

ECE 60141 / BME 66100
Foundations of Computational Imaging
Lecture 1: What is Computational Imaging?
Chapter 1

Charles A. Bouman

Purdue University

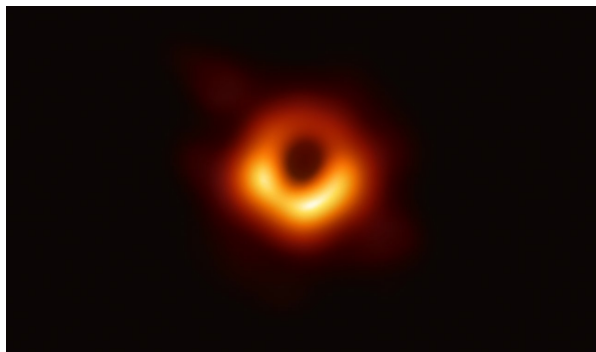
Monday, August 24, 2026

Today's Big Ideas

- Computational imaging forms images by computation, because practical sensors almost never produce a usable image on their own.
- Recovering the unknown image from the measurements is an inverse problem.
- Model-based methods solve that inverse problem with a forward model that simulates the sensor and a prior model that describes real images.
- A prior model is necessary rather than optional, because every algorithm assumes something and real images fill only a thin part of the space of all pixel arrays.
- Real images are 2-D, noncausal, and non-Gaussian, so the course starts with the simplest 1-D causal Gaussian models and builds up from there.

Reading: Chapter 1, *Foundations of Computational Imaging*.

April 10, 2019: the first image of a black hole



Credit: Event Horizon Telescope Collaboration

- This image was not taken by a camera — it was *computed*. So what exactly did the telescope measure?
- The **Event Horizon Telescope** is an Earth-sized array of radio telescopes, aimed at the supermassive black hole in Messier 87. It measured radio-frequency data — and an algorithm turned that data into the picture above.

What is computational imaging?

- Traditional imaging: optics form the image; the sensor records something that already looks like a picture.
- Computational imaging: the sensor records *data* — which may look nothing like a picture at all.
- So where does the image come from?

What is computational imaging?

- Traditional imaging: optics form the image; the sensor records something that already looks like a picture.
- Computational imaging: the sensor records *data* — which may look nothing like a picture at all.
- So where does the image come from?

The image you see is the solution to an *inverse problem*.

CT, MRI, PET, electron microscopy, synthetic aperture radar, computational photography — and your cellphone camera — all work this way.

The core idea of this course

Model the sensor (the forward model), model the image (the prior), and compute the reconstruction by optimization or sampling.

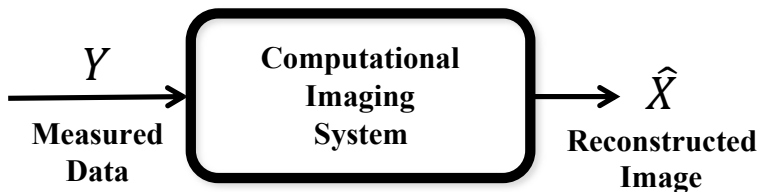
A promise for the semester



- Same patient, same scan, same GE 64-slice CT scanner: direct reconstruction (left) vs. model-based iterative reconstruction (right).
- Which image would you rather have your radiologist read?
- The right image did not come from a better scanner. It came from better *mathematics* — finer detail *and* less noise, from the same data.

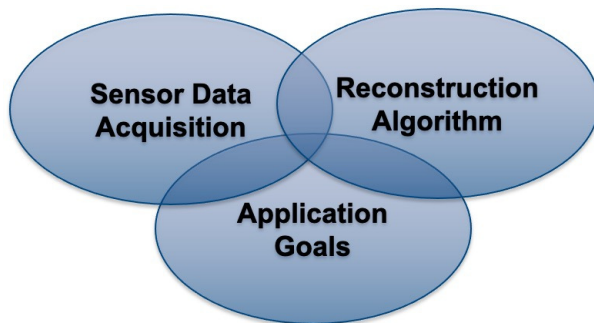
By the end of this course, you will understand exactly why the image on the right is possible.

Sensors do not make images — algorithms do



- Practical sensors almost never directly generate usable images.
- A cellphone CMOS sensor needs color dematrixing, deblurring, denoising, and registration before you see a picture.
- In CT or SAR, the raw data is not even recognizable as an image.

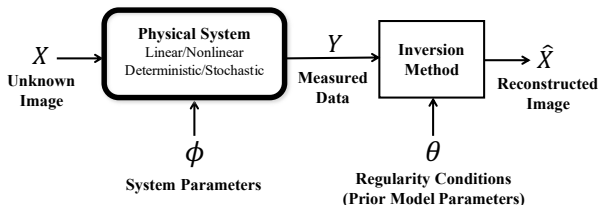
Co-design of sensor and algorithm



- Suppose perfect measurements cost \$10 million, and imperfect measurements plus a good algorithm cost \$100,000. Which system should we build?
- Design the sensor and the reconstruction algorithm *together* to meet the application goals at minimum cost. The goal is the most *informative* measurements, not *perfect* measurements.
- **Mick Jagger's theorem** states that "you can't always get the sensor data you want, but if you try sometimes, you get what you need."

