ASSESSING/CHOOSING MEDICAL ALGORITHMS FOR GPU SYSTEMS

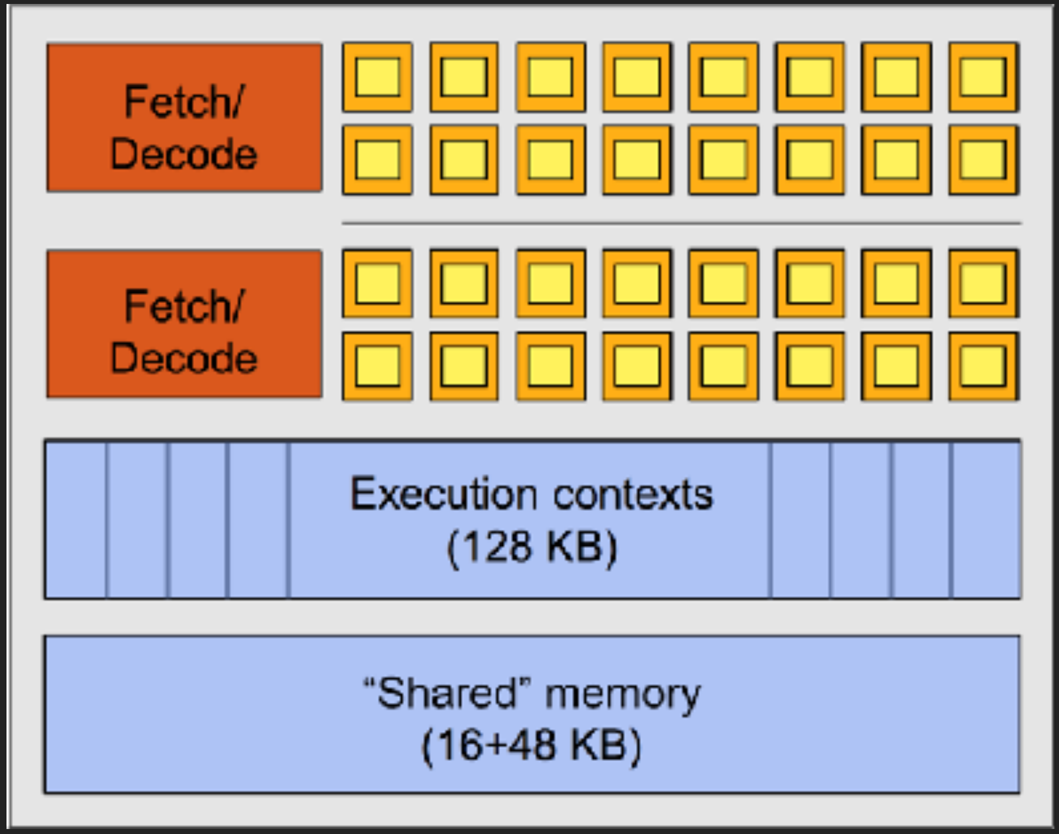
WILSON TANG

WHAT IS A GPU?

