Ray-weighted constrained-conjugate-gradient tomographic reconstruction for security applications

Jeffrey S. Kallman and Stephen G. Azevedo
Lawrence Livermore National Laboratory
LLNL-PRES-562193
IM# 623919

May 22, 2013

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Summary/Future work

- Summary
  - We have implemented and accelerated a constrained conjugate gradient algorithm (CCG)* using the adjoint method for computing the error gradient and incorporating the capability to use ray weighting.
  - We are investigating ray weighting by
    - powers of ray transmission
      - There is theoretical justification for setting the power to 1**
      - We have found that values greater than 1 sometimes yield more uniform reconstructions for homogeneous materials.
      - We are still determining appropriate limits for the power and appropriate circumstances for using ray weighting
    - sigmoidal functions of ray transmission
      - Do not work as well as powers of ray transmission
- Future work
  - Demonstrate reduction in feature space size
  - Assess robustness across different types of clutter, threats, etc.

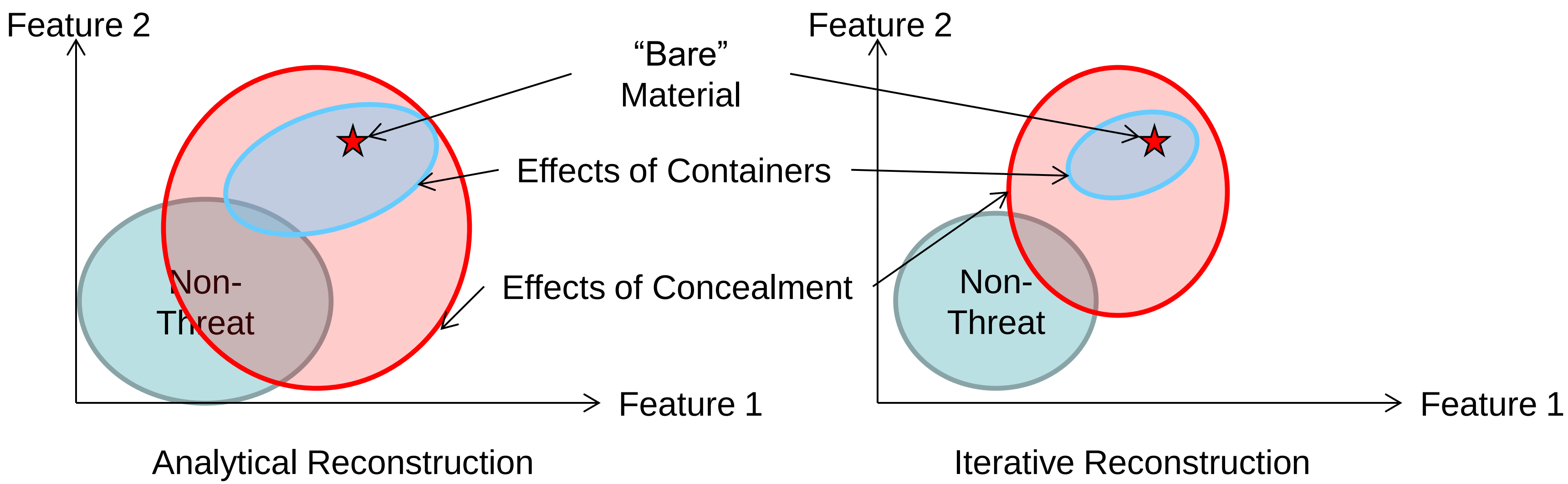
---

* D. M. Goodman, E. M. Johansson and T. W. Lawrence, "On applying the conjugate-gradient algorithm to image processing problems," *Multivariate Analysis: Future Directions*, Elsevier Science Publishers, 1993.

** See pages 536-7 and Appendix A of K. Sauer and C. Bouman, "A Local Update Strategy for Iterative Reconstruction from Projections," IEEE Trans. Sig. Proc., Vol. 41, No. 2, pp. 534-548, Feb. 1993.

Goal of this Iterative Reconstruction Work

- Show that iterative reconstruction techniques can reduce the effects of containers and concealment, and thus improve probability of detection / probability of false alarm.
- We sometimes refer to this spread as a cloud



Features can include x-ray attenuation coefficients, Zeff, density, texture, kurtosis

Error and Its Gradient

- Projection difference error is given by

$$E[\mu(\vec{r})] = \frac{1}{2}\sum_{m=1}^{M} w_m \left[I_m\left(s_{final}\right) - I_{m,observed}\left(s_{final}\right)\right]^2$$

where E is the error, μ is attenuation, r is position, M is the number of rays, m is the ray index, w is the ray weight, I is the ray intensity, s is the position along the ray.

- This error yields a Frechet derivative for every voxel of the form:

$$\nabla_f E\left(voxel_i\right) = \sum_{m=1}^{M} w_m I_m(s) \tilde{I}_m(s) P_{i,m}$$

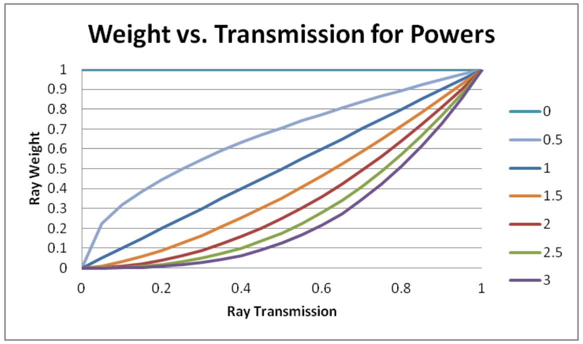
where $P_{i,m}$ is the projection of the ith voxel on the mth ray, $\tilde{I}_m(s)$ is the adjoint ray intensity, and the product $I_m(s)\tilde{I}_m(s)$ is a constant for each position s along ray m. See Appendix for the derivation of this Frechet derivative.

- This Frechet derivative can be used in a conjugate gradient algorithm.

Ray Weighting Policies

- Ray weighting policy is being investigated
  - Weighting by ray transmission to a power
    - From the literature
    - Works best so far

$$w_m = \left(\frac{I}{I_0} \right)^p$$

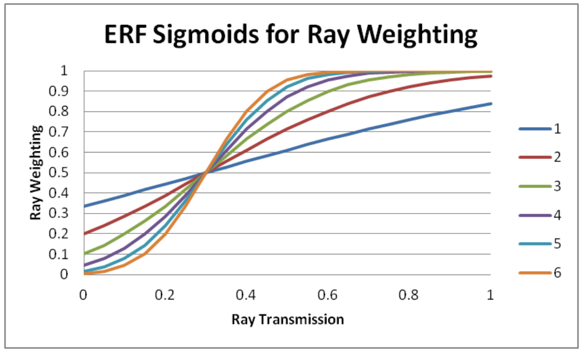


curve color indicates p

  - Weighting by sigmoidal function of ray transmission
    - Trying something different

$$w_m = 0.5 + 0.5\,erf\left(s\left(\frac{I}{I_0} - x \right) \right)$$

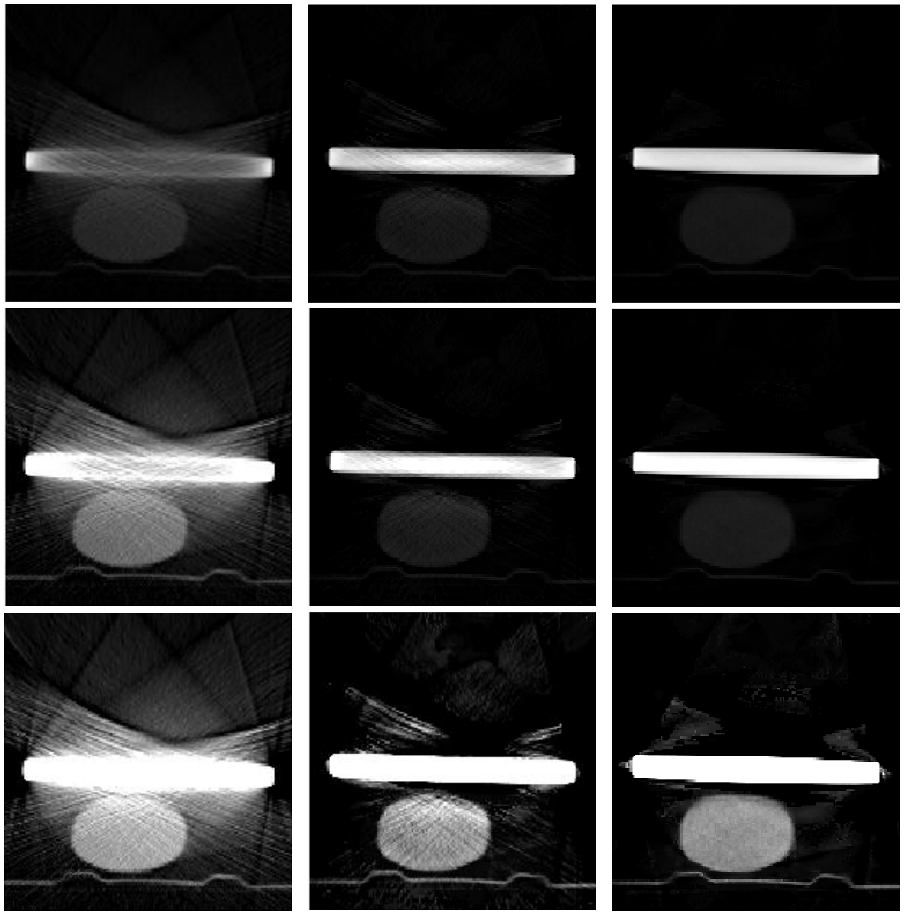
$$erf(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$$

