Inverse Problems

Ivo Ihrke

Outline

- Theory
  - example 1D deconvolution
  - Fourier method
  - Algebraic method
    - discretization
    - matrix properties
    - regularization
    - solution methods
- Computed Tomography (CT)
  - Radon transform
  - Filtered Back-Projection
  - natural phenomena
  - glass objects

Inverse Problem - Definition

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

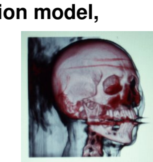
$$o = M(m)$$

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

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

Inverse Problems - Examples

- forward problem – volume rendering
  - given voxel data and image formation model, compute a view of the object
  - m are the volume coefficents, c is the ray that determines the pixel's value o (observation)



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

Inverse Problems - Examples

- inverse problem – CT
  - given the pixel values o, the ray geometry c and the image formation model, compute the volume densities m
  - invert

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

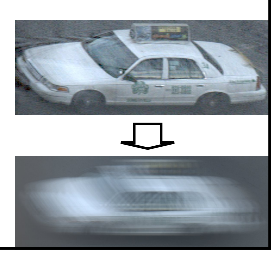
  - n is a noise component
  - we will later see how to do this

3D

Inverse Problems - Examples

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

$$o = m \otimes k$$



Inverse Problems - Examples

- 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 again noise

Inverse Problems - Theory

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

$$o = m \otimes k \Leftrightarrow F(o) = F(m) \cdot F(k)$$

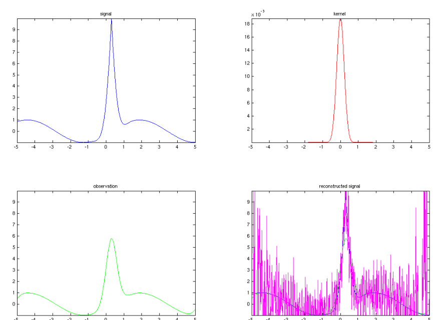
- deconvolution: $F(m) = \dfrac{F(o)}{F(k)}$
- problems
  - division by zero
  - Gibbs phenomenon (ringing artifacts)

A One-Dimensional Example – Deconvolution Spectral

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

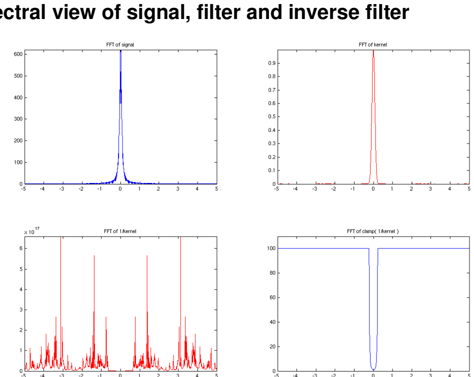
A One-Dimensional Example – Deconvolution Spectral

reconstruction is noisy even if data is perfect !



A One-Dimensional Example – Deconvolution Spectral

spectral view of signal, filter and inverse filter



A One-Dimensional Example - Deconvolution Spectral

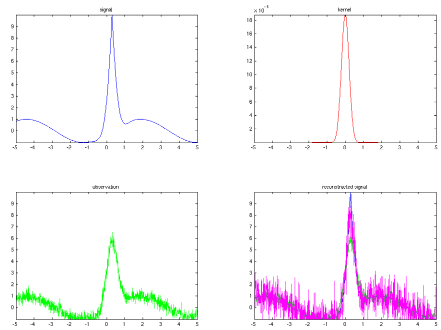
- solution: restrict frequency response of high pass filter (clamping)

$$G = \begin{cases} \frac{1}{F(k)} & \text{, if} \frac{1}{|F(k)|} < \gamma \\ \gamma \frac{F(k)}{|F(k)|} & \text{, else} \end{cases}$$

$$M = O \cdot G$$

A One-Dimensional Example - Deconvolution Spectral

reconstruction with clamped inverse filter



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

$$m(t) = \sum_i^N m_i \phi_i(t)$$

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)

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

$$\mathbf{o} = \mathbf{M}\mathbf{m}$$ 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)

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

unknown

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

A One-Dimensional Example – Deconvolution Algebraic

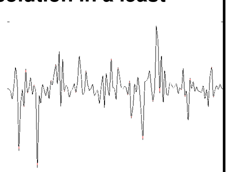
- normal equations – in case you forgot

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

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

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

→ apply to deconvolution problem



solution is completely broken !