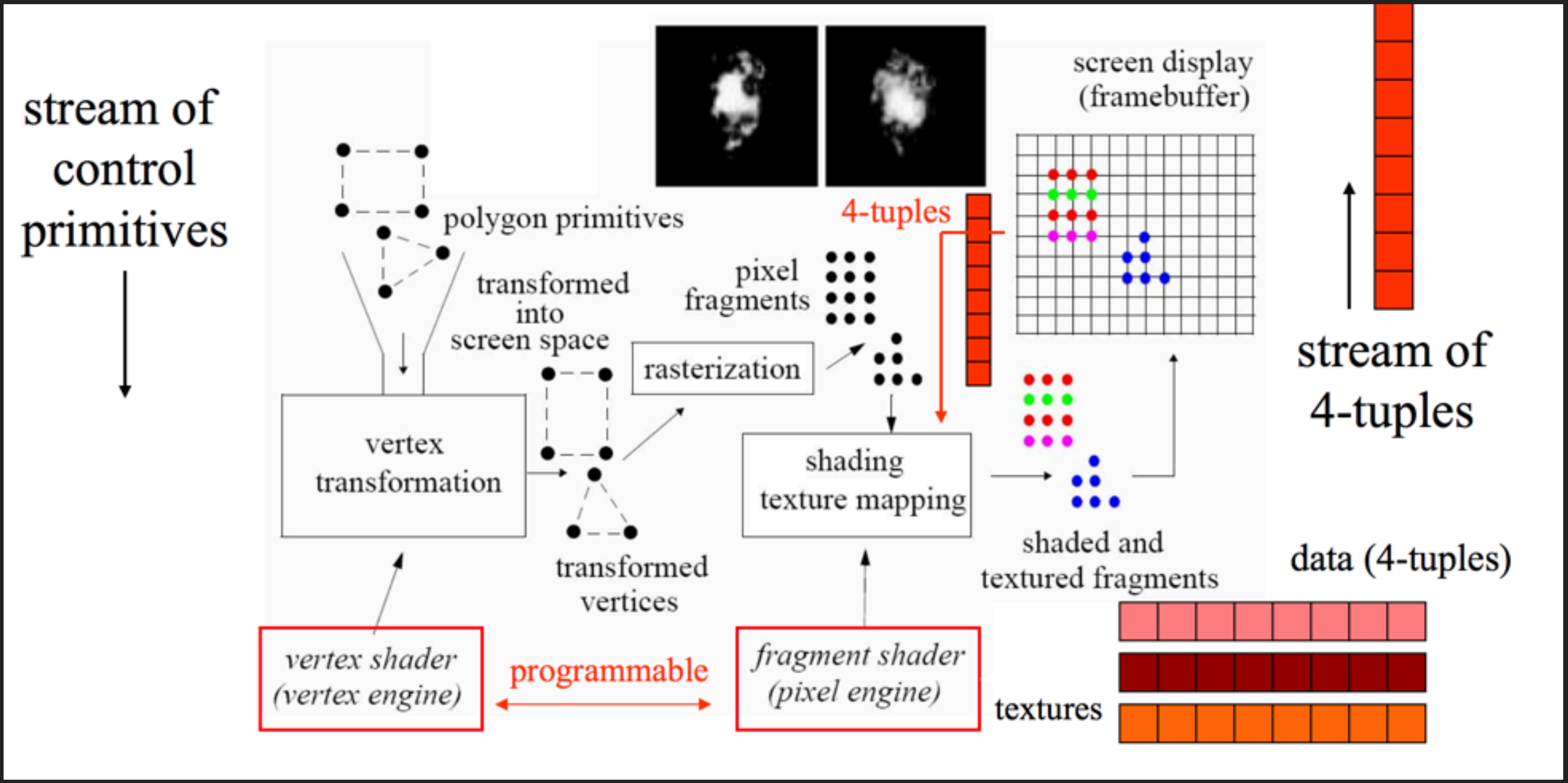
- Graphics Processing Unit
  - A Highly Parallel Programmable Processor (Pipeline)
    - Large Computational Capabilities
    - Single Instruction Multiple Data Parallelism
    - Throughput is Greater than Latency

GPU ARCHITECTURE

- Initially: Strict pipelined task-parallel architecture
- Modern: Single unified data-parallel programmable unit.



GPU COMPUTATIONAL VIEW



WHAT ALGORITHMS ARE GOOD FOR GPUS?

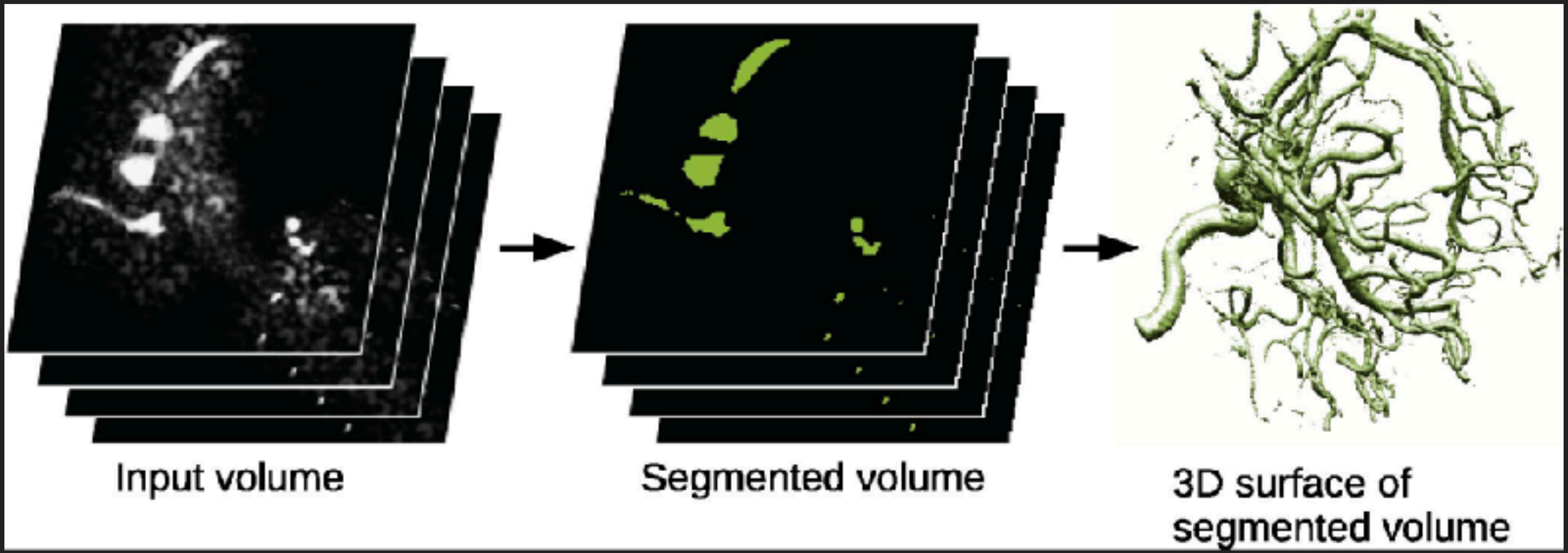
- Data Parallelism
  - Parallel Architecture of GPU
- High Thread Count
  - How many individual parts the calculation can be divided into and executed in parallel.
  - E.G 512x512 pixels could result in 262,144 threads
- No Branch Divergence (if-else)
  - If two or more threads in an AUE execute different execution paths, all execution paths have to be performed for all threads in that AUE
- Low Memory Usage
  - Limit in data memory, Use less data than total memory
- Little to No Memory Synchronization.
  - Atomic operations to synchronize require other operations to wait on it to finish