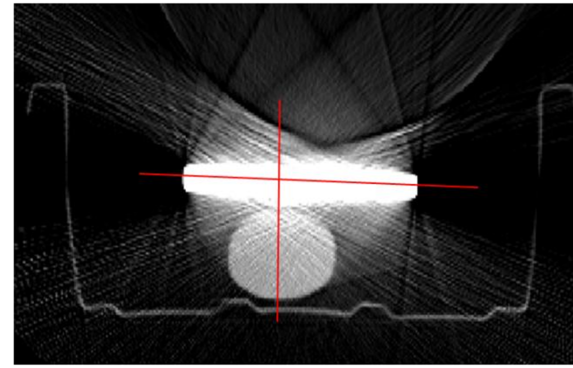


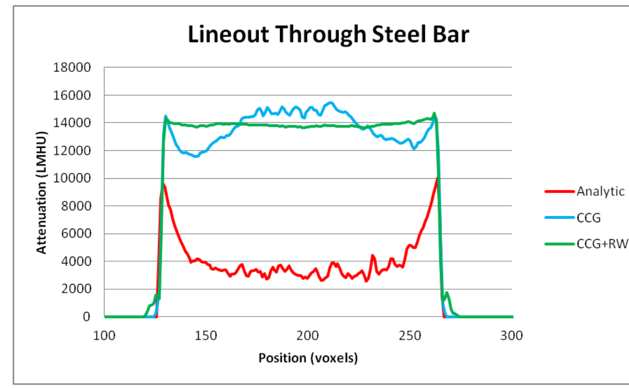
x = 0.3, curve color indicates s

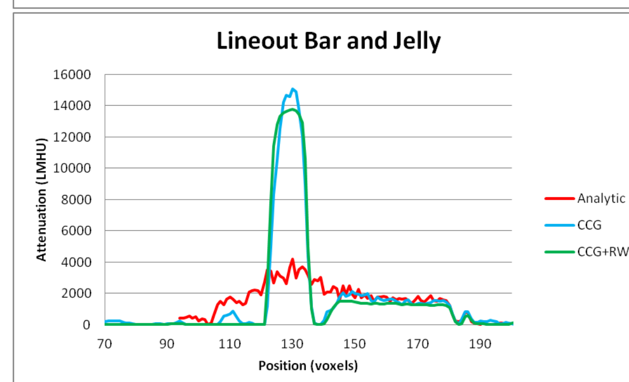
Example of Streak Artifacts and Their Reduction

	Analytic	CCG	CCG+ Ray Weighting*
Window Max at Max Steel			
Window Max at Mean Steel			
Window Max at Max Jelly			



