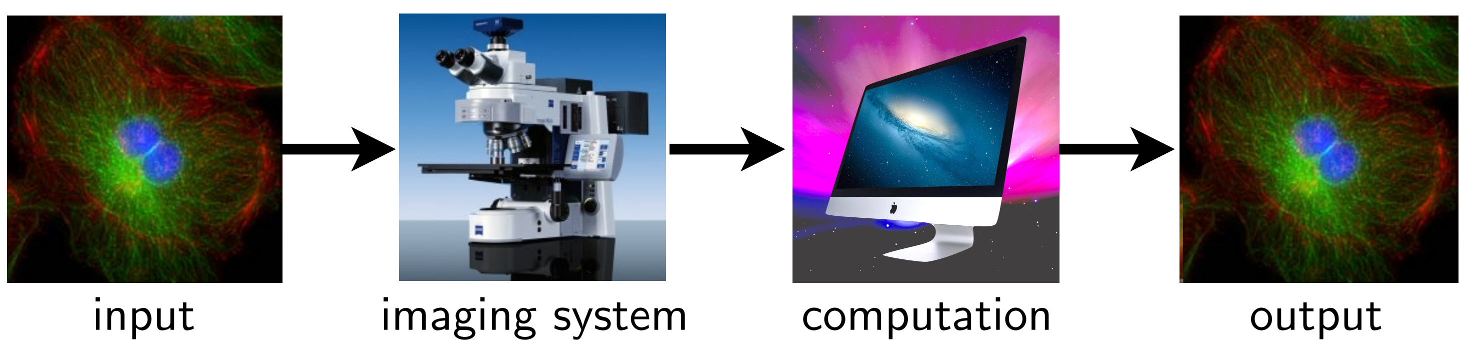


Imaging technology is going through a paradigm shift with computation at its core

Past: Can I see?

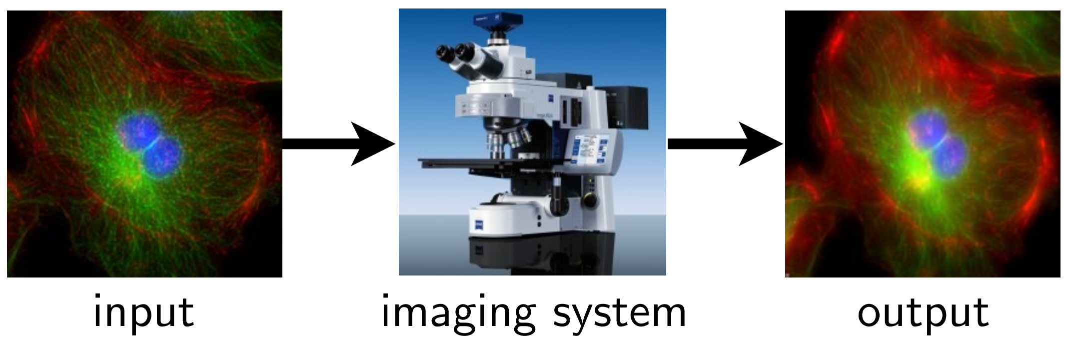


Present: Can I see better?

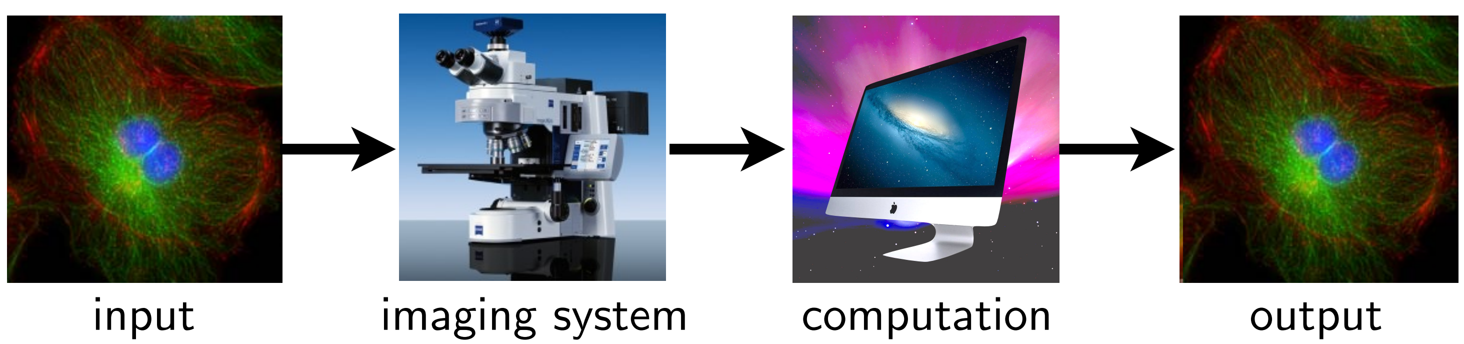


Imaging technology is going through a paradigm shift with computation at its core

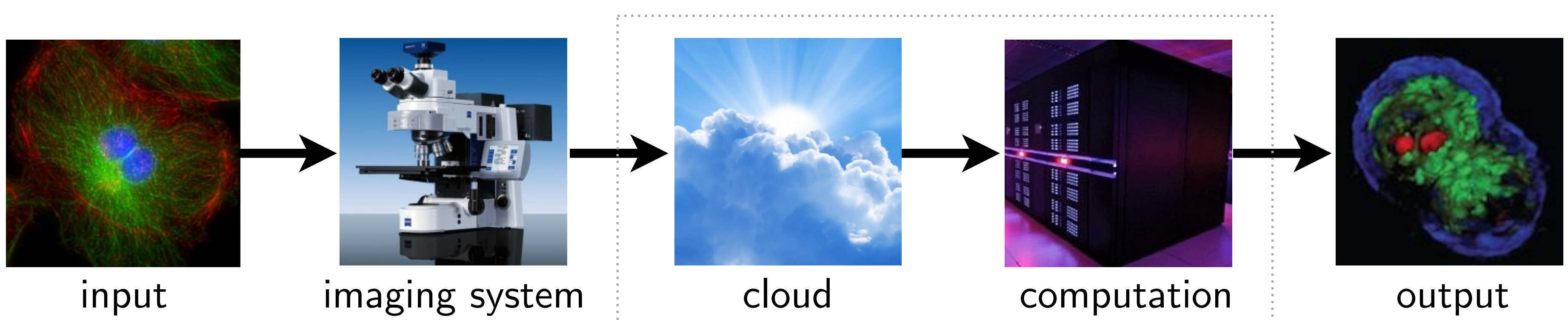
Past: Can I see?



Present: Can I see better?



Future: Can I see more?



Today we will talk about

- **Forward models in imaging**
  Relating the unknowns to the measured data
- **Notions of ill-posedness and regularization**
  When measurements are not enough
- **Optimization at large scales**
  When analytical solutions are not enough
- **Plug-and-Play Priors (PnP) at large scales**
  When traditional optimization is not enough

Today we will talk about

- **Forward models in imaging**
  Relating the unknowns to the measured data

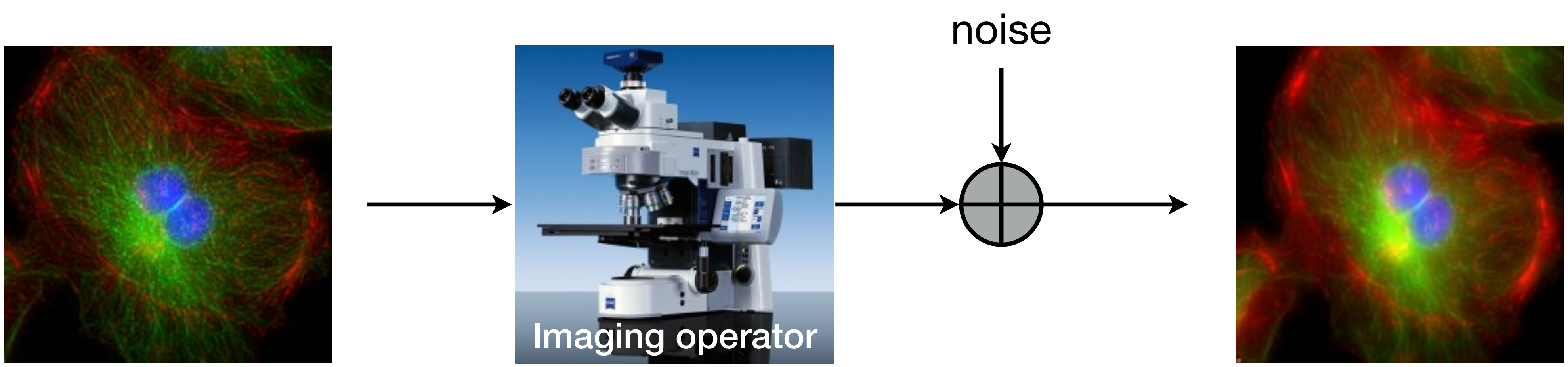
- Notions of ill-posedness and regularization
  When measurements are not enough

- Optimization at large scales
  When analytical solutions are not enough

- Plug-and-Play Priors (PnP) at large scales
  When traditional optimization is not enough

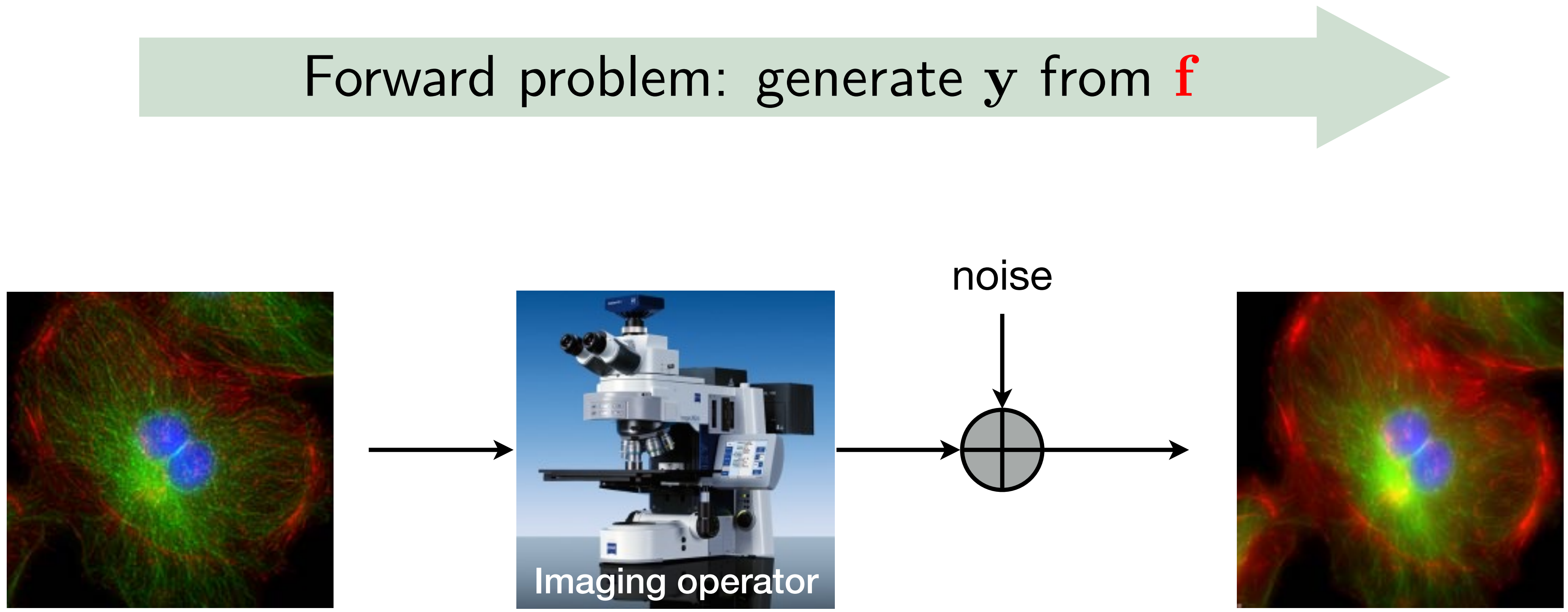
Forward model relates
the unknown object to the observed data

Forward model relates the unknown object to the observed data



$\mathbf{f}$ $\quad$ $\mathbf{H}$ $\quad$ $\mathbf{y} = \mathbf{H}\mathbf{f} + \mathbf{e}$

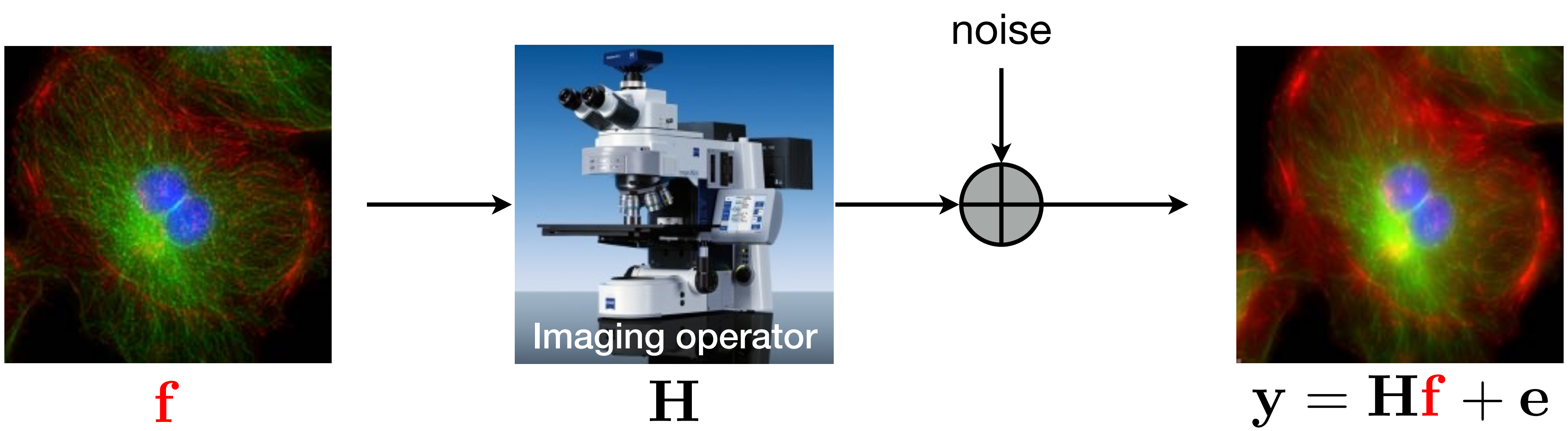
Forward model relates the unknown object to the observed data



$\mathbf{f}$ $\mathbf{H}$ $\mathbf{y} = \mathbf{H}\mathbf{f} + \mathbf{e}$

Forward model relates the unknown object to the observed data

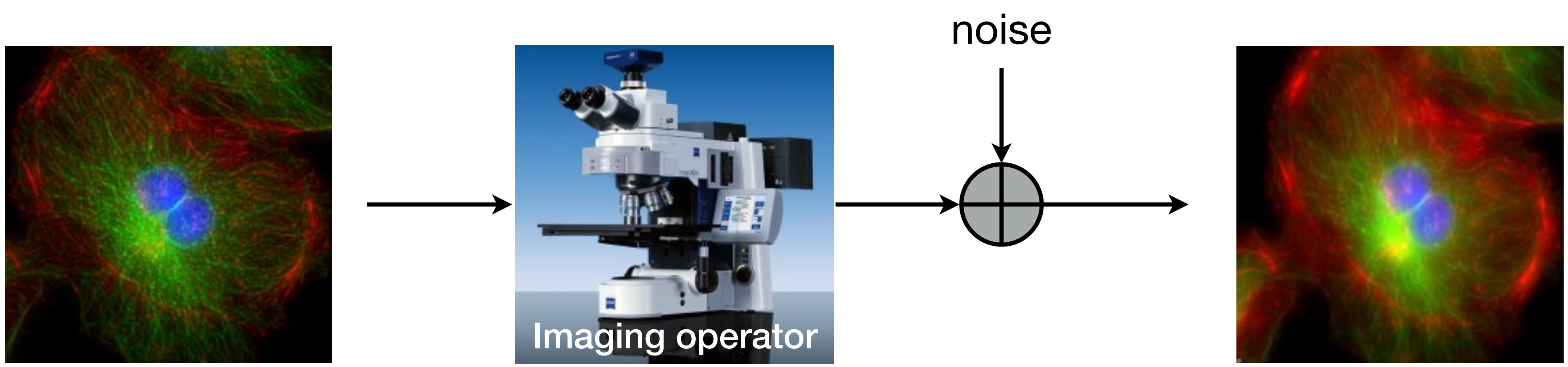
Forward problem: generate $\mathbf{y}$ from $\mathbf{f}$



Inverse problem: recover $\mathbf{f}$ from $\mathbf{y}$

Forward model relates the unknown object to the observed data

Forward problem: generate $\mathbf{y}$ from $\mathbf{f}$



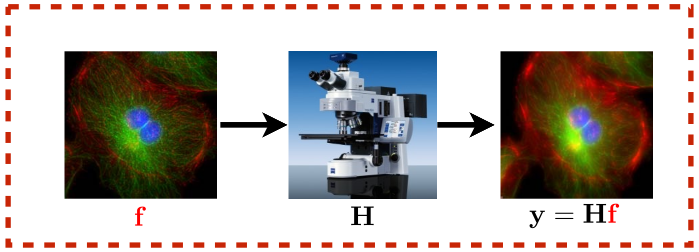
$\mathbf{f}$ $\quad$ $\mathbf{H}$ $\quad$ $\mathbf{y} = \mathbf{H}\mathbf{f} + \mathbf{e}$

Inverse problem: recover $\mathbf{f}$ from $\mathbf{y}$

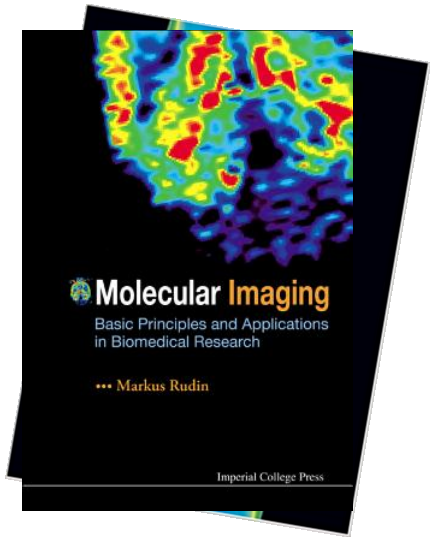
Question: Which problem is harder to solve?

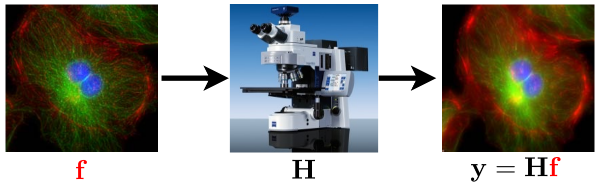
Forward models can be represented as integrals

Forward models can be represented as integrals



Forward models can be represented as integrals



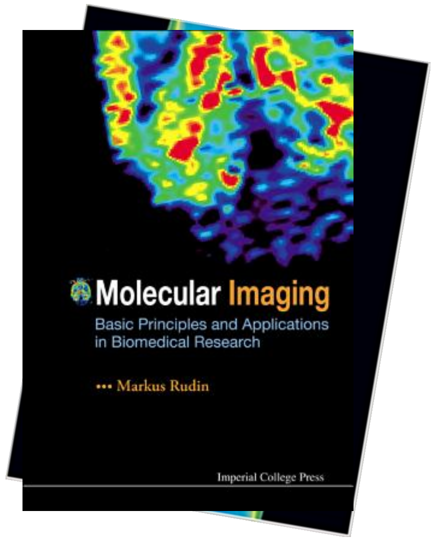


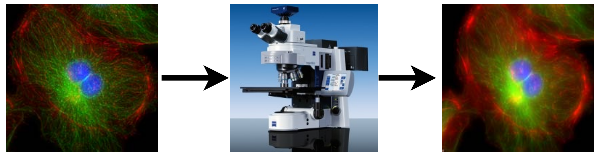
Unser and Tafti, "An Introduction to Sparse Stochastic Processes," 2014

Forward models can be represented as integrals

Unknown molecular/anatomical map: $f(\boldsymbol{r}), \quad \boldsymbol{r} = (x, y, z, t) \in \mathbb{R}^d$

defined over a continuum in space-time



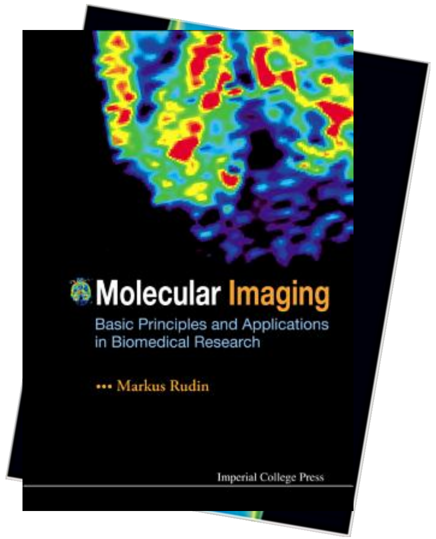


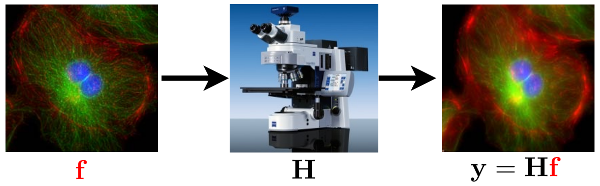
Unser and Tafti, “An Introduction to Sparse Stochastic Processes,” 2014

Forward models can be represented as integrals

Unknown molecular/anatomical map: $f(\boldsymbol{r}), \quad \boldsymbol{r} = (x, y, z, t) \in \mathbb{R}^d$

Space of finite-energy functions: $f \in L_2(\mathbb{R}^d)$





Unser and Tafti, "An Introduction to Sparse Stochastic Processes," 2014

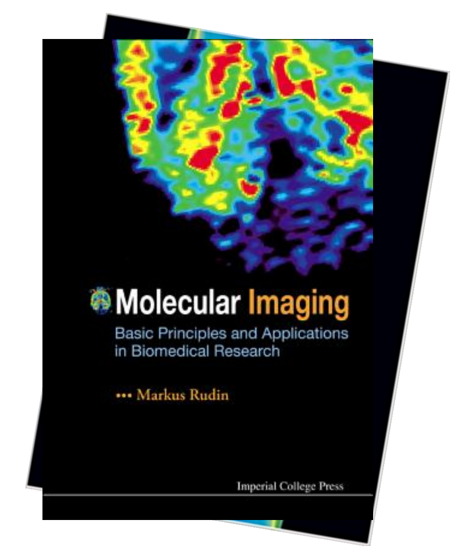
Forward models can be represented as integrals

Unknown molecular/anatomical map: $f(\boldsymbol{r}), \quad \boldsymbol{r} = (x, y, z, t) \in \mathbb{R}^d$

Space of finite-energy functions: $f \in L_2(\mathbb{R}^d)$

Imaging operator: $\mathrm{H} : s \mapsto \mathbf{y} = (y_1, \ldots, y_m) = \mathrm{H}\{f\}$

from continuum to finite dimensional: $\mathrm{H} : L_2(\mathbb{R}^d) \to \mathbb{R}^m$



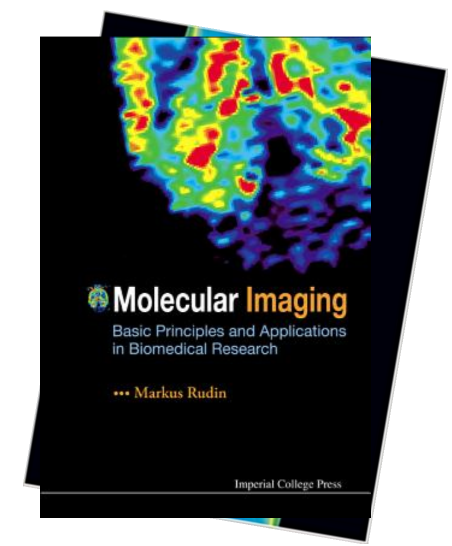
Unser and Tafti, "An Introduction to Sparse Stochastic Processes," 2014

Forward models can be represented as integrals

Unknown molecular/anatomical map: $f(\boldsymbol{r}), \quad \boldsymbol{r} = (x, y, z, t) \in \mathbb{R}^d$

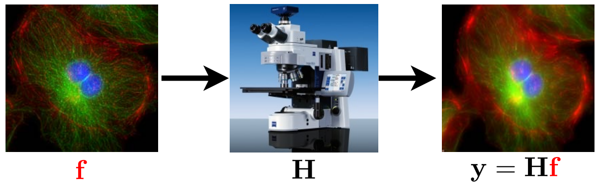
Space of finite-energy functions: $f \in L_2(\mathbb{R}^d)$

Imaging operator: $\mathrm{H} : s \mapsto \mathbf{y} = (y_1, \dots, y_m) = \mathrm{H}\{f\}$



Linearity assumption: $\forall \alpha_1, \alpha_2 \in \mathbb{R}, \quad \forall f_1, f_2 \in L_2(\mathbb{R}^d)$

$$\mathrm{H}\{\alpha_1 f_1 + \alpha_2 f_2\} = \alpha_1 \mathrm{H}\{f_1\} + \alpha_2 \mathrm{H}\{f_2\}$$



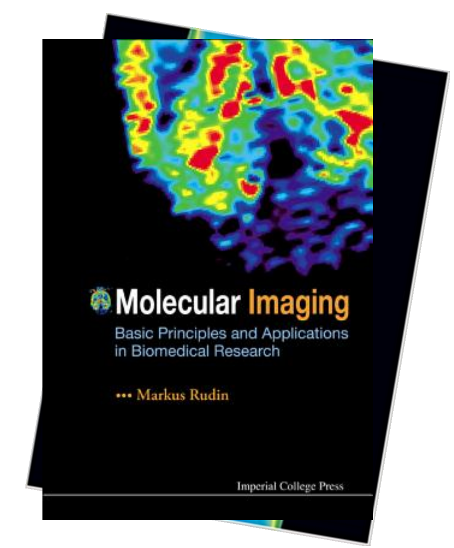
Unser and Tafti, "An Introduction to Sparse Stochastic Processes," 2014

