The Analysis and Classification of Images

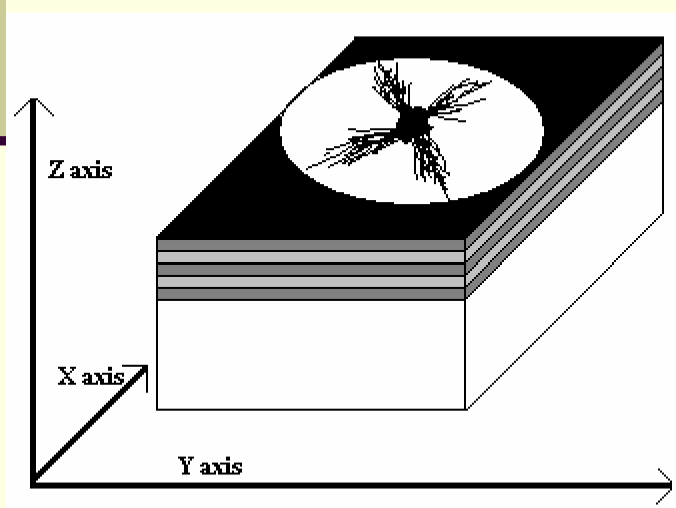
Generated by X-Ray Computed Tomography

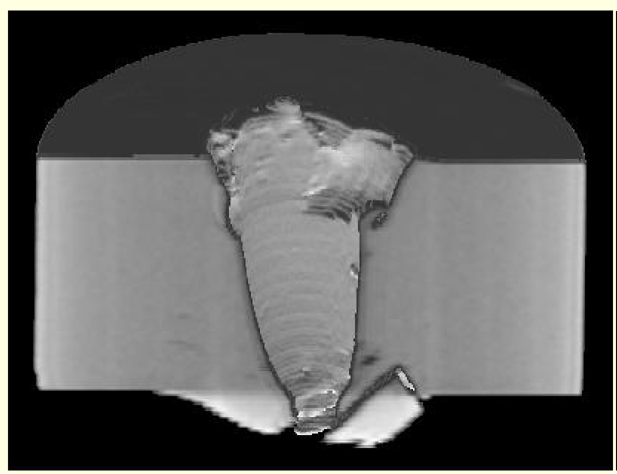
By: Michael Lowery University of California – Santa Cruz

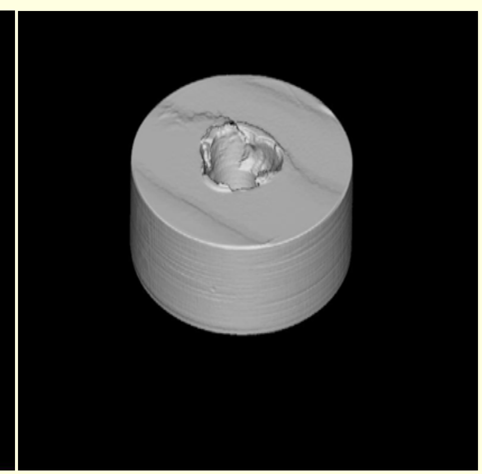
In Collaboration with: Jeff Wheeler, University of Kentucky – Lexington, KY

Introduction and Overview

- X Ray Computed Tomography (XCT)
  - Python script used to convert XCT images to binary format
  - Images processed based on XCT image size
    - 512x512 Image would have a 512x512 pixel matrix
  - Calculations based on Central Axis of Penetration

