# Interactive Visualization of Very Large Multiresolution Scientific Data Sets

UNIVERSITY of NEW HAMPSHIRE

R. Daniel Bergeron

## Modern Science Research

- Much of today's science research is driven by 3 principal components†:
  - Data
    - sampled data or simulation output
  - Data
    - many Gb, even Tb - too much to visualize
  - Data
    - many Tb, even Pb - how to store and access it

†Taken from the realtor's mantra: location, location, location



## Modern Science Research

- Much of today's science research is driven by 3 principal components:
  - Data
    - generation
  - Data
    - accessing/visualizing/analyzing
  - Data
    - understanding



## Data Visualization

- Visualization goal
  - *presentational* visualization
    - how do you show something in the data
  - *exploratory* visualization
    - how do you learn something from the data
- Exploratory visualization mantra†
  - *overview* then *focus*

†Due to Ben Schneiderman

# Overview then Focus

- Use a low resolution data representation to get an *overview* visualization
- *Zoom in* to regions of interest
  - *visual* zoom and

    simultaneous
  - *resolution* zoom

  data size stays constant (more or less)



# Case Study Overview

- Challenges of visualizing simulation data
- Focus on unsteady MHD simulation
- Application framework
  - Time Series Data
    - Multi/Adaptive resolution techniques
    - Error model
  - STAR data
    - Space Time Adaptive Resolution data



# Very Large Datasets

- Numerical simulation produces GBs and TBs of time series data
- How can we visualize this *interactively* on a commodity workstation?
- Key ideas
  - overview then focus (the visualization mantra)
  - know the error in the data
  - only read what you need



# Interactive Visualization Model

- Generates multiresolution data (in both spatial and temporal domains)
- Initial view is at a coarse enough level to support interactivity (depends on platform)
- Zoom into spatially and/or temporally focused view at higher resolution
  - where the data is "interesting", and
  - where the data has high error
- Goal: memory demand stays constant

## Implementation Issues

- Multiresolution data generation and access
- Adaptive resolution data generation/access
- Efficient I/O and network access to multidimensional data
- Writing rendering algorithms for MR and AR data



## Support for Large Scientific Data

- Granite Scientific Database System (Java)
  - General support for rectilinear, multisource, multidimensional, multiresolution data
  - Special features for I/O optimization based on iteration-aware prefetching and caching
- STARview visualization environment (C++)
  - Focused on multiresolution time series data
  - Eases implementation of renderers



## STARview Goals

- Space Time multi/Adaptive Resolution data hierarchy
- Provide a transparent uniform resolution interface to MR and AR data so renderers don't have to know about it.
- Supports MR and AR data in both the spatial and temporal domains.
- Supports access to error data for the lower resolution representations

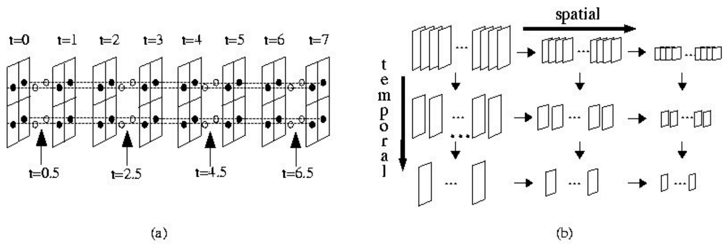


## STAR Data Model

- Space Time multi/Adaptive Resolution data hierarchy
- STAR Tree child node
  - reduced *spatial* resolution
  - reduced *temporal* resolution
- STARgen application creates hierarchy from original data
- Arbitrary mixing of spatial and temporal data
- Use wavelet transformation algorithm
- Generate error

# Space/Time Wavelets

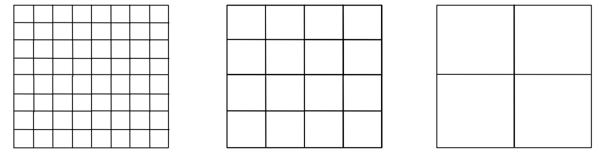
- Spatial wavelet transform applied to data from each step of time series
- Temporal wavelet transform applied to all data at corresponding positions in all steps



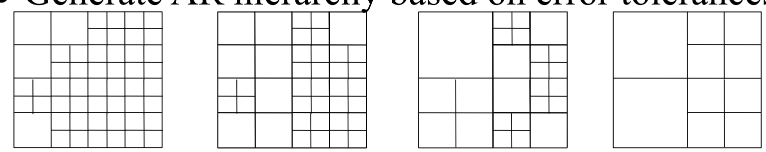



# Spatial Adaptive Resolution Data

- Given multiresolution hierarchy



- Generate AR hierarchy based on error tolerances

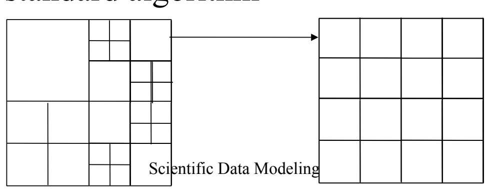


Error: < 5%   < 10%   < 15%   < 20%



# Using Spatial AR Data

- Using AR directly requires specialized algorithms
- Or, convert AR to uniform resolution
  - Pick target uniform resolution
  - Average higher resolution AR regions
  - Expand lower resolution AR regions
  - Apply standard algorithm





