Visualization at Earth Simulator Center

Earth Simulator Center, JAMSTEC, Japan
Akira Kageyama

with F. Araki, S. Kawahara, and N. Ohno

Outline

1. Introduction
2. Our projects
    - MovieMaker
    - VFIVE
    - others
3. Future

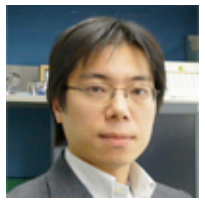
Visualization Group of our Center

6 research groups in Earth Simulator Center

- Geosciences
  - 2 groups on GCMs
    - Atmosphere, ocean, and coupled model.
  - 1 group on solid Earth
    - Geodynamo, mantle convection, plate, and earthquakes.
- Advanced simulation (holistic simulation) methods
  - 2 groups
    - Cloud formation, aurora, space plasmas, combustion, friction…
- Visualization
  - 1 group

Visualization Group

- Born in 2003
- 4 researchers
  - A. Kageyama (Group Leader)
  - F. Araki
  - S. Kawahara
  - N. Ohno

Hardwares

- Visualization computers
  - SGI Onyx3800
    - 12 processors, 24GB main memory, 11TB HDD
  - SGI Onyx4
    - 4 processors, 16GB
  - Apple XServe G5
    - 14 processors (dual core)
- CAVE
  - 4 screens (3m x 3m)
- Other available systems for the visualization
  - Earth Simulator
  - SGI Altix4700
  - NEC SX-8R

Visualization Requirements

- Simulation type
  - Fluid-type simulations >> MD-type simulations
  - GCMs, geodynamo, mantle convection, …
- Geometry
  - Spherical >> others
- Large data size
- Most of them prefer CUI rather than GUI.

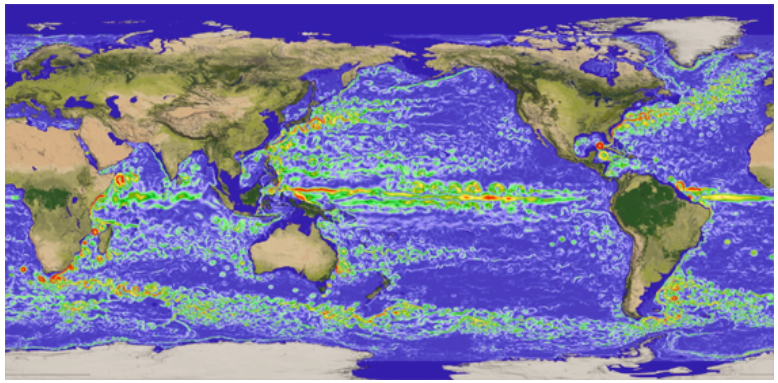
Typical data size

- For a still image
  - O(1) GB/ time step / variable
    - Atmospheric GCM
      - 3840 x 1920 x 96 x 4 B = 2.6 GB
    - Oceanic GCM
      - 3600 x 1500 x 54 x 4 B = 1.1 GB
    - Geodynamo
      - 1538 x 514 x 511 x 2 x 4B = 3.2 GB
- For a movie
  - 1000 time steps
  - O(1) TB / variable

  ==> *MovieMaker* project:

“Culture Complex”

- Atmospheric & oceanic researchers
  - No strong need for advanced visualization.
  - No strong need for 3-D visualization.
    - 2-D cross section, or a map, is just enough(?)
    - Division of work: Model builder, programmer…
  - A visualization software (*GrADS*) is just fine(?)



http://www.es.jamstec.go.jp/esc/research/AtmOcn/virtual_atm-ocn.en.html

“Culture Complex”

- Solid earth science and other groups
  - A researcher does everything.
  - Strong need for new visualization methods.
  - They want visualization software with source code.
    - To revise and run the code by themselves.
  - 3-D visualization is essential.
- Our visualization research is mainly driven by these groups.

Projects in Visualization Group

- *MovieMaker*
  - A parallel visualization software
- *VFIVE*
  - A virtual reality visualization software for CAVE.
- Others

Outline

1. Introduction
2. Our projects
   - MovieMaker
   - VFIVE
   - others
3. Future

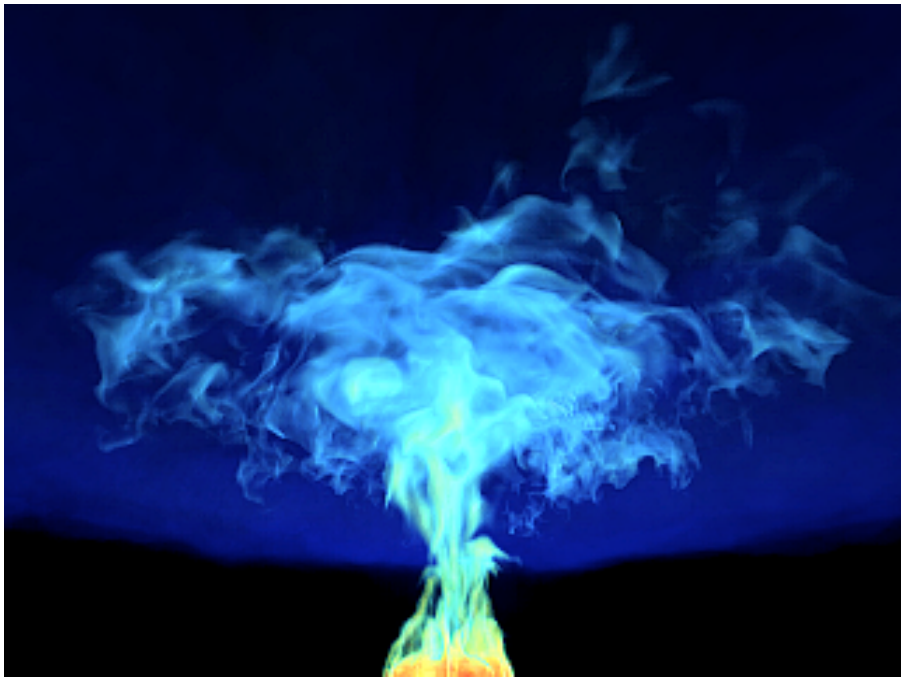
MovieMaker

- A parallel visualization software
- Any resolution
- Implemented visualization methods:
  - Volume rendering
  - Isosurface
  - Slice planes
  - Stream lines
- Developed by all group members.
  - Development leader: H. Uehara.

Uehata et al., *J. Plasma Physics*, 72, pp.841–844 (2006)

Snapshots from MovieMaker

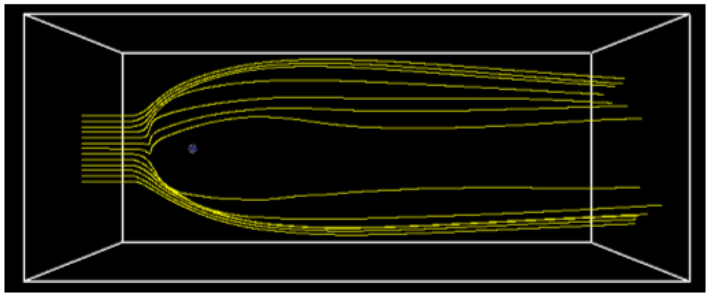
Eruption cloud simulation
by Y. Suzuki, IFREE, JAMSTEC

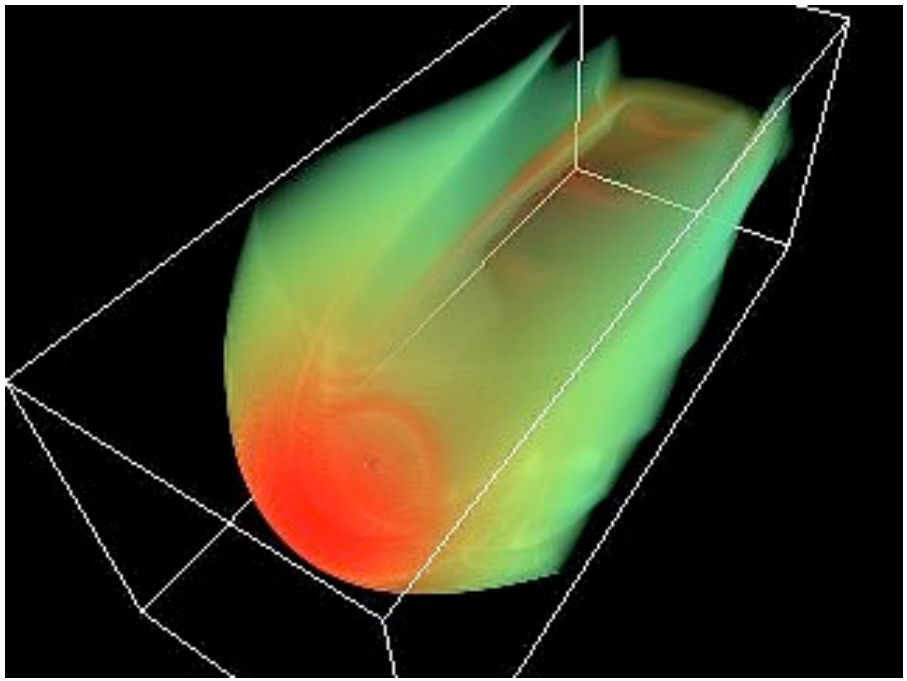


Grid size: 2040 x 2040 x 384 (= about 6GB single precesion data)
Image size: 1024 x 768
Rendering time per frame: 35 -- 101 sec

