Ivo Ihrke

-- INVERSE PROBLEMS --

Inverse Problems

Tomography

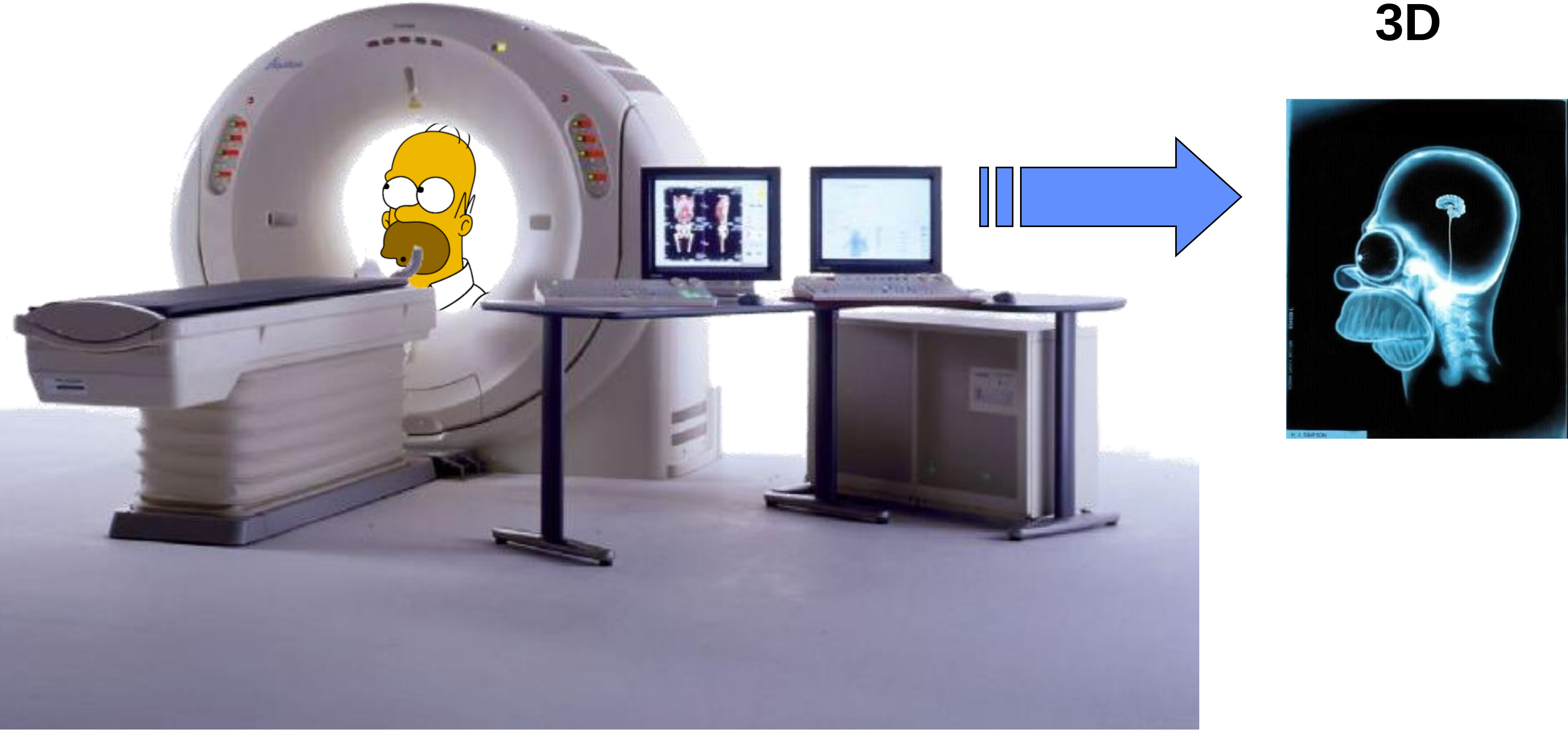
Overview

- Tomography
  - Absorption / emission
  - Fourier Slice Theorem and Filtered Back Projection
  - Algebraic Reconstruction
  - Applications

Outline

- Computed Tomography (CT)
  - Radon transform
  - Filtered Back-Projection
  - natural phenomena
  - glass objects

Computed Tomography (CT)



Some History

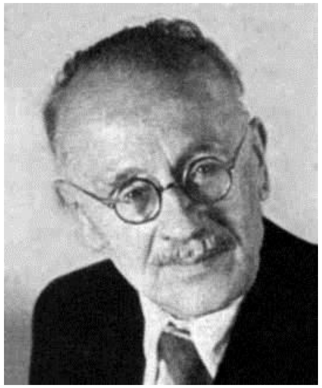
- Radon transform (1917)

$$\mathcal{R}\{f\}(\alpha, s) = \int_{c_{\alpha,s}} f \circ c_{\alpha,s}(t)dt$$

  - Radon: Inverse transform exists

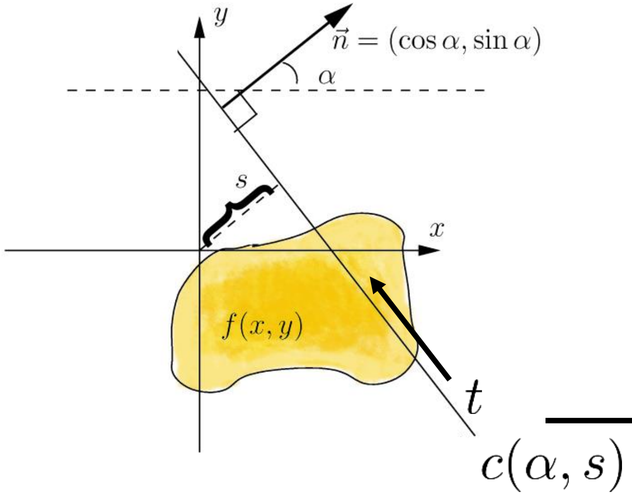
    if all (α, s) are covered
  - First numerical application

    Viktor Ambartsumian (1936, astrophysics)



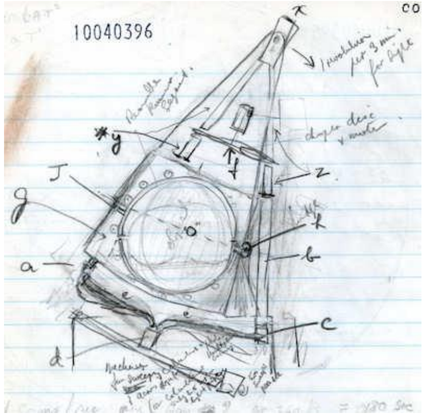
Johann Radon (1887-1956)

Viktor Ambartsumian (1909-1996)



Some History

- CT Scanning

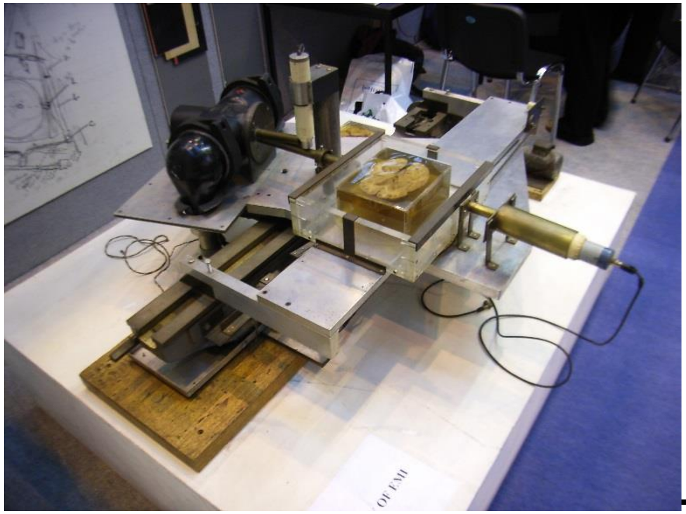


Sketch of the invention

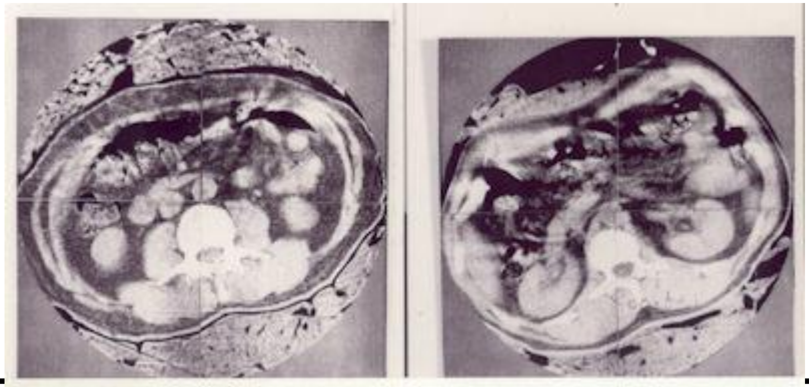
Godfrey Hounsfield (1919-2004)

Allan Cormack (1924-1998)

- 1979 Nobel prize in Physiology or Medicine



Prototype scanner

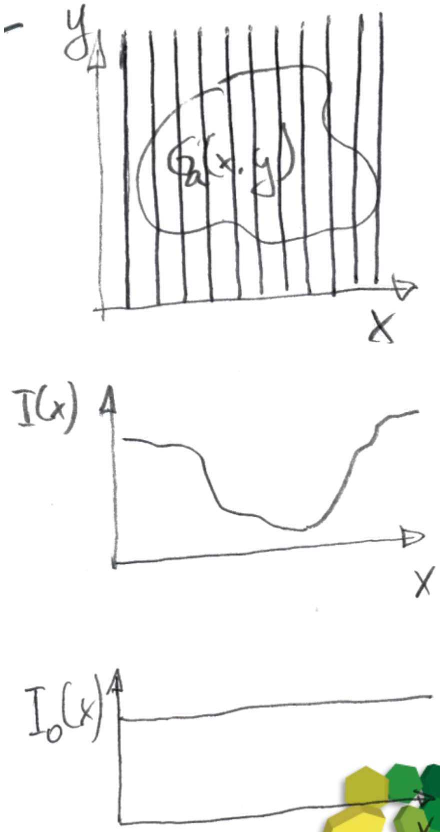


Hounsfield's abdomen

The math

- X-rays are attenuated by body tissue and bones
  - Attenuation is spatially variant (attenuation coeff. $\sigma_a(x,y)$)

$$\begin{aligned} I(x) &= I_0(x)e^{-\int_c \sigma_a(x,y)dy} \\ \Rightarrow \quad \frac{I(x)}{I_0(x)} &= e^{-\int_c \sigma_a(x,y)dy} \\ \Rightarrow \quad \log\frac{I(x)}{I_0(x)} &= -\int_c \sigma_a(x,y)dy \end{aligned}$$



  - $I(x), I_0(x)$ are known, determine $\sigma_a(x,y)$
  - Ill-posed for only one direction α
    - Need all

Well-Posed and Ill-Posed Problems

- Definition [Hadamard1902]
  - a problem is well-posed if
    1. a solution exists
    2. the solution is unique
    3. the solution continually depends on the data
  - a problem is ill-posed if it is not well-posed

Inverse Problems

Tomography

-- Fourier-Based Techniques --

Computed Tomography

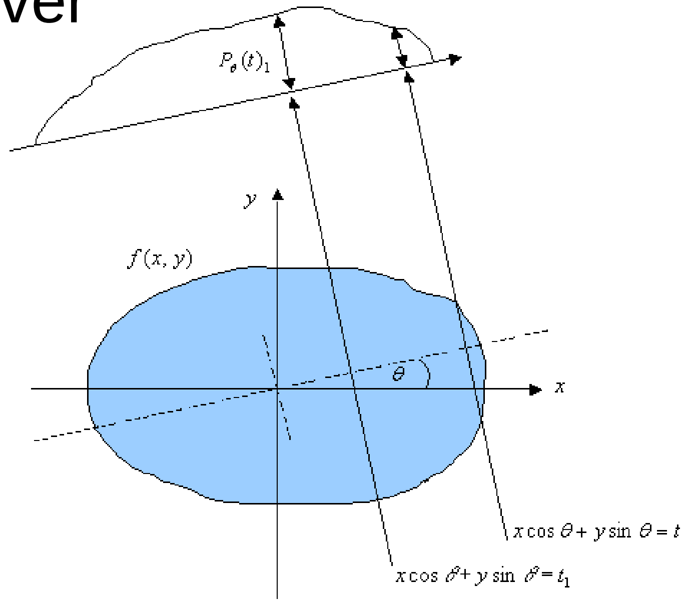
- tomography is the problem of computing a function from its projections
- a projection is a set of line integrals over function m along some ray c

$$o = \int_c m(c(s))ds$$

- invert this equation (noise is present)

$$o = \int_c m(c(s))ds + n$$

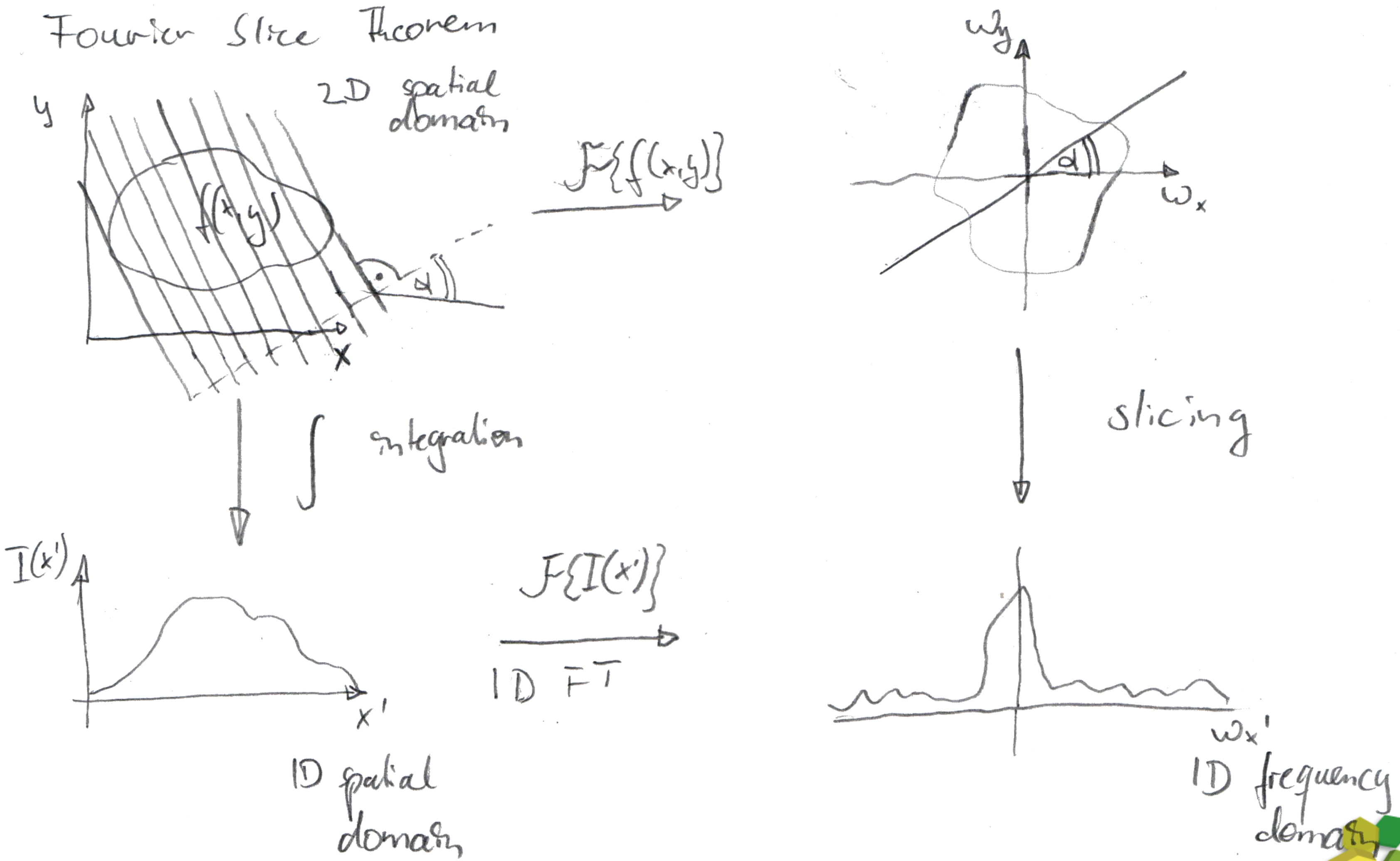
- if infinitely many projections are available this is possible (Radon transform) [Radon1917]


$P_\theta(t)_1$
y
$f(x, y)$
θ
x
$x\cos\theta + y\sin\theta = t$
$x\cos\theta + y\sin\theta = t_1$

Computed Tomography – Frequency Space Approach

- Fourier Slice Theorem
- The Fourier transform of an orthogonal projection is a slice of the Fourier transform of the function

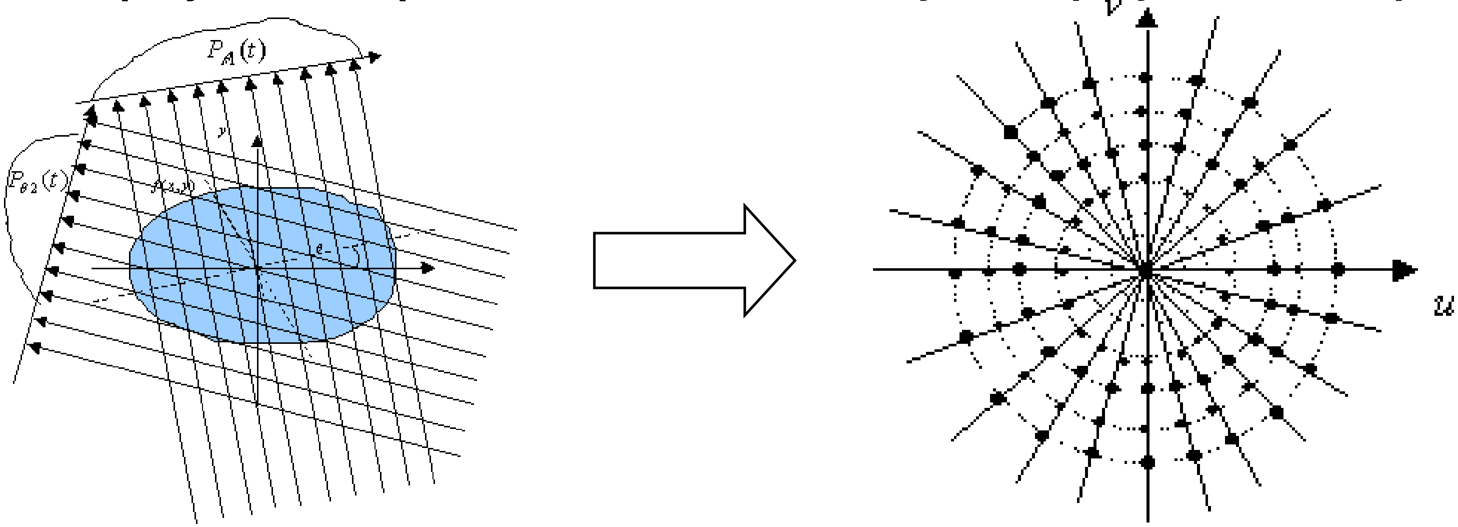
Computed Tomography – FST



Computed Tomography – Frequency Space Approach

- for recovery of the 2D function we need several slices

several projections, spatial domain **many more projections, frequency domain**



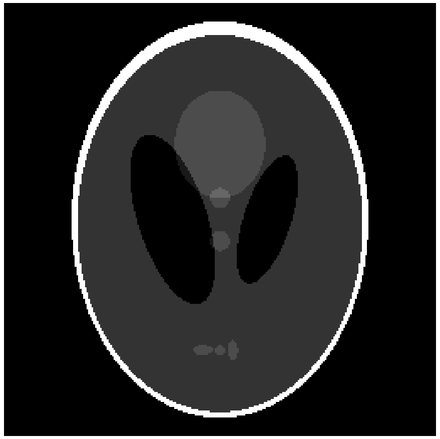
- slices are usually interpolated onto a rectangular grid
- inverse Fourier transform
- gaps for high frequency components

  → artifacts

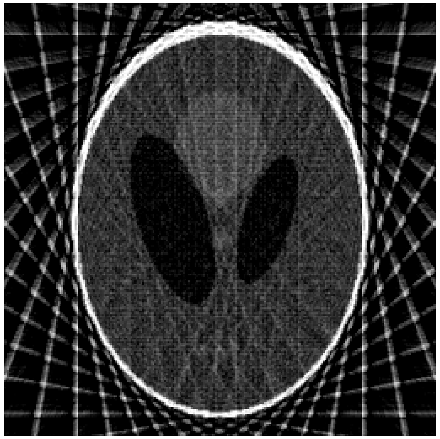
Frequency Space Approach - Example

without noise !

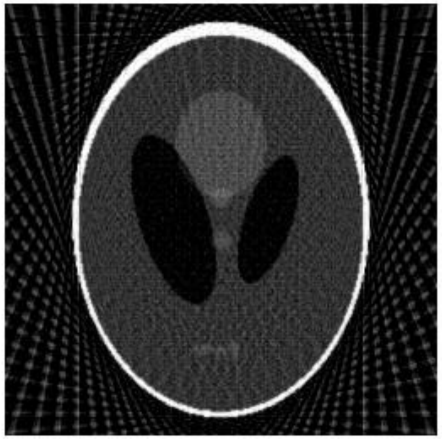
original (Shepp-Logan head phantom)



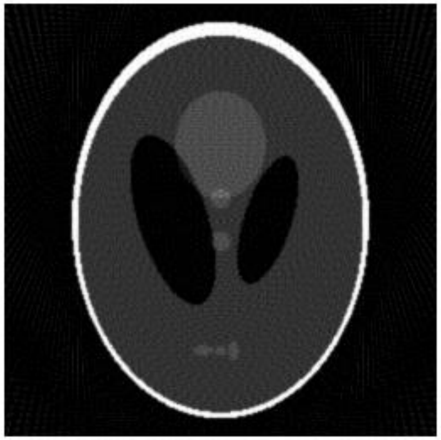
reconstruction from 18 directions



reconstruction from 36 directions



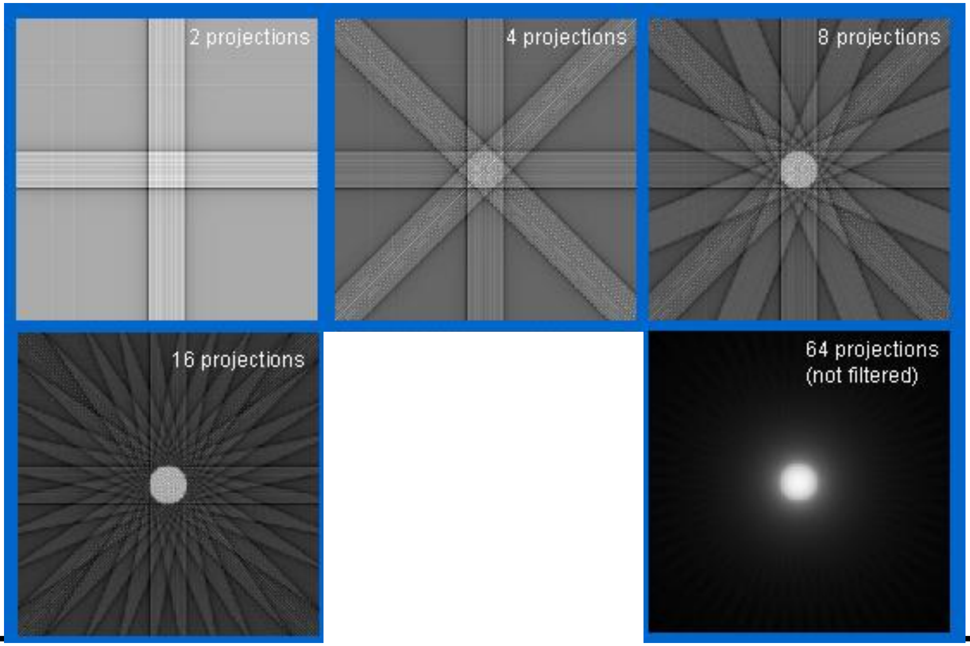
reconstruction from 90 directions



Filtered Back-Projection

- Fourier transform is linear
  - → we can sum the inverse transforms of the lines in frequency space instead of performing the inverse transform of the sum of the lines

backprojection:



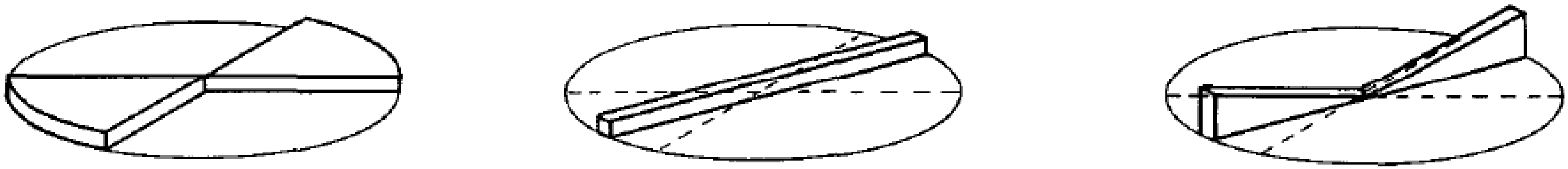
Filtered Back-Projection

- Why filtering ?
- discrete nature of measurements gives unequal weights to samples
- compensate

would like to have wedge shape for one discrete measurement

have a bar shape (discrete measurement)

compensate to have equal volume under filter

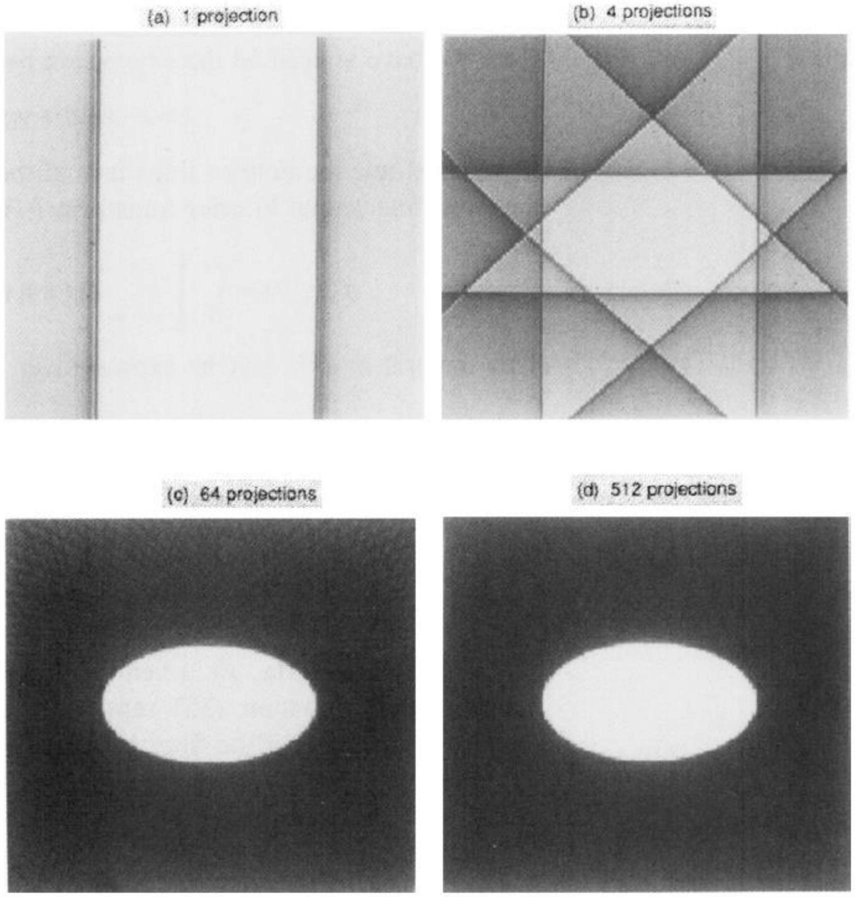


frequency domain

high pass filter

Filtered Back-Projection (FBP)

- high pass filter 1D projections in spatial domain
- back-project
- blurring is removed
  - FBP can be implemented on the GPU
  - projective texture mapping



Frequency Space based Methods

- Advantages
  - Fast processing
  - Incremental processing (FBP)
- Disadvantages
  - need orthogonal projections
  - sensitive to noise because of high pass filtering
  - Frequency-space artifacts, e.g. ringing
  - Equal angular view spacing (or adaptive filtering)

Inverse Problems

Tomography

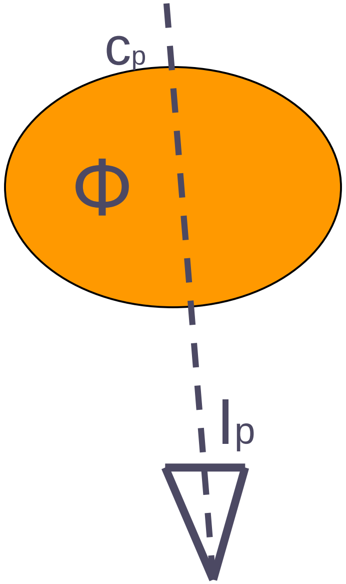
-- Algebraic Techniques --

Algebraic Reconstruction Techniques (ART)

- object described by Φ, a density field of e.g. emissive soot particles
- pixel intensities are line integrals along line of sight

$$I_p = \int_c \phi \, ds$$

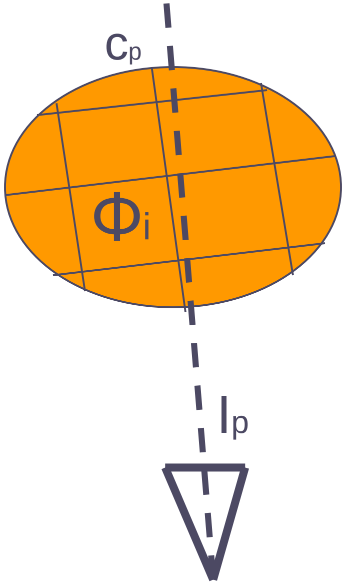
- Task: Given intensities, compute Φ



ART

- Algebraic Reconstruction Technique (ART)
- Discretize unknown Φ using a linear combination of basis functions Φ_i

$$I_p = \int_c \left(\sum_i a_i \phi_i \right) ds$$



- → linear system p = Sa

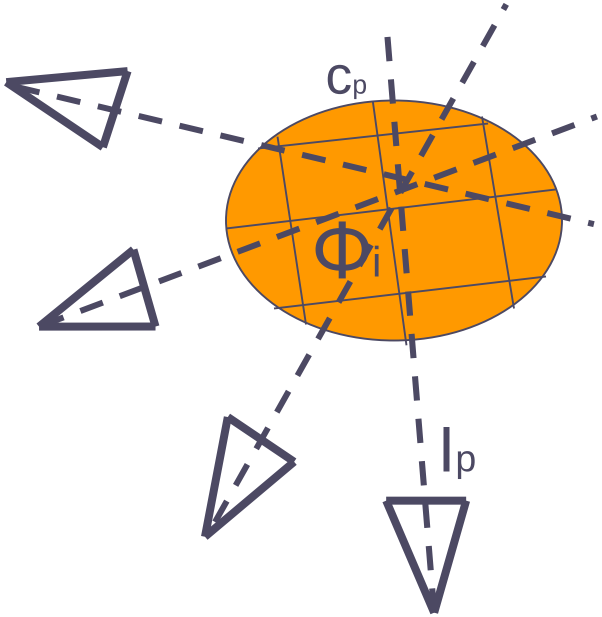
$$I_p = \sum_i a_i \left(\int_{c_p} \phi_i \, ds \right)$$

ART

- Discretize unknown Φ using a linear combination of basis functions Φi

$$I_p = \int_c \left(\sum_i a_i \phi_i \right) ds$$

$$I_p = \sum_i a_i \left(\int_{c_p} \phi_i \, ds \right)$$



- Need several views

ART – Matrix Structure

$$I_p = \sum_i a_i \left(\int_{c_p} \phi_i \, ds \right)$$

$$I = Sa$$

Basis functions

i →

pixels

p ↓

$$\begin{array}{|c|c|c|c|c|c}
\hline
\int_{c_1} \phi_1 ds & \int_{c_1} \phi_2 ds & \int_{c_1} \phi_3 ds & \int_{c_1} \phi_4 ds & \int_{c_1} \phi_5 ds & \cdots \\
\hline
\int_{c_2} \phi_1 ds & \int_{c_2} \phi_2 ds & \int_{c_2} \phi_3 ds & \int_{c_2} \phi_4 ds & \int_{c_2} \phi_5 ds & \\
\hline
\int_{c_3} \phi_1 ds & \int_{c_3} \phi_2 ds & \int_{c_3} \phi_3 ds & \int_{c_3} \phi_4 ds & \int_{c_3} \phi_5 ds & \\
\hline
\vdots & & & & &
\end{array}$$

invert LS in a least squares sense:

$$\mathbf{a = (S^T S)^{-1} S^T I}$$

Frequency Space based Methods - Disadvantages

- Advantages
  - Accomodates flexible acquisition setups
  - Can be made robust to noise (next lecture)
  - Arbitrary or adaptive discretization
  - Can be implemented on GPU
- Disadvantages
  - May be slow
  - May be memory-consumptive

