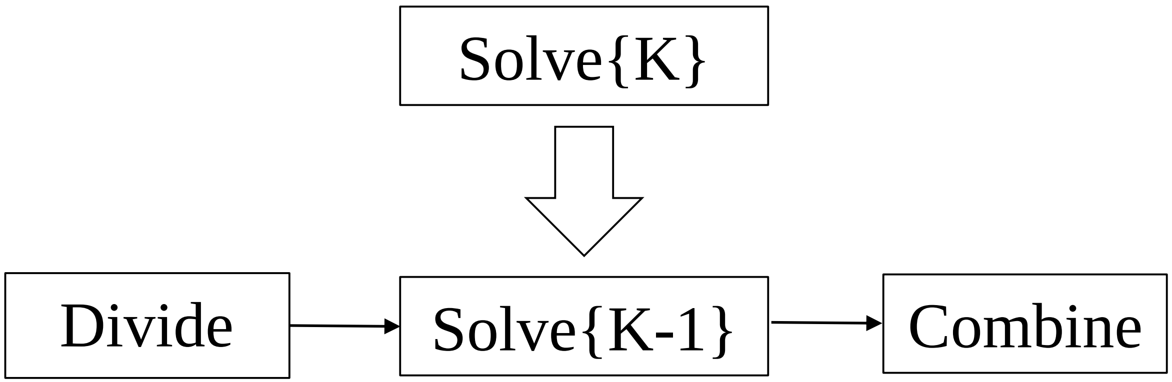
ECE 420

Lecture 8

Mar 11 2019

Divide and Conquer Algorithms

- Given a problem of a particular size, decompose into simpler subproblems
- Recursively apply the divide and conquer strategy until a sufficiently small problem is produced
  - Directly solve the subproblem
- Combine results from solved subproblems to reconstitute solution to original problem



Divide and Conquer Subcomponents

- Divide sub-algorithm
  - Given a problem of order N, create K subproblems of order N_{sub}
  - In many D&C algorithms, $K = 2,\ N_{sub} = N/2$
- Base-case solver
  - For sufficiently small order N, "directly" form the solution
  - N might be small enough that analytical or trivial solutions can be used
- Combine sub-algorithm
  - Given solutions to the K subproblems of order N_{sub}, determine the solution to the problem of order N

What Makes D&C Fast?

- Replace a higher order complexity algorithm, e.g. $O(N^2)$
- Efficient Divide and Combine sub algorithms
  - Much lower complexity than original algorithm
- Extremely simple or trivial base case solutions
- Recursive nature of algorithm involves $O(\log N)$ stages
- Total accounting for D&C algorithm generally gives an order reduction $N \to \log N$

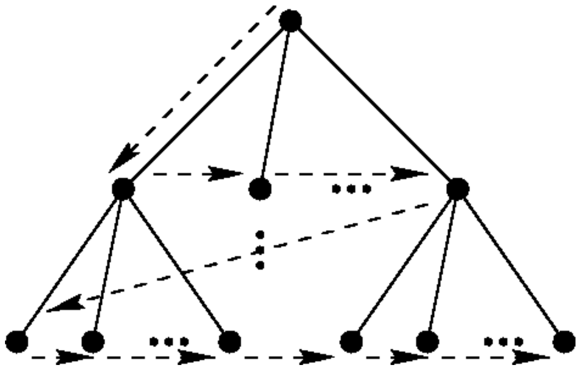
What Makes D&C Difficult?

- Not every problem admits this sort of efficient subproblem decomposition
- From an implementation perspective, the code becomes more complicated
  - Replace original problem solver with multiple subproblem routines
  - Sometimes the original problem solver code is used as the base case solver
- Additional design decisions regarding control flow, parallelism, memory footprint
- Sometimes a D&C algorithm can add significant enough challenges in implementation that the original algorithm outperforms in practice!

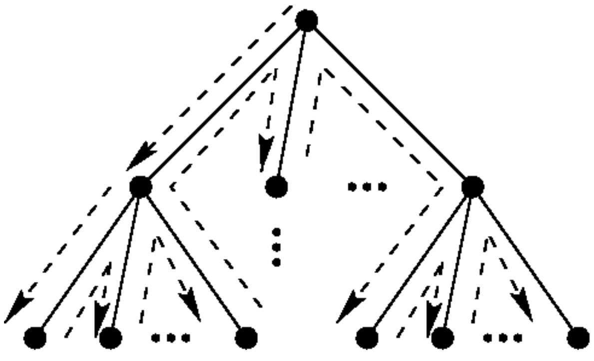
Control Flow for D&C Algorithms

- Recursive nature spawns progressively more subproblems to be processed
- What sequence should these subproblems be addressed?

Breadth First



Depth First



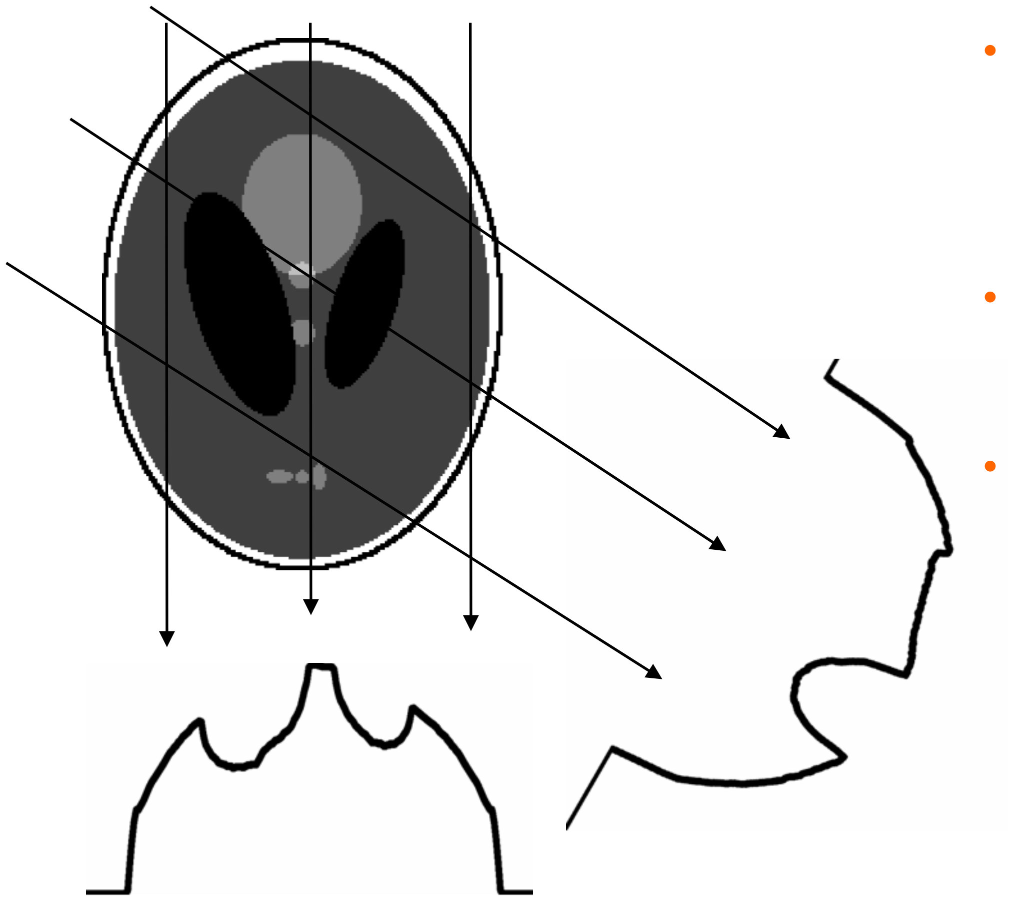
- How to implement an effective parallel version?
- Impact on amount of required memory as subproblems are created/destroyed?