Forward models can be represented as integrals

Unknown molecular/anatomical map: $f(\boldsymbol{r}), \quad \boldsymbol{r} = (x, y, z, t) \in \mathbb{R}^d$

Space of finite-energy functions: $f \in L_2(\mathbb{R}^d)$

Imaging operator: $\mathrm{H} : s \mapsto \mathbf{y} = (y_1, \ldots, y_m) = \mathrm{H}\{f\}$

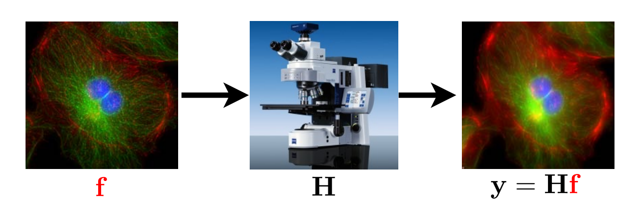


Linearity assumption: $\forall \alpha_1, \alpha_2 \in \mathbb{R}, \quad \forall f_1, f_2 \in L_2(\mathbb{R}^d)$

$$\mathrm{H}\{\alpha_1 f_1 + \alpha_2 f_2\} = \alpha_1 \mathrm{H}\{f_1\} + \alpha_2 \mathrm{H}\{f_2\}$$

$$\Rightarrow \quad [\mathbf{y}]_m = y_m = \langle h_m, f \rangle = \int_{\mathbb{R}^d} h_m(\boldsymbol{r}) f(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}$$

by the Riesz representation theorem



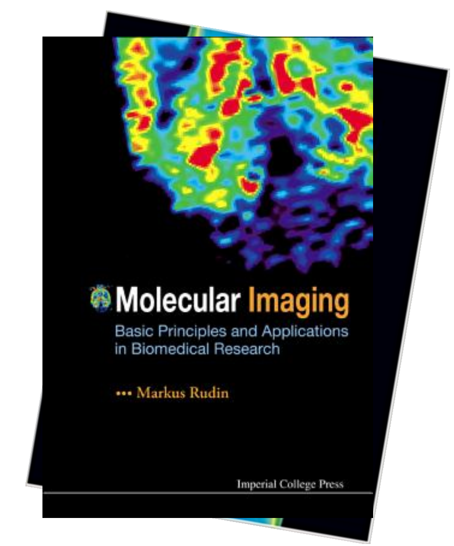
Unser and Tafti, "An Introduction to Sparse Stochastic Processes," 2014

Forward models can be represented as integrals

Unknown molecular/anatomical map: $f(\boldsymbol{r}), \quad \boldsymbol{r} = (x, y, z, t) \in \mathbb{R}^d$

Space of finite-energy functions: $f \in L_2(\mathbb{R}^d)$

Imaging operator: $\mathrm{H} : s \mapsto \mathbf{y} = (y_1, \ldots, y_m) = \mathrm{H}\{f\}$

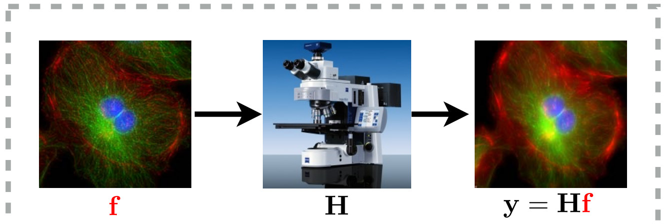


Linearity assumption: $\forall \alpha_1, \alpha_2 \in \mathbb{R}, \quad \forall f_1, f_2 \in L_2(\mathbb{R}^d)$

$$\mathrm{H}\{\alpha_1 f_1 + \alpha_2 f_2\} = \alpha_1 \mathrm{H}\{f_1\} + \alpha_2 \mathrm{H}\{f_2\}$$

impulse response of *m*th detector

$$\Rightarrow \quad [\mathbf{y}]_m = y_m = \langle h_m, f \rangle = \int_{\mathbb{R}^d} h_m(\boldsymbol{r}) f(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}$$



Unser and Tafti, "An Introduction to Sparse Stochastic Processes," 2014

Example imaging operator:
Fourier transform is extensively used in MRI

Example imaging operator:
Fourier transform is extensively used in MRI

"Images are obviously made of sine waves..."

Example imaging operator: Fourier transform is extensively used in MRI

Fourier transform: $\mathcal{F} : L_2(\mathbb{R}^d) \to L_2(\mathbb{R}^d)$

$$\hat{f}(\boldsymbol{\omega}) = \mathcal{F}\{f\} = \int_{\mathbb{R}^d} f(\boldsymbol{r})\, \mathrm{e}^{-\mathrm{j}\langle \boldsymbol{\omega}, \boldsymbol{r} \rangle}\, \mathrm{d}\boldsymbol{r}$$

Example imaging operator:
Fourier transform is extensively used in MRI

Fourier transform: $\mathcal{F} : L_2(\mathbb{R}^d) \to L_2(\mathbb{R}^d)$

$$\hat{f}(\boldsymbol{\omega}) = \mathcal{F}\{f\} = \int_{\mathbb{R}^d} f(\boldsymbol{r})\, \mathrm{e}^{-\mathrm{j}\langle \boldsymbol{\omega}, \boldsymbol{r} \rangle}\, \mathrm{d}\boldsymbol{r}$$

Inverse Fourier transform (reconstruction formula)

$$f(\boldsymbol{r}) = \mathcal{F}^{-1}\{f\} = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{f}(\boldsymbol{\omega})\, \mathrm{e}^{\mathrm{j}\langle \boldsymbol{\omega}, \boldsymbol{r} \rangle}\, \mathrm{d}\boldsymbol{\omega}$$ (a.e.)

Example imaging operator:
Fourier transform is extensively used in MRI

Fourier transform: $\mathcal{F} : L_2(\mathbb{R}^d) \to L_2(\mathbb{R}^d)$

$$\hat{f}(\boldsymbol{\omega}) = \mathcal{F}\{f\} = \int_{\mathbb{R}^d} f(\boldsymbol{r})\, \mathrm{e}^{-\mathrm{j}\langle \boldsymbol{\omega}, \boldsymbol{r} \rangle}\, \mathrm{d}\boldsymbol{r}$$

Inverse Fourier transform (reconstruction formula)

$$f(\boldsymbol{r}) = \mathcal{F}^{-1}\{f\} = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{f}(\boldsymbol{\omega})\, \mathrm{e}^{\mathrm{j}\langle \boldsymbol{\omega}, \boldsymbol{r} \rangle}\, \mathrm{d}\boldsymbol{\omega}$$

As a measurement function: $h_m(\boldsymbol{r}) = \mathrm{e}^{-\mathrm{j}\langle \boldsymbol{\omega}_m, \boldsymbol{r} \rangle}$ (*complex sinusoid*)

$$y_m = \langle h_m, f \rangle = \int_{\mathbb{R}^d} h_m(\boldsymbol{r})\, f(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}$$

Example imaging operator:
Fourier transform is extensively used in MRI

Example imaging operator:
Fourier transform is extensively used in MRI

Example imaging operator:
Fourier transform is extensively used in MRI

Linear forward model for MRI

$$\hat{s}(\boldsymbol{\omega}_m) = \int_{\mathbb{R}^3} s(\boldsymbol{r}) \mathrm{e}^{-\mathrm{j}\langle \boldsymbol{\omega}_m, \boldsymbol{r} \rangle} \, \mathrm{d}\boldsymbol{r}$$

sampling of Fourier transform

$\boldsymbol{r} = (x, y, z) \quad \boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$

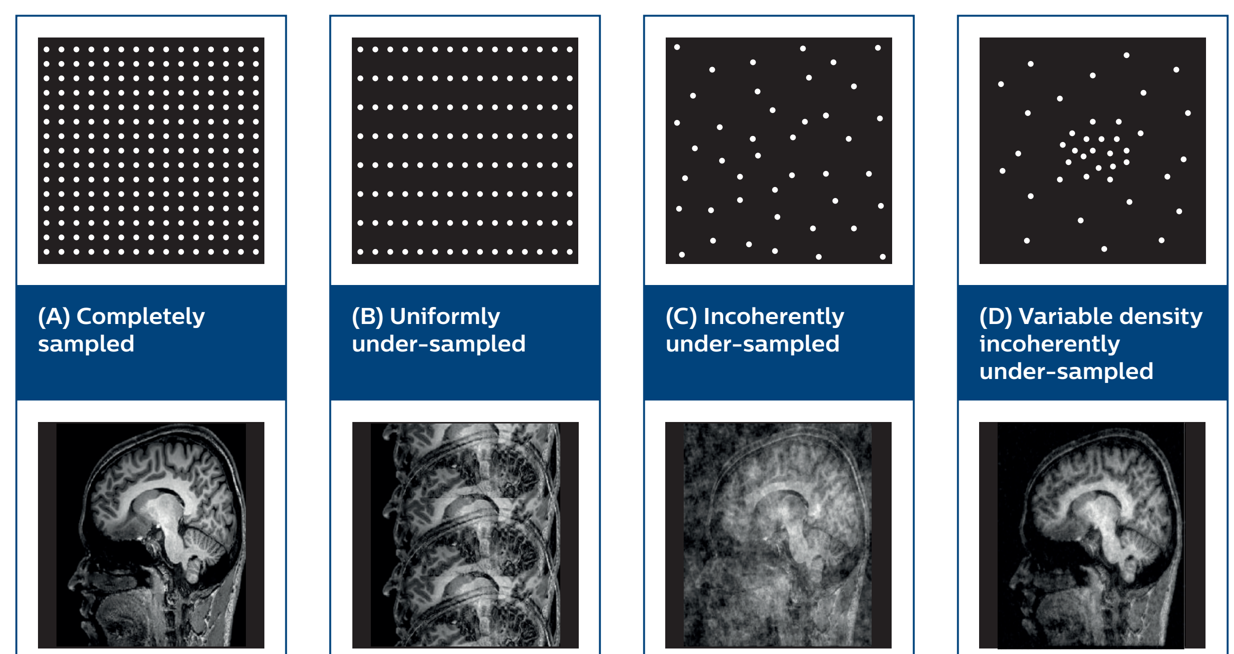
Example imaging operator:
Fourier transform is extensively used in MRI

Linear forward model for MRI

$$\hat{s}(\boldsymbol{\omega}_m) = \int_{\mathbb{R}^3} s(\boldsymbol{r}) \mathrm{e}^{-\mathrm{j}\langle \boldsymbol{\omega}_m, \boldsymbol{r} \rangle} \, \mathrm{d}\boldsymbol{r}$$

sampling of Fourier transform

$$\boldsymbol{r} = (x, y, z) \quad \boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$$



[Source]

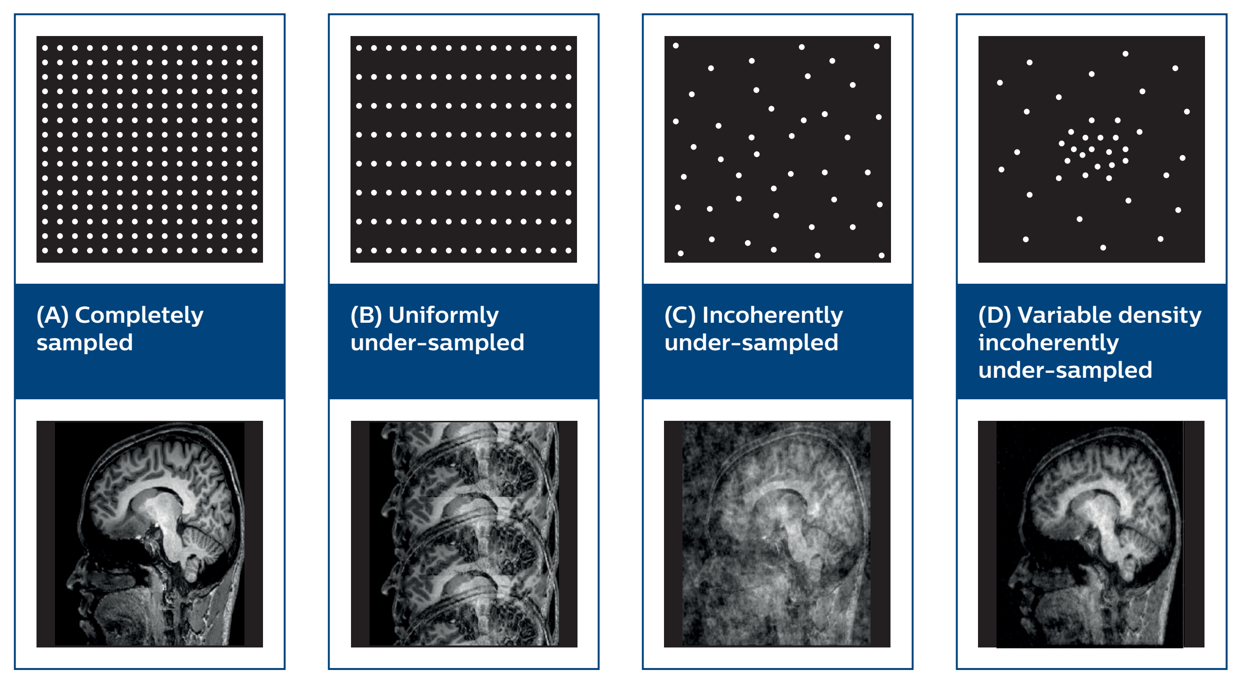
Example imaging operator:
Fourier transform is extensively used in MRI

Linear forward model for MRI

$$\hat{s}(\boldsymbol{\omega}_m) = \int_{\mathbb{R}^3} s(\boldsymbol{r}) \mathrm{e}^{-\mathrm{j}\langle \boldsymbol{\omega}_m, \boldsymbol{r} \rangle} \, \mathrm{d}\boldsymbol{r}$$

Extended forward model with coil sensitivity

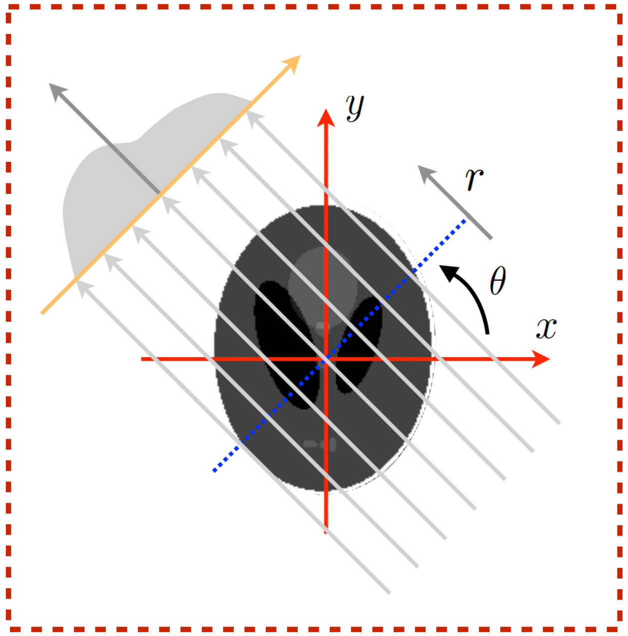
$$\hat{s}_w(\boldsymbol{\omega}_m) = \int_{\mathbb{R}^3} w(\boldsymbol{r}) s(\boldsymbol{r}) \mathrm{e}^{-\mathrm{j}\langle \boldsymbol{\omega}_m, \boldsymbol{r} \rangle} \, \mathrm{d}\boldsymbol{r}$$



[Source]

Example imaging operator:
Radon transform is extensively used in tomography

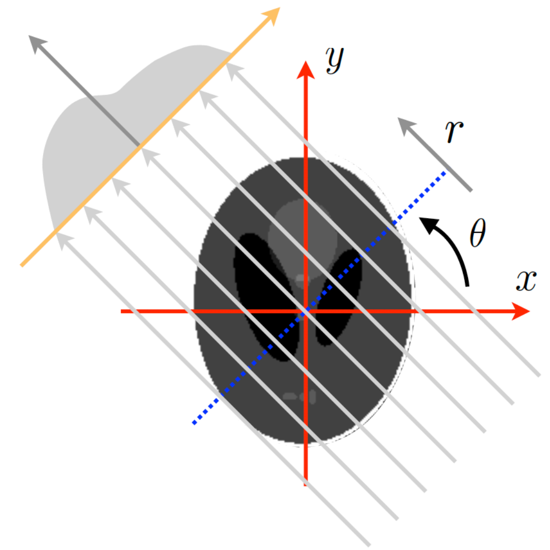
Example imaging operator:
Radon transform is extensively used in tomography



Unser and Tafti, “An Introduction to Sparse Stochastic Processes,” 2014

Example imaging operator: Radon transform is extensively used in tomography

Projection geometry: $\boldsymbol{r} = t\boldsymbol{\theta} + r\boldsymbol{\theta}^{\perp}, \quad \boldsymbol{\theta} = (\cos\theta, \sin\theta)$



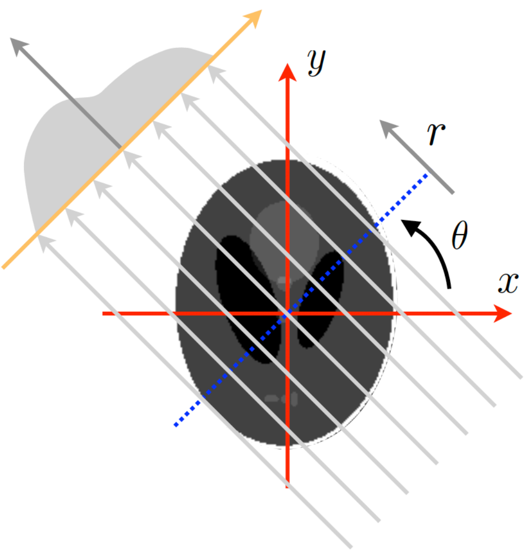
Unser and Tafti, "An Introduction to Sparse Stochastic Processes," 2014

Example imaging operator:
Radon transform is extensively used in tomography

Projection geometry: $\boldsymbol{r} = t\boldsymbol{\theta} + r\boldsymbol{\theta}^{\perp}, \quad \boldsymbol{\theta} = (\cos\theta, \sin\theta)$

Radon transform computes line integrals of the object:

$$\mathrm{R}_{\theta}\{f(\boldsymbol{r})\}(t) = \int_{\mathbb{R}} f(t\boldsymbol{\theta} + r\boldsymbol{\theta}^{\perp})\,\mathrm{d}r = \int_{\mathbb{R}^2} f(\boldsymbol{r})\delta(t - \langle \boldsymbol{r}, \boldsymbol{\theta}\rangle)\,\mathrm{d}\boldsymbol{r}$$



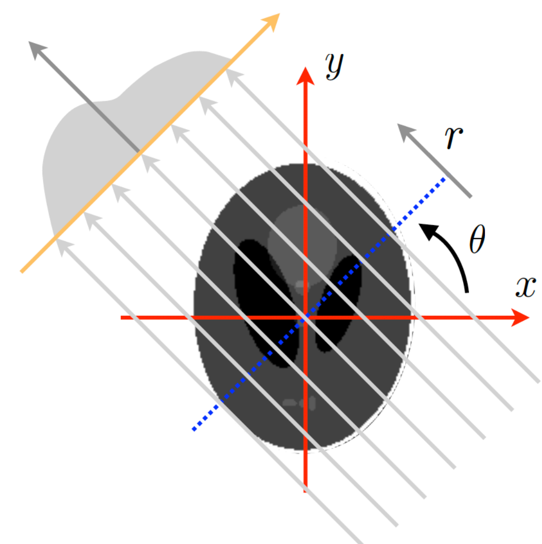
Unser and Tafti, "An Introduction to Sparse Stochastic Processes," 2014

Example imaging operator:
Radon transform is extensively used in tomography

Projection geometry: $\boldsymbol{r} = t\boldsymbol{\theta} + r\boldsymbol{\theta}^{\perp}, \quad \boldsymbol{\theta} = (\cos\theta, \sin\theta)$

Radon transform computes line integrals of the object:

$$\mathrm{R}_{\theta}\{f(\boldsymbol{r})\}(t) = \int_{\mathbb{R}} f(t\boldsymbol{\theta} + r\boldsymbol{\theta}^{\perp})\,\mathrm{d}r$$
$$= \int_{\mathbb{R}^2} f(\boldsymbol{r})\delta(t - \langle \boldsymbol{r}, \boldsymbol{\theta} \rangle)\,\mathrm{d}\boldsymbol{r}$$



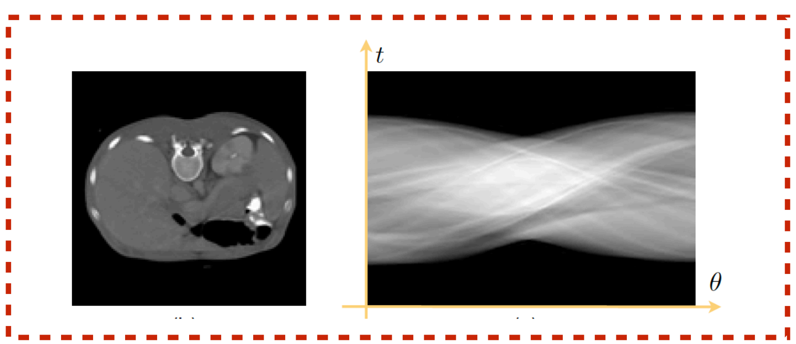


image and its sinogram

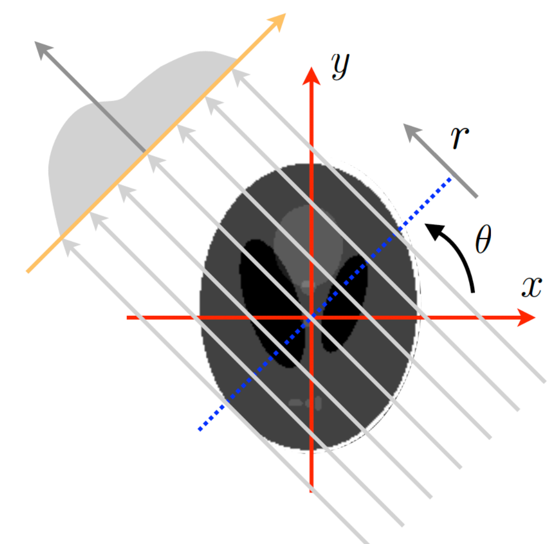
Unser and Tafti, "An Introduction to Sparse Stochastic Processes," 2014

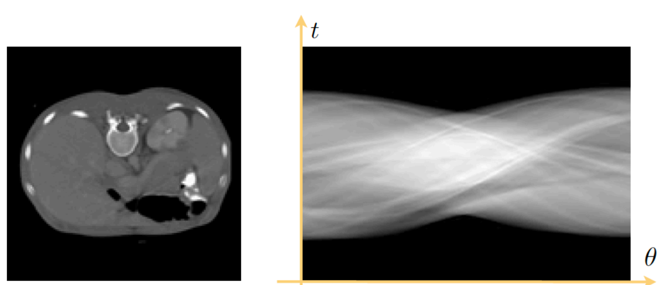
Example imaging operator: Radon transform is extensively used in tomography

Projection geometry: $\boldsymbol{r} = t\boldsymbol{\theta} + r\boldsymbol{\theta}^{\perp}, \quad \boldsymbol{\theta} = (\cos\theta, \sin\theta)$

Radon transform computes line integrals of the object:

$$\mathrm{R}_\theta\{f(\boldsymbol{r})\}(t) = \int_{\mathbb{R}} f(t\boldsymbol{\theta} + r\boldsymbol{\theta}^{\perp})\,\mathrm{d}r$$
$$= \int_{\mathbb{R}^2} f(\boldsymbol{r})\delta(t - \langle \boldsymbol{r}, \boldsymbol{\theta}\rangle)\,\mathrm{d}\boldsymbol{r}$$





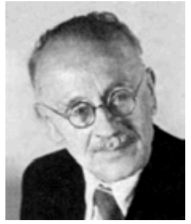
As a measurement function: $h_m(\boldsymbol{r}) = \delta(t_m - \langle \boldsymbol{r}, \theta_m\rangle)$

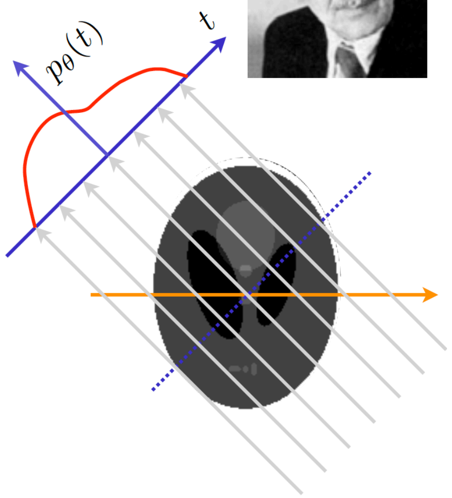
Unser and Tafti, “An Introduction to Sparse Stochastic Processes,” 2014

Central slice theorem relates projections to the Fourier transform of the object

Central slice theorem relates projections to the Fourier transform of the object

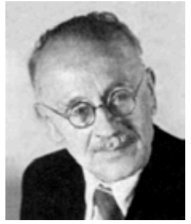
Radon transform: $p_\theta(t) = \mathsf{R}_\theta\{f\}(t,\theta)$