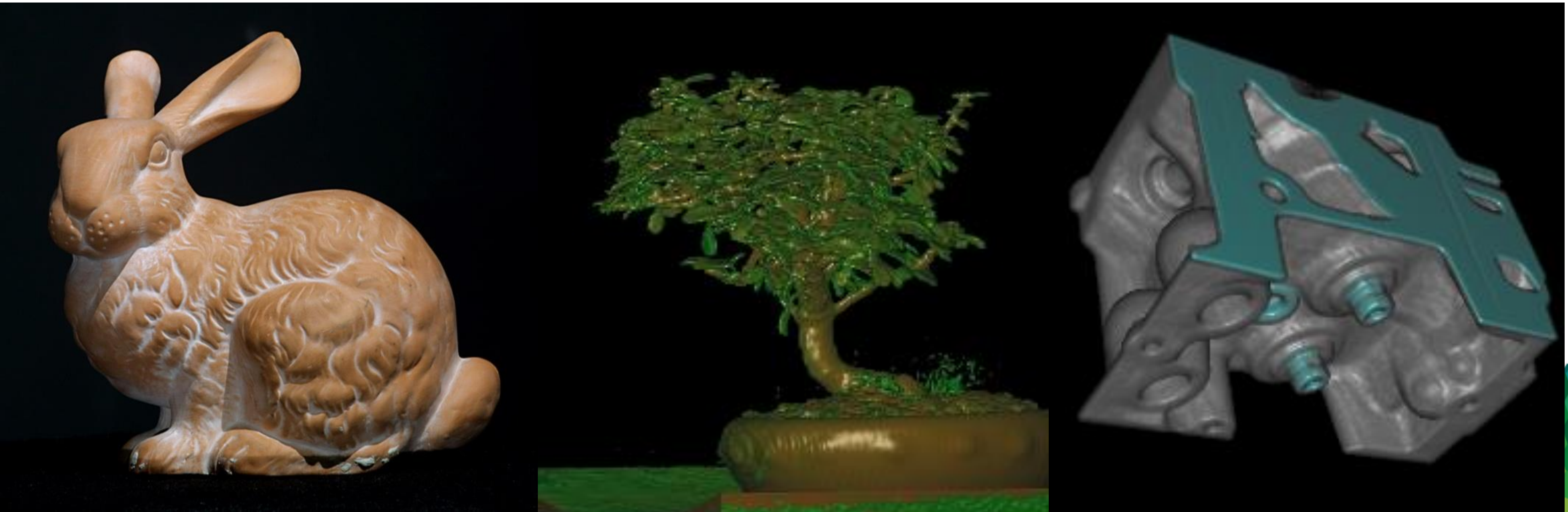
Inverse Problems

Tomography

-- Applications --

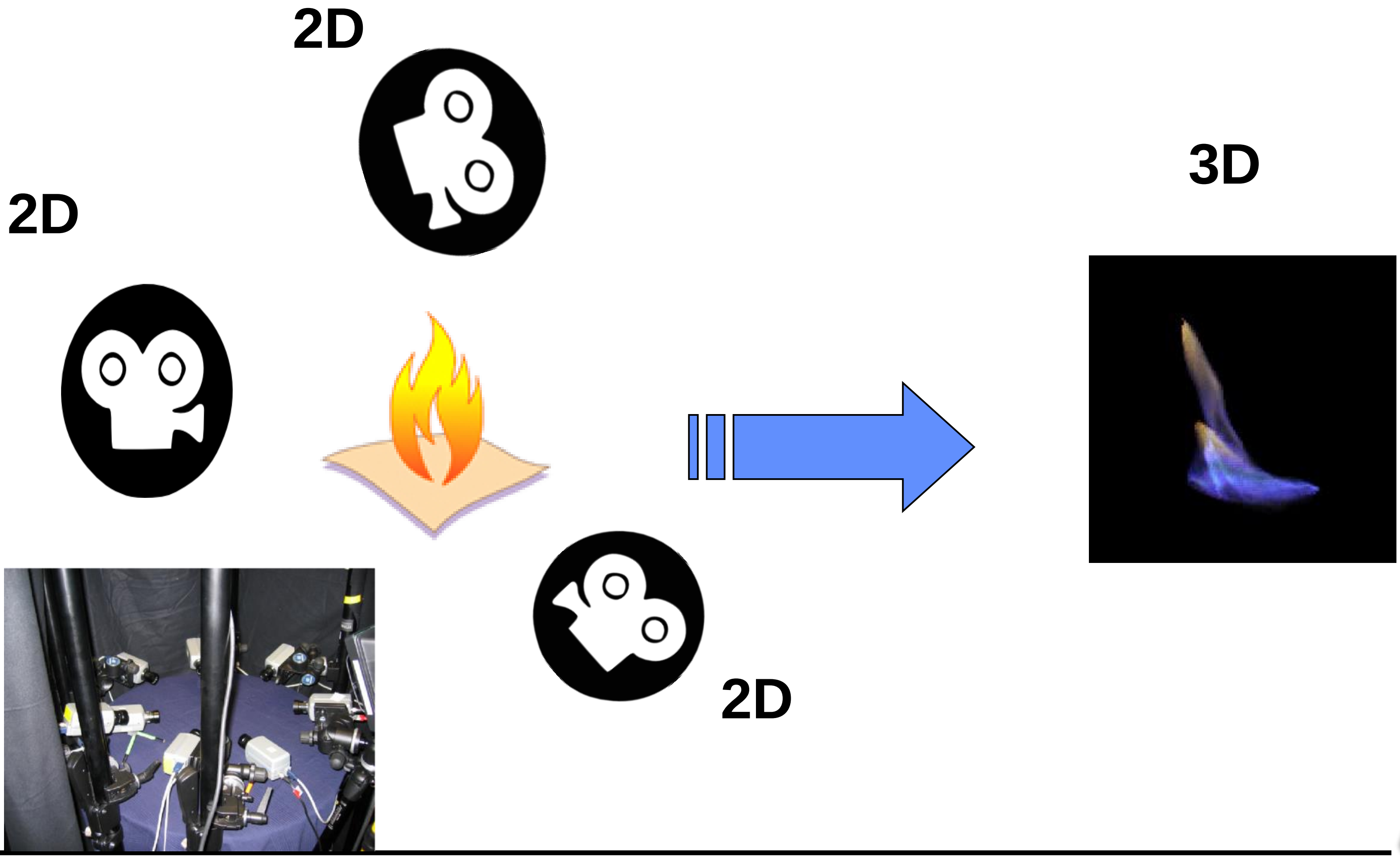
CT Applications in measurement and quality control

- Acquisition of difficult to scan objects
- Visualization of internal structures (e.g. cracks)
- No refraction



Tomographic Imaging in Graphics

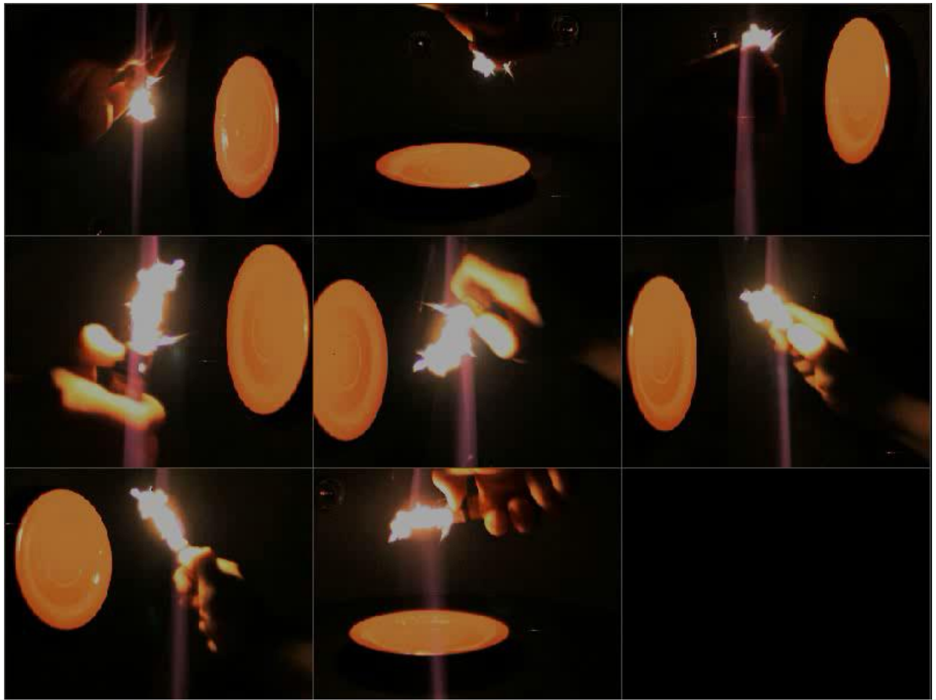
reconstruction of flames using a multi-camera setup



Flame tomography

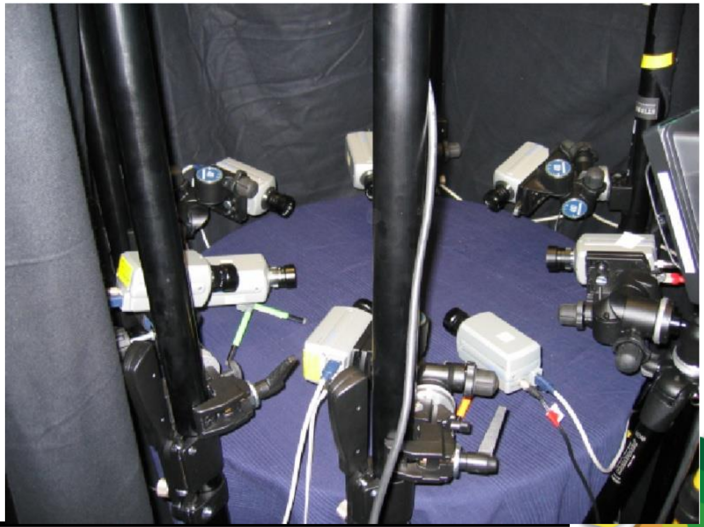
- Calibrated, synchronized camera setup
  - 8 cameras, 320 x 240 @ 15 fps

8 input views in
original camera orientation



[Ihrke' 04]

Camera setup

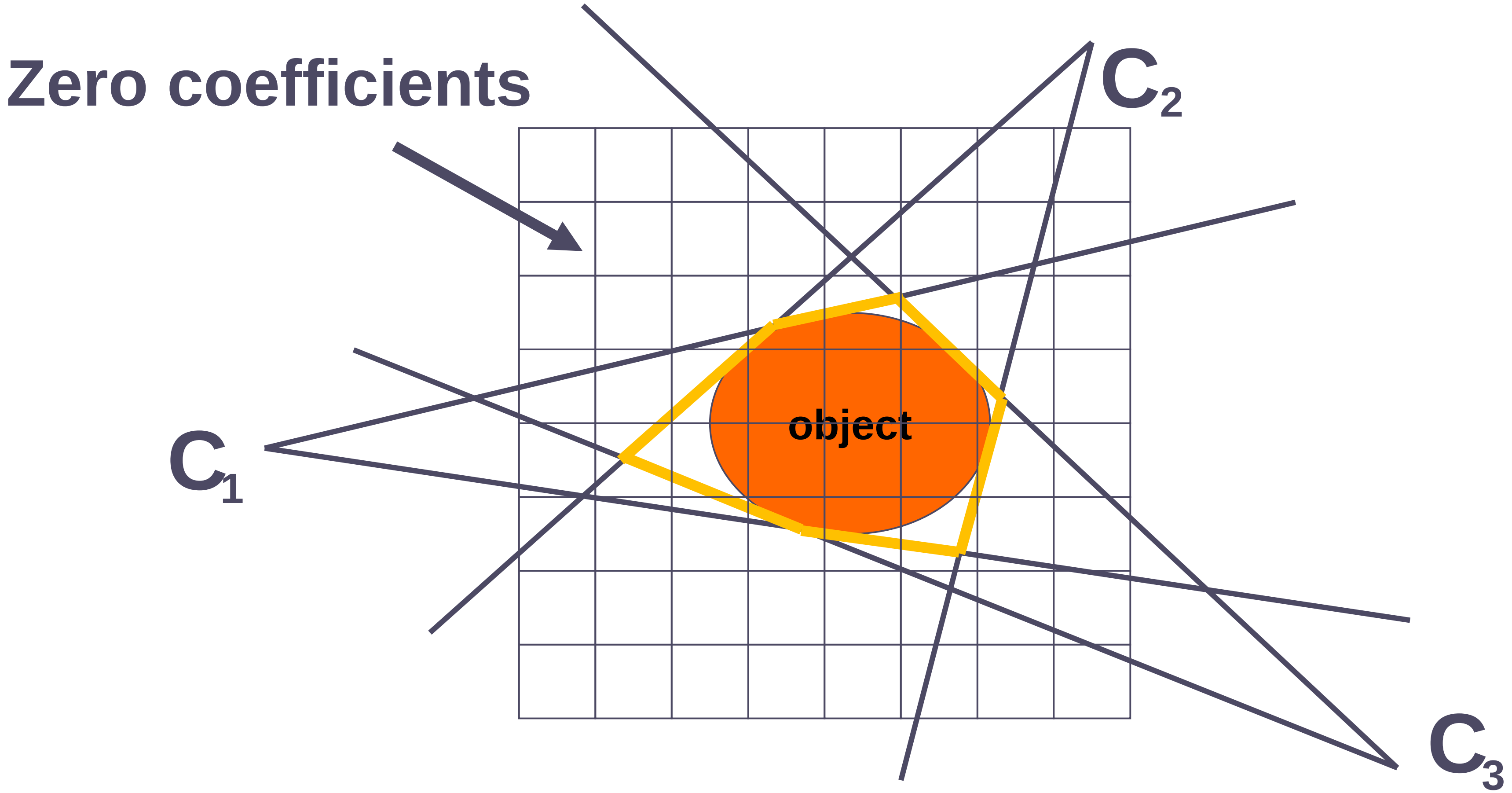


Sparse View ART - Practice

- Large number of projections is needed
- In case of dynamic phenomena
- → many cameras
  - expensive
  - inconvenient placement
- straight forward application of ART with few cameras not satisfactory

[Ihrke' 04]

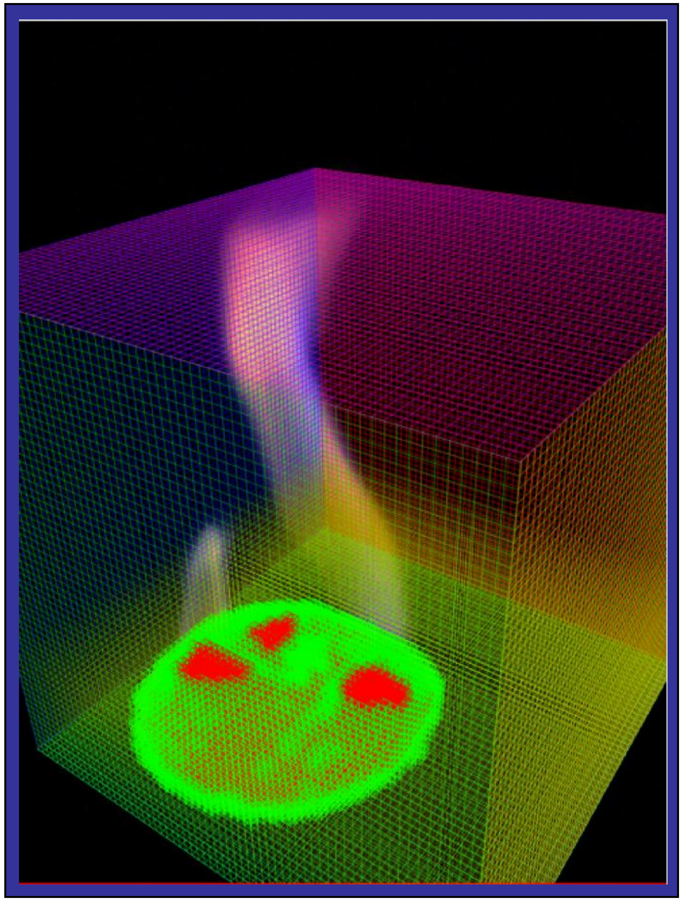
Visual Hull Restricted Tomography



[Ihrke' 04]

Visual Hull Restricted Tomography

- Only a small number of voxels contribute
- Remove voxels that do not contribute from linear system
- Complexity of inversion is significantly reduced

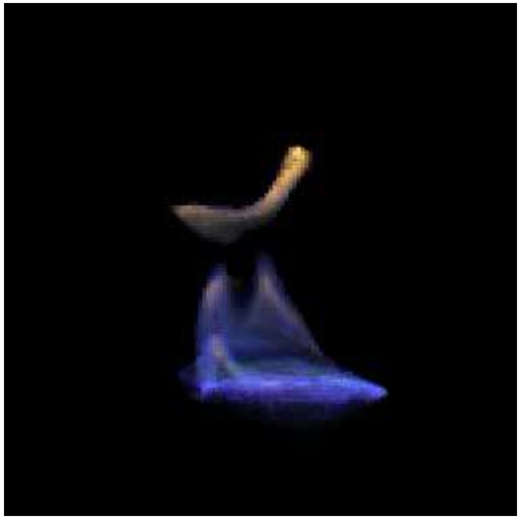


[Ihrke' 04]

Animated Flame Reconstruction

[Ihrke' 04]

frame 86

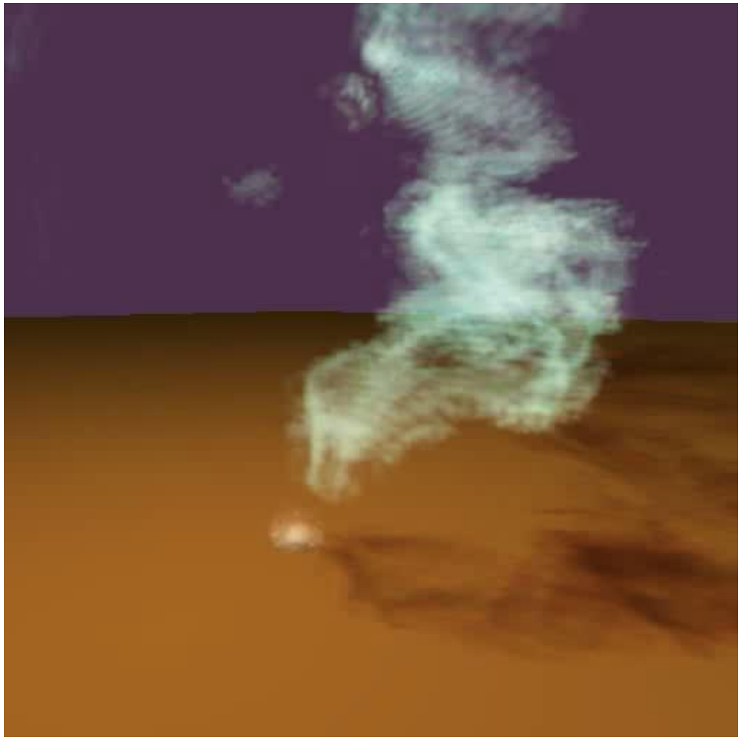


frame 194

animated reconstructed flames

Smoke Reconstructions

[Ihrke' 06]

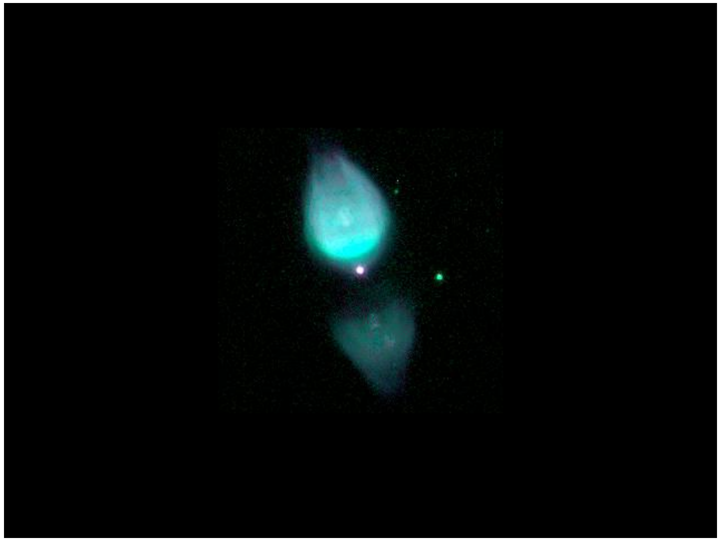


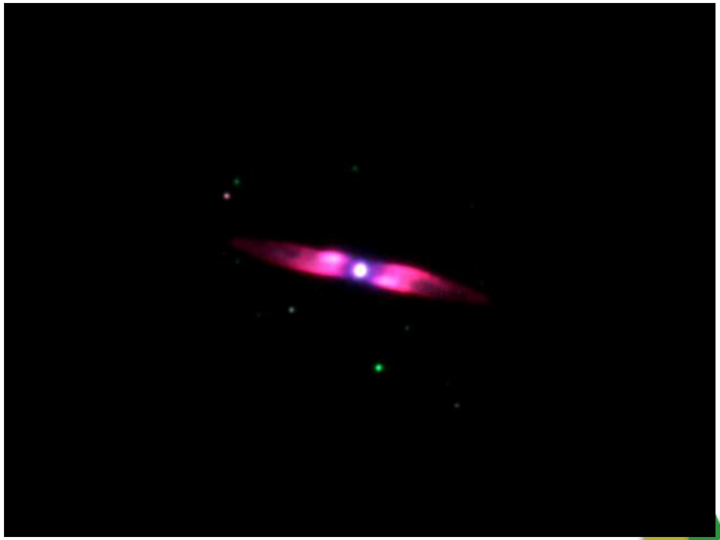
3D Reconstruction of Planetary Nebulae

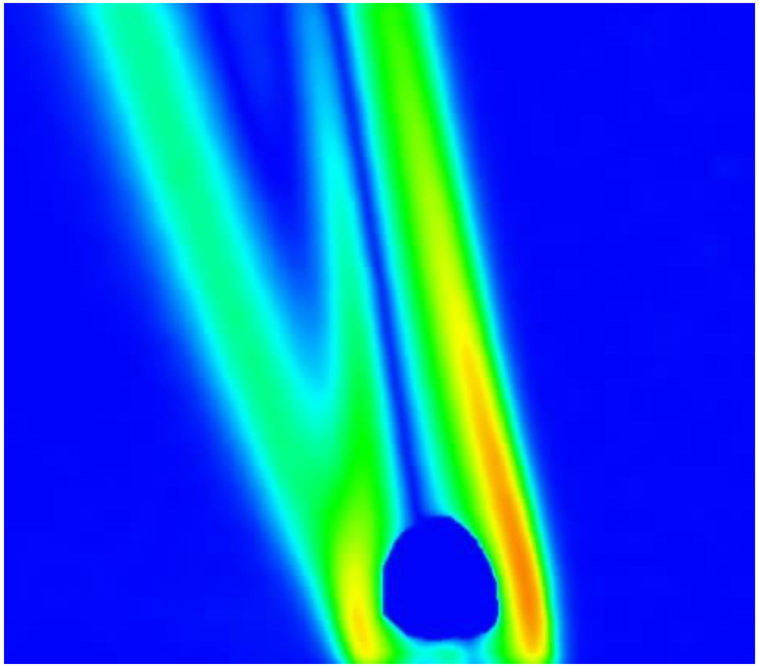
- only one view available

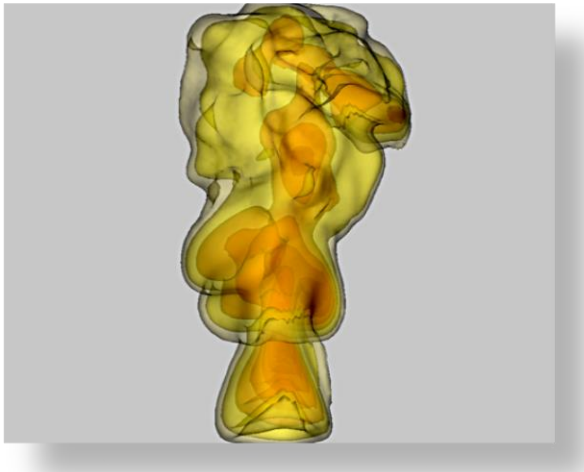
[Magnor04]

- exploit axial symmetry
- essentially a 2D problem

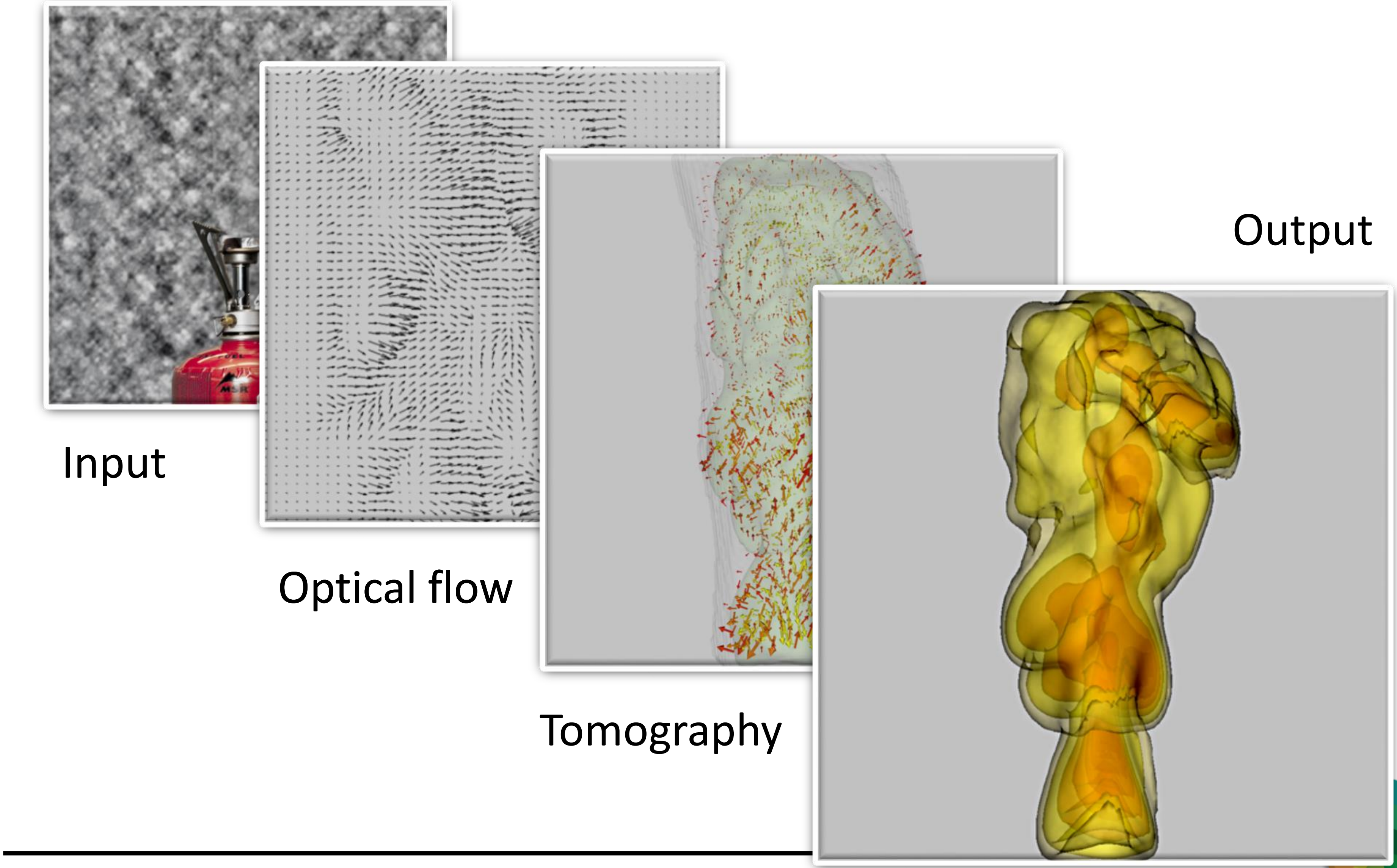








Schlieren Tomography



Input

Optical flow

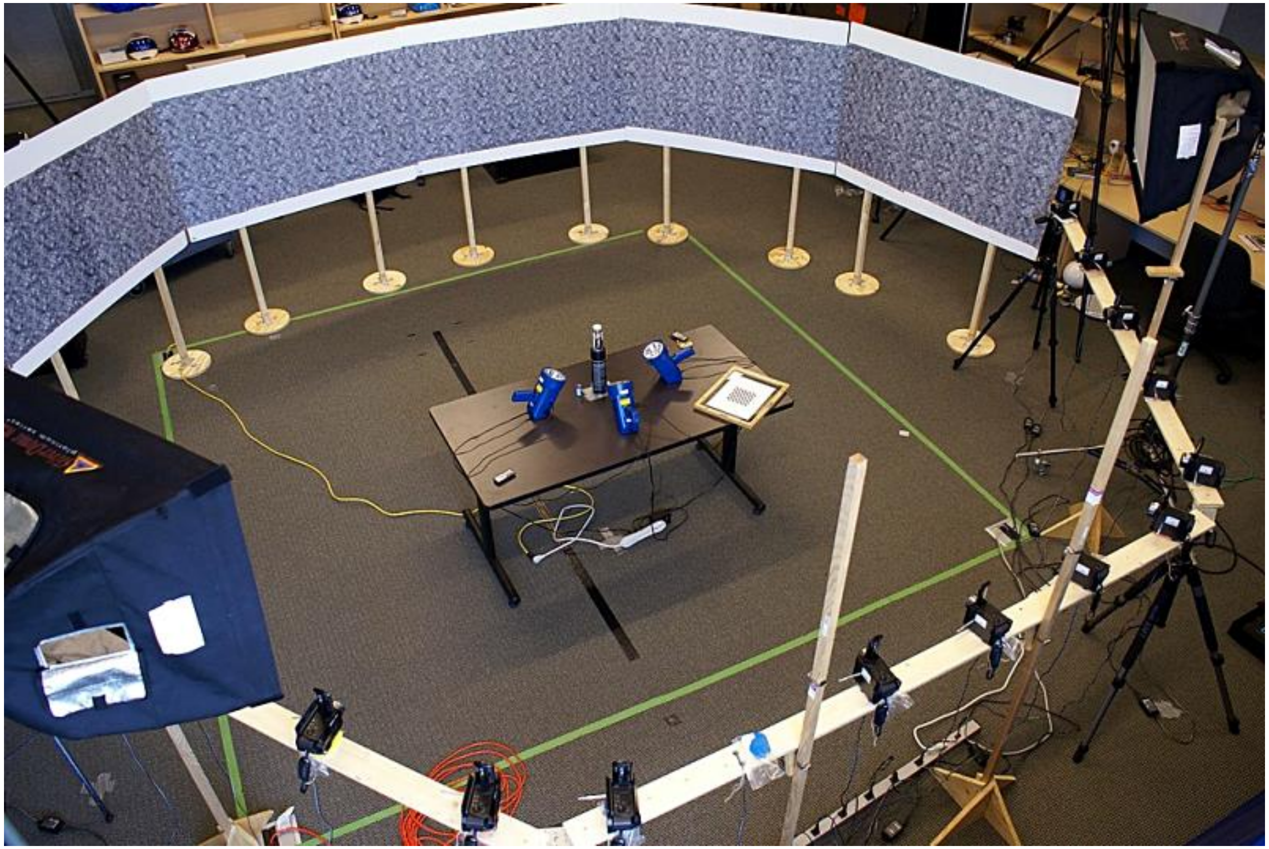
Tomography

Output

Schlieren Tomography - Acquisition

16 camera array (consumer camcorders)