Snapshots from MovieMaker

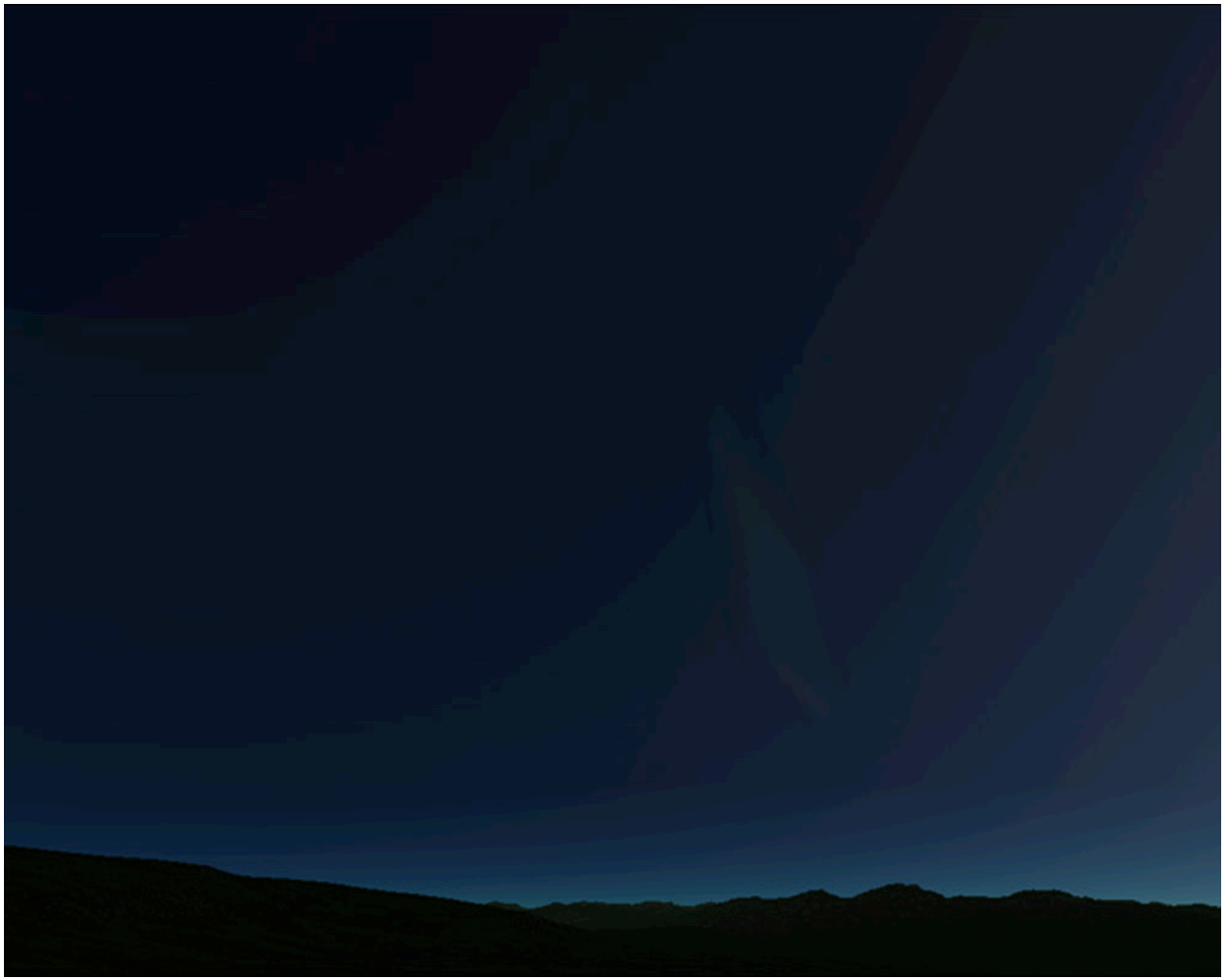
MHD simulation of Earth's magnetosphere
by Prof. T. Ogino, Nagoya, Univ.





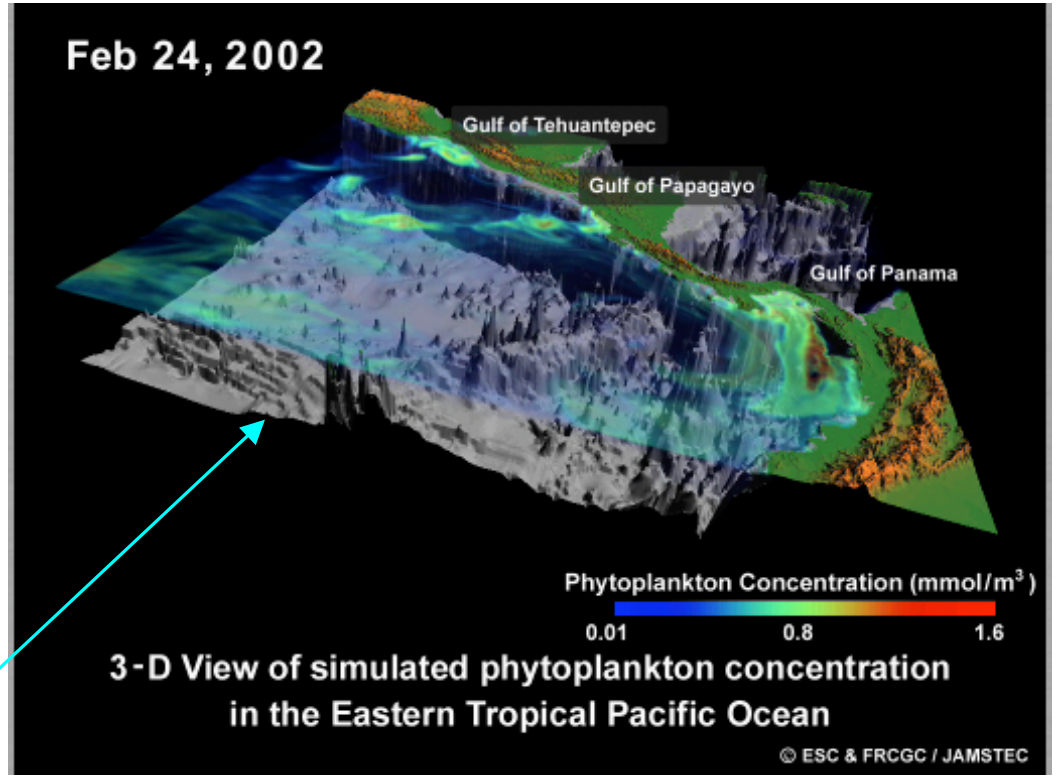
Grid size: 500 x 500 x 200
Rendering time: < 40 per frame.
By Onyx3800 12 cpu.

Examples of MovieMaker



Auroral formation

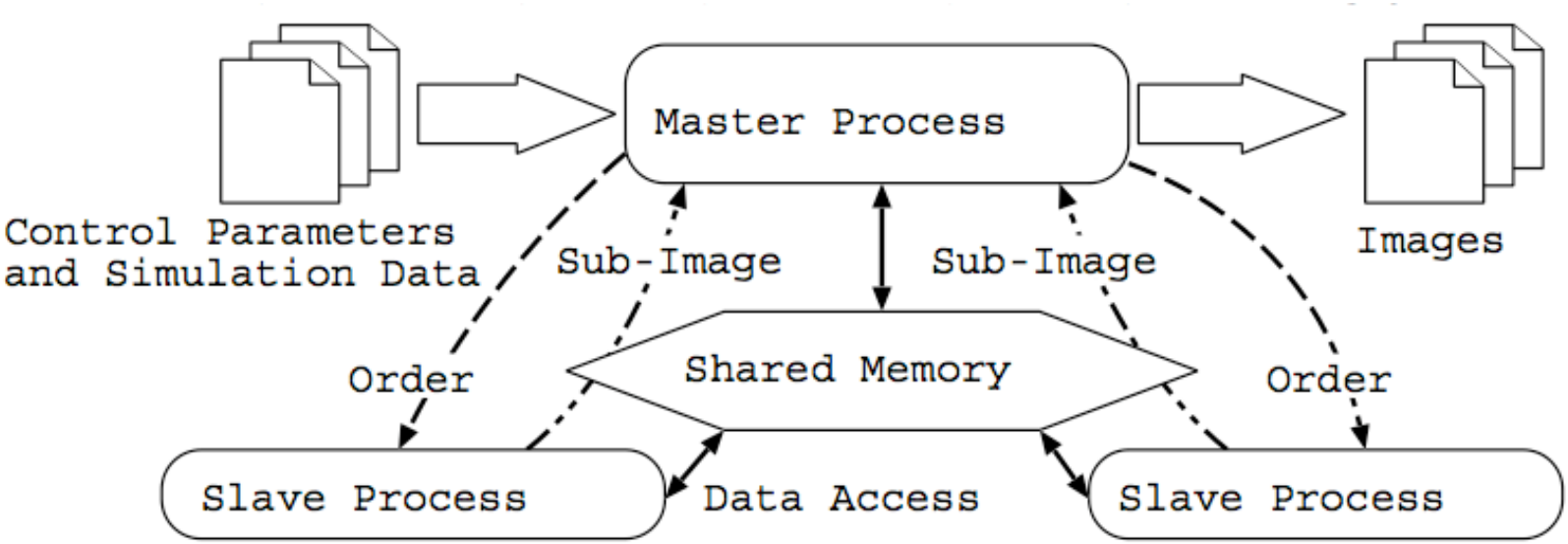
Snapshots from MovieMaker



Topography can be imposed.