D&C Algorithm Implementations

- A well implemented and tuned divide and conquer algorithm can dramatically outperform conventional algorithms
- Sometimes this part of the implementation requires as much or more effort than devising the algorithm's structure!
  - Significant gains can be had over "naïve" implementations
- Evolving computer architectures can necessitate revisiting design decisions and retuning periodically
- General-purpose divide-and-conquer algorithms can have far-reaching impact

Fast Algorithms for Tomography

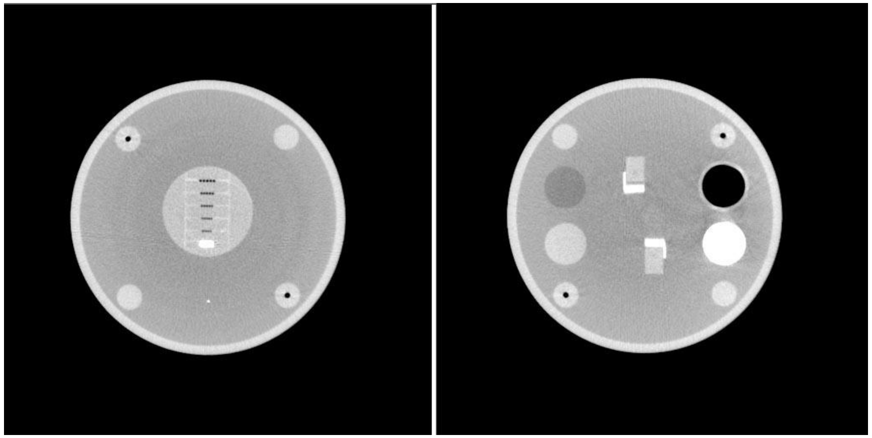
Tomography



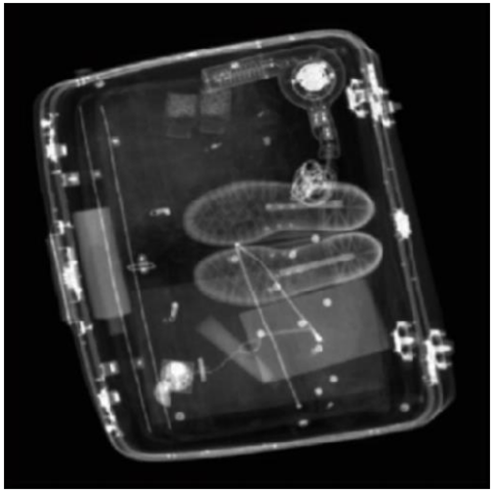
- Projections (line integrals) of an object acquired by a detector array
- Projections taken at a set of different angles
- Reconstruct internal representation of object from this data

Applications

Medical

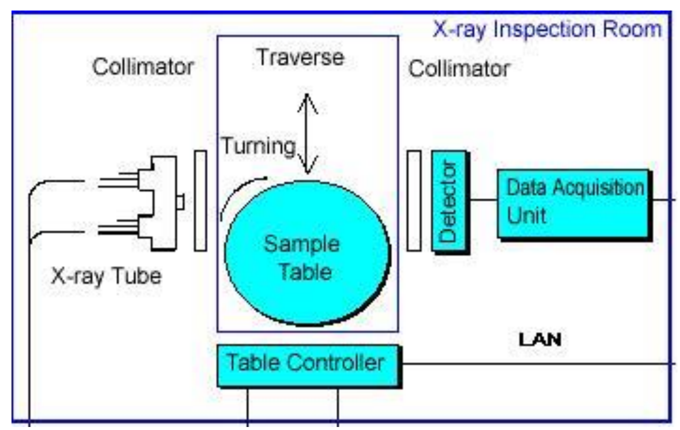


Industrial



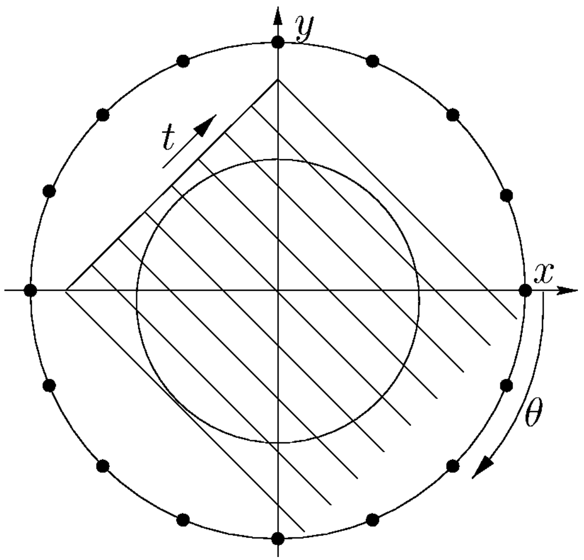
Security

CT Scanners


X-ray Inspection Room
Collimator
Traverse
Collimator
Turning
Detector
Data Acquisition Unit
Sample Table
X-ray Tube
LAN
Table Controller

