Computational Optical Imaging -
Optique Numerique

-- Deconvolution --

Winter 2013

Ivo Ihrke
Volumetric 3D

Volume Scanning
3D Scanning of “Unscannable” Objects
Triangulation scanning

- Traditional triangulation scanners illuminate the surface of an object
- Detect **laser line on object surface**
Standard Triangulation scanning

- problematic cases for classical scanners

- translucency
- transparency
- low albedo
Light Sheet Range Scanning

- What if we could see light rays propagating through space?
- *detecting where laser sheet ends might*
Surface Scanning - Setup

- Glass tank
- Laser
- Laser sheet
- Dye
- Camera
Surface Scanning

- Laser sheet hitting various surfaces in fluorescent liquid

object begins where fluorescence ends
Surface Scanning

Example stack of input images
Surface Scanning

- Space-time analysis of image stack

We don't know what happens behind the 1st surface! Refraction, reflection, scattering, ...
Results

[Hullin, Fuchs, Ihrke, Seidel, Lensch - SIGGRAPH 2008]
Acquisition
Object
Another Object
Mouse slices
Surface Extraction - Marching Cubes
Reconstruction

photograph
Volumetric 3D

Volume Scanning - More Robust Beams
Volume Slicing Microscopy

Citation

Ivo Ihrke / Winter 2013
Bessel Beams

- Ring system of the Bessel beam
- Main lobe of the Bessel beam
- Scattered light in the ring system
- Ideal detection volume
Bessel Beams in Volume Slicing Microscopy

Citation
Principle of operation

Gaussian beam

Without object

With object

**Healing property**

Citation

Ivo Ihrke / Winter 2013
Sectioned Bessel Beams

Fig. 3

Citation
Citation
Volumetric 3D

Volume Scanning – Resilience against Scattering
Confocal scanning microscopy

light source
pinhole
Confocal scanning microscopy

- Image blur $\Rightarrow 1/r^2$ falloff
- Same for the light source
- $1/r^4$ falloff
Confocal scanning microscopy

light source

pinhole

pinhole

photocell
Confocal scanning microscopy

- Light source
- Pinhole
- Photocell
Two-Photon Microscopy

- Non-linear excitation reduces scattering in out-of-focus regions

[1-photon excitation][2-photon excitation]

[constant][1/z^2]

[Boston University, Biomicroscopy Lab]
Deconvolution

Ivo Ihrke
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
Ivo Ihrke / Winter 2013
Applications - Microscopy

Images courtesy Meyer Instruments
Ivo Ihrke / Winter 2013
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
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:

\[ \Rightarrow \mathcal{F}\{m\} = \frac{\mathcal{F}\{o\}}{\mathcal{F}\{k\}} \]

- problems
  - division by zero
  - Gibbs phenomenon
    - (ringing artifacts)
A One-Dimensional Example – Deconvolution Spectral

- most common: \( \mathcal{F}\{k\} \) is a low pass filter

\[
1 \rightarrow \frac{1}{\mathcal{F}\{k\}}, \text{ the inverse filter, is high pass}
\]

- \( \rightarrow \) 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)

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

\[
\mathcal{F}\{m\} = \mathcal{F}\{o\} \cdot \mathcal{F}\{g\}
\]
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

\[ \mathcal{F}\{g\}(\omega) := \frac{1}{\mathcal{F}\{k\}(\omega)} \frac{\left|\mathcal{F}\{k\}\right|^2(\omega)}{\left|\mathcal{F}\{k\}\right|^2(\omega) + \left|\frac{1}{\text{SNR}(\omega)}\right|} \]
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)

$$o(x) = \{m \otimes k\}(x)$$

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

$$= \int_{-\infty}^{\infty} \sum_{i=0}^{N} m_i \phi_i(t)k(x-t)dt$$

$$= \sum_{i=0}^{N} m_i \int_{-\infty}^{\infty} \phi_i(t)k(x-t)dt$$

$$= \sum_{i=0}^{N} m_i \{\phi_i \otimes k\}(x)$$

$$o = Mm \quad \text{linear system}$$
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)

\[
o(x) = \{m \otimes k\}(x) \\
= \int_{-\infty}^{\infty} m(t)k(x-t)dt \\
= \int_{-\infty}^{\infty} \sum_{i=0}^{N} m_i \phi_i(t)k(x-t)dt \\
= \sum_{i=0}^{N} m_i \int_{-\infty}^{\infty} \phi_i(t)k(x-t)dt \\
= \sum_{i=0}^{N} m_i \{\phi_i \otimes k\}(x)
\]

\[ o = \mathbf{Mm} \quad \text{linear system} \]
A One-Dimensional Example – Deconvolution Algebraic

- normal equations

\[
\min_x \|Ax - b\|_2^2 = \min_x (Ax - b)^T (Ax - b) = \min x f(x)
\]

\[
\nabla f = 2A^T Ax - 2A^T b = 0
\]

→ solve \( A^T Ax = A^T b \) 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:

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

- we will check the singular values
  - Ok, since \( A^T A \) is SPD (symmetric, positive semi-definite)
    \( \Rightarrow \) non-negative eigenvalues
- Singular values are the square root of the eigenvalues

Ivo Ihrke / Winter 2013
A One-Dimensional Example – Deconvolution Algebraic

- matrix $M^T M$ 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

Log-Plot!
A One-Dimensional Example – Deconvolution Algebraic

- Why is this bad?