A One-Dimensional Example – Deconvolution Algebraic

- Why ?
- analyze distribution of eigenvalues
- remember

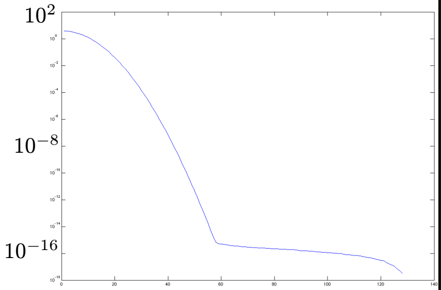
$$\det(\mathbf{A}) = \Pi_{i=0}^N \lambda_i, \quad \det(\mathbf{A}) = 0 \Rightarrow \text{matrix } \mathbf{A} \text{ is under-determined}$$

- actually we will check the singular values (square root of eigenvalues of $\mathbf{A}^T\mathbf{A}$)

A One-Dimensional Example – Deconvolution Algebraic

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

Log-Plot !



A One-Dimensional Example – Deconvolution Algebraic

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

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

- Inverse of M:

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

- singular values are diagonal elements of D
- inversion: $\mathbf{D}^{-1} = \mathrm{diag}(\frac{1}{\mathbf{D}_{ii}})$

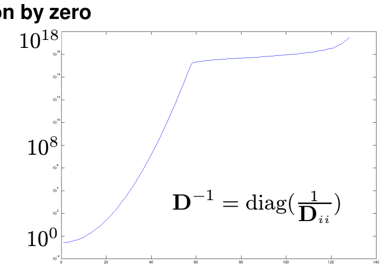
A One-Dimensional Example – Deconvolution Algebraic

- computing model parameters from observations:

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

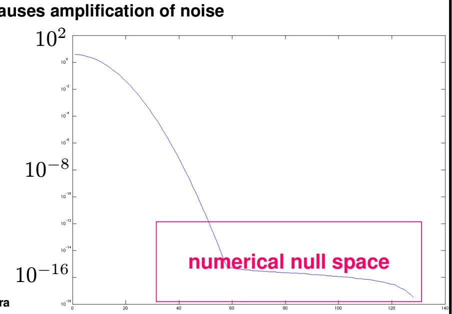
- again: amplification of noise
- potential division by zero

Log-Plot !



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
  - 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

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

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

Truncated Singular Value Decompostion

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

$$\mathbf{d}^+ = \begin{cases} \frac{1}{\mathbf{D}_{ii}} & , \mathbf{D}_{ii} > \epsilon \\ 0 & , \text{else} \end{cases} \qquad \mathbf{M}^+ = \mathbf{V}\mathbf{D}^+\mathbf{U}^T$$

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

- ϵ is called the *regularization* parameter

Regularization

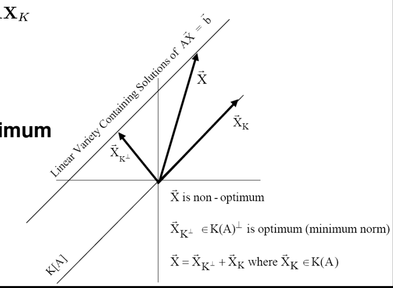
- countering the effect of ill-conditioned problems is called regularization
- an ill-conditioned problem behaves like a singular (under-constrained) system
- family of solutions exist
- → impose additional knowledge to pick a favorable solution
- TSVD results in minimum norm solution

