Fast converging iterative algorithms for PET

Jeffrey A. Fessler

EECS Department, BME Department, and
Nuclear Medicine Division of Dept. of Internal Medicine
The University of Michigan

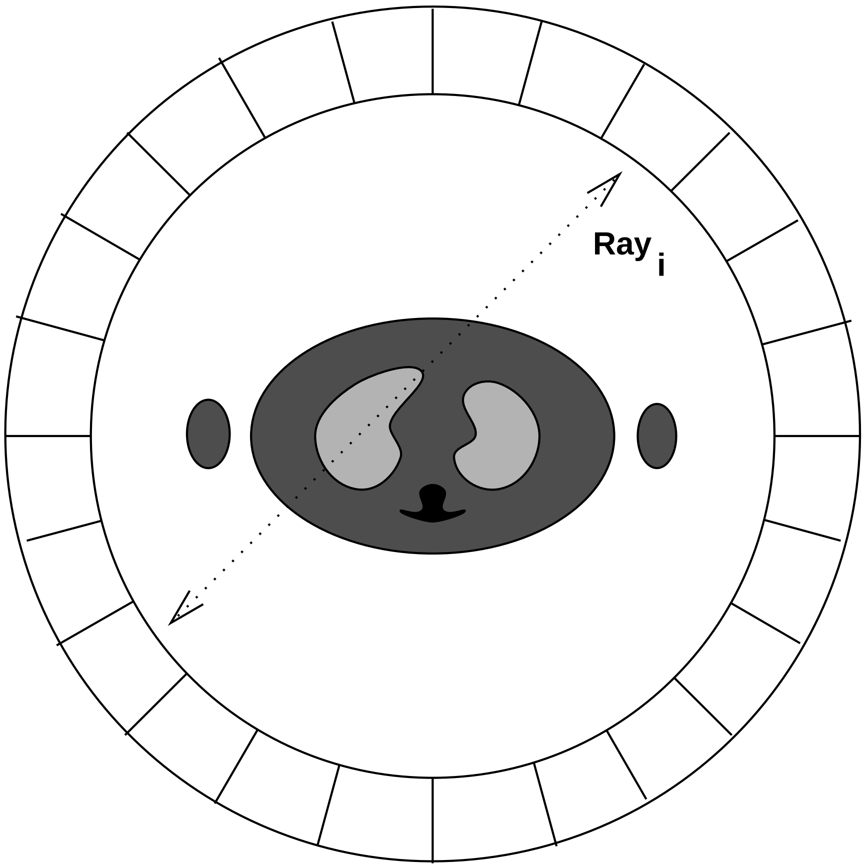
VIII Symposium on the Medical Applications of Cyclotrons

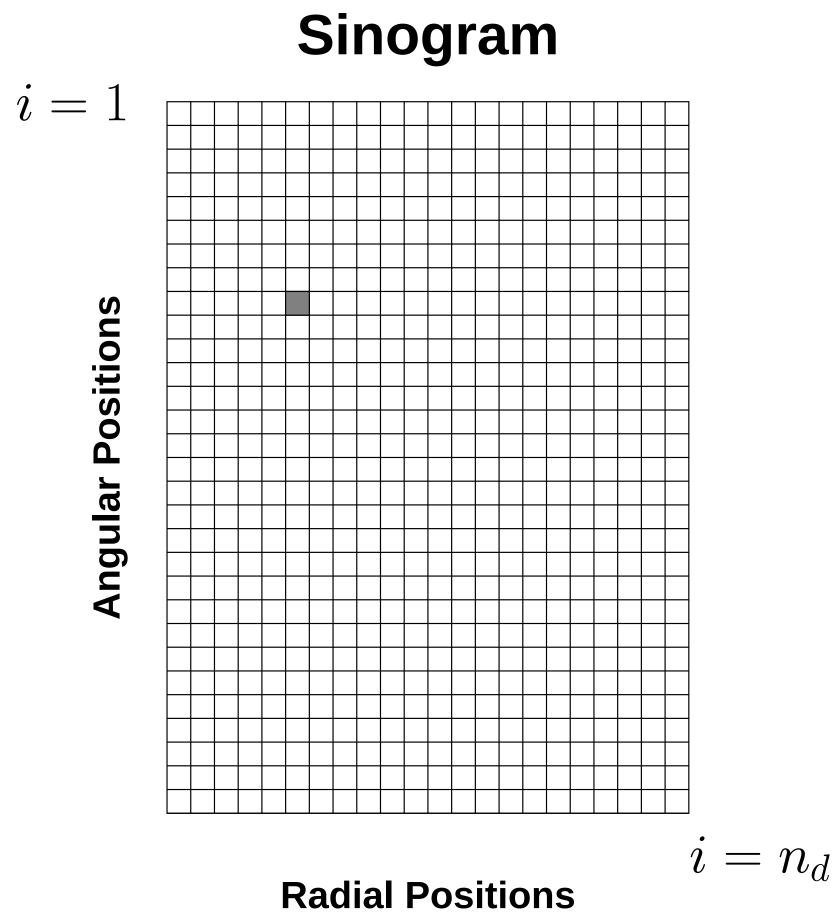
May 23, 1999

Outline

- Problem statement
- Choices / tradeoffs / considerations
  - 1. Object parameterization
  - 2. System physical modeling
  - 3. Statistical modeling of measurements
  - 4. Objective functions and regularization
  - 5. Iterative algorithms
- Examples
- Open problems

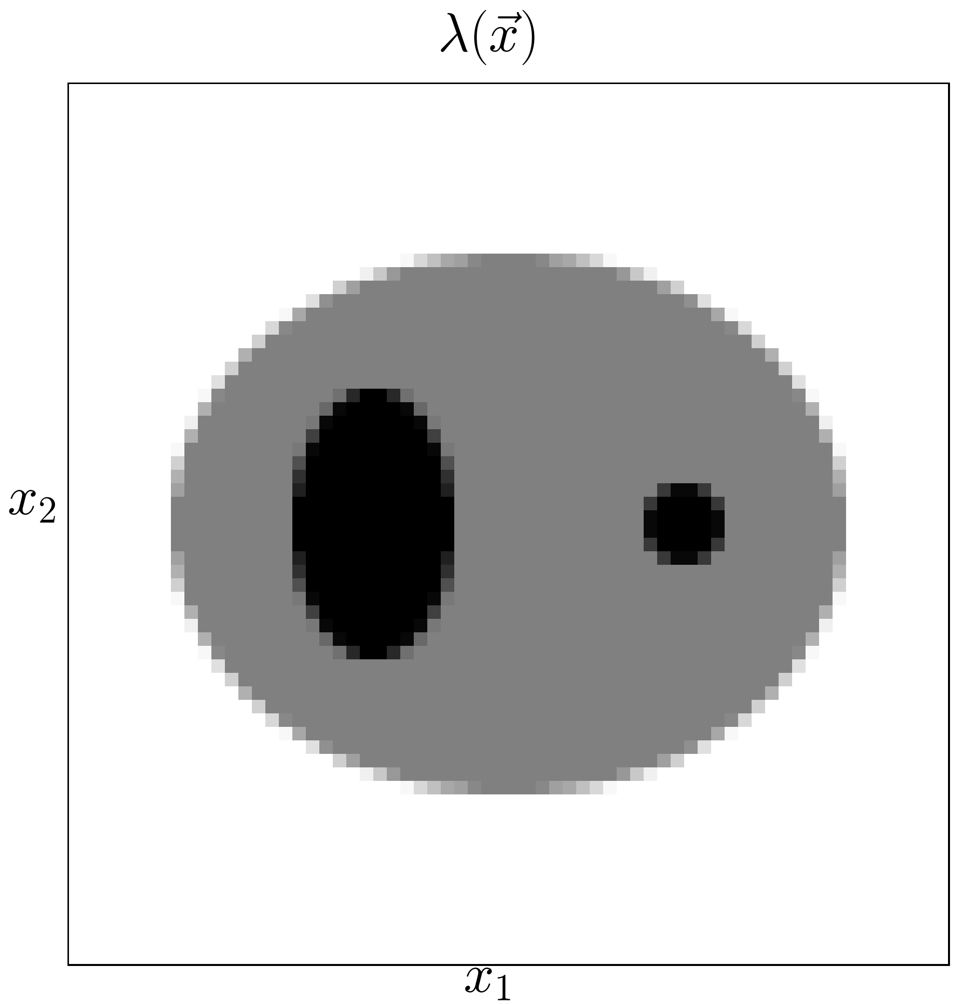
PET Data Collection



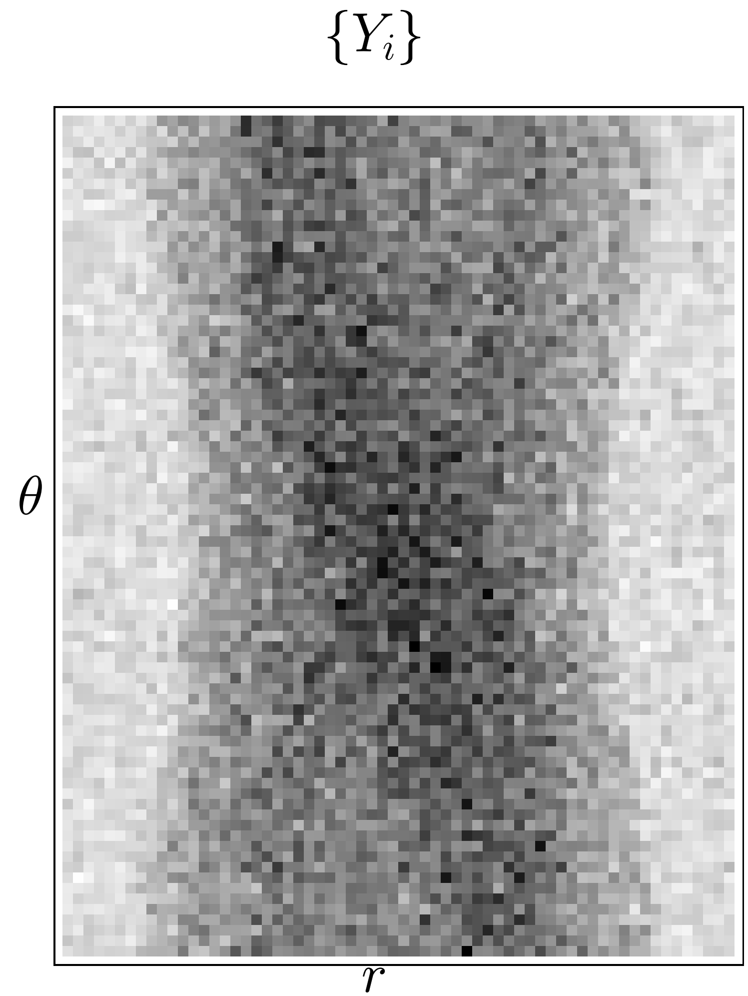


$$n_d \approx (n_{\text{crystals}})^2$$

PET Reconstruction Problem - Illustration



Image



Sinogram

Reconstruction Methods

(Simplified View)

Analytical (FBP)	**Iterative** (OSEM?)

Reconstruction Methods

ANALYTICAL

FBP
BPF
Gridding
...

ITERATIVE

Algebraic
(y = Ax)

ART
MART
SMART
...

Statistical

Least Squares

CG
CD
ISRA
...

Poisson Likelihood

EM (etc.)
OSEM
SAGE
CG
Int. Point
GCA
PSCD
FSCD ...