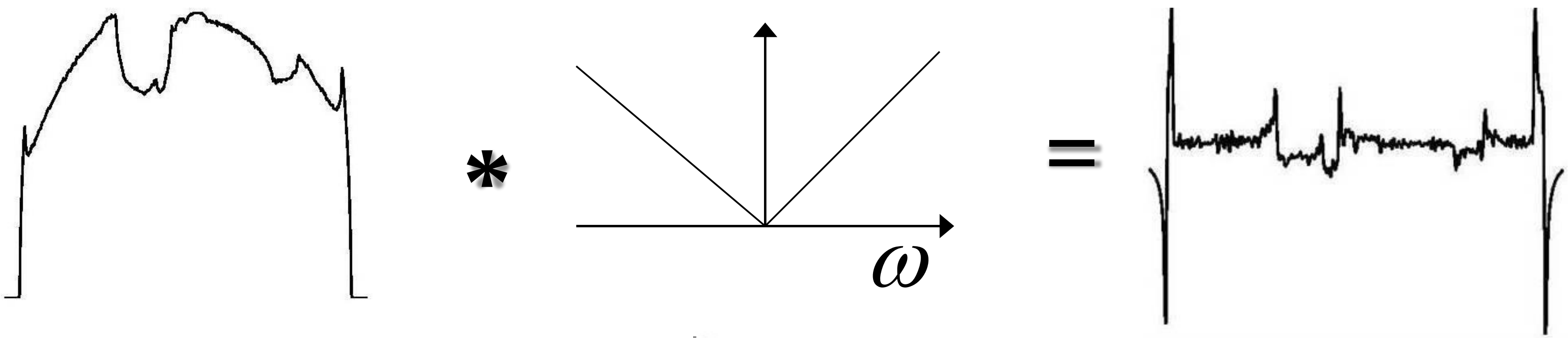
Parallel Beam Tomography

- Classical formulation is parallel beam
  - Projections taken on a set of parallel rays with particular direction of arrival
  - Projections generated by x-ray source that emits rays radially

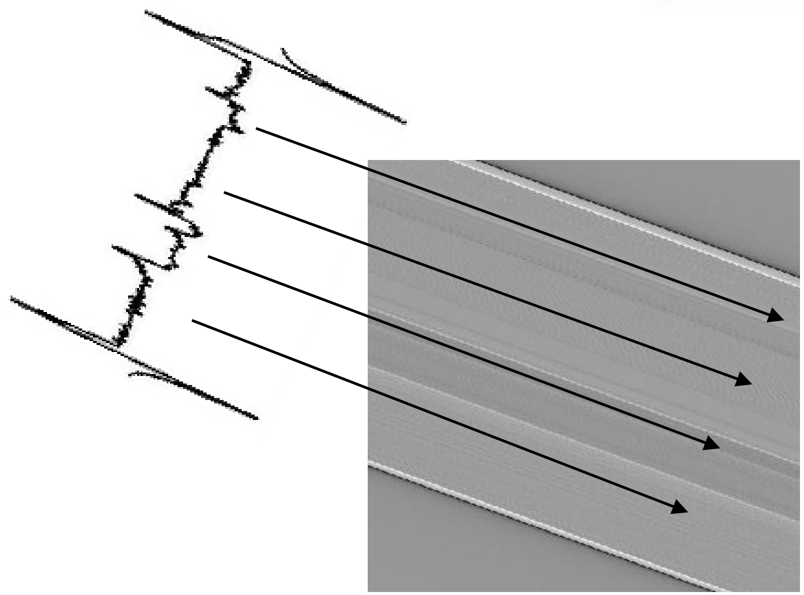


Filtered Backprojection

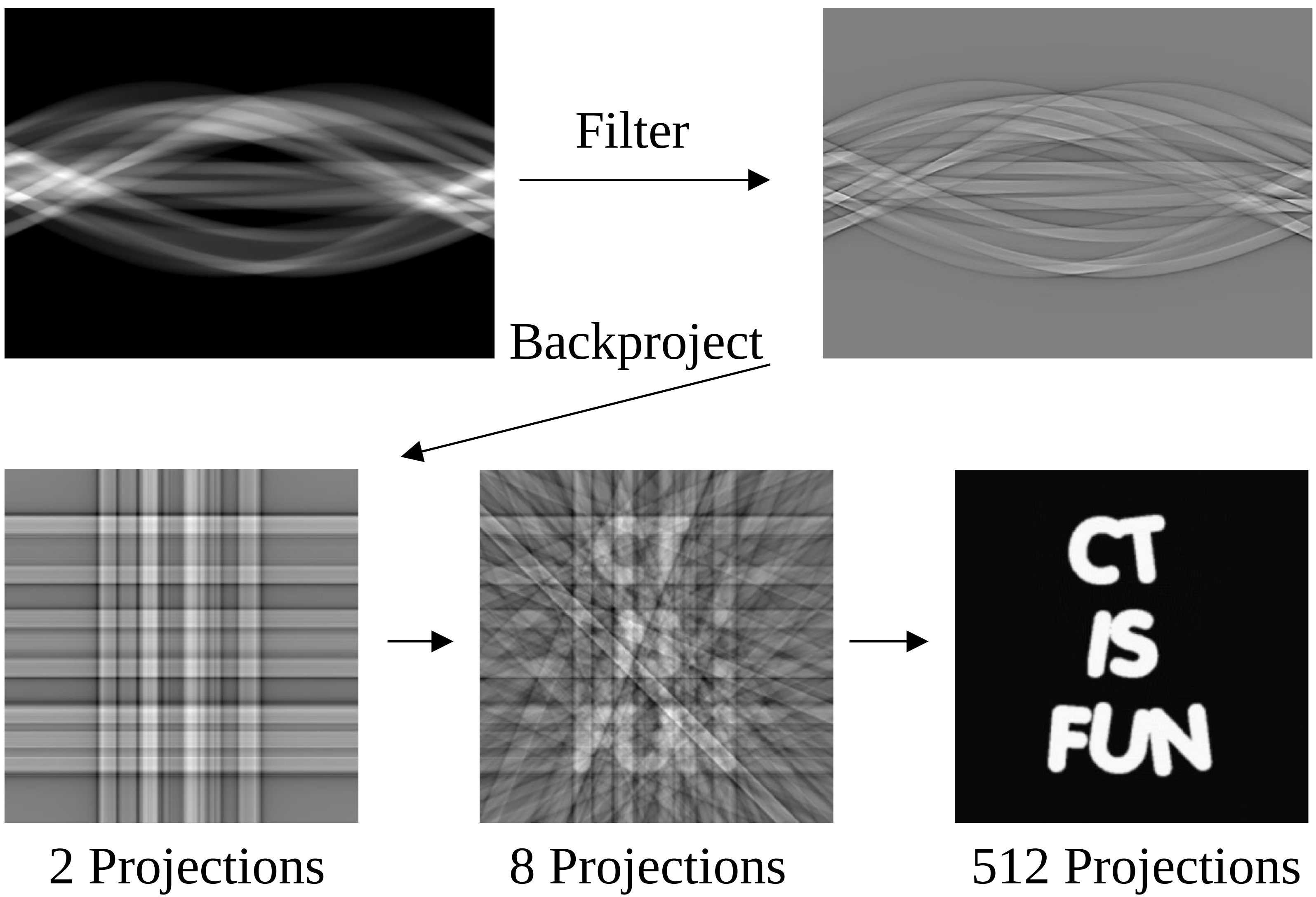
Filter Step



Backprojection Step



Filtered Backprojection



FBP Algorithms

- Benefits
  - Straightforward Implementation
  - Projections filtered and processed independently
  - Good reconstruction accuracy
- Complexity
  - Rule of thumb (mathematically motivated) number of projections needed proportional to size of image
  - Each pixel receives a contribution from each projection
  - Complexity of algorithm is $O(NPlog\ N) = O(N^2 logN)$ for the filtering step and $O(N^2 P) = O(N^3)$ for the backprojection step