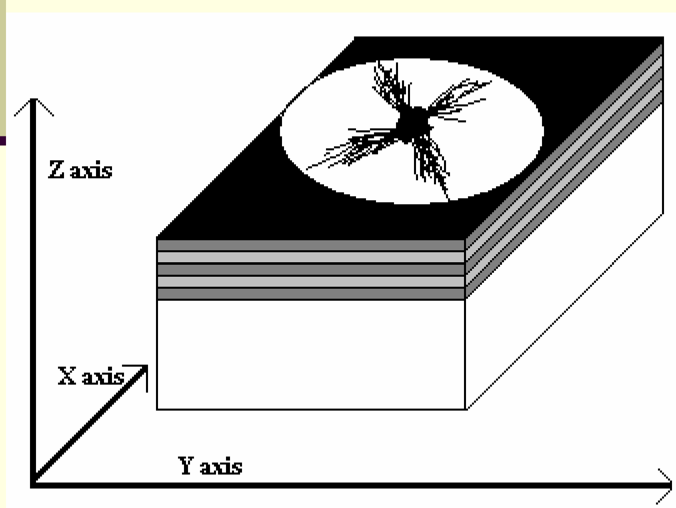


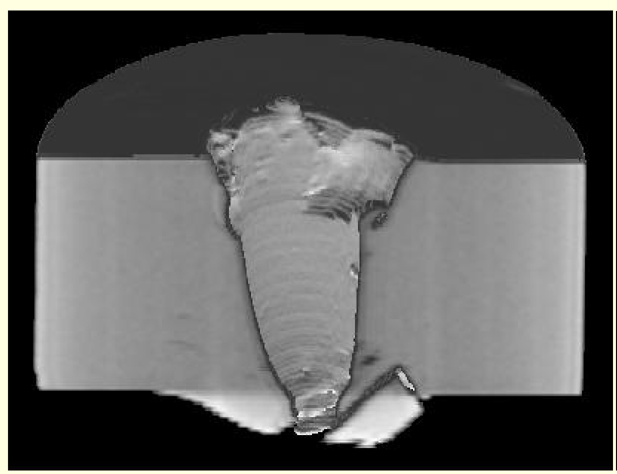


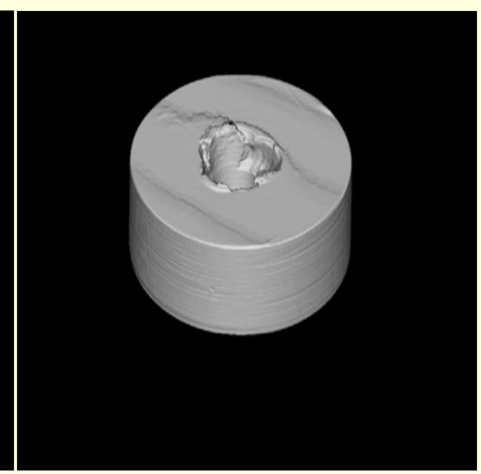


Introduction and Overview

- X Ray Computed Tomography (XCT)
  - Python script used to convert XCT images to binary format
  - Images processed based on XCT image size
    - 512x512 Image would have a 512x512 pixel matrix
  - Calculations based on Central Axis of Penetration