Techniques in MovieMaker

- Master/Slave model for the parallel processing
  - Dynamic load balance

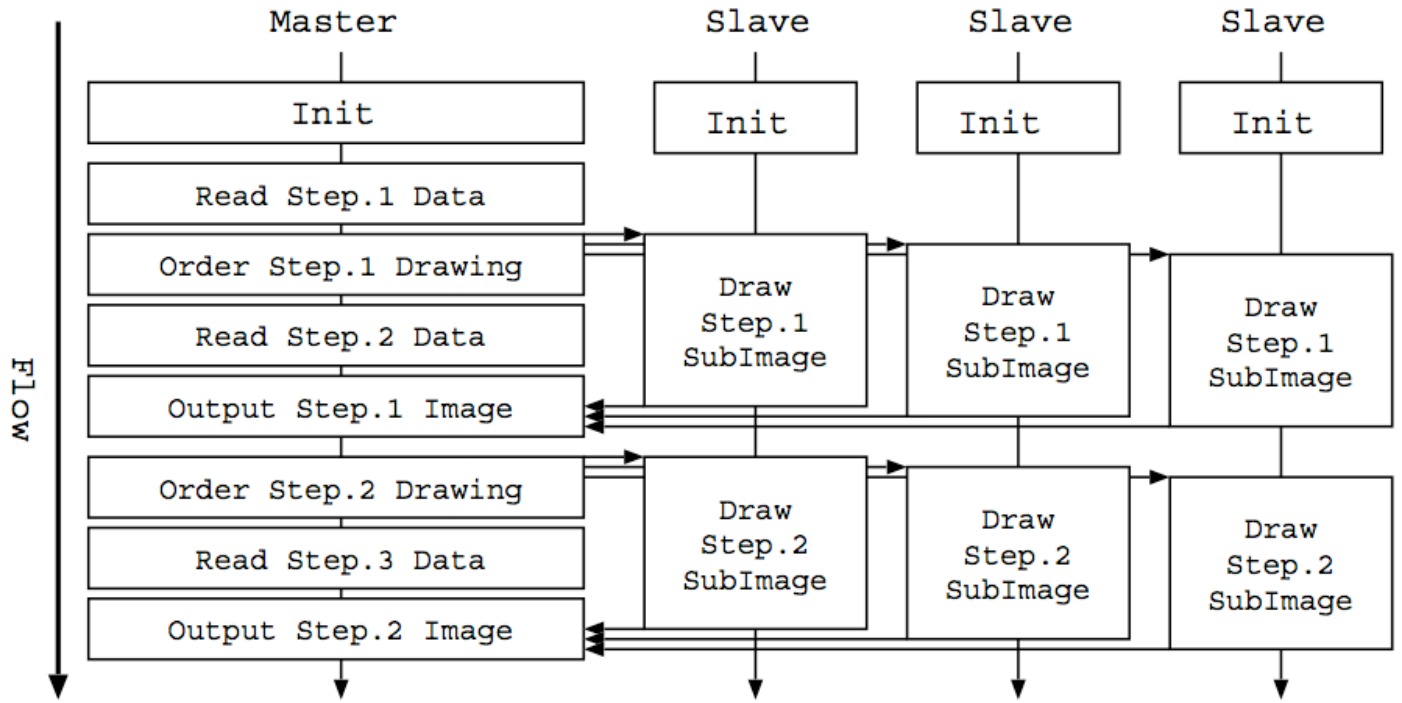


Techniques in MovieMaker

Overlapping of

(1) Data read for each time step (by the master process)

(2) Rendering of sub-images (by slave processes)



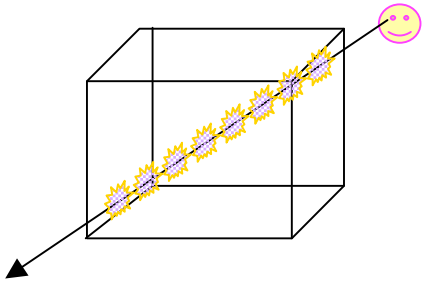
Variations of MovieMaker

- Base grid system:
  - MovieMaker on Cartesian coordinates
  - MovieMaker on spherical polar coordinates
  - MovieMaker on Yin-Yang coordinates
- Rendering method:
  - (Partially) hardware rendering version
  - Purely software rendering version, named "*Armada*", developed by N. Ohno

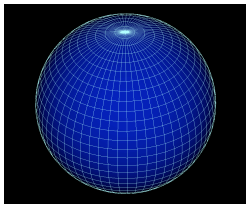
MovieMaker for different geometry

- Cartesian version

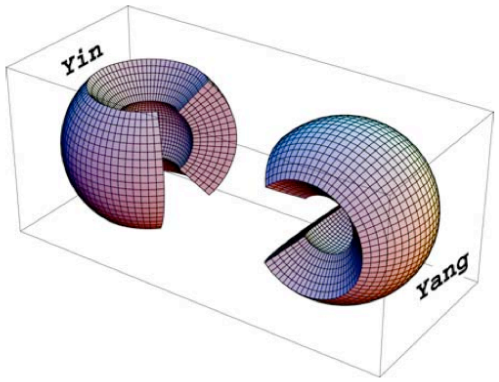
  Ray casting in the cartesian geometry

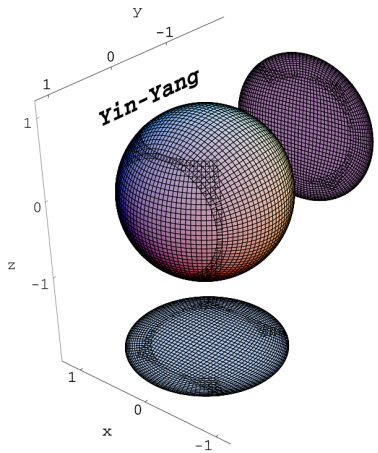


- Spherical polar version



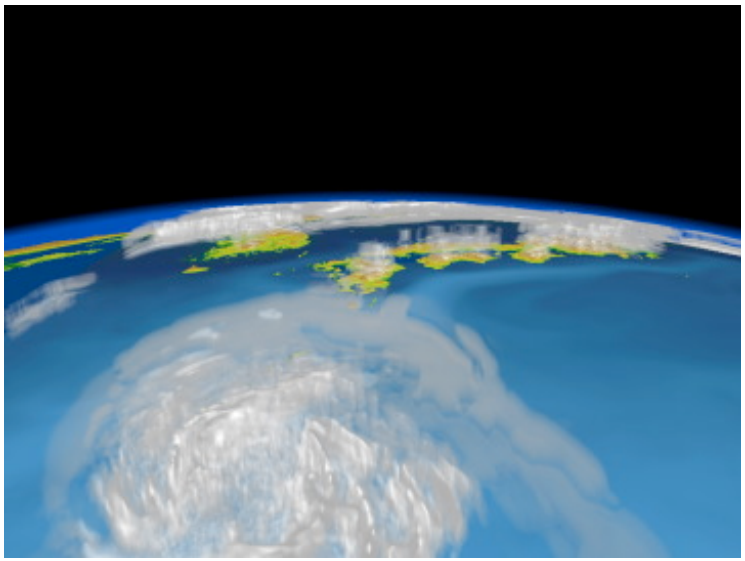
- Yin-Yang grid version





Samples of spherical MovieMakers

- MovieMaker on Spherical grid.
- Volume rendering with topography.