# Temporal Adaptive Resolution Data

- Want intelligent data reduction techniques
- Error tolerance δ used to remove less important time slices - those with less change
- Time between time steps is non-uniform
  - Recreate uniform sample via interpolation
- Framework allows any kind of interpolator

# Quality of MR and AR data

- Scientists do not like discarding data
- Integration of error with the data is key
- Uncertainty visualization informs scientist
- Only delete time steps *not significantly different* from surrounding steps (based on $\delta$)
- Only abstract spatial regions with low error
- Tradeoff is that we can handle larger data interactively



# Data Quality Issues

- Can you trust the low resolution data?
  - no (at least not blindly)
- Must compute *error* of low resolution data
  - error must be spatially (and temporally) computed, so you know where the data is unreliable
- Provide *error visualization* tools

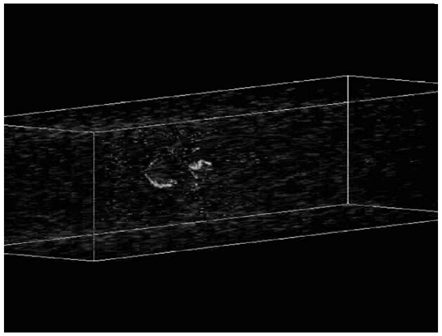


# Solar Wind Simulation

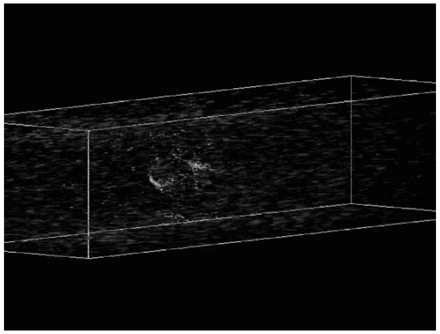
- Models interaction between solar wind and Earth's magnetosphere
- Simulation records magnetic field, particle velocity, and current density
- Data is a 3D time series
- Data points sampled on a structured grid
- 87 time steps, total data size is 15GB



# Solar Wind Unsteady Flow



392 x 112 x 112



196 x 56 x 56

Note: color is mapped to particle speed

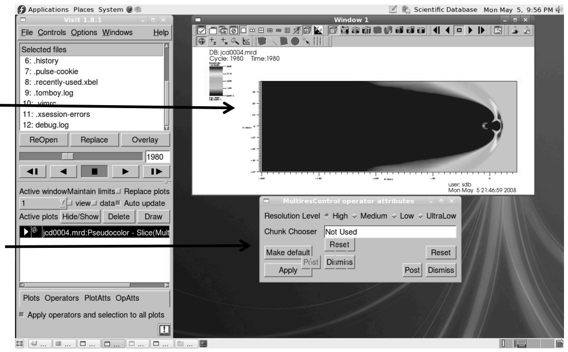
# STAR / VisIt Interface

- STAR database plugin
  - Accesses STAR multiresolution data hierarchy
- STAR operator plugin
  - User controls resolution via an operator plugin
  - Interaction with operator plugin triggers data reload



# STAR/VisIt MR Support

- VisIt state after a STAR data object opened
- 1 slice of one high resolution time step; shows density variate
- STAR operator dialog to control data resolution