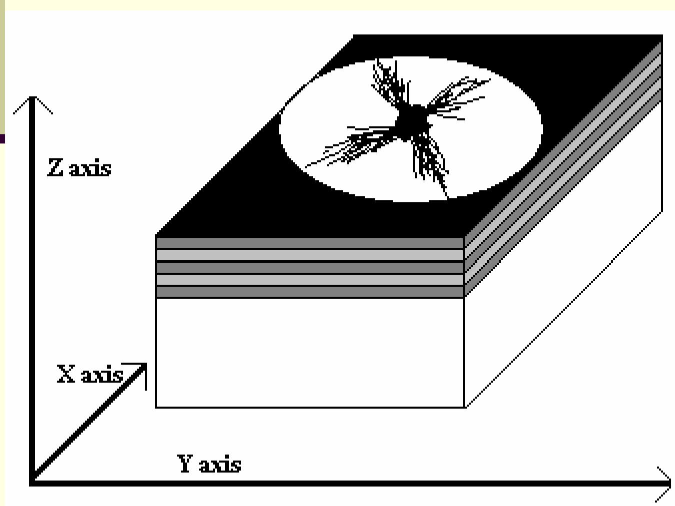


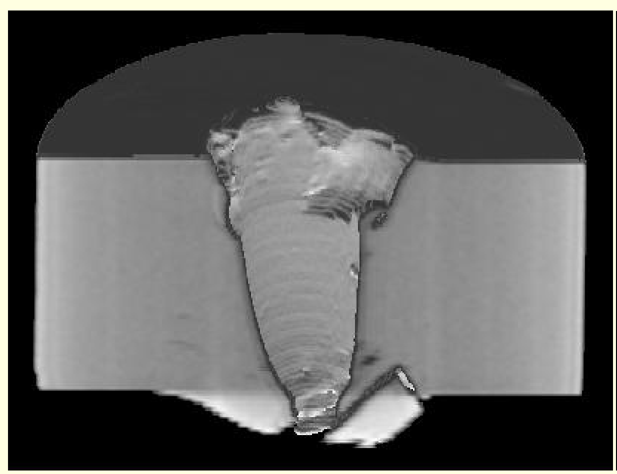


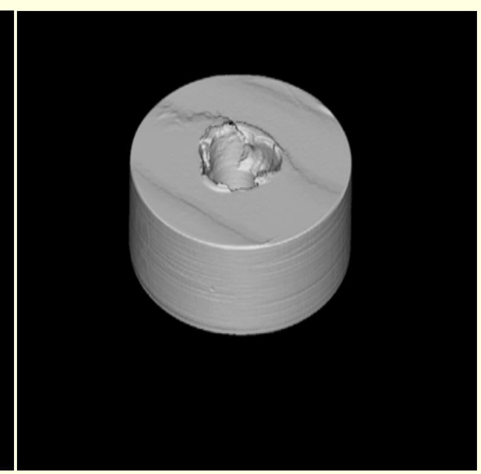


Introduction and Overview

- X Ray Computed Tomography (XCT)
  - Python script used to convert XCT images to binary format
  - Images processed based on XCT image size
    - 512x512 Image would have a 512x512 pixel matrix
  - Calculations based on Central Axis of Penetration