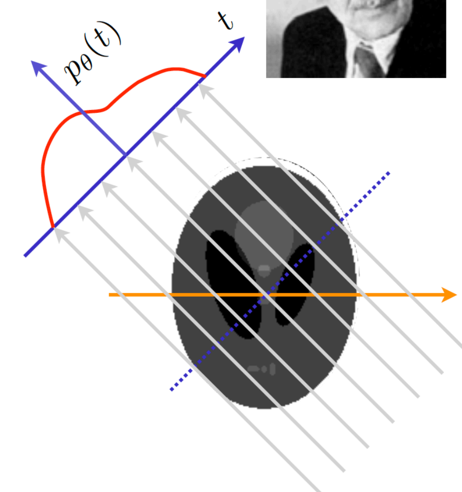




Unser and Tafti, "An Introduction to Sparse Stochastic Processes," 2014

Central slice theorem relates projections to the Fourier transform of the object





Radon transform: $p_\theta(t) = \mathrm{R}_\theta\{f\}(t, \theta)$

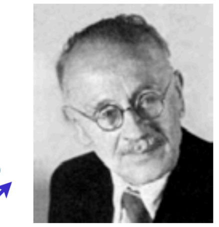
1D and 2D Fourier relationships:

$$\hat{p}_\theta(\omega) = \mathcal{F}_{\text{1D}}\{p_\theta\}(\omega)$$ 1D Fourier of data

$$\hat{f}(\boldsymbol{\omega}) = \mathcal{F}_{\text{2D}}\{f\}(\boldsymbol{\omega}) = \hat{f}_{\text{pol}}(\omega, \theta)$$ 2D Fourier of image

Unser and Tafti, "An Introduction to Sparse Stochastic Processes," 2014

Central slice theorem relates projections to the Fourier transform of the object

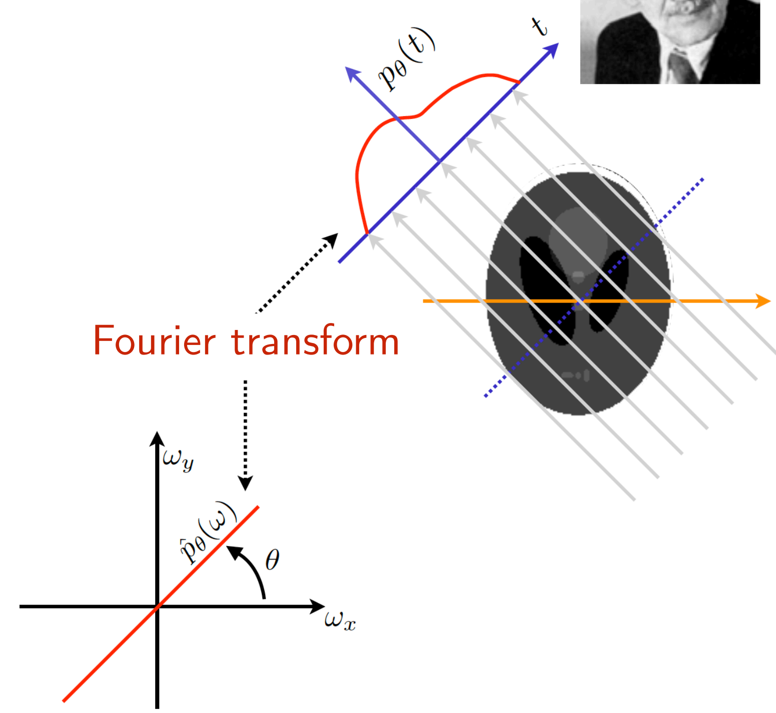


Radon transform: $p_\theta(t) = \mathrm{R}_\theta\{f\}(t, \theta)$

1D and 2D Fourier relationships:

$$\hat{p}_\theta(\omega) = \mathcal{F}_{\text{1D}}\{p_\theta\}(\omega)$$

$$\hat{f}(\boldsymbol{\omega}) = \mathcal{F}_{\text{2D}}\{f\}(\boldsymbol{\omega}) = \hat{f}_{\text{pol}}(\omega, \theta)$$



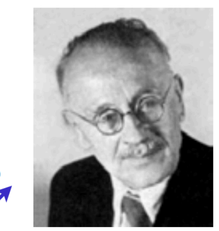
Central-slice theorem relates projections to Fourier sampling:

$$\hat{p}_\theta(\omega) = \hat{f}(\omega \cos\theta, \omega \sin\theta) = \hat{f}_{\text{pol}}(\omega, \theta)$$

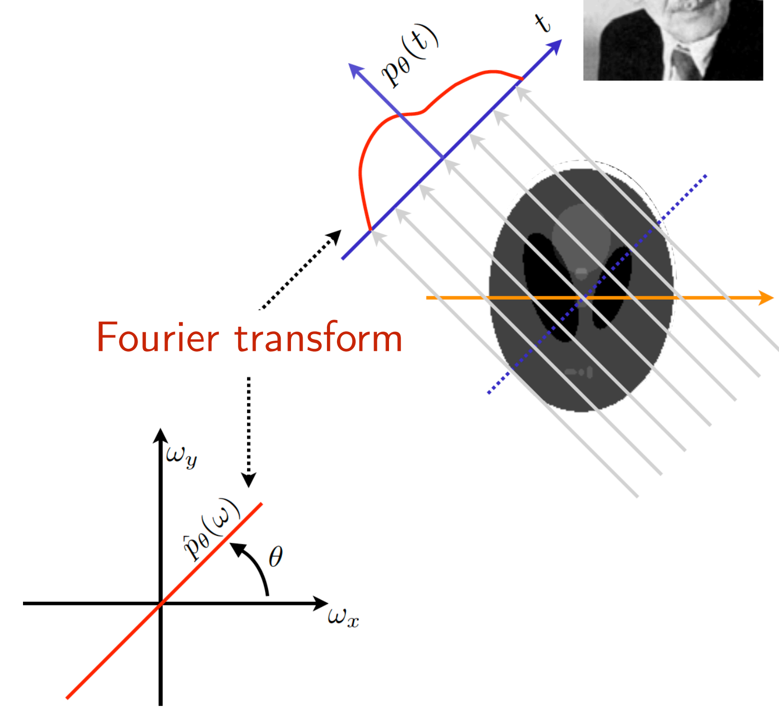
Establishes Fourier relationship between data and image

Unser and Tafti, "An Introduction to Sparse Stochastic Processes," 2014

Central slice theorem relates projections to the Fourier transform of the object



Radon transform: $p_\theta(t) = \mathsf{R}_\theta\{f\}(t,\theta)$



1D and 2D Fourier relationships:

$$\hat{p}_\theta(\omega) = \mathcal{F}_{\text{1D}}\{p_\theta\}(\omega)$$

$$\hat{f}(\boldsymbol{\omega}) = \mathcal{F}_{\text{2D}}\{f\}(\boldsymbol{\omega}) = \hat{f}_{\text{pol}}(\omega,\theta)$$

Central-slice theorem relates projections to Fourier sampling:

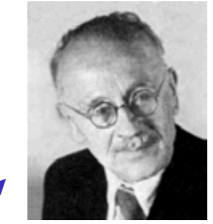
$$\hat{p}_\theta(\omega) = \hat{f}(\omega\cos\theta, \omega\sin\theta) = \hat{f}_{\text{pol}}(\omega,\theta)$$

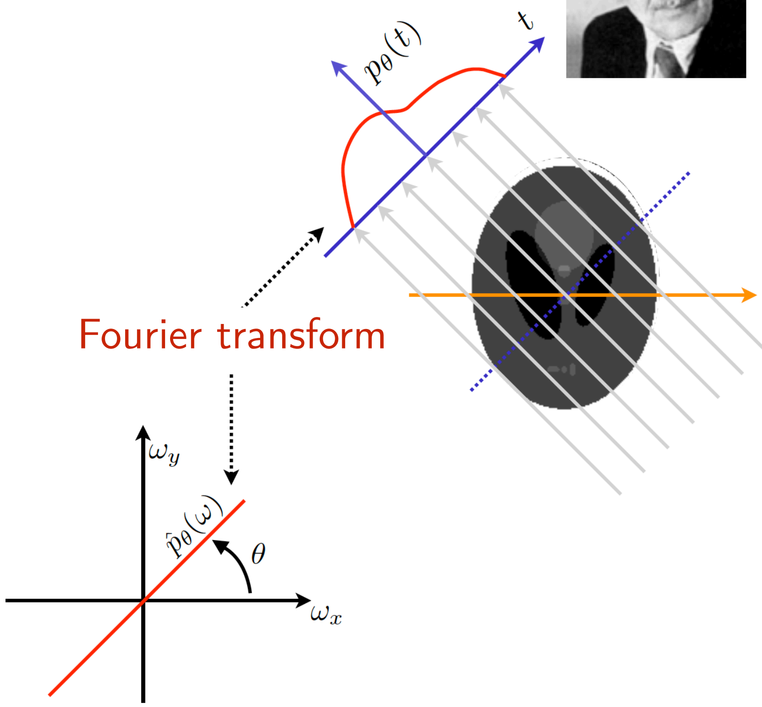
Establishes Fourier relationship between data and image

Proof for angle zero:

$$\hat{f}(\omega,0) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\,\mathrm{e}^{-\mathrm{j}\omega x}\,\mathrm{d}x\mathrm{d}y = \int_{-\infty}^{+\infty} \underbrace{\left(\int_{-\infty}^{+\infty} f(x,y)\,\mathrm{d}y\right)}_{p_0(x)} \mathrm{e}^{-\mathrm{j}\omega x}\,\mathrm{d}x = \hat{p}_0(x)$$

Central slice theorem relates projections to the Fourier transform of the object





Radon transform: $p_\theta(t) = \mathsf{R}_\theta\{f\}(t,\theta)$

1D and 2D Fourier relationships:

$$\hat{p}_\theta(\omega) = \mathcal{F}_{\text{1D}}\{p_\theta\}(\omega)$$

$$\hat{f}(\boldsymbol{\omega}) = \mathcal{F}_{\text{2D}}\{f\}(\boldsymbol{\omega}) = \hat{f}_{\text{pol}}(\omega,\theta)$$

Central-slice theorem relates projections to Fourier sampling:

$$\hat{p}_\theta(\omega) = \hat{f}(\omega\cos\theta, \omega\sin\theta) = \hat{f}_{\text{pol}}(\omega,\theta)$$

Establishes Fourier relationship between data and image

Proof for angle zero:

Question: How to generalize to other angles?

$$\hat{f}(\omega,0) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\,\mathrm{e}^{-\mathrm{j}\omega x}\,\mathrm{d}x\mathrm{d}y = \int_{-\infty}^{+\infty}\underbrace{\left(\int_{-\infty}^{+\infty} f(x,y)\,\mathrm{d}y\right)}_{p_0(x)}\mathrm{e}^{-\mathrm{j}\omega x}\,\mathrm{d}x = \hat{p}_0(x)$$

Most imaging systems can be characterized with a forward model

Most imaging systems can be characterized with a forward model

Modality	Radiation	Forward model	Variations
2D or 3D tomography	coherent x-ray	$y_i = \mathrm{R}_{\boldsymbol{\theta}_i} x$	parallel, cone beam, spiral sampling
3D deconvolution microscopy	fluorescence	$y = \mathrm{H}x$	brightfield, confocal, light sheet
structured illumination microscopy (SIM)	fluorescence	$y_i = \mathrm{H}\mathrm{W}_i x$ H: PSF of microscope W_i: illumination pattern	full 3D reconstruction, non-sinusoidal patterns
Positron Emission Tomography (PET)	gamma rays	$y_i = \mathrm{H}_{\boldsymbol{\theta}_i} x$	list mode with time-of-flight
Magnetic resonance imaging (MRI)	radio frequency	$y = \mathrm{F}x$	uniform or non-uniform sampling in k space
Cardiac MRI (parallel, non-uniform)	radio frequency	$y_{t,i} = \mathrm{F}_t \mathrm{W}_i x$ W_i: coil sensitivity	gated or not, retrospective registration
Optical diffraction tomography	coherent light	$y_i = \mathrm{W}_i \mathrm{F}_i x$	with holography or grating interferometry

Washington University in St. Louis

Most imaging systems can be characterized with a forward model

Modality	Radiation	Forward model	Variations
2D or 3D tomography	coherent x-ray	$y_i = \mathrm{R}_{\boldsymbol{\theta}_i} x$	parallel, cone beam, spiral sampling
3D deconvolution microscopy	fluorescence	$y = \mathrm{H}x$	brightfield, confocal, light sheet
structured illumination microscopy (SIM)	fluorescence	$y_i = \mathrm{H}\mathrm{W}_i x$ H: PSF of microscope W_i: illumination pattern	full 3D reconstruction, non-sinusoidal patterns
Positron Emission Tomography (PET)	gamma rays	$y_i = \mathrm{H}_{\boldsymbol{\theta}_i} x$	list mode with time-of-flight
Magnetic resonance imaging (MRI)	radio frequency	$y = \mathrm{F}x$	uniform or non-uniform sampling in k space
Cardiac MRI (parallel, non-uniform)	radio frequency	$y_{t,i} = \mathrm{F}_t \mathrm{W}_i x$ W_i: coil sensitivity	gated or not, retrospective registration
Optical diffraction tomography	coherent light	$y_i = \mathrm{W}_i \mathrm{F}_i x$	with holography or grating interferometry

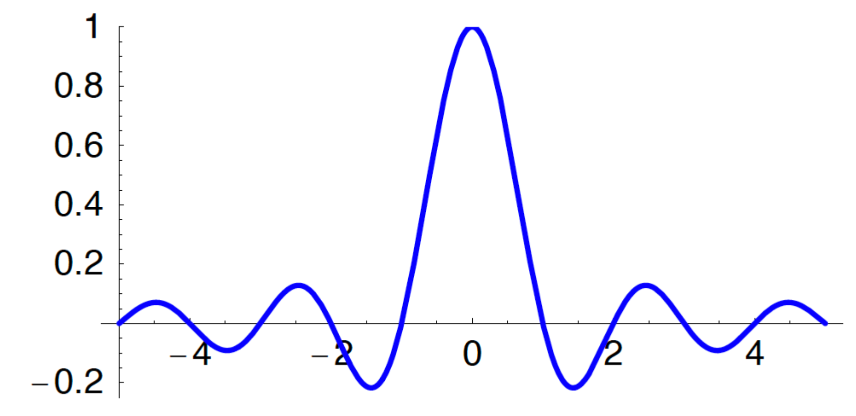
Currently active collaborations at CIG

Discretization: Continuous domain formalism easily reduces to a noisy linear system

Discretization: Continuous domain formalism easily reduces to a noisy linear system

Representation with basis functions:

$$f(\boldsymbol{r}) = \sum_{\boldsymbol{k} \in \Omega} f[\boldsymbol{k}] \beta_{\boldsymbol{k}}(\boldsymbol{r})$$



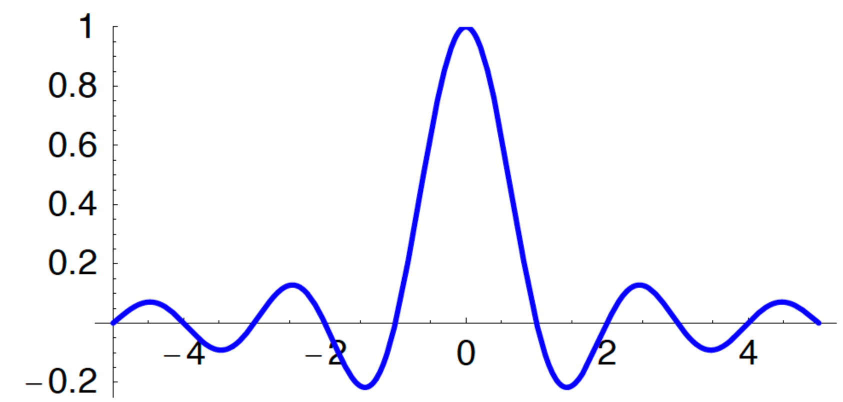
Question: What type of representation is offered by **sinc**?

Unser and Tafti, "An Introduction to Sparse Stochastic Processes," 2014

Discretization: Continuous domain formalism easily reduces to a noisy linear system

Representation with basis functions:

$$f(\boldsymbol{r}) = \sum_{\boldsymbol{k}\in\Omega} f[\boldsymbol{k}]\beta_{\boldsymbol{k}}(\boldsymbol{r})$$



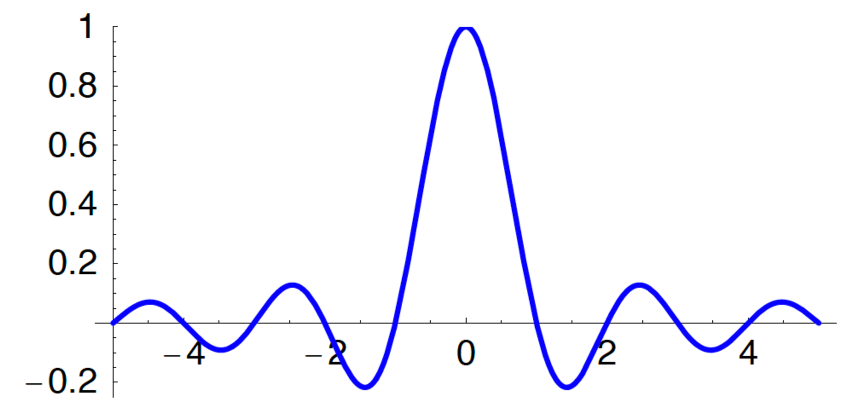
Signal vector: $\mathbf{f} = \{f[\boldsymbol{k}]\}_{\boldsymbol{k}\in\Omega} \in \mathbb{R}^n$

Unser and Tafti, “An Introduction to Sparse Stochastic Processes,” 2014

Discretization: Continuous domain formalism easily reduces to a noisy linear system

Representation with basis functions:

$$f(\boldsymbol{r}) = \sum_{\boldsymbol{k}\in\Omega} f[\boldsymbol{k}]\beta_{\boldsymbol{k}}(\boldsymbol{r})$$



Signal vector: $\mathbf{f} = \{f[\boldsymbol{k}]\}_{\boldsymbol{k}\in\Omega} \in \mathbb{R}^n$

Discretized measurement model:

$$y_i = \int_{\mathbb{R}^d} f(\boldsymbol{r})h_i(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} + e_i = \langle f, h_i\rangle + e_i, \quad (i = 1, \dots, m)$$

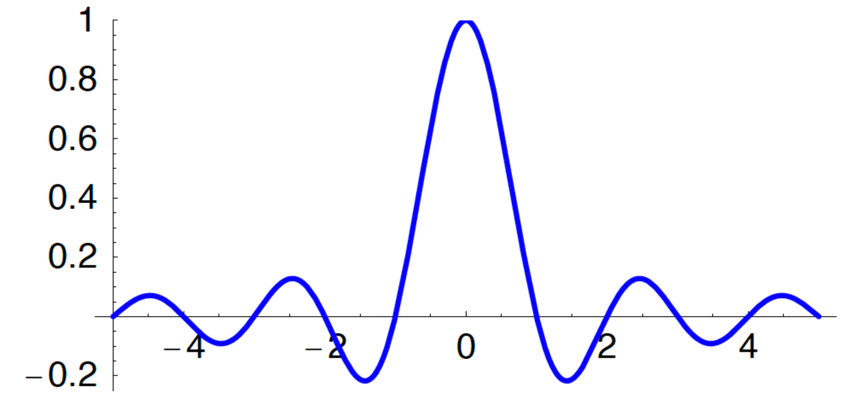
Question: What are the sources of noise?

Unser and Tafti, "An Introduction to Sparse Stochastic Processes," 2014

Discretization: Continuous domain formalism easily reduces to a noisy linear system

Representation with basis functions:

$$f(\boldsymbol{r}) = \sum_{\boldsymbol{k}\in\Omega} f[\boldsymbol{k}]\beta_{\boldsymbol{k}}(\boldsymbol{r})$$



Signal vector: $\mathbf{f} = \{f[\boldsymbol{k}]\}_{\boldsymbol{k}\in\Omega} \in \mathbb{R}^n$

Discretized measurement model:

$$y_i = \int_{\mathbb{R}^d} f(\boldsymbol{r})h_i(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} + e_i = \langle f, h_i\rangle + e_i, \quad (i = 1, \ldots, m)$$

$$\Rightarrow \boxed{\mathbf{y} = \mathbf{H}\mathbf{f} + \mathbf{e}} \qquad [\mathbf{H}]_{i,\boldsymbol{k}} = \langle h_i, \beta_{\boldsymbol{k}}\rangle = \int_{\mathbb{R}^d} h_m(\boldsymbol{r})\beta_{\boldsymbol{k}}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}$$

linear system of equations

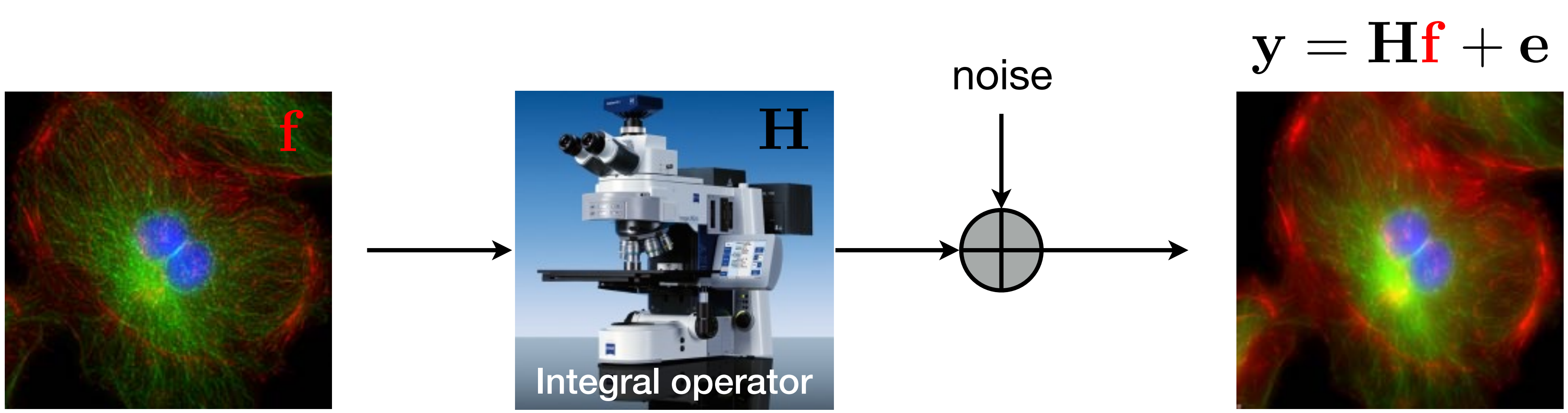
Unser and Tafti, "An Introduction to Sparse Stochastic Processes," 2014

To conclude "forward models"

Many imaging problems reduce to solving large and noisy linear systems

$$\mathbf{y} = \mathbf{H}\mathbf{f} + \mathbf{e}$$

Setting up the right forward model is a big step towards being able to form high quality images



Today we will talk about

- Forward models in imaging
  Relating the unknowns to the measured data

- **Notions of ill-posedness and regularization**
  When measurements are not enough

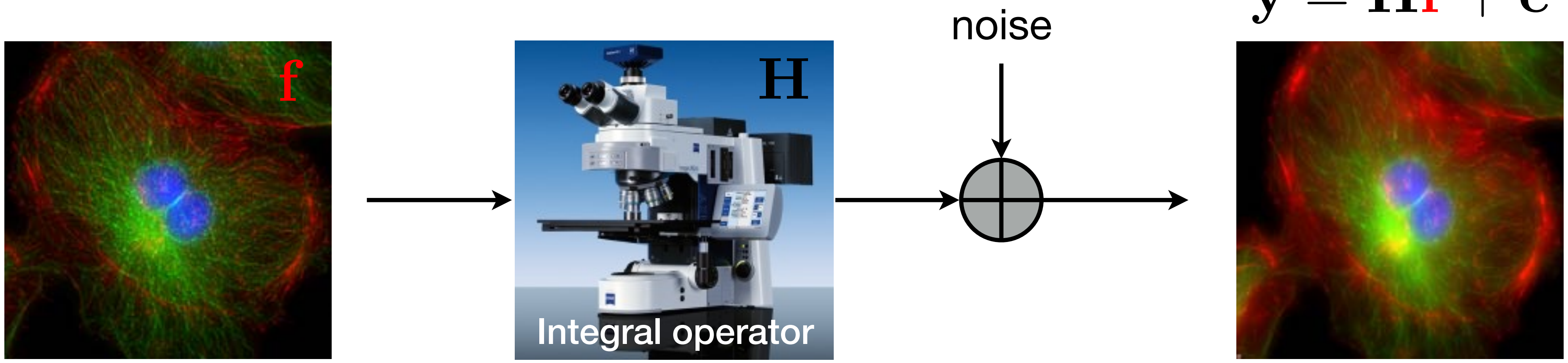
- Optimization at large scales
  When analytical solutions are not enough

- Plug-and-Play Priors (PnP) at large scales
  When traditional optimization is not enough

What makes imaging inverse problems difficult?

What makes imaging inverse problems difficult?