Minimum Norm Solution

- K is the null-space of A

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

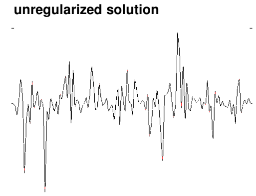
- $\mathbf{X}_{K^\perp}$ is the minimum norm solution



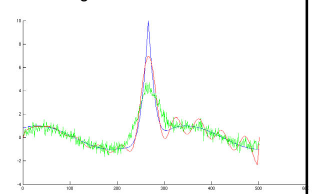
Example – 1D Deconvolution

- 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 $O(N^3)$
- today impossible to compute for systems larger than $\approx 6000^2$ (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)

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

- minimize modified quadratic form

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

- modified normal equations:

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

Modified Normal Equations

- include data term, smoothness term and blending parameter

data — smoothness

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

blending (regularization) parameter

Tikhonov Regularization

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

data fitting — minimum norm

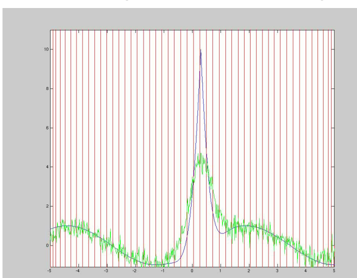
$$\min_{\mathbf{x}} \quad (\mathbf{A}\mathbf{x} - \mathbf{b})^T(\mathbf{A}\mathbf{x} - \mathbf{b}) + \lambda \mathbf{x}^T\mathbf{x}$$

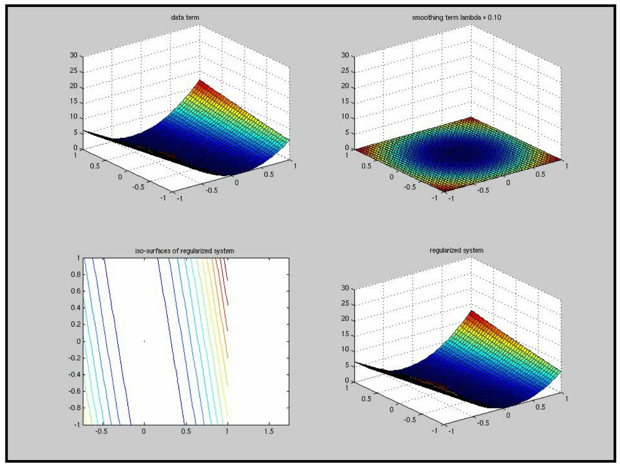
regularization parameter

- 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)





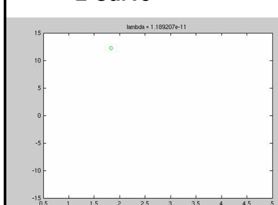
L-Curve criterion [Hansen98]

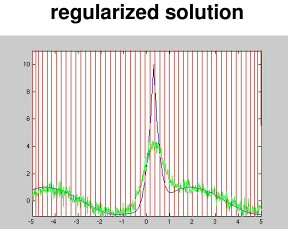
- need automatic way of determining λ
- want solution with small oscillations
- also want good data fit
- log-log plot of norm of residual (data fitting error) vs. norm of the solution (measure of oscillations in solution)

L-Curve Criterion

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

L-Curve — **regularized solution**





L-Curve Criterion

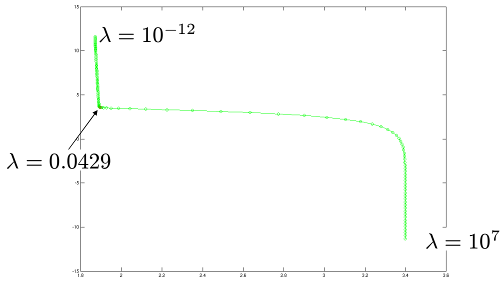
- compute L-Curve by solving inverse problem with choices of λ over a large range, e.g. $(10^{-12} - 10^{4})$
- point of highest curvature on resulting curve corresponds to optimal regularization parameter
- curvature computation