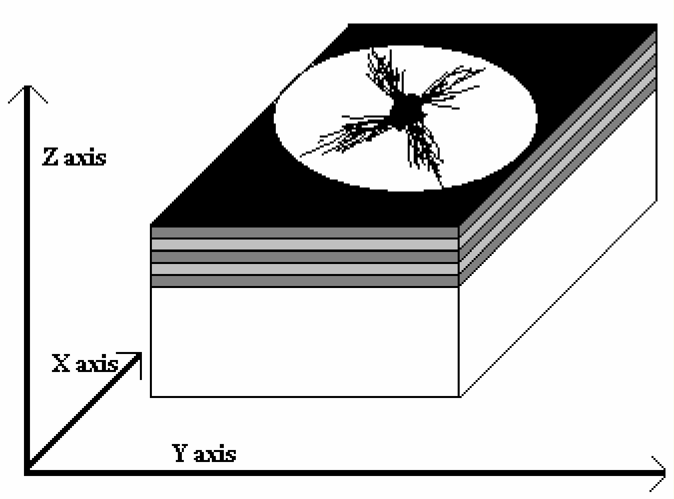


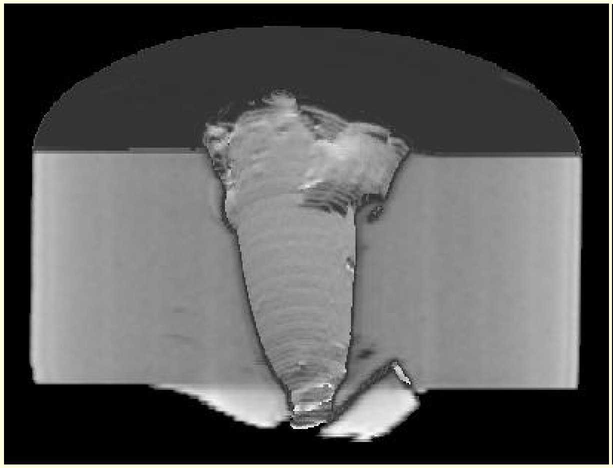




Introduction and Overview

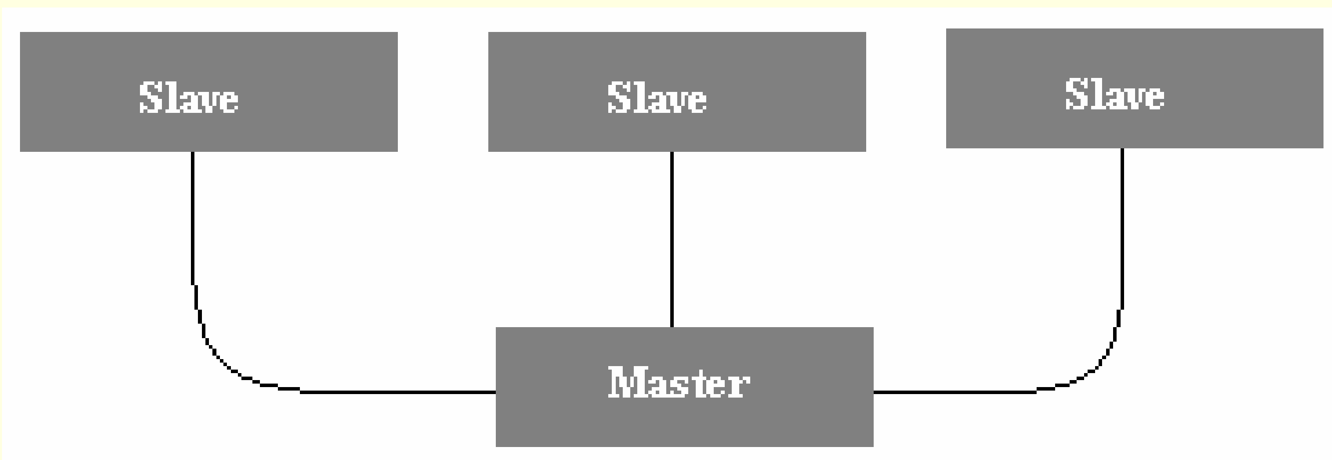
- X Ray Computed Tomography (XCT)
  - Python script used to convert XCT images to binary format
  - Images processed based on XCT image size
    - 512x512 Image would have a 512x512 pixel matrix
  - Calculations based on Central Axis of Penetration





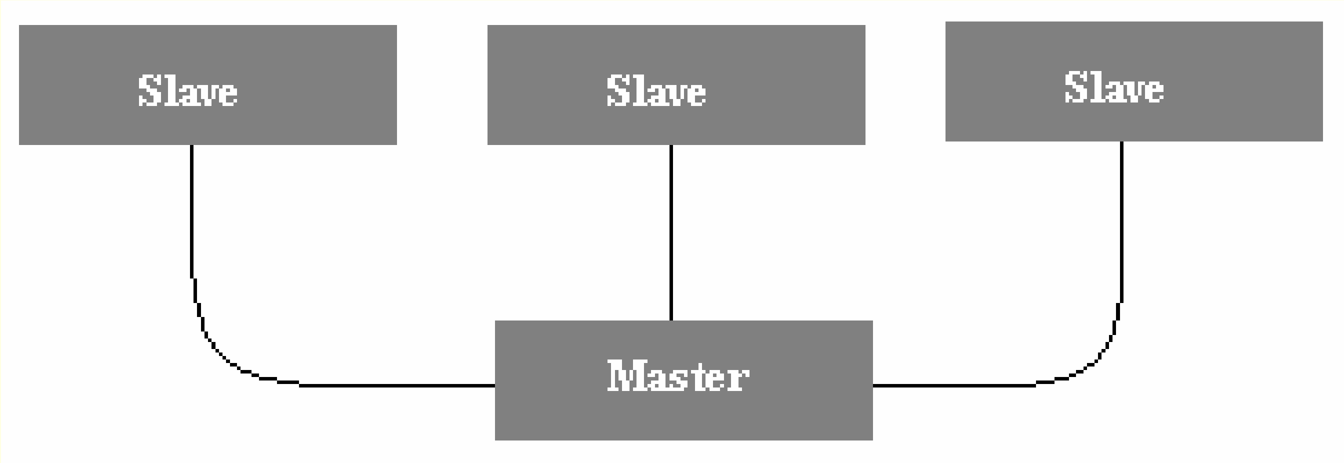
Computing Environment

- Parallel Computing Hardware
  - 8x 2Ghz Dell Precision Workstations w/1GB Ram
  - Interconnection via Gigabit Ethernet
- Parallel Computing Software
  - OSCAR (Open Source Cluster Application Resources) http://oscar.openclustergroup.org
    - Communication between nodes facilitated via LAM-MPI
- Parallel Algorithm is Embarrassingly Parallel
    - Work is sent to nodes based on a simple work distribution algorithm
      - Node 1 gets 10%, Node 2 gets 10% …
    - Upon Completion, all finished work is collected by the Master Node