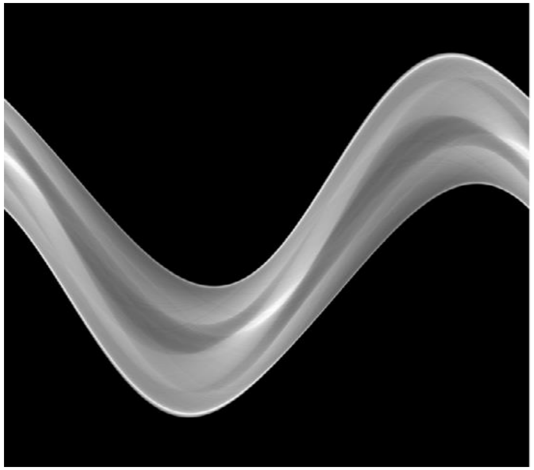
Fast Backprojection Algorithms

- Backprojection step dominates computational cost
- Fast algorithms reduce complexity of this step
- Results in 10-50x speedup
- Speed savings can allow
  - Less expensive (commodity) hardware
  - Even faster reconstructions (Real time imaging, iterative algorithms, or higher throughput)

Hierarchical Parallel Beam Backprojection

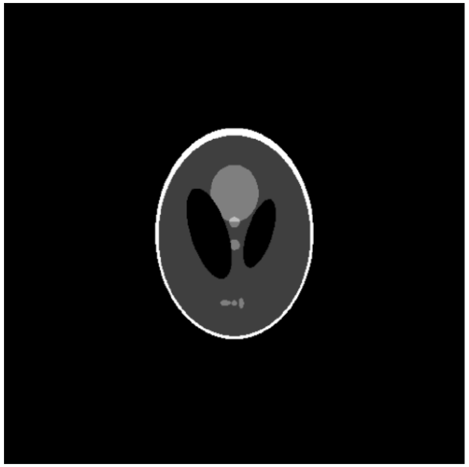
- Two properties used in formulation
- Shift property
  - A shift in the spatial domain corresponds to an appropriate shift of projection data
- Sampling Requirements
  - The number of projections required to reconstruct an object is proportional to the radius of support of the object