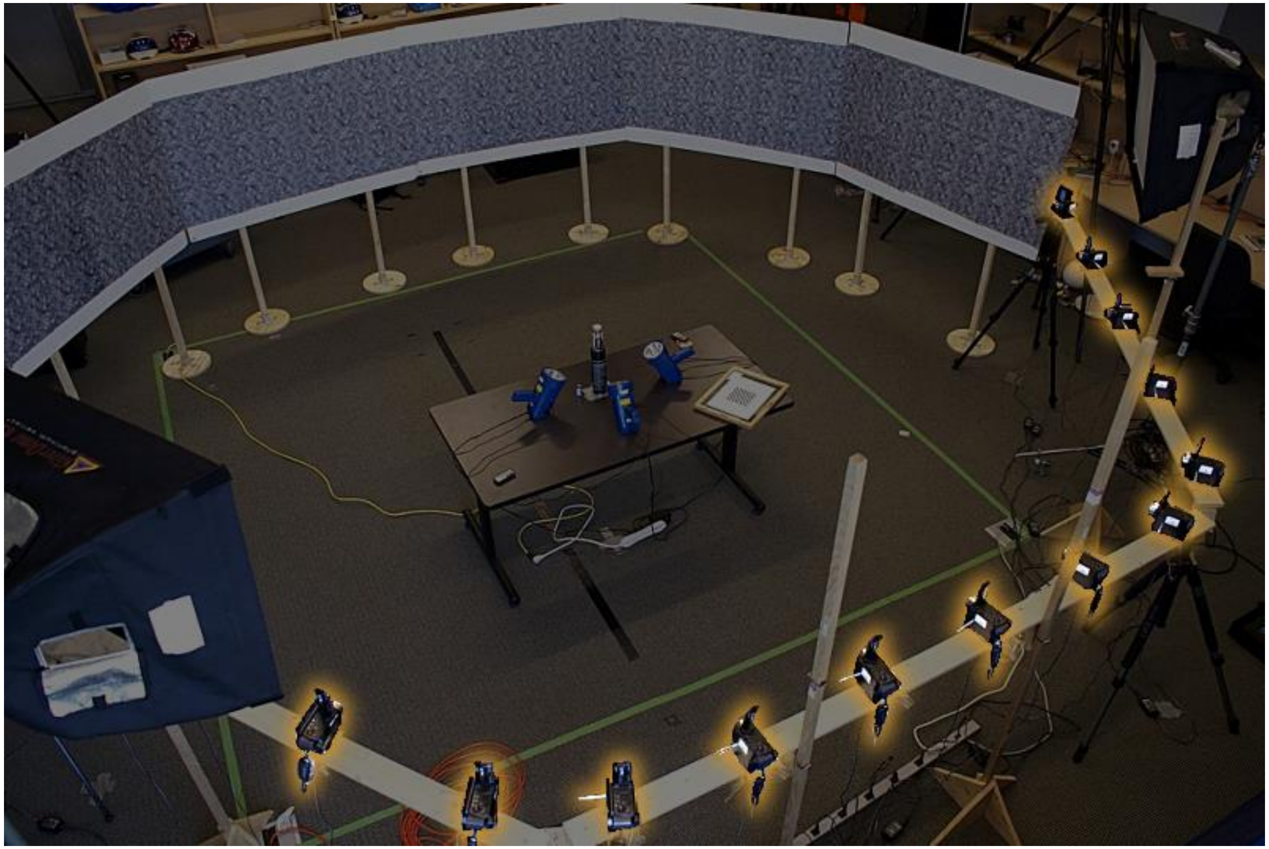
Synchronization & rolling shutter compensation



Schlieren Tomography - Acquisition

16 camera array (consumer camcorders)

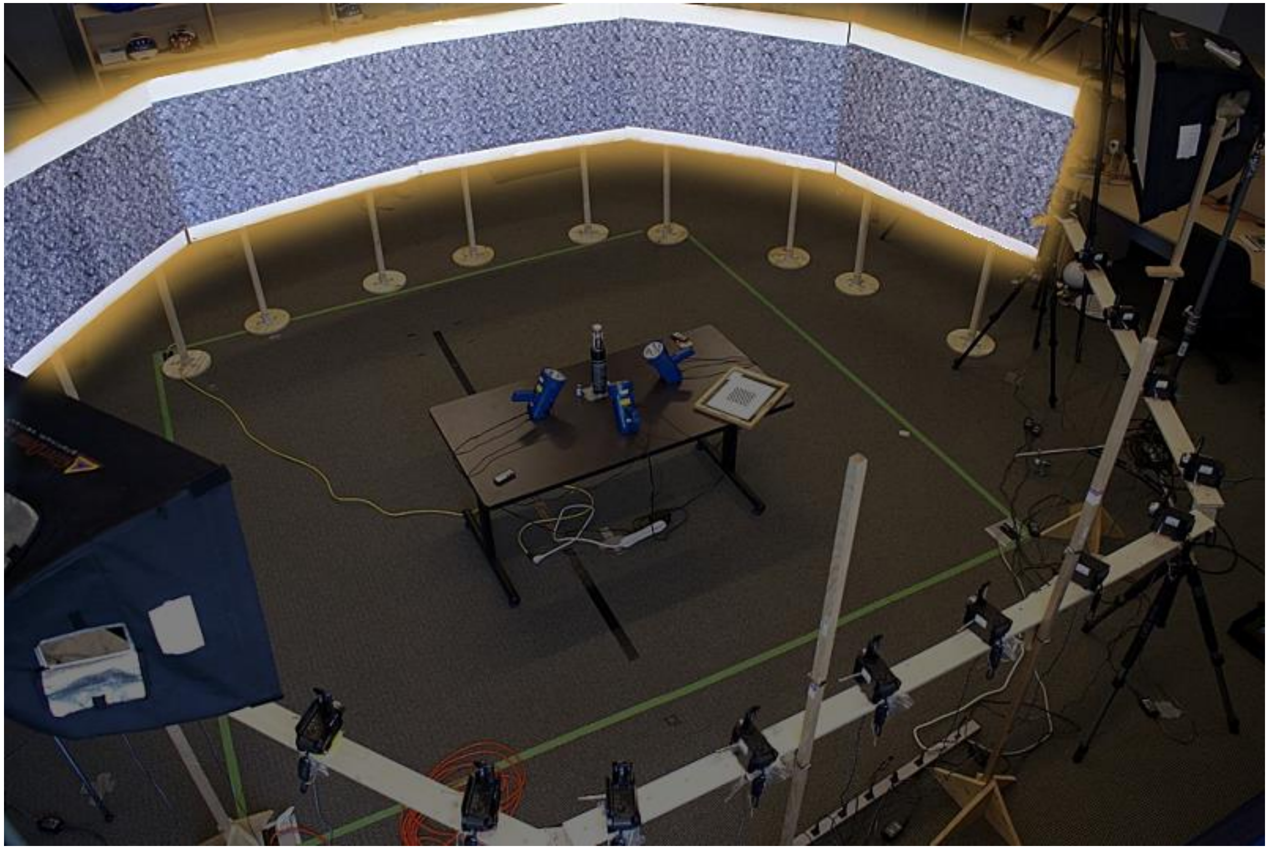
Synchronization & rolling shutter compensation



Schlieren Tomography - Acquisition

16 camera array (consumer camcorders)

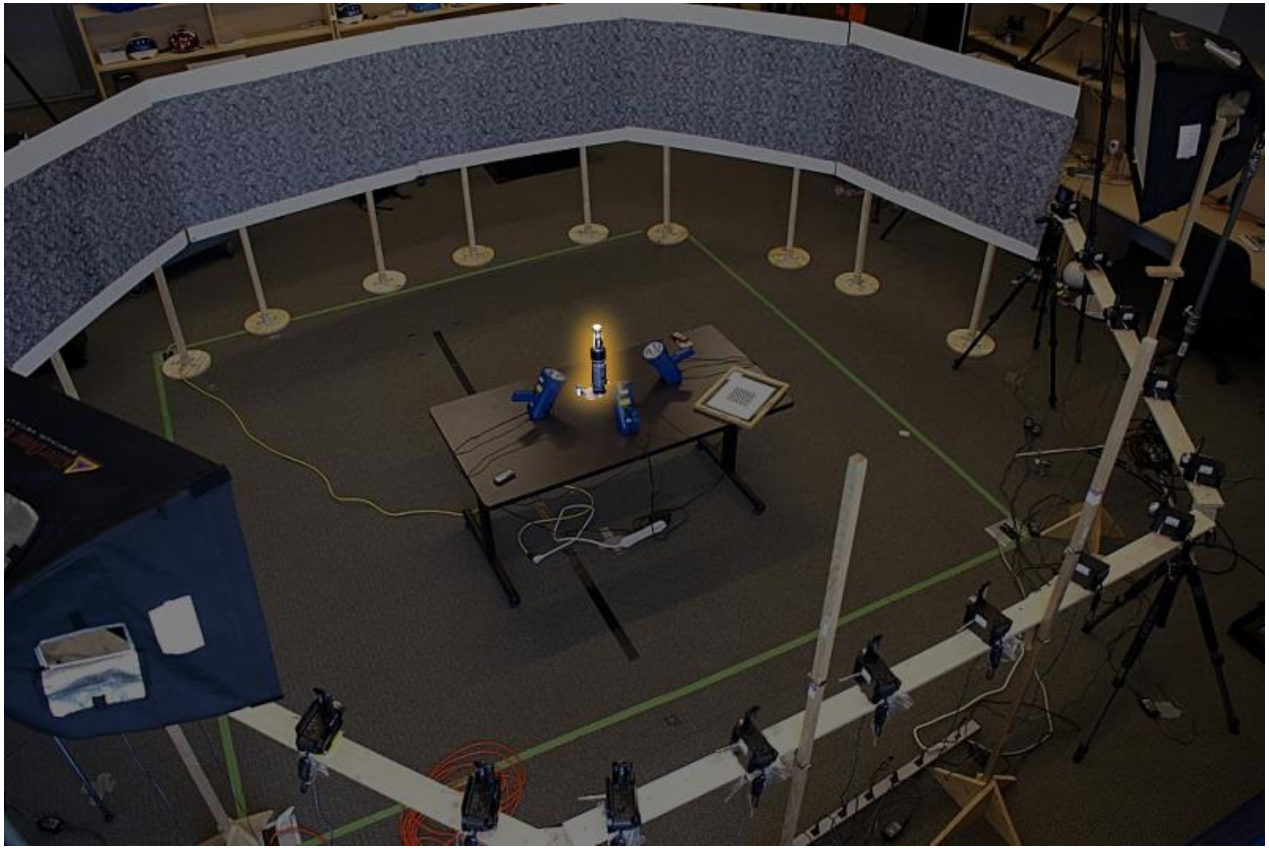
Synchronization & rolling shutter compensation



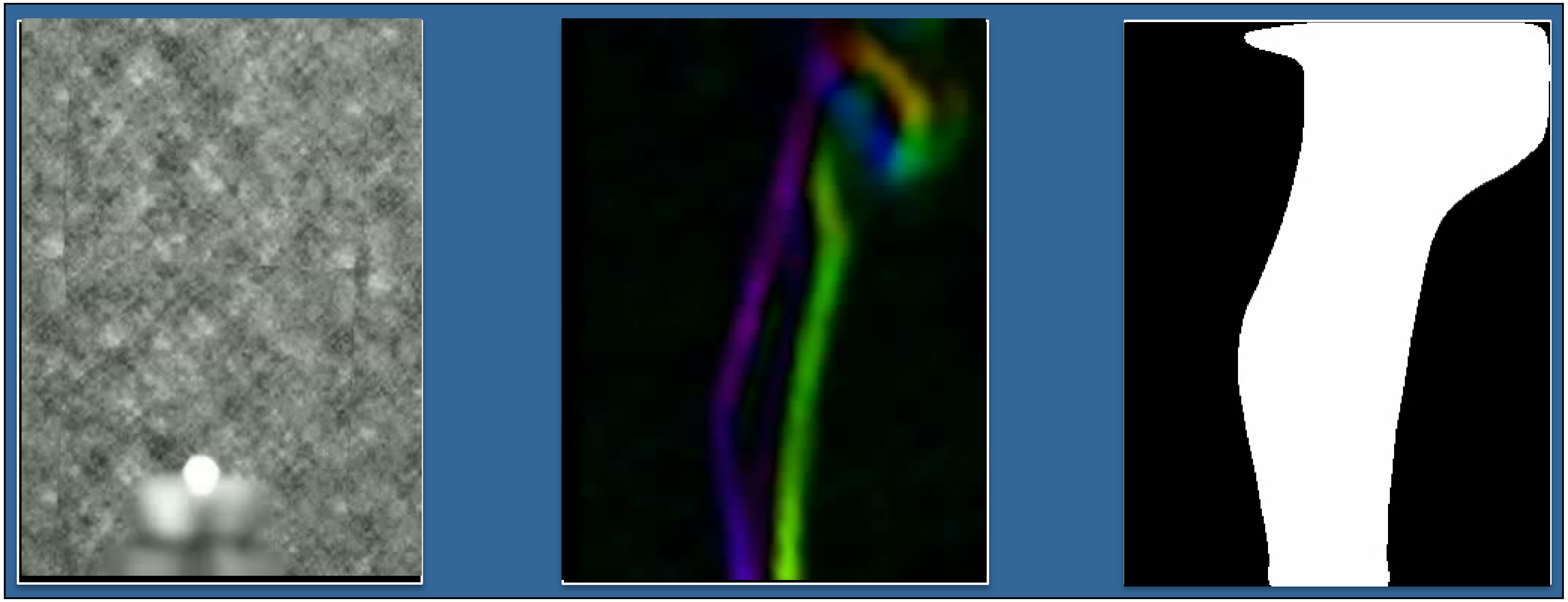
Schlieren Tomography - Acquisition

16 camera array (consumer camcorders)

Synchronization & rolling shutter compensation



Schlieren CT – Image Processing



Input

Optical flow

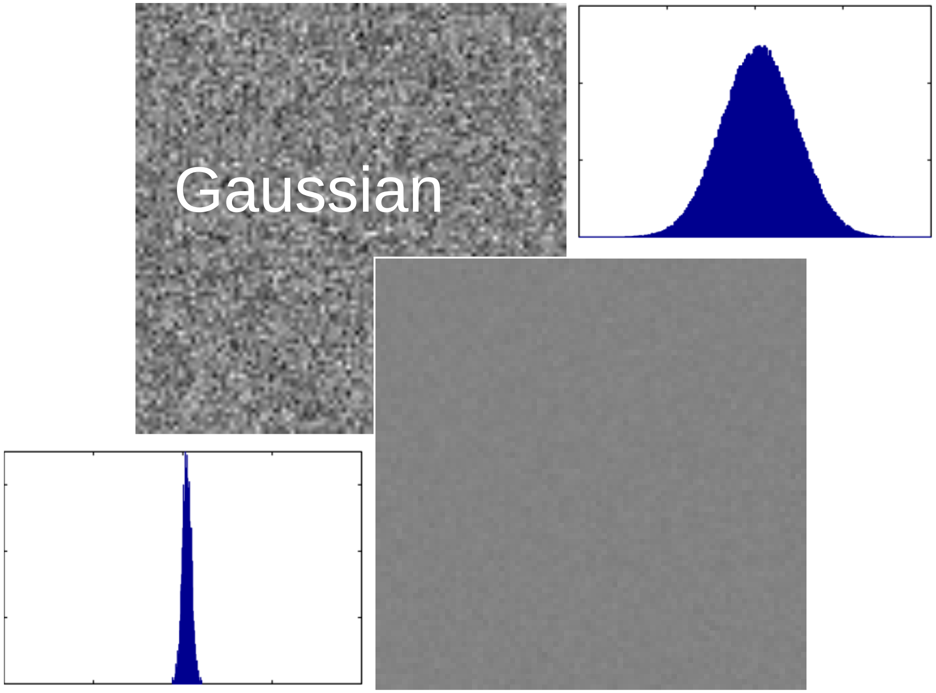
Mask

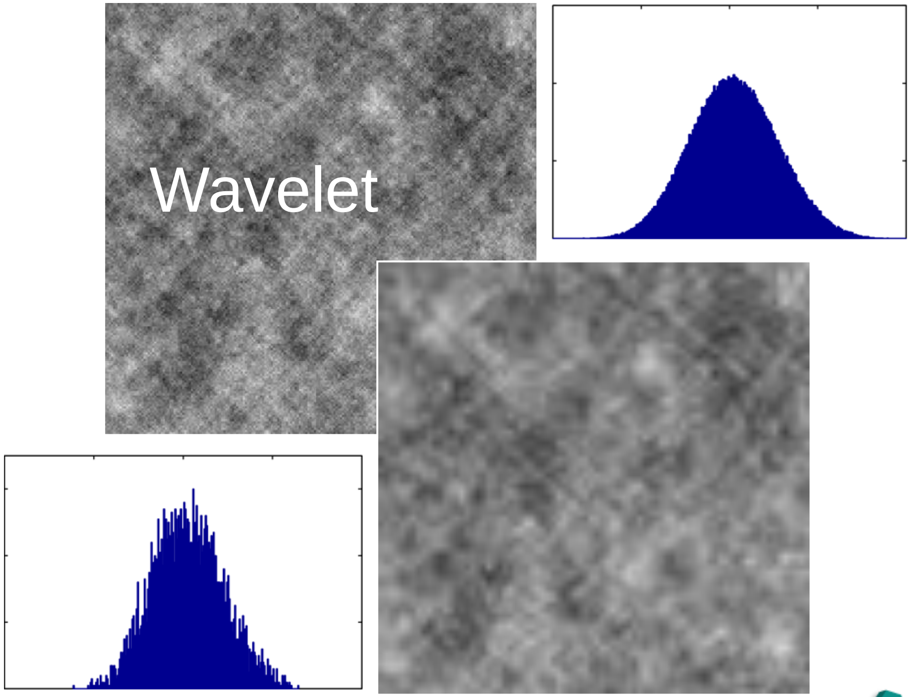
Schlieren CT – Background Pattern

High frequency detail everywhere

Decouple pattern resolution from sensor

Wavelet noise [Cook 05]





Schlieren CT - Image Formation

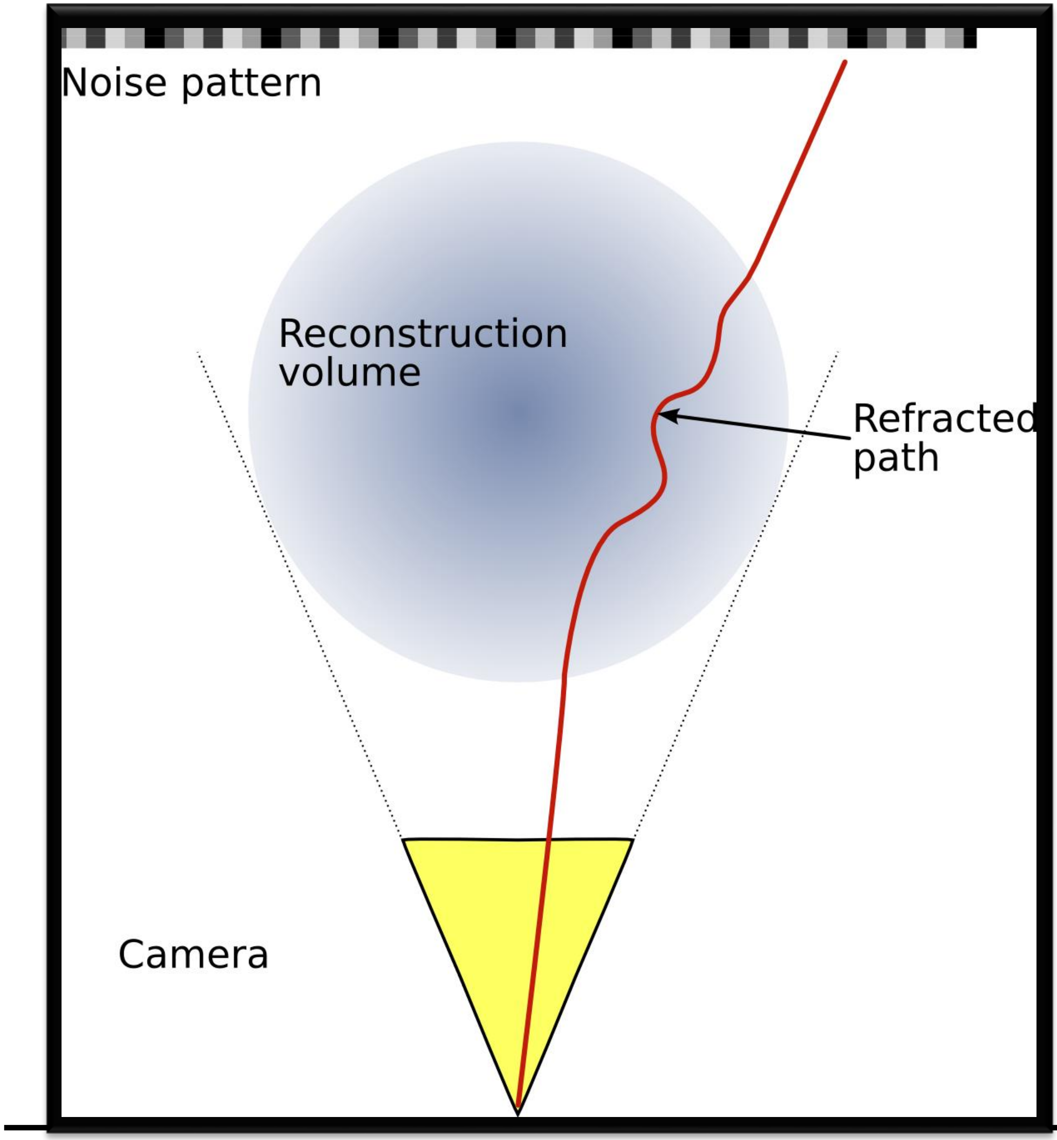
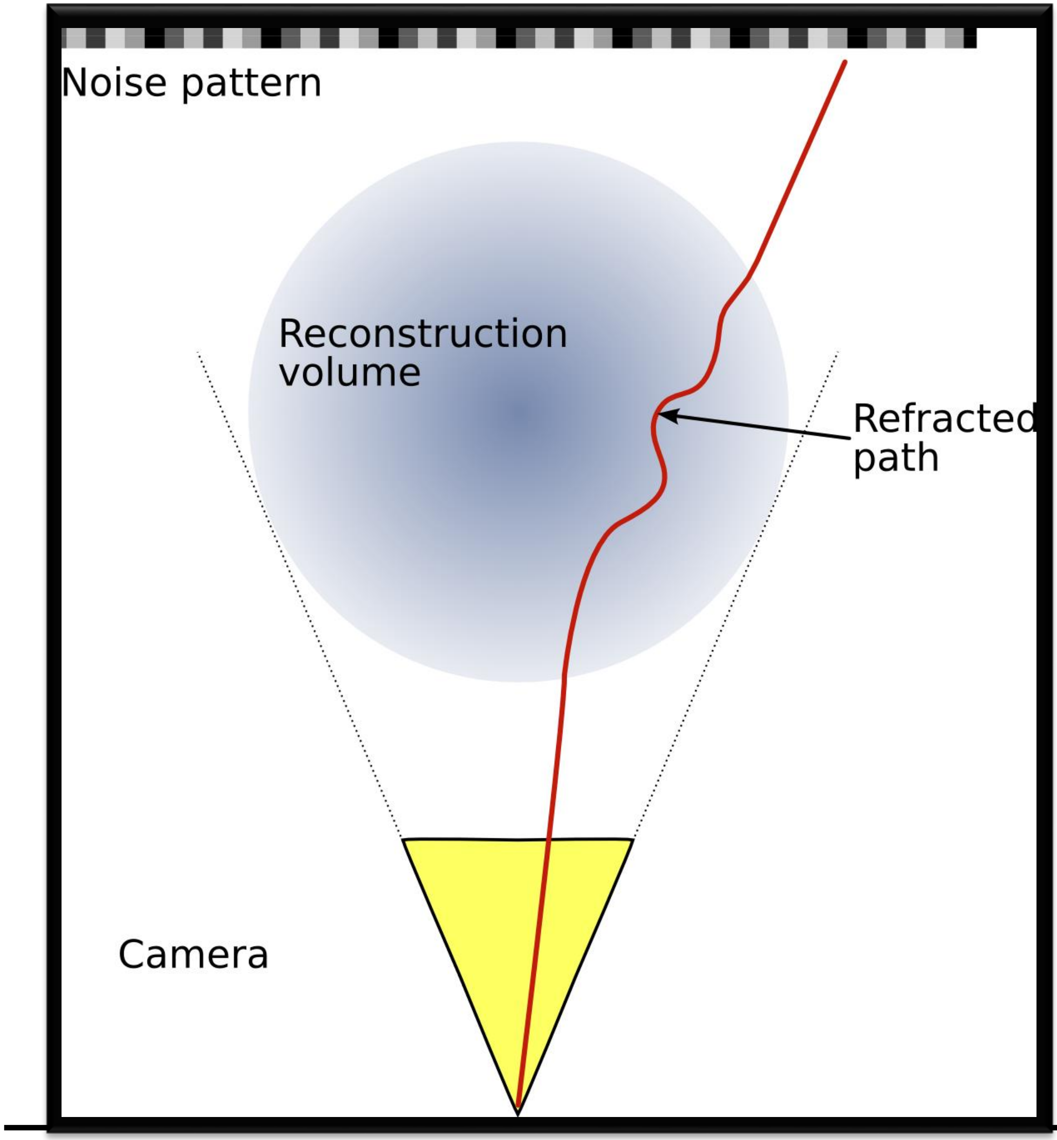


Image formation in continuously refracting media

Curved Rays

Described well by Ray Equation of Geometric Optics

Schlieren CT - Image Formation



$$\frac{d}{ds}\left(n\frac{d\mathbf{x}}{ds}\right)=\nabla n$$

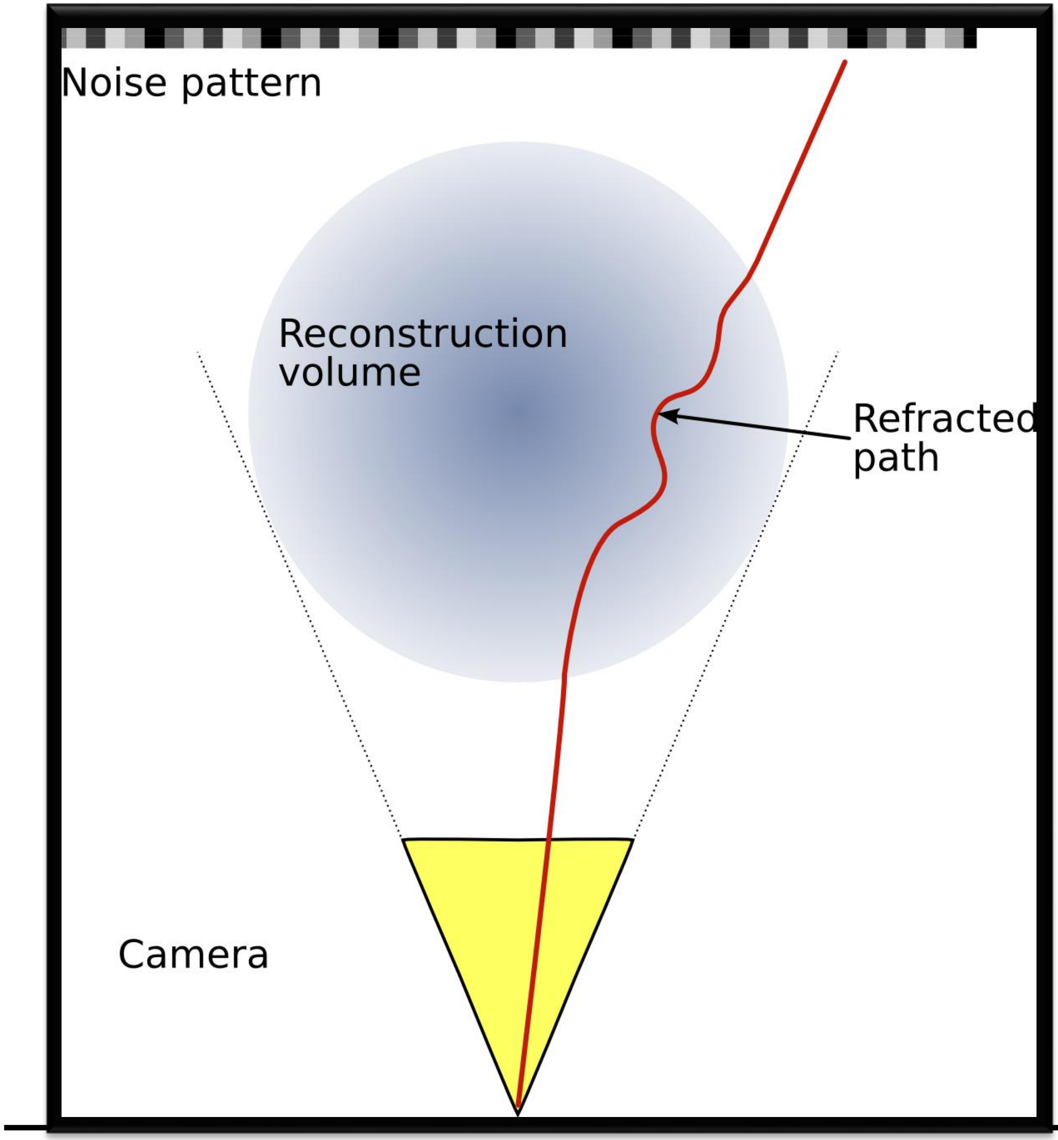
Continuous ray tracing, e.g. [Stam 96, Ihrke 07]

Set of 1st order ODE's :

$$n\frac{d\mathbf{x}}{ds}=\mathbf{d}$$

$$\frac{d\mathbf{d}}{ds}=\nabla n$$

Schlieren CT - Ray equation



$$\frac{d}{ds}\left(n\frac{d\mathbf{x}}{ds}\right) = \nabla n$$

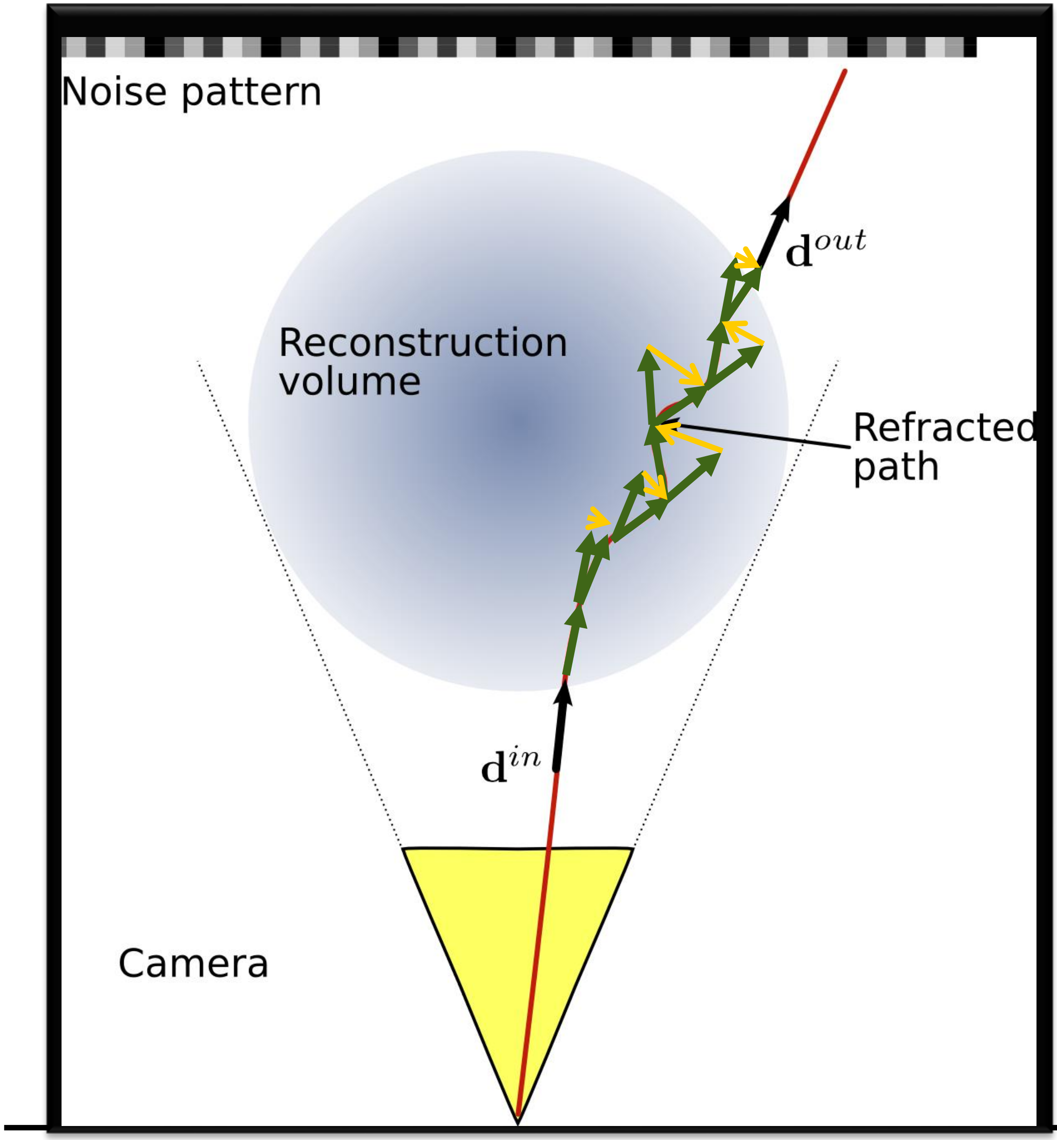
Continuous ray tracing, e.g. [Stam 96, Ihrke 07]