History of Statistical Image Reconstruction

- First use of iterative methods for tomography

  X-ray CT?
- Weighted least squares for 3D SPECT

  (Goitein, NIM, 1972)
- First proposal of Poisson likelihood for emission tomography

  (Rockmore and Macovski, TNS, 1976)
- First proposal of Poisson likelihood for transmission tomography

  (Rockmore and Macovski, TNS, 1977)
- First EM algorithm for Poisson emission model

  (Shepp and Vardi, TMI, 1982)
- First EM algorithm for Poisson transmission model

  (Lange and Carson, JCAT, 1984)
- Late 1990's - commercial availability of OSEM

  (Hudson and Larkin, TMI, 1994)

Why Statistical Methods?

- Object constraints (*e.g.* nonnegativity)
- Accurate models of physics (quantitative accuracy)
  (*e.g.* nonuniform attenuation in SPECT)
- System response models (*possibly* improved spatial resolution)
- Appropriate statistical models (less variance)
  (FBP treats all rays equally)
- Side information (*e.g.* MRI or CT boundaries)
- Nonstandard geometries (“missing” data)

Disadvantages?

- Computation time
- Model complexity
- Software complexity
- Less predictable (due to nonlinearities), especially for some methods
  e.g. Huesman (1984) FBP ROI variance for kinetic fitting

Five Categories of Choices

1. Object parameterization: $\lambda(\vec{x})$ vs $\underline{\lambda}$
2. System physical model: $s_i(\vec{x})$
3. Measurement statistical model $Y_i \sim \boxed{?}$
4. Objective function: data-fit / regularization
5. Algorithm / initialization

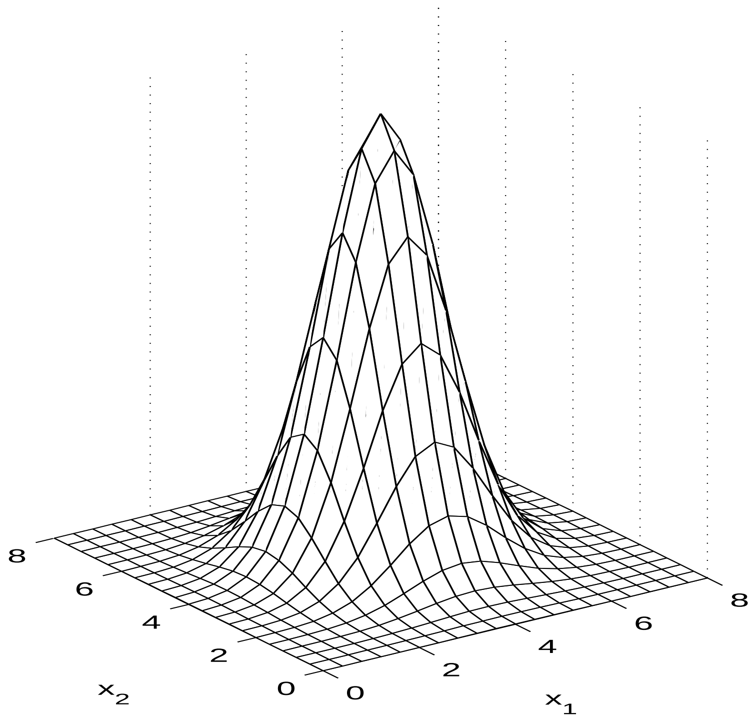
No perfect choices - one can critique all approaches!

Choices impact:

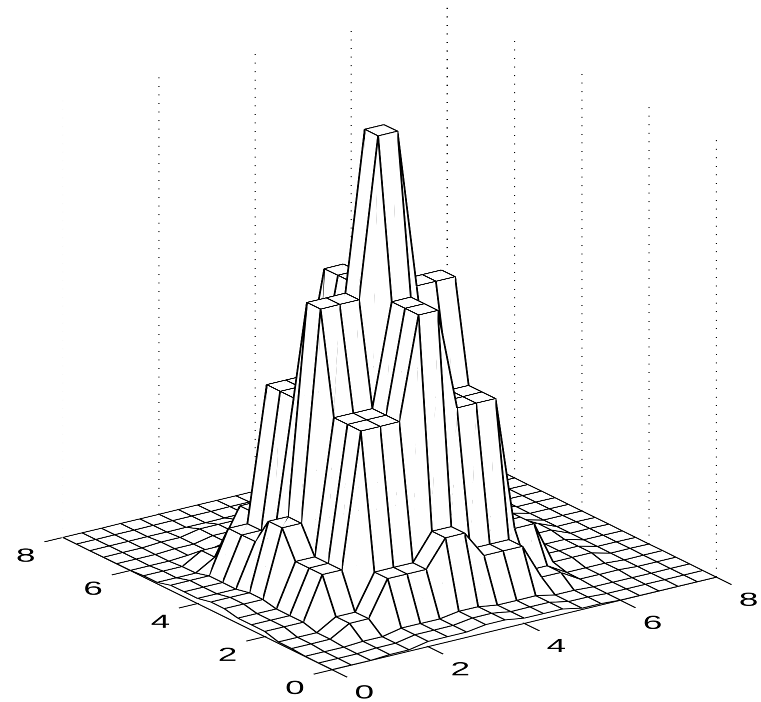
- Image spatial resolution
- Image noise
- Quantitative accuracy
- Computation time
- Memory
- Algorithm complexity

Choice 1. Object Parameterization

Radioisotope spatial distribution $\rightarrow$ $\lambda(\vec{x}) \approx \tilde{\lambda}(\vec{x}) = \sum_{j=1}^{n_p} \lambda_j \, b_j(\vec{x})$ $\leftarrow$ Series expansion "basis functions"



Object $\lambda(\vec{x})$



Pixelized approximation $\tilde{\lambda}(\vec{x})$

Basis Functions

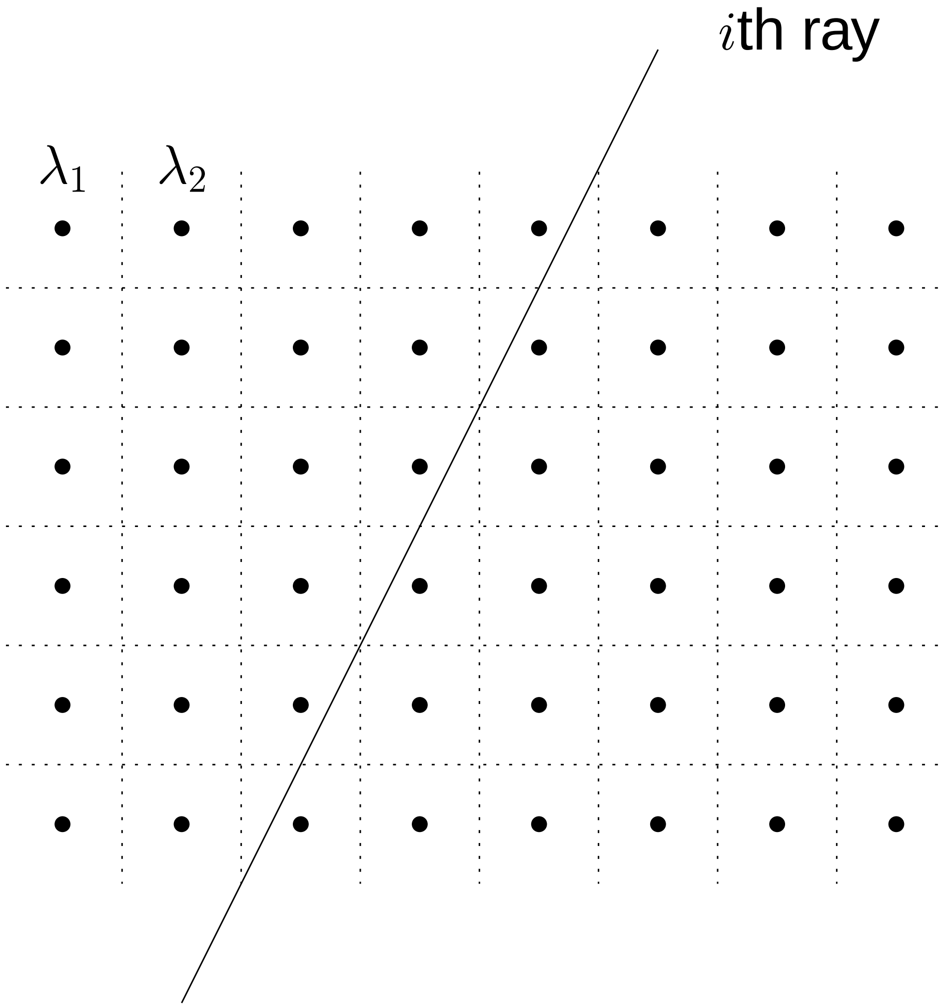
Choices

- Fourier series
- Circular harmonics
- Wavelets
- Kaiser-Bessel windows
- Overlapping disks
- B-splines (pyramids)
- Polar grids
- Logarithmic polar grids
- "Natural pixels"
- Point masses
- pixels / voxels
- ...

Considerations

- Represent object $\lambda(\vec{x})$ "well" with moderate n_p
- system matrix elements $\{a_{ij}\}$ "easy" to compute
- The $n_d \times n_p$ system matrix: $\boldsymbol{A} = \{a_{ij}\}$, should be sparse (mostly zeros).
- Easy to represent nonnegative functions
  e.g., if $\lambda_j \geq 0$, then $\lambda(\vec{x}) \geq 0$, *i.e.* $b_j(\vec{x}) \geq 0$.

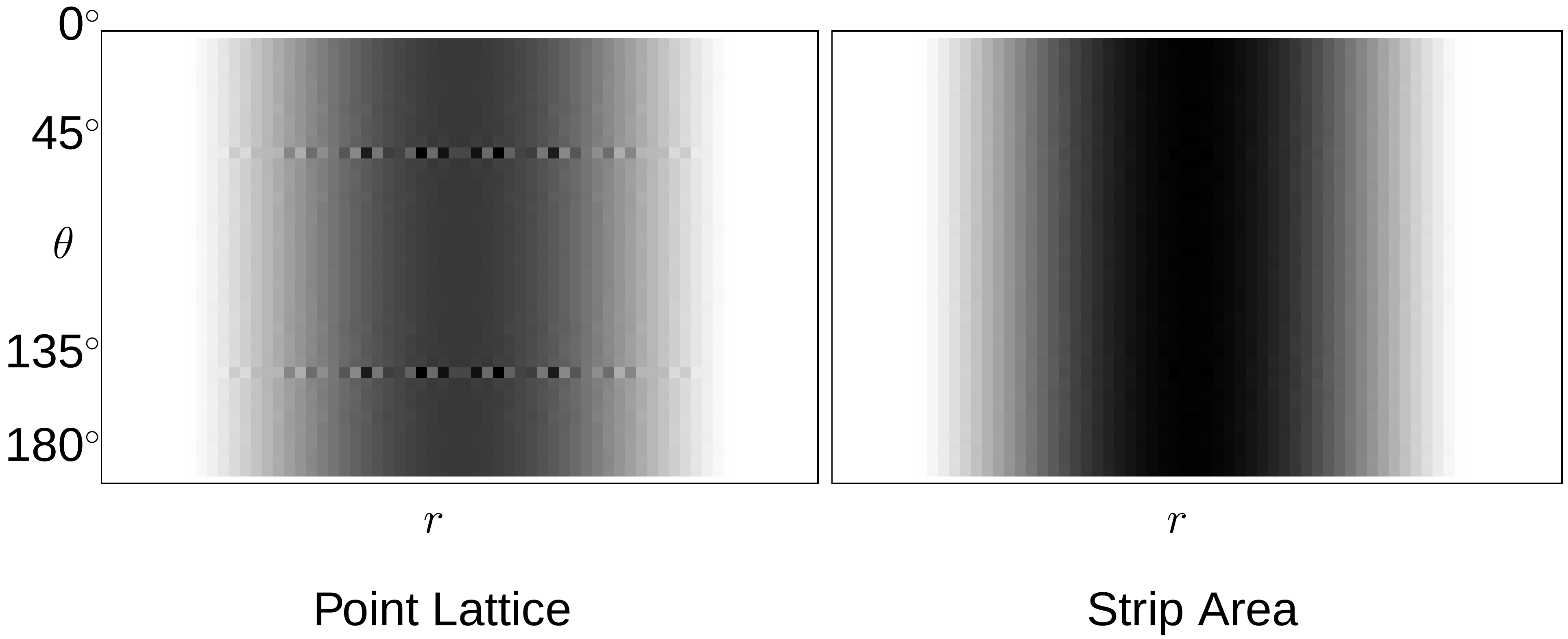
Point-Lattice Projector/Backprojector



a_{ij}'s determined by linear interpolation

Point-Lattice Artifacts

Projections (sinograms) of uniform disk object:



Choice 2. System Model

System matrix $\boldsymbol{A} = \{a_{ij}\}$ elements:

$$a_{ij} = \text{P[decay in the } j\text{th pixel is recorded by the } i\text{th detector unit]}$$

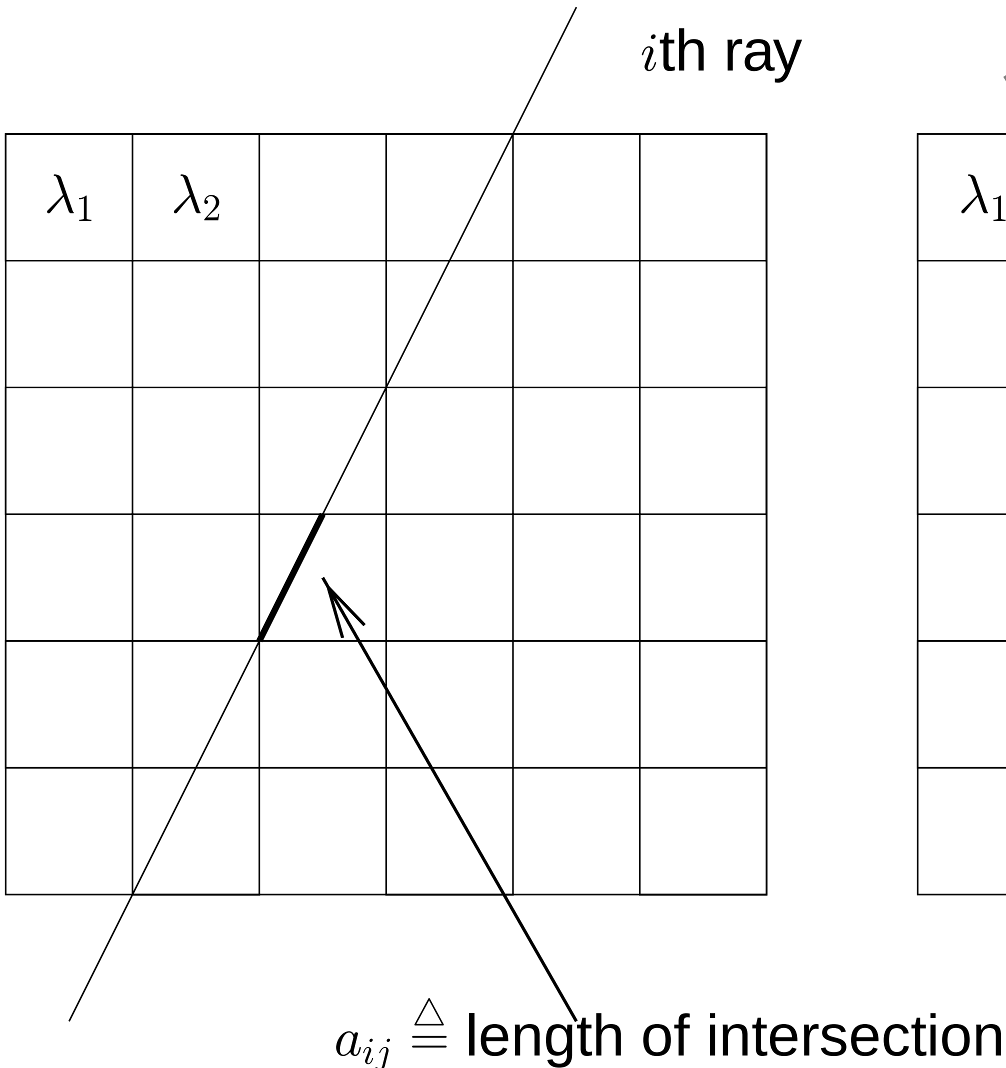
Physical effects

- scanner **geometry**
- solid angles
- detector efficiency
- attenuation
- scatter
- collimation
- detector response
- dwell time at each angle
- dead-time losses
- positron range
- noncolinearity
- ...

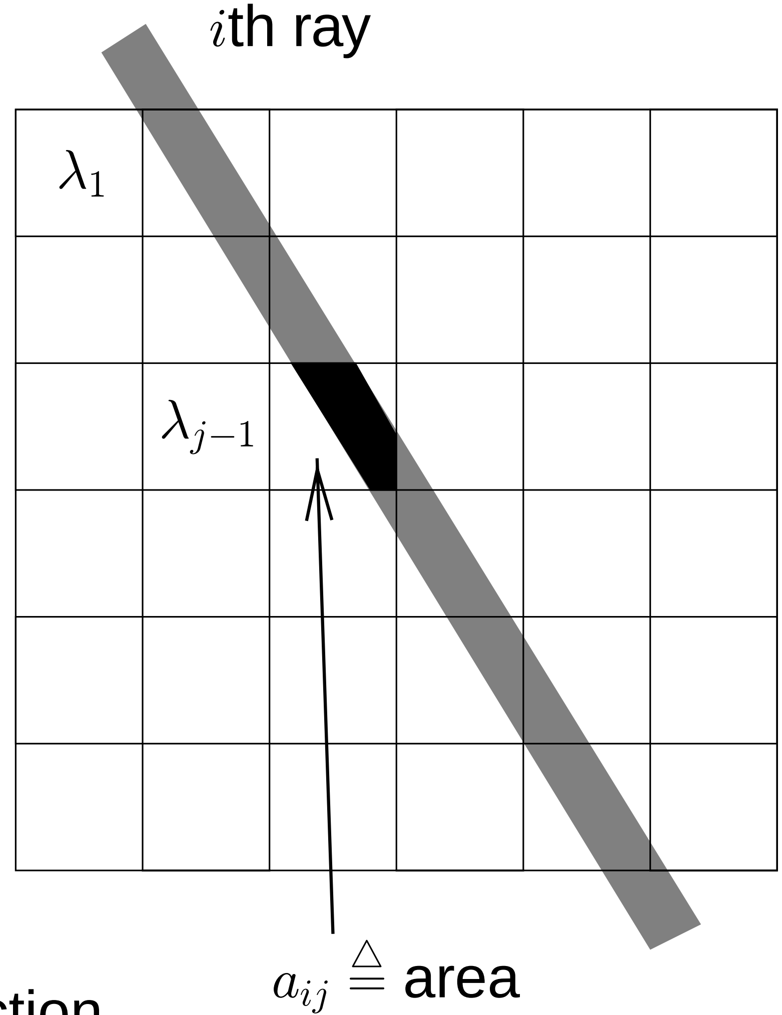
Considerations

- Accuracy vs computation and storage vs compute-on-fly
- Model uncertainties
  (*e.g.* calculated scatter probabilities based on noisy attenuation map)
- Artifacts due to over-simplifications

“Line Length” System Model



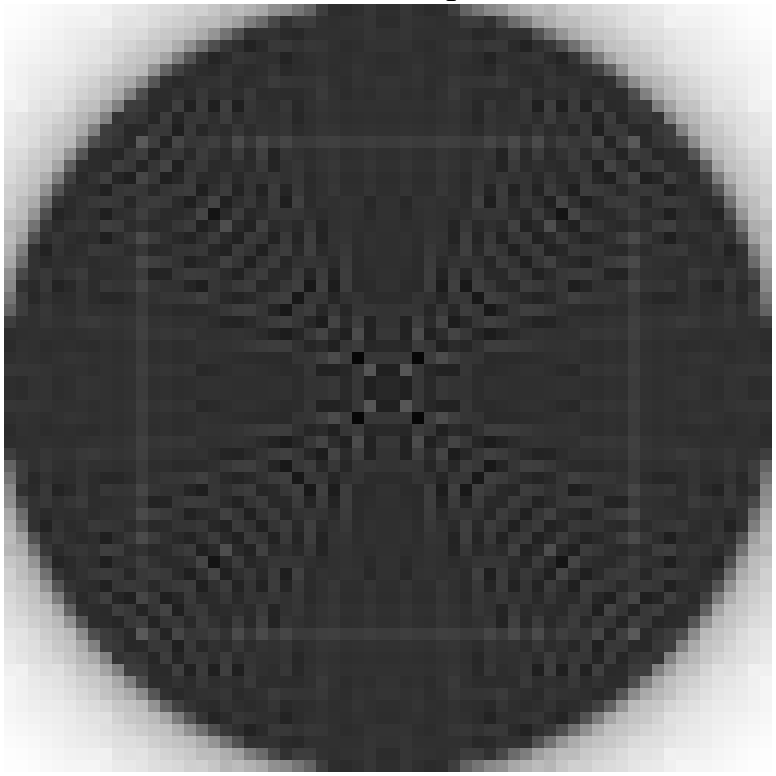
“Strip Area” System Model



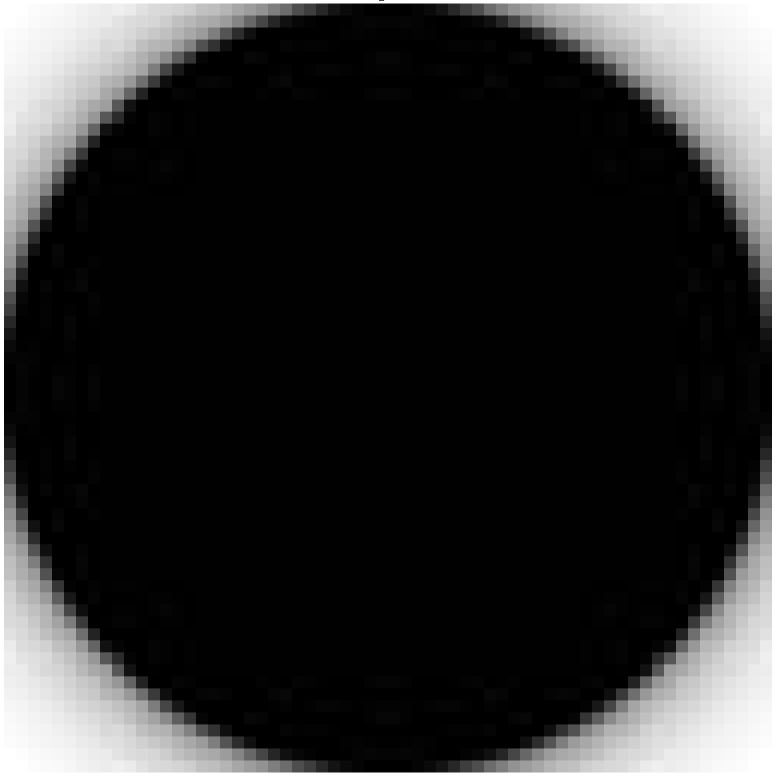
Sensitivity Patterns

$$\sum_{i=1}^{n_d} a_{ij} \approx s(\underline{x}_j) = \sum_{i=1}^{n_d} s_i(\underline{x}_j)$$

Line Length



Strip Area



Forward- / Back-projector "Pairs"