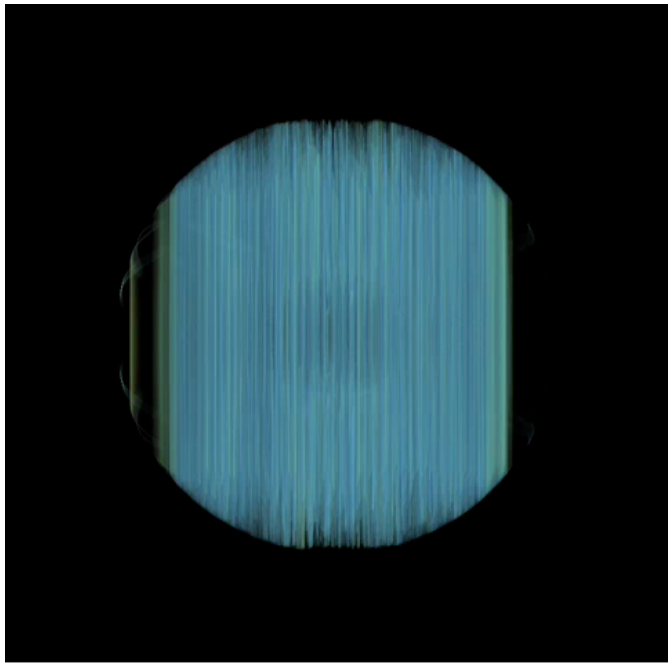
Atmosphere & Ocean
coupled model

Hurricane Katrina

Samples of spherical MovieMaker

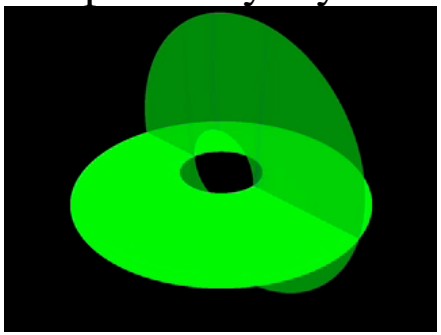
- MovieMaker on Yin-Yang grid

Volume rendering

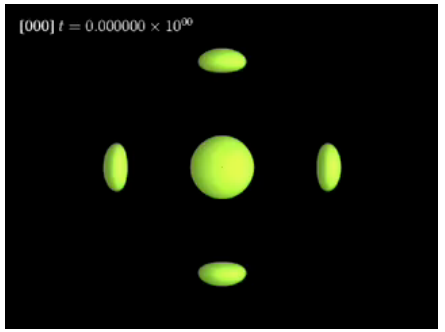


Geodynamo simulation

Slice planes by ray casting



Geodynamo simulation



Mantle convection

Hardware Rendering MovieMaker

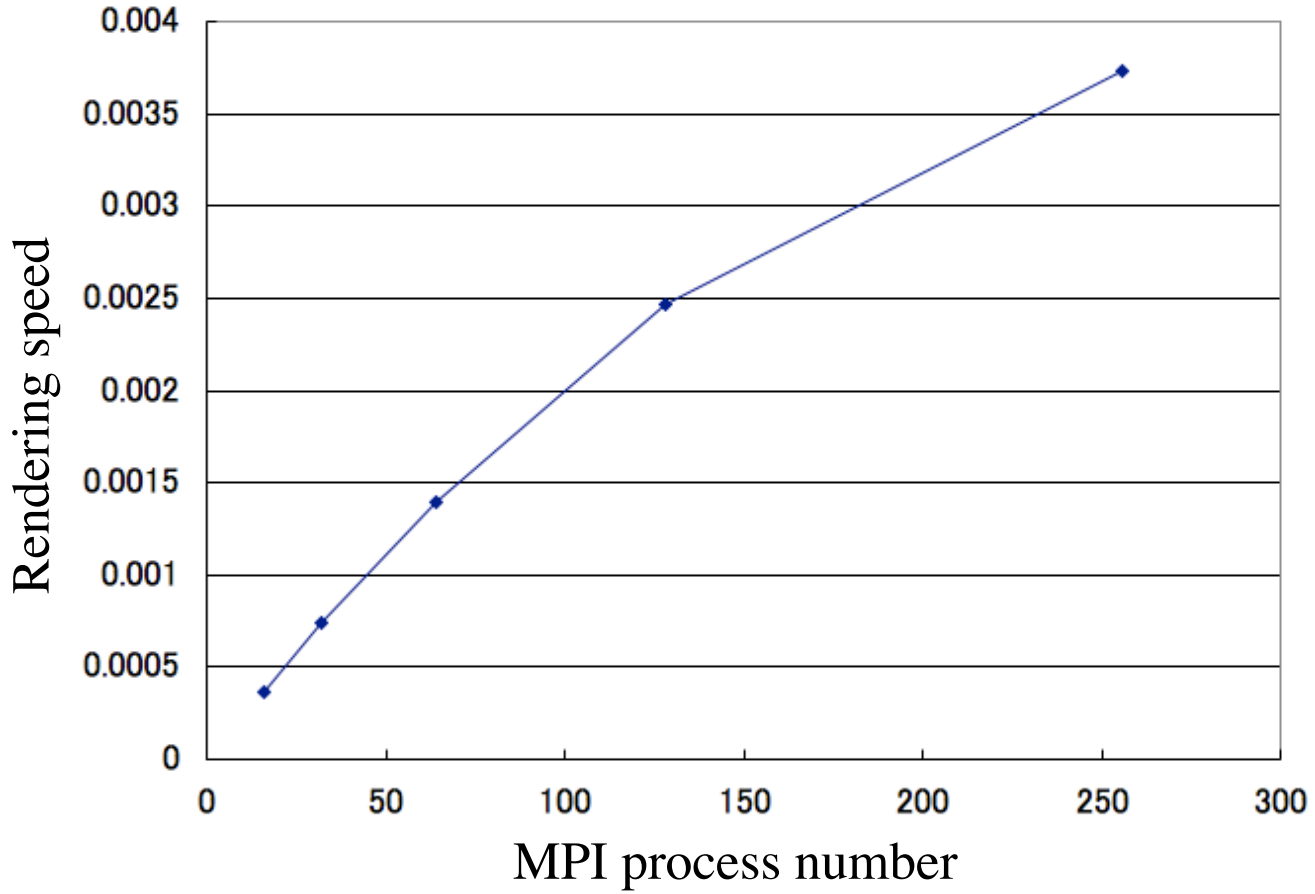
- Volume rendering: Software (ray casting)
- Isosurface: Hardware (OpenGL)
  - Polygon generation by “marching cubes”.
- Stream lines: OpenGL
- Original MovieMaker
  - For GWS (SGI Onyx3800, Onyx4)

Software Rendering MovieMaker

- *Armada*
- Developed by N. Ohno.
- Everything is done by software
  - Isosurface and slice planes by the ray casting.
  - Semi-transparent surfaces.
- Don’t need graphics board any more.
  - Tried on Altix, SX-8R, and Earth Simulator.
- Parallelization scaling is not bad.
  - MPI & OpenMP
- High vectorization ratio!

Performance of *Armada*

- Parallelization on Altix4700



Performance of *Armada*

- Speed test on Onyx, Altix, SX-8R & Earth SImulator

		Scalar (original) time (vec. ratio)	Vector time (vec. ratio)
Onyx3800	Case1	580	-
	Case2	382.7	-
	Case3	320.0	-
Altix4700 (Interactive node)	Case1	399.3	-
	Case2	263.0	-
	Case3	203.6	-
SX-8R	Case1	1400.0 (8.02%)	72.3 (99.23%)
	Case2	888.3 (6.49%)	55.2 (99.07%)
	Case3	761.9 (4.69%)	46.7 (99.17%)
Earth Simulator	Case1	-	164.8 (99.25%)
	Case2	-	125.9 (99.09%)
	Case3	-	112.3 (99.20%)

Speed up by the vectorization

Vector machines are faster than Altix!

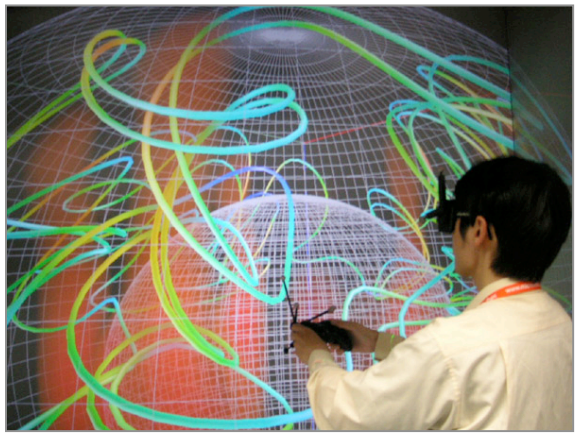
Volume rendering and isosurface on Yin-Yang grid.

From our Experience on Parallel Vis.

- We took a DIY-approach and it was effective.
  - We wrote MovieMaker from scratch.
  - Spent a lot of time to code, but we got flexibility:
    - Easy to accept requests from simulation researchers.
    - Easy to implement new method.
    - Easy to adopt to different data coordinates.
    - Easy to adopt to different computer architecture.
- Vector machines are fast for visualization, too.