Set of ODE's :

$$n\frac{d\mathbf{x}}{ds} = \mathbf{d}$$

$$\boxed{\frac{d\mathbf{d}}{ds} = \nabla n}$$

Schlieren CT - Ray equation



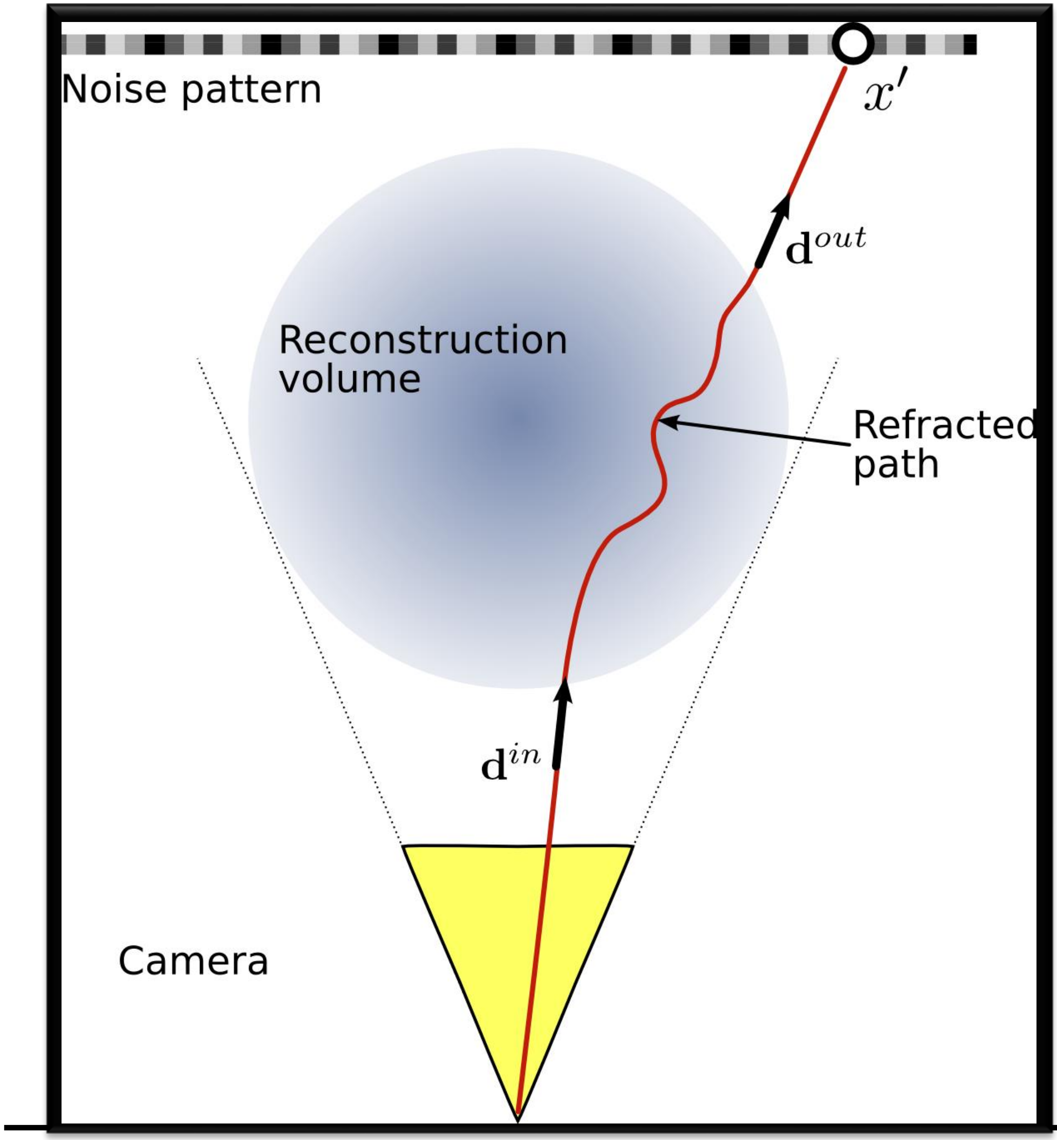
Integrating

$$\frac{d\mathbf{d}}{ds} = \nabla n$$

yields

$$\mathbf{d}^{out} = \mathbf{d}^{in} + \int_c \nabla n \, ds$$

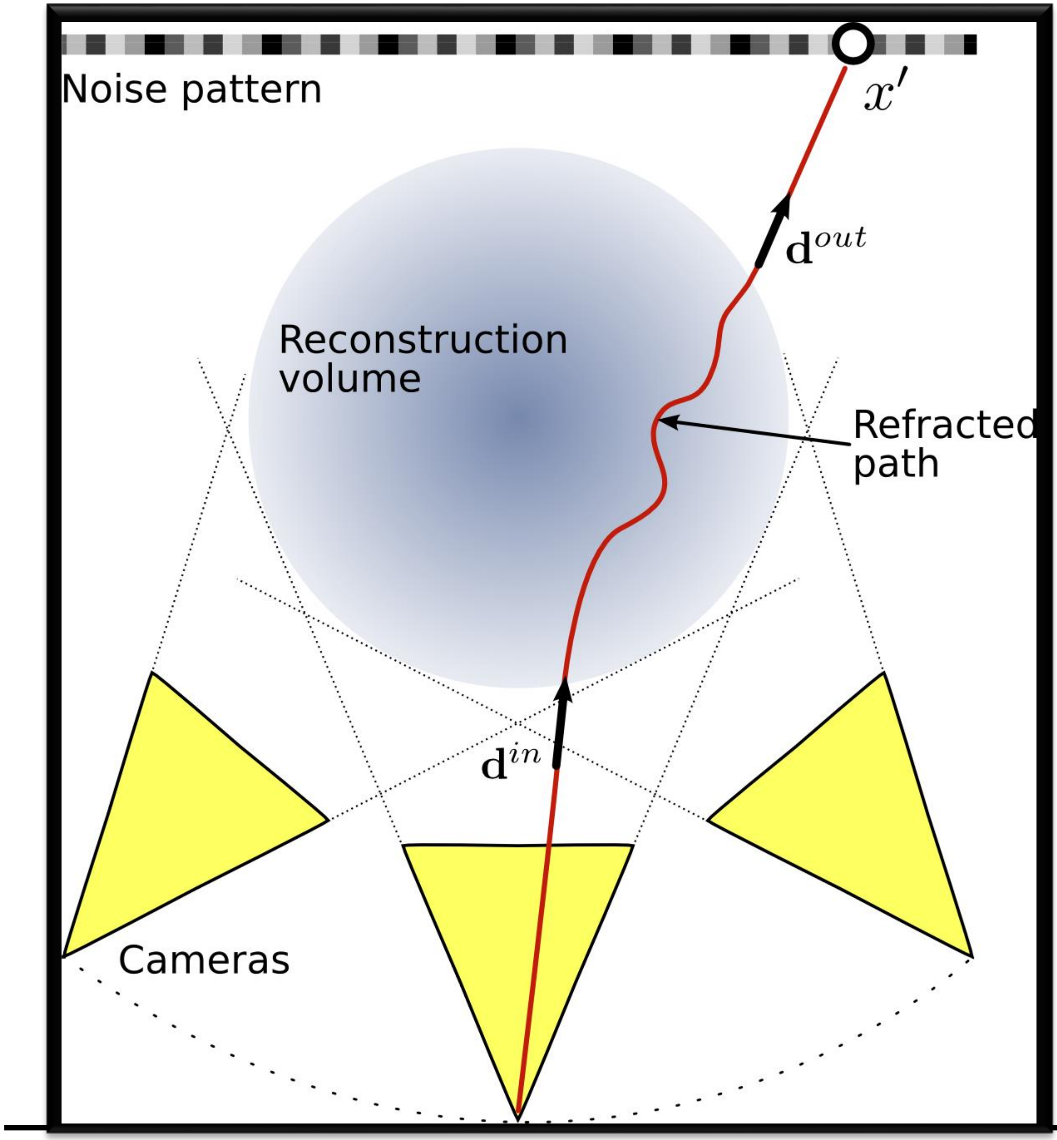
Schlieren Tomography



Basic equation for Schlieren Tomography

$$\mathbf{d}^{out} - \mathbf{d}^{in} = \int_c \nabla n \, ds$$

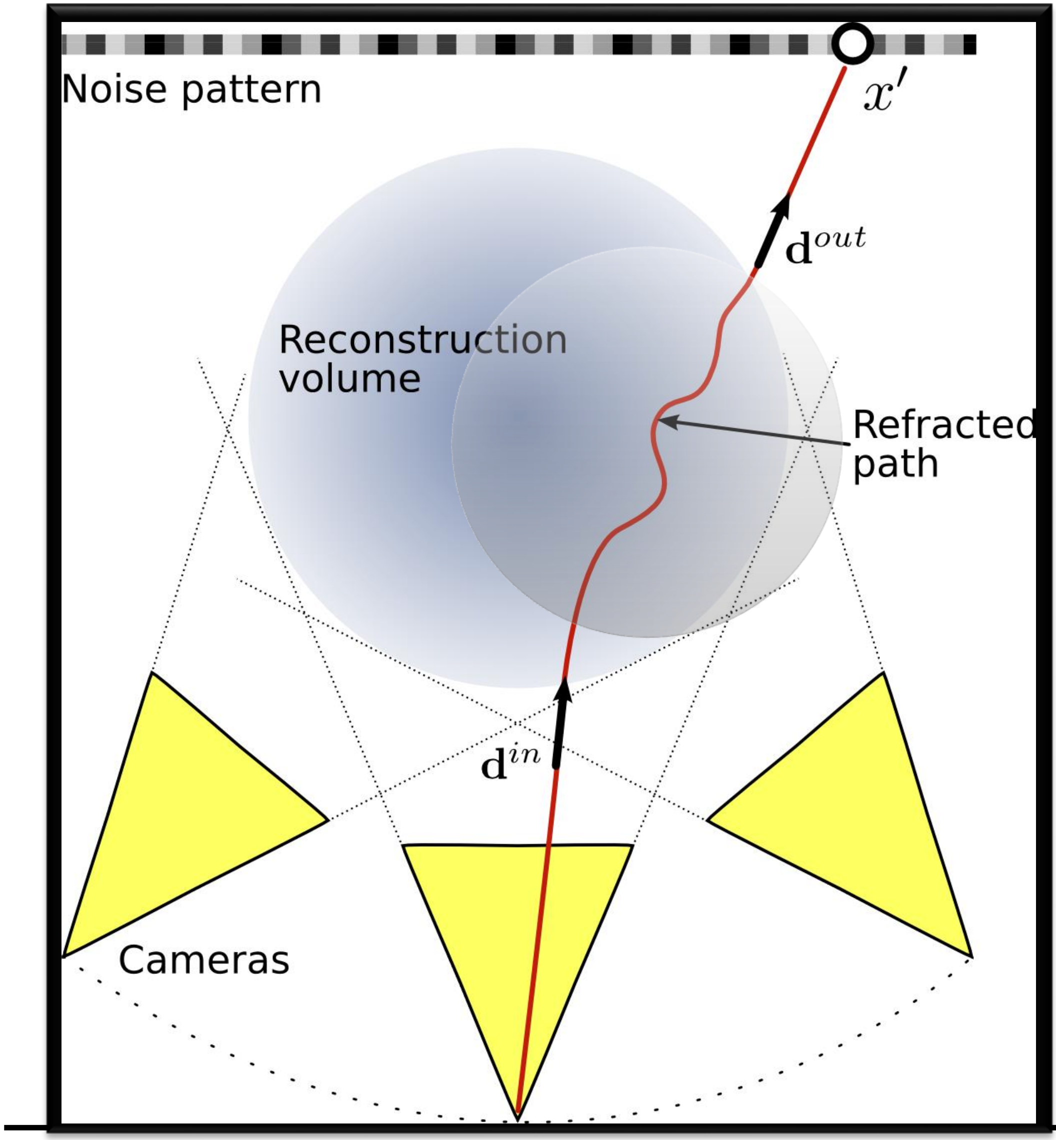
Schlieren Tomography



Based on measurements of line integrals from different orientations

$$\mathbf{d}^{out} - \mathbf{d}^{in} = \int_{c} \nabla n \, ds$$

Schlieren Tomography



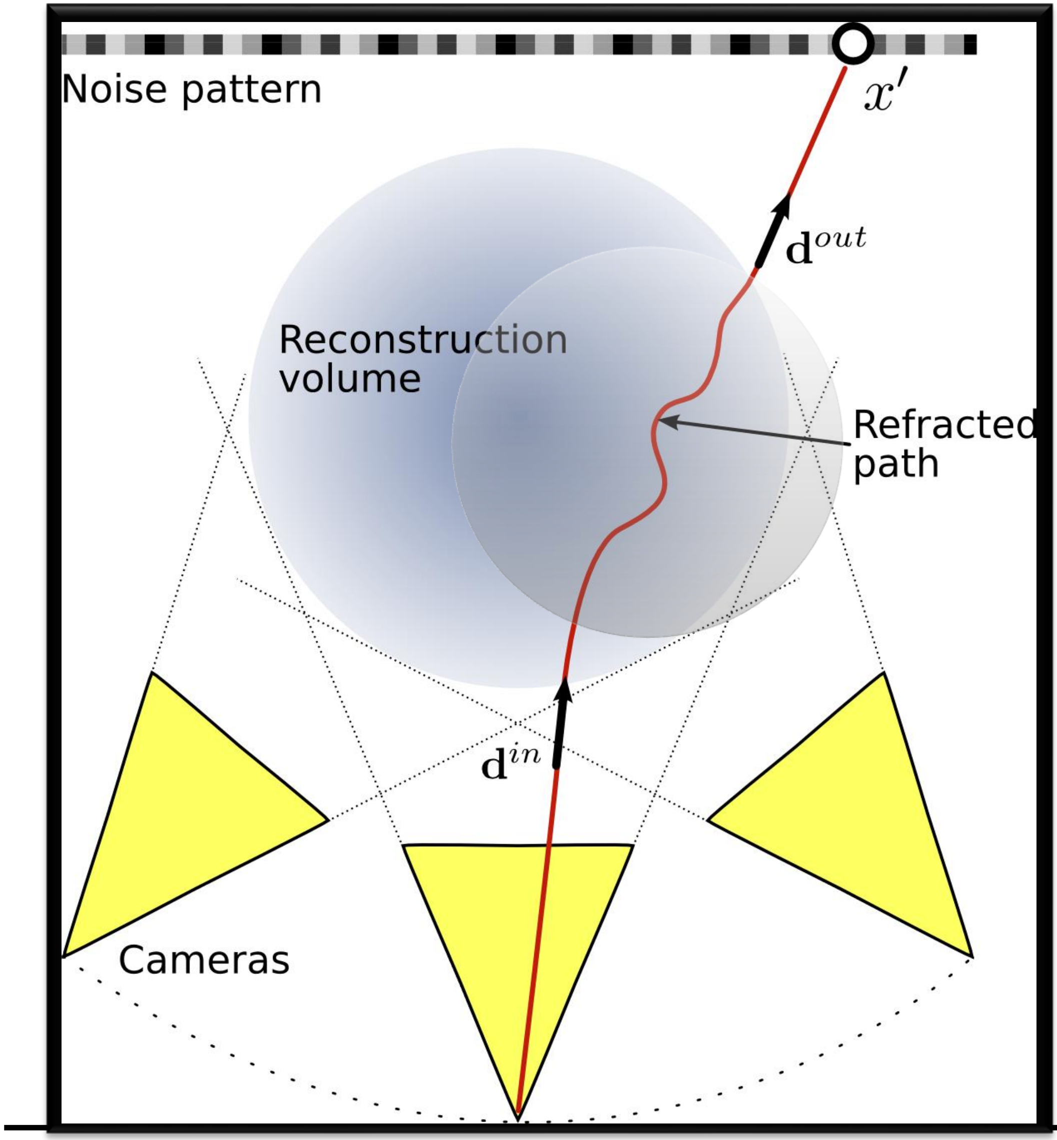
Ray path must be known

BUT: unknown refractive index

In practice, ray bending negligible [Venkatakrishnan'04]

$$\mathbf{d}^{out} - \mathbf{d}^{in} = \int_c \nabla n \, ds$$

Schlieren Tomography



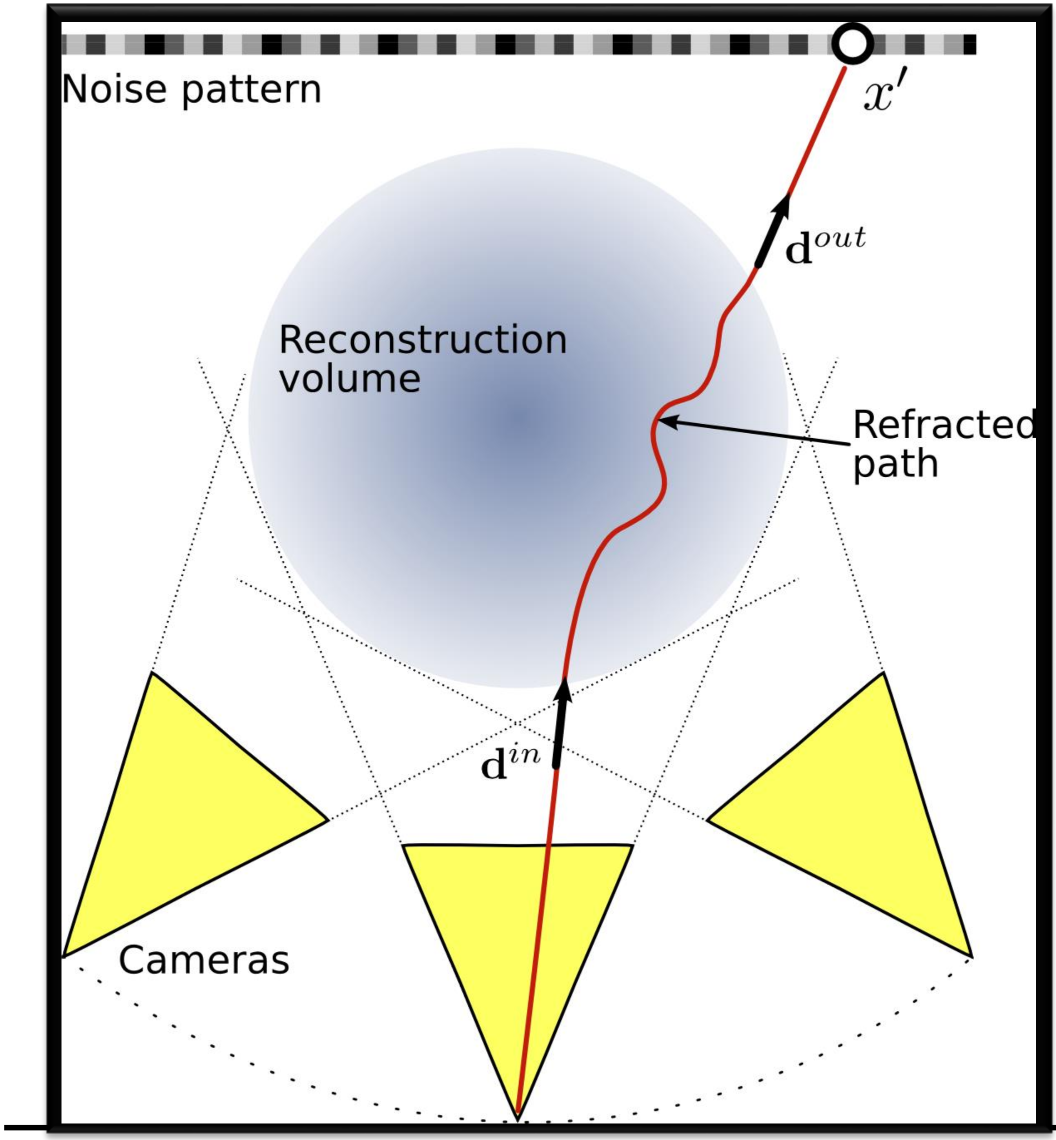
Ray path must be known

BUT: unknown refractive index

Affects integration path only, equation still holds approximately!

$$\mathbf{d}^{out} - \mathbf{d}^{in} = \int_c \nabla n \, ds$$

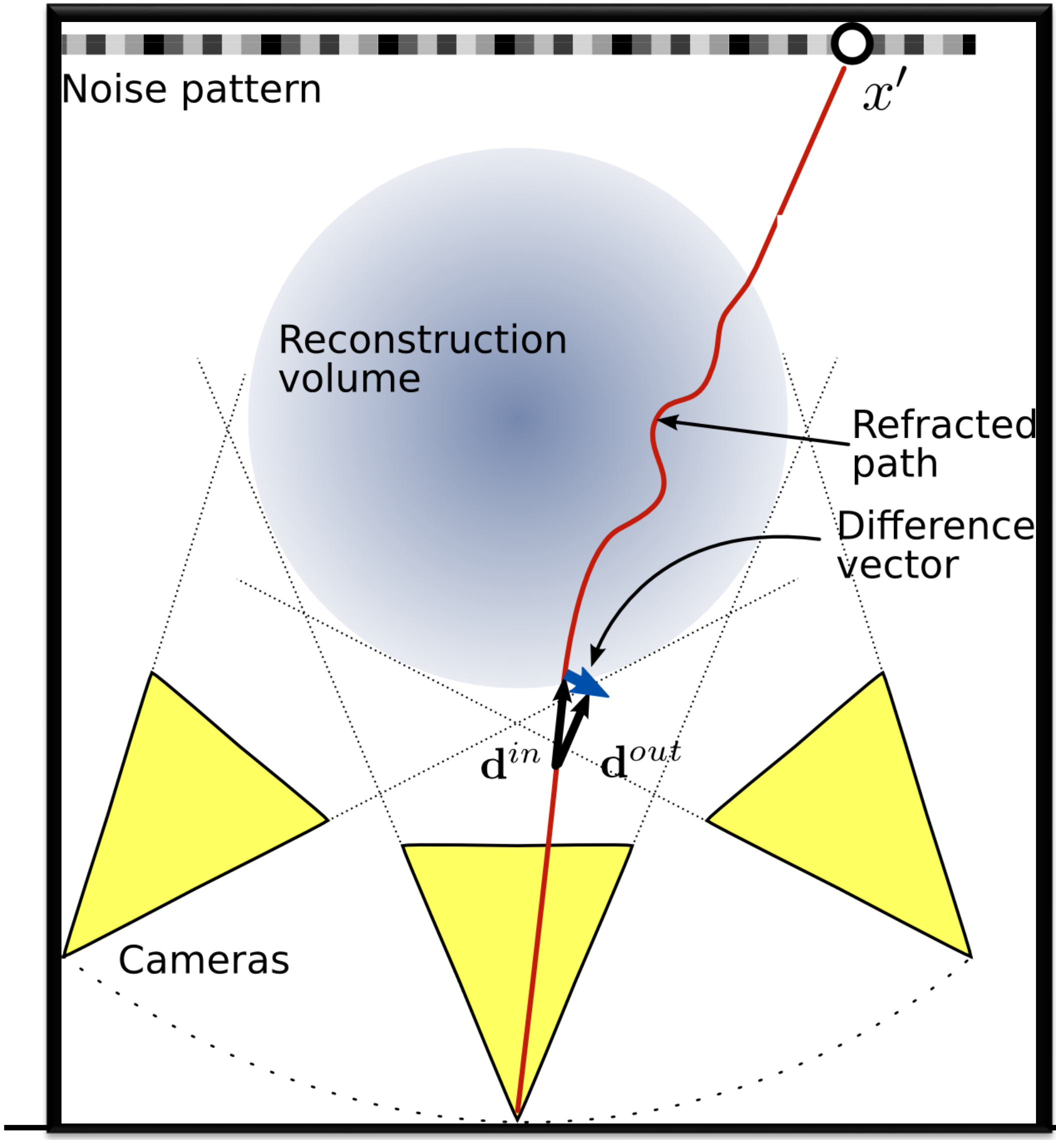
Schlieren Tomography - Measurements



Measure difference vector

$$\mathbf{d}^{out} - \mathbf{d}^{in} = \int_c \nabla n \, ds$$

Schlieren Tomography - Measurements



Measure difference vector

Component parallel to optical axis is lost

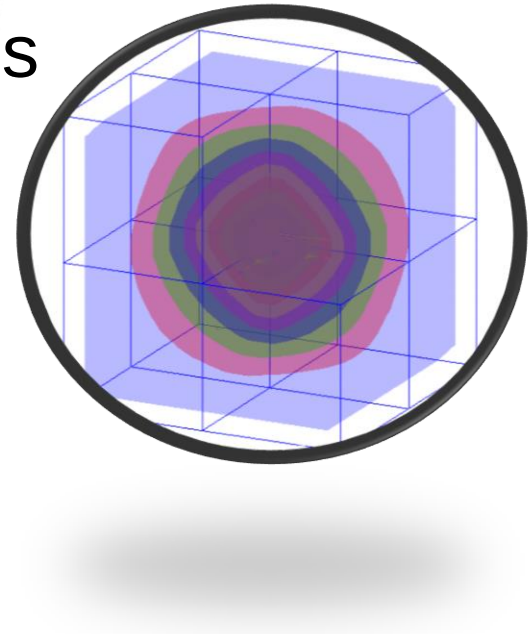
$$\mathbf{d}^{out} - \mathbf{d}^{in} = \int_c \nabla n \, ds$$

Schlieren Tomography –Linear System

Vector-valued tomographic problem

Discretize gradient

Radially symmetric basis functions



$$\overline{\nabla n} = \sum_i \mathbf{n}_i \phi_i$$

Linear system in

$$\overline{\mathbf{d}}^{out} - \mathbf{d}^{in} = \int \sum_i \mathbf{n}_i \phi_i ds = \sum_i \mathbf{n}_i \int \phi_i \, ds$$

Schlieren Tomography - Integration

Given ∇n from tomography

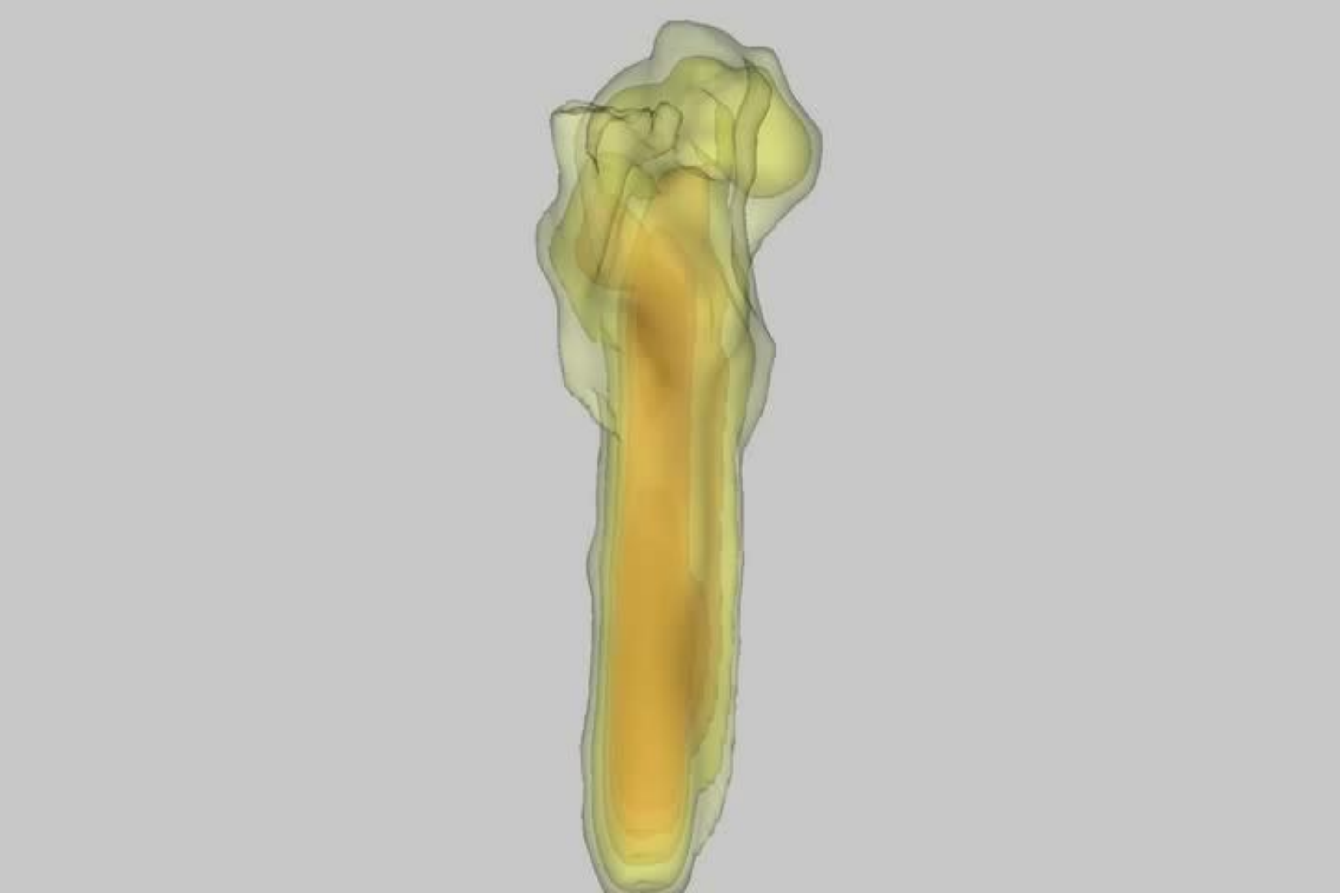
Compute n from definition of Laplacian

$$\nabla \cdot \nabla n = \Delta n$$

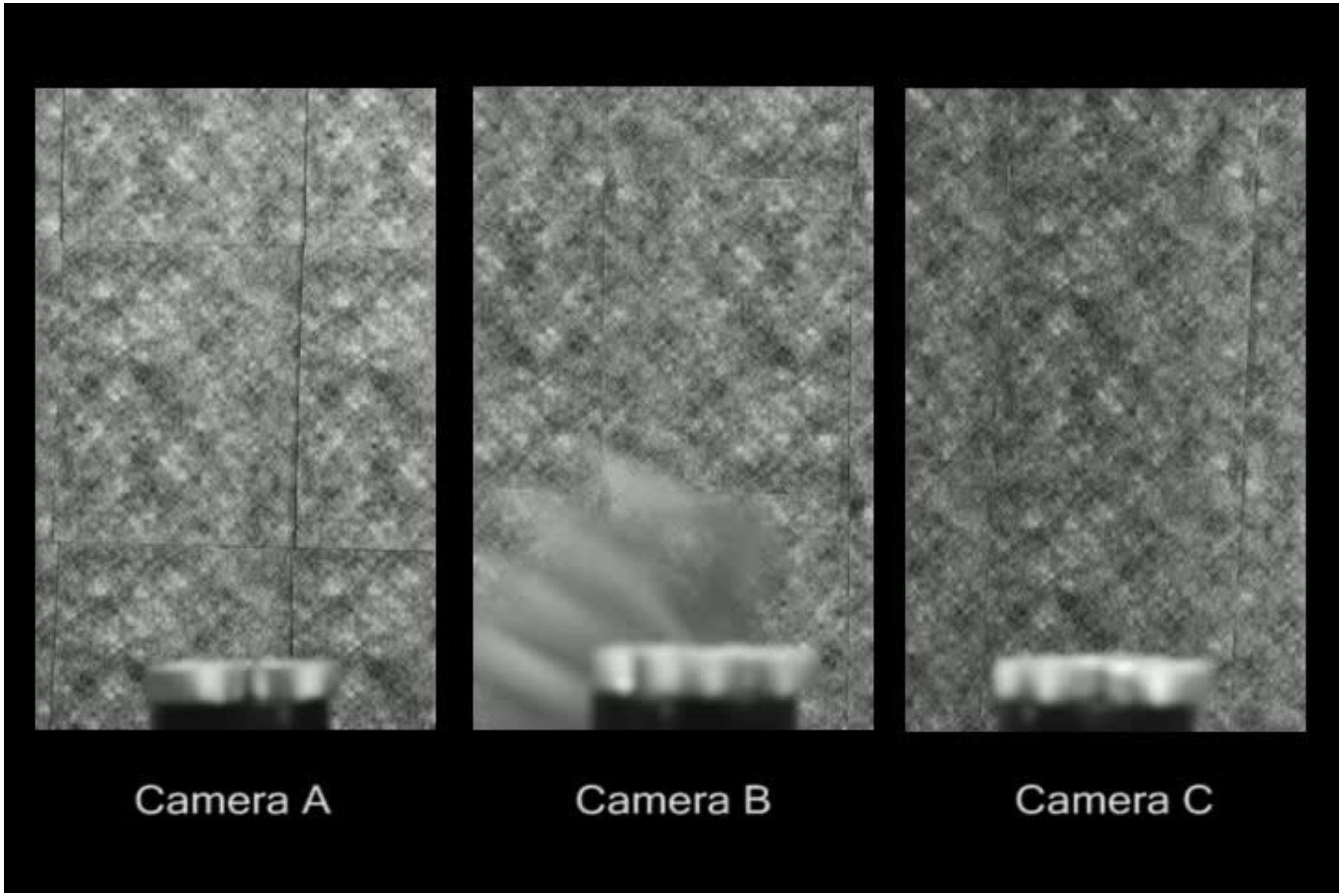
Solve Poisson equation to get refractive index