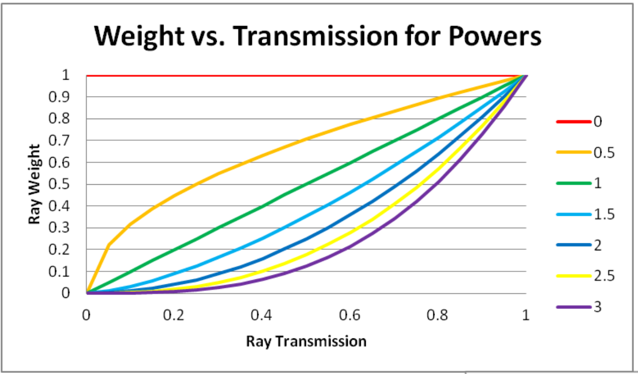


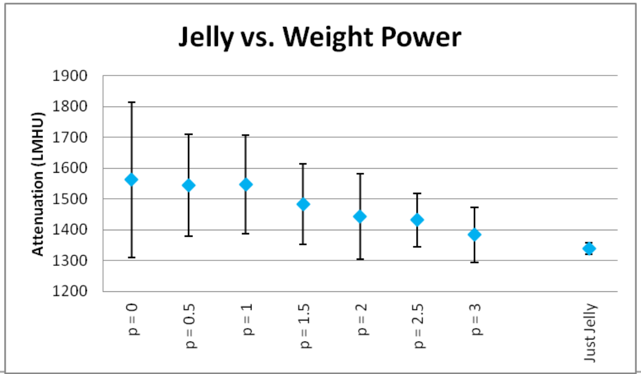


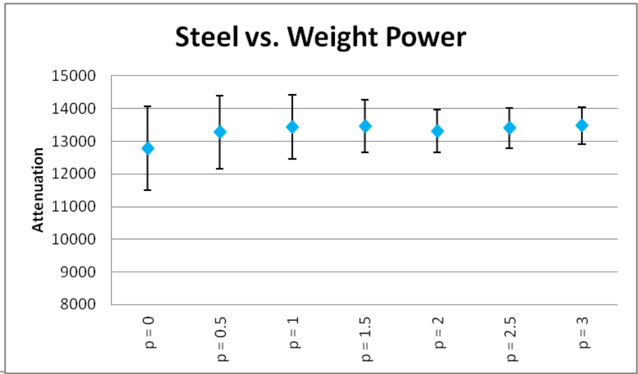


* Using ray transmission power weighting. Power = 2.5

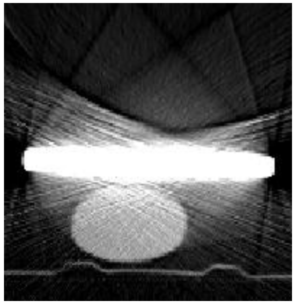
Power Law Ray Weighting Examining Jelly

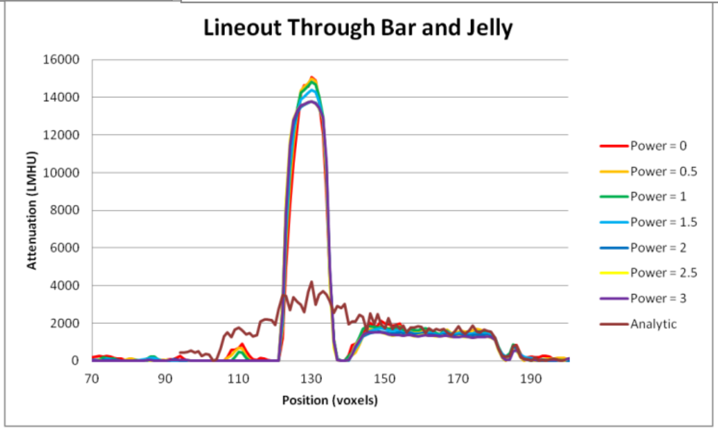


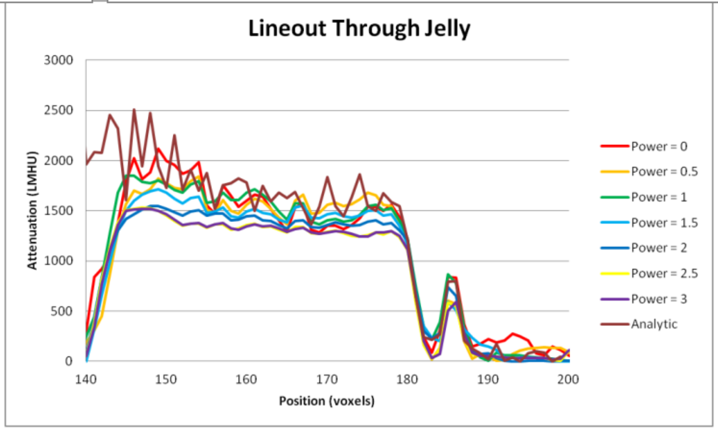




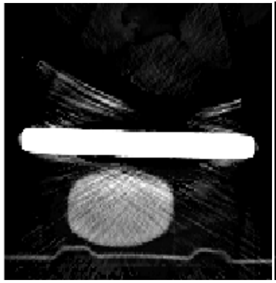
Analytic Reconstruction





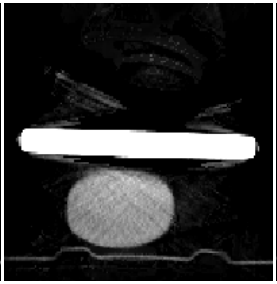


Power = 0

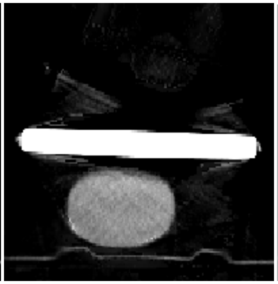


Power = 0.5

Power = 1

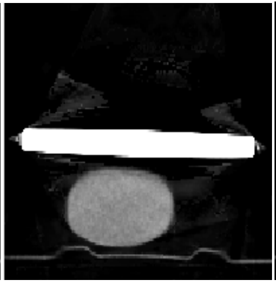


Power = 1.5

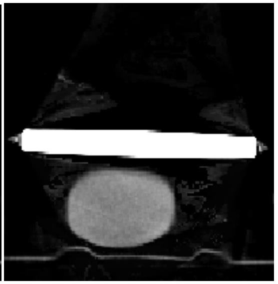


Power = 2

Power = 2.5

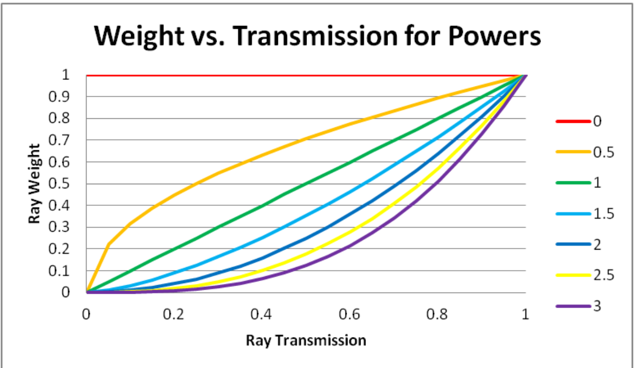


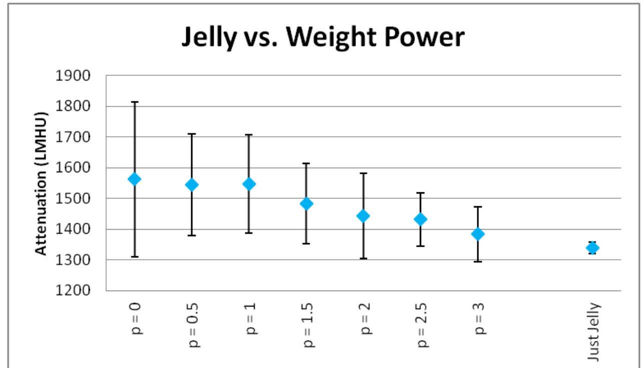
Power = 3

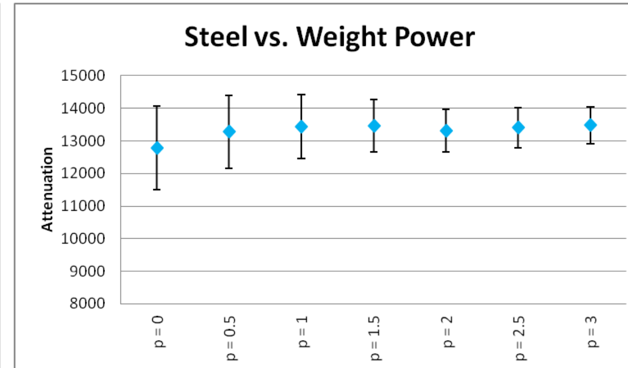


All of the images on the bottom row have the same imaging window

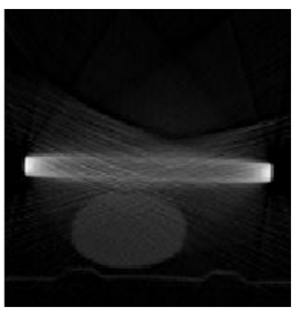
Power Law Ray Weighting Examining Iron

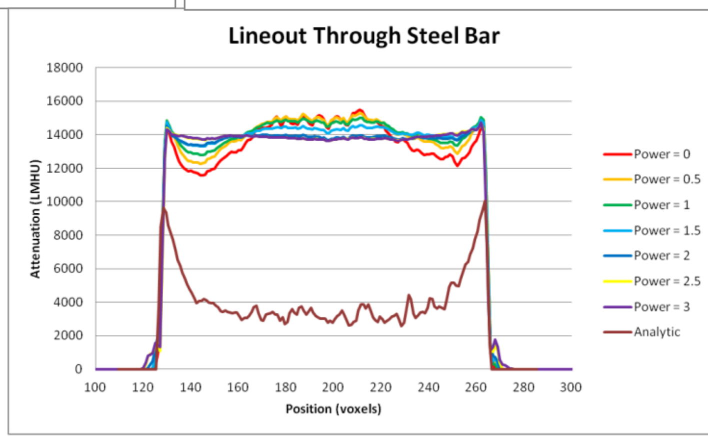


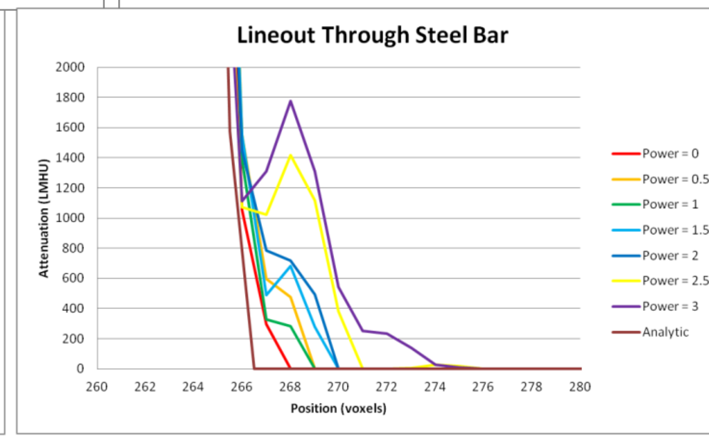




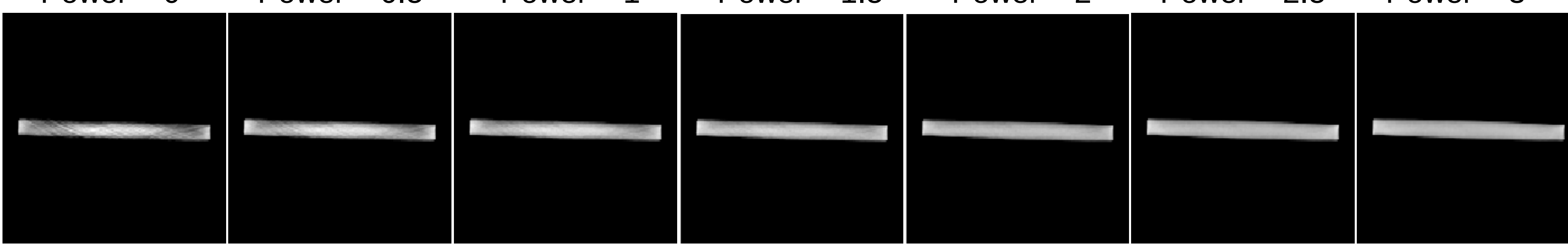
Analytic Reconstruction





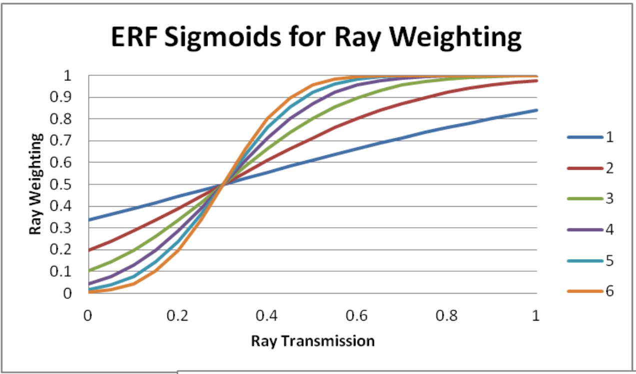


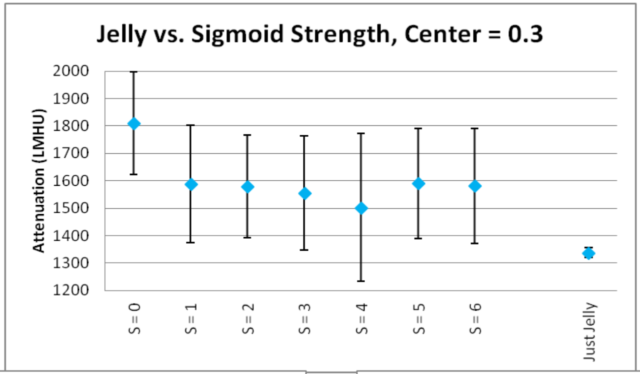
Power = 0 | Power = 0.5 | Power = 1 | Power = 1.5 | Power = 2 | Power = 2.5 | Power = 3