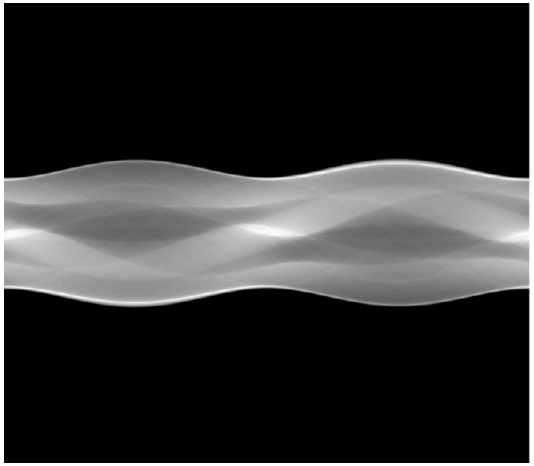
Shift Property



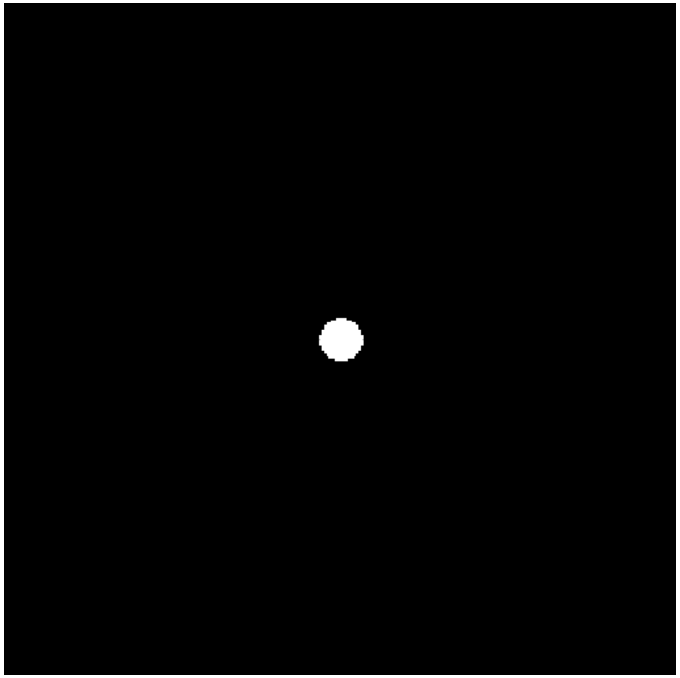
Image

Projection Data

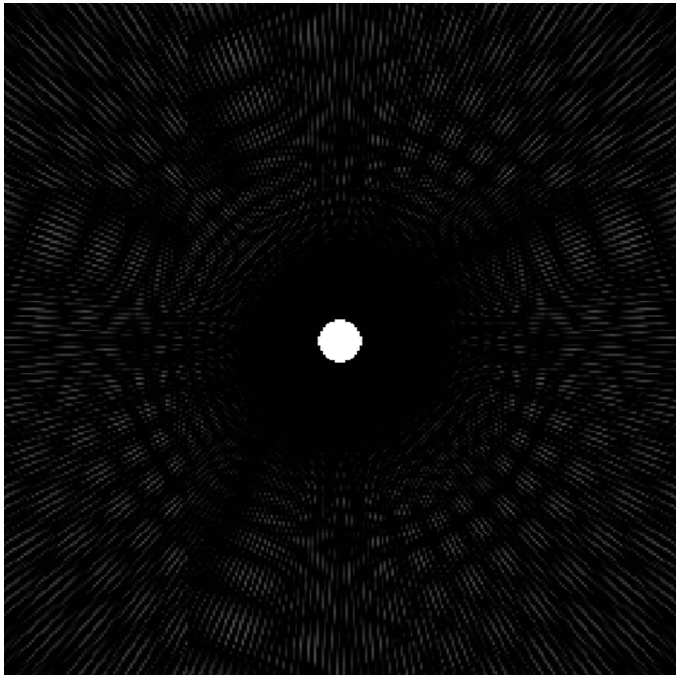




Projection Reduction Example

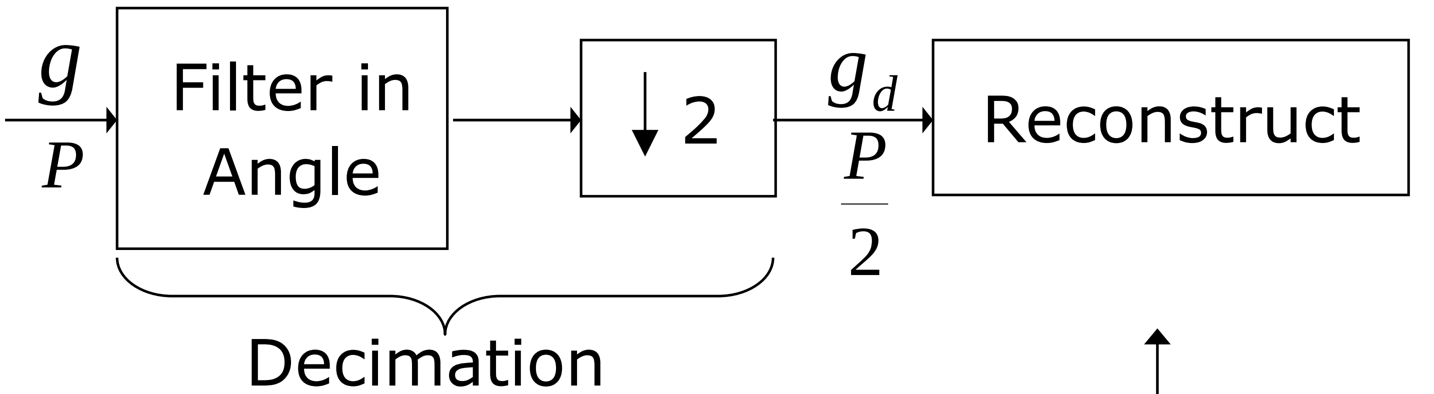


Large Projection Set

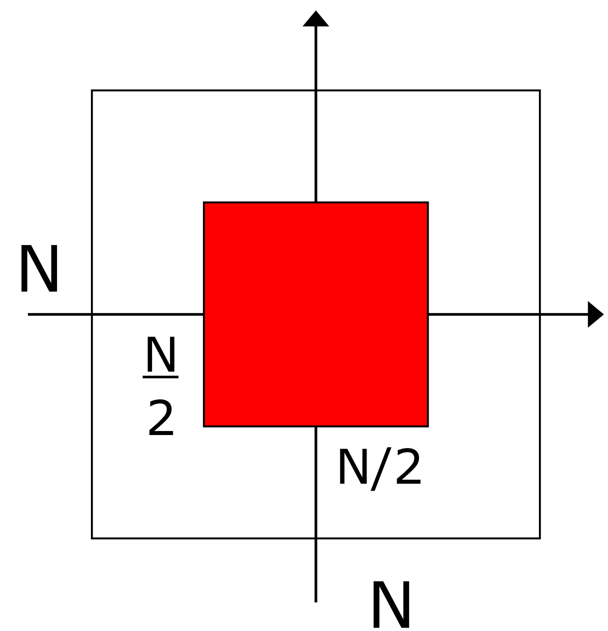


Reduced Projection Set

Reduction in Projections

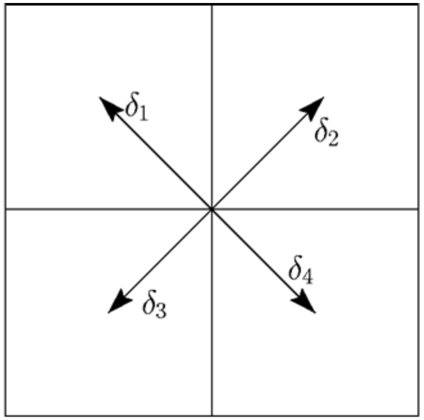


Can be used to reconstruct an object of 1/2 the size, centered at the origin, with 1/2 the number of projections



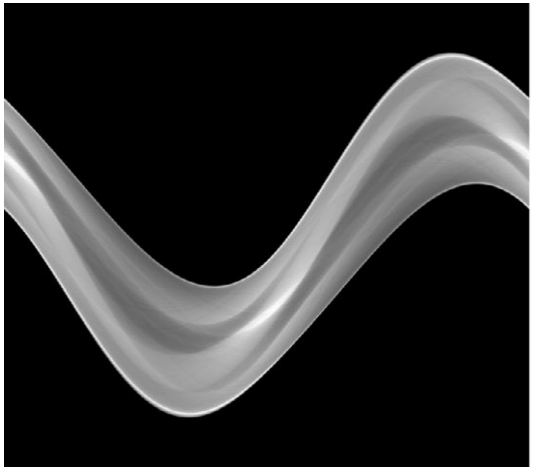
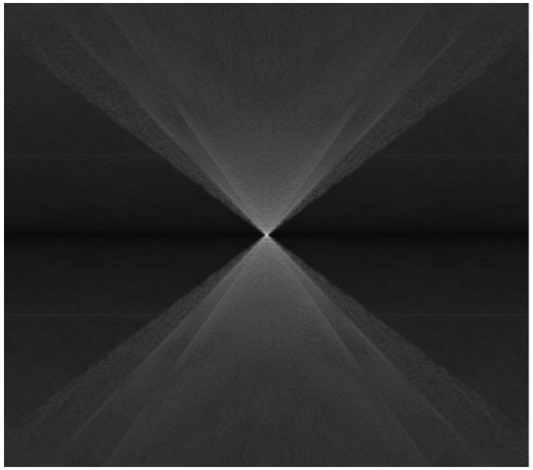
HPBB Method

- Decomposition into quadrants
  - Each is half the size, so each should require half the number of projections



- Apply shifting and sampling properties
  - Shifting projections according to $-\delta$ equivalent to moving subregion to origin
  - By sampling theorem, can now reduce the number of projections by a factor of 2
  - Backproject reduced projection set onto subregion and place into appropriate location in output image