- Inconsistent gradient field due to noise and other measurement error
- Anisotropic diffusion

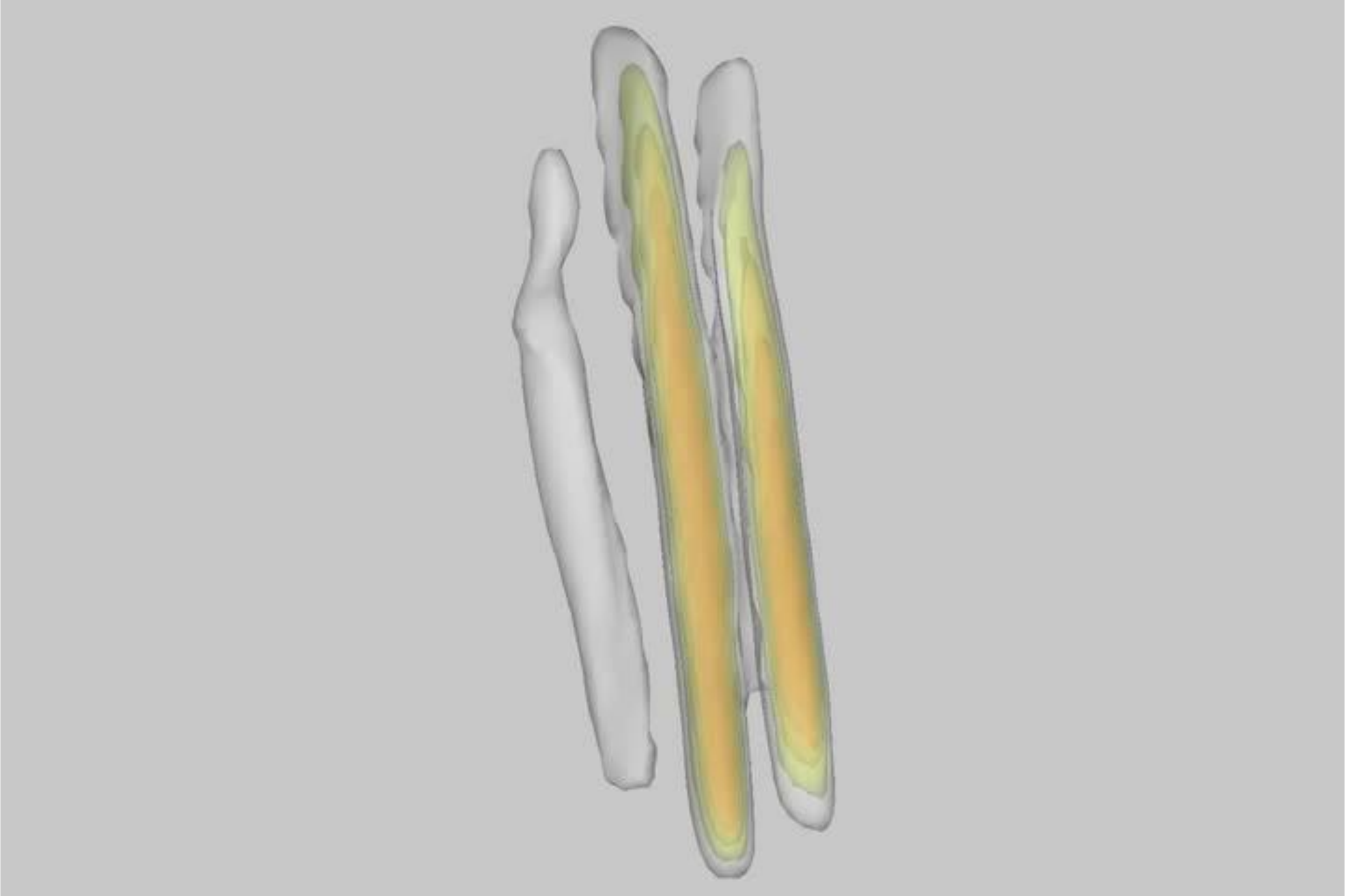
Schlieren Tomography - Results



Schlieren Tomography - Results



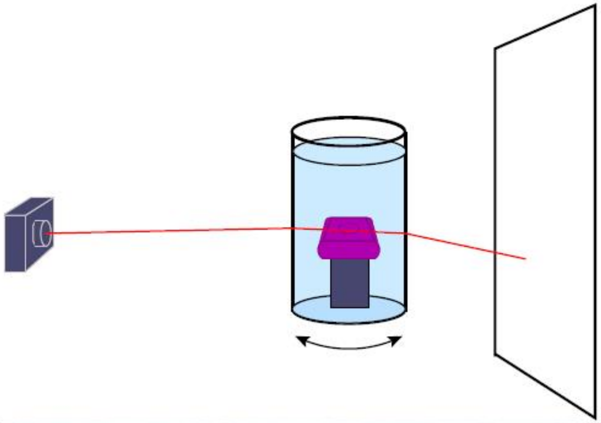
Schlieren Tomography - Results

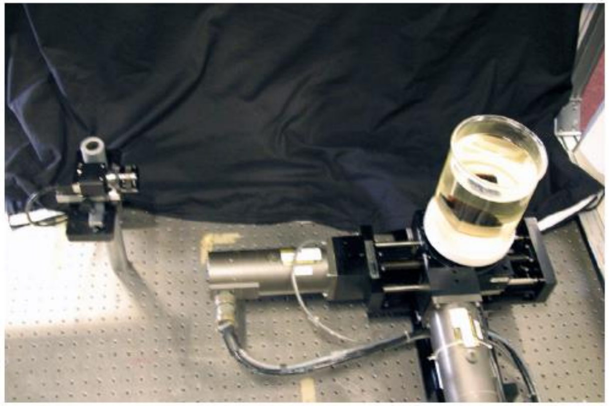


3D Scanning of Glass Objects [Trifonov06]

- visible light tomography of glass objects
  - needs straight ray pathes
- compensate for refraction
  - immerse glass object in water
  - add refractive index matching agent

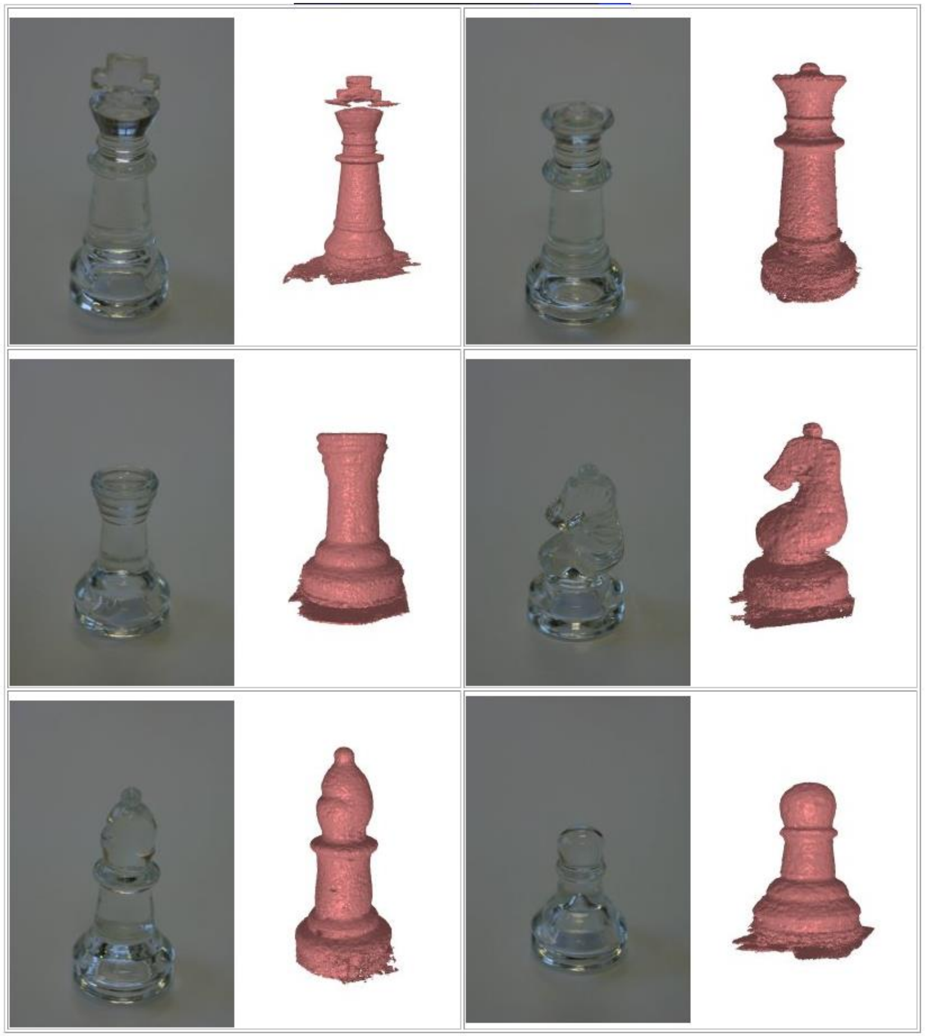
  → "ray straightening"
- apply tomographic reconstruction



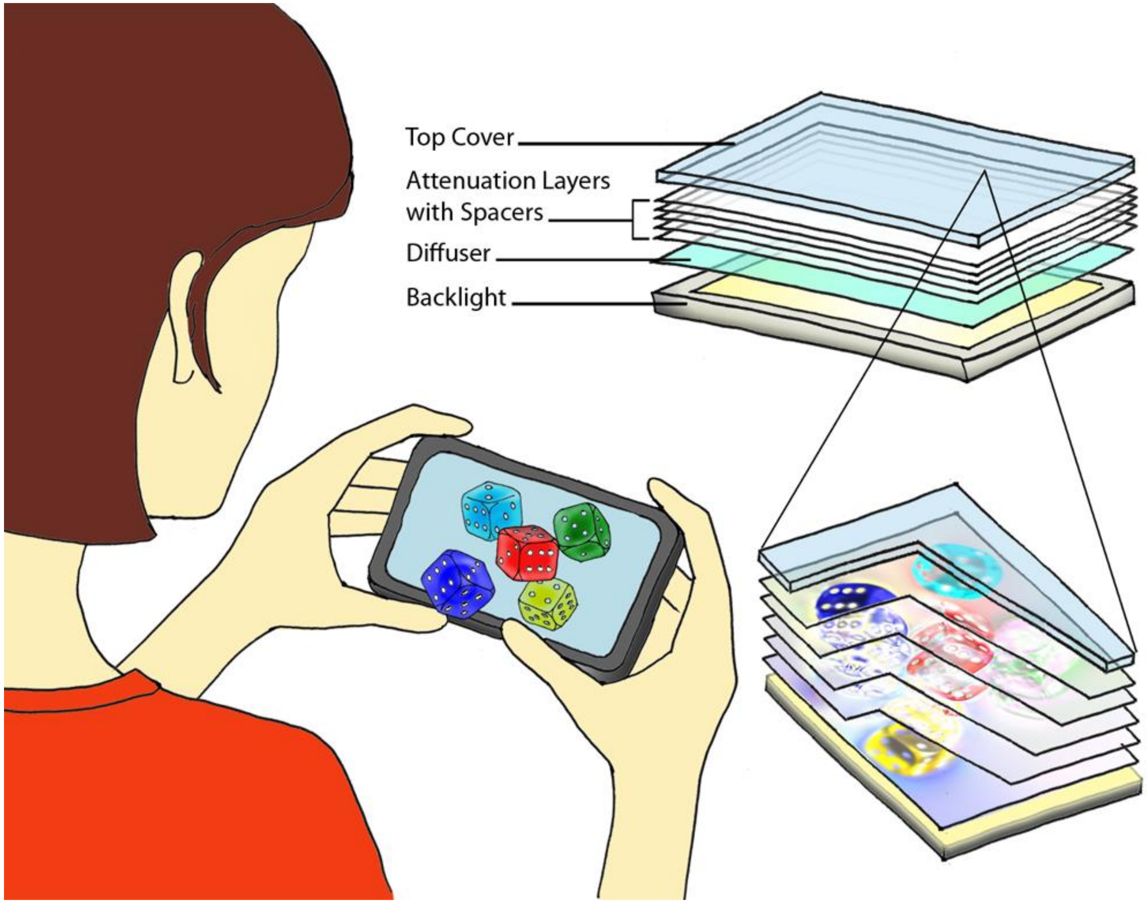


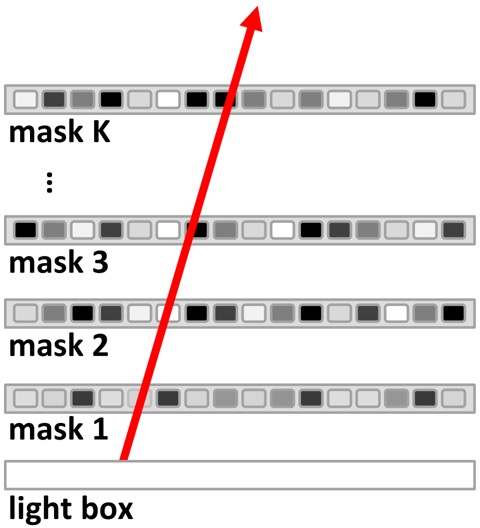
3D Scanning of Glass Objects [Trifonov06]

- Tomographic reconstruction results in volume densities
- use marching cubes to extract object surfaces



Layered 3D: Multi-Layer Displays

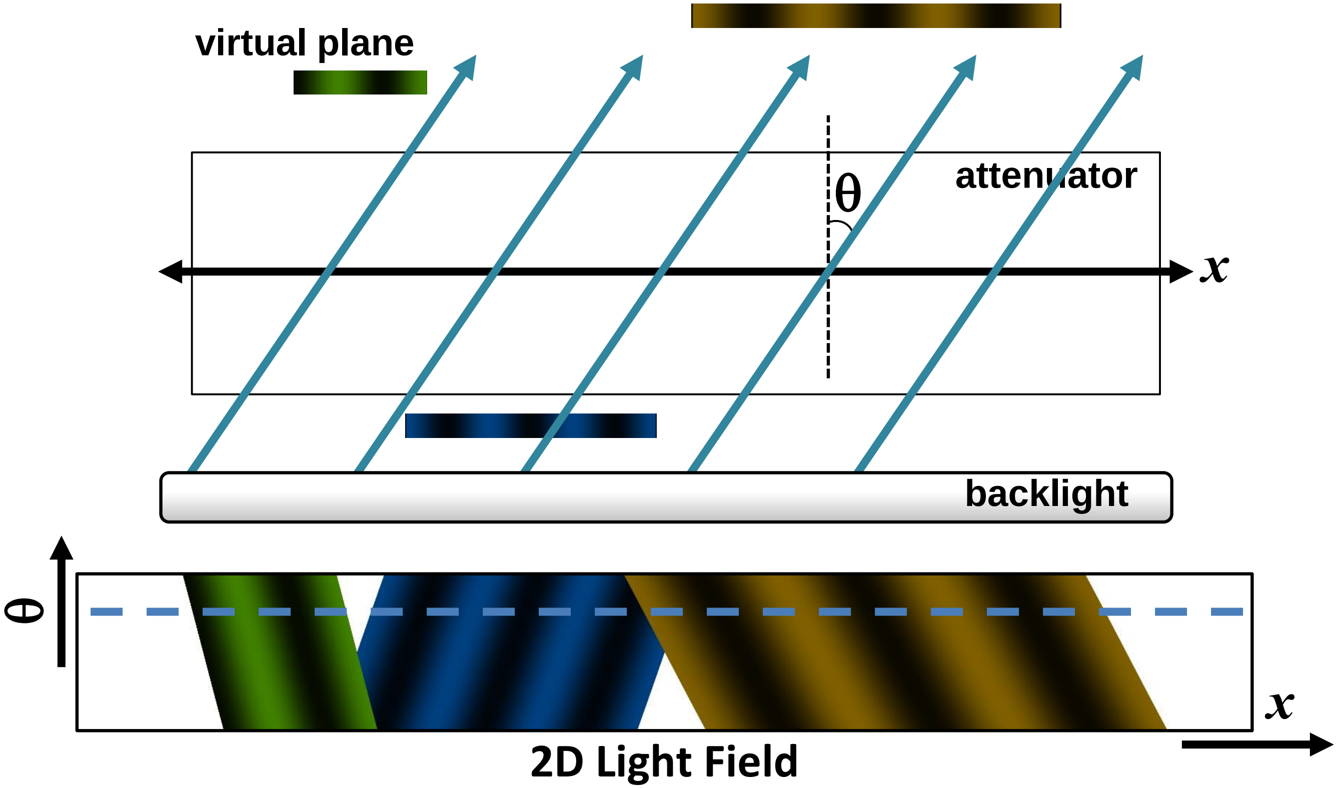




Layered 3D

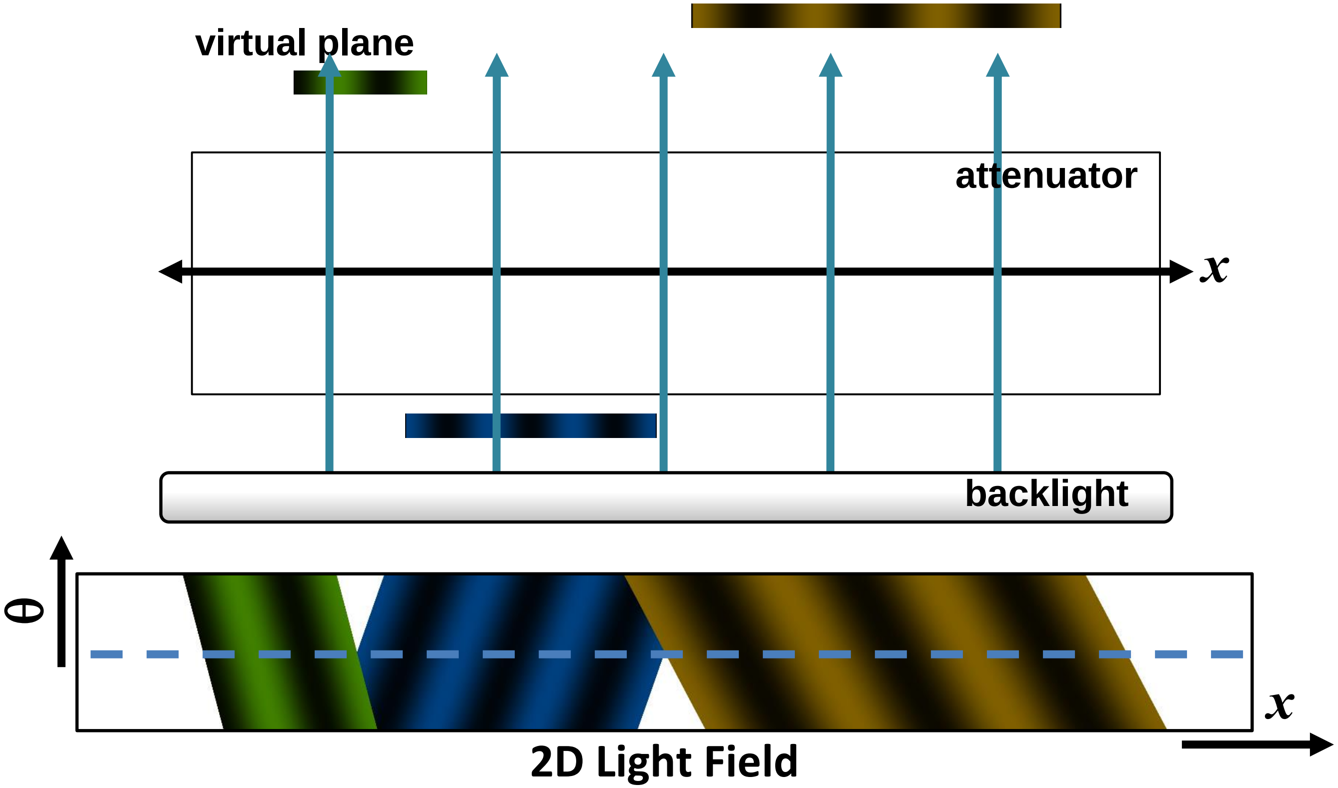
[Wetzstein'11]

Tomographic Light Field Synthesis



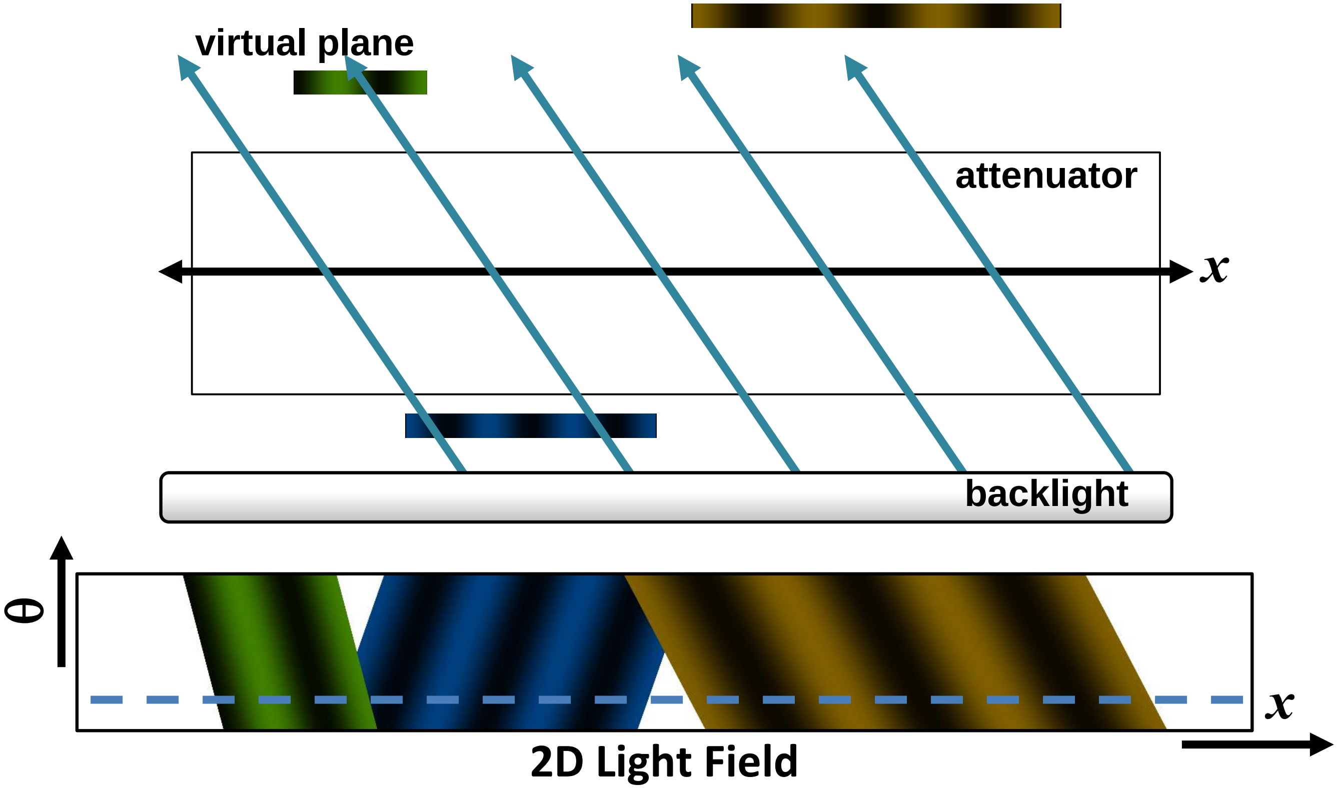
[Wetzstein'11]

Tomographic Light Field Synthesis



[Wetzstein'11]

Tomographic Light Field Synthesis



[Wetzstein'11]

Tomographic Light Field Synthesis

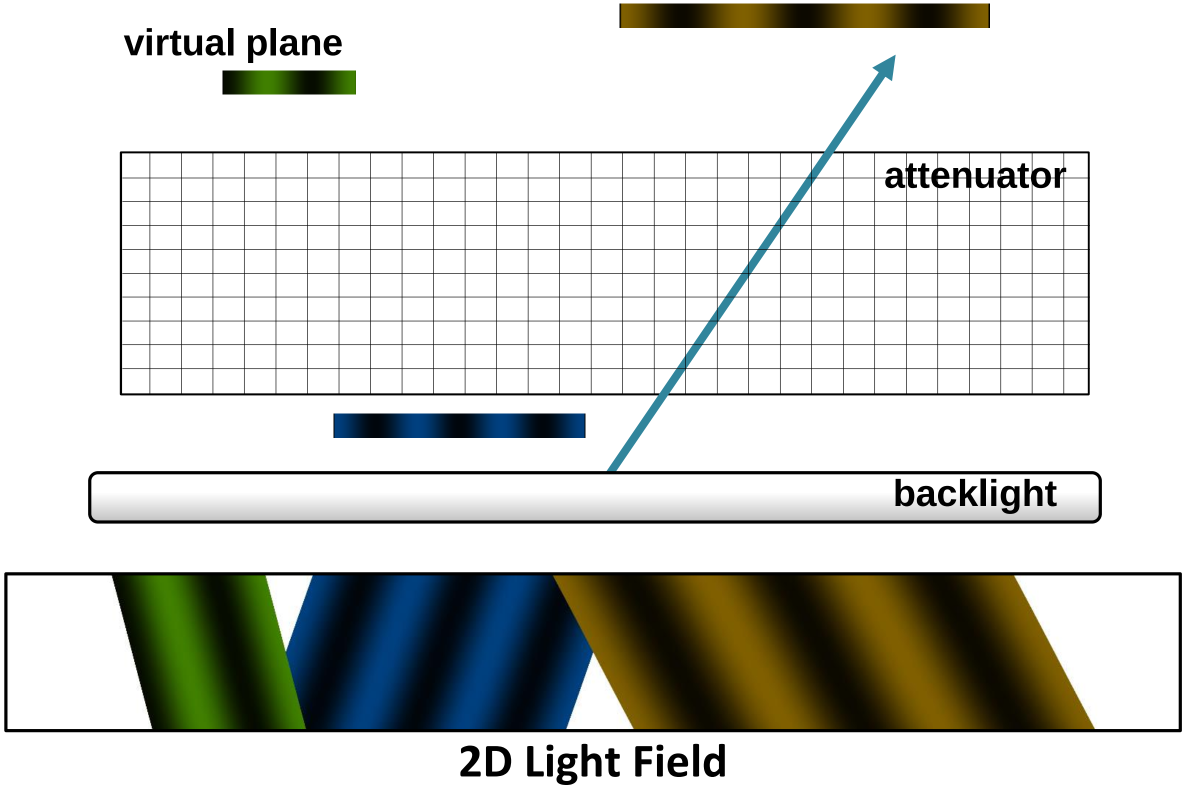


Image formation model:

$$L(x,\theta) = I_0 e^{-\int_C \mu(r)\,dr}$$

$$\bar{L}(x,\theta) = \ln\left(\frac{L(x,\theta)}{I_0}\right) = -\int_C \mu(r)\,dr$$

$$\bar{\mathbf{l}} = -\mathbf{P}\boldsymbol{\alpha}$$

Tomographic synthesis:

$$\arg\min_{\boldsymbol{\alpha}} \left\|\bar{\mathbf{l}} + \mathbf{P}\boldsymbol{\alpha}\right\|^2, \text{ for } \boldsymbol{\alpha} \geq 0$$

[Wetzstein’11]

Tomographic Light Field Synthesis

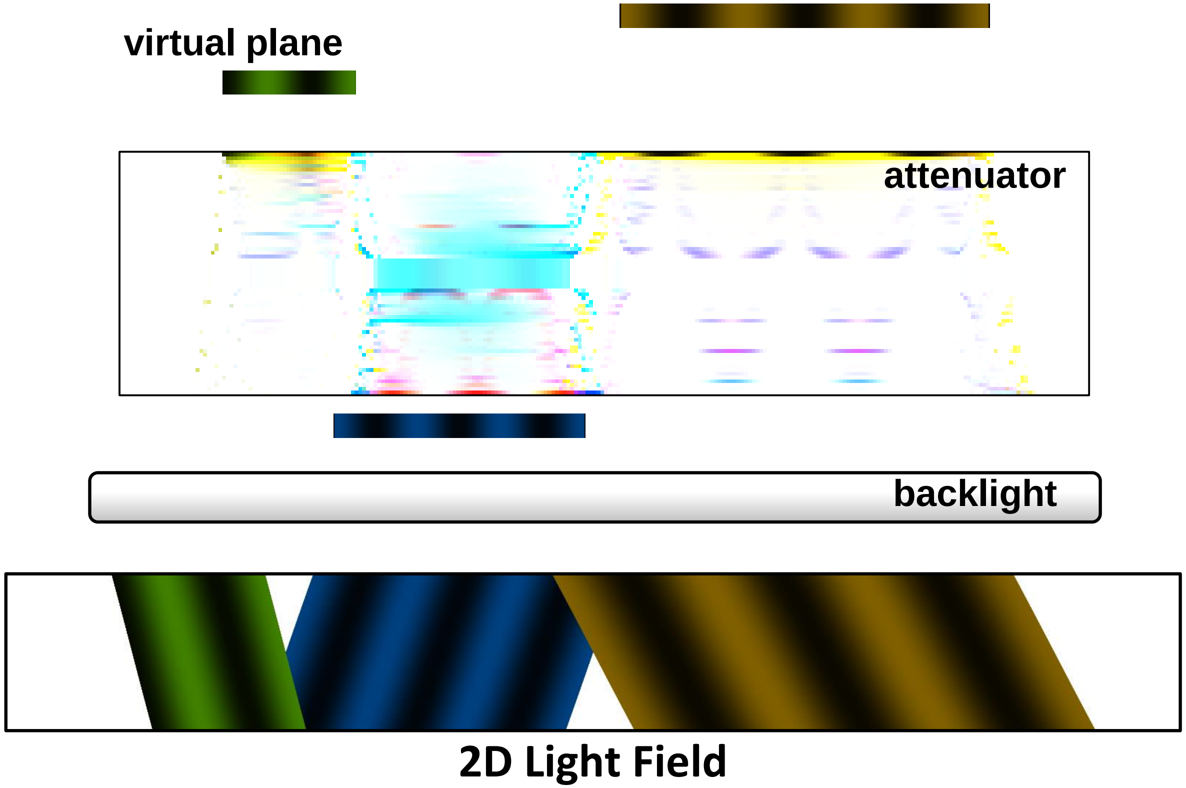


Image formation model:

$$L(x,\theta) = I_0 e^{-\int_C \mu(r)\,dr}$$

$$\bar{L}(x,\theta) = \ln\left(\frac{L(x,\theta)}{I_0}\right) = -\int_C \mu(r)\,dr$$