Forward projection (image domain to projection domain):

$$E[Y_i] = \int s_i(\vec{x}) \lambda(\vec{x}) \, d\vec{x} = \sum_{j=1}^{n_p} a_{ij} \lambda_j = [\boldsymbol{A} \underline{\lambda}]_i, \quad \text{or} \quad E[\underline{Y}] = \boldsymbol{A} \underline{\lambda}$$

Backprojection (projection domain to image domain):

$$\boldsymbol{A}' \underline{y} = \left\{ \sum_{i=1}^{n_d} a_{ij} y_i \right\}_{j=1}^{n_p}$$

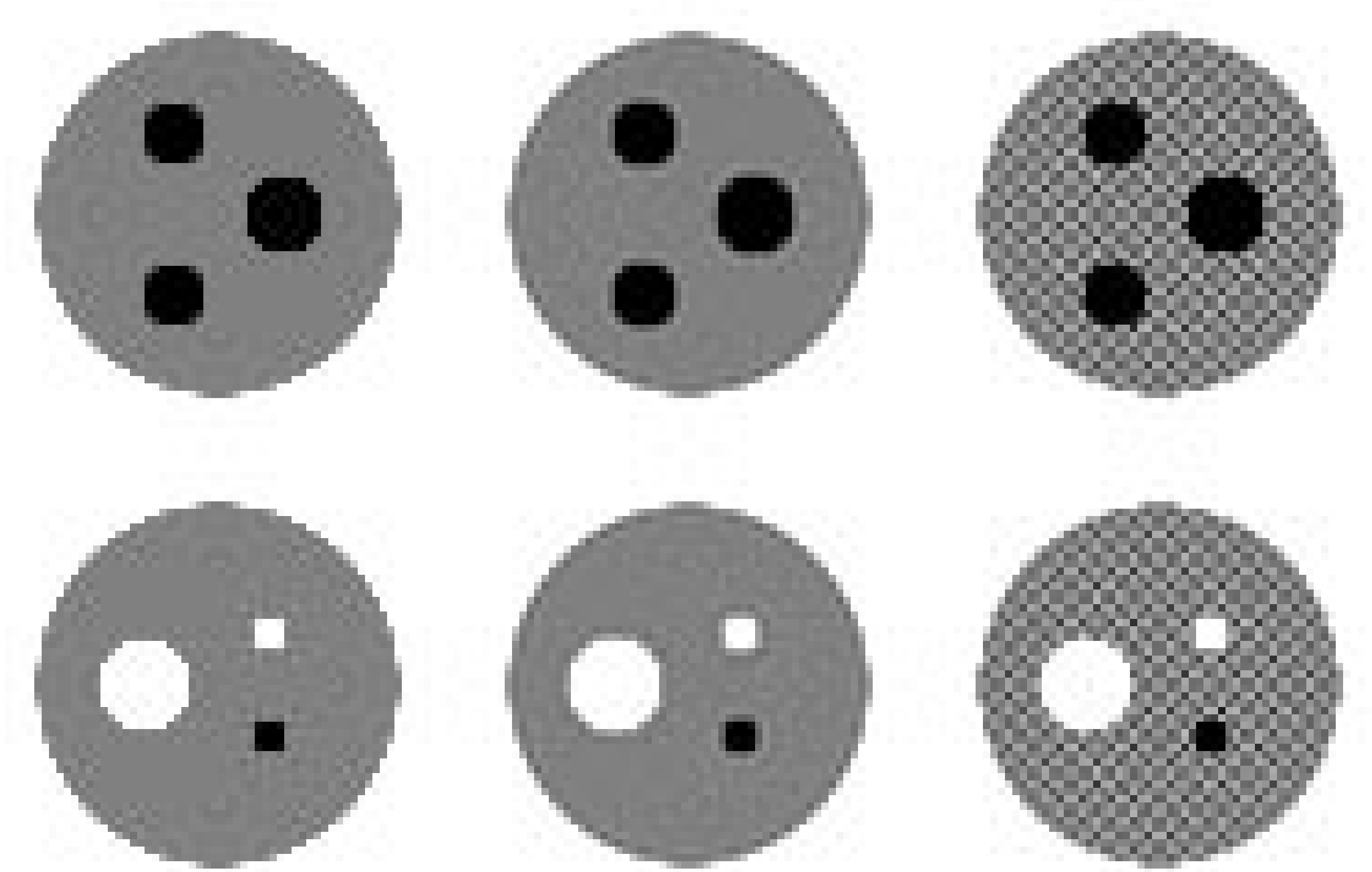
Often $\boldsymbol{A}'$ is implemented as $\boldsymbol{B} \underline{y}$ for some "backprojector" $\boldsymbol{B} \neq \boldsymbol{A}'$

Least-squares solutions (for example):

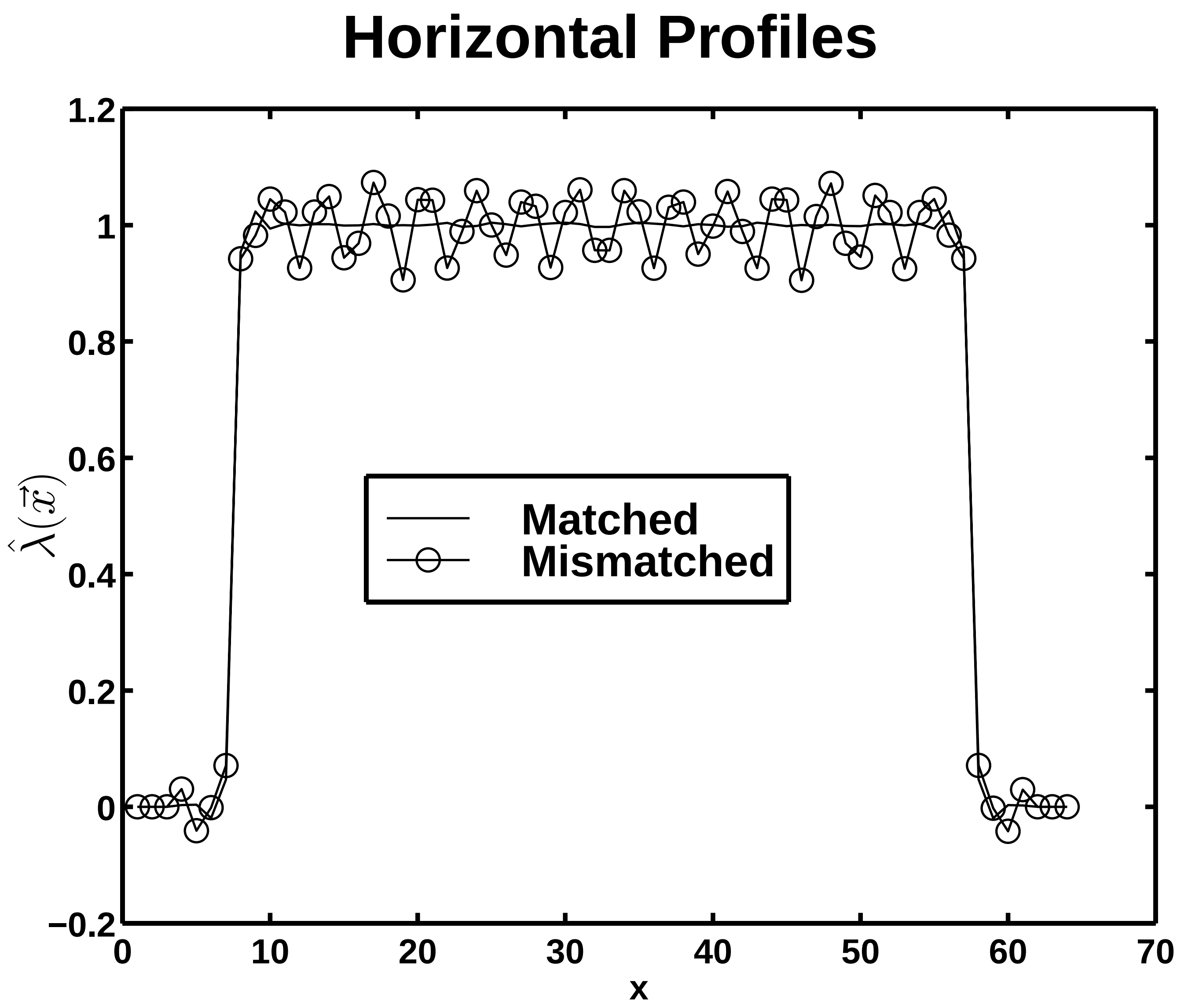
$$\underline{\hat{\lambda}} = [\boldsymbol{A}' \boldsymbol{A}]^{-1} \boldsymbol{A}' \underline{y} \neq [\boldsymbol{B} \boldsymbol{A}]^{-1} \boldsymbol{B} \underline{y}$$

Mismatched Backprojector $B \neq A'$ (3D PET)

$\underline{\lambda}$	$\hat{\underline{\lambda}}$ (PWLS-CG)	$\hat{\underline{\lambda}}$ (PWLS-CG)



($64 \times 64 \times 4$)	Matched	Mismatched


Horizontal Profiles
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
0
10
20
30
40
50
60
70
x
$\hat{\lambda}(\vec{x})$
Matched
Mismatched

Choice 3. Statistical Models

After modeling the system physics, we have a deterministic "model:"

$$\underline{Y} \approx E[\underline{Y}] = \boldsymbol{A}\underline{\lambda} + \underline{r}.$$

Statistical modeling is concerned with the " $\approx$ " aspect.

Random Phenomena

- Number of tracer atoms injected N
- Spatial locations of tracer atoms $\{\vec{X}_k\}_{k=1}^N$
- Time of decay of tracer atoms $\{T_k\}_{k=1}^N$
- Positron range
- Emission angle
- Photon absorption
- Compton scatter
- Detection $S_k \neq 0$
- Detector unit $\{S_k\}_{i=1}^{n_d}$
- Random coincidences
- Deadtime losses
- ...

Statistical Model Considerations

- More accurate models:
  - can lead to lower variance images,
  - can reduce bias
  - may incur additional computation,
  - may involve additional algorithm complexity
  (*e.g.* proper transmission Poisson model has nonconcave log-likelihood)
- Statistical model errors (*e.g.* deadtime)
- Incorrect models (*e.g.* log-processed transmission data)

Statistical Model Choices

- "None." Assume $\underline{Y} - \underline{r} = \boldsymbol{A}\underline{\lambda}$. "Solve algebraically" to find $\underline{\lambda}$.
- White Gaussian noise. Ordinary least squares: minimize $\|\boldsymbol{Y} - \boldsymbol{A}\underline{\lambda}\|^2$
- Non-White Gaussian noise. Weighted least squares: minimize

$$\|\boldsymbol{Y} - \boldsymbol{A}\underline{\lambda}\|^2_{\boldsymbol{W}} = \sum_{i=1}^{n_d} w_i \left(y_i - [\boldsymbol{A}\underline{\lambda}]_i\right)^2, \quad \text{where} \quad [\boldsymbol{A}\underline{\lambda}]_i \triangleq \sum_{j=1}^{n_p} a_{ij}\lambda_j$$