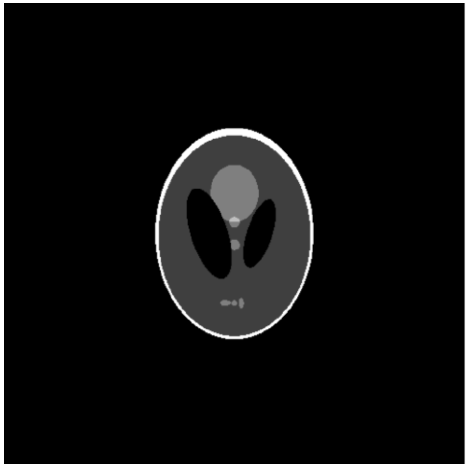
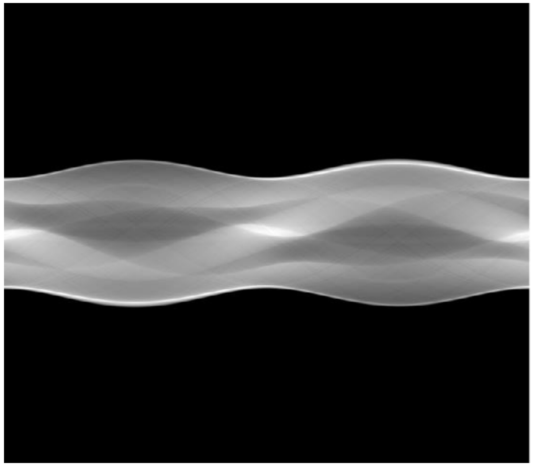
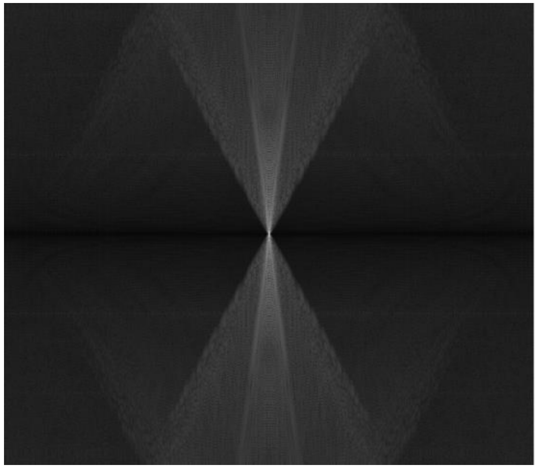
Angular Bandwidth Reduction