Outline

1. Introduction
2. Our projects
   - MovieMaker
   - VFIVE
   - others
3. Future

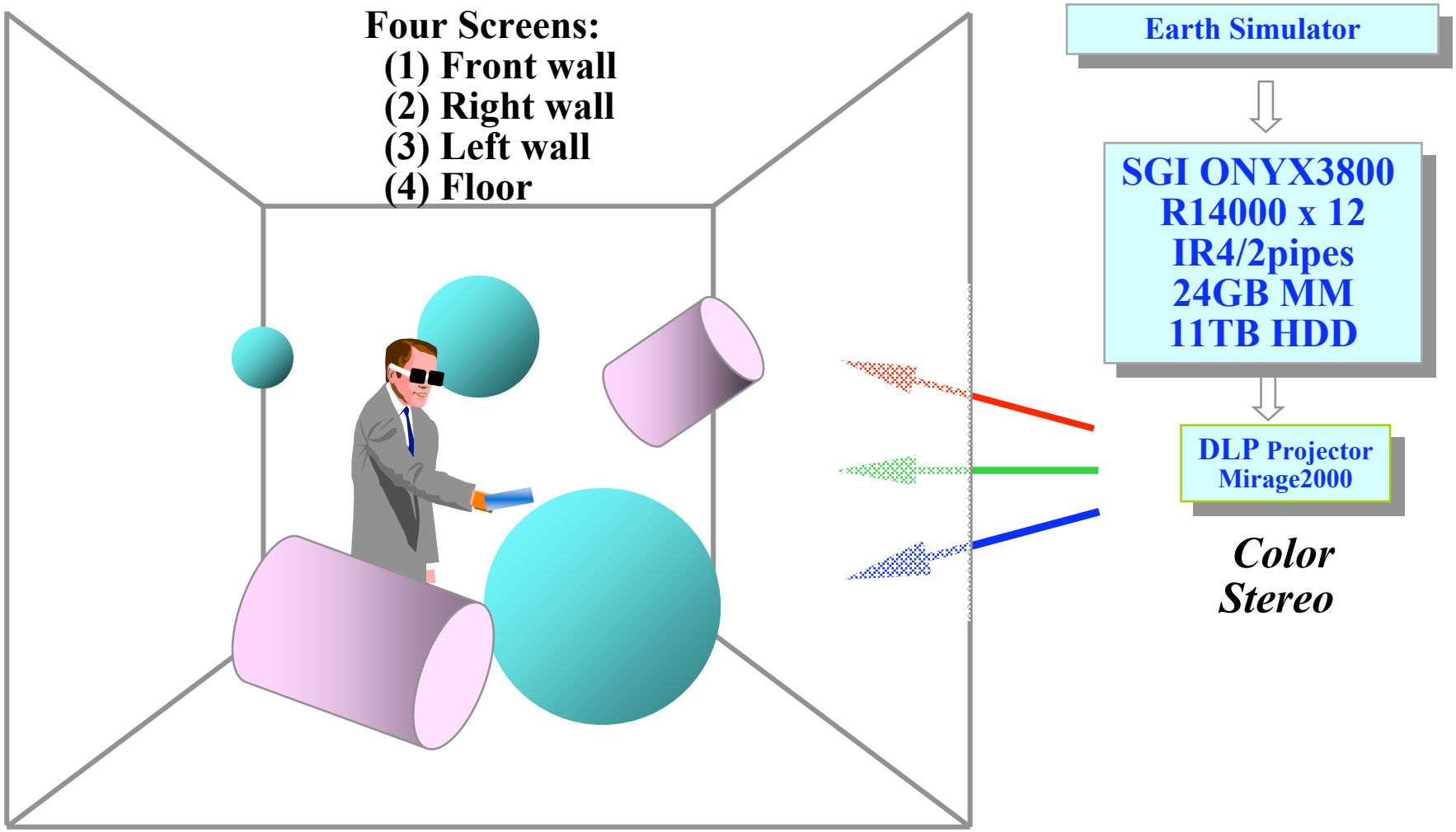


VFIVE

- A Virtual Reality (VR) visualization software
- For CAVE systems
- Originally developed by A. Kageyama
  - Since 1997
- Now developed by N. Ohno

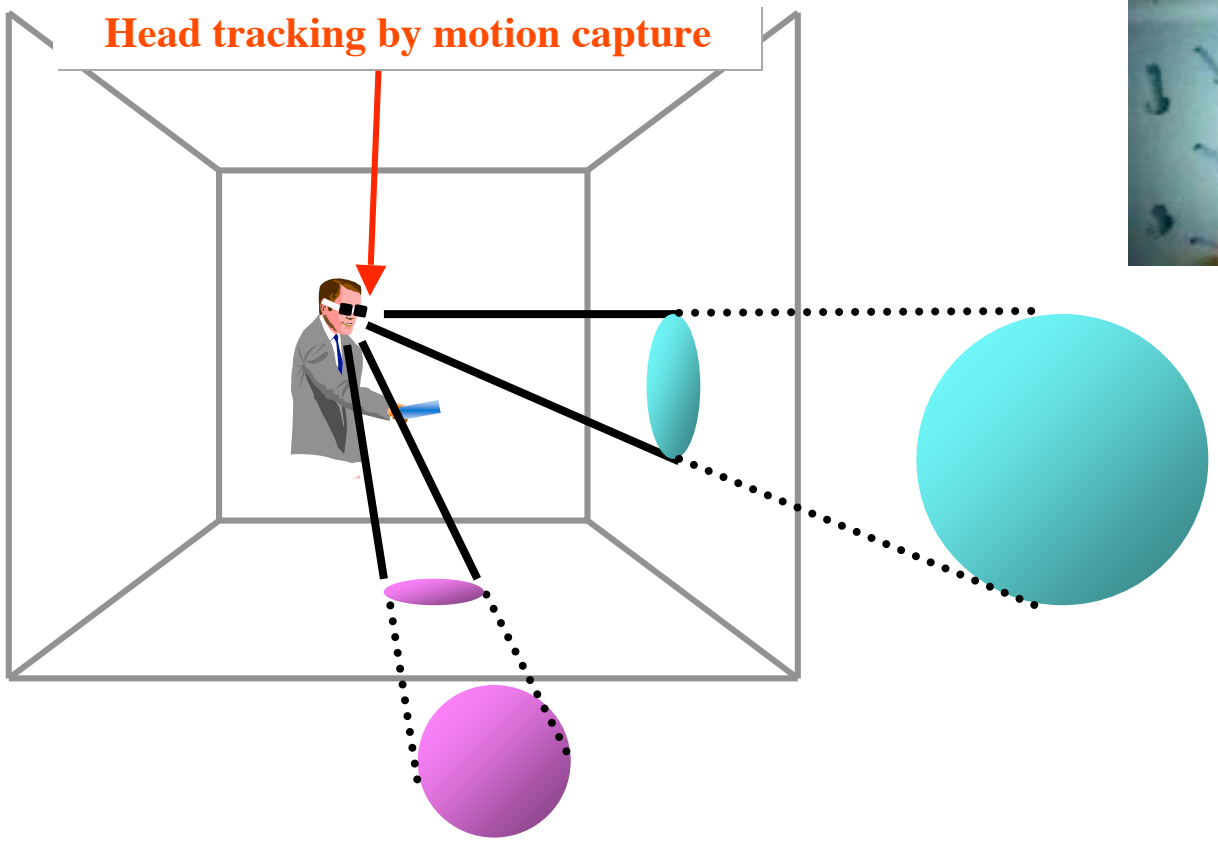
Akira Kageyama, et al.,
Visualization of Vector Field by Virtual Reality,
Prog. Theor. Phys. Suppl., 138 (2000)