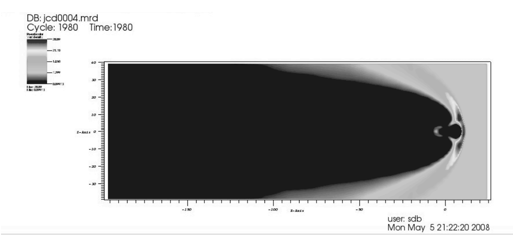


Any VisIt rendering can be applied to any compatible STAR data

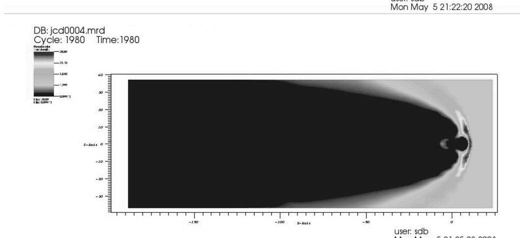


# STAR/VisIt MR Support 2

- Medium resolution



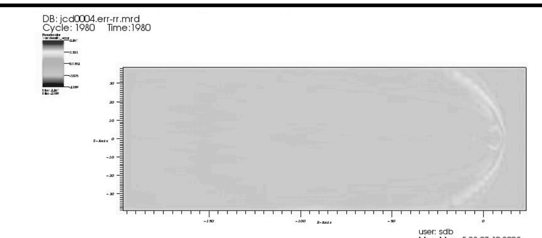

- Low resolution

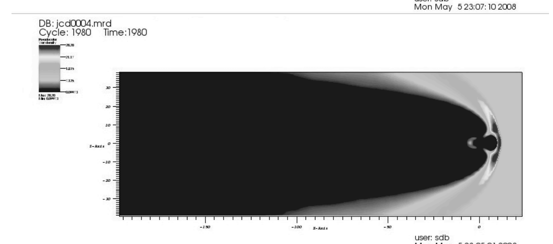





# STAR Error Data

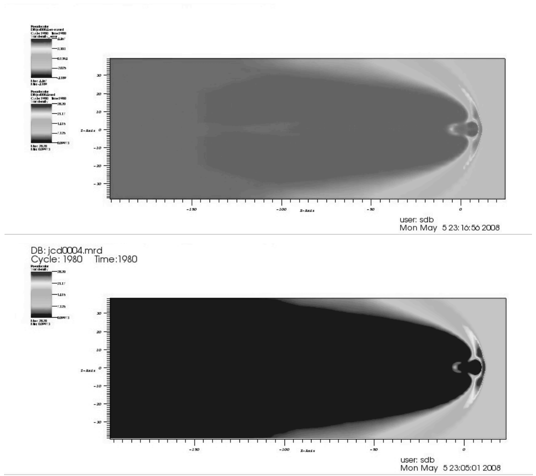
- STAR error data is generated at same resolution as the lower resolution data
- Top is error of resolution 2 and bottom is resolution 2 data.



# STAR/Visit Error Data

- Error is just another data set to VisIt; top is error data drawn with opacity at 50 % superimposed on the medium resolution data,
- VisIt lets you drag a slider to to change opacity dynamically or swap views between the error and data.






# Managing Large Data

- User specifies upper memory limit
- Time Series Data that exceeds this limit is loaded at a lower temporal resolution
- Intermediate slices are interpolated
- Scientist can zoom in spatially
  - Automatically increase spatial resolution in response to a reduction in spatial range



# Results

| Data Size | Show Error? | Memory Used for Data Storage | Average Frame Rate |
|---|---|---|---|
| 512x512x3000 | N | No Imposed Limit | ∅ |
| 512x512x3000 | N | 650MB Limit | 14 fps |
| | | | |
| 256x256x3000x2 | Y | No Imposed Limit | ∅ |
| 256x256x3000x2 | Y | 750MB Limit | 10 fps |
| 128x128x3000x2 | Y | No Imposed Limit | 12 fps |
| 64x64x3000x2 | Y | No Imposed Limit | 30 fps |

- Showing error requires twice as much data
- Keeping interesting data in memory yields interactive frame rates

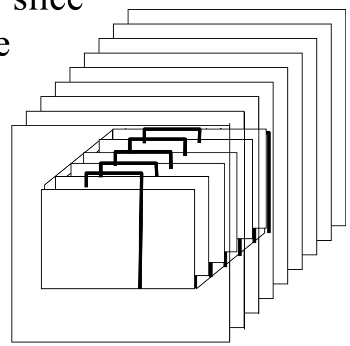


# Out-of-core Visualization

- MR and AR data and subset access reduce data needed to make a visualization
- Sometimes still need to create images from data that simply doesn’t fit in memory
- Many visualization techniques don't need all data in memory at once
- Interactivity, however, demands efficient I/O (or network) data retrieval

# I/O Optimization

- Consider a 3D matrix stored by slice and a subregion that is too large for available memory
  - If viewed along slice storage axis, access matches storage.
  - If viewed from right, access does not match storage



- Application creates an *iterator*
  - Defines access pattern <u>in advance</u>, so I/O system can predict what data to *pre-fetch* and save in *cache*



Text

# Iteration-Aware Caching

- Preliminary results are promising
  - 39GB visible woman data set
  - 2.5 to 12 <u>times</u> faster
- Same notion works to reduce network access costs for remote data
  - improvement achieved by addressing *latency overhead*



# Conclusions

- Principal goal: combine space and time multiresolution into unified data model
- Focus on simulation of MHD phenomena
- Integrate error model into application
- Make it useful for scientists creating simulations
- Minimize difficulty in creating renderers



# Recent Related Work

- Integrate MR error analysis into simulation [HiPC 2011]
  - Save data based on significance
  - Simulate at higher resolution, save at lower
- Lossy wavelet data compression [VDA 2012]
  - Save some detail coefficient blocks
  - Reduce precision of detail coefficients (to a byte)
  - Can reduce error significantly for small increase in space and IO