The inverse problem

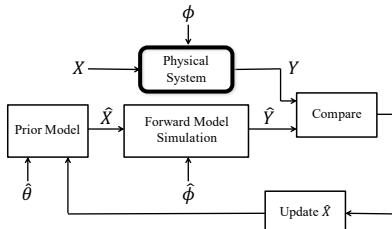
- There is something we want to know: an image X .
- Nature hands us measurements instead: Y .
- The **inverse problem** is to compute an estimate $\hat{X}$ of the image X from the measurements Y .



- Two more players: ϕ (system calibration) and θ (amount of regularization) must often be estimated too.
- Microscopes, cellphone cameras, radio telescopes, PET scanners — all share this same structure.

The model-based approach

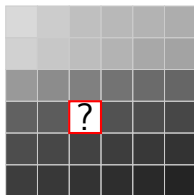
If we guessed an image, how could we tell whether the guess was right?



- The **forward model** $p(y|x)$ simulates the sensor. It predicts the measurements the guess would produce — geometry, sensitivity, noise.
- Compare the prediction with the real measurements; update the guess; repeat.
- The **prior model** $p(x)$ keeps every guess looking like a real image.

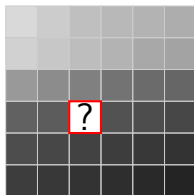
We compute the inverse by modeling the *forward* process — so it works even when no analytical inverse exists.

A thought experiment



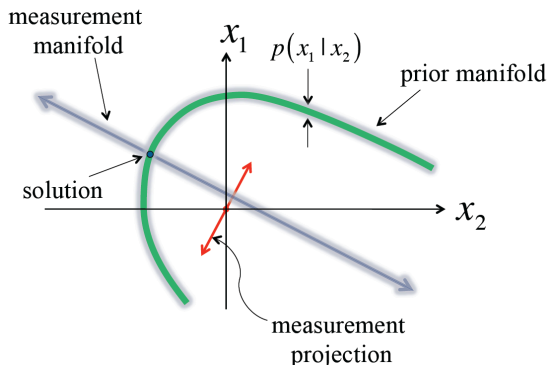
- One pixel of this image is covered. Can you guess roughly what belongs there?

A thought experiment



- One pixel of this image is covered. Can you guess roughly what belongs there?
- Of course you can. Why?
- Because almost no arrays of random pixel values are pictures. Real images have structure — each pixel is predictable from its neighbors.

Why priors work: images lie on a thin manifold



- A **thin manifold** is the small, curved subset of $[0, 1]^N$ that real images occupy. Predictability from neighbors is what makes it thin.
- Measurements $Y = AX + W$ constrain the solution to another thin manifold.
- Even with fewer measurements than unknowns ($M < N$), a good solution exists at the *intersection* of the two.

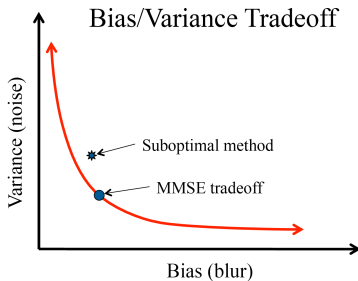
“But aren't priors cheating?”

Shouldn't we just reconstruct what the data say — no assumptions?

“But aren't priors cheating?”

Shouldn't we just reconstruct what the data say — no assumptions?

There is no such thing as “no assumptions.”



- Every inversion algorithm makes assumptions — a resolution limit, a reconstruction filter.
- Zero bias often means very large (even infinite) variance. Variance looks like noise; bias looks like blur or artifacts. You saw both in the CT pair earlier.
- So the goal is not “no assumptions” — it is the most *accurate* prior model, which introduces the least harmful bias.

Roadmap: from simple models to real images

- Real images are best modeled as **2-D, noncausal, and non-Gaussian** — adjacent-pixel differences are Laplacian, not Gaussian. That is the hardest box in this table:

	1-D	2-D
Gaussian	causal AR	GMRF
Non-Gaussian	Markov process	MRF
Discrete	Markov chain	discrete MRF

- We are not going to jump straight into the hardest box. We start in the simplest one — 1-D, causal, Gaussian — and build up, one step at a time.
- So for the first few weeks, some examples will not look much like CT scans or black holes. This table is why.
- Noncausal models in 2-D have loops, which break the standard 1-D tools — that is why Markov random fields are a central theme.

Where we are headed



- Everything we do this semester is about one thing: how measured data become images like these.
- Logistics, grading, the labs, and this week's to-do list are in **Lecture 0** on the course page — read it this week.

Welcome to the course!

Summary

- Computational imaging computes the image $\hat{X}$ from raw sensor data Y , because the sensor output is not itself a usable image.
- The inverse problem is to estimate the unknown image X from the measurements Y , together with the nuisance parameters ϕ and θ .
- Model-based methods combine the forward model $p(y|x)$ with the prior model $p(x)$, and they return the $\hat{X}$ that balances the two.
- Real images lie on a thin manifold in $[0, 1]^N$, so measurements $Y = AX + W$ can determine a good solution even when $M < N$.
- Real images are 2-D, noncausal, and non-Gaussian, so the course starts with 1-D causal Gaussian models and works toward Markov random fields.

Reading: Chapter 1 **Next lecture:** Chapter 2.1, Random variables and expectation.