- Singular Value Decomposition: U, V are orthonormal, D is diagonal

  \[ M = UDV^T \]

- Inverse of M:

  \[ M^{-1} = (UDV^T)^{-1} \]
  \[ = V^{-T}D^{-1}U^{-1} \]
  \[ = VD^{-1}U^T \]

- Singular values are diagonal elements of D

- Inversion:

  \[ D^{-1} = \text{diag}\left(\frac{1}{D_{i,i}}\right) \]
A One-Dimensional Example – Deconvolution Algebraic

- computing model parameters from observations:
  \[ m = M^{-1} o = V D^{-1} U^T o \]
- again: amplification of noise
- potential division by zero

Log-Plot!

\[ D^{-1} = \text{diag} \left( \frac{1}{D_{i,i}} \right) \]
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

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

- smaller condition number \( \rightarrow \) 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)

\[
d^+ = \begin{cases} 
\frac{1}{D_{i,i}} & \text{if } D_{i,i} > \epsilon \\
0 & \text{else}
\end{cases}
\]

\[
D^+ = \text{diag}(d^+)
\]

\[
M^+ = VD^+U^T
\]

- \( \epsilon \) is called the regularization parameter
Minimum Norm Solution

- Let $K[A]$ be the null-space of $A$ and $X_K \in K$

  \[\begin{align*}
  \Rightarrow AX_K &= 0 \\
  \Rightarrow AX &= A(X_{K\perp} + X_K) \\
  &= AX_{K\perp} + AX_K \\
  &= AX_{K\perp} + 0 \\
  &= AX_{K\perp} \\
  &= b
  \end{align*}\]

- $X_{K\perp}$ is the minimum norm solution
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 \( \mathcal{O}(N^3) \)

<table>
<thead>
<tr>
<th>system size</th>
<th>512</th>
<th>1024</th>
<th>2048</th>
<th>4096</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVD time (in s)</td>
<td>0.27</td>
<td>1.75</td>
<td>12.54</td>
<td>96.28</td>
</tr>
</tbody>
</table>

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

- today impractical to compute for systems larger than \( \geq 16384^2 \)
  (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)

  \[
  \min_x \quad \alpha \|Ax - b\|^2_2 + (1 - \alpha)\|Rx\|^2_2 = \\
  \min_x \quad \alpha (Ax - b)^T (Ax - b) + (1 - \alpha)x^T R^T Rx = \\
  \min_x \quad \hat{f}(x)
  \]

- minimize modified quadratic form

  \[\nabla \hat{f}(x) = 2\alpha A^T Ax - 2A^T b + 2(1 - \alpha)R^T Rx = 0
  \]

- regularized normal equations:

  \[(\alpha A^T Ax + (1 - \alpha)R^T R)x = A^T b\]
Modified Normal Equations

- include data term, smoothness term and blending parameter

\[ (\alpha A^T A x + (1 - \alpha) R^T R) x = A^T b \]

data \hspace{5em} \text{Prior information (popular: smoothness)}

blending (regularization) parameter
Tikhonov Regularization

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

\[
\min_{x} (Ax - b)^T (Ax - b) + \lambda x^T x
\]

- large $\lambda$ will result in smooth solution, small $\lambda$ fits the data well

- find good trade-off
Tikhonov Regularization - Example

- reconstruction for different choices of $\lambda$
- small lambda, many oscillations
- large lambda, smooth solution (in the limit constant)
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 $\lambda$
- start with $\lambda = 10^{-12}$

L-Curve regularized solution
L-Curve Criterion

- compute L-Curve by solving inverse problem with choices of $\lambda$ 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

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

- find maximum $\kappa$ and use corresponding $\lambda$ 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]
  - Examples
    - Jacobi
    - Gauss-Seidel
    - Successive Over-Relaxation (SOR)
  - use fixed-point iteration
    \[ \mathbf{x}^{t+1} = G\mathbf{x}^t + \mathbf{c} \]
    - 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 $A^T A$
  - 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 $65536 \times 65536$ 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

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

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

discretized model: will be zero over a wide range of values
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 $A^T A$ explicitly – it is usually dense
Computational Photography Example

Hardware-based Stabilization of Deconvolution (Flutter Shutter Camera)

Slides by Ramesh Raskar
Traditional Camera
Shutter is OPEN
Flutter Shutter Camera
Shutter is OPEN and CLOSED
Comparison of Blurred Images

Ivo Ihrke / Winter 2013
Implementation
Completely Portable
Blurring == Convolution

Log space

Zeros!

Traditional Camera: Box Filter
Flutter Shutter: Coded Filter

Ivo Ihrke / Winter 2013
References


[Raskar06] - Ramesh Raskar, Amit Agrawal, and Jack Tumblin, Coded Exposure Photography: Motion Deblurring using Fluttered Shutter, ACM SIGGRAPH 2006