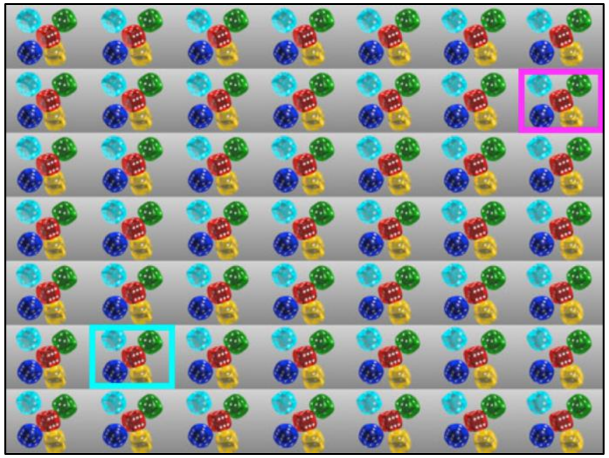
$$\bar{\mathbf{I}} = -\mathbf{P}\boldsymbol{\alpha}$$

Tomographic synthesis:

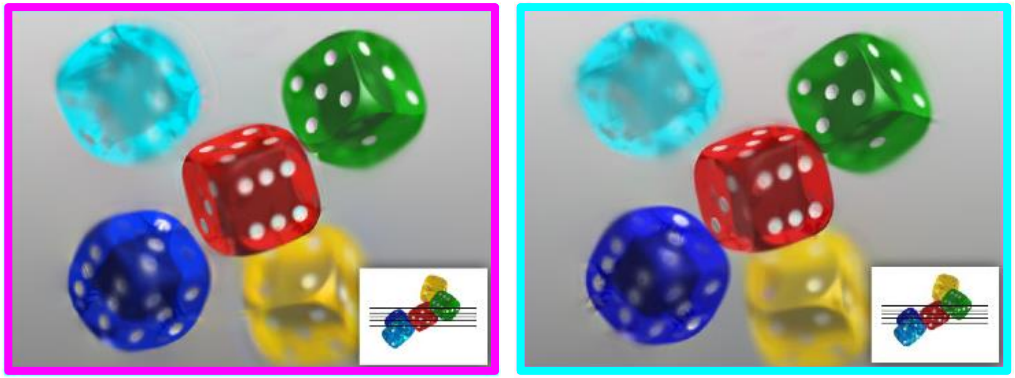
$$\underset{\boldsymbol{\alpha}}{\arg\min} \left\| \bar{\mathbf{I}} + \mathbf{P}\boldsymbol{\alpha} \right\|^2, \text{ for } \boldsymbol{\alpha} \geq 0$$

[Wetzstein’11]

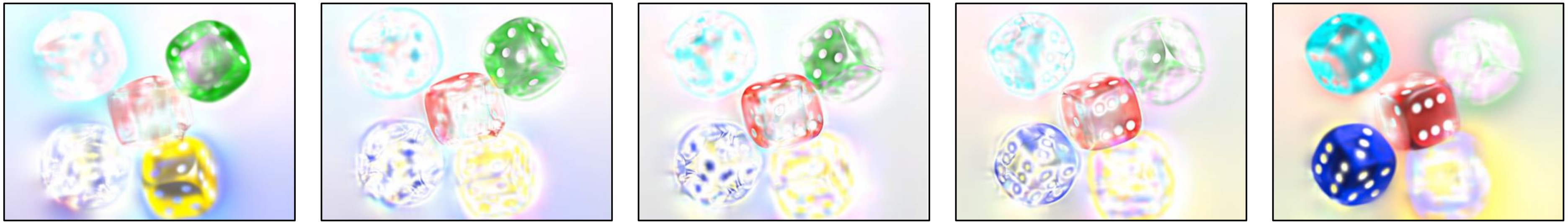
Multi-Layer Light Field Decomposition



Target 4D Light Field



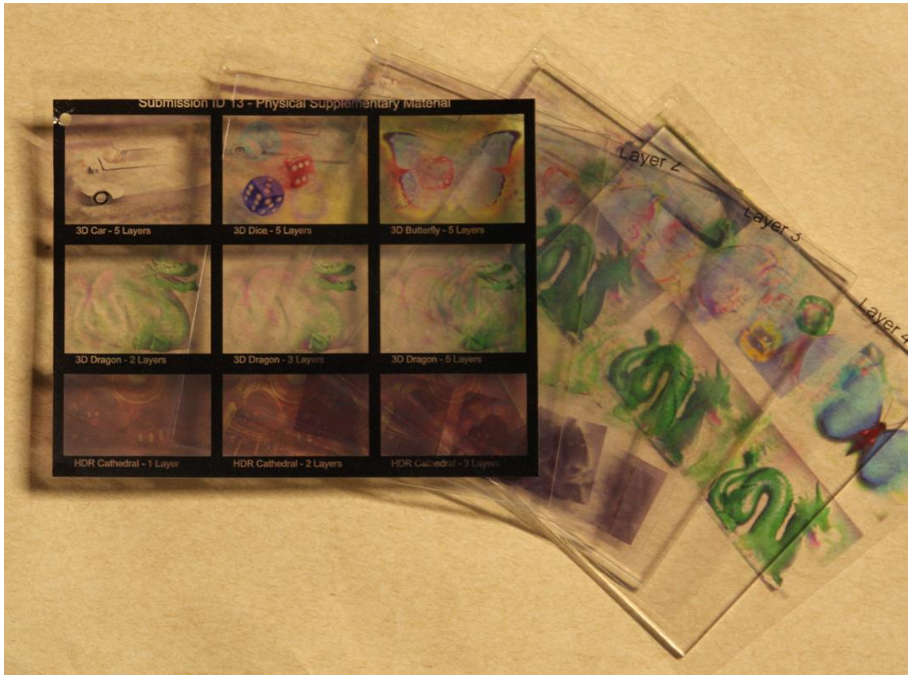
Reconstructed Views



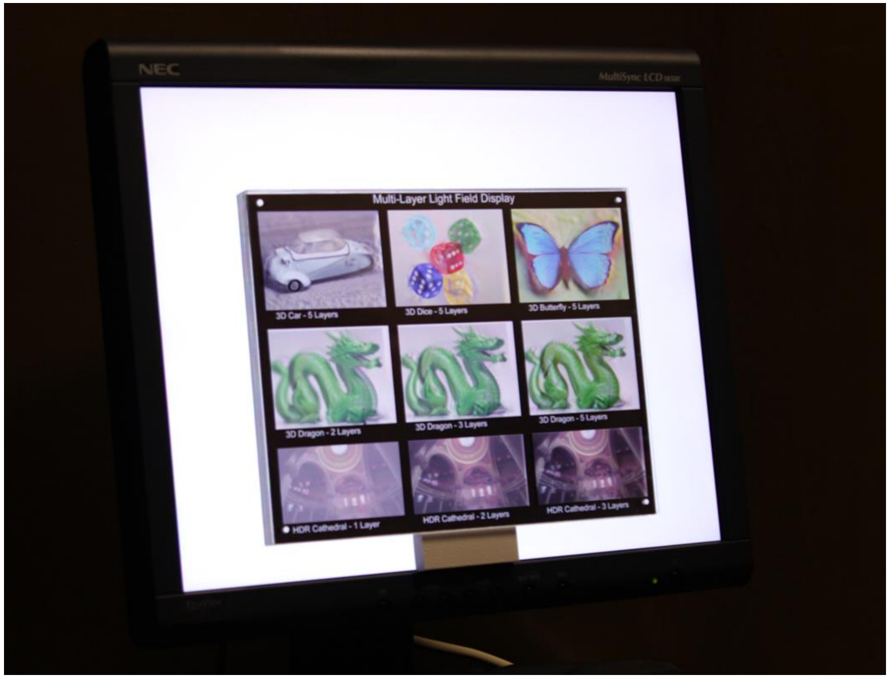
Multi-Layer Decomposition

[Wetzstein'11]

Prototype Layered 3D Display

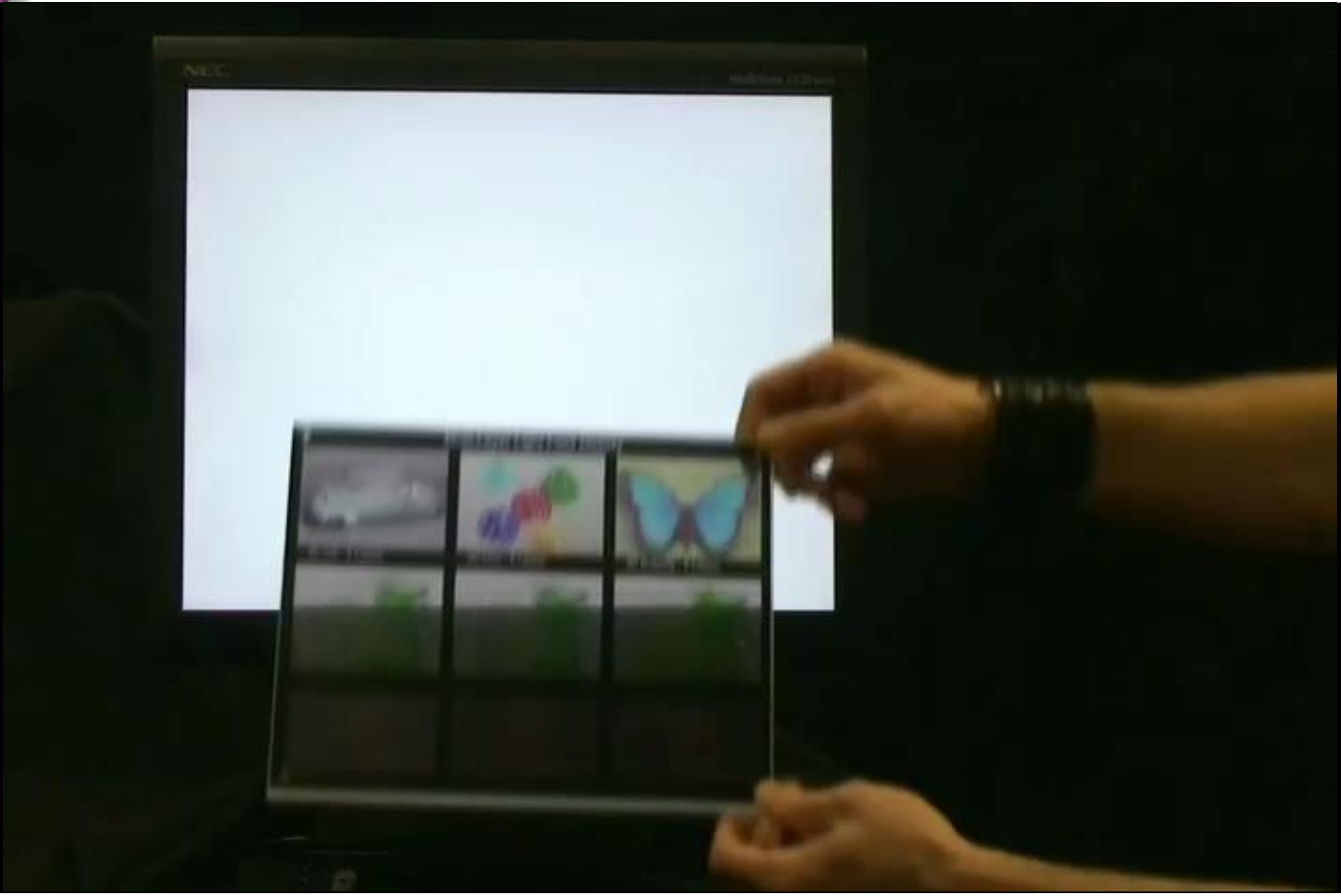


Transparency stack with acrylic spacers



Prototype in front of LCD (backlight source)

[Wetzstein’11]



[Wetzstein'11]

Inverse Problems - Deconvolution

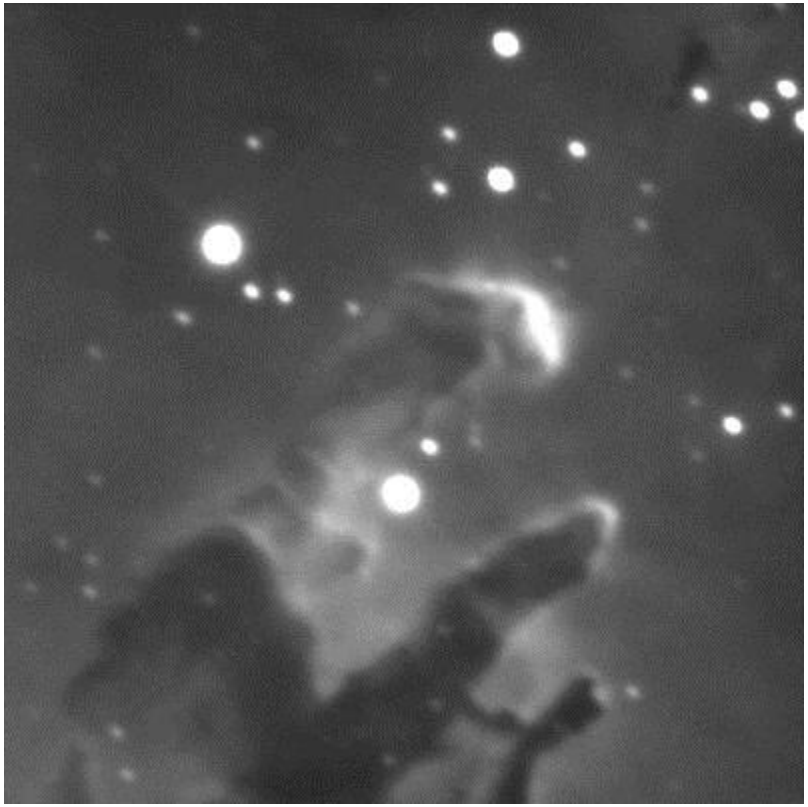
Deconvolution

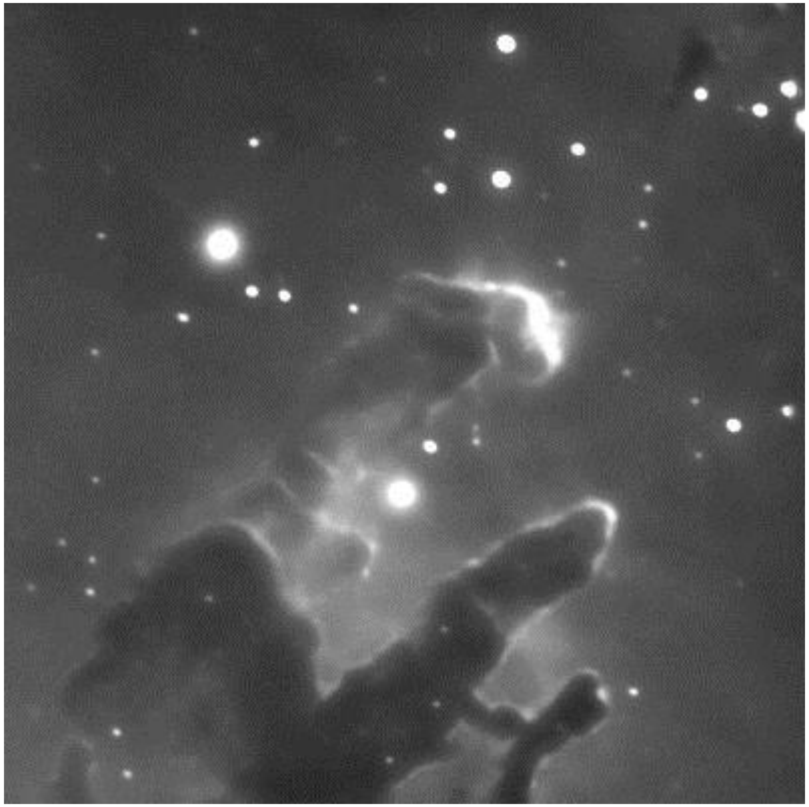
Outline

- Deconvolution Theory
  - example 1D deconvolution
  - Fourier method
  - Algebraic method
    - discretization
    - matrix properties
    - regularization
    - solution methods
- Deconvolution Examples

Applications - Astronomy

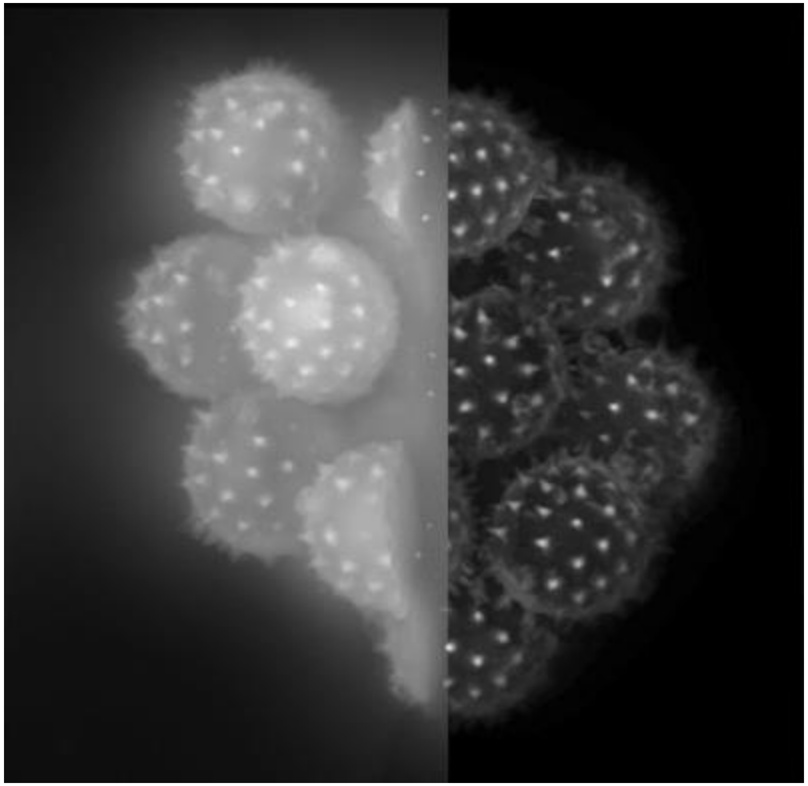
- BEFORE AFTER

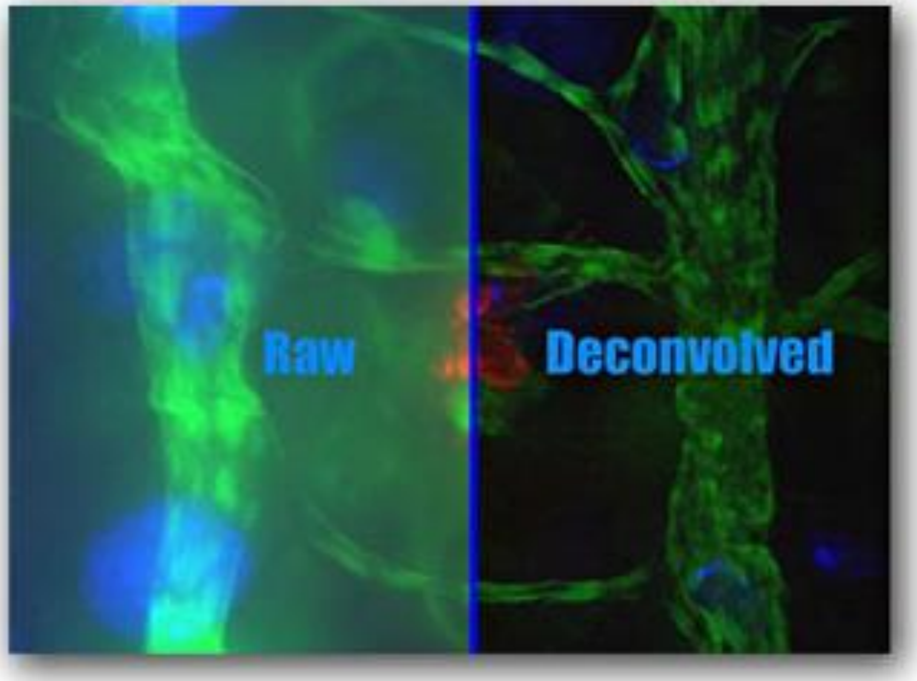




Images courtesy of Robert Vanderbei

Applications - Microscopy





Images courtesy Meyer Instruments

Inverse Problem - Definition

- forward problem
  - given a mathematical model M and its parameters m, compute (predict) observations o

$$o = M(m)$$

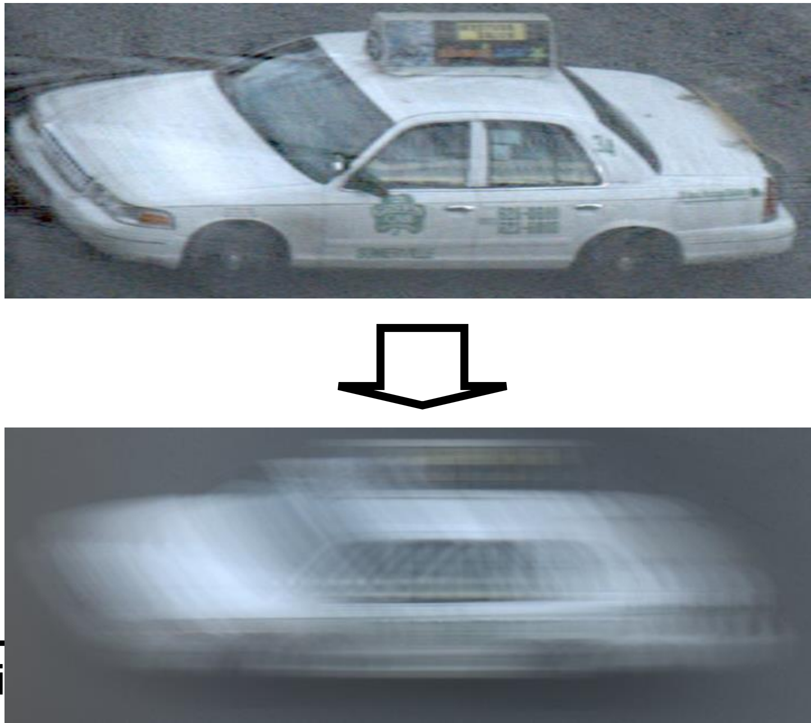
- inverse problem
  - given observations o and a mathematical model M, compute the model's parameters

$$m = M^{-1}(o)$$

Inverse Problems – Example Deconvolution

- forward problem – convolution
  - example blur filter
  - given an image m and a filter kernel k, compute the blurred image o

$$o = m \otimes k$$

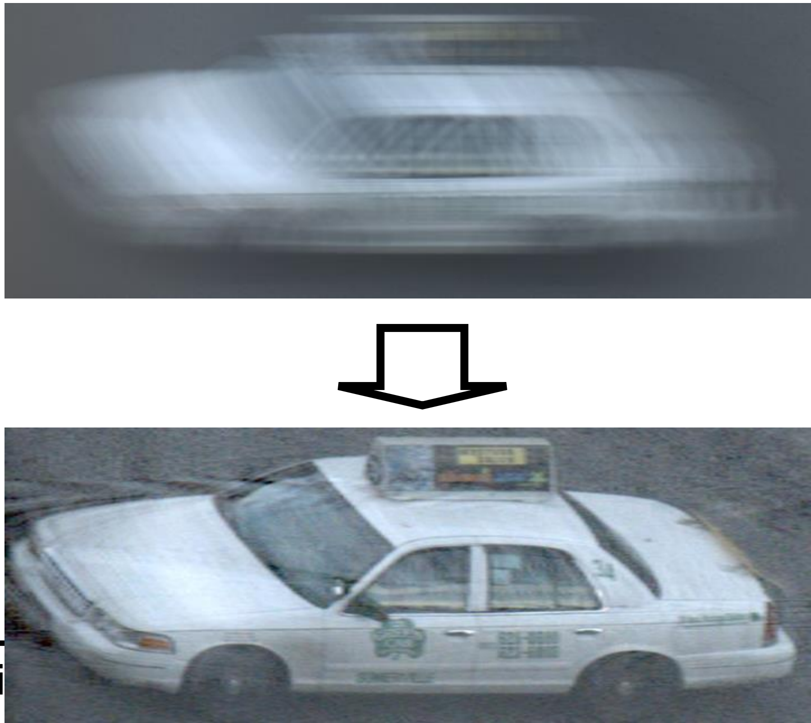


Inverse Problems – Example Deconvolution

- inverse problem – deconvolution
  - example blur filter
  - given a blurred image o and a filter kernel k, compute the sharp image
  - need to invert

$$o = m \otimes k + n$$

  - n is noise



Inverse Problems - Deconvolution

Deconvolution

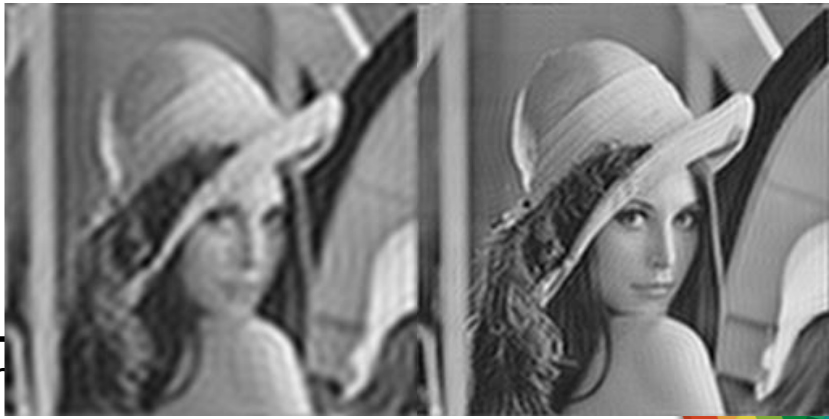
--Fourier Solution--

Deconvolution - Theory

- deconvolution in Fourier space
- convolution theorem (F is the Fourier transform):

$$o = m \otimes k, \quad \Rightarrow \mathcal{F}\{o\} = \mathcal{F}\{m\} \cdot \mathcal{F}\{k\}$$

- deconvolution: $\Rightarrow \mathcal{F}\{m\} = \dfrac{\mathcal{F}\{o\}}{\mathcal{F}\{k\}}$
- problems
  - division by zero
  - Gibbs phenomenon (ringing artifacts)

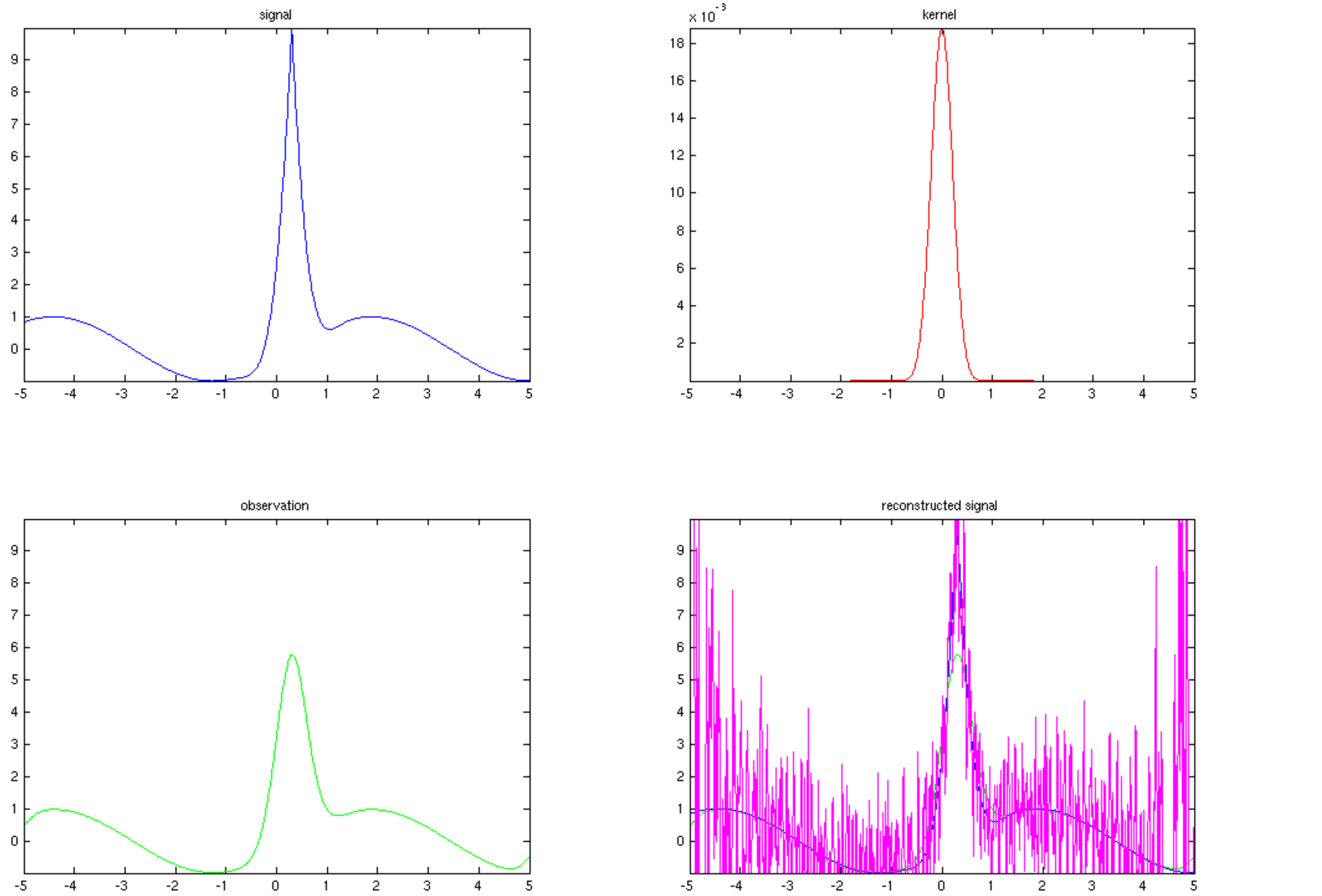


A One-Dimensional Example – Deconvolution Spectral

- most common: $\mathcal{F}\left\{k\right\}$ is a low pass filter
- → $\dfrac{1}{\mathcal{F}\left\{k\right\}}$, the inverse filter, is high pass
- → amplifies noise and numerical errors

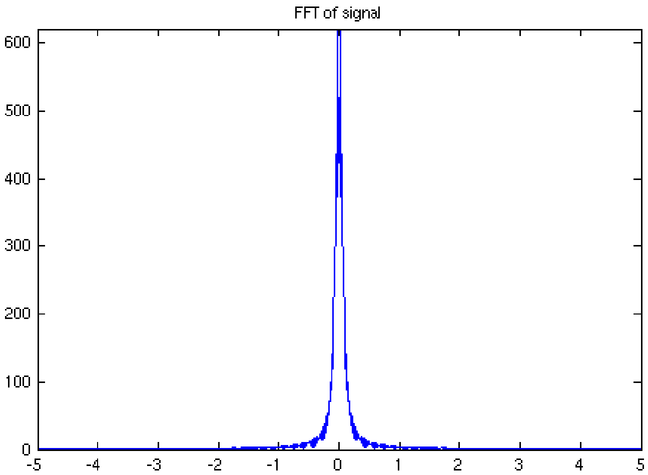
A One-Dimensional Example – Deconvolution Spectral

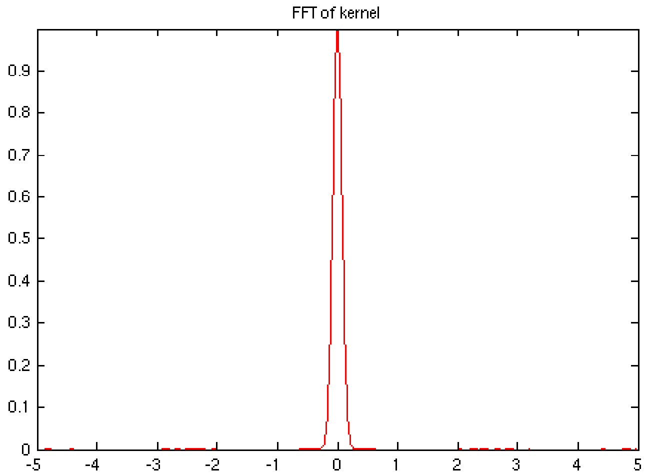
- reconstruction is noisy even if data is perfect !
  - Reason: numerical errors in representation of function