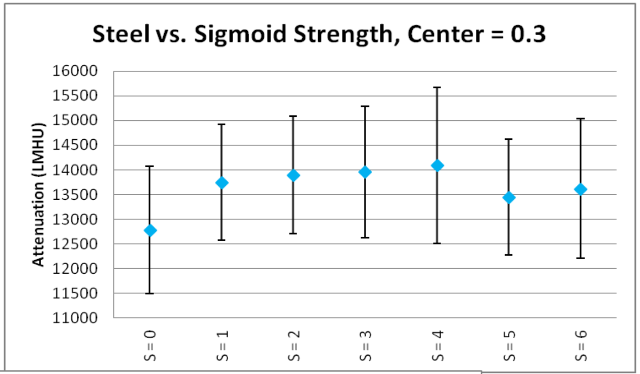


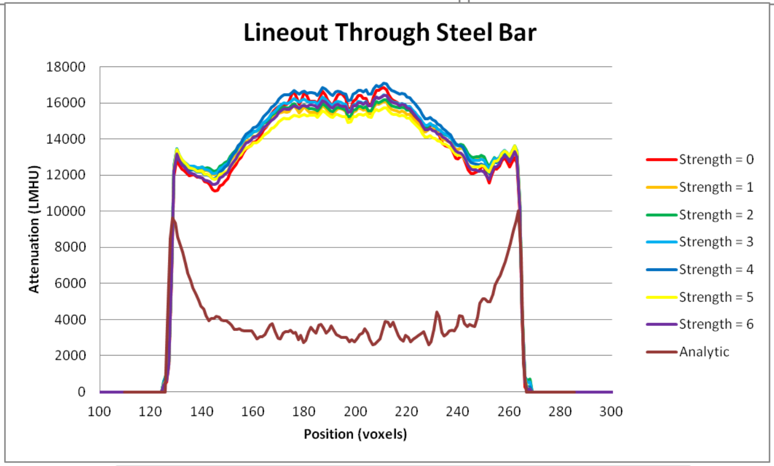
All of the images on the bottom row have the same imaging window