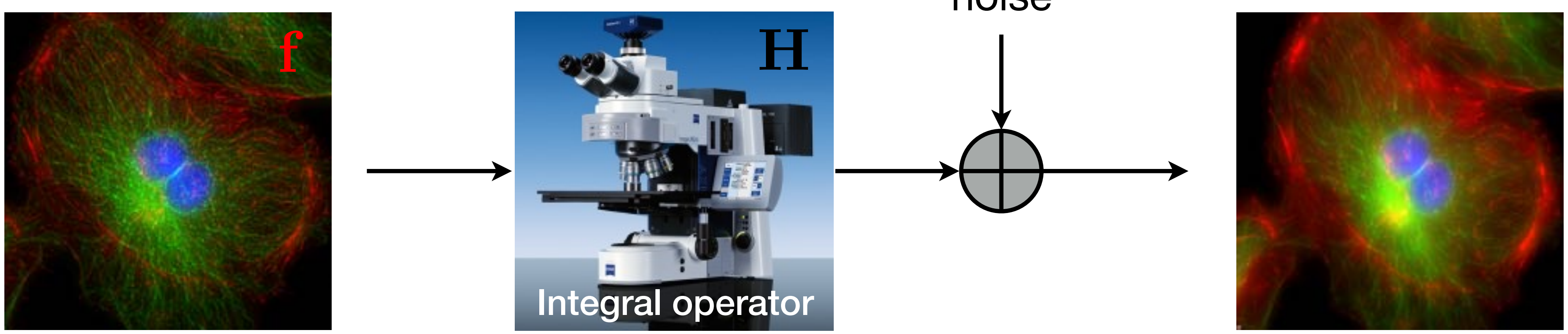
$$\mathbf{y} = \mathbf{H}\mathbf{f} + \mathbf{e}$$



Problem: recover $\mathbf{f}$ from noisy measurements $\mathbf{y}$

What makes imaging inverse problems difficult?

$$\mathbf{y} = \mathbf{H}\mathbf{f} + \mathbf{e}$$

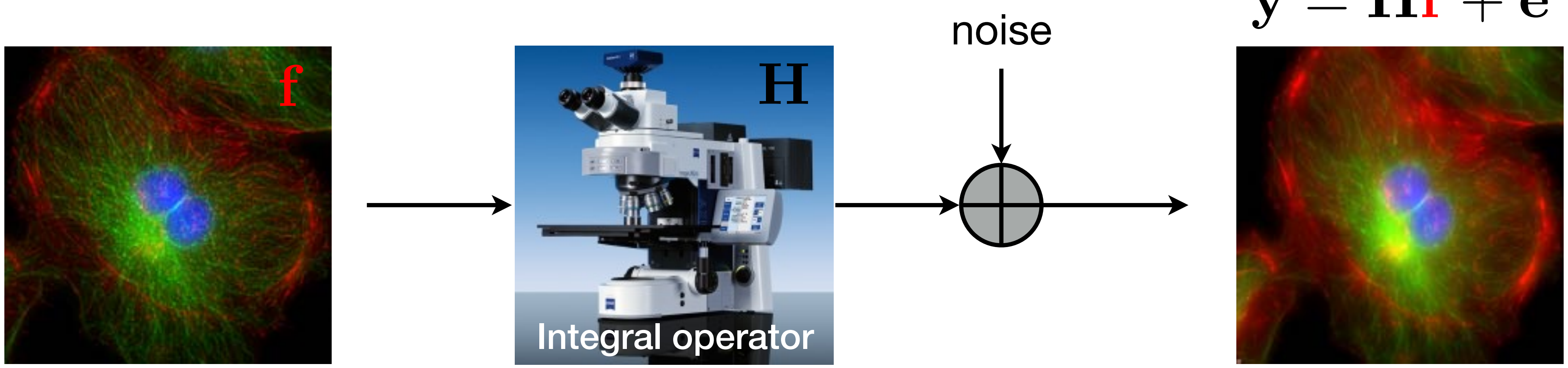


Problem: recover $\mathbf{f}$ from noisy measurements $\mathbf{y}$

Question: Why can't we simply compute the inverse $\mathbf{f} = \mathbf{H}^{-1}\mathbf{y}$?

What makes imaging inverse problems difficult?

$$\mathbf{y} = \mathbf{H}\mathbf{f} + \mathbf{e}$$



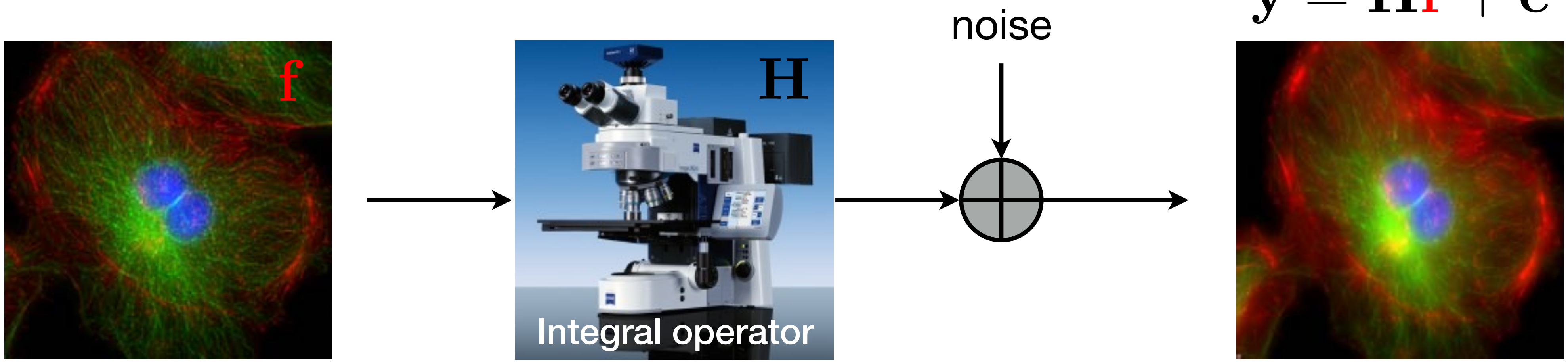
Problem: recover $\mathbf{f}$ from noisy measurements $\mathbf{y}$

Question: Why can't we simply compute the inverse $\mathbf{f} = \mathbf{H}^{-1}\mathbf{y}$?

1) Difficult to invert the matrix as it is non-square or too large

What makes imaging inverse problems difficult?

$$\mathbf{y} = \mathbf{H}\mathbf{f} + \mathbf{e}$$



Problem: recover $\mathbf{f}$ from noisy measurements $\mathbf{y}$

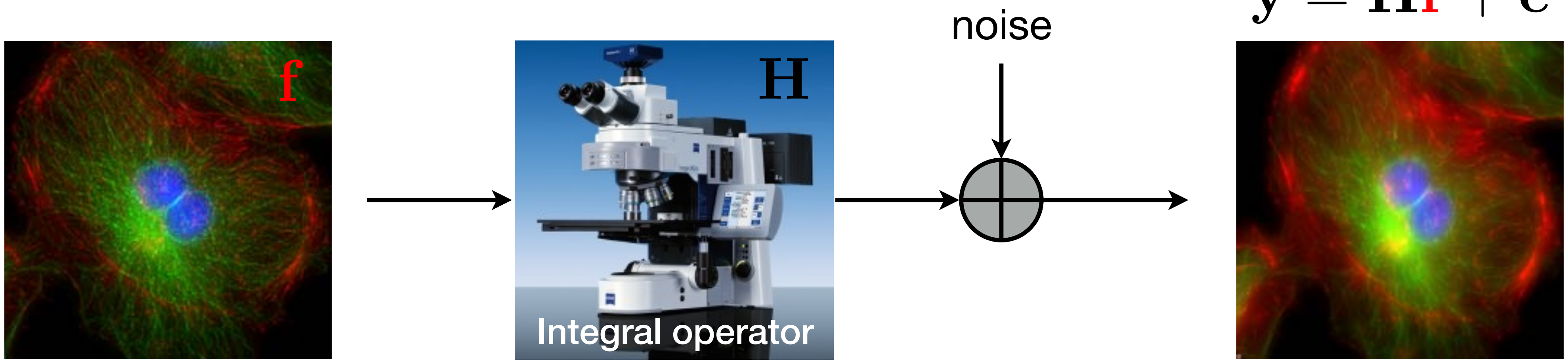
Question: Why can't we simply compute the inverse $\mathbf{f} = \mathbf{H}^{-1}\mathbf{y}$?

1) Difficult to invert the matrix as it is non-square or too large

2) Measurements do not uniquely describe the object

What makes imaging inverse problems difficult?

$$\mathbf{y} = \mathbf{H}\mathbf{f} + \mathbf{e}$$

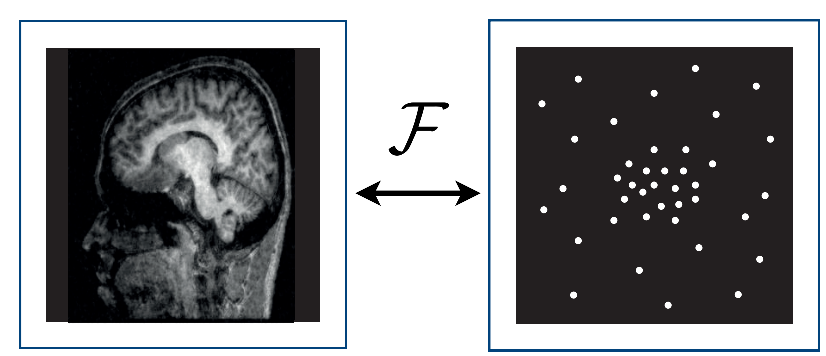


Problem: recover $\mathbf{f}$ from noisy measurements $\mathbf{y}$

Question: Why can't we simply compute the inverse $\mathbf{f} = \mathbf{H}^{-1}\mathbf{y}$?

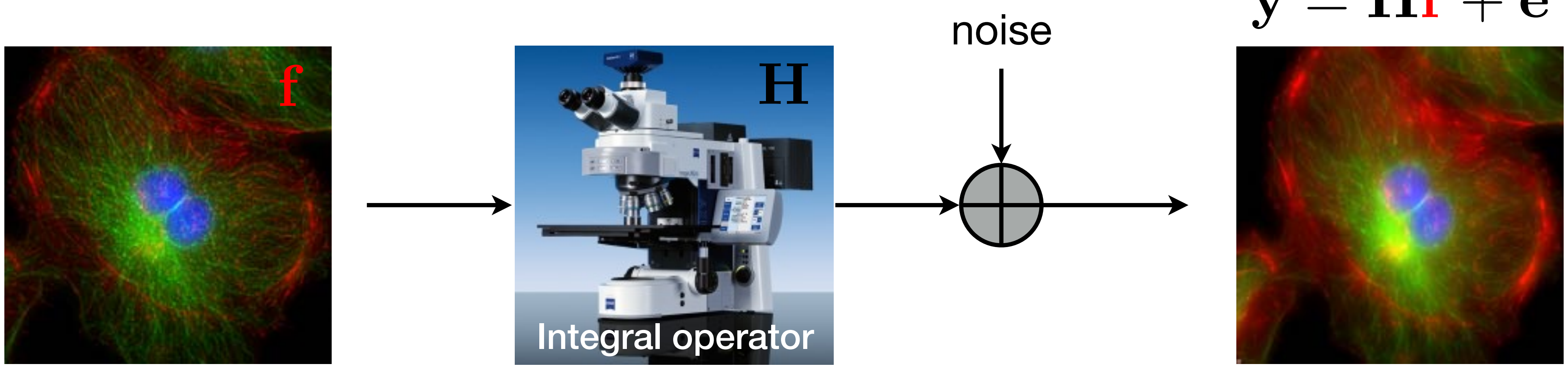
1) Difficult to invert the matrix as it is non-square or too large

2) Measurements do not uniquely describe the object



What makes imaging inverse problems difficult?

$$\mathbf{y} = \mathbf{H}\mathbf{f} + \mathbf{e}$$



Problem: recover $\mathbf{f}$ from noisy measurements $\mathbf{y}$

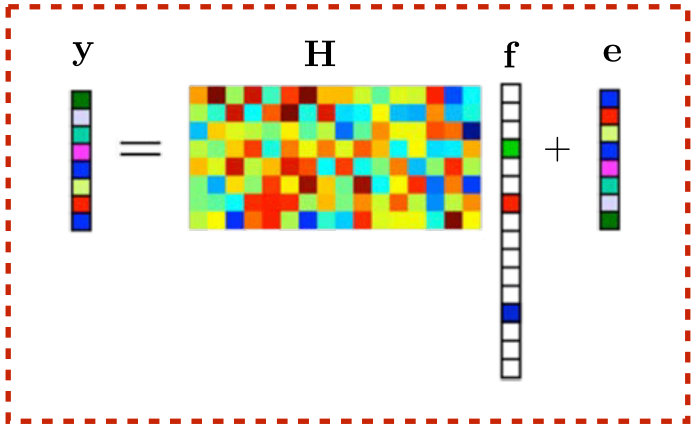
Question: Why can't we simply compute the inverse $\mathbf{f} = \mathbf{H}^{-1}\mathbf{y}$?

1) Difficult to invert the matrix as it is non-square or too large
2) Measurements do not uniquely describe the object
3) Noise amplification (related but not equal to 2)

Regularization framework enables the selection of the most suitable solution among alternatives

Regularization framework enables the selection of the most suitable solution among alternatives

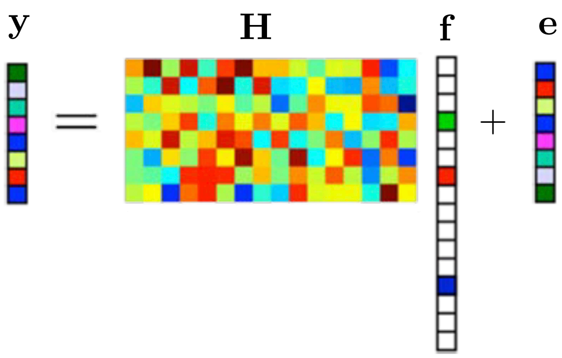
Consider a noisy linear system with noise of bounded norm



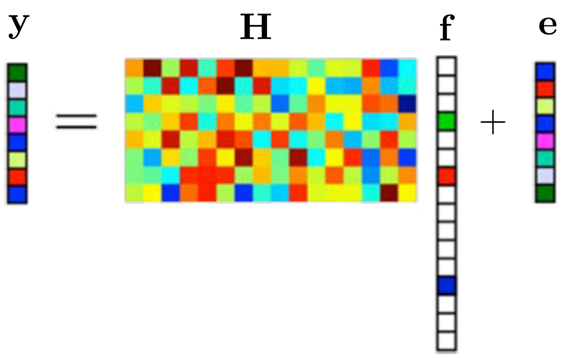
Regularization framework enables the selection of the most suitable solution among alternatives

Consider a noisy linear system
with noise of bounded norm

$$\mathbf{y} = \mathbf{H}\mathbf{f} + \mathbf{e} \quad \text{such that} \quad \|\mathbf{y} - \mathbf{H}\mathbf{f}\|_{\ell_2}^2 \leq \sigma^2$$



Regularization framework enables the selection of the most suitable solution among alternatives



Consider a noisy linear system with noise of bounded norm

$$\mathbf{y} = \mathbf{H}\mathbf{f} + \mathbf{e} \quad \text{such that} \quad \|\mathbf{y} - \mathbf{H}\mathbf{f}\|_{\ell_2}^2 \leq \sigma^2$$

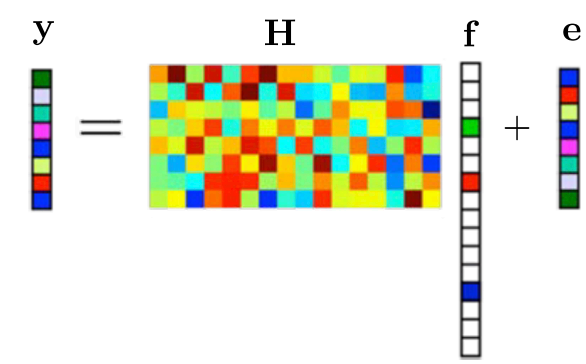
We consider a constrained optimization problem

$$\text{minimize } \mathcal{R}(\mathbf{f}) \text{ subject to } \|\mathbf{H}\mathbf{f} - \mathbf{y}\|_{\ell_2}^2 \leq \sigma^2$$

- The "regularizer" picks the solution which we think is best
- Allows us to infuse prior knowledge into the problem

Regularization framework enables the selection of the most suitable solution among alternatives

Consider a noisy linear system with noise of bounded norm



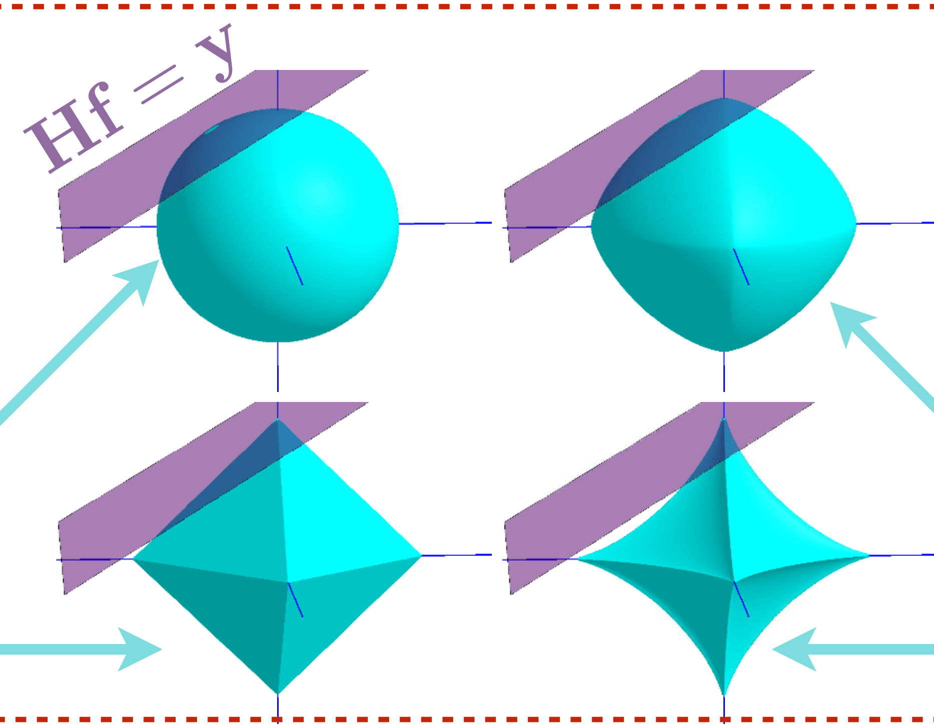
$$\mathbf{y} = \mathbf{H}\mathbf{f} + \mathbf{e} \quad \text{such that} \quad \|\mathbf{y} - \mathbf{H}\mathbf{f}\|_{\ell_2}^2 \leq \sigma^2$$

We consider a constrained optimization problem

$$\text{minimize } \mathcal{R}(\mathbf{f}) \text{ subject to } \|\mathbf{H}\mathbf{f} - \mathbf{y}\|_{\ell_2}^2 \leq \sigma^2$$

Examples when no noise

Elad, "Sparse and Redundant Representations," 2010



$\mathcal{R}(\mathbf{f}) = \|\mathbf{f}\|_{\ell_2}^2$

$\mathcal{R}(\mathbf{f}) = \|\mathbf{f}\|_{\ell_1}$

$\mathcal{R}(\mathbf{f}) = \|\mathbf{f}\|_{\ell_{1.5}}^{1.5}$

$\mathcal{R}(\mathbf{f}) = \|\mathbf{f}\|_{\ell_{0.7}}^{0.7}$

Question: How to regularize in imaging?

Question: How to regularize in imaging?

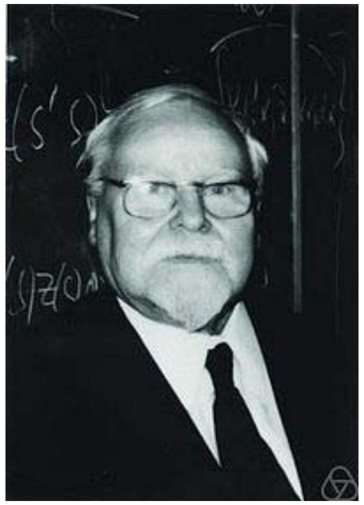
minimize $\mathcal{R}(\mathbf{f})$ subject to $\|\mathbf{Hf} - \mathbf{y}\|_{\ell_2}^2 \leq \sigma^2$

Question: How to regularize in imaging?

Classical approach: Tikhonov regularization

$$\mathcal{R}(\mathbf{f}) = \|\mathbf{D}\mathbf{f}\|_{\ell_2}^2$$

Assumption:
image is smooth



Andrey N. Tikhonov (1906-1993)

minimize $\mathcal{R}(\mathbf{f})$ subject to $\|\mathbf{H}\mathbf{f} - \mathbf{y}\|_{\ell_2}^2 \leq \sigma^2$

Question: How to regularize in imaging?

Classical approach: Tikhonov regularization

$$\mathcal{R}(\mathbf{f}) = \|\mathbf{D}\mathbf{f}\|_{\ell_2}^2 \quad \Rightarrow \quad \widehat{\mathbf{f}}_{\mathsf{Tikh}} = (\mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{H}^\mathsf{T}\left[\mathbf{H}(\mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{H}^\mathsf{T}\right]^{-1}\mathbf{y}$$

unique closed-form solution

minimize $\mathcal{R}(\mathbf{f})$ subject to $\|\mathbf{H}\mathbf{f} - \mathbf{y}\|_{\ell_2}^2 \leq \sigma^2$

Question: How to regularize in imaging?

Classical approach: Tikhonov regularization

$$\mathcal{R}(\mathbf{f}) = \|\mathbf{D}\mathbf{f}\|_{\ell_2}^2 \quad \Rightarrow \quad \widehat{\mathbf{f}}_{\mathsf{Tikh}} = (\mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{H}^\mathsf{T}\left[\mathbf{H}(\mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{H}^\mathsf{T}\right]^{-1}\mathbf{y}$$

Assumption:
image is smooth

Question: Is image smoothness a reasonable assumption?

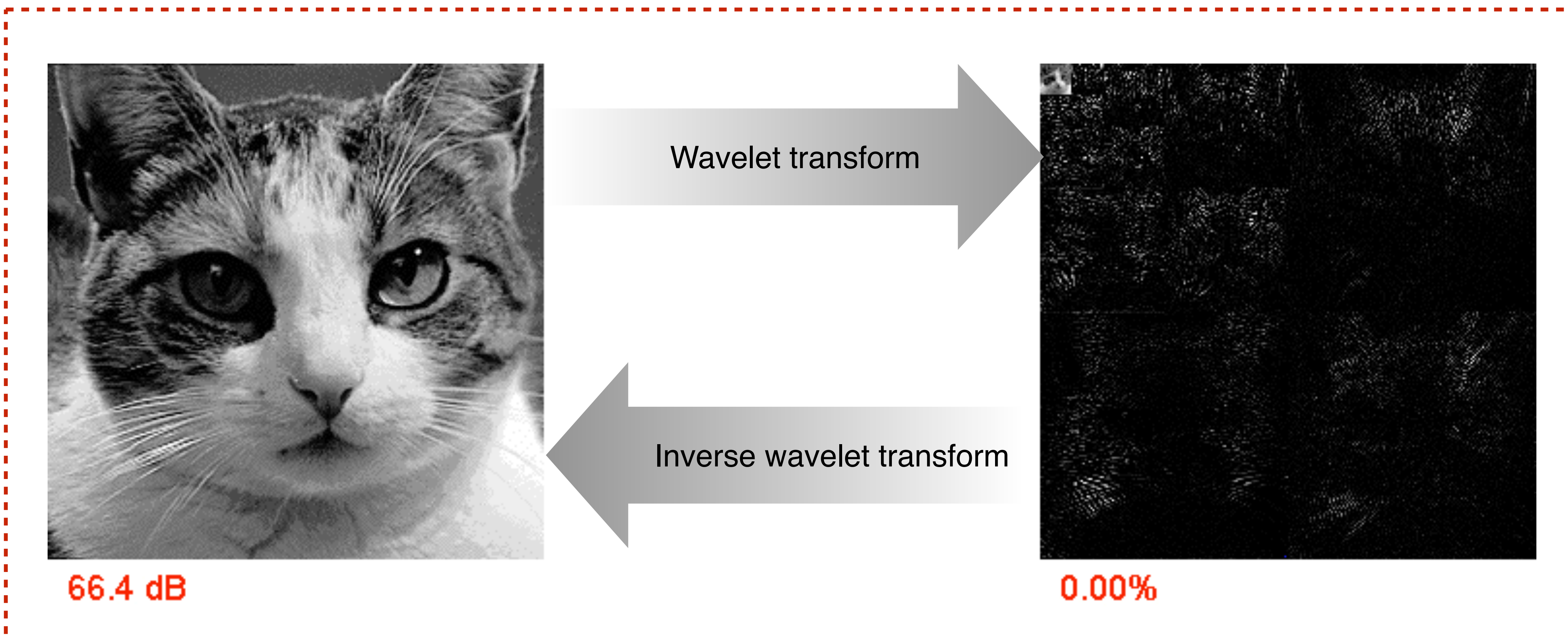
minimize $\mathcal{R}(\mathbf{f})$ subject to $\|\mathbf{H}\mathbf{f} - \mathbf{y}\|_{\ell_2}^2 \leq \sigma^2$

Question: How to regularize in imaging?