- Ordinary Poisson model (ignoring or precorrecting for background)

$$Y_i \sim \text{Poisson}\{[\boldsymbol{A}\underline{\lambda}]_i\}$$

- Poisson model

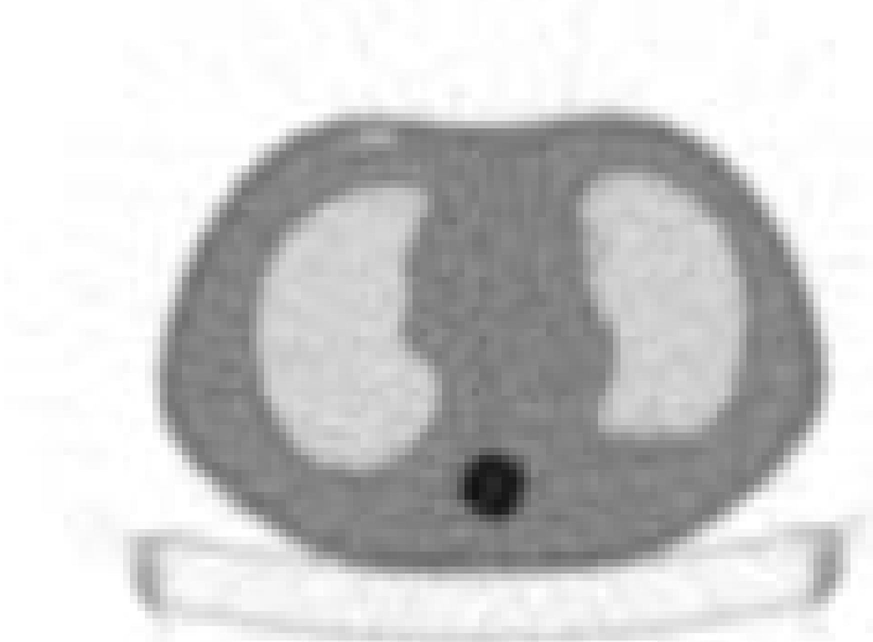
$$Y_i \sim \text{Poisson}\{[\boldsymbol{A}\underline{\lambda}]_i + r_i\}$$

- Shifted Poisson model (for randoms precorrected PET)

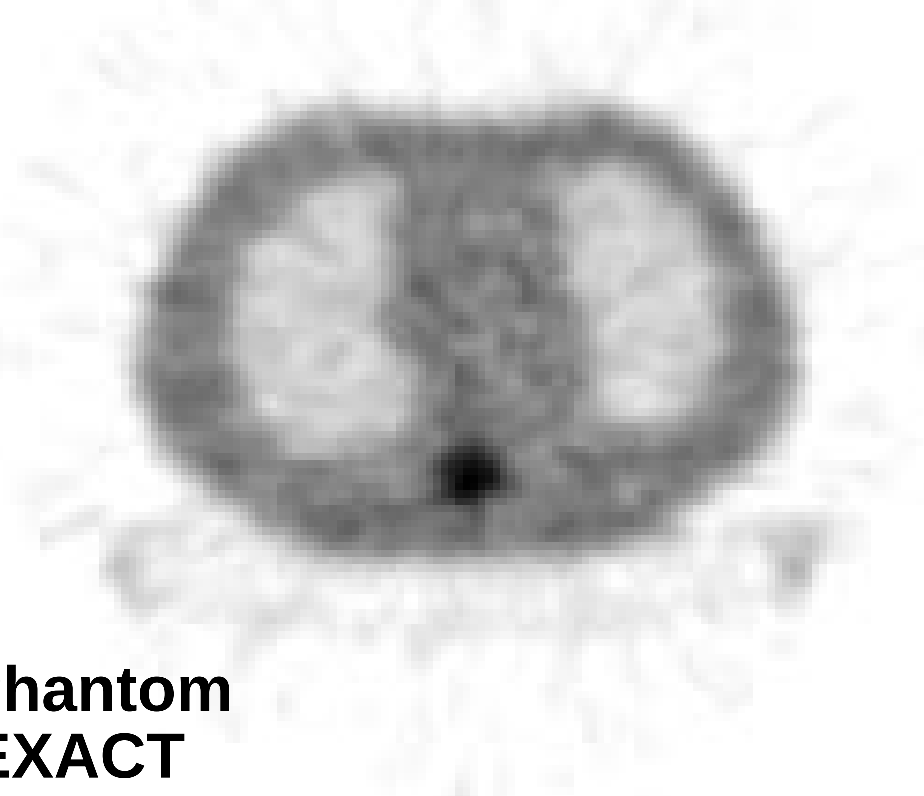
$$Y_i = Y_i^{\text{prompt}} - Y_i^{\text{delay}} \sim \text{Poisson}\{[\boldsymbol{A}\underline{\lambda}]_i + 2r_i\} - 2r_i$$

Transmission Phantom

FBP 7hour



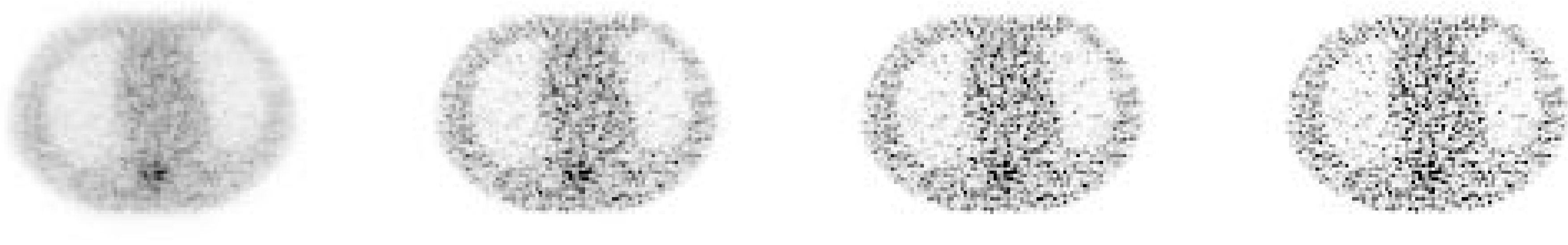
FBP 12min



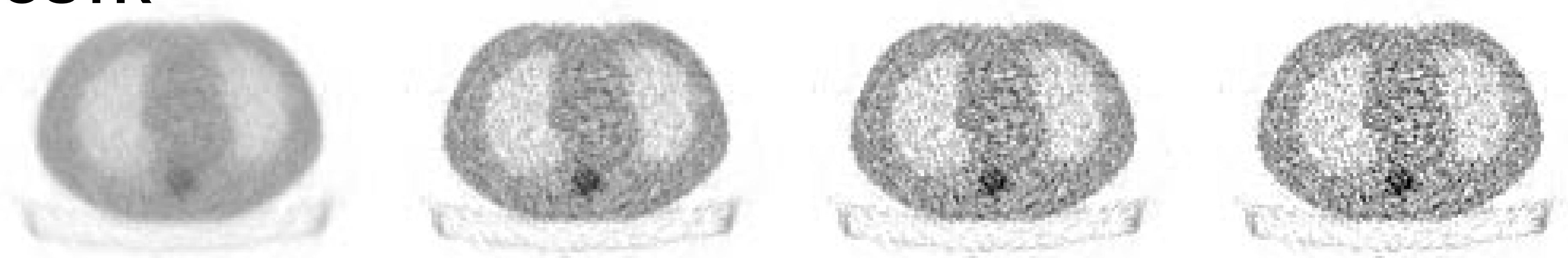
Thorax Phantom
ECAT EXACT

Effect of statistical model

OSEM



OSTR



Iteration: 1 3 5 7

Choice 4. Objective Functions

Components:

- *Data-fit* term
- *Regularization* term (and regularization parameter β)
- Constraints (*e.g.* nonnegativity)

$$\Phi(\underline{\lambda}) = \text{DataFit}(\underline{Y}, \boldsymbol{A}\underline{\lambda} + \underline{r}) - \beta \cdot \text{Roughness}(\underline{\lambda})$$

$$\boxed{\hat{\underline{\lambda}} \triangleq \arg\max_{\underline{\lambda} \geq \underline{0}} \Phi(\underline{\lambda})}$$

"Find the image that 'best fits' the sinogram data"

Actually *three* choices to make for Choice 4 ...

Distinguishes "statistical methods" from "algebraic methods" for "$\underline{Y} = \boldsymbol{A}\underline{\lambda}$."

Why Objective Functions?

(vs "procedure" *e.g.* adaptive neural net with wavelet denoising)

Theoretical reasons

ML is based on maximizing an objective function: the log-likelihood

- ML is asymptotically consistent
- ML is asymptotically unbiased
- ML is asymptotically efficient (under true statistical model...)
- Penalized-likelihood achieves uniform CR bound asymptotically

Practical reasons

- Stability of estimates (if Φ and algorithm chosen properly)
- Predictability of properties (despite nonlinearities)
- Empirical evidence (?)

Choice 4.1: Data-Fit Term

- Least squares, weighted least squares (quadratic data-fit terms)
- Reweighted least-squares
- Model-weighted least-squares
- Norms robust to outliers
- Log-likelihood of statistical model. Poisson case:

$$L(\underline{\lambda}; \underline{Y}) = \log P[\underline{Y} = \underline{y}; \underline{\lambda}] = \sum_{i=1}^{n_d} y_i \log([\boldsymbol{A}\underline{\lambda}]_i + r_i) - ([\boldsymbol{A}\underline{\lambda}]_i + r_i) - \log y_i!$$

Poisson probability mass function (PMF):

$$P[\underline{Y} = \underline{y}; \underline{\lambda}] = \Pi_{i=1}^{n_d} e^{-\bar{y}_i} \bar{y}_i^{y_i} / y_i! \text{ where } \underline{\bar{y}} \triangleq \boldsymbol{A}\underline{\lambda} + \underline{r}$$

Considerations

- Faithfulness to statistical model vs computation
- Effect of statistical modeling errors

Choice 4.2: Regularization

Forcing too much "data fit" gives noisy images
Ill-conditioned problems: small data noise causes large image noise

Solutions:

- **Noise-reduction methods**
  - Modify the *data* (prefilter or extrapolate sinogram data)
  - Modify an *algorithm* derived for an ill-conditioned problem (stop before converging, post-filter)
- **True regularization methods**
  Redefine the *problem* to eliminate ill-conditioning
  - Use bigger pixels (fewer basis functions)
  - Method of sieves (constrain image roughness)
  - Change objective function by <u>adding a roughness penalty</u> / prior

$$R(\underline{\lambda}) = \sum_{j=1}^{n_p} \sum_{k \in \mathcal{N}_j} \psi(\lambda_j - \lambda_k)$$

Noise-Reduction vs True Regularization

Advantages of **“noise-reduction”** methods

- Simplicity (?)
- Familiarity
- Appear less subjective than using penalty functions or priors
- Only fiddle factors are # of iterations, amount of smoothing
- Resolution/noise tradeoff usually varies with iteration
  (stop when image looks good - in principle)