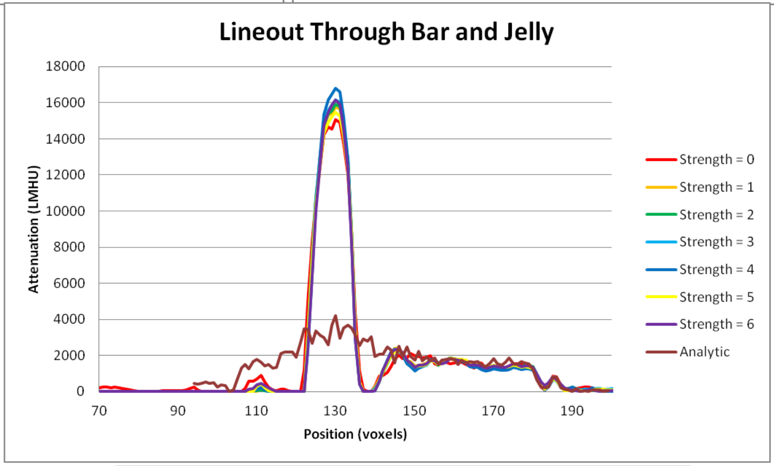
Sigmoid Ray Weighting Centered at 0.3

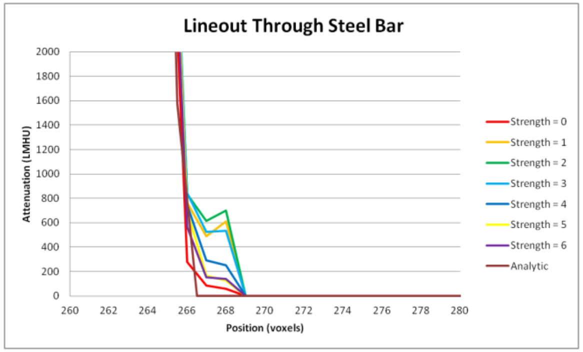


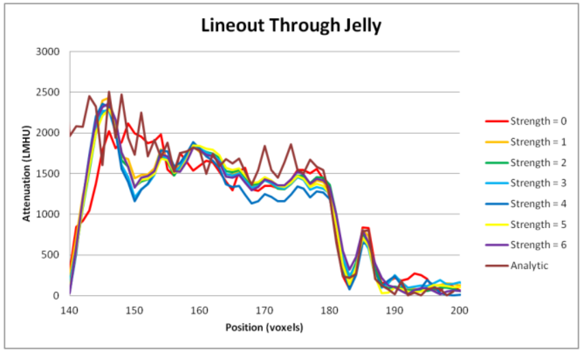




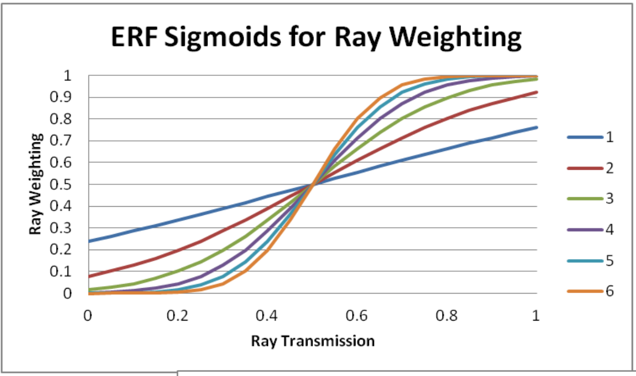


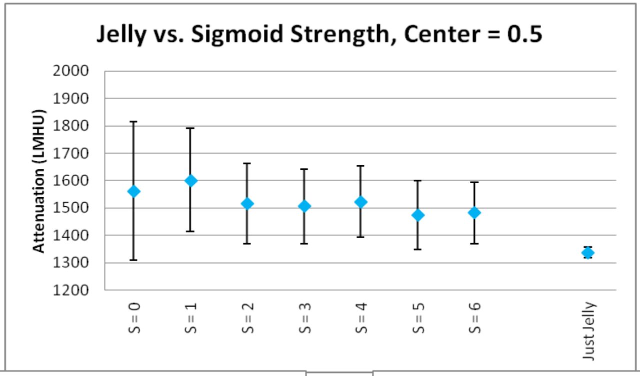


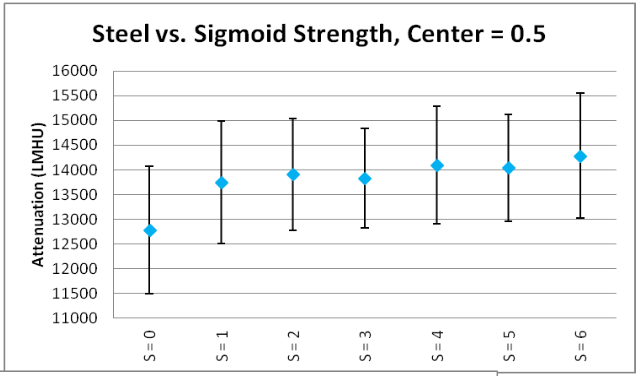


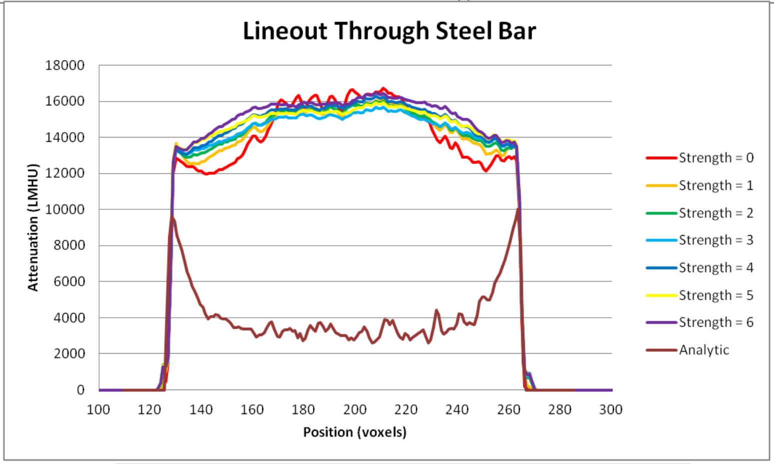


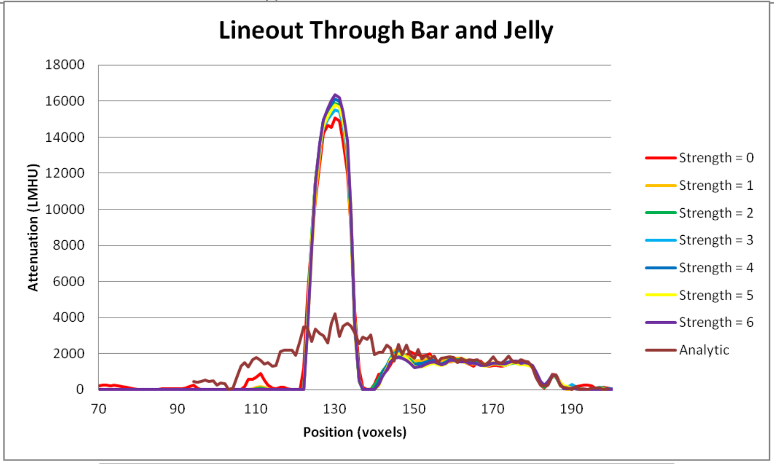
Sigmoid Ray Weighting Centered at 0.5

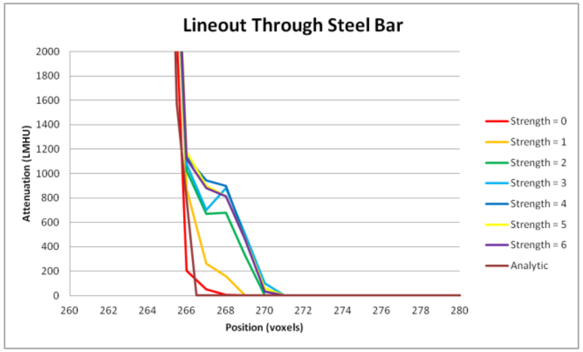


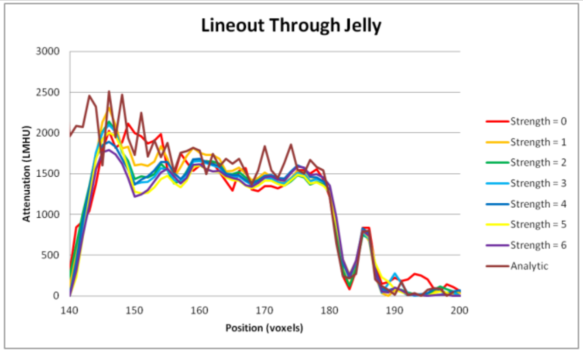












Ray Weighting Does Not Always Improve Results

Analytic | Power = 0 | Power = 1 | Power = 2 | Power = 3

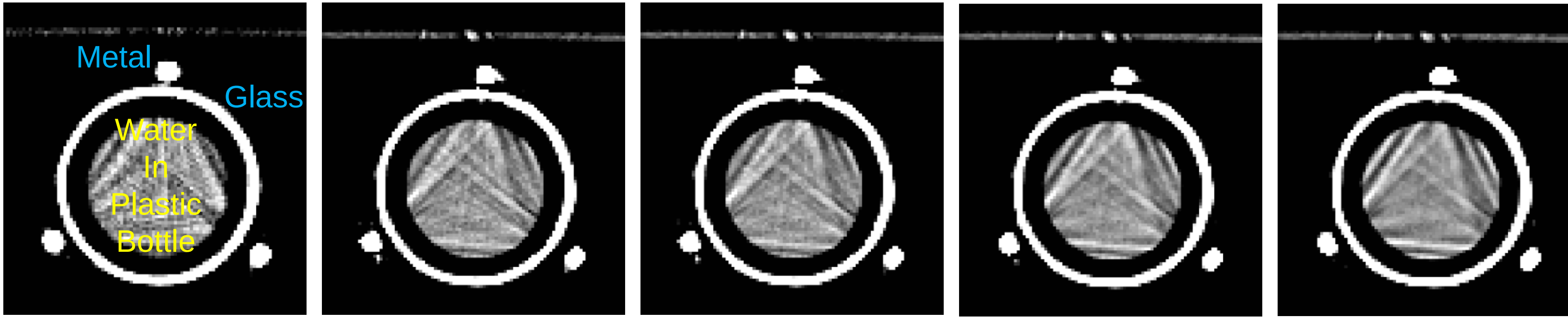
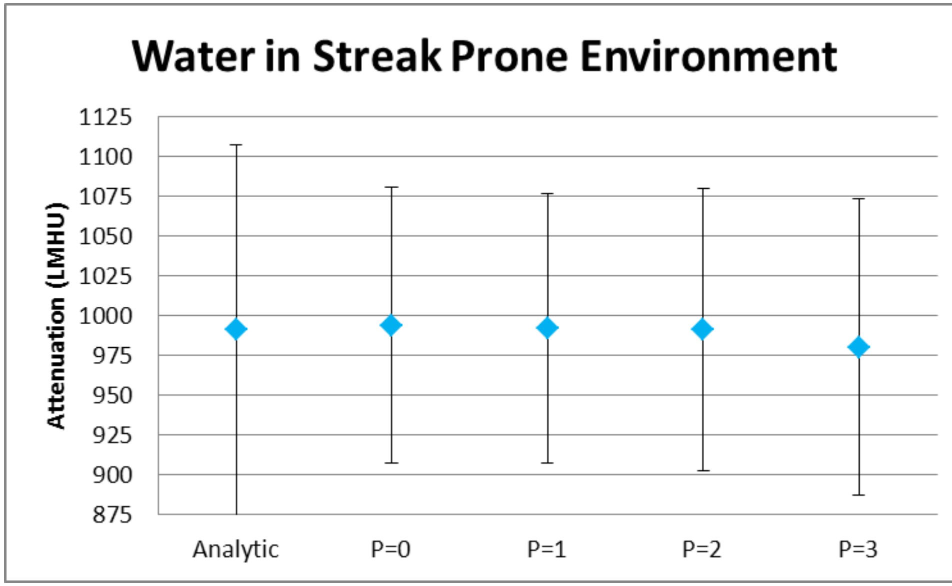


Image window is 621-1258

Notes:

- Iterative reconstruction without ray weighting moves mean attenuation of water sample 0.3% closer to 1000 than analytic reconstruction and decreases Standard Deviation of the voxels by 25%.
- Ray weighting does not improve these results.



Summary/Future work

- Summary
  - We have implemented a fully constrained conjugate gradient algorithm (CCG)* using the adjoint method for computing the error gradient and incorporated the capability to use ray weighting.
  - We are investigating ray weighting by
    - powers of ray transmission
      - There is theoretical justification for setting the power to 1**
      - We have found that values greater than 1 sometimes yield more uniform reconstructions for homogeneous materials.
      - We are still determining appropriate limits for the power and appropriate circumstances for using ray weighting
    - sigmoidal functions of ray transmission
      - Does not work as well as powers of ray transmission
- Future work
  - Demonstrate reduction in feature space size
  - Assess robustness across different types of clutter, threats, etc.

---

* D. M. Goodman, E. M. Johansson and T. W. Lawrence, "On applying the conjugate-gradient algorithm to image processing problems," *Multivariate Analysis: Future Directions*, Elsevier Science Publishers, 1993.

** See pages 536-7 and Appendix A of K. Sauer and C. Bouman, "A Local Update Strategy for Iterative Reconstruction from Projections," IEEE Trans. Sig. Proc., Vol. 41, No. 2, pp. 534-548, Feb. 1993.

Outline

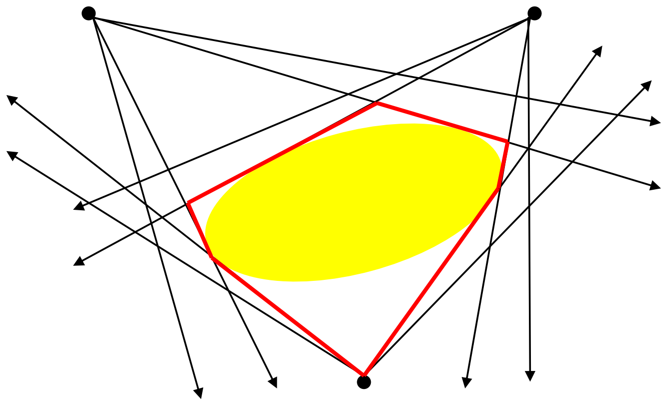
- High Level Overview of Conjugate Gradient Reconstruction
- Model Pruning: a Minor Tweak for Speed
- Error Approximation: a Major Tweak for Speed
- Approximating the Error
- Caveats
- Example: Clock Phantom
- Example: Clock-like Data

High Level Overview of CG Reconstruction

- Generate a model to fill space (blobs, regular voxels, pieces).
- Incorporate *a priori* information (or set everything to zero).
- Determine interactions between model and all rays.
- Perform an iteration of the conjugate gradient method:
  - Execute forward model for each ray
  - Determine mismatch between forward model and data for each ray.
  - Distribute error gradient to parts of model that interact with ray.
  - Generate appropriate direction given error gradient, regularization, and prior descent direction.
  - Perform line minimization to find minimum error in that direction.
  - If error is small enough, exit, otherwise repeat conjugate gradient iteration.
- Generate the output image.