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

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

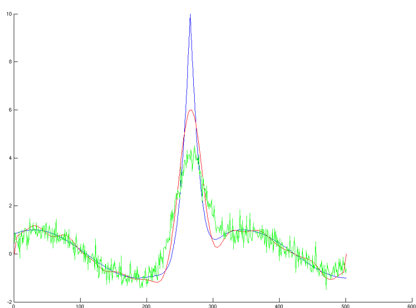
L-Curve Criterion – Example 1D Deconvolution

- 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 $O(N^3)$
  - SVD $O(N^3)$
- direct solution methods are too time-consuming
- Solution: approximate iterative solution

Iterative Solution Methods for Large Linear Systems

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

    $\mathbf{x}_{k+1} = \mathbf{G}\mathbf{x}_k + \mathbf{c}$
  - 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 $\mathbf{A}^T\mathbf{A}$
  - 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

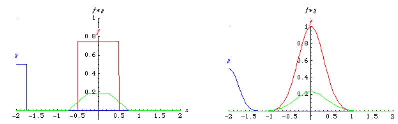
- iterative solution methods require only matrix-vector multiplications
- most efficient if matrix $\mathbf{A}$ is *sparse*
- sparse matrix means lots of zero entries
- back to our hypothetical 65536x65536 matrix
- memory consumption for full matrix:

$2^{16} \cdot 2^{16} \cdot 8\text{bytes} = 32\text{GB}$

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

Iterative Solution Methods for Large Linear Systems

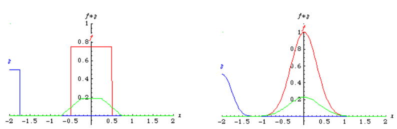
- answer: use a discretization with basis functions that have local support, i.e. which are themselves zero over a wide range
- for deconvolution the filter kernel should also be locally supported



discretized model: $o = \sum_i^N m_i(\phi_i \otimes k)$

Iterative Solution Methods for Large Linear Systems

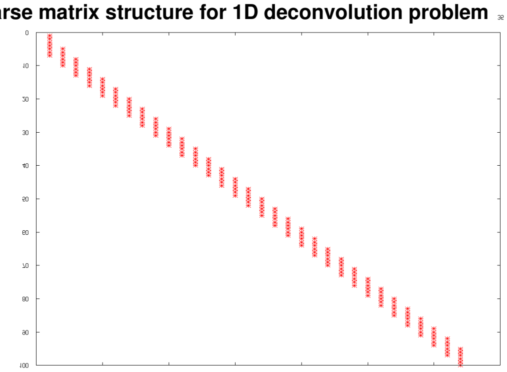
- answer: use a discretization with basis functions that have local support, i.e. which are themselves zero over a wide range
- for deconvolution the filter kernel should also be locally supported



discretized model: $o = \sum_i^N m_i(\phi_i \otimes k)$ will be zero over a wide range of values

Iterative Solution Methods for Large Linear Systems

sparse matrix structure for 1D deconvolution problem



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 $\mathbf{A}^T\mathbf{A}$ explicitly – it is usually dense

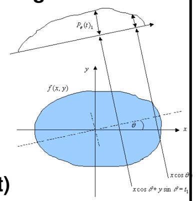
Computed Tomography (CT)

3D

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



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

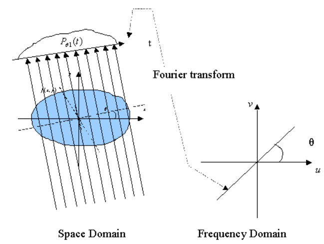
- invert this equation (noise is present)

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

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

Computed Tomography – Frequency Space Approach

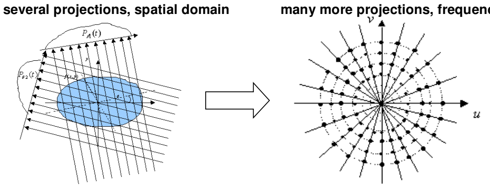
- Fourier Slice Theorem
- the Fourier transform of an orthogonal projection is a slice of the Fourier transform of the function !



