Introduction to Medical Imaging

Lecture 11: Cone-Beam CT Theory

Klaus Mueller

Computer Science Department
Stony Brook University

Available cone-beam reconstruction methods:
- exact
- approximate
- algebraic

Our discussion:
- exact (now)
- approximate (next)

The Radon transform and its inverse are important mechanisms to understand cone-beam CT

**Cone-Beam Transform**

\[ D\mu(\vec{a}(t), \vec{\beta}) = \left. \int_0^\infty \mu(\vec{a}(t) + s\vec{\beta}) ds, \ (\vec{a}, \vec{\beta}) \in \Gamma \times S^2 \right] \]

\( \vec{a}(t) \) is the source position along trajectory \( \Gamma \).
\( \vec{\beta} \) is the unit vector pointing along a particular x-ray beam.

The cone-beam transform reflects the data acquisition process of measuring line integrals of the attenuation coefficient \( \mu \).

**2D Radon Transform**

The analytical approach of reconstruction by projections has to be done in the context of the Radon transform \( \mathcal{R} \):

\[ \mathcal{R}\mu(\rho, \theta) = \int \int_\Gamma \delta(r \cdot \vec{\theta} - \rho) \cdot \mu(r) \, dl = \int dl \mu(\rho \cdot \vec{\theta} + l \cdot \vec{\theta}_l) \]

Thus in the 2D case the Radon transform \( \mathcal{R}\mu \) is identical to the measured cone beam transform \( D\mu \):

\[ D\mu(\vec{a}, \vec{\beta}_l) = \mathcal{R}\mu(\rho, \vec{\theta}) \]

with projection angle \( \theta \).

from: Dr. Günter Lauritsch, Siemens
3D Radon Transform

In three dimensions the Radon transform $\mathcal{R}$ is a plane integral

$$\mathcal{R}_\mu (\rho, \vartheta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mu(x, y) \delta(x \cdot \vartheta - \rho) \cdot dx \cdot dy$$

which is a severe complication compared to the 2D case. As we will see the link to the measured cone beam transform $D_{\mu}$ is not trivial.

from: Dr. Günter Lauritsch, Siemens

---

Fourier-Slice Theorem in 2D

The radial 1D Fourier transform $F_\rho \mathcal{R}_\mu (\rho, \vartheta)$ of the Radon transform $\mathcal{R}_\mu$ along $\vartheta$ is equal to the 2D Fourier transform $F_\omega \mu (\omega_x, \omega_y)$ of the object $\mu$ along $\vartheta$ perpendicular to the direction of the projection.

from: Dr. Günter Lauritsch, Siemens

---

Fourier-Slice Theorem in 3D

The radial 1D Fourier transform $F_\rho \mathcal{R}_\mu (\rho, \vartheta)$ of the Radon transform $\mathcal{R}_\mu$ along $\vartheta$ is equal to the 3D Fourier transform $F_\omega \mu (\omega_x, \omega_y, \omega_z)$ of the object $\mu$ along $\vartheta$ perpendicular to the direction of the projection.

from: Dr. Günter Lauritsch, Siemens

---

Exact Reconstruction in 2D and 3D

In 2D:
- use 2D inversion formula: the filtered backprojection procedure
- we have seen a spatial technique, only performing filtering in the frequency domain (in a polar grid)
- but may also interpolate the polar grid in the frequency domain and invert the resulting cartesian lattice
- employ linogram techniques for the latter (see later)

In 3D:
- use 3D inversion formula: not nearly as straightforward than 2D inversion
- full frequency-space methods also exist
- more details next (on all)
**Exact Inversion Formula**

The basic 3D inversion filtered backprojection formula, due to Natterer (1986):

\[
\hat{f}(x) = \frac{-1}{8\pi^2} \int_{S^2} \frac{\partial^2}{\partial \rho^2} \mathcal{R} f(\phi) \, d\theta.
\]

- \(\theta\) is the angle, a unit vector on a unit sphere
- \(x, \rho\) are object and Radon space coordinates, resp.: \(|\rho| = x \cdot \theta\)
- involves a 2\(^{nd}\) derivative of the 3D Radon transform
- the second derivative operator can be treated as a convolution kernel

Some manipulations can reduce the second derivative to a first derivative, along with convolution operators

\[
\hat{f}(x) = \frac{1}{2} \int_{S^2} \frac{-1}{4\pi^2} \frac{\partial^2}{\partial \rho^2} \mathcal{R} f(\phi) \, d\theta = \frac{1}{2} \int_{S^2} \frac{-1}{2\pi^2 \rho^2} \frac{\partial}{\partial \rho} \left[ \frac{1}{2\pi^2 \rho} \mathcal{R} f(\phi) \right] \, d\theta
\]