Classical approach: Tikhonov regularization

$$\mathcal{R}(\mathbf{f}) = \|\mathbf{D}\mathbf{f}\|_{\ell_2}^2 \quad \Rightarrow \quad \widehat{\mathbf{f}}_{\mathsf{Tikh}} = (\mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{H}^\mathsf{T}\left[\mathbf{H}(\mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{H}^\mathsf{T}\right]^{-1}\mathbf{y}$$

Modern approach: Transform-domain sparsity



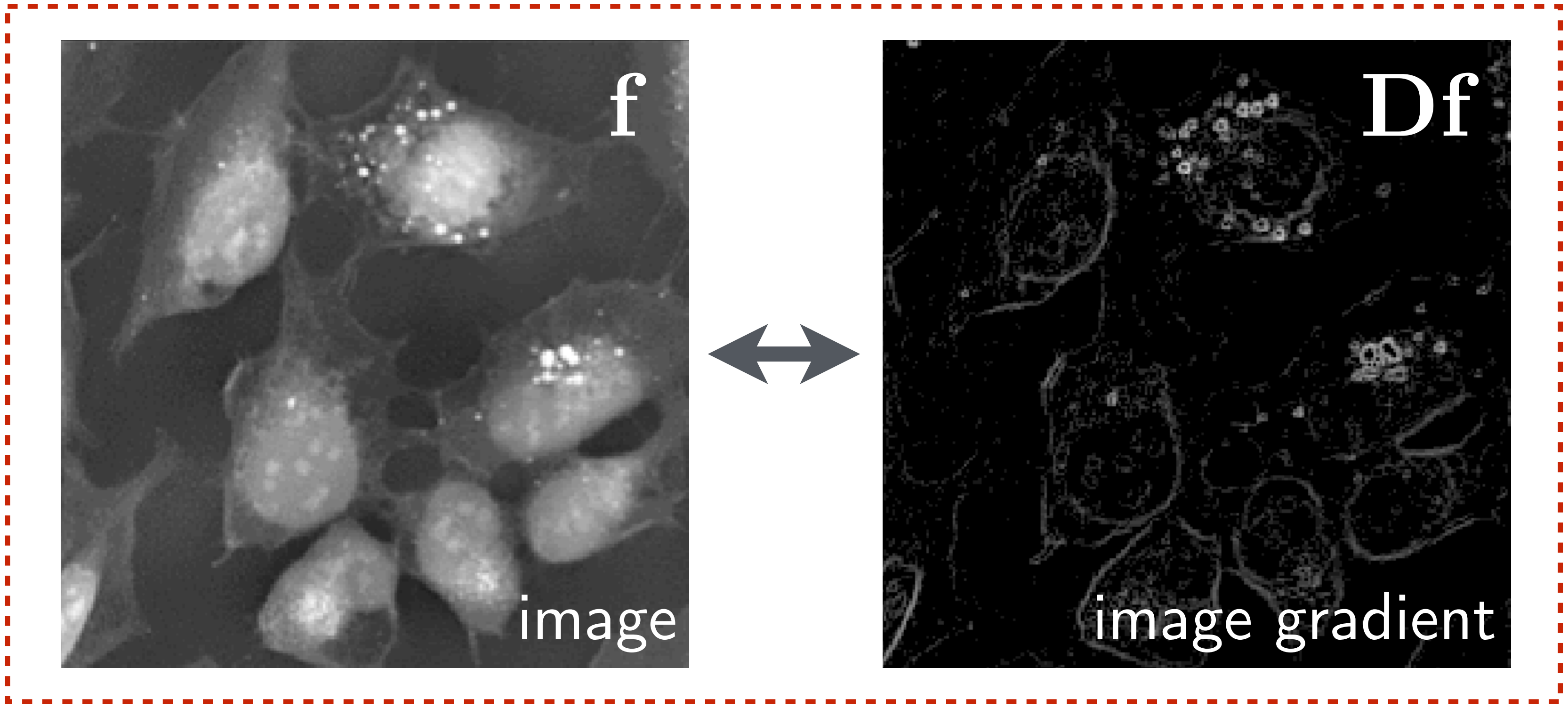
minimize $\mathcal{R}(\mathbf{f})$ subject to $\|\mathbf{H}\mathbf{f} - \mathbf{y}\|_{\ell_2}^2 \leq \sigma^2$

Question: How to regularize in imaging?

Classical approach: Tikhonov regularization

$$\mathcal{R}(\mathbf{f}) = \|\mathbf{D}\mathbf{f}\|_{\ell_2}^2 \quad \Rightarrow \quad \widehat{\mathbf{f}}_{\mathsf{Tikh}} = (\mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{H}^\mathsf{T}\left[\mathbf{H}(\mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{H}^\mathsf{T}\right]^{-1}\mathbf{y}$$

Modern approach: Transform-domain sparsity



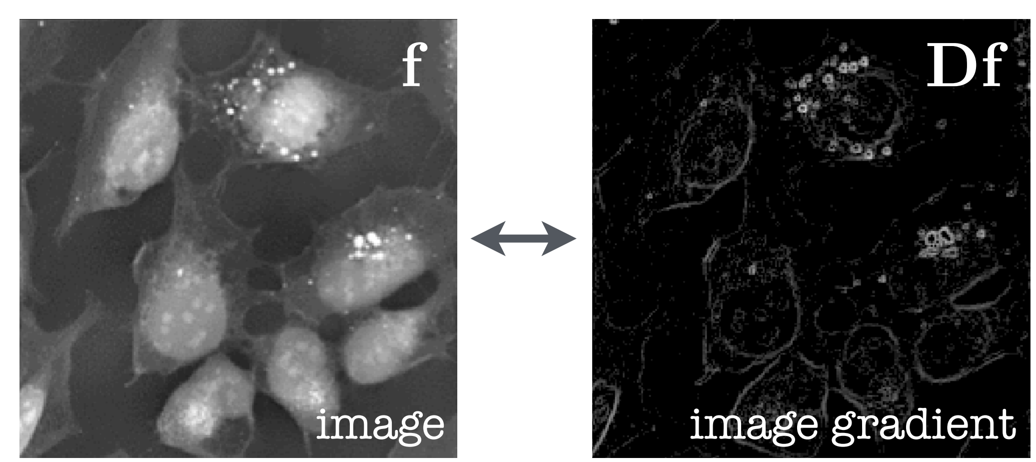
minimize $\mathcal{R}(\mathbf{f})$ subject to $\|\mathbf{H}\mathbf{f} - \mathbf{y}\|_{\ell_2}^2 \leq \sigma^2$

Question: How to regularize in imaging?

Classical approach: Tikhonov regularization

$$\mathcal{R}(\mathbf{f}) = \|\mathbf{D}\mathbf{f}\|_{\ell_2}^2 \quad \Rightarrow \quad \widehat{\mathbf{f}}_{\mathsf{Tikh}} = (\mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{H}^\mathsf{T}\left[\mathbf{H}(\mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{H}^\mathsf{T}\right]^{-1}\mathbf{y}$$

Modern approach: Transform-domain sparsity



Question: How to regularize in imaging?

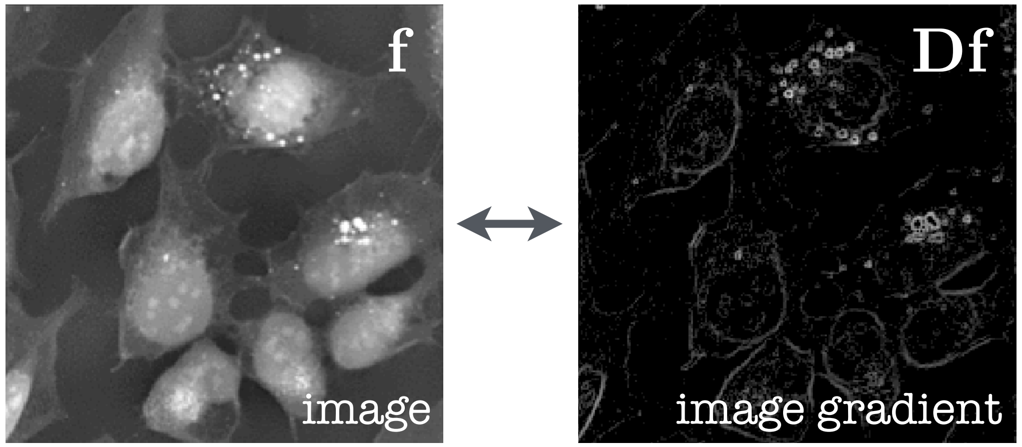
Classical approach: Tikhonov regularization

$$\mathcal{R}(\mathbf{f}) = \|\mathbf{D}\mathbf{f}\|_{\ell_2}^2 \quad \Rightarrow \quad \widehat{\mathbf{f}}_{\mathsf{Tikh}} = (\mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{H}^\mathsf{T}\left[\mathbf{H}(\mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{H}^\mathsf{T}\right]^{-1}\mathbf{y}$$

Modern approach: Transform-domain sparsity

$$\mathcal{R}(\mathbf{f}) = \|\mathbf{D}\mathbf{f}\|_{\ell_0} = \#\{i : [\mathbf{D}\mathbf{f}]_i \neq 0\}$$

intractable nonconvex optimiazation



Question: How to regularize in imaging?

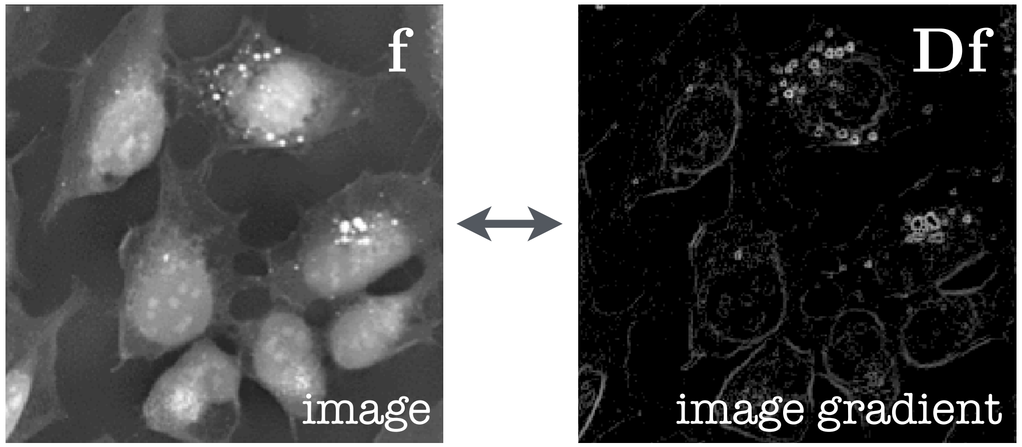
Classical approach: Tikhonov regularization

$$\mathcal{R}(\mathbf{f}) = \|\mathbf{D}\mathbf{f}\|_{\ell_2}^2 \quad \Rightarrow \quad \widehat{\mathbf{f}}_{\mathsf{Tikh}} = (\mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{H}^\mathsf{T}\left[\mathbf{H}(\mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{H}^\mathsf{T}\right]^{-1}\mathbf{y}$$

Modern approach: Transform-domain sparsity

$$\mathcal{R}(\mathbf{f}) = \|\mathbf{D}\mathbf{f}\|_{\ell_1}$$

- convex (but nondifferentiable)
- promotes sparsity



To conclude "regularization"

Many imaging problems are ill-posed: there are infinitely many solutions

$$\mathbf{y} = \mathbf{H}\mathbf{f} + \mathbf{e}$$

Regularization is a strategy to select the solution that "makes sense"

$$\text{minimize } \mathcal{R}(\mathbf{f}) \text{ subject to } \|\mathbf{H}\mathbf{f} - \mathbf{y}\|_{\ell_2}^2 \leq \sigma^2$$

Classical image regularizers are linear, but increasingly they are nonlinear

(20th) $\mathcal{R}(\mathbf{f}) = \|\mathbf{D}\mathbf{f}\|_{\ell_2}^2 \quad \Rightarrow \quad \mathcal{R}(\mathbf{f}) = \|\mathbf{D}\mathbf{f}\|_{\ell_1}$ (21st)

Today we will talk about

- Forward models in imaging
  Relating the unknowns to the measured data
- Notions of ill-posedness and regularization
  When measurements are not enough
- **Optimization at large scales**
  When analytical solutions are not enough
- Plug-and-Play Priors (PnP) at large scales
  When traditional optimization is not enough

Proximal operator corresponds to image denoising

Proximal operator corresponds to image denoising

A more convenient formulation

$$\text{min } \mathcal{R}(\mathbf{f}) \text{ subject to } \|\mathbf{Hf} - \mathbf{y}\|_{\ell_2}^2 \leq \sigma^2 \Leftrightarrow \min_{\mathbf{f}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{Hf}\|_{\ell_2}^2 + \lambda \mathcal{R}(\mathbf{f}) \right\}$$

constrained optimization

unconstrained optimization

Proximal operator corresponds to image denoising

A more convenient formulation

$$\min \mathcal{R}(\mathbf{f}) \text{ subject to } \|\mathbf{Hf} - \mathbf{y}\|_{\ell_2}^2 \leq \sigma^2 \;\Leftrightarrow\; \min_{\mathbf{f}} \left\{ \frac{1}{2}\|\mathbf{y} - \mathbf{Hf}\|_{\ell_2}^2 + \lambda \mathcal{R}(\mathbf{f}) \right\}$$

Image denoising corresponds to identity measurement matrix

$$\min_{\mathbf{f}} \left\{ \frac{1}{2}\|\mathbf{y} - \mathbf{f}\|_{\ell_2}^2 + \lambda \mathcal{R}(\mathbf{f}) \right\}$$

Question: Can you comment on convexity?

Proximal operator corresponds to image denoising

A more convenient formulation

$$\text{min } \mathcal{R}(\mathbf{f}) \text{ subject to } \|\mathbf{Hf}-\mathbf{y}\|_{\ell_2}^2 \leq \sigma^2 \;\Leftrightarrow\; \min_{\mathbf{f}} \left\{ \frac{1}{2}\|\mathbf{y}-\mathbf{Hf}\|_{\ell_2}^2 + \lambda \mathcal{R}(\mathbf{f}) \right\}$$

Image denoising corresponds to identity measurement matrix

$$\min_{\mathbf{f}} \left\{ \frac{1}{2}\|\mathbf{y}-\mathbf{f}\|_{\ell_2}^2 + \lambda \mathcal{R}(\mathbf{f}) \right\}$$

For a convex regularizer,
the objective is strongly convex
=> there is a unique minimizer

Proximal operator corresponds to image denoising

A more convenient formulation

$$\text{min } \mathcal{R}(\mathbf{f}) \text{ subject to } \|\mathbf{Hf}-\mathbf{y}\|_{\ell_2}^2 \leq \sigma^2 \;\Leftrightarrow\; \min_{\mathbf{f}} \left\{ \frac{1}{2}\|\mathbf{y}-\mathbf{Hf}\|_{\ell_2}^2 + \lambda\mathcal{R}(\mathbf{f}) \right\}$$

Image denoising corresponds to identity measurement matrix

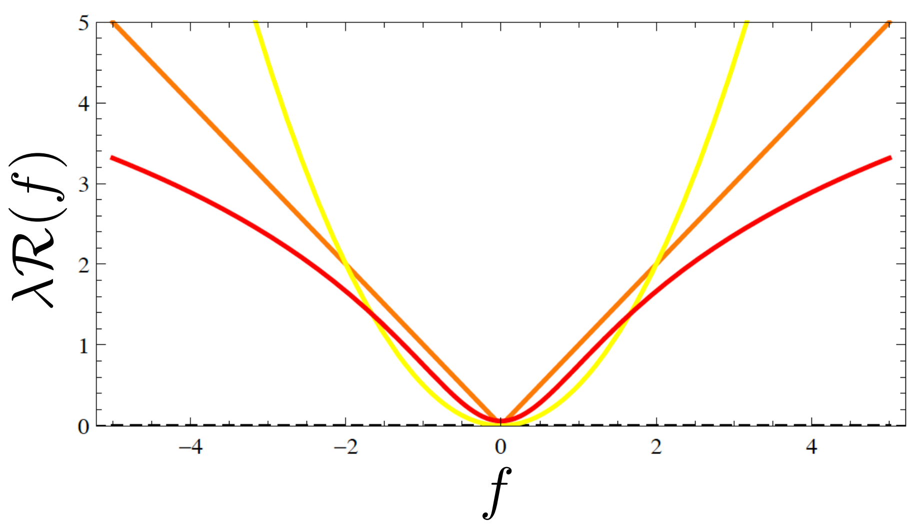
$$\min_{\mathbf{f}} \left\{ \frac{1}{2}\|\mathbf{y}-\mathbf{f}\|_{\ell_2}^2 + \lambda\mathcal{R}(\mathbf{f}) \right\}$$

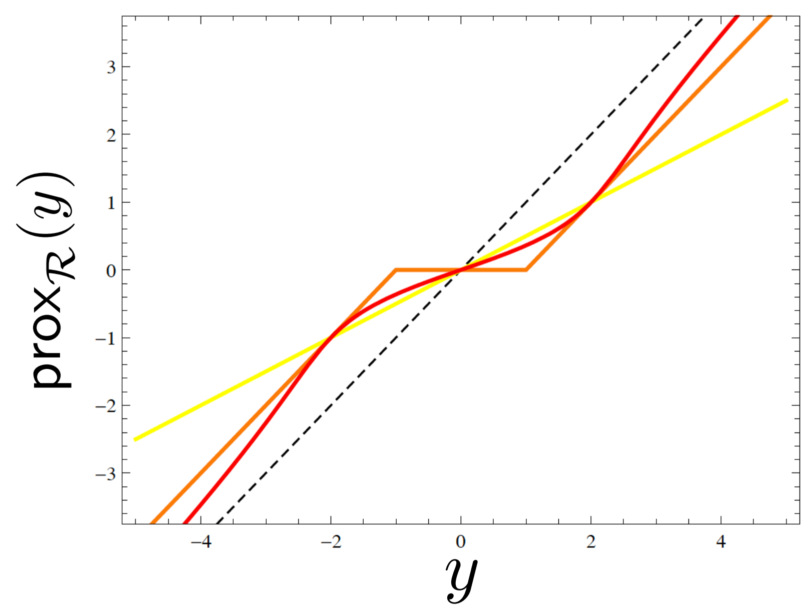
We can thus define the prox operator that solves the denoising problem

$$\text{prox}_{\lambda\mathcal{R}}(\mathbf{y}) \triangleq \arg\min_{\mathbf{f}} \left\{ \frac{1}{2}\|\mathbf{y}-\mathbf{f}\|_{\ell_2}^2 + \lambda\mathcal{R}(\mathbf{f}) \right\}$$

Proximal operator corresponds to image denoising

Some examples of poitwise proximals





linear attenuation		ℓ_2 minimization
soft-threshold		ℓ_1 minimization
shrinkage function	$\approx$	ℓ_p relaxation for $p \to 0$

Unser and Tafti, "An Introduction to Sparse Stochastic Processes," 2014

FISTA and ADMM are two popular algorithms for large-scale and nonsmooth optimization

FISTA and ADMM are two popular algorithms for large-scale and nonsmooth optimization

Consider the objective function

$$\mathcal{C}(\mathbf{f}) = \mathcal{D}(\mathbf{f}) + \mathcal{R}(\mathbf{f}) \quad \text{where} \quad \mathcal{D}(\mathbf{f}) \triangleq \frac{1}{2}\|\mathbf{H}\mathbf{f} - \mathbf{y}\|_{\ell_2}^2$$

data fit + regularizer

FISTA and ADMM are two popular algorithms for large-scale and nonsmooth optimization

Consider the objective function

$$\mathcal{C}(\mathbf{f}) = \mathcal{D}(\mathbf{f}) + \mathcal{R}(\mathbf{f}) \quad \text{where} \quad \mathcal{D}(\mathbf{f}) \triangleq \frac{1}{2}\|\mathbf{H}\mathbf{f} - \mathbf{y}\|_{\ell_2}^2$$

Fast iterative shrinkage/thresholding algorithm (FISTA) vs.
Alternating direction method of multipliers (ADMM)

$$\mathbf{z}^k \leftarrow \mathbf{s}^{k-1} - \gamma \nabla \mathcal{D}(\mathbf{s}^{k-1})$$
$$\mathbf{f}^k \leftarrow \mathsf{prox}_{\gamma\mathcal{R}}(\mathbf{z}^k)$$
$$\mathbf{s}^k \leftarrow \mathbf{f}^k + ((q_{k-1} - 1)/q_k)(\mathbf{f}^k - \mathbf{f}^{k-1})$$

ISTA: $q_k = 1 \Rightarrow O(1/t)$
FISTA: specific $q_k \Rightarrow O(1/t^2)$

$$\mathbf{z}^k \leftarrow \mathsf{prox}_{\gamma\mathcal{D}}(\mathbf{f}^{k-1} - \mathbf{s}^{k-1})$$
$$\mathbf{f}^k \leftarrow \mathsf{prox}_{\gamma\mathcal{R}}(\mathbf{z}^k + \mathbf{s}^{k-1})$$
$$\mathbf{s}^k \leftarrow \mathbf{s}^{k-1} + (\mathbf{z}^k - \mathbf{f}^k)$$

ADMM fast practical convergence

FISTA and ADMM are two popular algorithms for large-scale and nonsmooth optimization

Consider the objective function

$$\mathcal{C}(\mathbf{f}) = \mathcal{D}(\mathbf{f}) + \mathcal{R}(\mathbf{f}) \quad \text{where} \quad \mathcal{D}(\mathbf{f}) \triangleq \frac{1}{2}\|\mathbf{H}\mathbf{f} - \mathbf{y}\|_{\ell_2}^2$$

Fast iterative shrinkage/thresholding algorithm (FISTA) vs. Alternating direction method of multipliers (ADMM)

FISTA:

$$\mathbf{z}^k \leftarrow \mathbf{s}^{k-1} - \gamma \nabla \mathcal{D}(\mathbf{s}^{k-1})$$
$$\mathbf{f}^k \leftarrow \text{prox}_{\gamma\mathcal{R}}(\mathbf{z}^k)$$
$$\mathbf{s}^k \leftarrow \mathbf{f}^k + ((q_{k-1} - 1)/q_k)(\mathbf{f}^k - \mathbf{f}^{k-1})$$

ADMM:

$$\mathbf{z}^k \leftarrow \text{prox}_{\gamma\mathcal{D}}(\mathbf{f}^{k-1} - \mathbf{s}^{k-1})$$
$$\mathbf{f}^k \leftarrow \text{prox}_{\gamma\mathcal{R}}(\mathbf{z}^k + \mathbf{s}^{k-1})$$
$$\mathbf{s}^k \leftarrow \mathbf{s}^{k-1} + (\mathbf{z}^k - \mathbf{f}^k)$$

Question: Which one is computationally more efficient?

FISTA and ADMM are two popular algorithms for large-scale and nonsmooth optimization

Consider the objective function

$$\mathcal{C}(\mathbf{f}) = \mathcal{D}(\mathbf{f}) + \mathcal{R}(\mathbf{f}) \quad \text{where} \quad \mathcal{D}(\mathbf{f}) \triangleq \frac{1}{2}\|\mathbf{H}\mathbf{f} - \mathbf{y}\|_{\ell_2}^2$$

Fast iterative shrinkage/thresholding algorithm (FISTA) vs. Alternating direction method of multipliers (ADMM)

FISTA:

$$\mathbf{z}^k \leftarrow \mathbf{s}^{k-1} - \gamma \nabla \mathcal{D}(\mathbf{s}^{k-1})$$
$$\mathbf{f}^k \leftarrow \mathsf{prox}_{\gamma\mathcal{R}}(\mathbf{z}^k)$$
$$\mathbf{s}^k \leftarrow \mathbf{f}^k + ((q_{k-1} - 1)/q_k)(\mathbf{f}^k - \mathbf{f}^{k-1})$$

ADMM:

$$\mathbf{z}^k \leftarrow \mathsf{prox}_{\gamma\mathcal{D}}(\mathbf{f}^{k-1} - \mathbf{s}^{k-1})$$
$$\mathbf{f}^k \leftarrow \mathsf{prox}_{\gamma\mathcal{R}}(\mathbf{z}^k + \mathbf{s}^{k-1})$$
$$\mathbf{s}^k \leftarrow \mathbf{s}^{k-1} + (\mathbf{z}^k - \mathbf{f}^k)$$

Per-iteration complexity of ADMM is generally higher

$$\nabla \mathcal{D}(\mathbf{f}) = \mathbf{H}^{\mathsf{T}}(\mathbf{H}\mathbf{f} - \mathbf{y})$$

$$\mathsf{prox}_{\gamma\mathcal{D}}(\mathbf{f}) = [\mathbf{I} + \gamma \mathbf{H}^{\mathsf{T}}\mathbf{H}]^{-1}(\mathbf{f} + \gamma \mathbf{H}^{\mathsf{T}}\mathbf{y})$$

requires matrix inversion

To conclude "optimization"

Many imaging problems are ill-posed:
there are infinitely many solutions

$$\mathbf{y} = \mathbf{H}\mathbf{f} + \mathbf{e}$$

Regularization is a strategy to select the solution that "makes sense"

$$\text{minimize } \mathcal{R}(\mathbf{f}) \text{ subject to } \|\mathbf{H}\mathbf{f} - \mathbf{y}\|_{\ell_2}^2 \leq \sigma^2$$

Classical image regularizers are linear,
but increasingly they are nonlinear

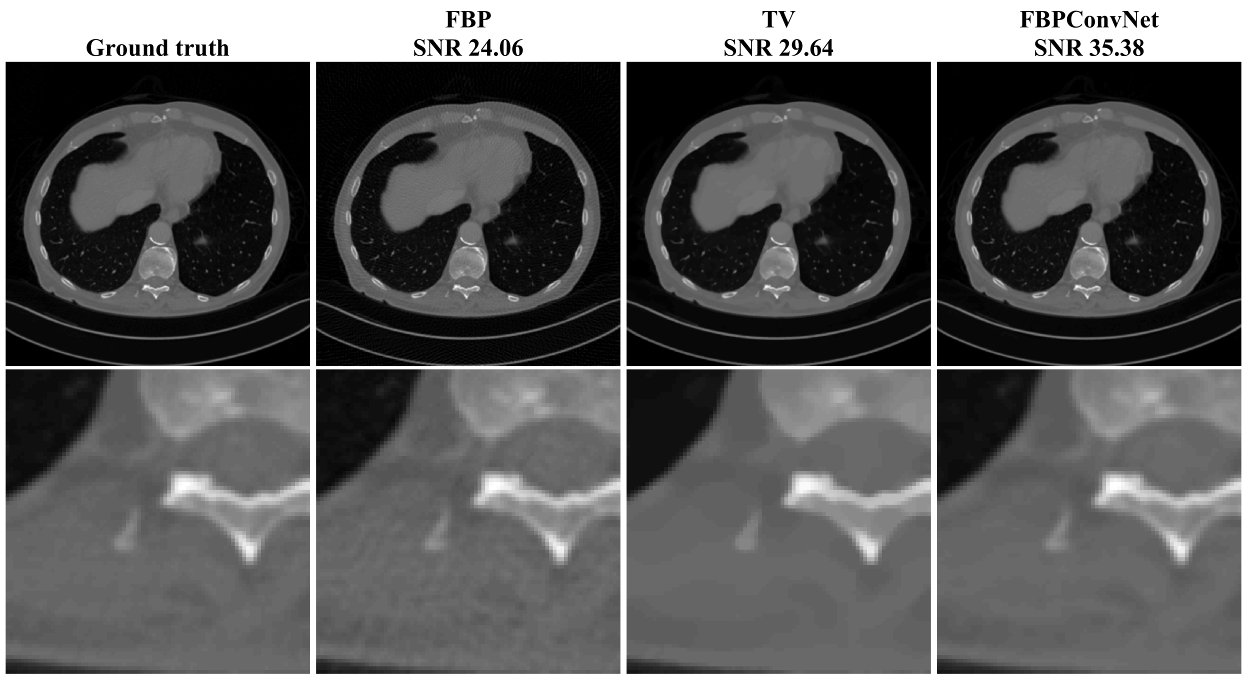
(20th) $\mathcal{R}(\mathbf{f}) = \|\mathbf{D}\mathbf{f}\|_{\ell_2}^2 \quad \Rightarrow \quad \mathcal{R}(\mathbf{f}) = \|\mathbf{D}\mathbf{f}\|_{\ell_1}$ (21st)