Computed Tomography – Frequency Space Approach

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

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



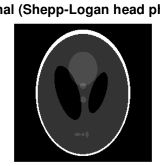
- slices are usually interpolated onto a rectangular grid
- inverse fourier transform
- gaps for high frequency components

→ artifacts

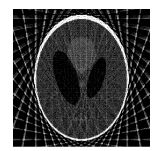
Frequency Space Approach - Example

without noise !

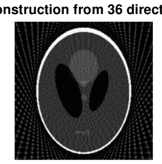
original (Shepp-Logan head phantom)



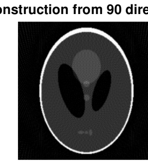
reconstruction from 18 directions



reconstruction from 36 directions

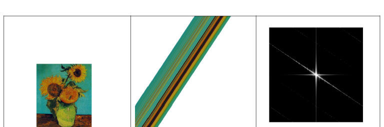


reconstruction from 90 directions



Filtered Back-Projection

- most commonly used CT algorithm
- principle:
  - we have seen the line in frequency space before!
  - light field lecture



(a1) Scene image (a2) EPI (a3) Fourier transform of EPI

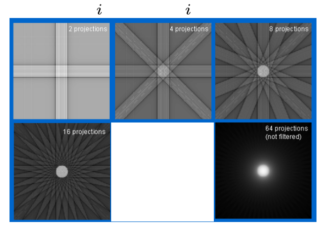
  - inverse Fourier transform of a line in frequency domain yields a projection smeared out in space in the spatial domain

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

$$F(\sum_i l_i) = \sum_i F(l_i)$$

backprojection:



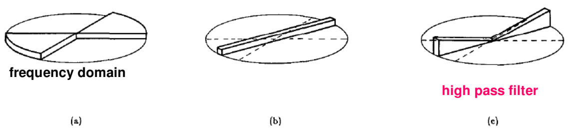
Filtered Back-Projection

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

would like to have wedge shape for one discrete measurement

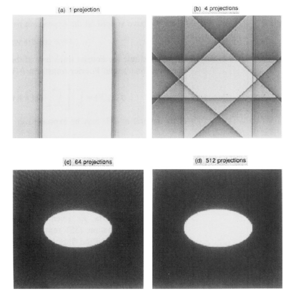
have a bar shape (discrete measurement)

compensate to have equal volume under filter



Filtered Back-Projection

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

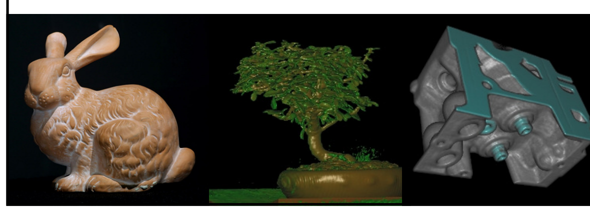


Frequency Space based Methods - Disadvantages

- need orthogonal projections
- need precise acquisition setup – optical axes of all projections must intersect in one point
- sensitive to noise because of high pass filtering

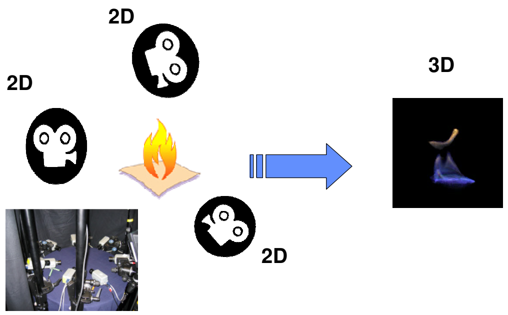
CT Applications in graphics

- acquisition of difficult to scan objects
- X-Rays are not refracted → can scan glass objects
- some CT scan examples – sorry no glass object in the examples



Tomographic Imaging - Graphics

- reconstruction of flames using a multi-camera setup

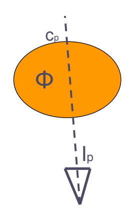


Algebraic Reconstruction Techniques

- 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



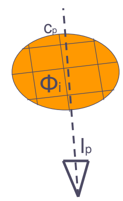
ART

- 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

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



- we have seen this before !