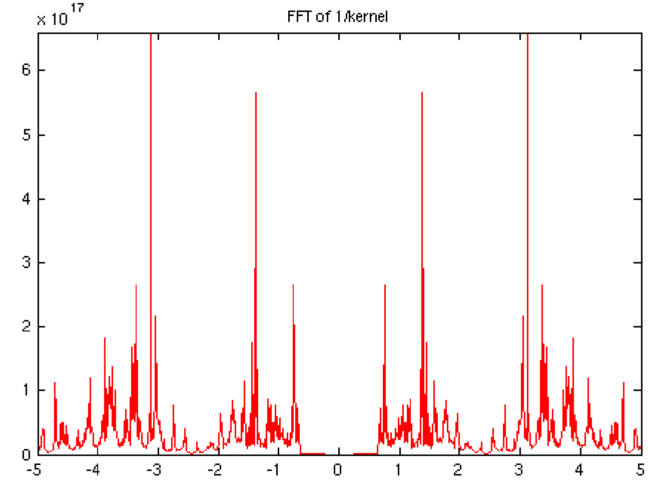


A One-Dimensional Example – Deconvolution Spectral

- spectral view of signal, filter and inverse filter







A One-Dimensional Example – Deconvolution Spectral

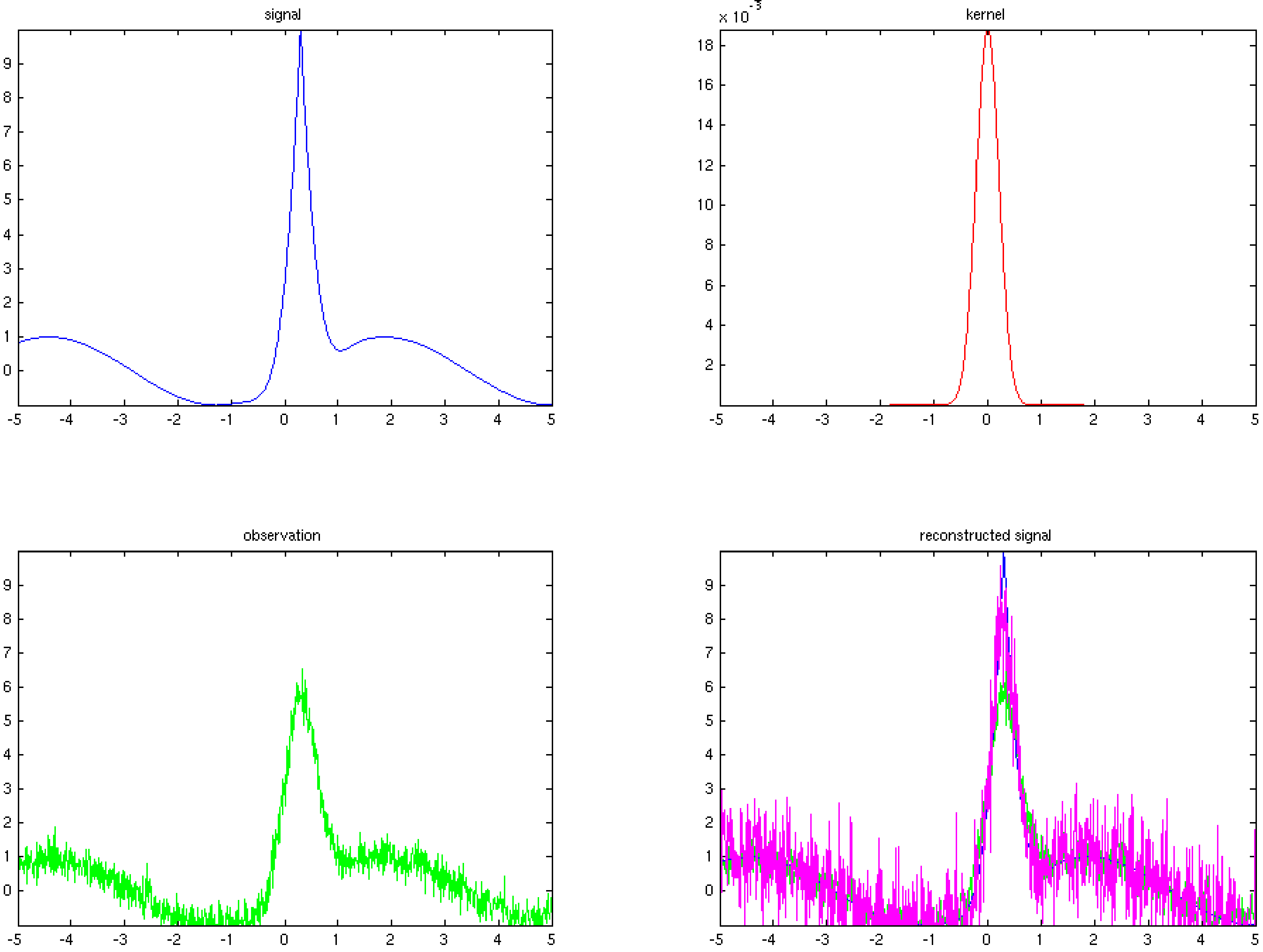
- solution: restrict frequency response of high pass filter (clamping)

$$\mathcal{F}\{g\} := \begin{cases} \frac{1}{\mathcal{F}\{k\}} & \text{if} \frac{1}{\mathcal{F}\{k\}} < \gamma \\ \gamma \frac{\mathcal{F}\{k\}}{|\mathcal{F}\{k\}|} & \text{else} \end{cases}$$

$$\mathcal{F}\{m\} = \mathcal{F}\{o\} \cdot \mathcal{F}\{g\}$$

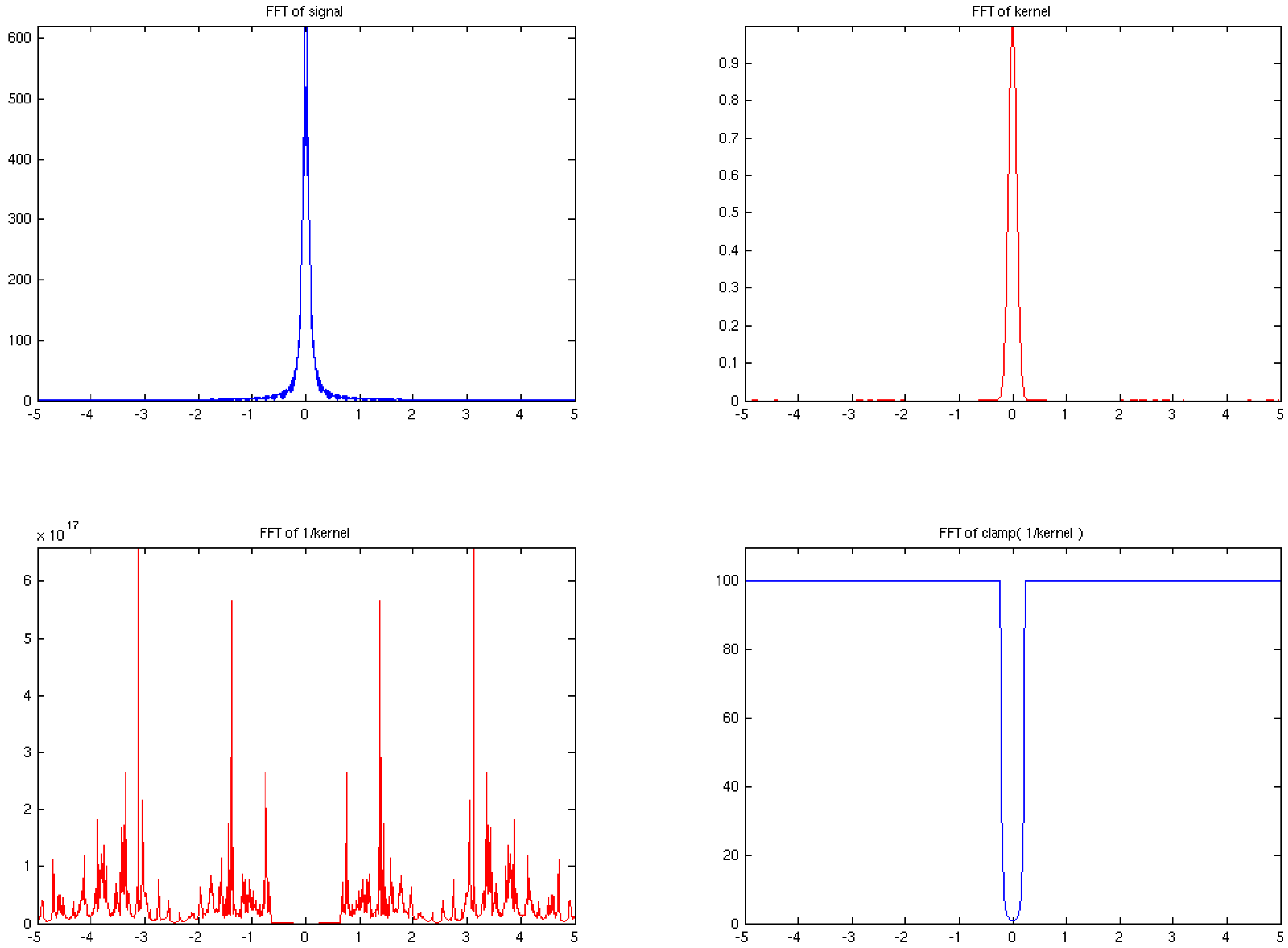
A One-Dimensional Example - Deconvolution Spectral

- reconstruction with clamped inverse filter



A One-Dimensional Example – Deconvolution Spectral

- spectral view of signal, filter and inverse filter



A One-Dimensional Example – Deconvolution Spectral

- Automatic per-frequency tuning: Wiener Deconvolution
  - Alternative definition of inverse kernel
  - Least squares optimal
  - Per-frequency SNR must be known

$$\mathfrak{F}\left\{g\right\}(\omega) \quad := \quad \frac{1}{\mathfrak{F}\left\{k\right\}(\omega)} \frac{|\mathfrak{F}\left\{k\right\}|^2(\omega)}{|\mathfrak{F}\left\{k\right\}|^2(\omega) + |\frac{1}{\mathrm{SNR}(\omega)}|}$$

Deconvolution

-- Algebraic Solution --

A One-Dimensional Example-Deconvolution Algebraic

- alternative: algebraic reconstruction
- convolution

$$o(x) = \int_{-\infty}^{\infty} m(t)k(x-t)dt$$

- discretization: linear combination of basis functions

$$m(t) = \sum_{i=0}^{N} m_i \phi_i(t)$$

A One-Dimensional Example – Deconvolution Algebraic

- discretization:
  - observations are linear combinations of convolved basis functions
  - linear system with unknowns m_i
  - often over-determined, i.e. more observations o than degrees of freedom (# basis functions)

$$
\begin{aligned}
o(x) &= \{m \otimes k\}(x) \\
&= \int_{-\infty}^{\infty} m(t)k(x-t)dt \\
&= \int_{-\infty}^{\infty} \sum_{i=0}^{N} m_i \phi_i(t)k(x-t)dt \\
&= \sum_{i=0}^{N} m_i \int_{-\infty}^{\infty} \phi_i(t)k(x-t)dt \\
&= \sum_{i=0}^{N} m_i \{\phi_i \otimes k\}(x)
\end{aligned}
$$

$$\mathbf{o} = \mathbf{M}\mathbf{m}$$ **linear system**

A One-Dimensional Example – Deconvolution Algebraic

- discretization:
  - observations are linear combinations of convolved basis functions
  - linear system with unknowns m_i
  - often over-determined, i.e. more observations o than degrees of freedom (# basis functions)

$$\begin{aligned}
o(x) &= \{m \otimes k\}(x) \\
&= \int_{-\infty}^{\infty} m(t)k(x-t)dt \\
&= \int_{-\infty}^{\infty} \sum_{i=0}^{N} m_i \phi_i(t) k(x-t)dt \\
&= \sum_{i=0}^{N} m_i \int_{-\infty}^{\infty} \phi_i(t) k(x-t)dt \\
&= \sum_{i=0}^{N} m_i \{\phi_i \otimes k\}(x)
\end{aligned}$$

unknown

$$\mathbf{o} = \mathbf{Mm}$$

linear system

A One-Dimensional Example – Deconvolution Algebraic

- normal equations

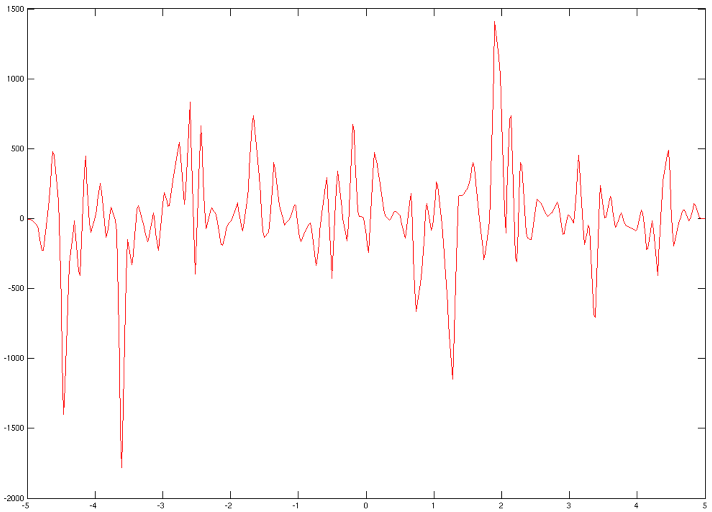
$$\min_{\mathbf{x}} ||\mathtt{A}\mathbf{x} - \mathbf{b}||_2^2 = \min_{\mathbf{x}} (\mathtt{A}\mathbf{x} - \mathbf{b})^T (\mathtt{A}\mathbf{x} - \mathbf{b}) = \min_{\mathbf{x}} f(\mathbf{x})$$

$$\bigtriangledown f = 2\mathtt{A}^T\mathtt{A}\mathbf{x} - 2\mathtt{A}^T\mathbf{b} = \mathbf{0}$$

→ solve $\mathtt{A}^T\mathtt{A}\mathbf{x} = \mathtt{A}^T\mathbf{b}$ to obtain solution in a least squares sense

→ apply to deconvolution

solution is completely broken !



A One-Dimensional Example – Deconvolution Algebraic

- Why ?
- analyze distribution of eigenvalues
- Remember:

$$\det \mathtt{A} = \prod_{i=0}^{N} \lambda_i \quad \text{and} \quad \det \mathtt{A} = 0 \Rightarrow$$ Matrix is under-determined

- we will check the singular values
  - Ok, since $\mathtt{A}^T \mathtt{A}$ is SPD (symmetric, positive semi-definite)

  → non-negative eigenvalues

- Singular values are the square root of the eigenvalues

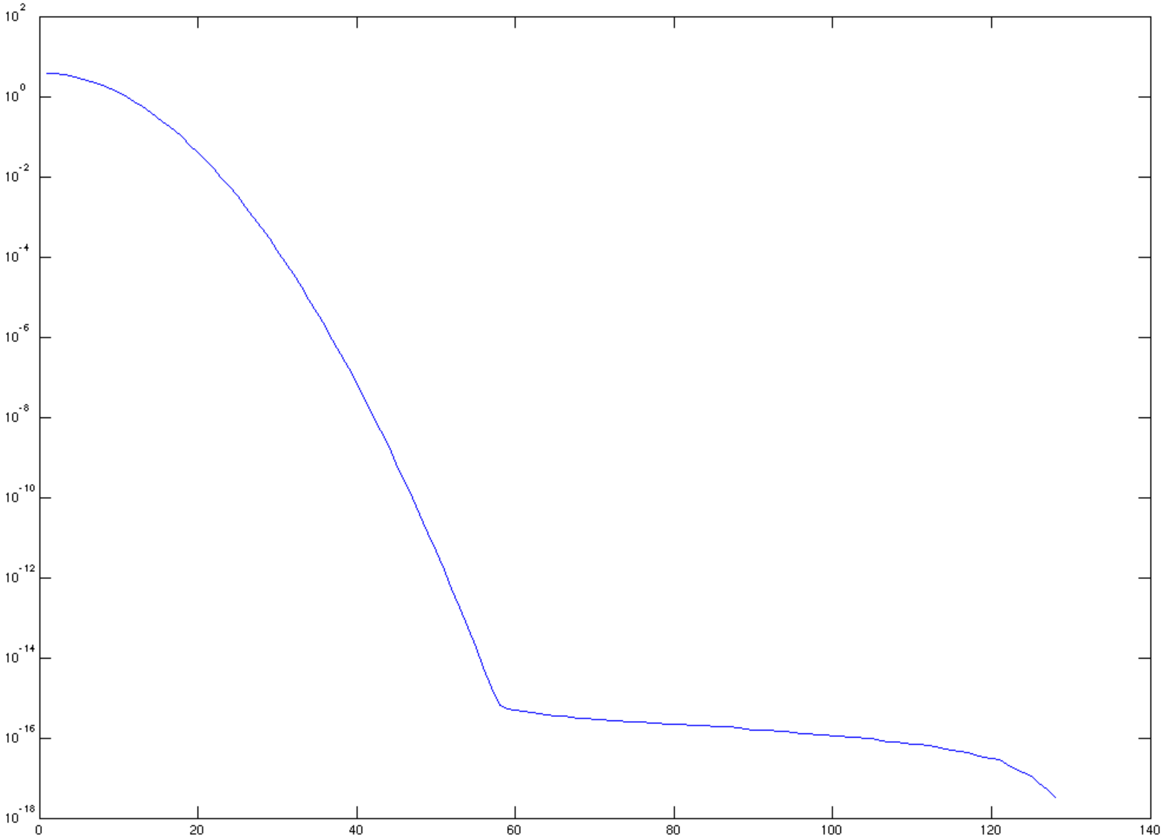
A One-Dimensional Example – Deconvolution Algebraic

- matrix $\mathtt{M}^T\mathtt{M}$ has a very wide range of singular values!
- more than half of the singular values are smaller than machine epsilon (10^{-16}) for double precision

10^2

Log-Plot ! 10^{-8}

10^{-16}



A One-Dimensional Example – Deconvolution Algebraic

- Why is this bad ?
- Singular Value Decomposition: U, V are orthonormal, D is diagonal

$$\mathtt{M} = \mathtt{U}\mathtt{D}\mathtt{V}^T$$

- Inverse of M:

$$\begin{aligned}\mathtt{M}^{-1} &= \left(\mathtt{U}\mathtt{D}\mathtt{V}^T\right)^{-1} \\ &= \mathtt{V}^{-T}\mathtt{D}^{-1}\mathtt{U}^{-1} \\ &= \mathtt{V}\mathtt{D}^{-1}\mathtt{U}^T\end{aligned}$$

- singular values are diagonal elements of D
- inversion:

$$\mathtt{D}^{-1} = \mathrm{diag}\left(\frac{1}{\mathtt{D}_{i,i}}\right)$$

A One-Dimensional Example – Deconvolution Algebraic

- computing model parameters from observations:

$$\mathbf{m} = \mathtt{M}^{-1}\mathbf{o} = \mathtt{V}\mathtt{D}^{-1}\mathtt{U}^T\mathbf{o}$$

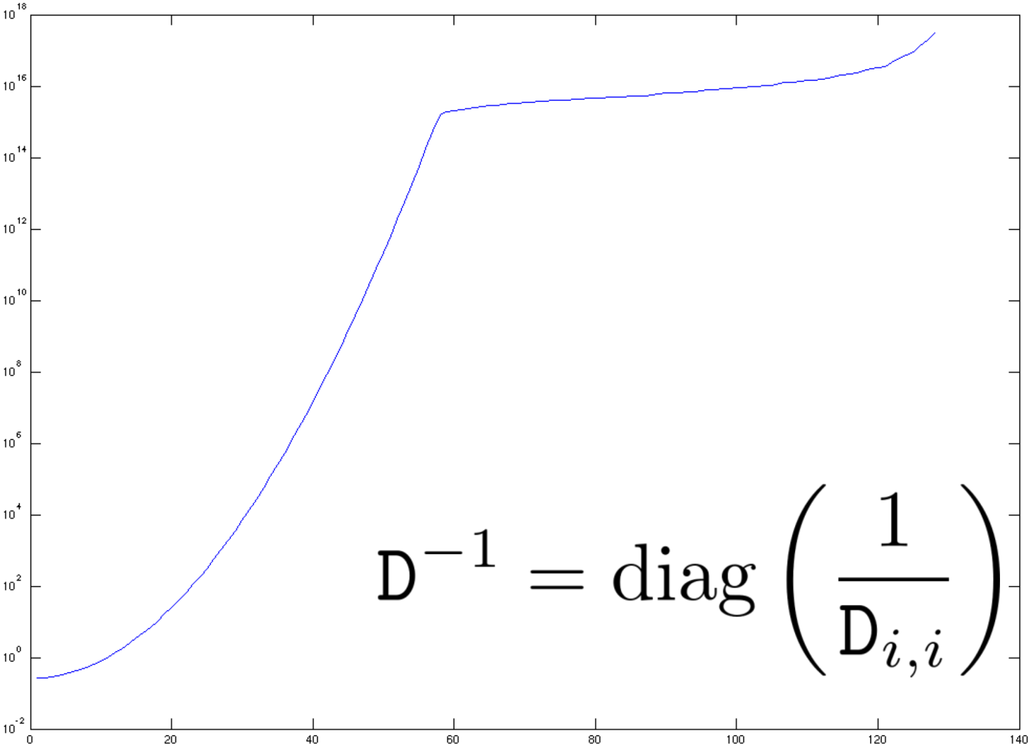
- again: amplification of noise
- potential division by zero

Log-Plot !

10^{18}

10^{8}

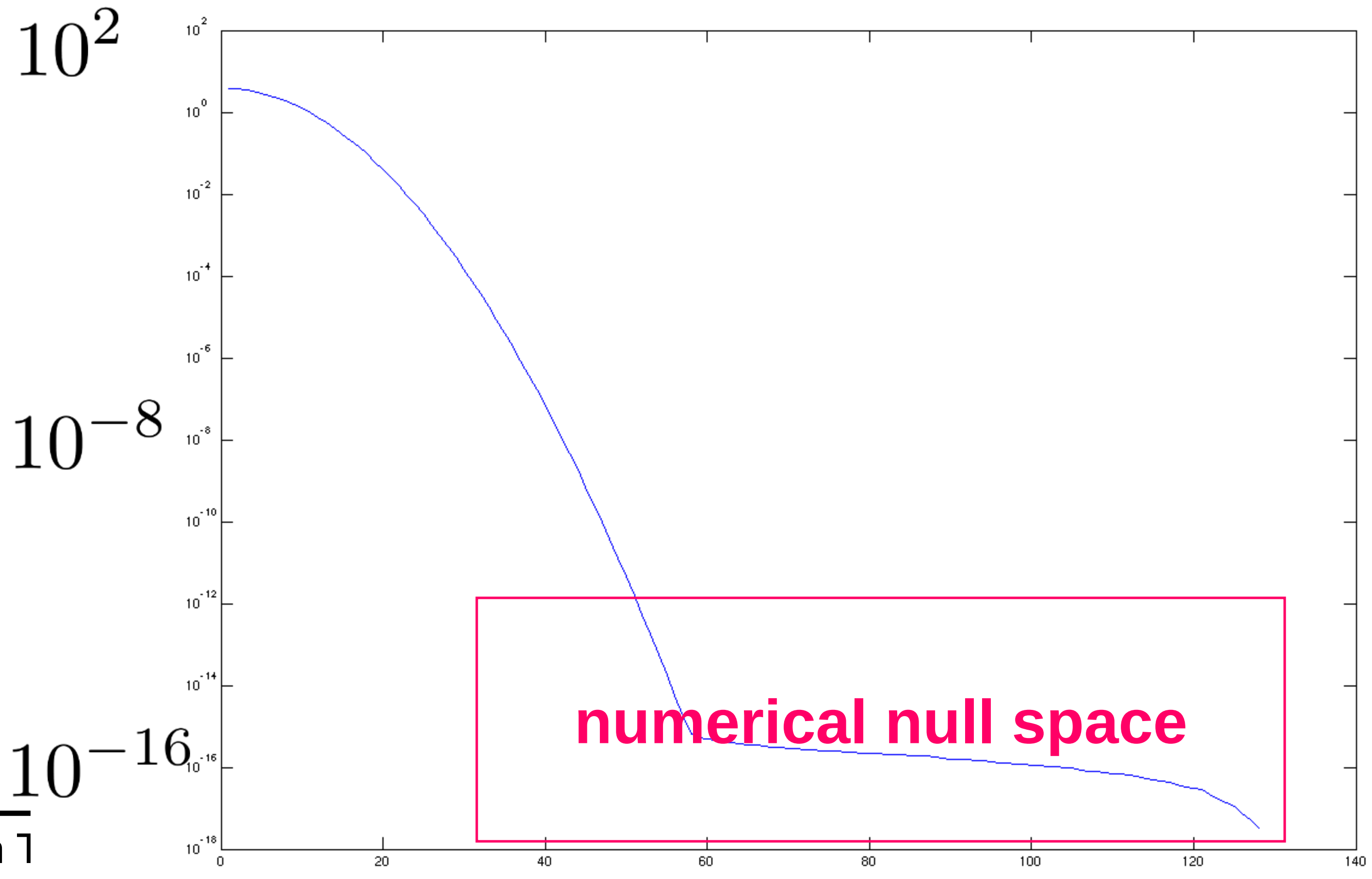
10^{0}



$$\mathtt{D}^{-1} = \mathrm{diag}\left(\frac{1}{\mathtt{D}_{i,i}}\right)$$

A One-Dimensional Example – Deconvolution Algebraic

- inverse problems are often ill-conditioned (have a numerical null-space)
- inversion causes amplification of noise



Well-Posed and Ill-Posed Problems

- Definition [Hadamard1902]
  - a problem is well-posed if
    1. a solution exists
    2. the solution is unique
    3. the solution continually depends on the data

Well-Posed and Ill-Posed Problems

- Definition [Hadamard1902]
  - a problem is ill-posed if it is not well-posed
    - most often condition (3) is violated
    - if model has a (numerical) null space, parameter choice influences the data in the null-space of the data very slightly, if at all
    - noise takes over and is amplified when inverting the model

Condition Number

- measure of ill-conditionedness: condition number
- measure of stability for numerical inversion
- ratio between largest and smallest singular value

$$\rho(\mathbf{A}) = \frac{\sigma_0}{\sigma_N}, \quad \sigma_0 > \ldots > \sigma_N \text{ are the singular values of } \mathbf{A}$$

- smaller condition number → less problems when inverting linear system
- condition number close to one implies near orthogonal matrix

Truncated Singular Value Decomposition

- solution to stability problems: avoid dividing by values close to zero
- Truncated Singular Value Decomposition (TSVD)

$$\mathbf{d}^{+} = \begin{cases} \frac{1}{\mathtt{D}_{i,i}} & \text{if } \mathtt{D}_{i,i} > \epsilon \\ 0 & \text{else} \end{cases}$$

$$\mathtt{D}^{+} = \operatorname{diag}\left(\mathbf{d}^{+}\right)$$

$$\mathtt{M}^{+} = \mathtt{V}\mathtt{D}^{+}\mathtt{U}^{T}$$

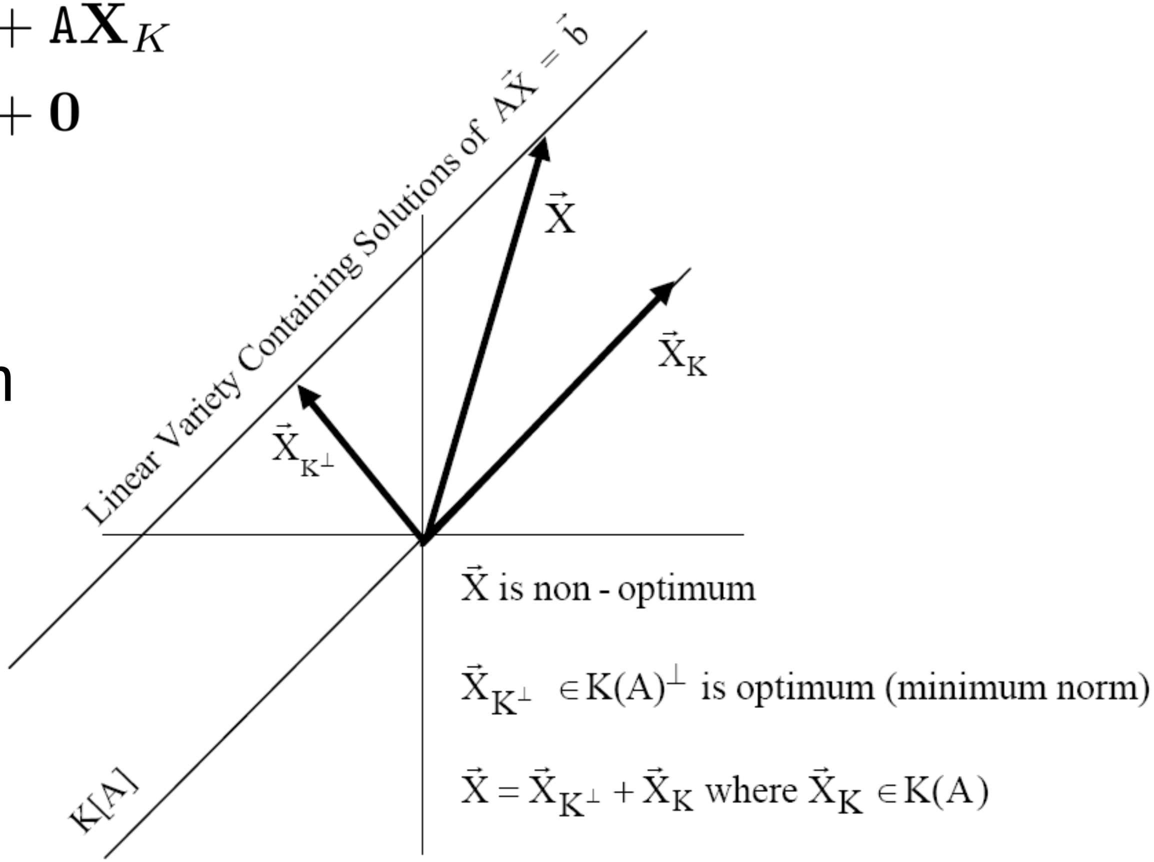
- ϵ is called the *regularization* parameter

Minimum Norm Solution

- Let $K[\mathtt{A}]$ be the null-space of A and $\mathbf{X}_K \in K$

$$\begin{aligned}
\Rightarrow \mathtt{A}\mathbf{X}_K &= \mathbf{0} \\
\Rightarrow \mathtt{A}\mathbf{X} &= \mathtt{A}(\mathbf{X}_{K^\perp} + \mathbf{X}_K) \\
&= \mathtt{A}\mathbf{X}_{K^\perp} + \mathtt{A}\mathbf{X}_K \\
&= \mathtt{A}\mathbf{X}_{K^\perp} + \mathbf{0} \\
&= \mathtt{A}\mathbf{X}_{K^\perp} \\
&= \mathbf{b}
\end{aligned}$$

- $\mathbf{X}_{K^\perp}$ is the minimum norm solution



$\vec{X}$ is non - optimum

$\vec{X}_{K^\perp} \in K(A)^\perp$ is optimum (minimum norm)

$\vec{X} = \vec{X}_{K^\perp} + \vec{X}_K$ where $\vec{X}_K \in K(A)$

Regularization

- countering the effect of ill-conditioned problems is called regularization

- an ill-conditioned problem behaves like a singular (i.e. under-constrained) system

- family of solutions exist

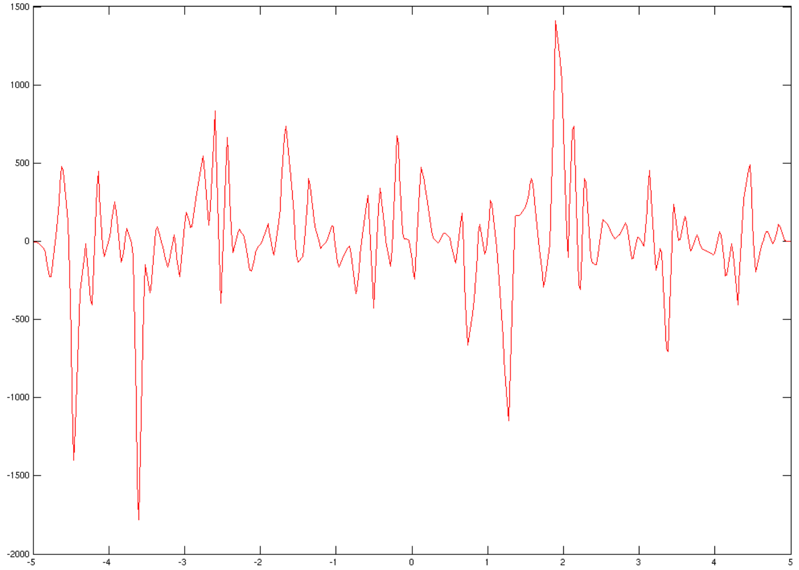
→impose additional knowledge to pick a favorable solution

- TSVD results in minimum norm solution

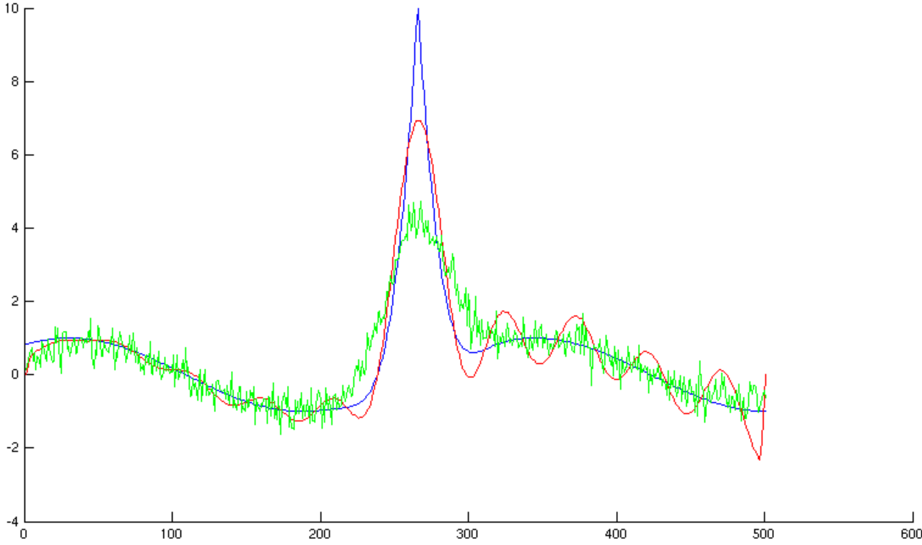
Example – 1D Deconvolution

- back to our example – apply TSVD
- solution is much smoother than Fourier deconvolution

unregularized solution



TSVD regularized solution $\epsilon = 10^{-6}$



Large Scale Problems

- consider 2D deconvolution
- 512x512 image, 256x256 basis functions

→ least squares problem results in matrix that is 65536x65536 !