Today we will talk about

- Forward models in imaging
  Relating the unknowns to the measured data
- Notions of ill-posedness and regularization
  When measurements are not enough
- Optimization at large scales
  When analytical solutions are not enough
- Plug-and-Play Priors (PnP) at large scales
  When traditional optimization is not enough

Deep learning is currently getting the best performance for image reconstruction

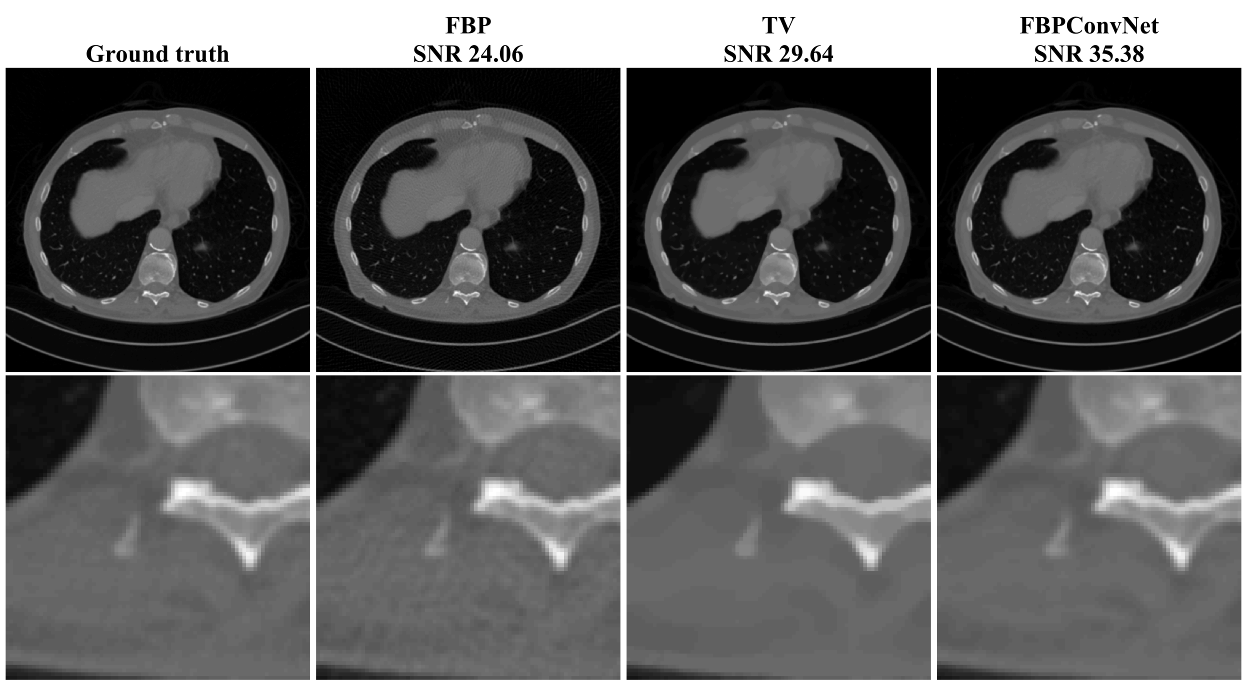
Deep learning is currently getting the best performance for image reconstruction



X-Ray CT

Jin *et al.*, 2016

Deep learning is currently getting the best performance for image reconstruction

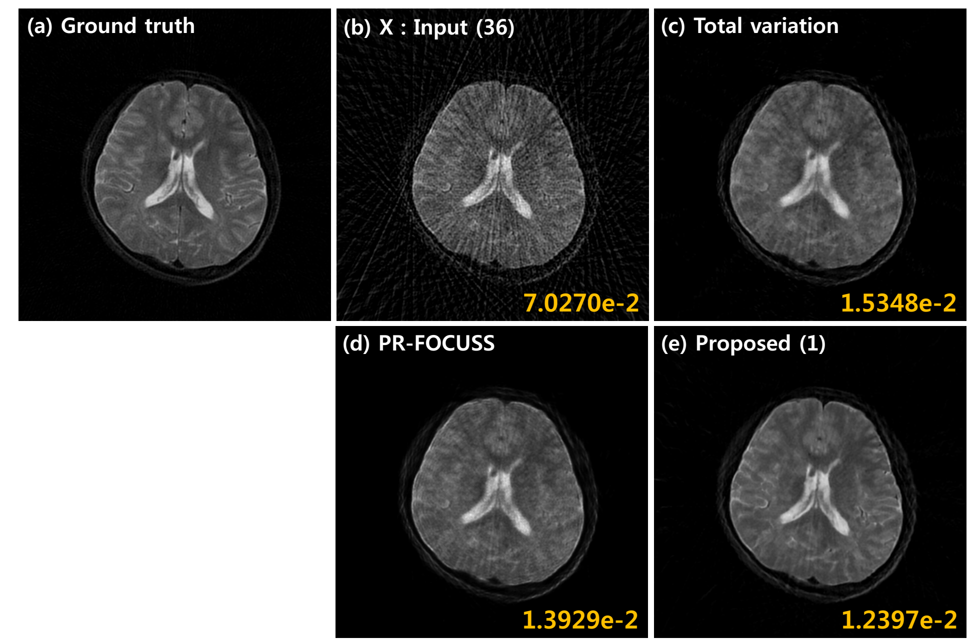


X-Ray CT

Jin *et al.*, 2016

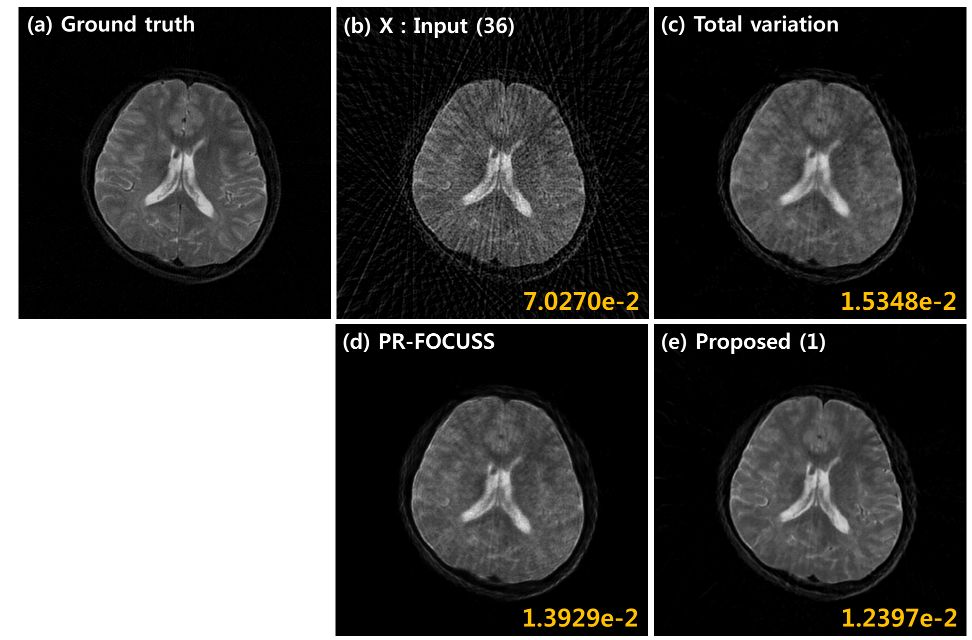
MRI

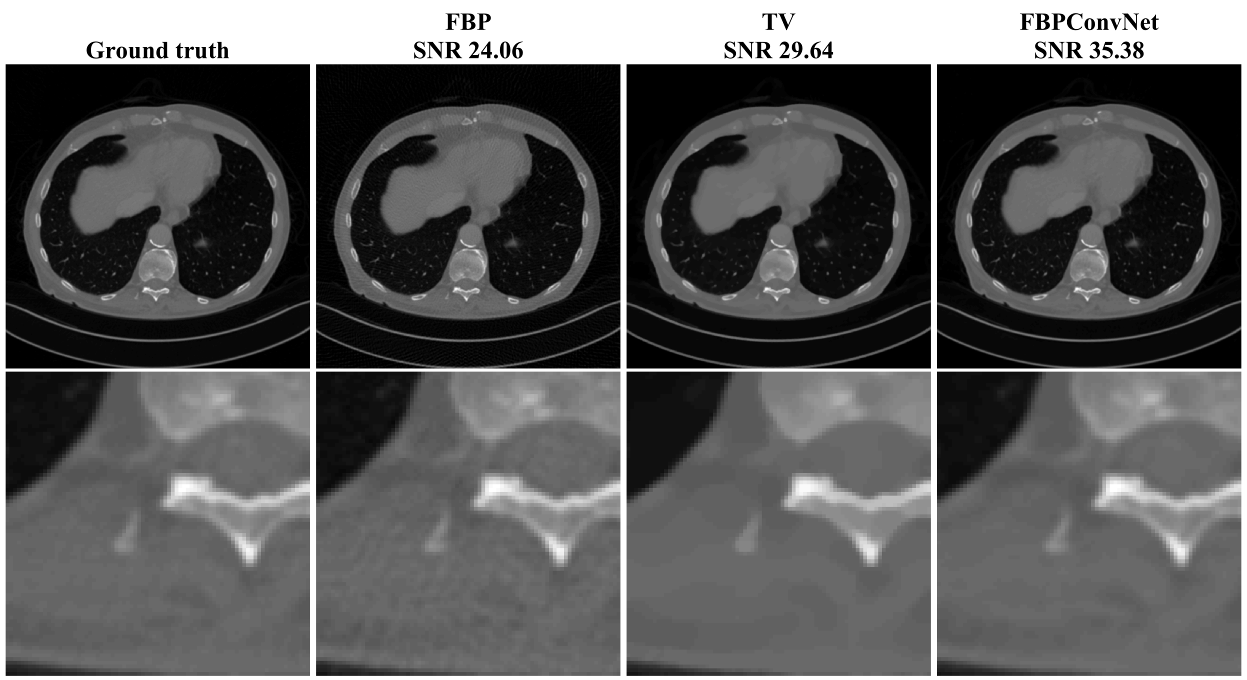
Han *et al.*, 2017



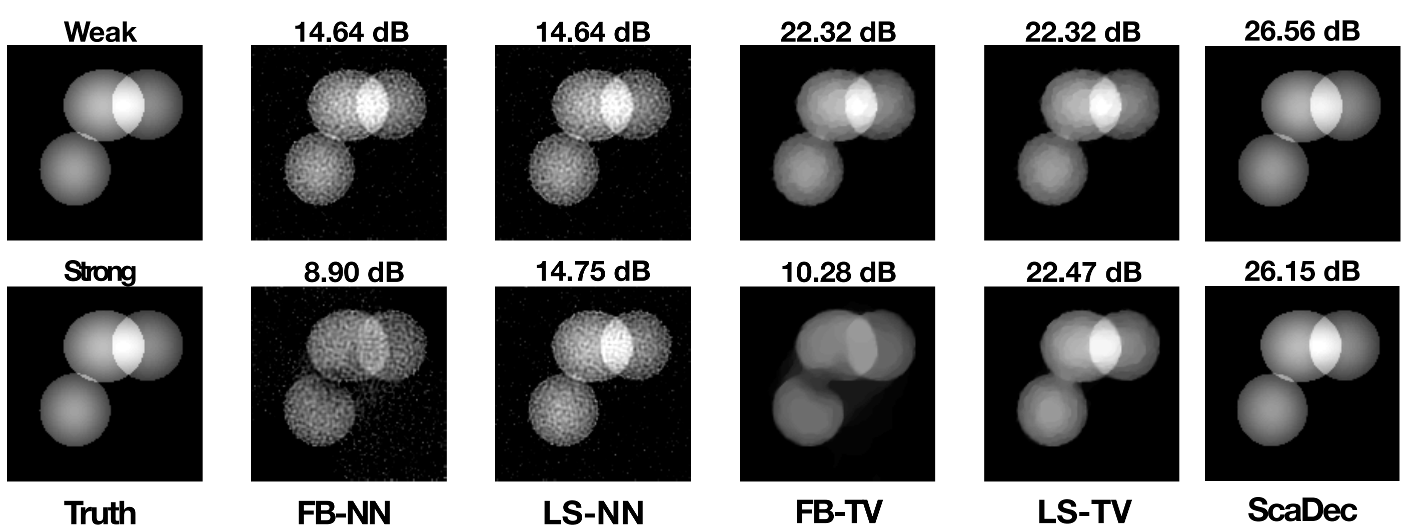
Deep learning is currently getting the best performance for image reconstruction

MRI Han *et al.*, 2017





X-Ray CT Jin *et al.*, 2016

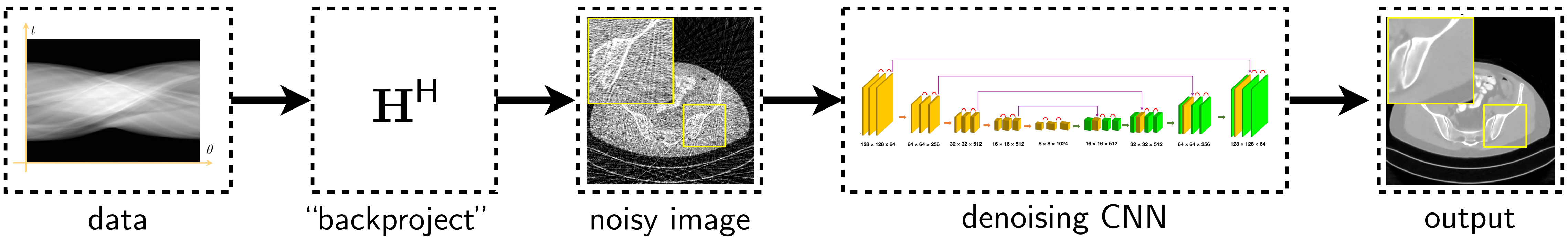


Diffraction Tomography Sun *et al.*, 2018

A well established deep learning pipeline: first backproject then denoise with a ConvNet

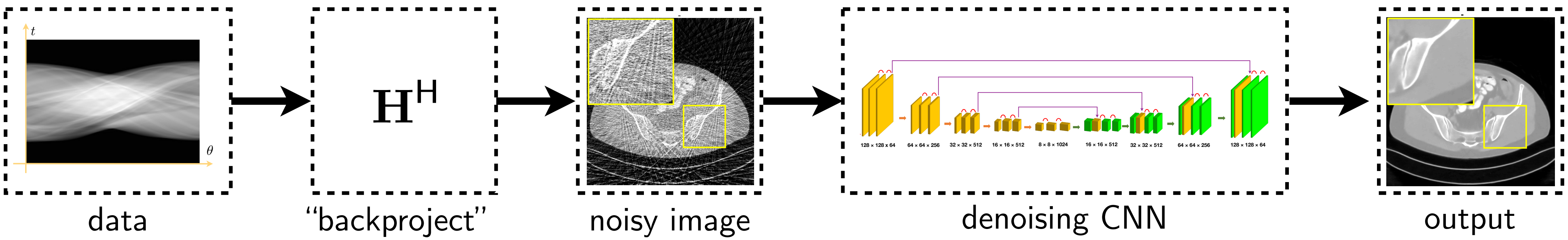
A well established deep learning pipeline: first backproject then denoise with a ConvNet

Data processing pipeline



A well established deep learning pipeline: first backproject then denoise with a ConvNet

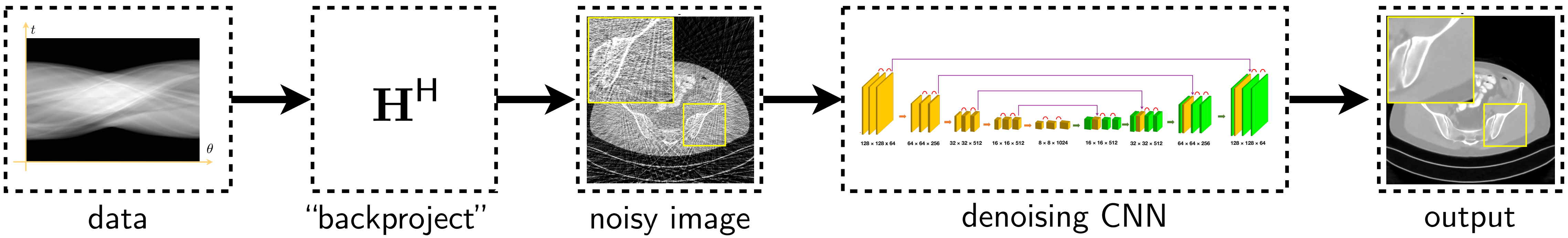
Data processing pipeline



Question: What are some of the key limitations of this approach?

A well established deep learning pipeline: first backproject then denoise with a ConvNet

Data processing pipeline



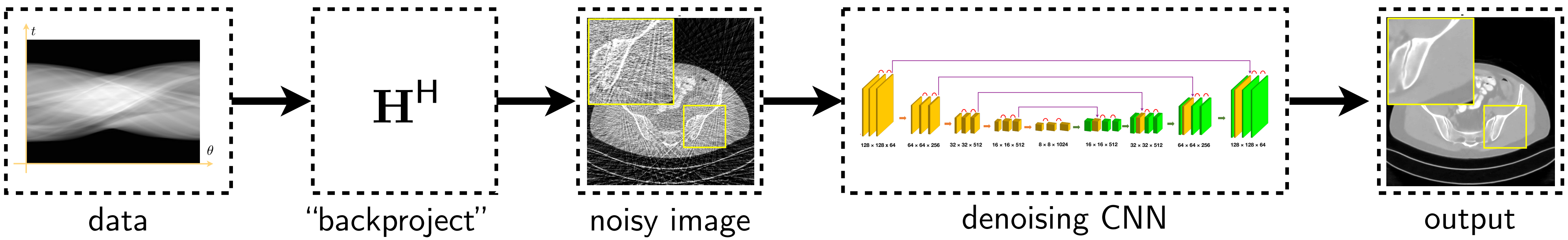
1) Implicit dependance of CNN on the forward model

Hard to decouple the individual contributions of D and R

Washington University in St. Louis

A well established deep learning pipeline: first backproject then denoise with a ConvNet

Data processing pipeline



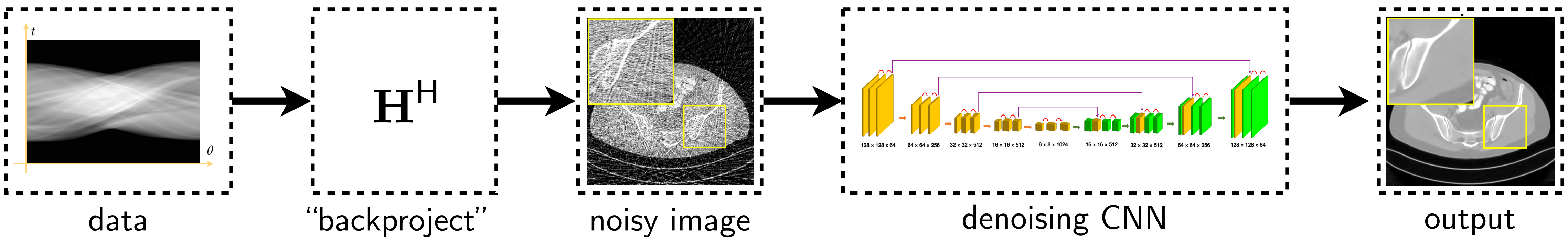
1) Implicit dependance of CNN on the forward model

2) Consistency with the measured data is unclear

No explicit measure of the deviation from the data

A well established deep learning pipeline: first backproject then denoise with a ConvNet

Data processing pipeline



1) Implicit dependance of CNN on the forward model

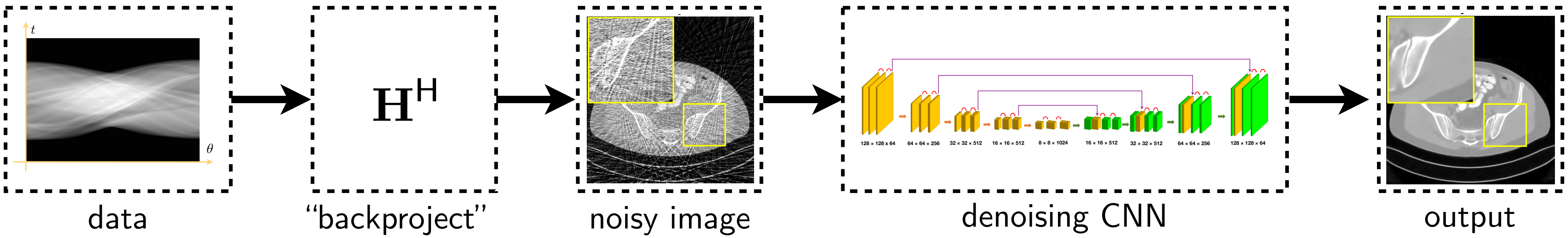
2) Consistency with the measured data is unclear

3) Difficult to impose nontrivial hard constraints on the image

Example: We absolutely need the image gradient to be smaller than epsilon

A well established deep learning pipeline: first backproject then denoise with a ConvNet

Data processing pipeline



1) Implicit dependance of CNN on the forward model

2) Consistency with the measured data is unclear

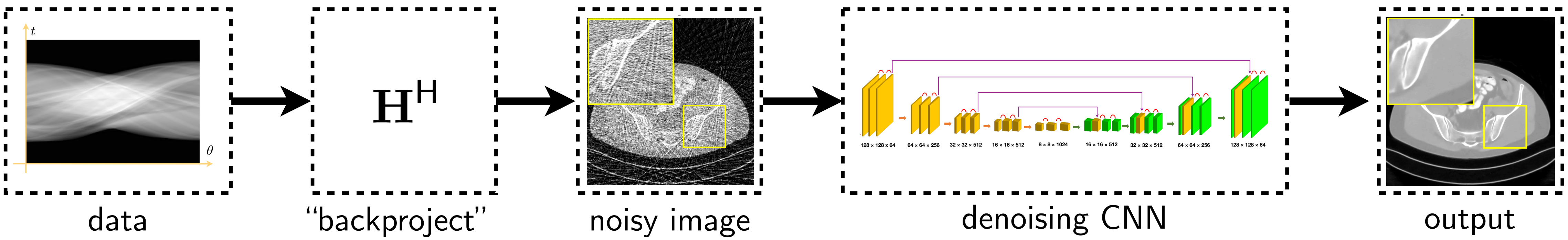
3) Difficult to impose nontrivial hard constraints on the image

4) Not principled: how to select the right architecture?

Variations in the problem are not explicitly linked to model parameters

A well established deep learning pipeline: first backproject then denoise with a ConvNet

Data processing pipeline



1) Implicit dependance of CNN on the forward model

2) Consistency with the measured data is unclear

3) Difficult to impose nontrivial hard constraints on the image

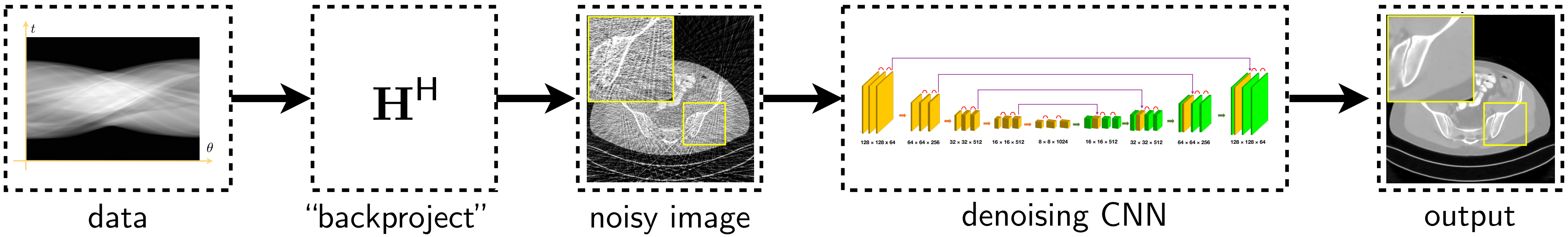
4) Not principled: how to select the right architecture?

5) Difficult to generalize to nonlinear forward models

What happens if there is no backprojection?