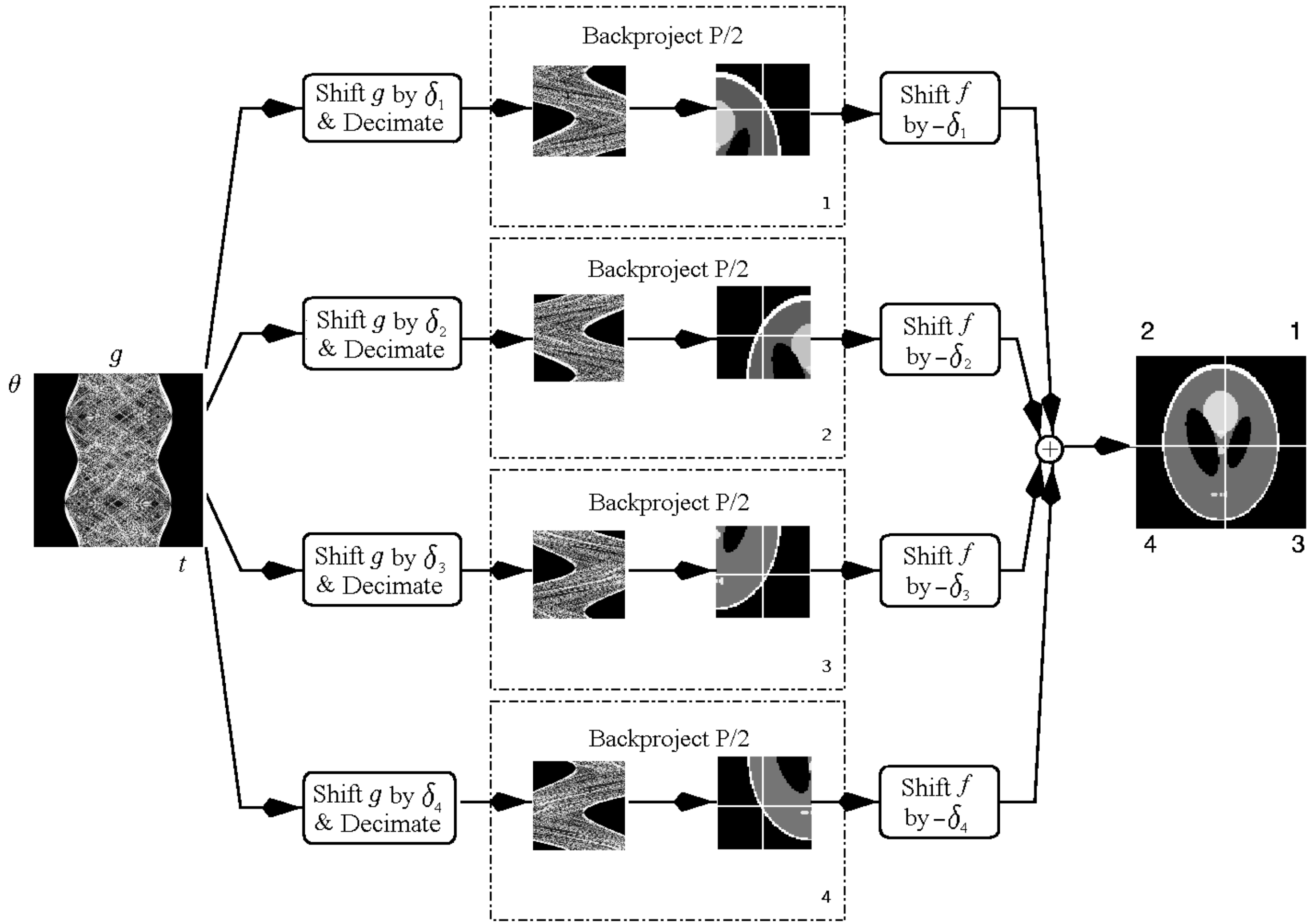
Image

Projection Data

FT of Proj Data

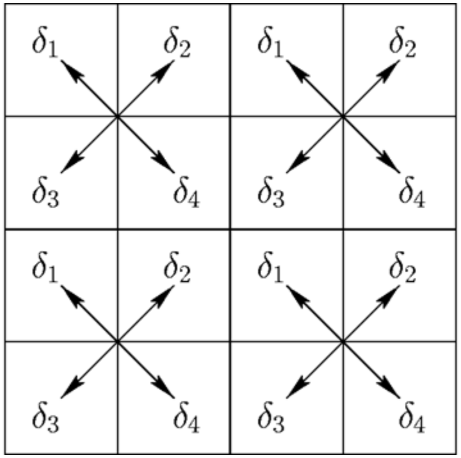




Decomposition Step

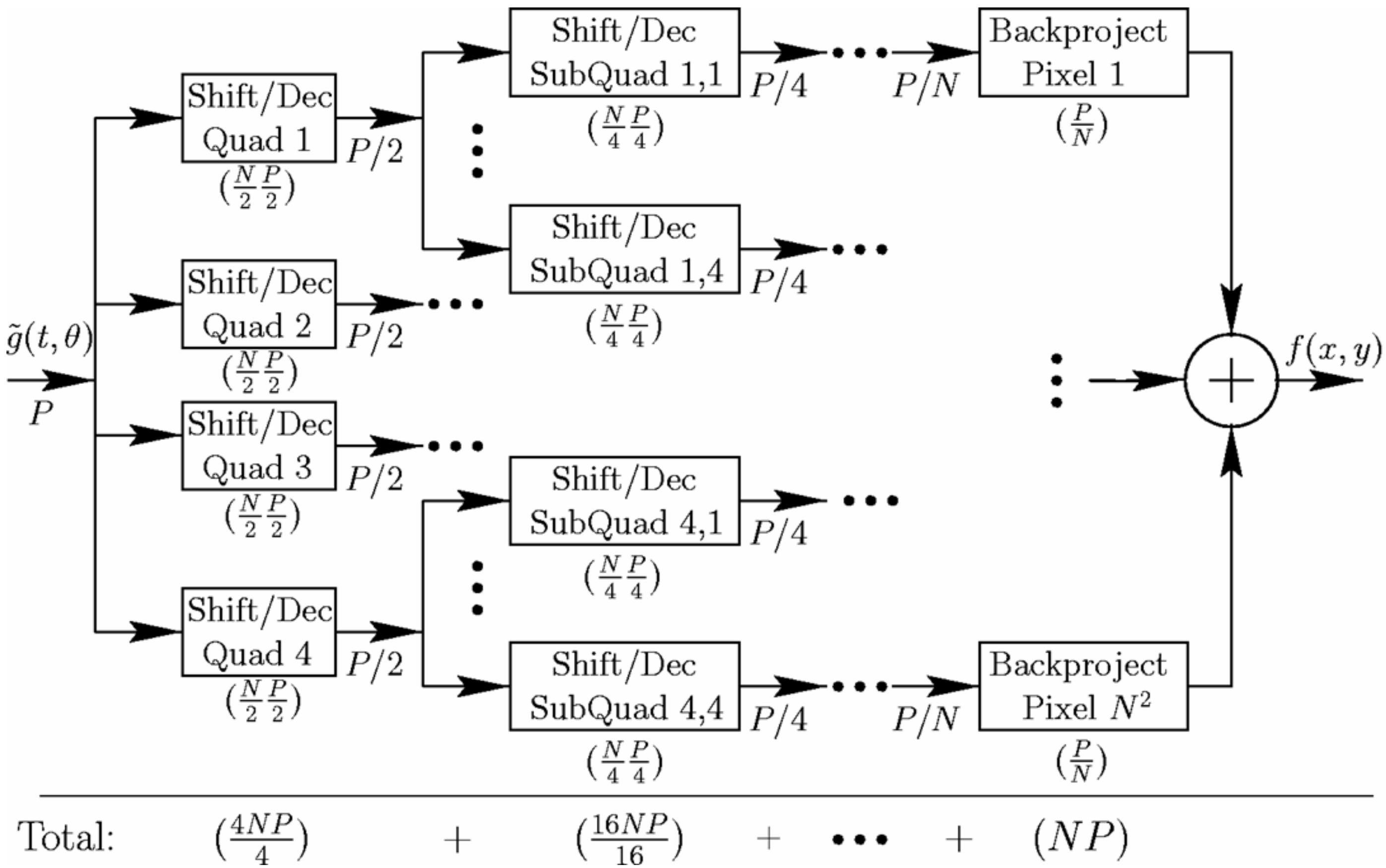


Recursive Decomposition

- Backprojection step is half the work
- Reduction in complexity through recursive decomposition
  - Further reduction in number of projections through shifting and decimation
  - Continue this decomposition until reaching size of one pixel



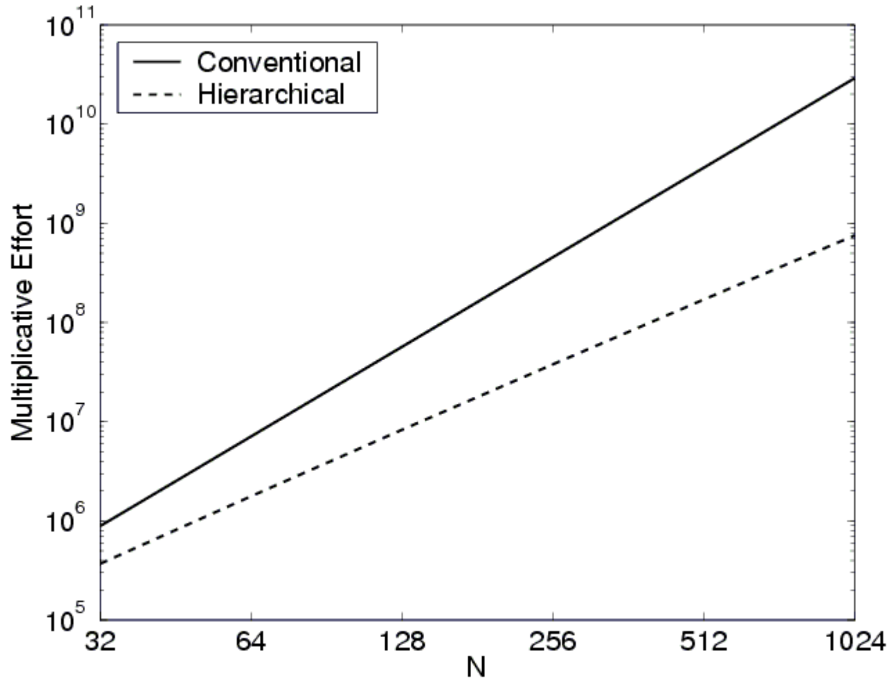
HPBB Complexity



HPBB Complexity

$O(NP)$ work per stage, $\log N$ stages

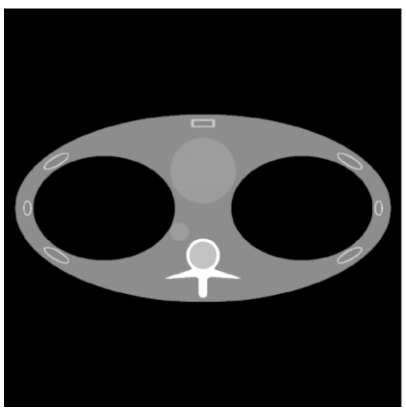
$O(NPlog\ N) = O(N^2 \log N)$ hierarchical vs. $O(N^3)$ for conventional