MEDICAL IMAGING AND GPUS

- Medical Imaging
  - Large Computational Requirements for Large Data Sets
    - MRI - k-space , CT - voxel
  - Real-Time Urgency in Clinical Setting
  - Common Computationally Complex Tasks
    - Segmentation: Identifying Specific Anatomical Structures
    - Reconstruction: Building 2D, 3D Images from Data

IMAGE SEGMENTATION

- Dividing the individual elements of an image or volume into a set of groups with a common property
    - E.G Identifying a Specific Organ in an Image



SEGMENTATION –BINARY THRESHOLDING

- Thresholding segments each voxel in the slice by assessing the voxel intensity with a threshold. The simplest binary segmentation follows the following form:

$$S(\overrightarrow{x}) = \begin{cases} 1 & \text{if } I(\overrightarrow{x}) \geqslant T \\ 0 & \text{else} \end{cases}$$

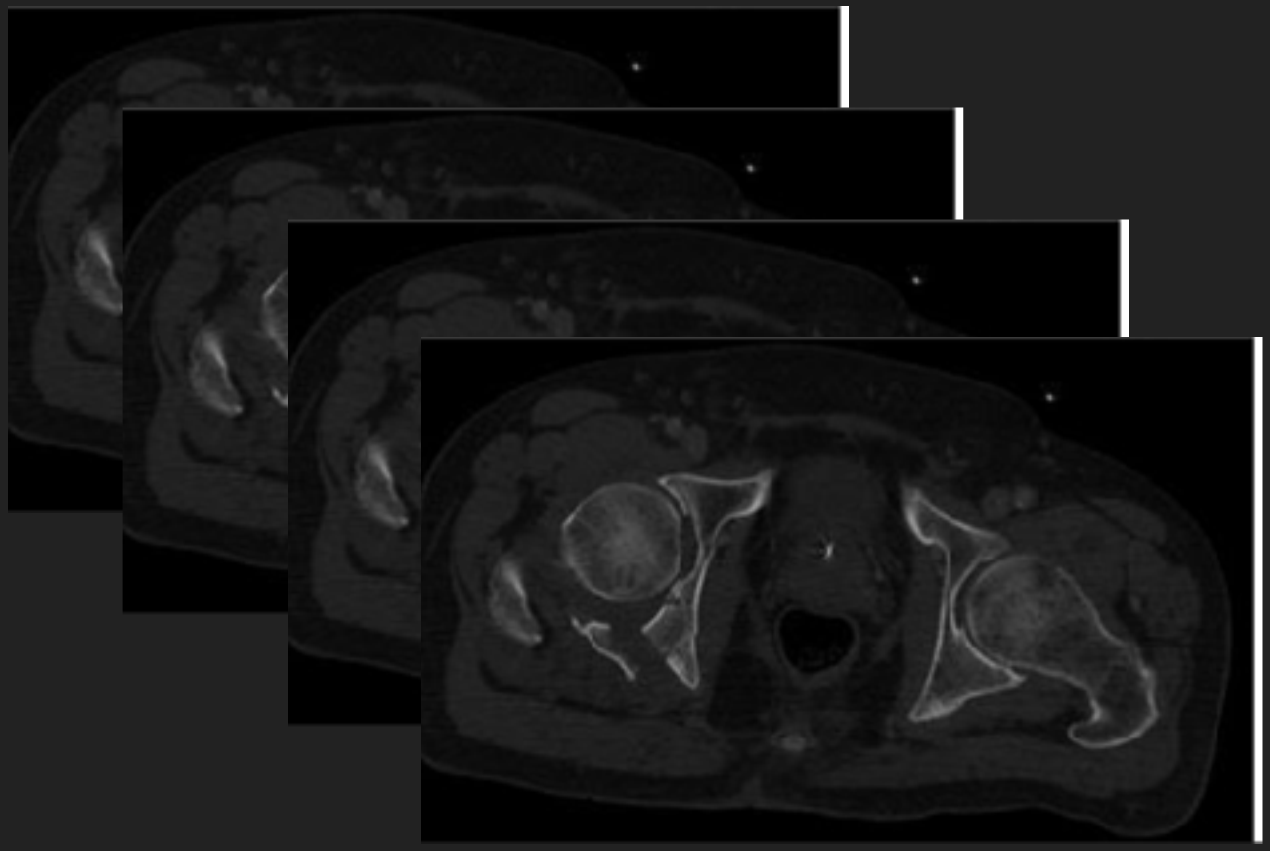
- Where T is the threshold, I(x) is the intensity, and S(x) is the resulting labeling.