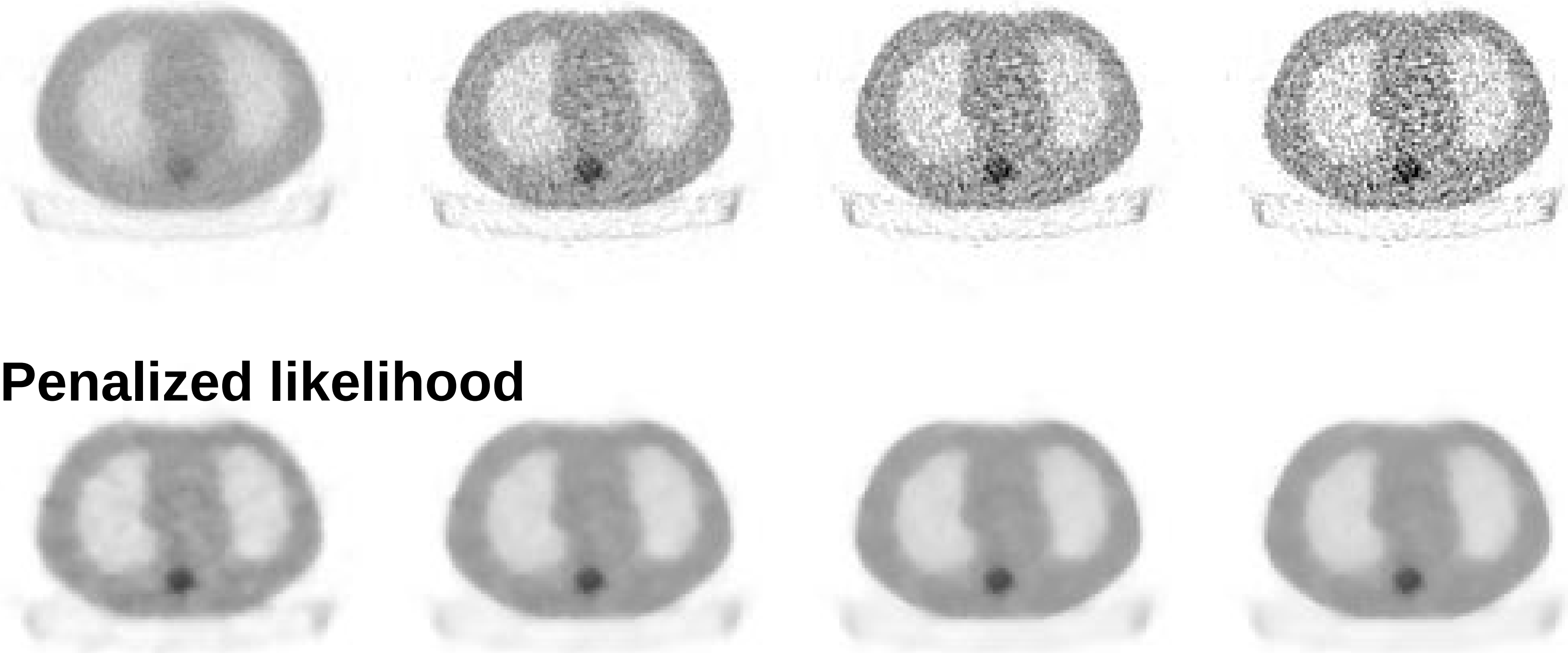
Advantages of **true regularization** methods

- Stability
- Predictability
- Resolution can be made object independent
- Controlled resolution (*e.g.* spatially uniform, edge preserving)
- Start with (*e.g.*) FBP image $\Rightarrow$ reach solution faster.

Unregularized vs Regularized Reconstruction

ML (unregularized) **(OSTR)**

Penalized likelihood



Iteration: 1 3 5 7

Roughness Penalty Function Considerations

$$R(\underline{\lambda}) = \sum_{j=1}^{n_p} \sum_{k \in \mathcal{N}_j} \psi(\lambda_j - \lambda_k)$$

- Computation
- Algorithm complexity
- Uniqueness of maximum of Φ
- Resolution properties (edge preserving?)
- # of adjustable parameters
- Predictability of properties (resolution and noise)

Choices

- separable vs nonseparable
- quadratic vs nonquadratic
- convex vs nonconvex

This topic is actively debated!

Nonseparable Penalty Function Example

Example

x_1	x_2	x_3
x_4	x_5	

$$R(\underline{x}) = (x_2 - x_1)^2 + (x_3 - x_2)^2 + (x_5 - x_4)^2$$
$$+ (x_4 - x_1)^2 + (x_5 - x_2)^2$$

2	2	2
2	1	

$R(\underline{x}) = 1$

3	3	1
2	2	

$R(\underline{x}) = 6$

1	3	1
2	2	

$R(\underline{x}) = 10$

Rougher images $\Rightarrow$ greater $R(\underline{x})$

Penalty Functions: Quadratic vs Nonquadratic

Phantom   Quadratic Penalty   Huber Penalty

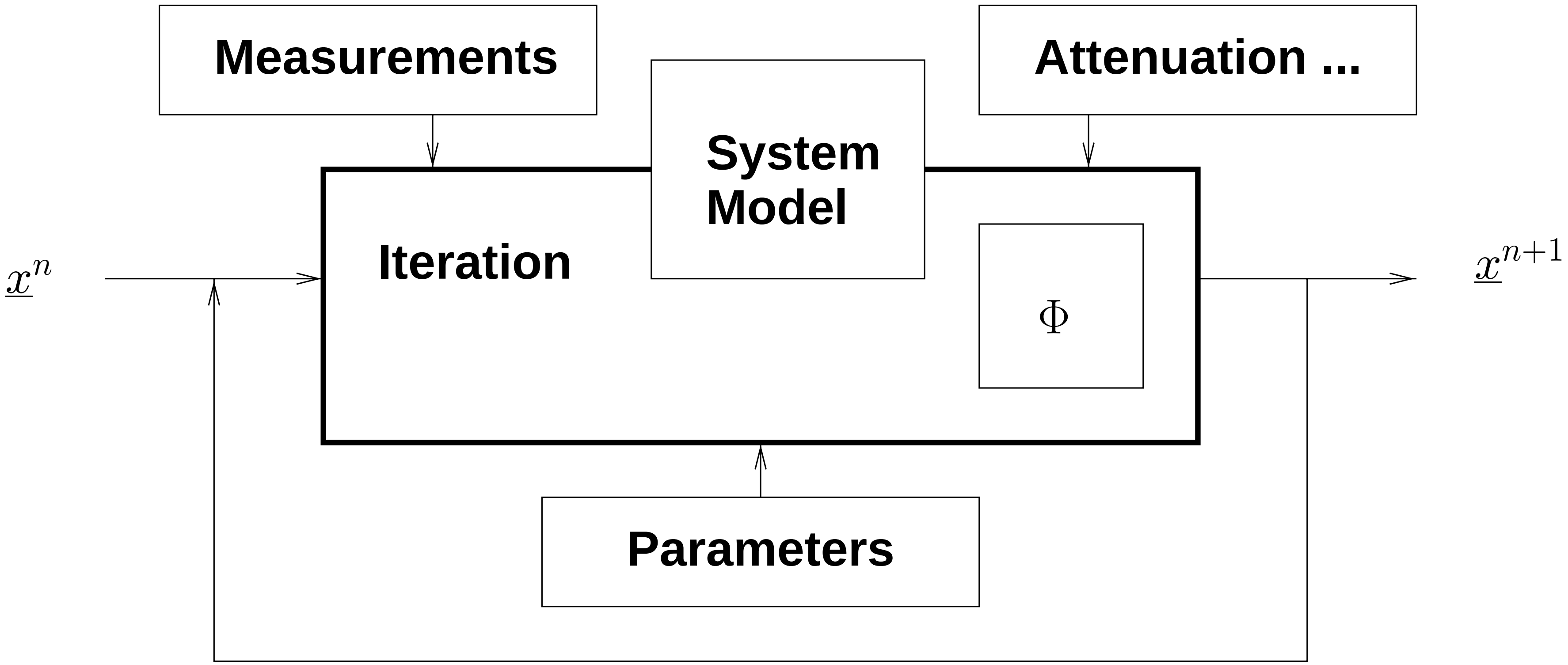
Summary of Modeling Choices

1. Object parameterization: $\lambda(\underline{x})$ vs $\underline{\lambda}$
2. System physical model: $s_i(\underline{x})$
3. Measurement statistical model $Y_i \sim \boxed{?}$
4. Objective function: data-fit / regularization / constraints

Reconstruction Method = Objective Function + Algorithm

5. Iterative algorithm
   ML-EM, MAP-OSL, PL-SAGE, PWLS+SOR, PWLS-CG, . . .

Choice 5. Algorithms



Deterministic iterative mapping: $\underline{x}^{n+1} = \mathcal{M}(\underline{x}^n)$

All algorithms are imperfect. No single best solution.

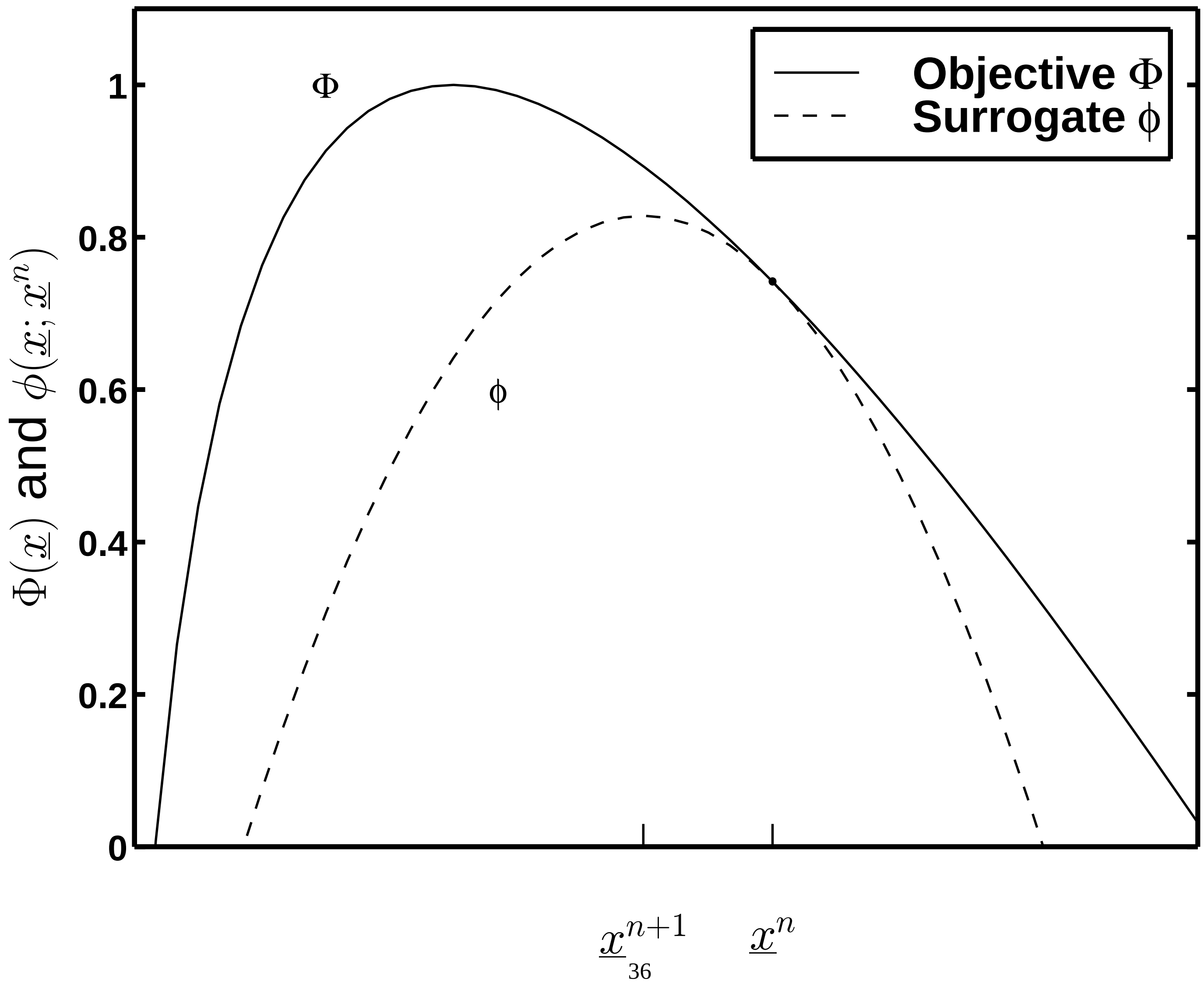
Ideal Algorithm

$$\underline{x}^\star \triangleq \arg\max_{\underline{x} \geq \underline{0}} \Phi(\underline{x}) \quad \text{(global maximum)}$$