- even worse in 3D (millions of unknowns)
- problem: SVD is $\mathcal{O}\left(N^3\right)$

system size	512	1024	2048	4096
SVD time (in s)	0.27	1.75	12.54	96.28

Intel Xeon 2-core (E5503) @ 2GHz (introduced 2010)

- today impractical to compute for systems larger than $> 16384^2$ (takes a couple of hours)
- Question: How to compute regularized solutions for large scale systems ?

Explicit Regularization

- Answer: modify original problem to include additional optimization goals (e.g. small norm solutions)

$$\begin{aligned}
\min_x \quad & \alpha||\mathtt{A}\mathbf{x}-\mathbf{b}||_2^2+(1-\alpha)||\mathtt{R}\mathbf{x}||_2^2 = \\
\min_x \quad & \alpha\left(\mathtt{A}\mathbf{x}-\mathbf{b}\right)^T\left(\mathtt{A}\mathbf{x}-\mathbf{b}\right)+(1-\alpha)\mathbf{x}^T\mathtt{R}^T\mathtt{R}\mathbf{x} = \\
\min_x \quad & \hat{f}(\mathbf{x})
\end{aligned}$$

- minimize modified quadratic form

$$\bigtriangledown\hat{f}(\mathbf{x}) = 2\alpha\mathtt{A}^T\mathtt{A}\mathbf{x}-2\mathtt{A}^T\mathbf{b}+2(1-\alpha)\mathtt{R}^T\mathtt{R}\mathbf{x}=\mathbf{0}$$

- regularized normal equations:

$$(\alpha\mathtt{A}^T\mathtt{A}\mathbf{x}+(1-\alpha)\mathtt{R}^T\mathtt{R})\mathbf{x}=\mathtt{A}^T\mathbf{b}$$

Modified Normal Equations

- include data term, smoothness term and blending parameter

data **Prior information (popular: smoothness)**

$$(\alpha \mathtt{A}^T \mathtt{A}\mathbf{x} + (1-\alpha)\mathtt{R}^T\mathtt{R})\mathbf{x} = \mathtt{A}^T\mathbf{b}$$

blending (regularization) parameter

Tikhonov Regularization

- setting $\mathtt{R} = \mathbb{1}$ and $\lambda = \frac{1-\alpha}{\alpha}$ we have a quadratic optimization problem with data fitting *and* minimum norm terms

data fitting **minimum norm**

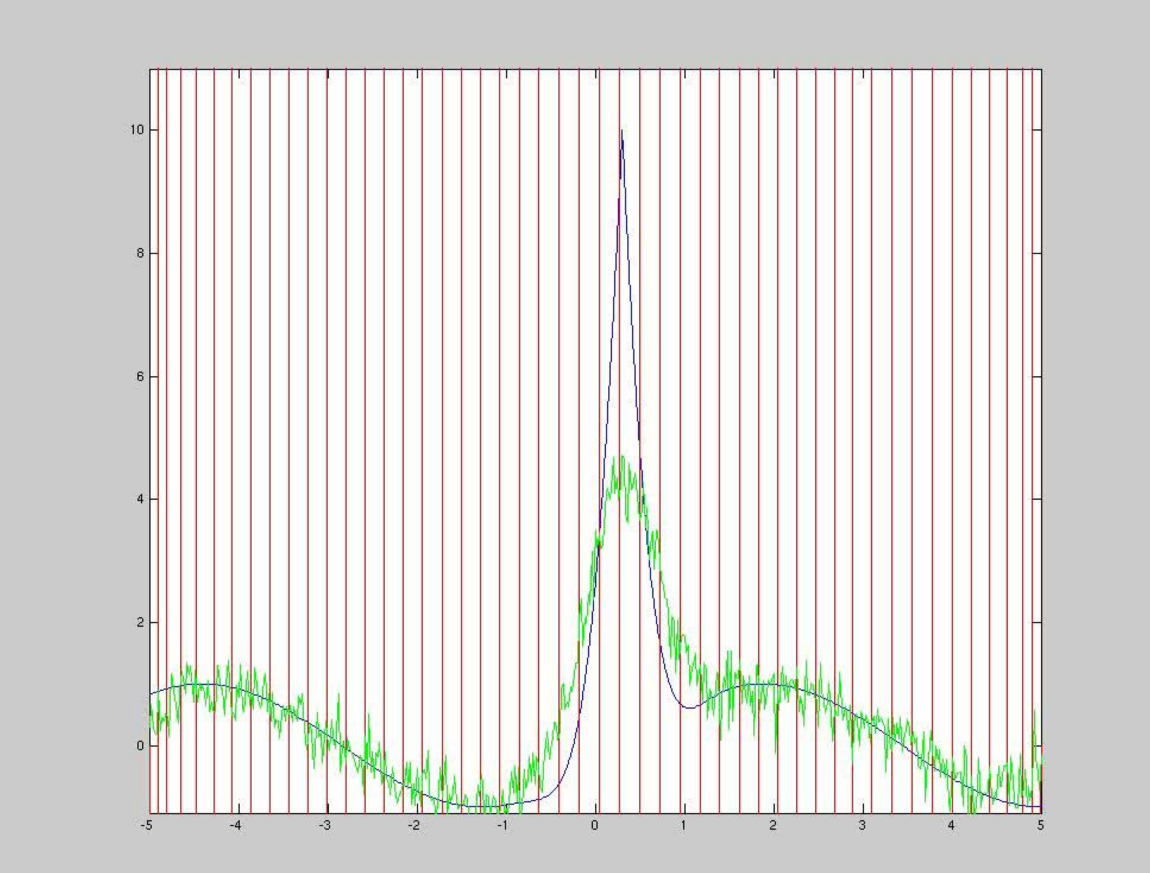
$$\min_{\mathbf{x}} (\mathtt{A}\mathbf{x} - \mathbf{b})^T (\mathtt{A}\mathbf{x} - \mathbf{b}) + \underset{\uparrow}{\lambda} \mathbf{x}^T \mathbf{x}$$

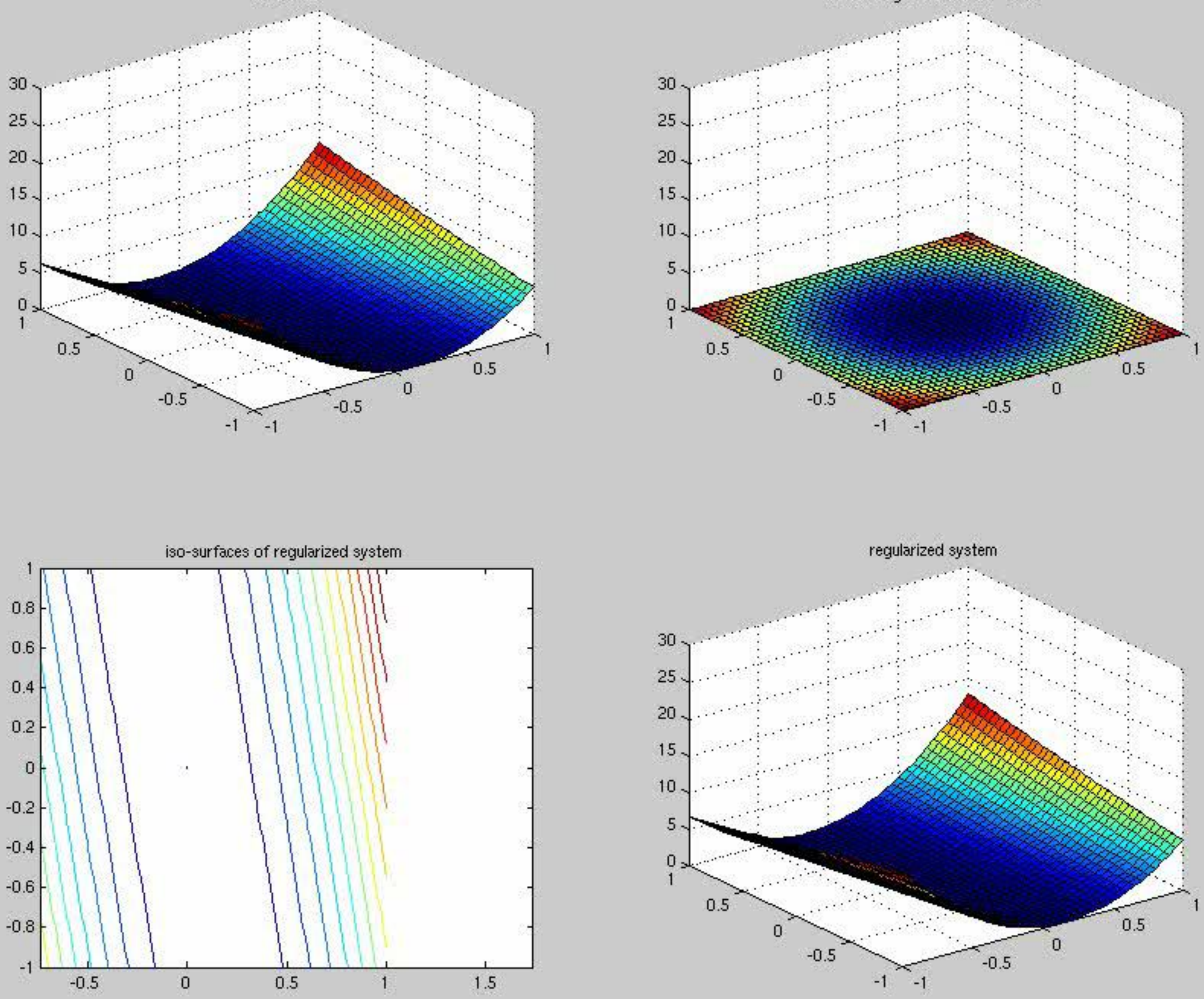
regularization parameter

- large λ will result in smooth solution, small λ fits the data well
- find good trade-off

Tikhonov Regularization - Example

- reconstruction for different choices of λ
- small lambda, many oscillations
- large lambda, smooth solution (in the limit constant)




30
25
20
15
10
5
0
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
30
25
20
15
10
5
0
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
iso-surfaces of regularized system
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.5
0
0.5
1
1.5
regularized system
30
25
20
15
10
5
0
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

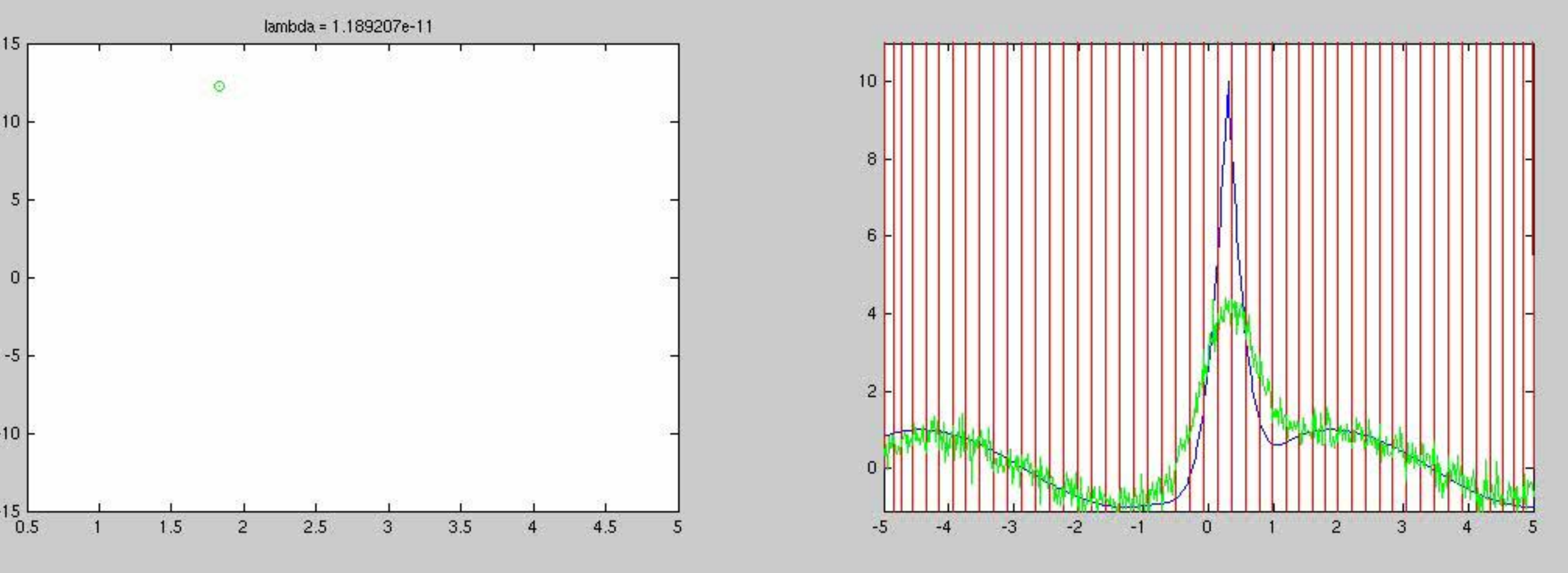
L-Curve criterion [Hansen98]

- need automatic way of determining
- want solution with small oscillations
- also want good data fit
- log-log plot of norm of residual (data fitting error) vs. norm of the solution (measure of oscillations in solution)

L-Curve Criterion

- video shows reconstructions for different λ
- start with $\lambda = 10^{-12}$

L-Curve **regularized solution**



L-Curve Criterion

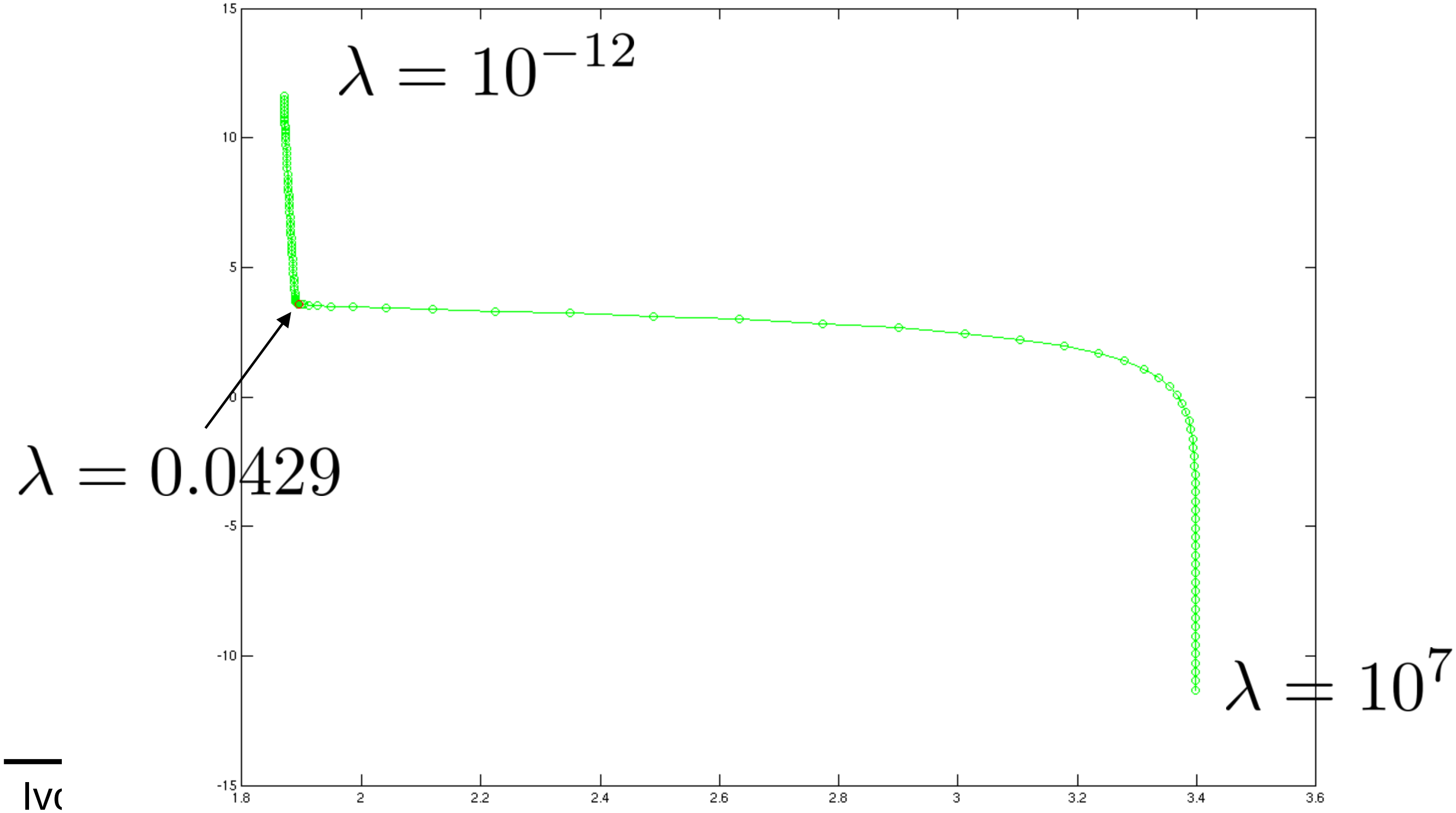
- compute L-Curve by solving inverse problem with choices of λ over a large range, e.g. $\lambda \in [10^{-12}, 10^{7}]$
- point of highest curvature on resulting curve corresponds to optimal regularization parameter
- curvature computation

$$\kappa = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{\frac{3}{2}}}$$

- find maximum κ and use corresponding λ to compute optimal solution

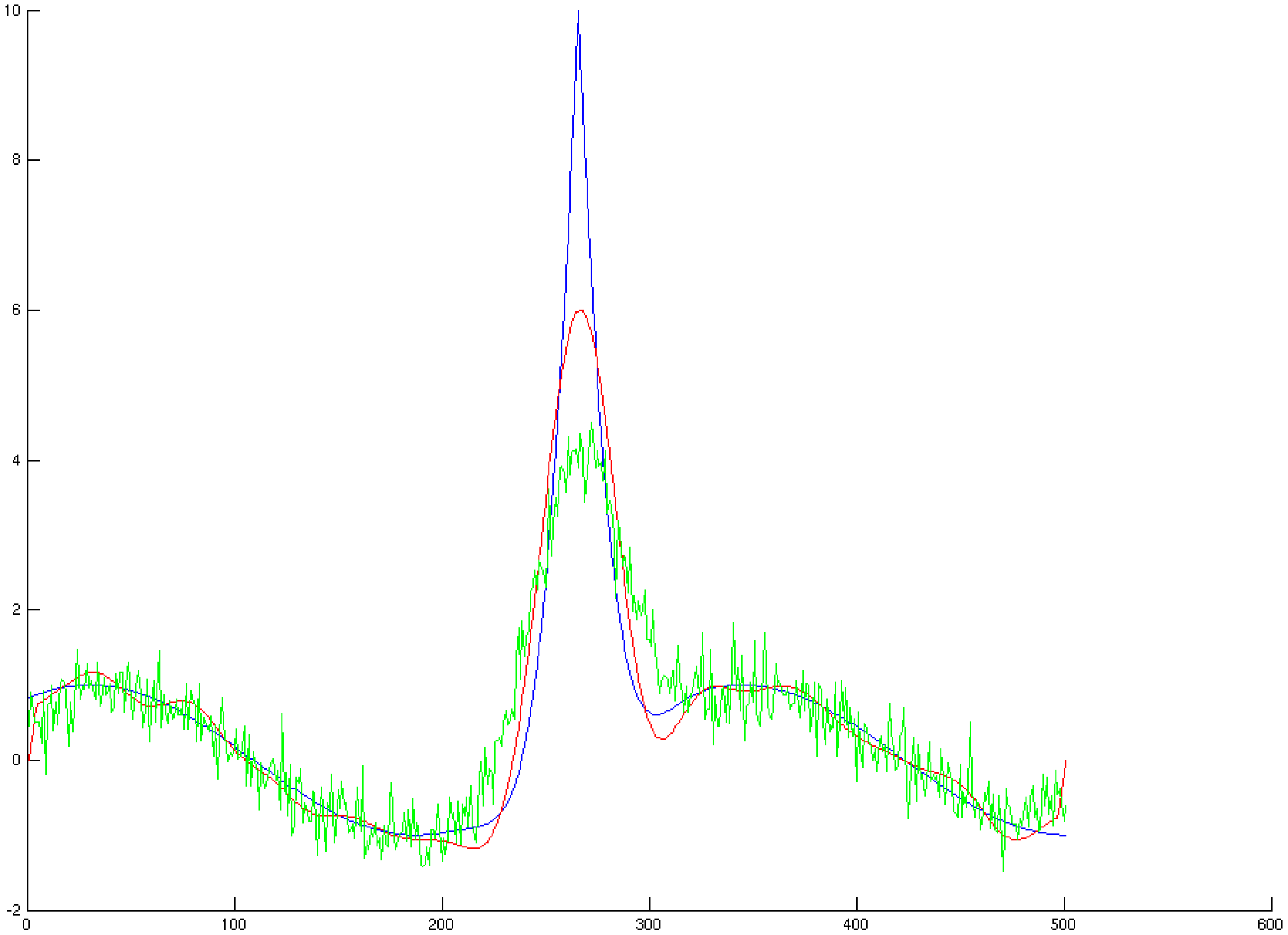
L-Curve Criterion – Example 1D Deconvolution

- L-curve with automatically selected optimal point
- optimal regularization parameter is different for every problem



L-Curve Criterion – Example 1D Deconvolution

- regularized solution (red) with optimal $\lambda = 0.0429$



Solving Large Linear Systems

- we can now regularize large ill-conditioned linear systems
- How to solve them ?
  - Gaussian elimination: $\mathcal{O}\left(N^3\right)$
  - SVD: $\mathcal{O}\left(N^3\right)$
- direct solution methods are too time-consuming
- Solution: approximate iterative solution

Iterative Solution Methods for Large Linear Systems

- stationary iterative methods [Barret94]
  - Examples
    - Jacobi
    - Gauss-Seidel
    - Successive Over-Relaxation (SOR)
  - use fixed-point iteration

$$\mathbf{x}^{t+1} = \mathbf{G}\mathbf{x}^{t} + \mathbf{c}$$

    - matrix G and vector c are constant throughout iteration
    - generally slow convergence
    - don't use for practical applications

Iterative Solution Methods for Large Linear Systems

- non-stationary iterative methods [Barret94]
  - conjugate gradients (CG)
    - symmetric, positive definite linear systems (SPD)
  - conjugate gradients for the normal equations

    short CGLS or CGNR
    - avoid explicit computation of $\mathtt{A}^T\mathtt{A}$
  - CG – type methods are good because
    - fast convergence (depends on condition number)
    - regularization built in !
    - number of iterations = regularization parameter
    - behave similar to truncated SVD

Iterative Solution Methods for Large Linear Systems

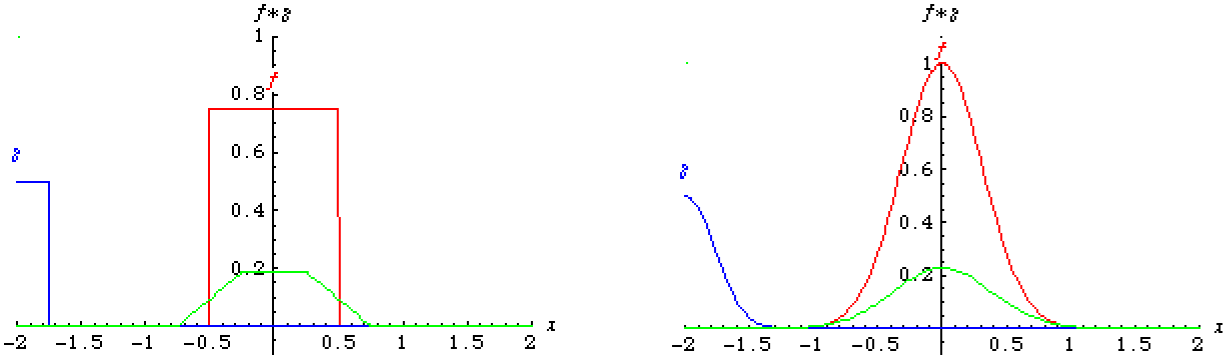
- iterative solution methods require only matrix-vector multiplications
- most efficient if matrix A is *sparse*
- sparse matrix means lots of zero entries
- back to our hypothetical 65536x65536 matrix
- memory consumption for full matrix:

$$2^{16} \times 2^{16} \times 8 \text{ bytes} = 32\,\text{Gbyte}$$

- sparse matrices store only non-zero matrix entries
- Question: How do we get sparse matrices ?

Iterative Solution Methods for Large Linear Systems

- answer: use a discretization with basis functions that have local support, i.e. which are themselves zero over a wide range
- for deconvolution the filter kernel should also be locally supported



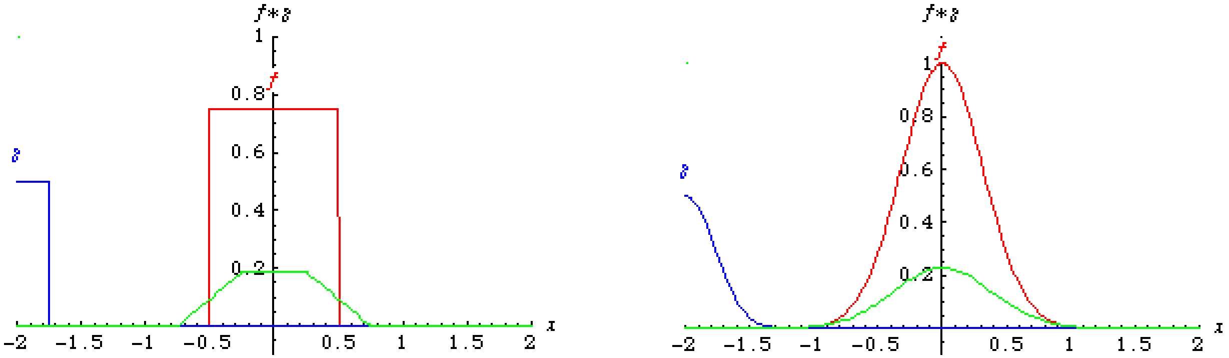
discretized model:

$$o = \sum_{i=0}^{N} m_i \left\{ \phi_i \otimes k \right\}$$

Iterative Solution Methods for Large Linear Systems

- answer: use a discretization with basis functions that have local support, i.e. which are themselves zero over a wide range

- for deconvolution the filter kernel should also be locally supported



discretized model:

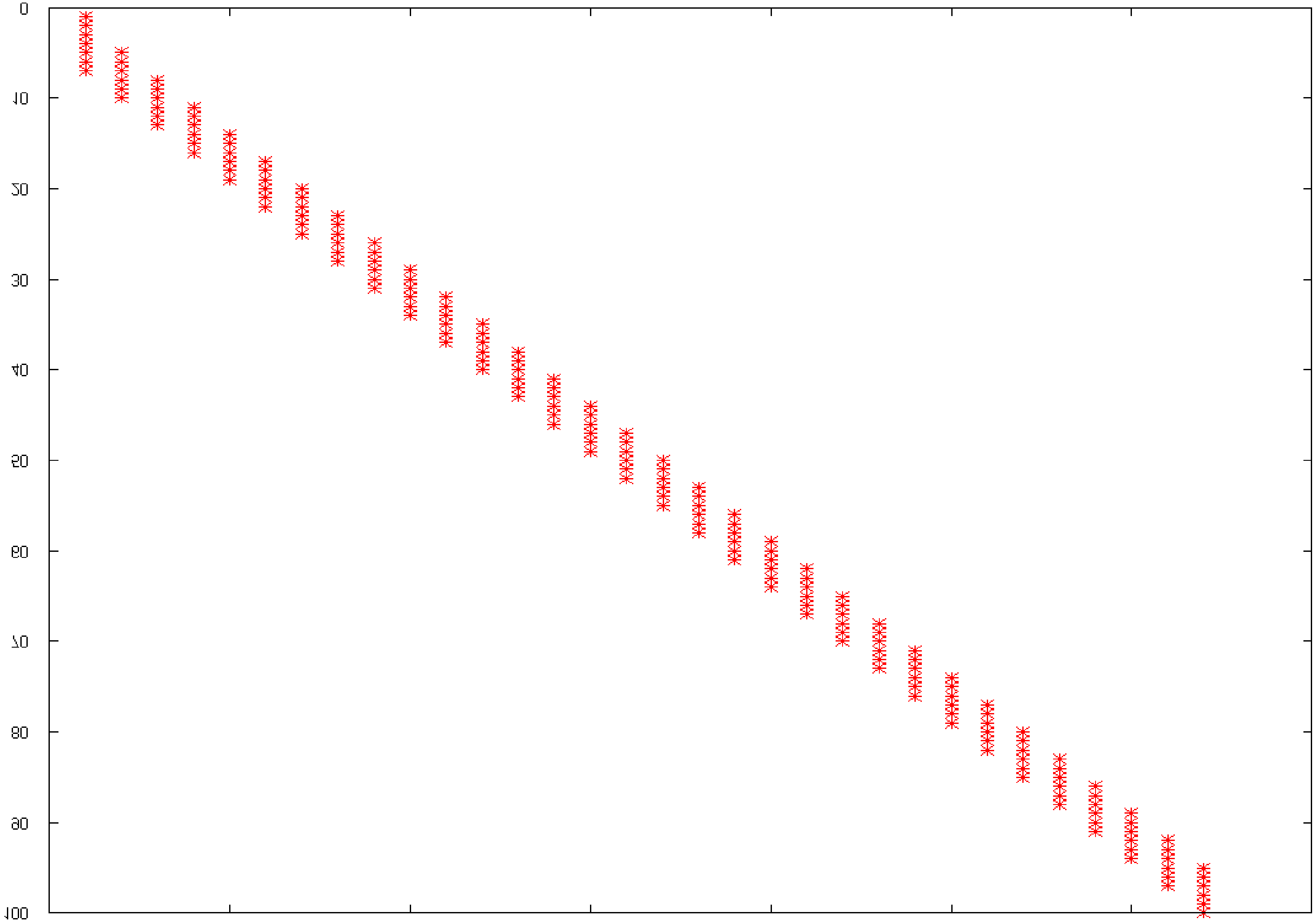
$$o = \sum_{i=0}^{N} m_i \{\phi_i \otimes k\}$$

will be zero over a wide range of values

Iterative Solution Methods for Large Linear Systems

sparse matrix structure for 1D deconvolution problem

32



Inverse Problems – Wrap Up

- inverse problems are often ill-posed
- if solution is unstable – check condition number
- if problem is small $< 4000^2$ use TSVD and Matlab
- otherwise use CG if problem is symmetric (positive definite), otherwise CGLS
- if convergence is slow try Tikhonov regularization – it's simple
  - improves condition number and thus convergence
- if problem gets large $> 15000^2$ make sure you have a sparse linear system!
- if system is sparse, avoid computing $\mathtt{A}^T\mathtt{A}$ explicitly – it is usually dense