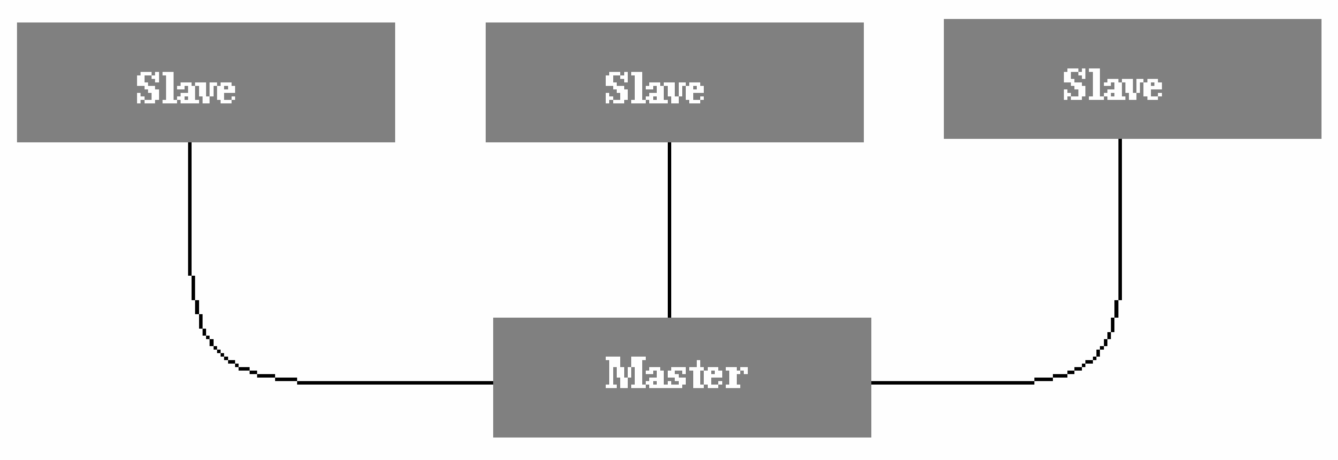
Computing Environment

- Parallel Computing Hardware
  - 8x 2Ghz Dell Precision Workstations w/1GB Ram
  - Interconnection via Gigabit Ethernet
- Parallel Computing Software
  - OSCAR (Open Source Cluster Application Resources) http://oscar.openclustergroup.org
    - Communication between nodes facilitated via LAM-MPI
- Parallel Algorithm is Embarrassingly Parallel
    - Work is sent to nodes based on a simple work distribution algorithm
      - Node 1 gets 10%, Node 2 gets 10% …
    - Upon Completion, all finished work is collected by the Master Node



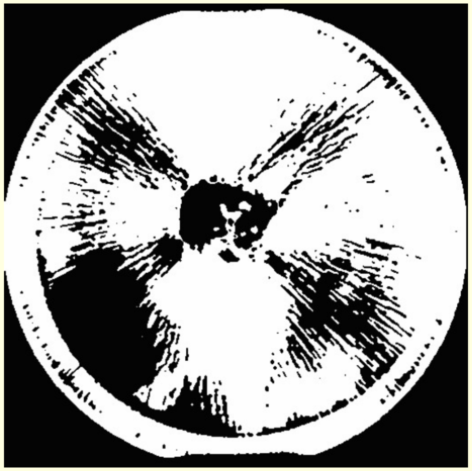
Computing Environment

- Parallel Computing Hardware
  - 8x 2Ghz Dell Precision Workstations w/1GB Ram
  - Interconnection via Gigabit Ethernet
- Parallel Computing Software
  - OSCAR (Open Source Cluster Application Resources) http://oscar.openclustergroup.org
    - Communication between nodes facilitated via LAM-MPI
- Parallel Algorithm is Embarrassingly Parallel
    - Work is sent to nodes based on a simple work distribution algorithm
      - Node 1 gets 10%, Node 2 gets 10% …
    - Upon Completion, all finished work is collected by the Master Node