ASSESSING BINARY THRESHOLDING ALGORITHM

- Data Parallelism: High. Completely data parallel, each voxel can be classified independently.
- Thread Count: High. Equal to the number of voxels/pixels processed.
- Branch Divergence: None
- Data Memory: Low. Only data needed to be stored is the S(x), labeled results
- Data Synchronization: None. No need for synchronization due to complete parallel segmentation
- Medical Image Timing: Urgent to Medium. Segmentation is typically used as a further diagnosis step in assessing serious diseases.

RECONSTRUCTION – COMPUTERIZED TOMOGRAPHY IMAGING

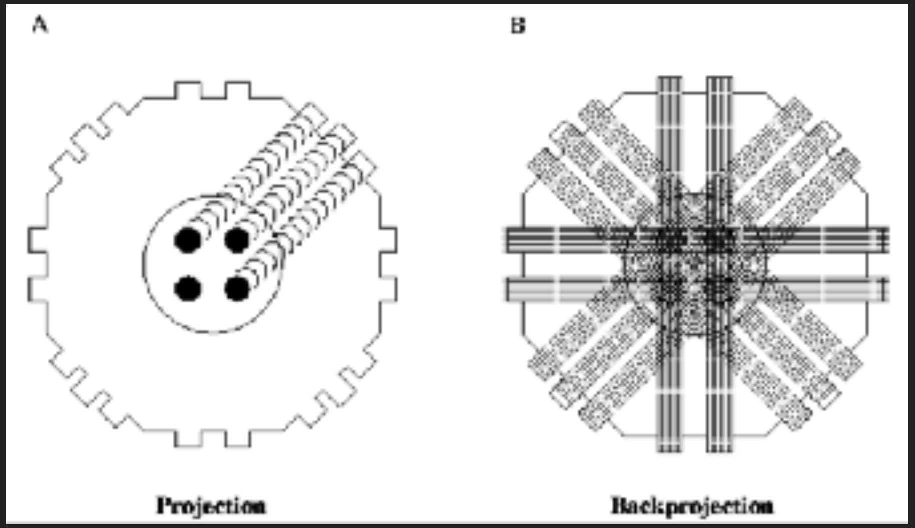
- CT Scan
  - Multiple X-ray images around body (1D) -> Cross-section of body (2D) -> Section of the body (3D)



CT IMAGE RECONSTRUCTION

- To reconstruct these images from a finite number of x-ray projections the detectors must measure a line integral of the X-ray attenuation through the scanned object.

- Back projection or an inverse Radon transform to reconstruct the original density of the image. Filtered projections are mapped to voxels(v_j) in parallel and used to reconstruct a 3D image.



$$P: p_i = \sum_{j=0}^{N^3-1} \left(v_j \cdot w_{ij} \right) \qquad B: v_j = \sum_{i=0}^{M_\varphi - 1} \left(p_i \cdot w_{ij} \right)$$

FBP

$$v_j = \sum_{p_i \in P_{set}} p_i w_{ij\_fdk} = \boxed{\sum_{p_i \in P_{set}} B(S)}$$

S: scanner projections

I: identity projection/volume

ASSESSING THRESHOLDING ALGORITHM

- Data Parallelism: High, the FBP algorithm can be executed on multiple scanner projections simultaneously before aggregating the data.
- Thread Count: High, Equal to the number of scanner projections needed to be processed.
- Branch Divergence: None
- Data Memory: High. Storage needed for multiple steps and the final image.
- Data Synchronization: Low, Data will need to be synchronized after the initial processing.
- Medical Image Timing: Urgent, CT scans can be used in diagnosing injuries such as internal bleeding and complicated fractures after accidents requiring low turn around times.

MOVING FORWARD/POTENTIAL EXTENSIONS

- Actual Implementation
  - CUDA Optimization on GPUs
- Potential for 1-2 orders of magnitude in timing while maintaining image quality