A CAVE at Earth Simulator Center

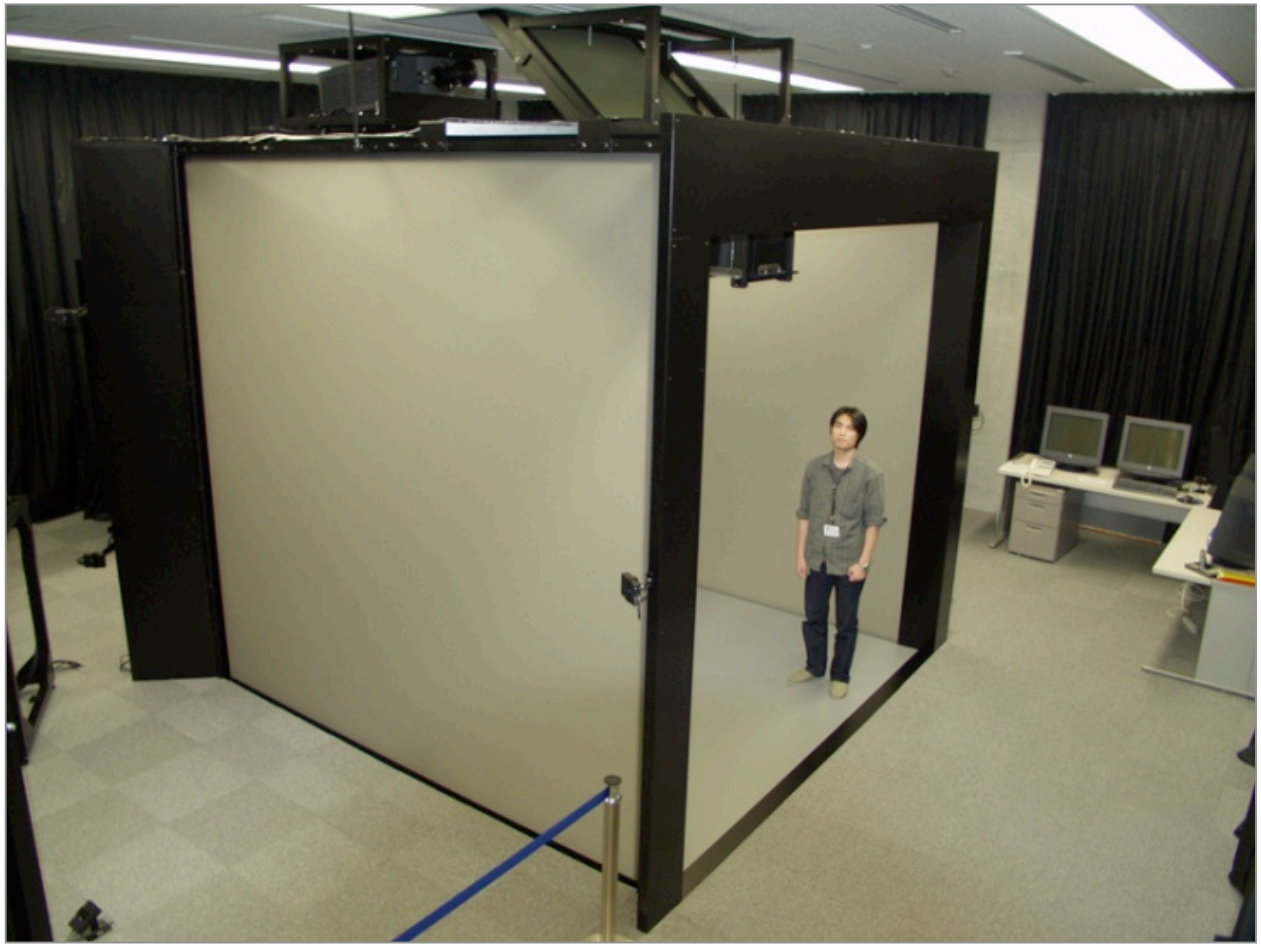


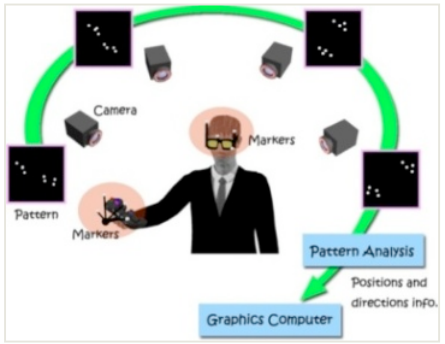
Head Tracking of CAVE System

You can walk in the CAVE room and observe 3D objects from any point.



Overview of our CAVE



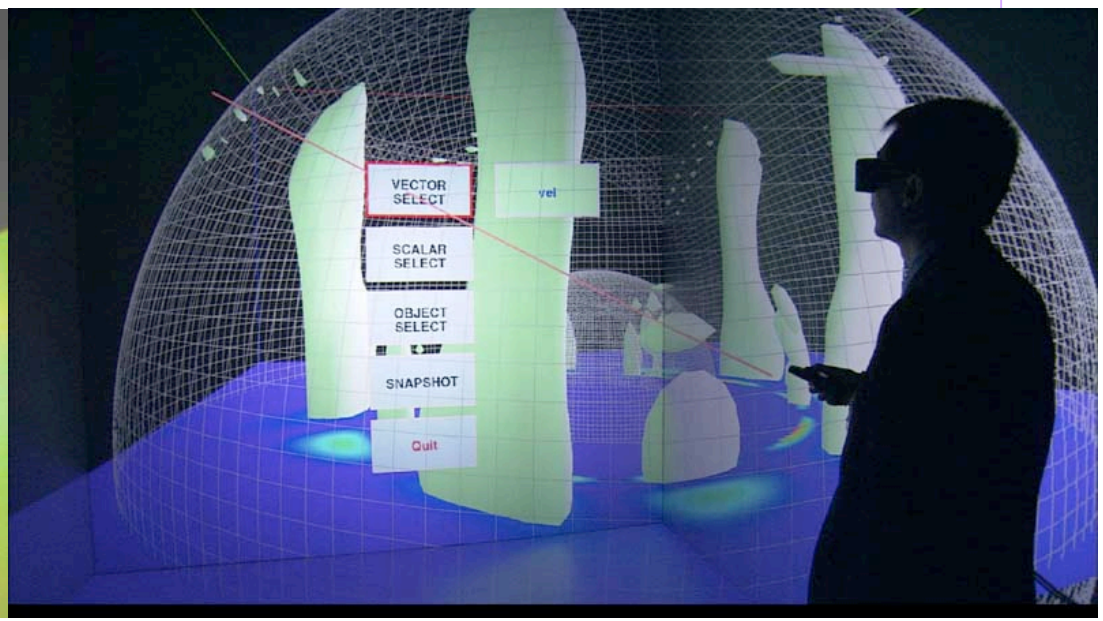


CAVE as a Visualization Tool

- CAVE is “old” and there are many in the world. But…
- Most of them seems to be used as just (an expensive) 3-D TV set!
- VR is not a 3-D TV; not just for an immersive & stereo-view.
- *Interactivity* is the key! Let’s use this function!

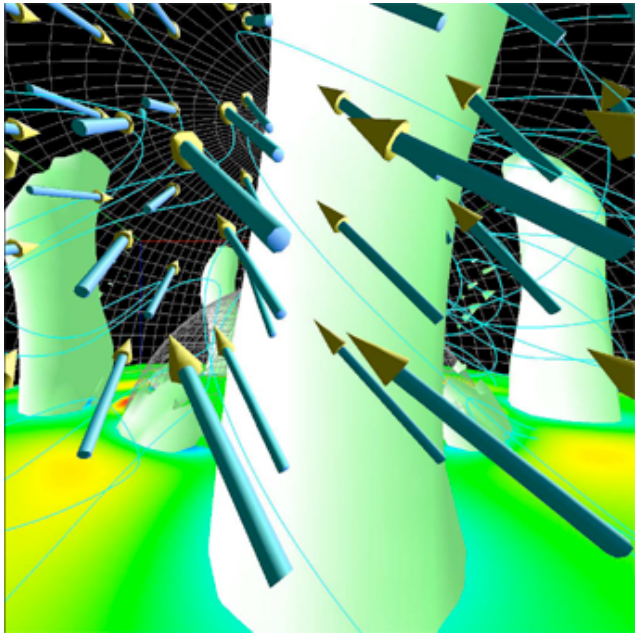
VFIVE

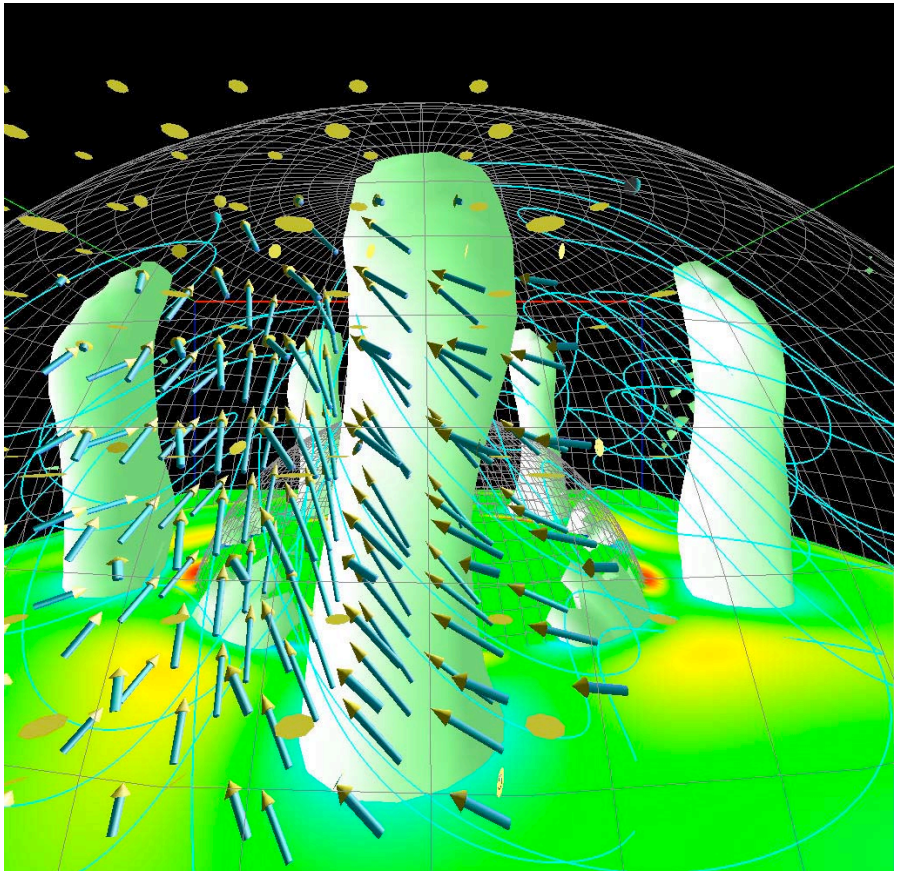
- General purpose VR visualization software
- C++ with OpenGL & CAVE lib.
- Runs on CAVEs, ImmersaDesk, & HMD.
- (Virtual) menu panel with (virtual) laser beam