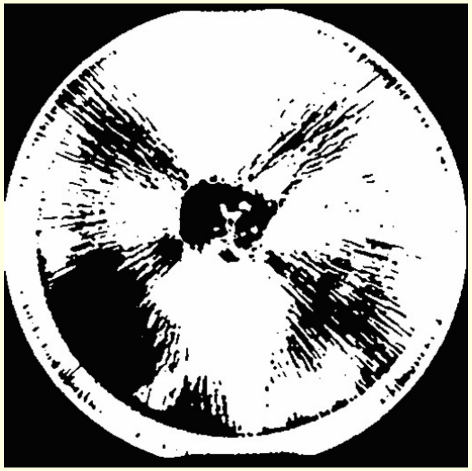
Calling Function and Notation

- Database Conventions and Notation:
  - The image data is stored in a vector, and accessed via special function
    - Given the (X, Y, Z) coordinate, the function returns it's value for analyses
      - Allows for multiple data extraction methodologies with out drastic code changes
  - Analyses Results Access:
    - F(X , Y , Z)
      - X : Method of decomposition
      - Y : Subsection size (User defined)
      - Z : Current subsection being analyzed
    - Decomposition Method:
      - 1) Square sub matrices
      - 2) Radian wedge slices
      - 3) Annular ring segments



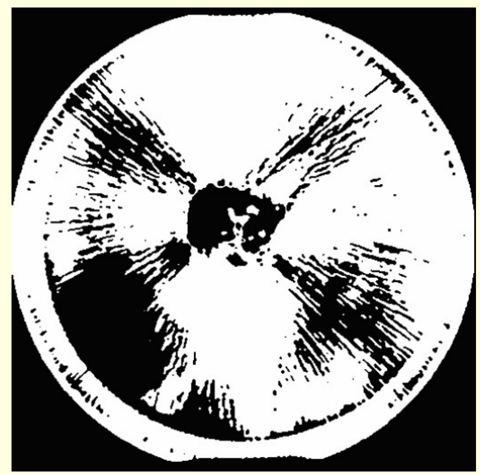
Calling Function and Notation

- Database Conventions and Notation:
  - The image data is stored in a vector, and accessed via special function
    - Given the (X, Y, Z) coordinate, the function returns it's value for analyses
      - Allows for multiple data extraction methodologies with out drastic code changes
  - Analyses Results Access:
    - F(X , Y , Z)
      - X : Method of decomposition
      - Y : Subsection size (User defined)
      - Z : Current subsection being analyzed
    - Decomposition Method:
      - 1) Square sub matrices
      - 2) Radian wedge slices
      - 3) Annular ring segments



Calling Function and Notation

- Database Conventions and Notation:
  - The image data is stored in a vector, and accessed via special function
    - Given the (X, Y, Z) coordinate, the function returns it's value for analyses
      - Allows for multiple data extraction methodologies with out drastic code changes
  - Analyses Results Access:
    - F(X , Y , Z)
      - X : Method of decomposition
      - Y : Subsection size (User defined)
      - Z : Current subsection being analyzed
    - Decomposition Method:
      - 1) Square sub matrices
      - 2) Radian wedge slices
      - 3) Annular ring segments



Square Matrix Decomposition

- Assume each Image is a 512 x 512 binary map
  - Divide the map into 32 x 32 pixel sub matrices

    Leaving a square of 16 isolated sub matrices

Use Calling Function F(1, 32, 7) to access sub divided matrices of 32 pixels square, and run average destruction analysis for image 7 if it isn't already available

Square sub matrices are the simplest decomposition, they allow rapid analysis and easy cross referencing between stored images