A well established deep learning pipeline: first backproject then denoise with a ConvNet

Data processing pipeline

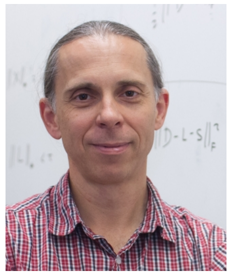


1) Implicit dependance of CNN on the forward model
2) Consistency with the measured data is unclear
3) Difficult to impose nontrivial hard constraints on the image
4) Not principled: how to select the right architecture?
5) Difficult to generalize to nonlinear forward models

Treating the denoising CNN as a proximal operator allows to separate the prior from the forward model

Treating the denoising CNN as a proximal operator allows to separate the prior from the forward model

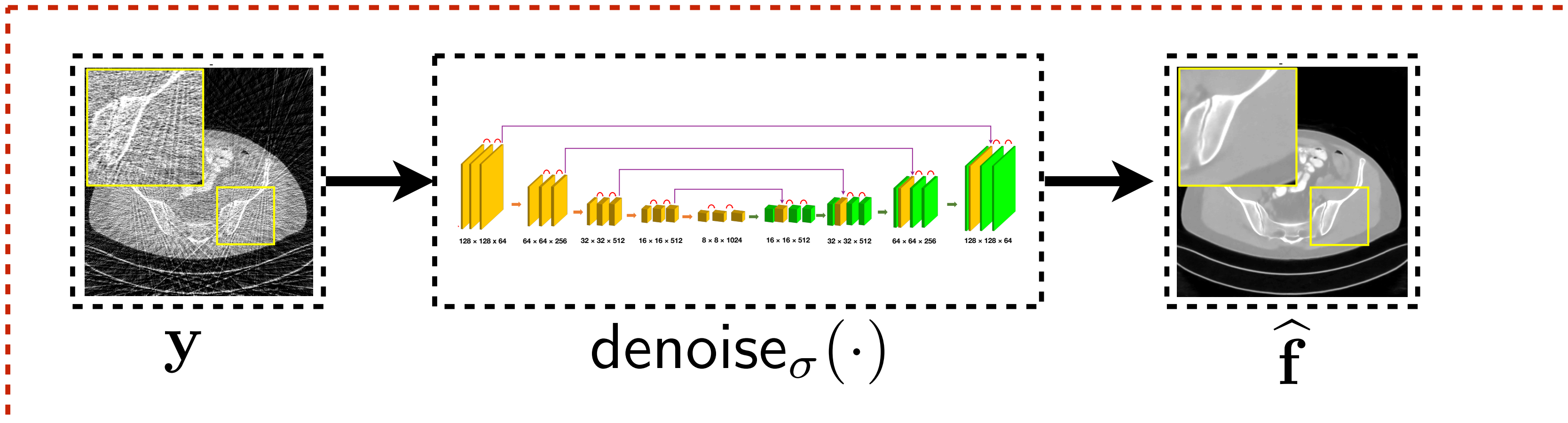




Venkatakrishnan *et al.*, “Plug-and-Play Priors for Model Based Reconstruction,” 2013.

Treating the denoising CNN as a proximal operator allows to separate the prior from the forward model

Train a CNN to denoise for various noise levels

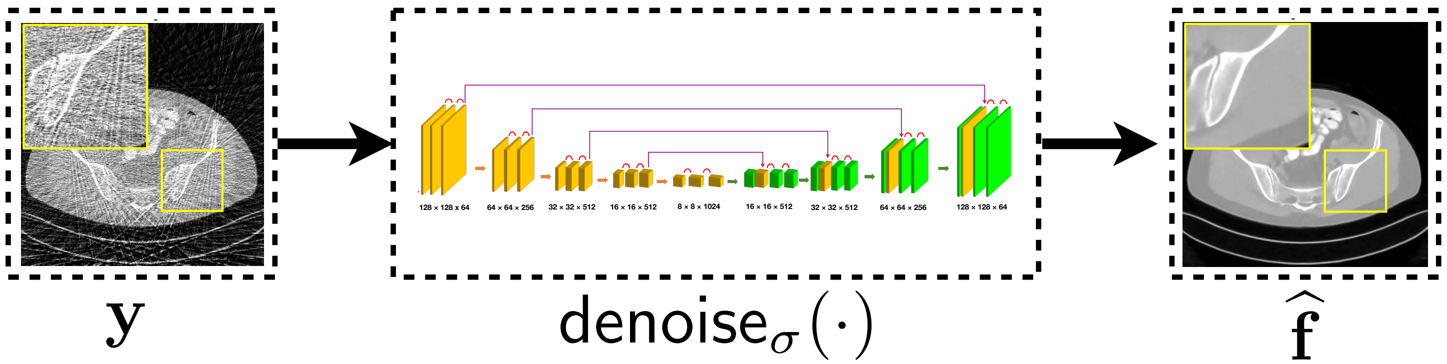


$\approx \text{prox}_{\gamma \mathcal{R}}(\mathbf{y})$

Venkatakrishnan *et al.*, “Plug-and-Play Priors for Model Based Reconstruction,” 2013.

Treating the denoising CNN as a proximal operator allows to separate the prior from the forward model

Train a CNN to denoise for various noise levels



Use the trained CNN as a Plug-and-Play Prior (PnP)

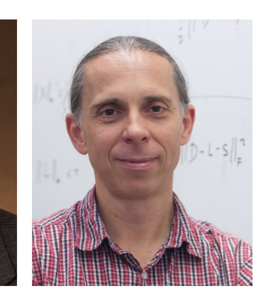
$$\mathbf{z}^k \leftarrow \mathbf{s}^{k-1} - \gamma \nabla \mathcal{D}(\mathbf{s}^{k-1})$$
$$\mathbf{f}^k \leftarrow \mathsf{denoise}_\sigma(\mathbf{z}^k)$$
$$\mathbf{s}^k \leftarrow \mathbf{f}^k + ((q_{k-1} - 1)/q_k)(\mathbf{f}^k - \mathbf{f}^{k-1})$$

PnP-FISTA

$$\mathbf{z}^k \leftarrow \mathsf{prox}_{\gamma\mathcal{D}}(\mathbf{f}^{k-1} - \mathbf{s}^{k-1})$$
$$\mathbf{f}^k \leftarrow \mathsf{denoise}_\sigma(\mathbf{z}^k + \mathbf{s}^{k-1})$$
$$\mathbf{s}^k \leftarrow \mathbf{s}^{k-1} + (\mathbf{z}^k - \mathbf{f}^k)$$

PnP-ADMM





Venkatakrishnan *et al.*, “Plug-and-Play Priors for Model Based Reconstruction,” 2013.

Plug-and-Play Priors (PnP) approach has been shown to yield state-of-the-art results

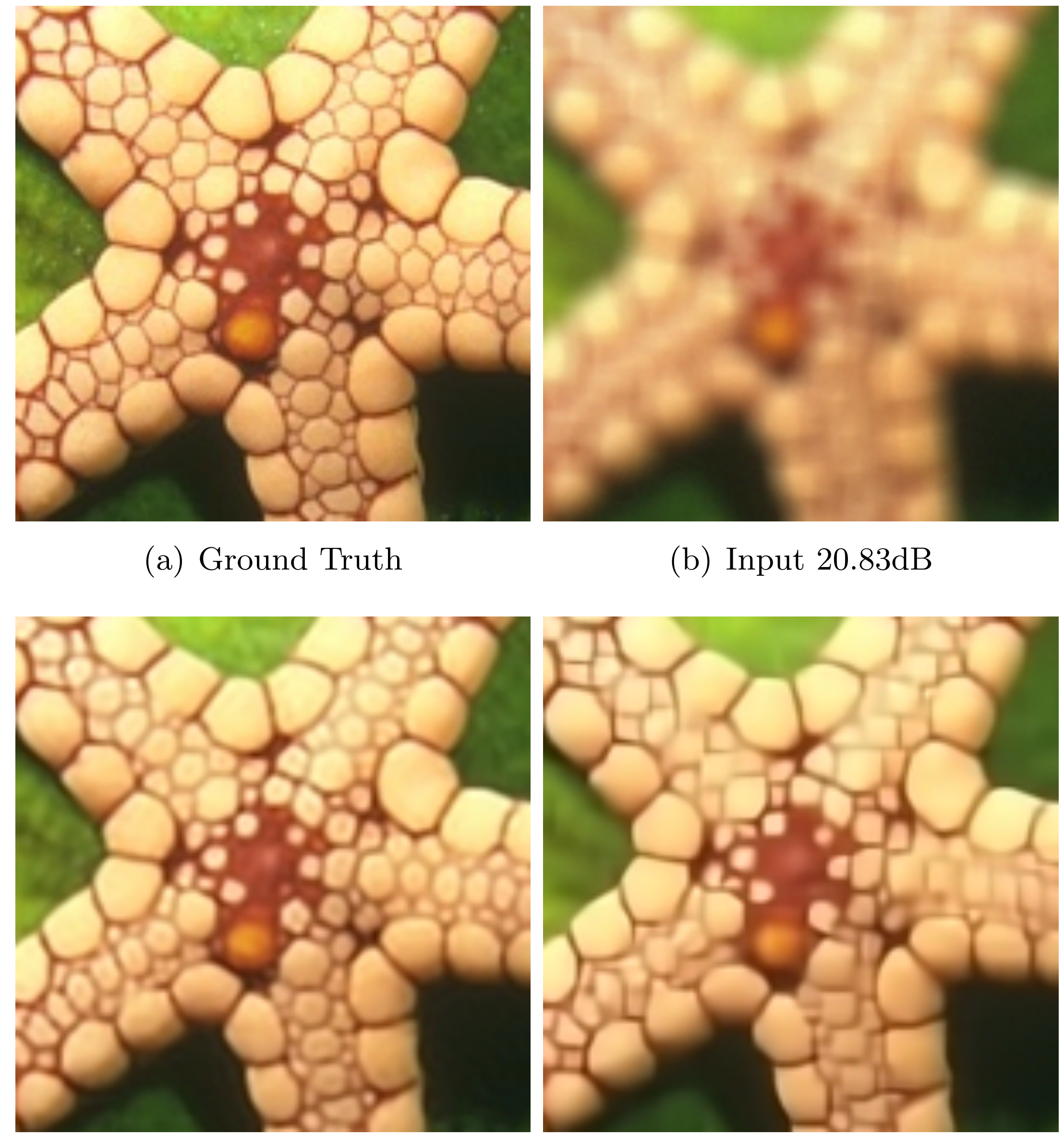
Plug-and-Play Priors (PnP) approach has been shown to yield state-of-the-art results

Method	Average PSNR (dB) over 10 images
TV	29.22
IDD-BM3D	30.92
ASDS-Reg	30.11
NCSR	31.09
PnP	**31.33**

Romano *et al.*, "The Little Engine That Could: Regularization by Denoising," 2017

Plug-and-Play Priors (PnP) approach has been shown to yield state-of-the-art results

Method	Average PSNR (dB) over 10 images
TV	29.22
IDD-BM3D	30.92
ASDS-Reg	30.11
NCSR	31.09
PnP	**31.33**



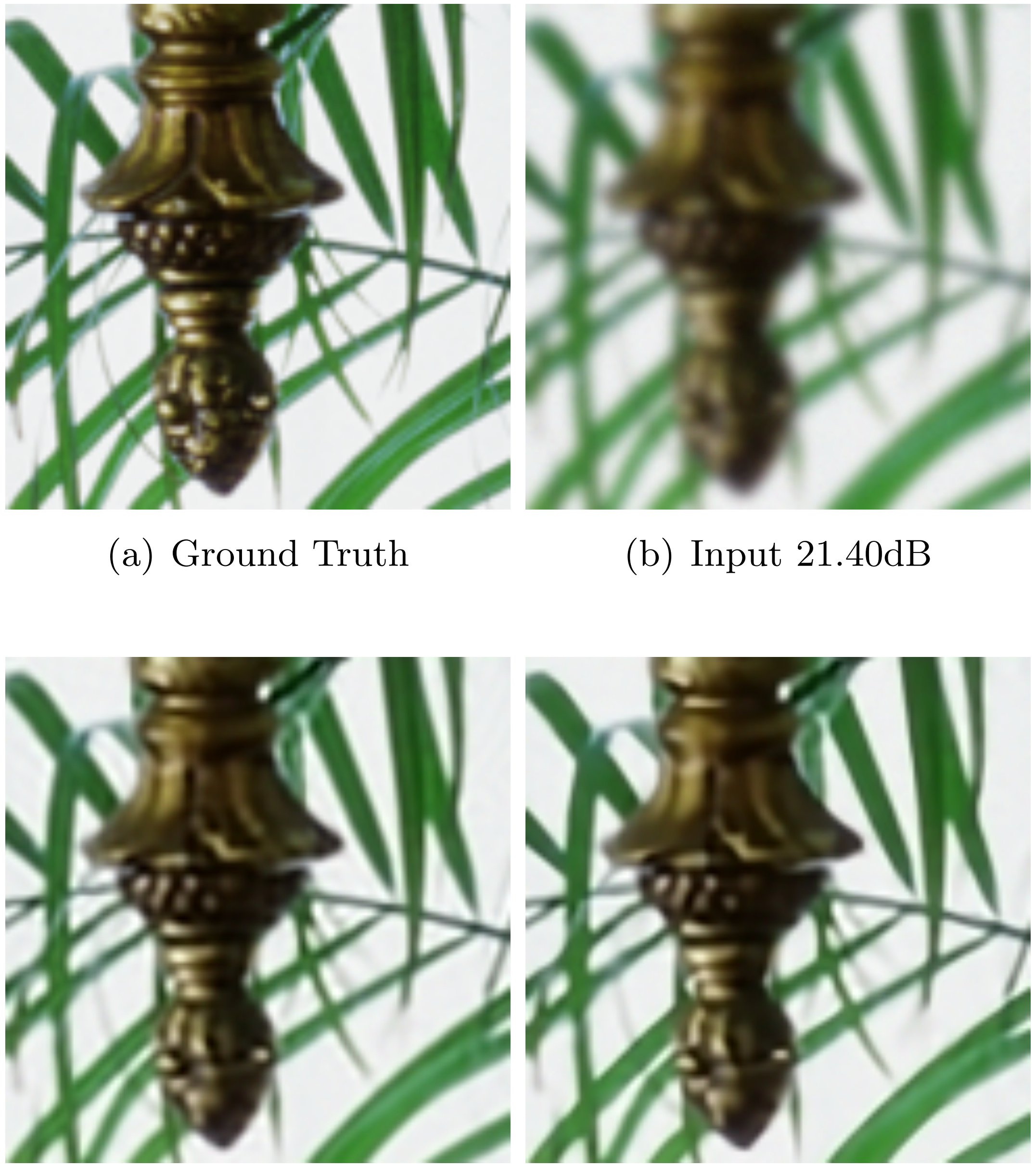
(a) Ground Truth (b) Input 20.83dB

(d) NCSR 28.39dB (e) P^3-TNRD 28.43dB

Romano *et al.*, “The Little Engine That Could: Regularization by Denoising,” 2017

Plug-and-Play Priors (PnP) approach has been shown to yield state-of-the-art results

Method	Average PSNR (dB) over 10 images
TV	29.22
IDD-BM3D	30.92
ASDS-Reg	30.11
NCSR	31.09
PnP	**31.33**



(a) Ground Truth (b) Input 21.40dB

(d) NCSR 30.03dB (e) P^3-TNRD 30.36dB

Romano *et al.*, "The Little Engine That Could: Regularization by Denoising," 2017

Can we say anything about convergence?

Can we say anything about convergence?

Sreehari *et al.*, "Plug-and-Play Priors for Bright Field Electron Tomography and Sparse Interpolation," 2016

Chan *et al.*, "Plug-and-Play ADMM for Image Restoration: Fixed-Point Convergence and Applications," 2016

Can we say anything about convergence?

Result #1: When $\mathcal{D}(\cdot)$ is convex and $\nabla \text{denoise}_\sigma(\cdot)$ is a symmetric matrix with eigenvalues in $[0, 1]$, then $\text{denoise}_\sigma(\cdot)$ is a proximal operator.

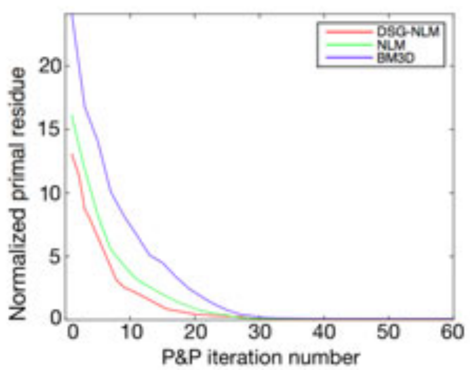
Result #2: When both $\nabla \mathcal{D}(\cdot)$ and $\text{denoise}_\sigma(\cdot)$ are bounded operators, PnP-ADMM with damping converges to a fixed point.

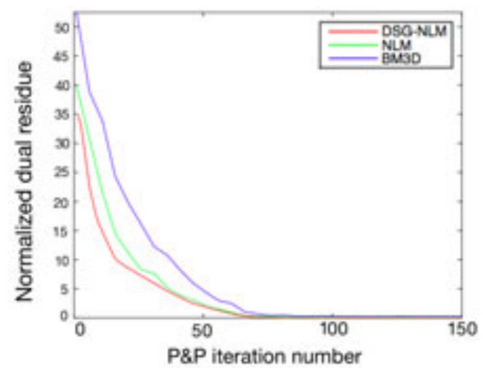
Sreehari *et al.*, "Plug-and-Play Priors for Bright Field Electron Tomography and Sparse Interpolation," 2016

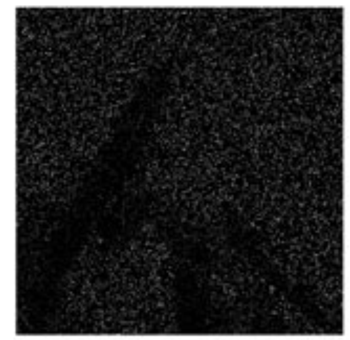
Chan *et al.*, "Plug-and-Play ADMM for Image Restoration: Fixed-Point Convergence and Applications," 2016

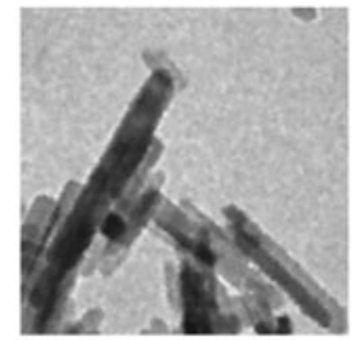
Can we say anything about convergence?

Result #1: When $\mathcal{D}(\cdot)$ is convex and $\nabla \text{denoise}_\sigma(\cdot)$ is a symmetric matrix with eigenvalues in $[0, 1]$, then $\text{denoise}_\sigma(\cdot)$ is a proximal operator.









Result #2: When both $\nabla \mathcal{D}(\cdot)$ and $\text{denoise}_\sigma(\cdot)$ are bounded operators, PnP-ADMM with damping converges to a fixed point.

DCNN [9]	20.72	21.30	18.91	21.68	16.10	23.39	22.33	22.99	22.46	20.23	21.01
SR [12]	20.67	21.30	18.86	21.51	16.37	23.15	22.19	22.85	22.26	20.33	20.95
SPSR [10]	20.85	21.58	19.18	21.85	16.59	23.52	22.42	23.05	22.53	20.50	21.21
TSE [52]	20.59	21.24	18.80	21.49	16.40	23.14	22.21	22.78	22.21	20.30	20.92
GPR [11]	21.55	22.68	19.90	22.77	17.70	24.57	23.51	24.37	23.63	21.35	22.20
Ours - M	**23.62**	**25.75**	**23.06**	**25.30**	**24.48**	**27.17**	**29.14**	**29.42**	**26.86**	**26.86**	**26.17**


DCNN


PnP-ADMM

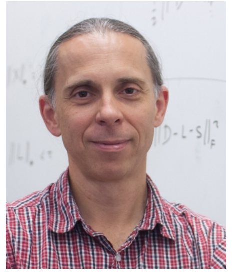
Sreehari *et al.*, "Plug-and-Play Priors for Bright Field Electron Tomography and Sparse Interpolation," 2016

Chan *et al.*, "Plug-and-Play ADMM for Image Restoration: Fixed-Point Convergence and Applications," 2016

Can we say anything about convergence?

Can we say anything about convergence?

Sun, Wohlberg, Kamilov, “An Online Plug-and-Play Algorithm for Regularized Image Reconstruction,” 2018


Can we say anything about convergence?

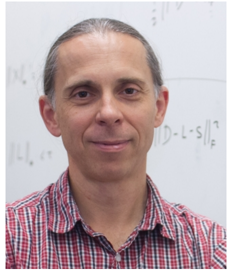
Useful definitions

$$\mathsf{P}(\mathbf{f}) \triangleq \mathsf{denoise}_\sigma(\mathbf{f} - \gamma \nabla \mathcal{D}(\mathbf{f}))$$

gradient-denoiser operator

$$\mathsf{fix}(\mathsf{P}) \triangleq \{\mathbf{f} \in \mathbb{R}^n : \mathbf{f} = \mathsf{P}(\mathbf{f})\}$$

its of fixed points


Sun, Wohlberg, Kamilov, "An Online Plug-and-Play Algorithm for Regularized Image Reconstruction," 2018

Washington University in St. Louis

Can we say anything about convergence?

Useful definitions

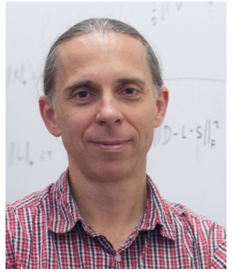
$$\mathsf{P}(\mathbf{f}) \triangleq \mathsf{denoise}_\sigma(\mathbf{f} - \gamma \nabla \mathcal{D}(\mathbf{f})) \qquad \mathsf{fix}(\mathsf{P}) \triangleq \{\mathbf{f} \in \mathbb{R}^n : \mathbf{f} = \mathsf{P}(\mathbf{f})\}$$

#1: Let $\mathsf{denoise}_\sigma(\cdot) = \mathsf{prox}_{\gamma\mathcal{R}}(\cdot)$. Then, $\mathbf{f}^* \in \mathsf{fix}(\mathsf{P})$ iff it minimizes $\mathcal{C} = \mathcal{D} + \mathcal{R}$

#2: Run PnP-ISTA with a nonexpansive denoiser for $t \geq 1$ iterations. Then

$$\min_{k \in \{1,\dots,t\}} \left\{ \|\mathbf{f}^{k-1} - \mathsf{P}(\mathbf{f}^{k-1})\|_{\ell_2}^2 \right\} = O(1/t)$$

#3: For nonexpansive denoisers, fixed points of PnP-ADMM coincide with fix(P)


Sun, Wohlberg, Kamilov, "An Online Plug-and-Play Algorithm for Regularized Image Reconstruction," 2018

PnP-SGD is an online extension useful when dealing with a large number of measurements

PnP-SGD is an online extension useful when dealing with a large number of measurements

Consider the following data-fidelity term

$$\mathcal{D}(\mathbf{f}) = \frac{1}{2I}\sum_{i=1}^{I} \|\mathbf{y}_i - \mathbf{H}_i\mathbf{f}\|_{\ell_2}^2 \quad \Rightarrow \quad \nabla\mathcal{D}(\mathbf{f}) = \frac{1}{I}\sum_{i=1}^{I} \mathbf{H}_i^{\mathsf{T}}(\mathbf{H}_i\mathbf{f} - \mathbf{y})$$

cost of computing the gradient is liner in the number of measurements

Sun, Wohlberg, Kamilov, "An Online Plug-and-Play Algorithm for Regularized Image Reconstruction," 2018

PnP-SGD is an online extension useful when dealing with a large number of measurements

Consider the following data-fidelity term

$$\mathcal{D}(\mathbf{f}) = \frac{1}{2I}\sum_{i=1}^{I} \|\mathbf{y}_i - \mathbf{H}_i\mathbf{f}\|_{\ell_2}^2 \quad \Rightarrow \quad \nabla\mathcal{D}(\mathbf{f}) = \frac{1}{I}\sum_{i=1}^{I} \mathbf{H}_i^{\mathsf{T}}(\mathbf{H}_i\mathbf{f} - \mathbf{y})$$

PnP-SGD can accelerate imaging by parallelizing the processing of each data item

$$\hat{\nabla}\mathcal{D}(\mathbf{s}^{k-1}) \leftarrow \mathsf{minibatchGradient}(\mathbf{s}^{k-1}, B)$$
$$\mathbf{z}^k \leftarrow \mathbf{s}^{k-1} - \gamma\hat{\nabla}\mathcal{D}(\mathbf{s}^{k-1})$$
$$\mathbf{f}^k \leftarrow \mathsf{denoise}_\sigma(\mathbf{z}^k)$$
$$\mathbf{s}^k \leftarrow \mathbf{f}^k + ((q_{k-1} - 1)/q_k)(\mathbf{f}^k - \mathbf{f}^{k-1})$$

use only B measurements per iteration instead of I

Sun, Wohlberg, Kamilov, "An Online Plug-and-Play Algorithm for Regularized Image Reconstruction," 2018

PnP-SGD is an online extension useful when dealing with a large number of measurements

Consider the following data-fidelity term

$$\mathcal{D}(\mathbf{f}) = \frac{1}{2I}\sum_{i=1}^{I}\|\mathbf{y}_i - \mathbf{H}_i\mathbf{f}\|_{\ell_2}^2 \quad \Rightarrow \quad \nabla\mathcal{D}(\mathbf{f}) = \frac{1}{I}\sum_{i=1}^{I}\mathbf{H}_i^{\mathsf{T}}(\mathbf{H}_i\mathbf{f} - \mathbf{y})$$

PnP-SGD can accelerate imaging by parallelizing the processing of each data item

$$\hat{\nabla}\mathcal{D}(\mathbf{s}^{k-1}) \leftarrow \mathsf{minibatchGradient}(\mathbf{s}^{k-1}, B)$$
$$\mathbf{z}^k \leftarrow \mathbf{s}^{k-1} - \gamma\hat{\nabla}\mathcal{D}(\mathbf{s}^{k-1})$$
$$\mathbf{f}^k \leftarrow \mathsf{denoise}_\sigma(\mathbf{z}^k)$$
$$\mathbf{s}^k \leftarrow \mathbf{f}^k + ((q_{k-1} - 1)/q_k)(\mathbf{f}^k - \mathbf{f}^{k-1})$$

Sun, Wohlberg, Kamilov, "An Online Plug-and-Play Algorithm for Regularized Image Reconstruction," 2018