VFIVE Visualization Methods

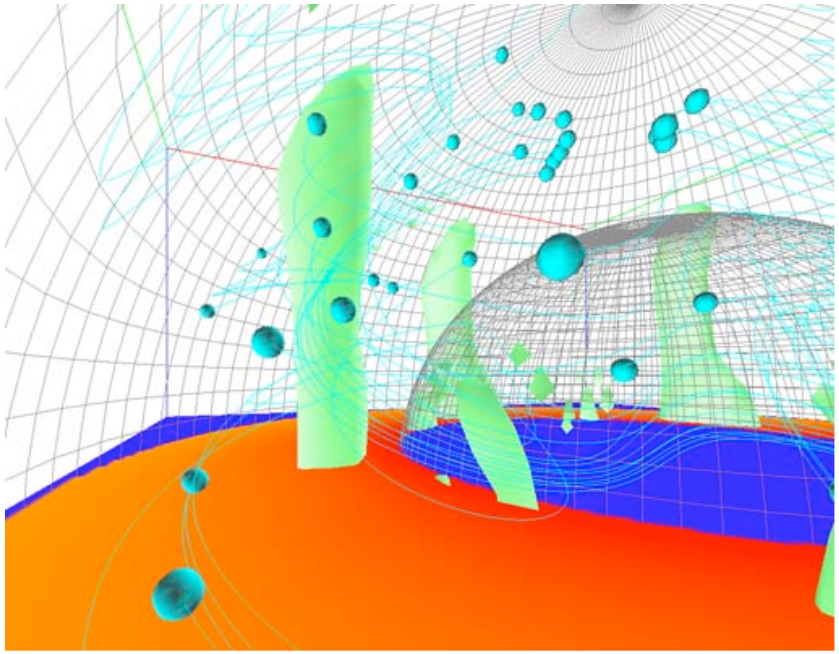
- "Local Arrows"
  - Follows your hand
- "Isosurface"

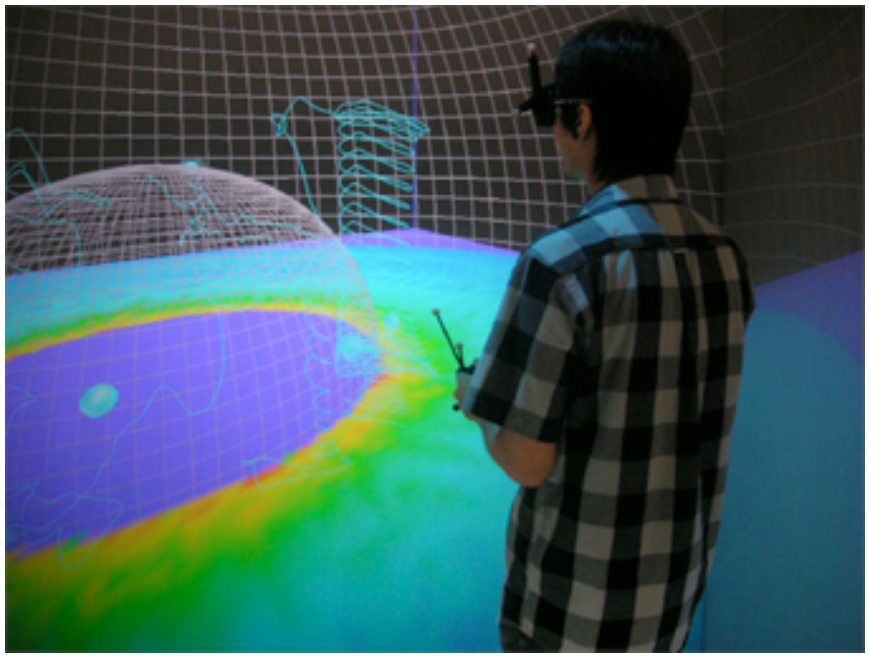




VFIVE Visualization Methods

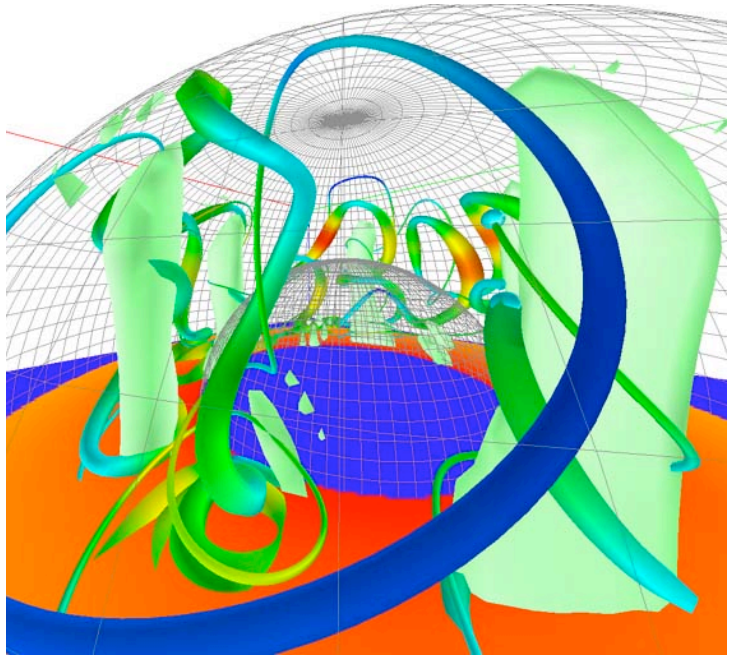
- "Tracer Particles" and "Stream (or Field) Lines"
  - Starts from your finger tip.





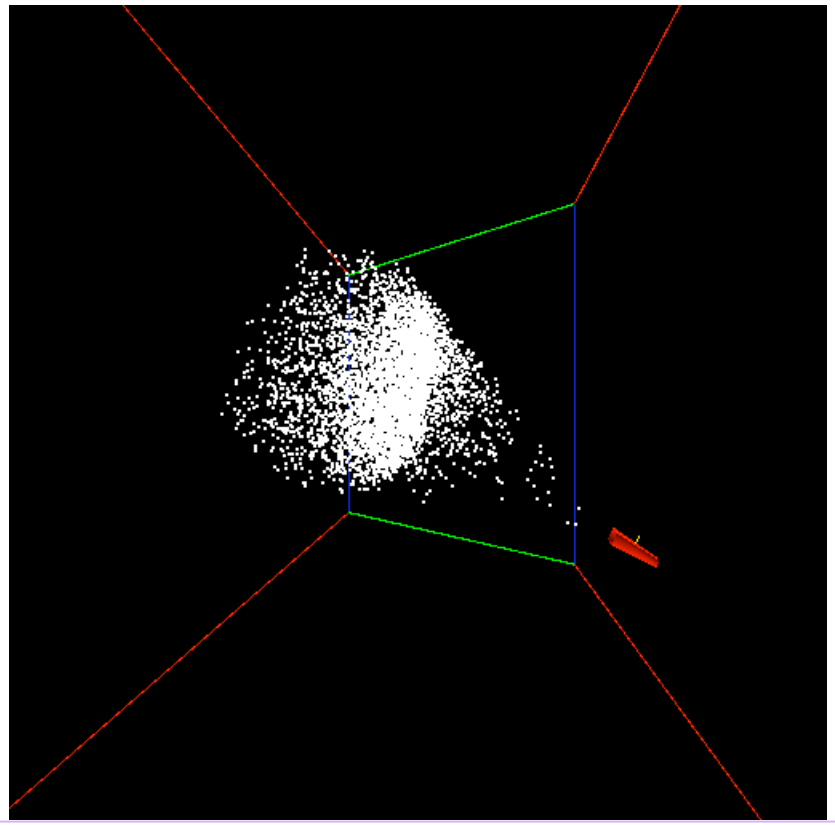
VFIVE Visualization Methods

- "Stream Line Tubes"
  - Again, starts from your finger tip.