Square Matrix Decomposition

- Assume each Image is a 512 x 512 binary map
  - Divide the map into 32 x 32 pixel sub matrices
    Leaving a square of 16 isolated sub matrices

Use Calling Function F(1, 32, 7) to access sub divided matrices of 32 pixels square, and run average destruction analysis for image 7 if it isn't already available

Square sub matrices are the simplest decomposition, they allow rapid analysis and easy cross referencing between stored images

Square Matrix Decomposition

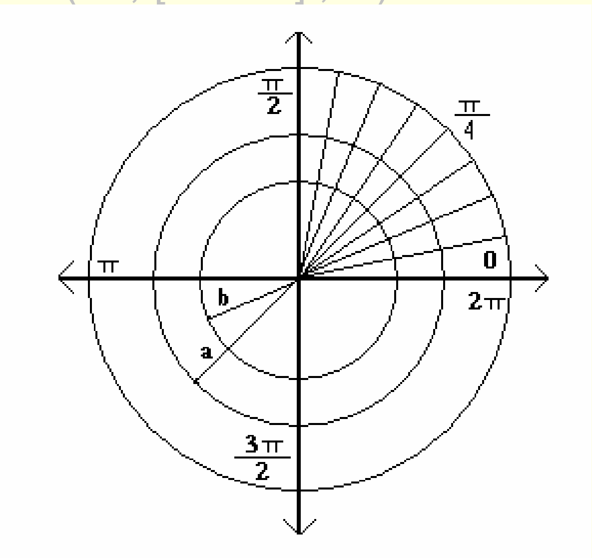
- Assume each Image is a 512 x 512 binary map
  - Divide the map into 32 x 32 pixel sub matrices
    Leaving a square of 16 isolated sub matrices

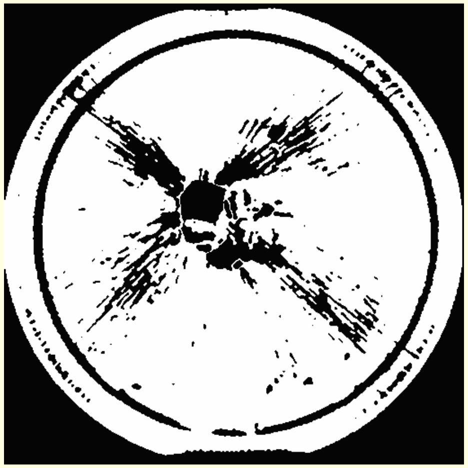
Use Calling Function F(1, 32, 7) to access sub divided matrices of 32 pixels square, and run average destruction analysis for image 7 if it isn't already available

Square sub matrices are the simplest decomposition, they allow rapid analysis and easy cross referencing between stored images

Radian Wedge Decomposition

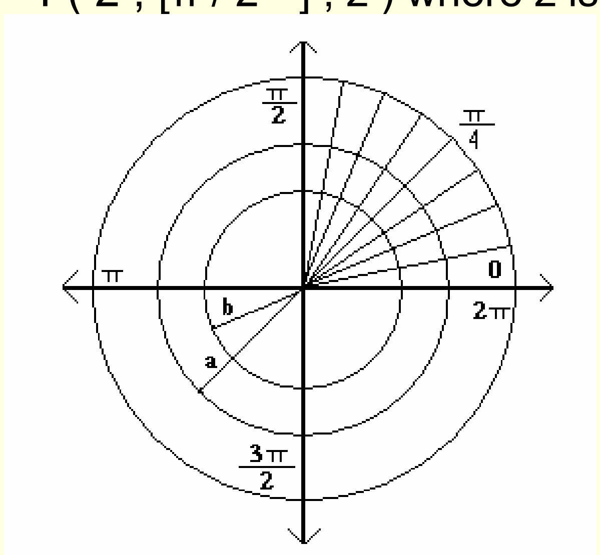
- Image Interval :
  - Let n range over the interval [0 : 2 π] in [π / 2^n] increments (π / 8, π / 16, π / 32 …)
  - Let (a > b) : Area in the annulus is defined as: $\pi(a^2 - b^2)$
    - User defined and adjustable depending on detail
    - Run MPI image analysis to determine destruction
    - Comparisons can be based on corresponding wedge segments from different images
    - Sections are accessed via the calling function: F(2 , [π / 2^n] , z) where z is the location of the marker