Model Pruning: a Minor Tweak for Speed

- In many cases there are many voxels in the model that are intersected with un-attenuated rays. These regions:
  - Waste computational resources
  - Are a potential source of error
- These voxels can be identified by examining the ratio of unattenuated rays / attenuated rays for each voxel.
  - If the ratio is larger than a user defined threshold (we use 0.25) the voxel attenuation is set to zero and frozen there.
- For compact objects this acts like a convex hull. For objects broken into pieces the regions between pieces may be eliminated.
- A threshold separating attenuating vs. non-attenuating rays must be set.



Error Approximation: a Major Tweak for Speed

- One iteration of the Conjugate Gradient Algorithm in a nutshell:

$$\vec{x}_{k+1} = \vec{x}_k + \alpha_k \vec{d}_k, \quad \vec{x}_0 = 0$$

$$\vec{d}_{k+1} = -\vec{g}_{k+1} + \beta_k \vec{d}_k, \quad \vec{d}_0 = -\vec{g}_0$$

$$\vec{g}_k = \nabla_f E(\vec{x}_k), \quad \beta_k = \frac{(\vec{g}_{k+1} - \vec{g}_k)^T \vec{g}_{k+1}}{\vec{g}_k^T \vec{g}_k}$$

- Major effort is the determination of α_k (the line search). It can require many evaluations of the error. Computing error for a system with 1E6 rays (1000 projections of 1000 data points each) on a 1E6 voxel grid can require > 2E9 operations.
- Approximating the error can reduce the effort to less than that required for two full evaluations of the error (a speedup of approximately 20 times, although the result is non-deterministic). Our approach to approximation is to choose a random subset of the rays to approximate the error.
  - We use a different subset for each iteration. We have been using a subset with sqrt(original vector length) rays.
  - This technique is related to conjugate gradient methods with inexact searches *.

* D. F. Shanno, “On the convergence of a new conjugate gradient algorithm,” SIAM J. Numer. Anal., Vol. 15, No. 6, pp. 1247-57, December 1978.

Approximating the Error

- The error we are using for CT reconstruction is the squared projection difference error given by

$$E[\mu(\vec{r})] = \frac{1}{2}\sum_{m=1}^{M} w_m \left[I_m(s_{final}) - I_{m,observed}(s_{final})\right]^2$$

  where E is the total error, μ is attenuation (essentially the x vector of the conjugate gradient), r is position, M is the number of rays, m is the ray index, w is the ray weight, I is the current modeled ray intensity, s_{final} is the detector position, and $I_{observed}$ is the data we are trying to match.

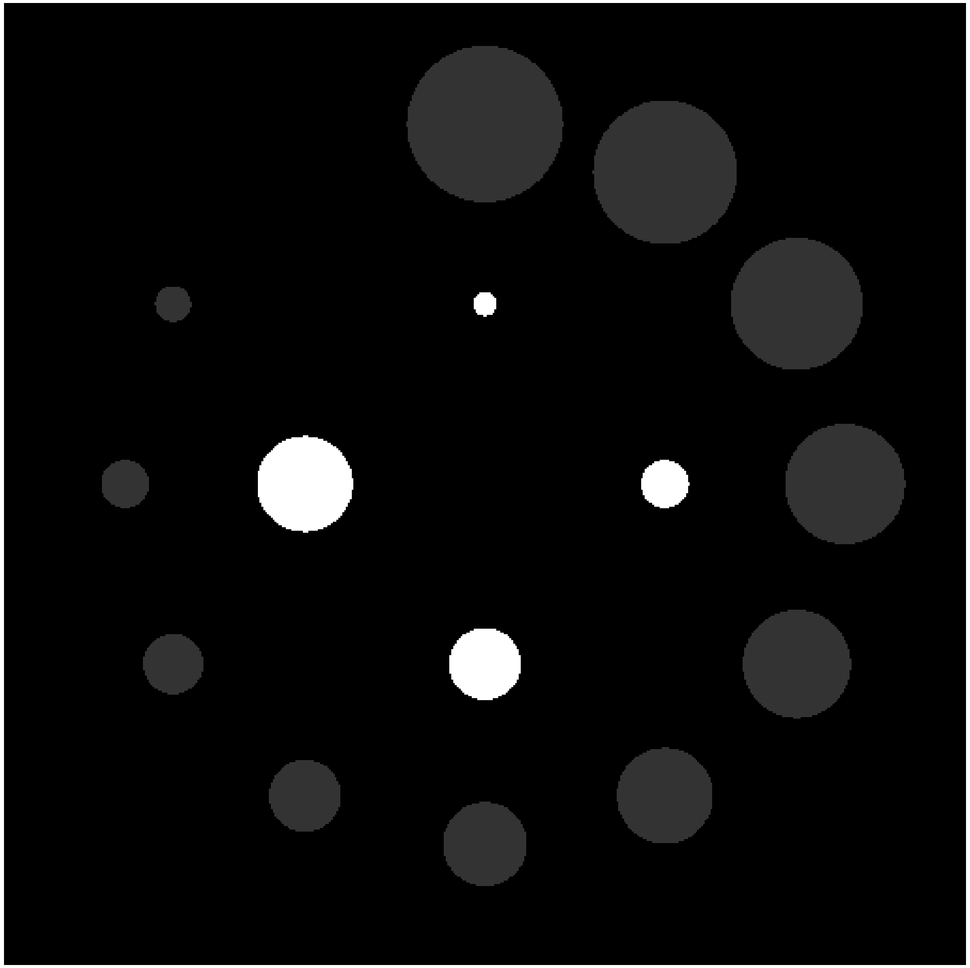
- The approximate error we use for CT reconstruction is the squared error of a random subset of the rays that have at least 0.1% mismatch between the modeled ray intensity and the detected ray intensity.
  - The value of the approximate error does not have to be close to the value of the true error. We need the minimum of the approximate error along the search direction to be near the minimum of the true error.
- One full error computation must be performed before the line search in order to generate the gradient for the entire problem.
- The majority of the implementation effort is in rewriting the error computation to deal with a set of selected rays and the selection of the rays themselves.

Caveats

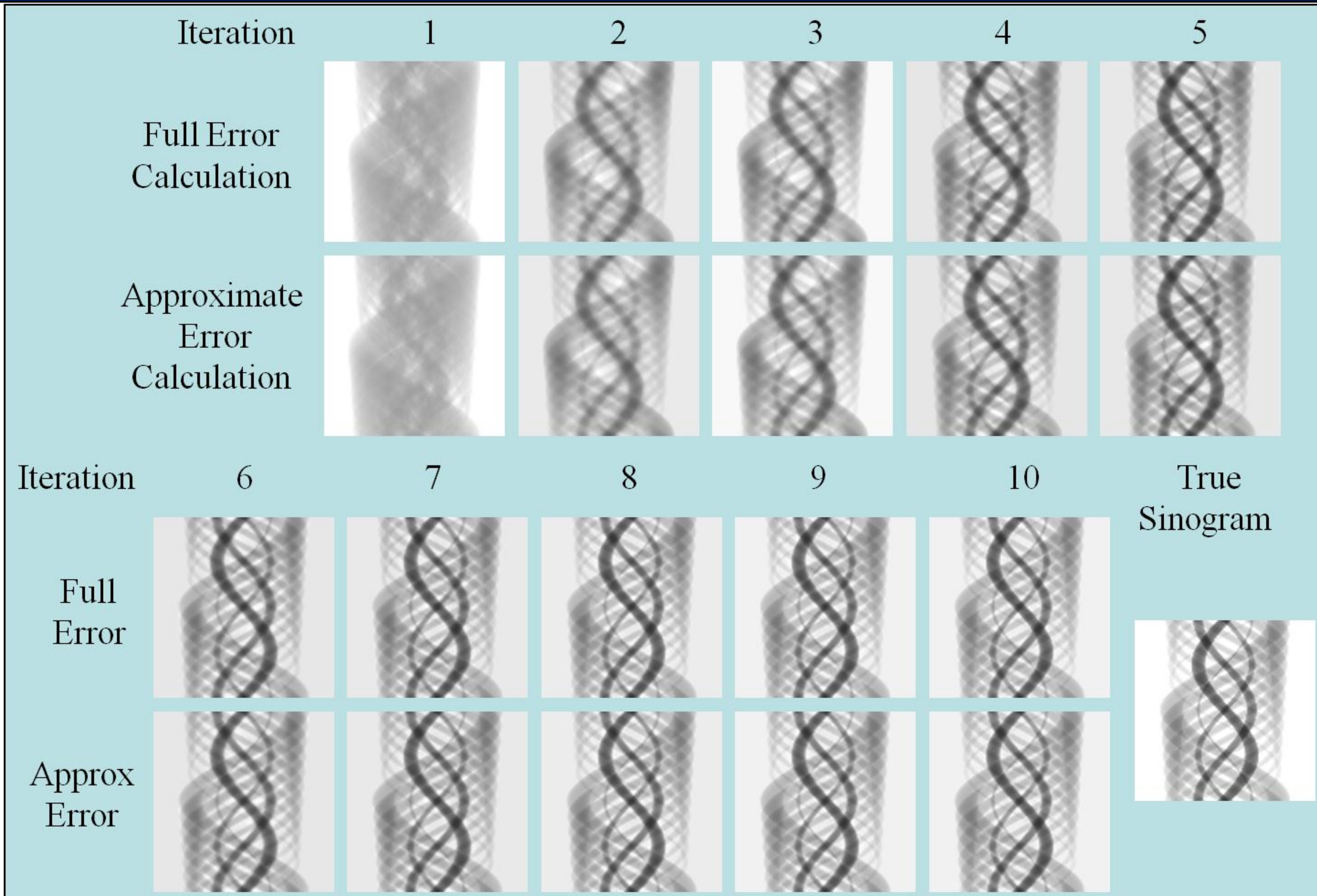
- Near the converged solution it becomes difficult to select an appropriate set of rays with which to approximate the error.
- At this point it is reasonable to switch to the full error conjugate gradient.
- Depending on when this switch occurs it may significantly reduce the time savings of this method.
- The same data will yield different results depending on which random sets of rays are used in the line search. This can be alleviated by switching to the full error conjugate gradient as the problem nears convergence.