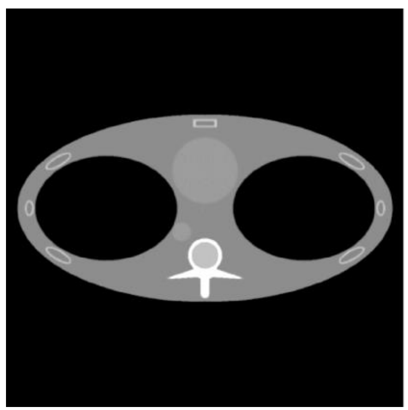
Algorithm Comparison

Conventional

Hierarchical



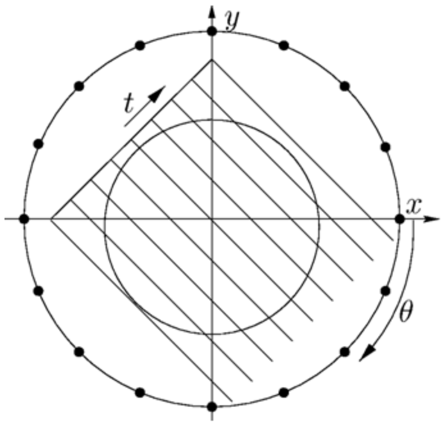
Conventional



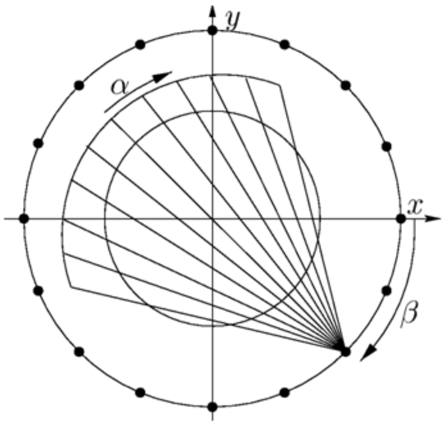
Hierarchical

	512^2 Image 1536 Projections
Conventional Algorithm	9.53 Seconds
Hierarchical Algorithm	0.17 Seconds
Speedup	56.1

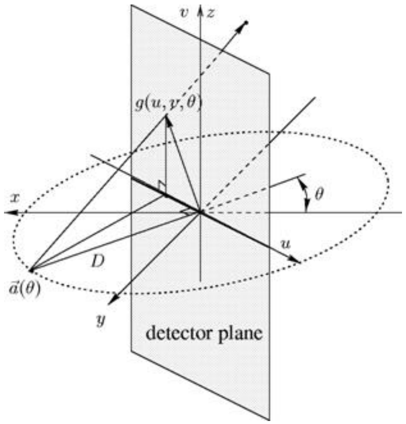
Family of Fast Algorithms



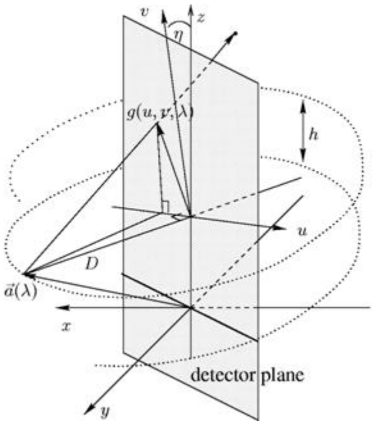
Parallel Beam



Fan Beam



Circular Cone Beam



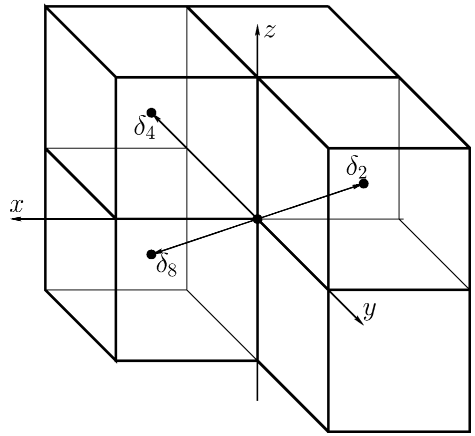
Helical Cone Beam

Extensions to Divergent-Beam Geometries

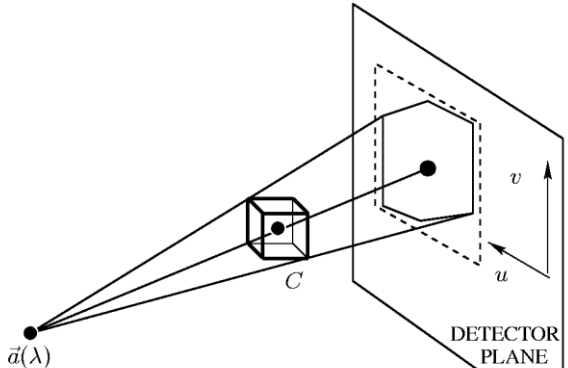
	Parallel Beam	Fan Beam	Cone Beam
FBP Algorithm	✓	✓	✓
Sampling Theorem	✓	✓ *	✗
Shift Property	✓	✗	✗

Hierarchical 3D Cone Beam Backprojection

3-D, Divergent Beam create a more complex implementation



Octant
Decomposition



2D Shifts of
Projection Data

Summary

- The computational cost of Filtered Backprojection algorithms is dominated by the backprojection step
- Hierarchical algorithms reduce complexity of the backprojection operation, greatly accelerating the reconstruction process
- This technique can be applied to a variety of scanning geometries
- The same approach can be used to accelerate reprojection, and create fast iterative reconstruction algorithms

This Week

- Assigned Lab Projects Wrap Up
- Final Project Proposals Due Sunday
  - Make sure your project Deliverable and Milestones are well defined! Include both ***what*** your team will be doing and ***how*** that will be demonstrated!
- After break: Final Project Proposal presentations and Assigned Lab Demos
  - Note that the proposal and presentation are not the same thing