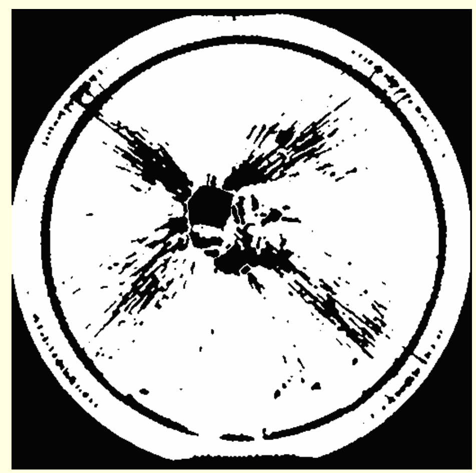




Radian Wedge Decomposition

- Image Interval :
  - Let n range over the interval [0 : 2 π] in [π / 2 ^{n}] increments (π / 8, π / 16, π / 32 …)
  - Let (a > b) : Area in the annulus is defined as: $\pi(a^2 - b^2)$
    - User defined and adjustable depending on detail
    - Run MPI image analysis to determine destruction
    - Comparisons can be based on corresponding wedge segments from different images
    - Sections are accessed via the calling function: F(2 , [π / 2 ^{n}] , z) where z is the location of the marker





Library and Processes

- Library coded using C++, Python, and mySQL
  - Calling function F(x , y , z) used to compare individual sub sections
    - Results that are not available are automatically computed and added to the database
  - Stored data can be called and its average compared with our current data

Variations and PhotoShop Filters

- PhotoShop Sharpening/Blurring Filters
  - Blurring filters decrease number of pixels
    - Borderline edges are smoothed creating a binary image with less detail – i.e. less pixels
    - Fewer pixels streamline processing
- Cross referencing can be done between different data extraction methods to verify accuracy between techniques.
- Currently under development: Graphical User Interface (GUI) for a user friendly version.

Variations and PhotoShop Filters

- PhotoShop Sharpening/Blurring Filters
  - Blurring filters decrease number of pixels
    - Borderline edges are smoothed creating a binary image with less detail – i.e. less pixels
    - Fewer pixels streamline processing
- Cross referencing can be done between different data extraction methods to verify accuracy between techniques.
- Currently under development: Graphical User Interface (GUI) for a user friendly version.

Variations and PhotoShop Filters

- PhotoShop Sharpening/Blurring Filters
  - Blurring filters decrease number of pixels
    - Borderline edges are smoothed creating a binary image with less detail – i.e. less pixels
    - Fewer pixels streamline processing
- Cross referencing can be done between different data extraction methods to verify accuracy between techniques.
- Currently under development: Graphical User Interface (GUI) for a user friendly version.

Conclusions and Summary

- Images are analyzed in the following manner:
  - Decomposed from n x m into sub squares, radian wedges, or annular rings
  - MPI parallelization computes average destruction percentages
  - Image Analyses stored and accessed via
    F(x , y , z)
  - Cross referencing is done from images already stored in the library
  - Finally, images can be altered in Photoshop to net different results

Conclusions and Summary

- Images are analyzed in the following manner:
  - Decomposed from n x m into sub squares, radian wedges, or annular rings
  - MPI parallelization computes average destruction percentages
  - Image Analyses stored and accessed via

    F(x , y , z)

  - Cross referencing is done from images already stored in the library
  - Finally, images can be altered in Photoshop to net different results

Personal Info

- Author:
  - Michael Lowery
    - University of California – Santa Cruz
    - Computational Mathematics
    - Phone:
      - (408) 712 - 7356
    - Email:
      - mplowery@gmail.com
    - Mailing Address:
      - 2081 Bohannon Dr.
      - Santa Clara, Ca 95050
- Co Author:
  - Jeffrey Wheeler
    - University of Kentucky – Lexington KY
  - Email:
    - birdjeff@scipp.ucsc.edu