Example: Clock Phantom

- Clock Type Phantom
- Dim circles have attenuation value 0.2
- Bright circles have attenuation value 1.0
- Region is 4 units on a side.

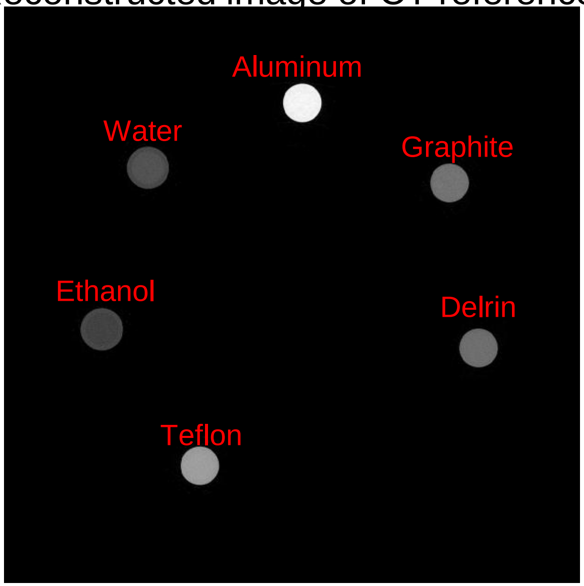


Approximate Error Line Search

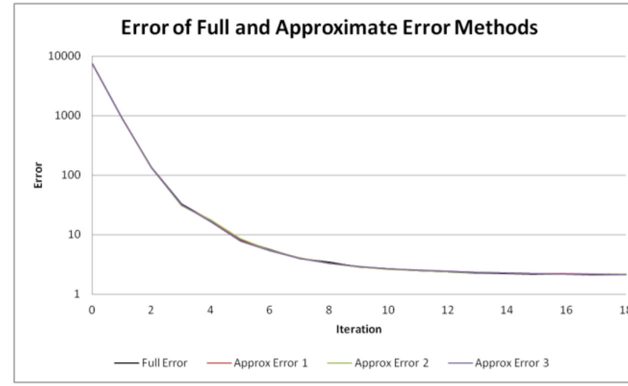


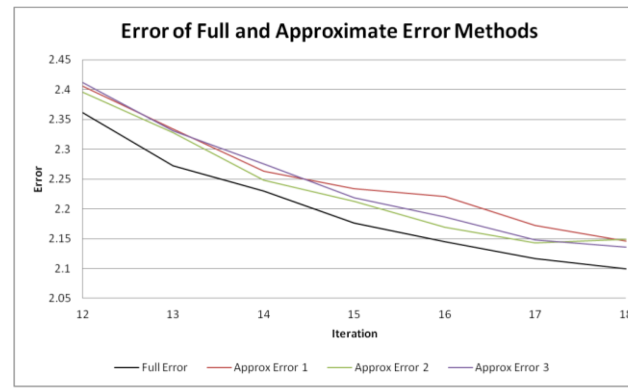
MicroCT References Example

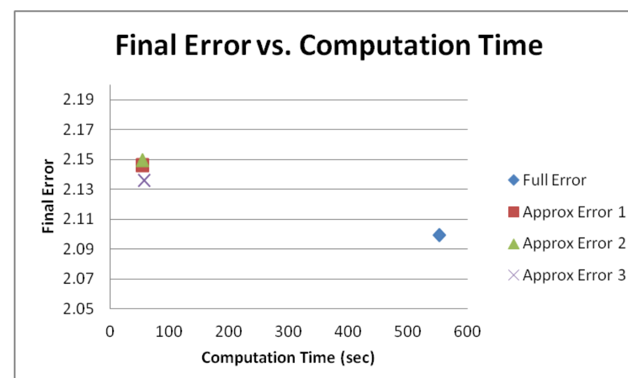
Reconstructed image of CT references



- Reconstruction of a CT slice with reference materials.
- 18 CCG iterations
  - Once with full error
  - Three times with approximate error
    - The plots show the full error (computed at the beginning of each approximate error line search and at the end of the reconstruction)
  - Approximate error results took 1/10th the time to generate.
    - Factor of 10 difference is actually smaller than usually observed because of the large amounts of empty space in the image eliminated by model pruning.

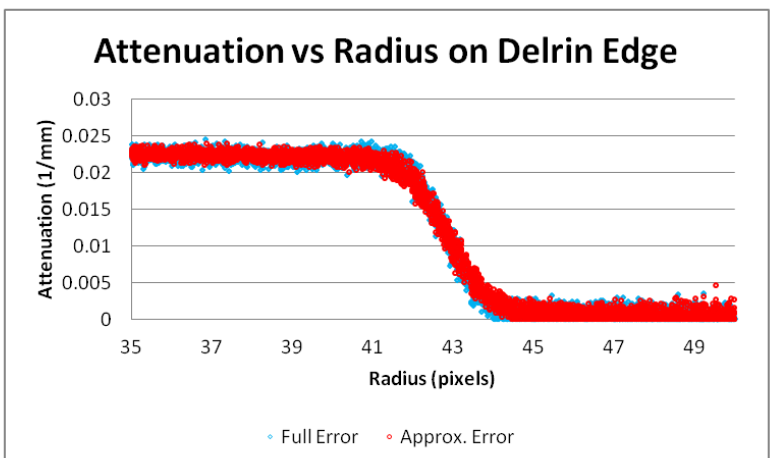






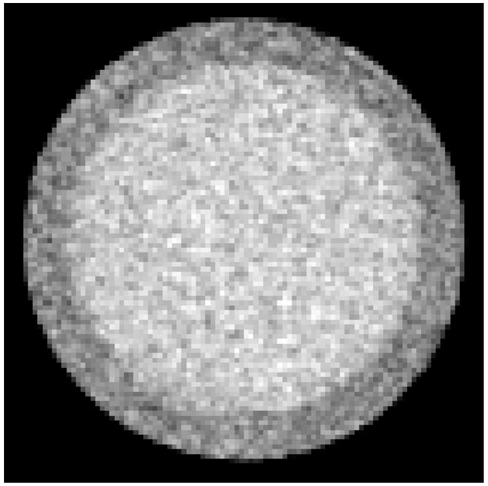
MicroCT References Example

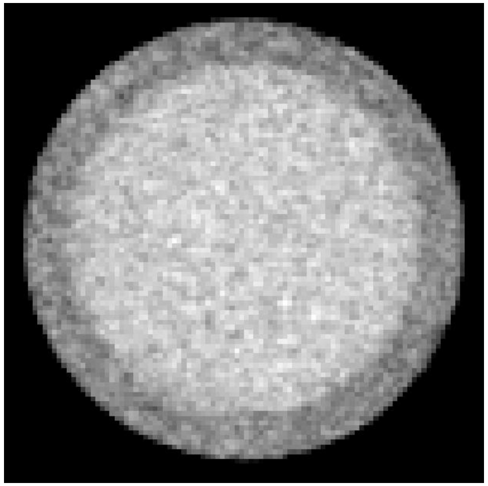
There are subtle differences between the reconstructions that are evident when viewed carefully. An example is the very slightly sharper edge on the Delrin sample seen in the graph.