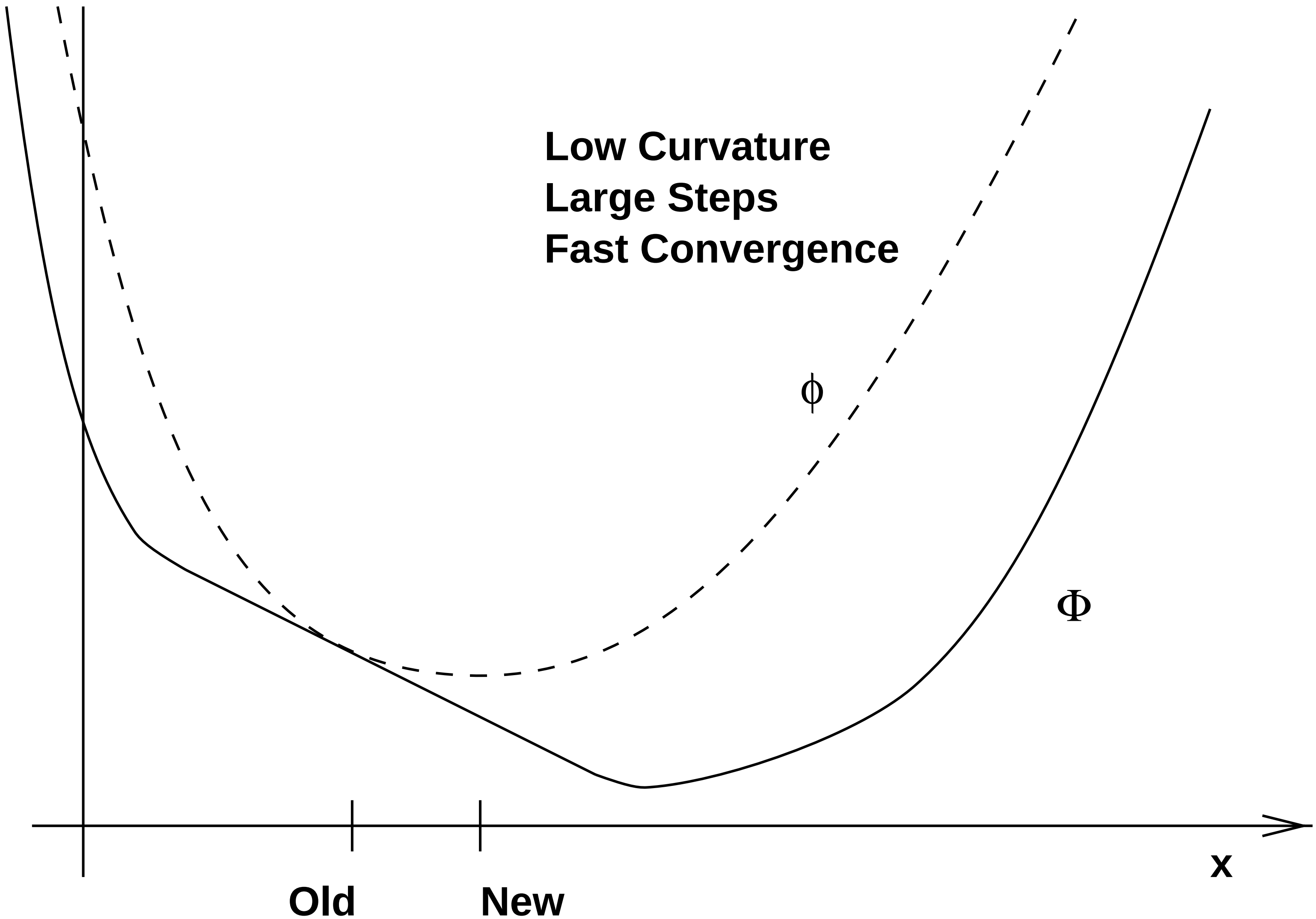
stable and convergent	$\{\underline{x}^n\}$ converges to $\underline{x}^\star$ if run indefinitely
converges quickly	$\{\underline{x}^n\}$ gets "close" to $\underline{x}^\star$ in just a few iterations
globally convergent	$\lim_n \underline{x}^n$ independent of starting image
fast	requires minimal computation per iteration
robust	insensitive to finite numerical precision
user friendly	nothing to adjust (*e.g.* acceleration factors)
monotonic	$\Phi(\underline{x}^n)$ increases every iteration
parallelizable	(when necessary)
simple	easy to program and debug
flexible	accommodates any type of system model (matrix stored by row or column or projector/backprojector)

Choices: forgo one or more of the above

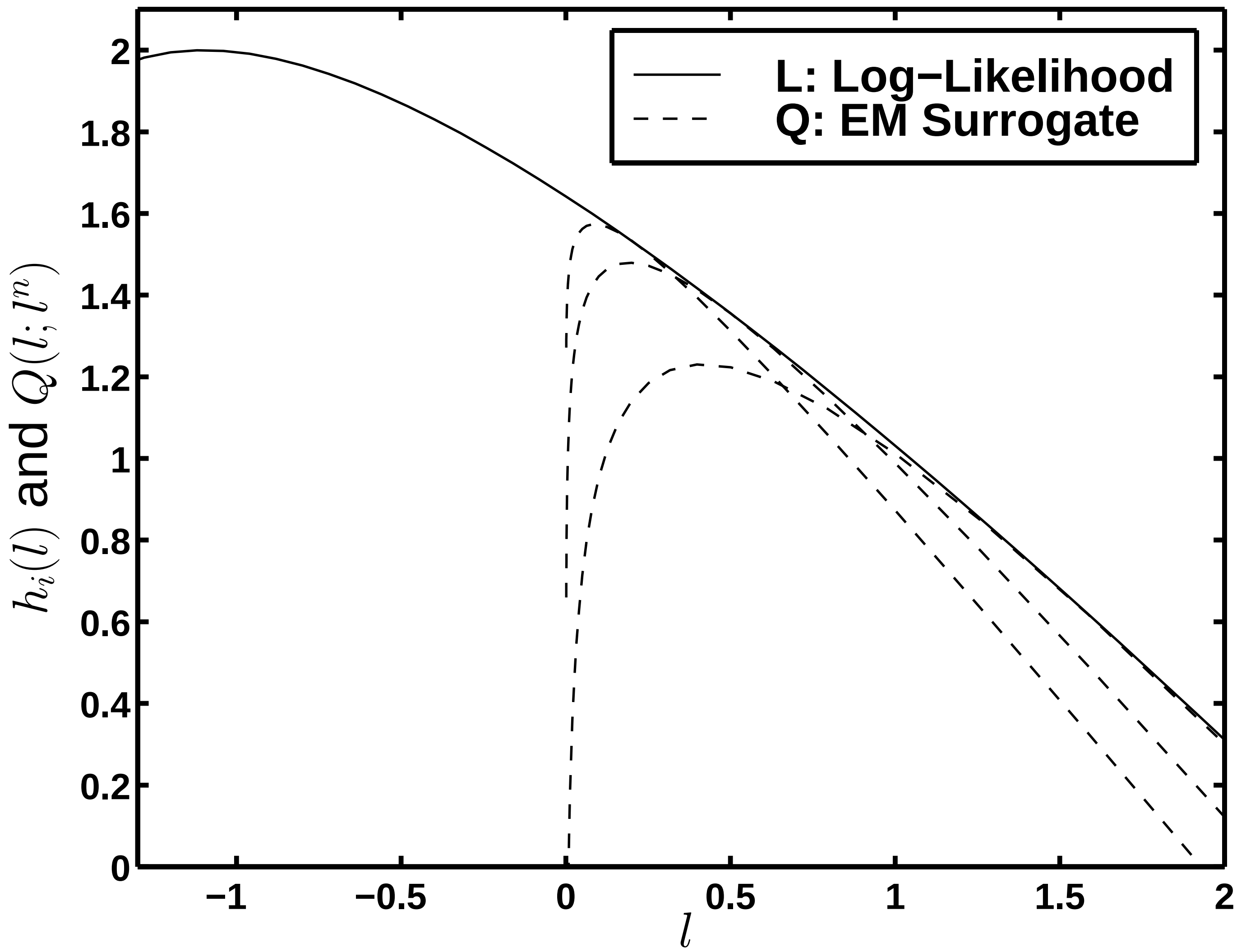
Optimization Transfer Illustrated



Convergence Rate: Fast



Slow Convergence of EM



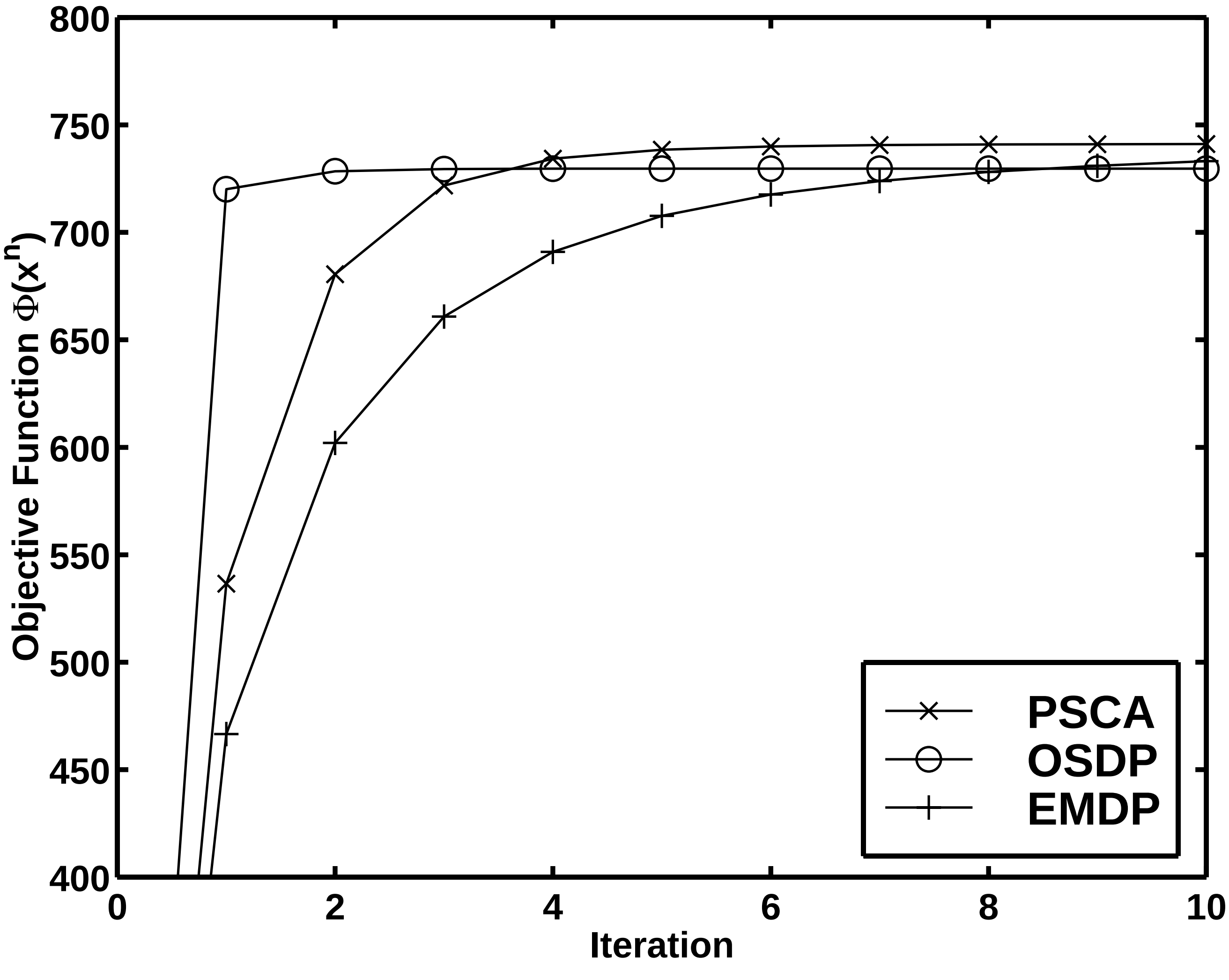
Paraboloidal Surrogates

- Not separable (unlike EM)
- Not self-similar (unlike EM)
- Poisson log-likelihood replaced by a series of least squares problems.
- Maximize each quadratic problem easily using coordinate ascent.