PnP-SGD converges to the same set of fixed points as batch PnP algorithms

Sun, Wohlberg, Kamilov, "An Online Plug-and-Play Algorithm for Regularized Image Reconstruction," 2018

PnP-SGD converges to the same set of fixed points as batch PnP algorithms

#4: Run PnP-SGD for $t \geq 1$ iterations under some mild assumptions. Then

$$\mathbb{E}\left[\min_{k\in\{1,\dots,t\}}\left\{\|\mathbf{f}^{k-1} - \mathsf{P}(\mathbf{f}^{k-1})\|_{\ell_2}^2\right\}\right] \leq C\left[\frac{\gamma^2\nu^2}{B} + \frac{2\gamma\nu}{\sqrt{B}}\|\mathbf{f}^0 - \mathbf{f}^*\|_{\ell_2} + \frac{\|\mathbf{f}^0 - \mathbf{f}^*\|_{\ell_2}^2}{t}\right]$$

Convergence in expectation. C is a constant. Note the case when B = t

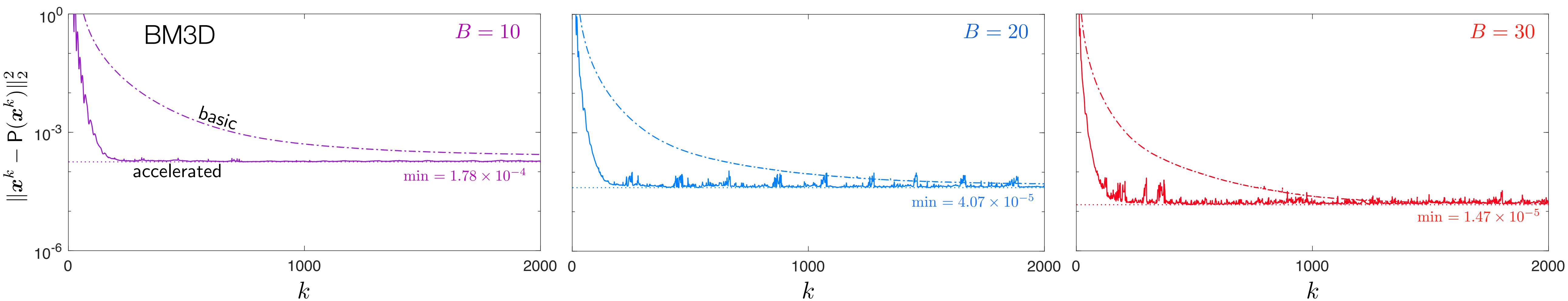
Sun, Wohlberg, Kamilov, "An Online Plug-and-Play Algorithm for Regularized Image Reconstruction," 2018

PnP-SGD converges to the same set of fixed points as batch PnP algorithms

Washington University in St. Louis

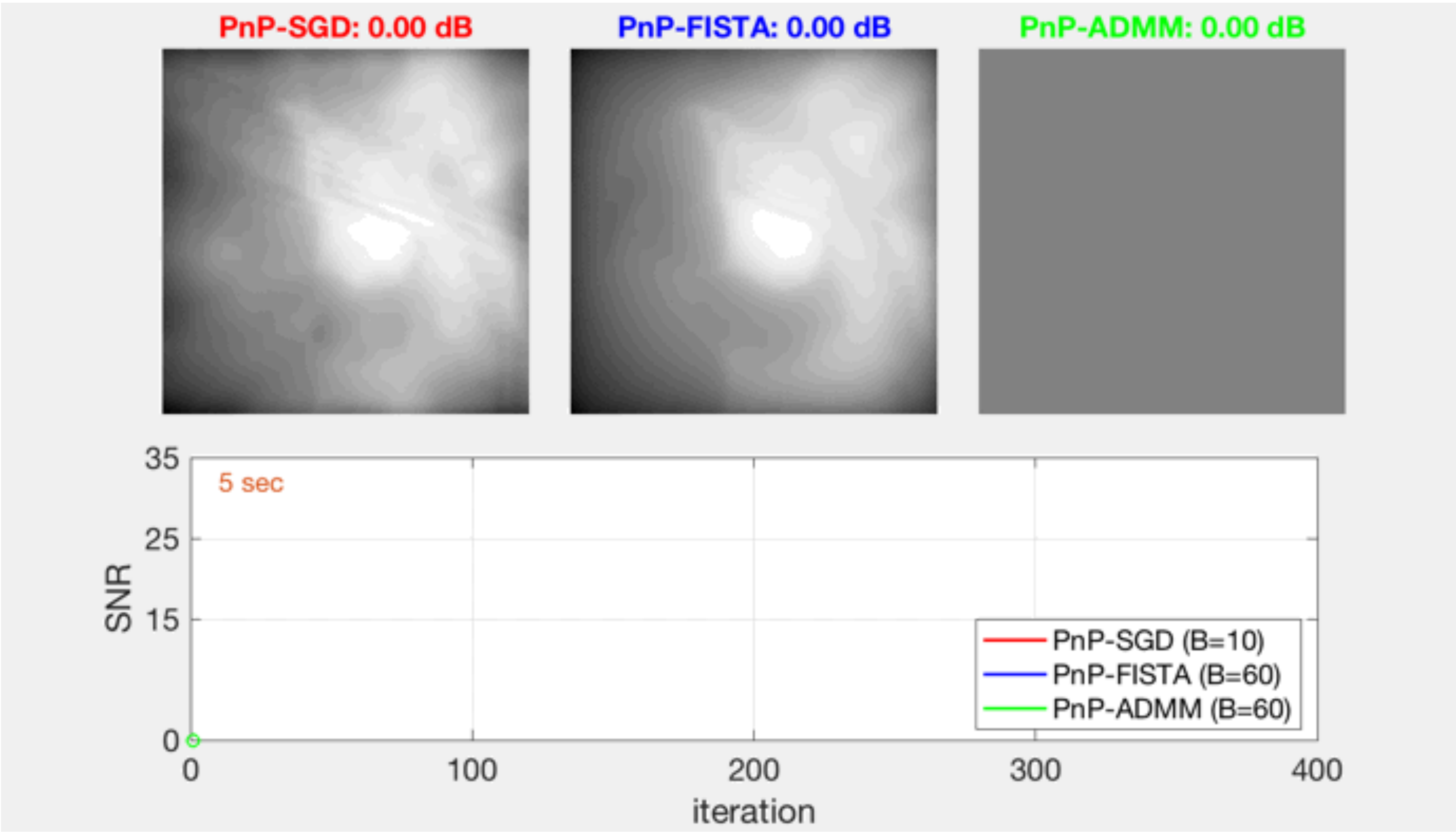
#4: Run PnP-SGD for $t \geq 1$ iterations under some mild assumptions. Then

$$\mathbb{E}\left[\min_{k\in\{1,\dots,t\}}\left\{\|\mathbf{f}^{k-1} - \mathsf{P}(\mathbf{f}^{k-1})\|_{\ell_2}^2\right\}\right] \leq C\left[\frac{\gamma^2\nu^2}{B} + \frac{2\gamma\nu}{\sqrt{B}}\|\mathbf{f}^0 - \mathbf{f}^*\|_{\ell_2} + \frac{\|\mathbf{f}^0 - \mathbf{f}^*\|_{\ell_2}^2}{t}\right]$$



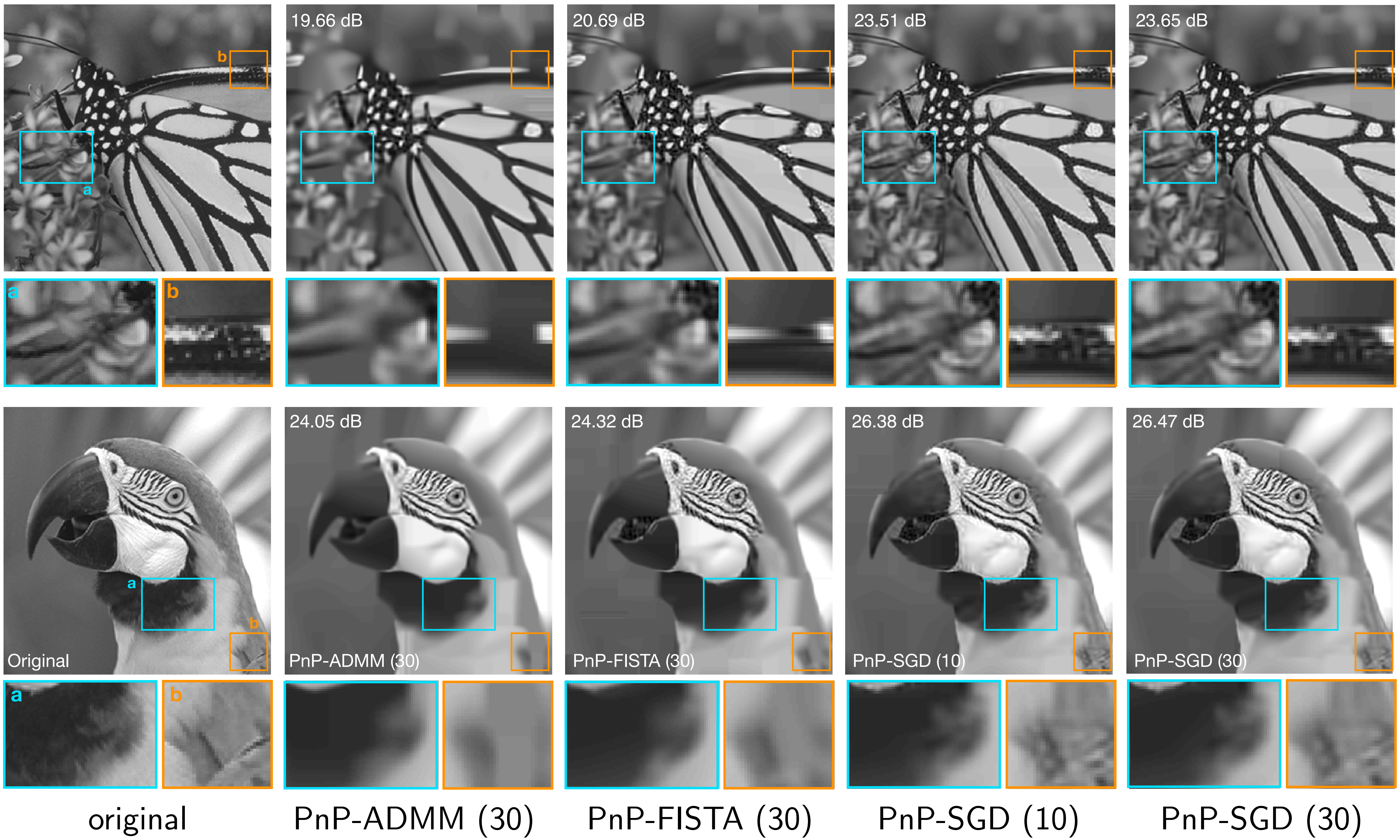
Sun, Wohlberg, Kamilov, "An Online Plug-and-Play Algorithm for Regularized Image Reconstruction," 2018

For many measurements PnP-SGD converges faster than batch algorithms



Sun, Wohlberg, Kamilov, “An Online Plug-and-Play Algorithm for Regularized Image Reconstruction,” 2018

For the same measurement budget, PnP-SGD gets much higher quality results



Sun, Wohlberg, Kamilov, "An Online Plug-and-Play Algorithm for Regularized Image Reconstruction," 2018

Conclusion

Image reconstruction is a fascinating research area that brings together physics, signal processing, nonlinear optimization, and machine learning

We are increasingly reliant on implicit regularization using nonlinear operators, such as deep neural networks or nonlinear filters

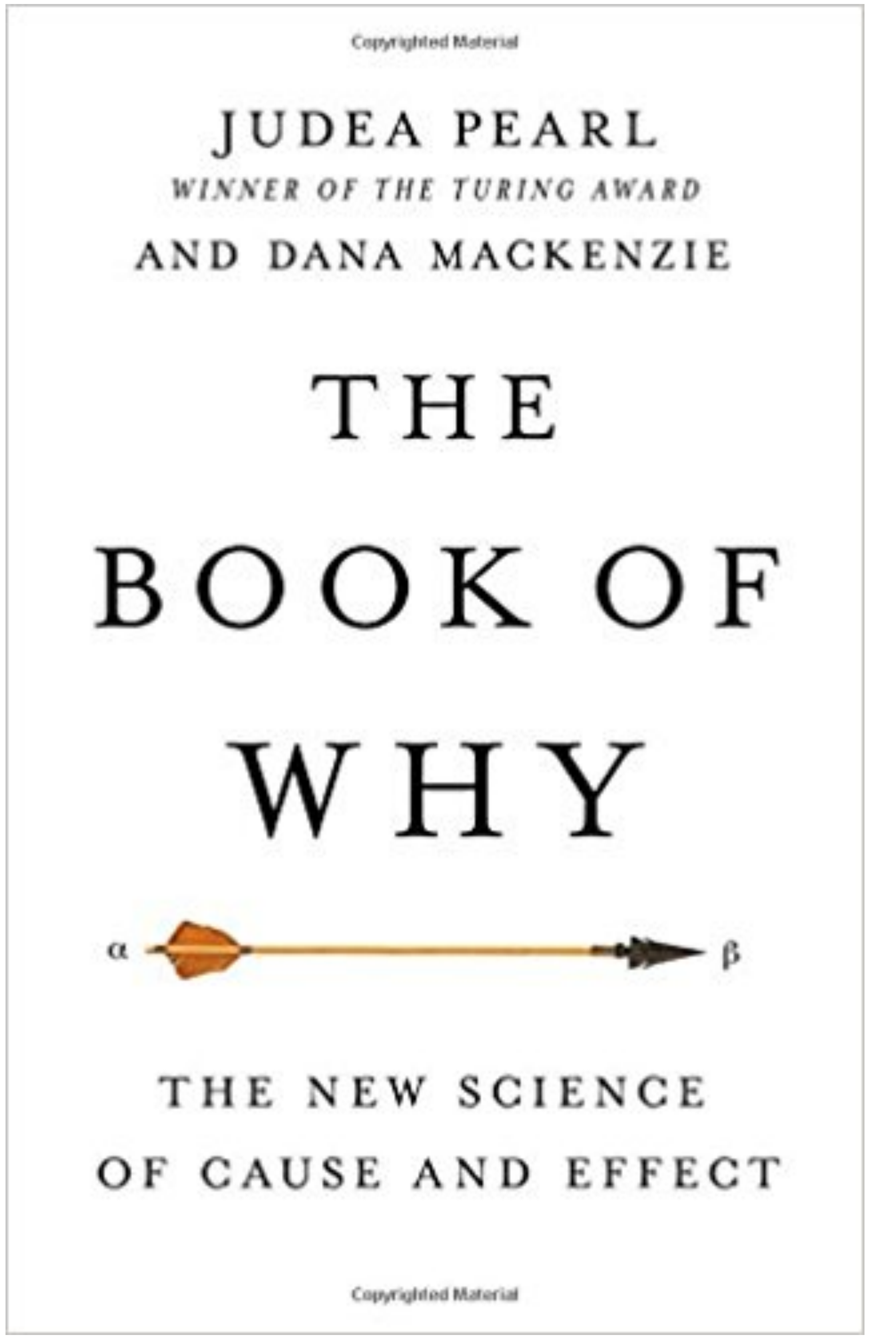
Plug-In SGD is a theoretically sound algorithm that can regularize at large-scales using nonlinear operators

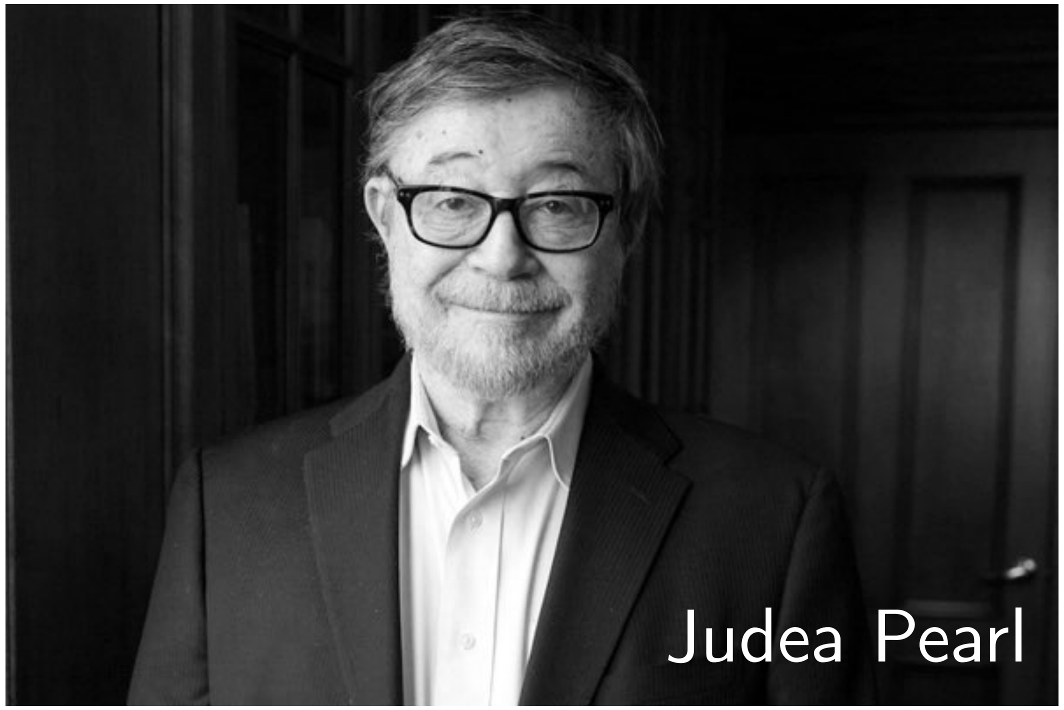
CONTACT INFO

Prof. Ulugbek Kamilov
Computational Imaging Group (CIG)
Washington University in St. Louis
Email: kamilov@wustl.edu
Web: http://cigroup.wustl.edu
Twitter: @wustlcig

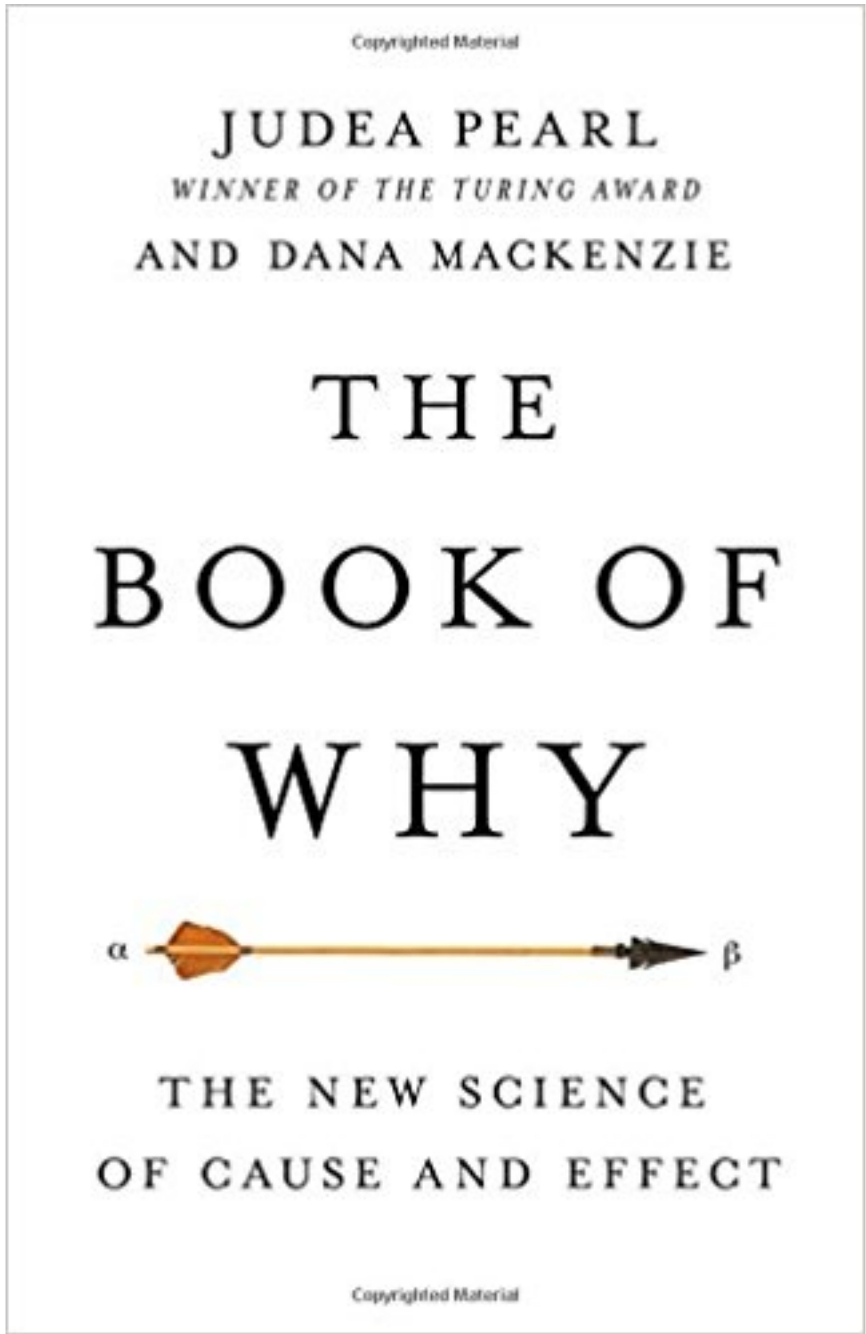
Judea Pearl won the Turing Award in 2011 for fundamental contributions to artificial intelligence

Judea Pearl won the Turing Award in 2011 for fundamental contributions to artificial intelligence



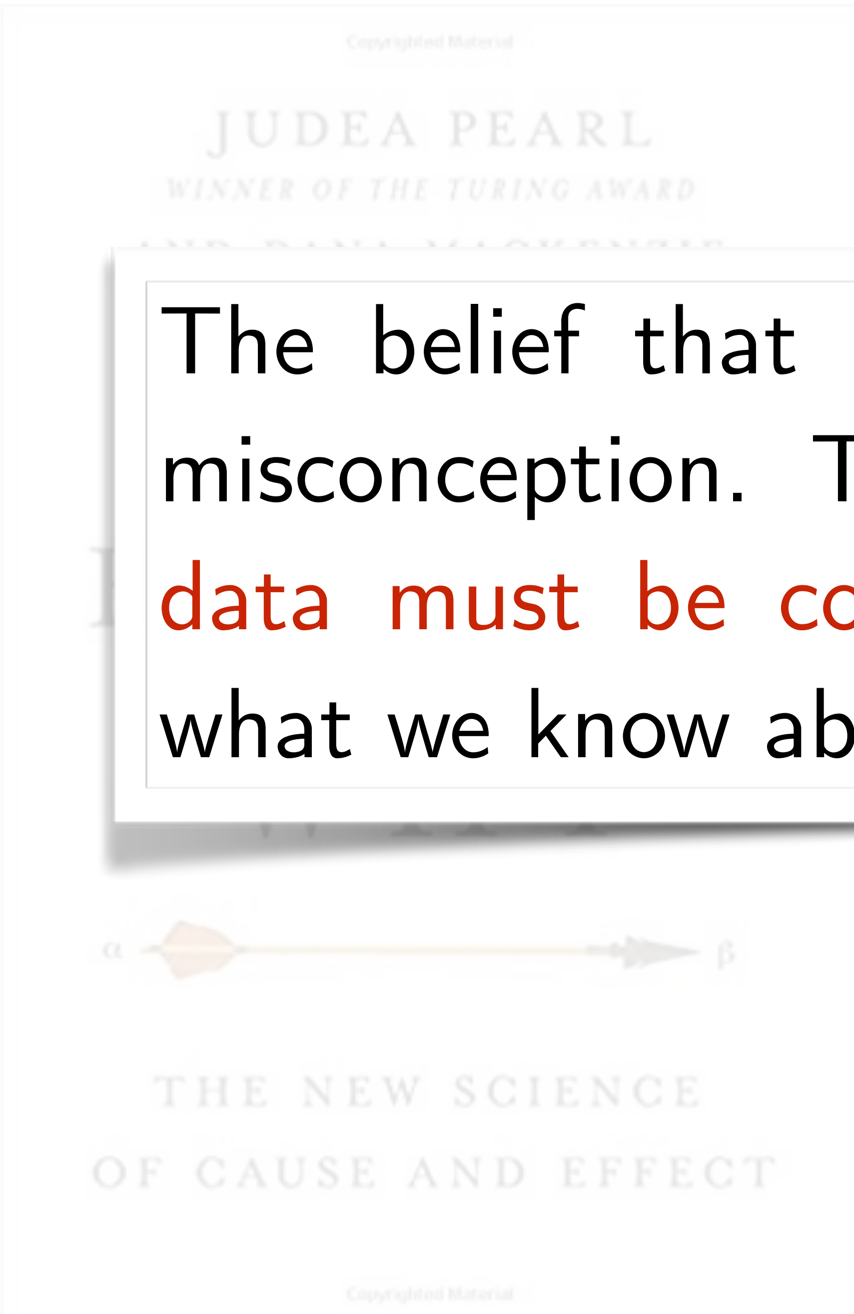


Judea Pearl won the Turing Award in 2011 for fundamental contributions to artificial intelligence



We live in an era that presumes Big Data to be the solution to all our problems (...) But I hope with this book to convince you that data are profoundly dumb. Data can tell you that the people who took a medicine recovered faster than those who did not take it, but they can't tell you why.

Judea Pearl won the Turing Award in 2011 for fundamental contributions to artificial intelligence



We live in an era that presumes Big Data to be the solution to all our problems. Courses in data science

> The belief that data can tell the full story is a misconception. To produce truly useful insights, **data must be combined with models** that infuse what we know about the problem.

Judea Pearl