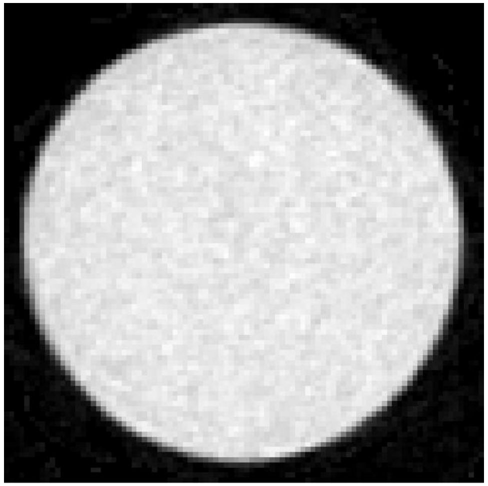
Full Error **Approximate Error**

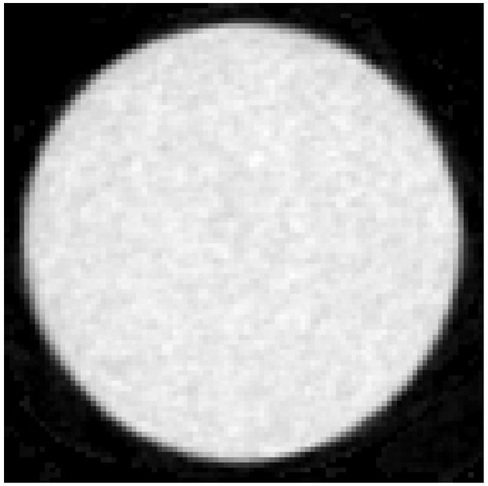
Water in plastic container





Delrin





Appendix B: Derivation of Error Gradient

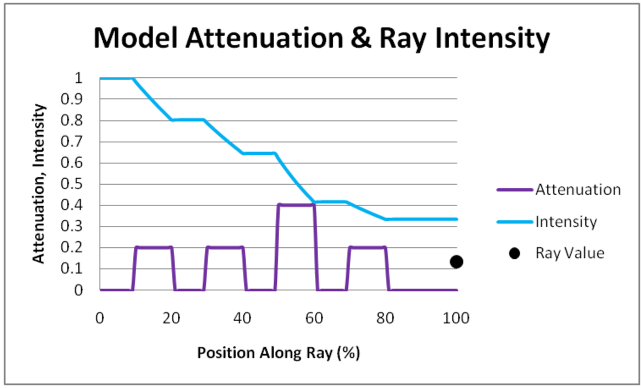
Outline

- One Dimensional Ray Equation
- Frechet Derivative and Variations
- Adjoint Ray Equation
- Manipulations to get Frechet Derivative
- Evaluation of Frechet Derivative
- Finite dimensional cases

One Dimensional Ray Equation

- Position along the ray is represented by *s*.
- Intensity at any point along the ray is represented by *I(s)*.
- Attenuation at any point along the ray is represented by $\mu(s)$.
- Initial Intensity $I(0) = I_0$.
- One dimensional ray equation is:

$$\frac{dI}{ds} + \mu(s)I(s) = -I_0\delta(s)$$



- Define the Error as:

$$E[\mu(s)] = \frac{1}{2}\left(I\left(s_{final}\right) - I_{observed}\left(s_{final}\right)\right)^2$$

- What we really want is the gradient of the error with respect to the attenuation distribution, the Frechet derivative.

Frechet Derivative and Variations

- The Frechet derivative, when integrated with the variation in the attenuation, gives the variation of the error (ΔE) :

$$\Delta E[\mu(s)] = \int \nabla_f E(s) \Delta\mu(s) ds$$

- The variation of the ray equation is given by:

$$\frac{d\Delta I}{ds} + \mu(s)\Delta I(s) + I(s)\Delta\mu(s) = 0$$

- And the variation of the error is given by:

$$\Delta E[\mu(s)] = \left(I(s_{final}) - I_{observed}(s_{final})\right)\Delta I(s_{final})$$

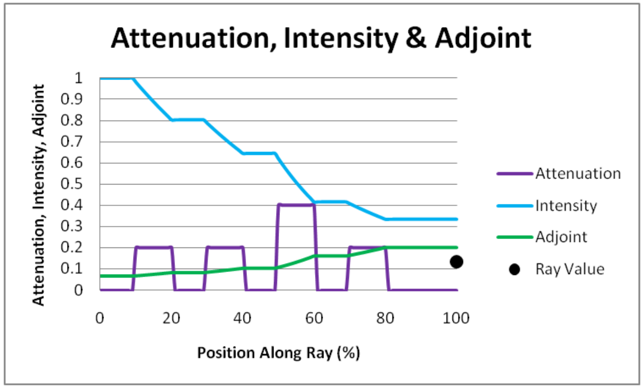
Adjoint Ray Equation

- The ray equation gives forward propagation. The adjoint ray equation gives backward propagation:

$$-\frac{d\tilde{I}}{ds}+\tilde{I}\mu=-\tilde{S}(s)$$

- The source term $\tilde{S}(s)$ is, in effect, an initial condition of

$$\tilde{S}(s)=\left(I\left(s_{final}\right)-I_{observed}\left(s_{final}\right)\right)\delta\left(s-s_{final}\right)$$



- So the variation of the error is given by

$$\Delta E[\mu(s)]=\int\tilde{S}\Delta I ds=-\int\left(-\frac{d\tilde{I}}{ds}+\tilde{I}\mu\right)\Delta I ds$$

Manipulations to get Frechet Derivative

- We use the identity:

$$\Delta I \frac{d\tilde{I}}{ds} = -\tilde{I}\frac{d\Delta I}{ds} + \frac{d}{ds}\left(\tilde{I}\Delta I\right)$$

- Realizing we can disregard the right hand term of the identity because it is zero at the endpoints we find the variation of the error:

$$\Delta E = -\int \left(\frac{d\Delta I}{ds} + \Delta I \mu \right) \tilde{I}\, ds$$

- Substituting from the variation of the ray equation we find the variation of the error is now in a form from which we can easily extract the Frechet derivative:

$$\Delta E[\mu(s)] = \int I(s)\tilde{I}(s)\Delta\mu(s)ds = \int \nabla_f E(s)\Delta\mu(s)ds$$

- The Frechet Derivative is thus:

$$\nabla_f E(s) = I(s)\tilde{I}(s)$$

How Can We Evaluate The Frechet Derivative?

- For simplicity consider a uniform attenuation distribution. Over the course of the forward propagation the intensity at any position is:

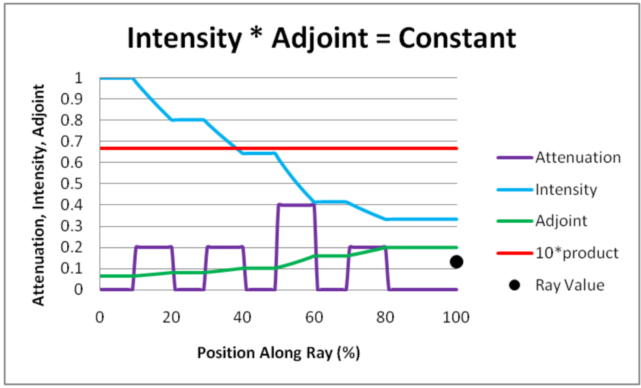
$$I(s) = I_0 e^{-\mu s}$$

- Suppose the result of the forward propagation is not the same as the observed intensity. The difference is the initial condition on the back propagation.
- Over the course of back propagation the intensity at any position is

$$\tilde{I}(s) = \left(I(s_{final}) - I_{observed}(s_{final})\right) e^{-\mu(s_{final} - s)}$$

- The resultant product at any position is constant:

$$I(s)\tilde{I}(s) = \left(I(s_{final}) - I_{observed}(s_{final})\right) I(s_{final})$$



- This works for **ANY** distribution of attenuation along the ray. And there is no need to actually do the back propagation.

Finite Dimensional Case

- If the distribution to be found is represented by the sum of basis functions $\phi_i(s)$ multiplied by parameters p_i:

$$\mu(s) = \sum_{i=1}^{N} p_i \phi_i(s)$$

- Then the finite dimensional Frechet derivative is given by

$$\frac{\partial E}{\partial p_i} = \int I(s)\tilde{I}(s)\phi_i(s)ds = I(s_{final})\tilde{I}(s_{final})P_i$$

- Considering the fact that product of the first two terms in the integral are constant, the finite dimensional Frechet derivative is the projection of the ray through the basis function, P_i, times a constant.