- many different variants have been proposed
  - for example: Kudo/Saito (1990), Smith (1985)

**Grangeat’s Algorithm**

Phase 1:
- from cone-beam data to derivatives of Radon data

Phase 2:
- from derivatives of Radon data to reconstructed 3D object

There are many ways to achieve Phase 2
- direct, \(O(N^3)\)
- a two-step procedure, \(O(N^4)\) [Marr et al, 1981]
- a Fourier method, \(O(N^3 \log N)\), [Axelsson/Danielsson, 1994]
- a divide-and-conquer strategy, \(O(N^3 \log N)\) [Basu/Bresler, 2002]
- we shall discuss the first three here

But first let us see how Radon data are generated from cone-beam data

**Transforming Cone-Beam to Radon Data**

**Strategy:**
- weigh detector data with a factor \(1/SA\)
- integrate along all intersections (lines) between the detector plane and the required Radon planes
  - there are \(N^2\) such lines (\(N\) lines and \(N\) rotations)
- take the derivative in the \(s\)-direction (in the detector plane perpendicular to \(t\))
- weight the 2D data set resulting from a single source position by the factor \(SC / \cos^2 \beta\)

The order of these operations can be switched since they are all linear (Grangeat swapped the order of operation 2 and 3)
Radon Data to Object: Direct Method

There are $O(N^3)$ data points in Radon (derivative) space. Each is due to a plane integral.

The direct method simply inserts the plane data into the object space, one by one:
- this is basically the expansion of a point into a plane, defined by $(\theta, \rho)$
- this gives rise to an $O(N^5)$ algorithm

Radon Data to Object: Two-Step Method

Each vertical plane holds all Radon points due to plane integrals of perpendicularly intersecting planes:
- filtered backprojection reduces the plane integrals to line integrals, confined to horizontal planes

The horizontal planes are then reconstructed with another filtered backprojection:
Each such operation is $O(N^3)$ and there are $O(N)$ of them, resulting in a complexity of $O(N^4)$

Radon Data to Object: Fourier Space Approach

from Axelsson/Danielsson
Radon Data to Object: Fourier Space Approach

Takes advantage of the $O(N \log N)$ complexity of the FFT at various steps

It also uses linograms [Edholm/Herman, 1987] to reduce 2D interpolation to 1D interpolation

The complexity is then $O(N^3 \log N)$

Long Object Problem

- Reconstruction of an ROI should be feasible from projection data restricted to the ROI and some surrounding.
- The basic 3D Radon inversion formula does not fulfill this request.

Tuy's Sufficiency Condition

To reconstruct a point $x$ of the object any plane containing $x$ must have at least one non-tangential intersection point with the source trajectory.

Concept of PI-Lines

For a point $x$ on a PI line any plane containing $x$ has at least one intersection point with the PI segment associated with the PI line.

The PI segment is that portion of the source trajectory needed for reconstructing the point $x$. 
Examples of Complete Trajectories

A prominent example of an incomplete trajectory

3D Radon Data Acquired by a Circular Trajectory

Challenges in Cone-Beam Reconstruction

The naive application of the 3D Radon inversion formula is prohibitive due to

- long object problem
- enormous computational expense

Simplifications have to be found to end up in an efficient and numerically stable reconstruction algorithm preferably in a shift-invariant 1D-filtered backprojection algorithm

Utilization of redundant data is obscure. Ideally redundancy in collected Radon planes has to be considered. However, this approach is suboptimal because:

- it is quite complicated
- underestimates the redundancy of data
- typically in cone beam, the data are highly redundant in approximation
The Feldkamp-Davis-Kress Algorithm

Approximate cone-beam algorithm
Works well for smaller cone-beam angles
Widely in use

The Feldkamp Algorithm: Details

for each projection
weight pixels by \( a/b \)
ramp-filter each column (along \( y_d \) direc.)
for each grid voxel \( v_j \)
project \( v_j \) onto image along cone-rays
interpolate voxel update \( dv_j \)
weight \( dv_j \) by depth factor \( c_j \): \( dv_j = dv_j \cdot c_j \)
add result to grid voxel: \( v_j = v_j + dv_j \)

\[
c_j = \frac{a^2}{(a + \sqrt{v_{jx}^2 + v_{jz}^2 \cos(\varphi - \varphi_k)})^2}
\]