ART – Matrix Structure

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

Basis functions i →

pixels p ↓

$\int_{c_1} \phi_1 ds$	$\int_{c_1} \phi_2 ds$	$\int_{c_1} \phi_3 ds$	$\int_{c_1} \phi_4 ds$	$\int_{c_1} \phi_5 ds$
$\int_{c_2} \phi_1 ds$	$\int_{c_2} \phi_2 ds$	$\int_{c_2} \phi_3 ds$	$\int_{c_2} \phi_4 ds$	$\int_{c_2} \phi_5 ds$
$\int_{c_3} \phi_1 ds$	$\int_{c_3} \phi_2 ds$	$\int_{c_3} \phi_3 ds$	$\int_{c_3} \phi_4 ds$	$\int_{c_3} \phi_5 ds$

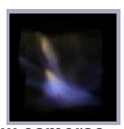
. . .

invert LS in a least squares sense:

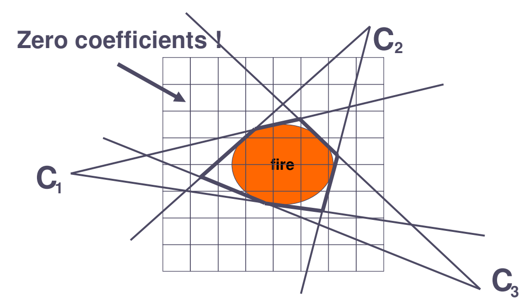
$$a = (S^T S)^{-1} S^T p$$

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

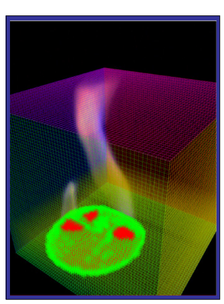


Visual Hull Restricted Tomography

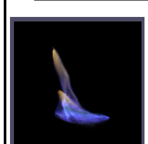


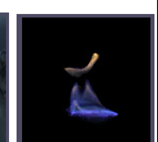
Visual Hull Restricted Tomography

- Only about 1/10 of the voxels contribute
- Remove voxels that do not contribute from linear system
- Complexity of inversion is significantly reduced

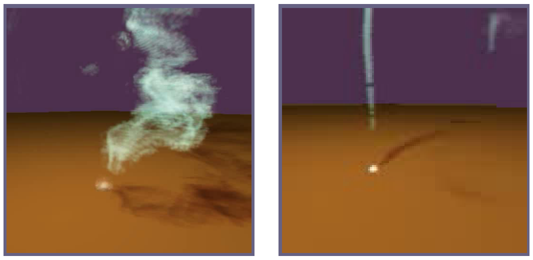


Flame Reconstructions



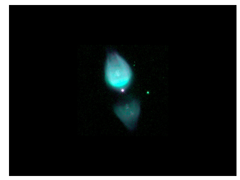


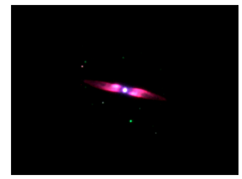
Smoke Reconstructions



3D Reconstruction of Planetary Nebulae [Magnor04]

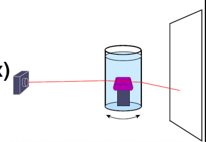
- only one view available
- exploit axial symmetry
- essentially a 2D problem





3D Scanning of Glass Objects [Trifonov06]

- uses visible light
- tomography needs straight ray pathes
- compensate for refraction
- put glass object into water
- add salt (increases refractive index)
- once refractive index is the same as that of the glass, ray pathes are straight
- can apply tomographic reconstruction



3D Scanning of Glass Objects [Trifonov06]

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