Advantages
- Fast converging
- Instrinsically monotone global convergence
- Fairly simple to derive / implement
- Nonnegativity easy (with coordinate ascent)

Disadvantages
- Coordinate ascent $\therefore$ column-stored system matrix

Convergence rate: PSCA vs EM



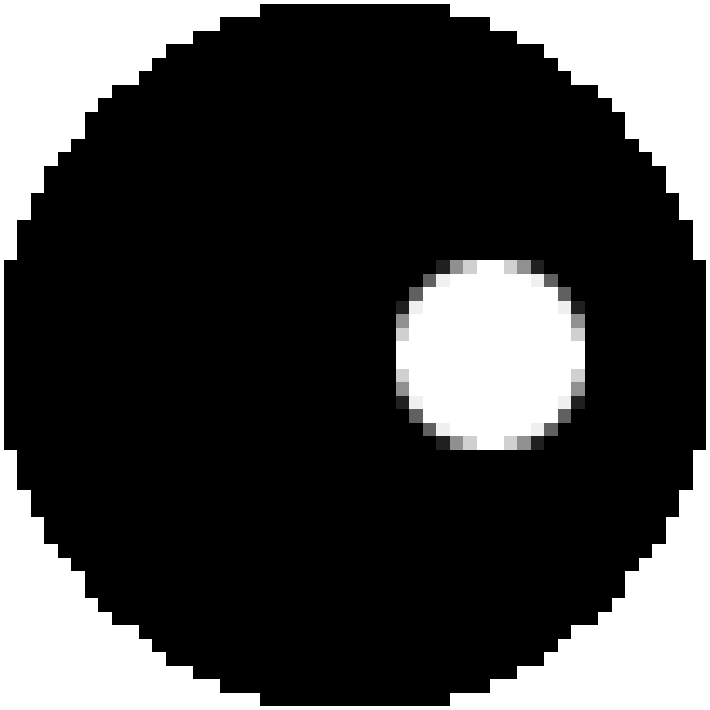
Ordered Subsets Algorithms

- The *backprojection* operation appears in every algorithm.
- Intuition: with half the angular sampling, the backprojection would look fairly similar.
- To "OS-ize" an algorithm, replace all backprojections with partial sums.

Problems with OS-EM

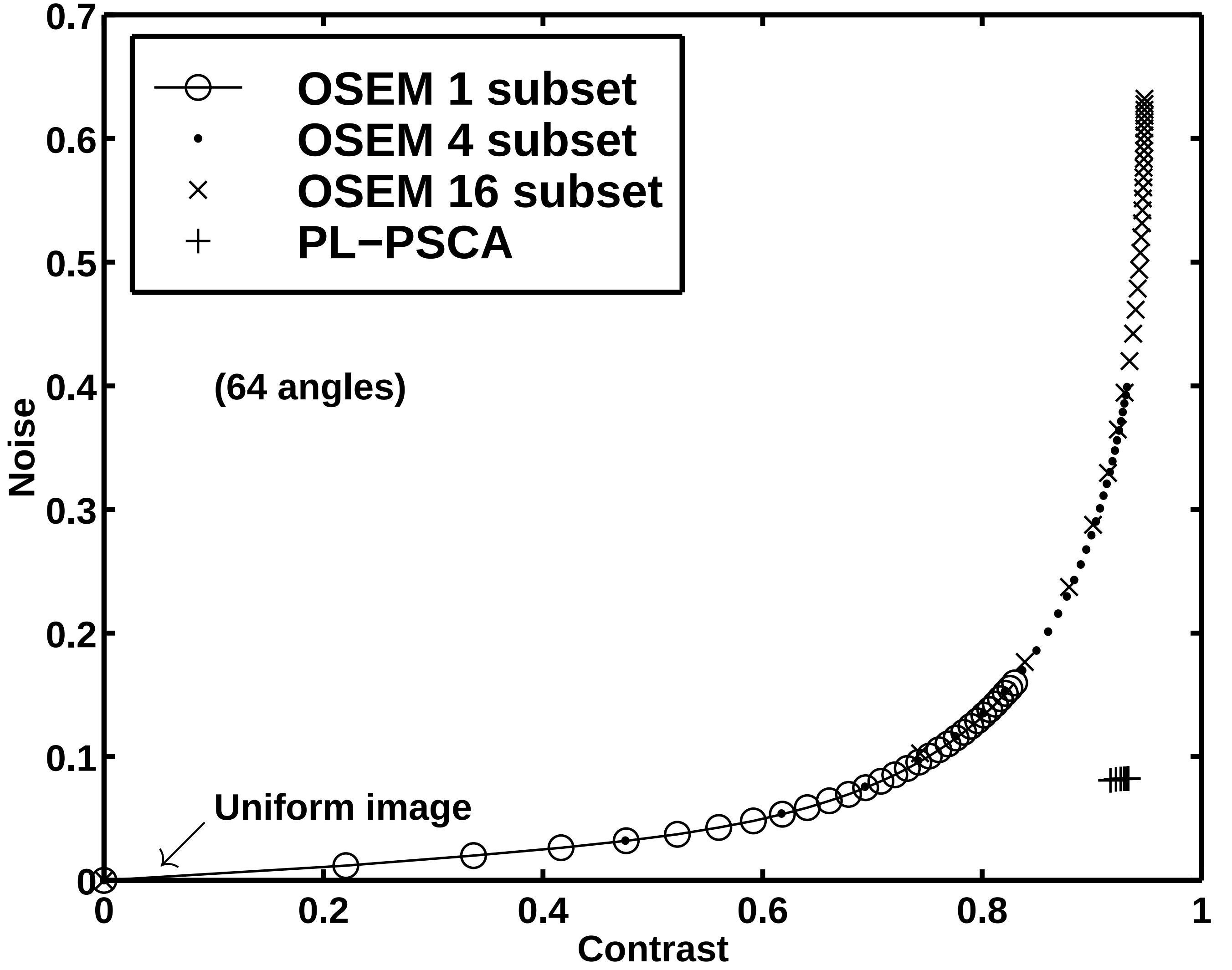
- Non-monotone
- Does not converge (may cycle)
- Byrne's RBBI approach only converges for consistent (noiseless) data
- $\therefore$ unpredictable
  - What resolution after n iterations?
  - Object-dependent, spatially nonuniform
  - What variance after n iterations?
  - ROI variance? (*e.g.* for Huesman's WLS kinetics)

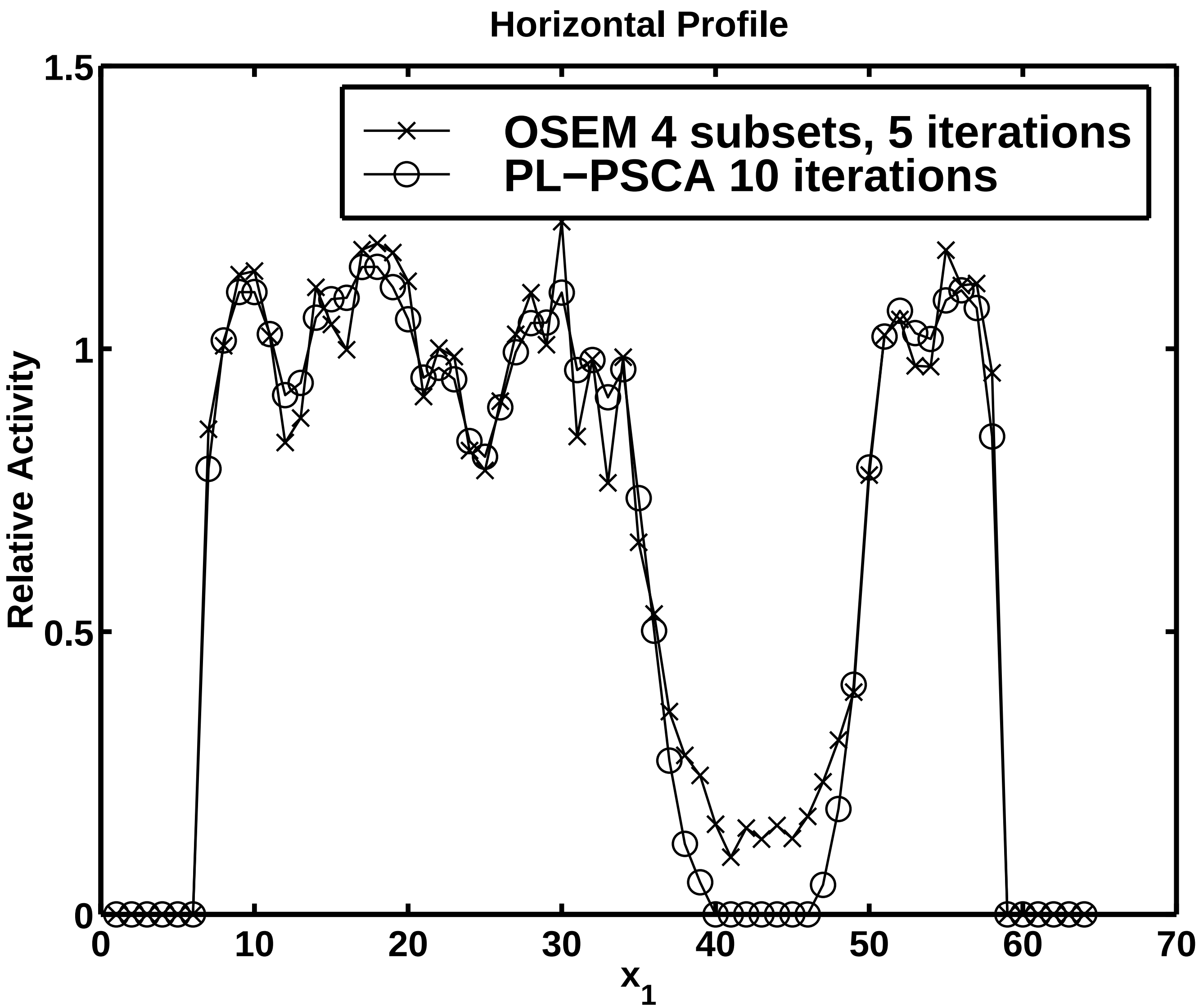
OSEM vs Penalized Likelihood



- 64×62 image
- 66×60 sinogram
- 10^6 counts
- 15% randoms/scatter
- uniform attenuation
- contrast in cold region
- within-region σ opposite side

Contrast-Noise Results




Horizontal Profile
OSEM 4 subsets, 5 iterations
PL−PSCA 10 iterations
Relative Activity
1.5
1
0.5
0
0
10
20
30
40
50
60
70
x1

Noise Properties

$$\mathrm{Cov}\{\underline{\hat{x}}\} \approx \left[\nabla^{20}\Phi\right]^{-1} \left[\nabla^{11}\Phi\right] \mathrm{Cov}\{\underline{Y}\} \left[\nabla^{11}\Phi\right]^{T} \left[\nabla^{20}\Phi\right]^{-1}$$

- Enables prediction of noise properties
- Useful for computing ROI variance for kinetic fitting

IEEE Tr. Image Processing, 5(3):493 1996

Summary

- General principles of statistical image reconstruction
- Optimization transfer
- Principles apply to transmission reconstruction
- Predictability of resolution / noise and controlling spatial resolution argues for regularized objective-function
- Still work to be done...

An Open Problem

Still no algorithm with all of the following properties:

- Nonnegativity easy
- Fast converging
- Intrinsically monotone global convergence
- Accepts any type of system matrix
- Parallelizable