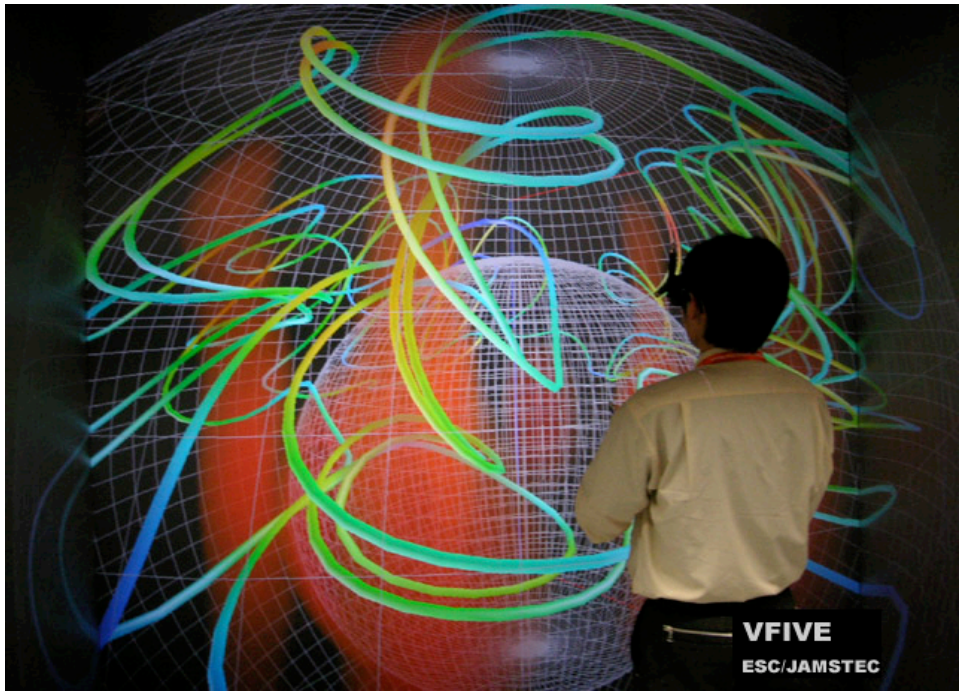
VFIVE Visualization Methods

- "Snowflakes in spotlight"
  - Many particles in virtual spotlight from your hand.



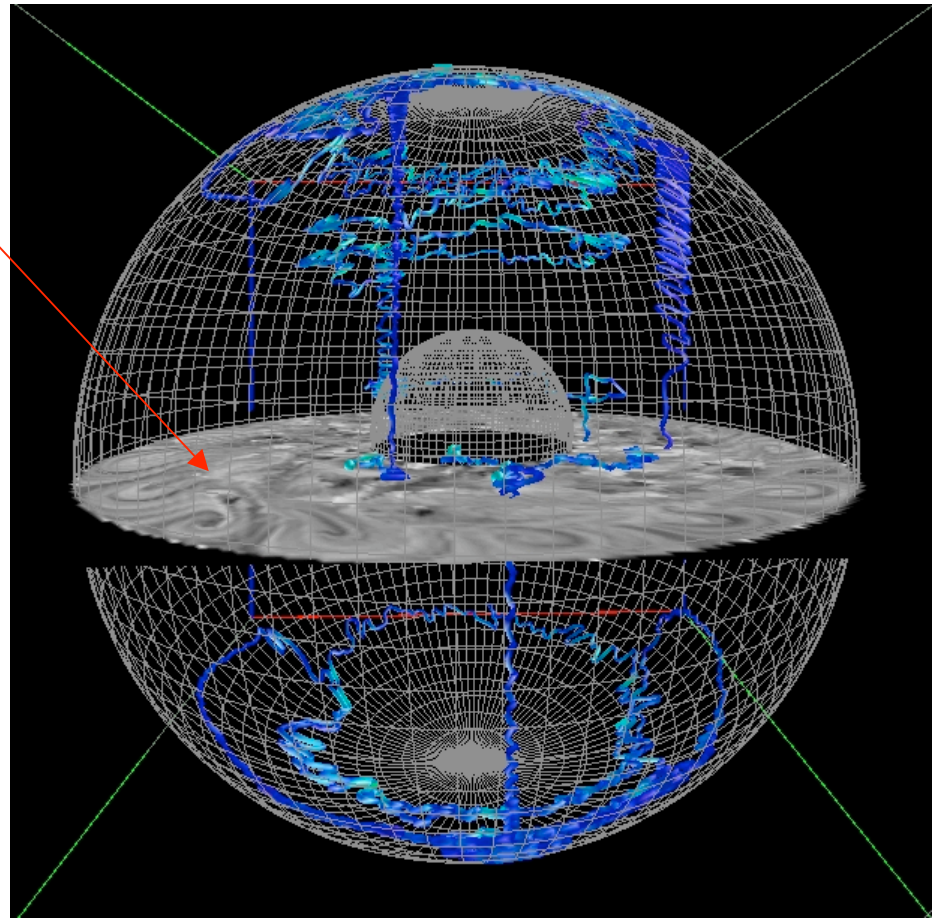
VFIVE Visualization Methods

- "Volume Rendering"
  - 3D texture map technique



VFIVE Visualization Methods

- “LIC”

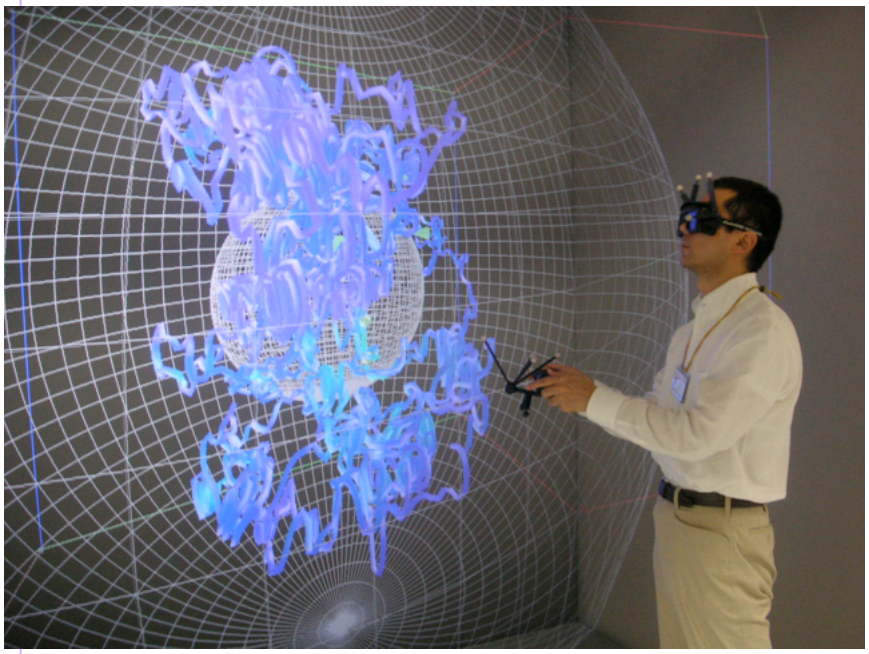


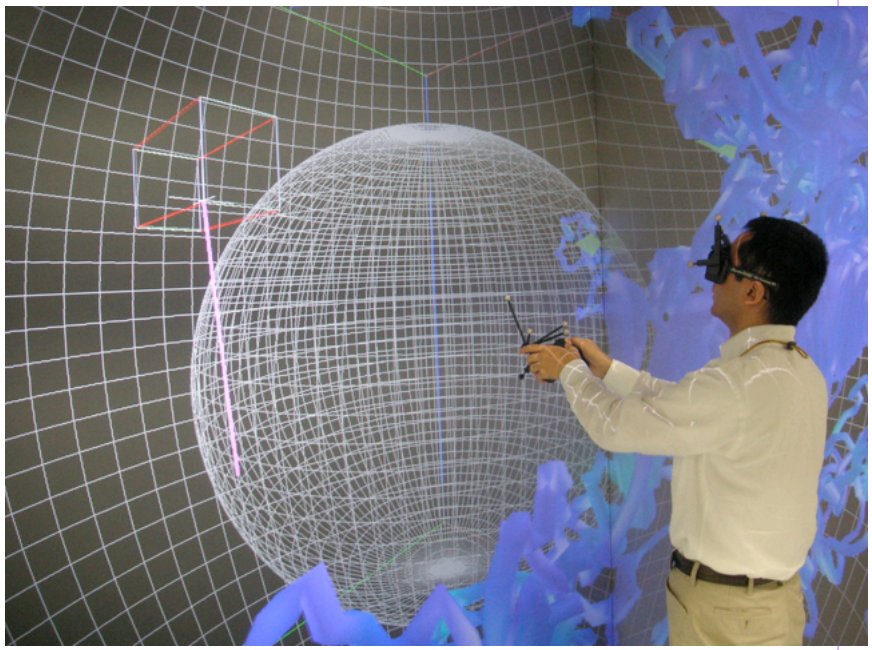
VFIVE

- movie

Recent Improvement of VFIVE

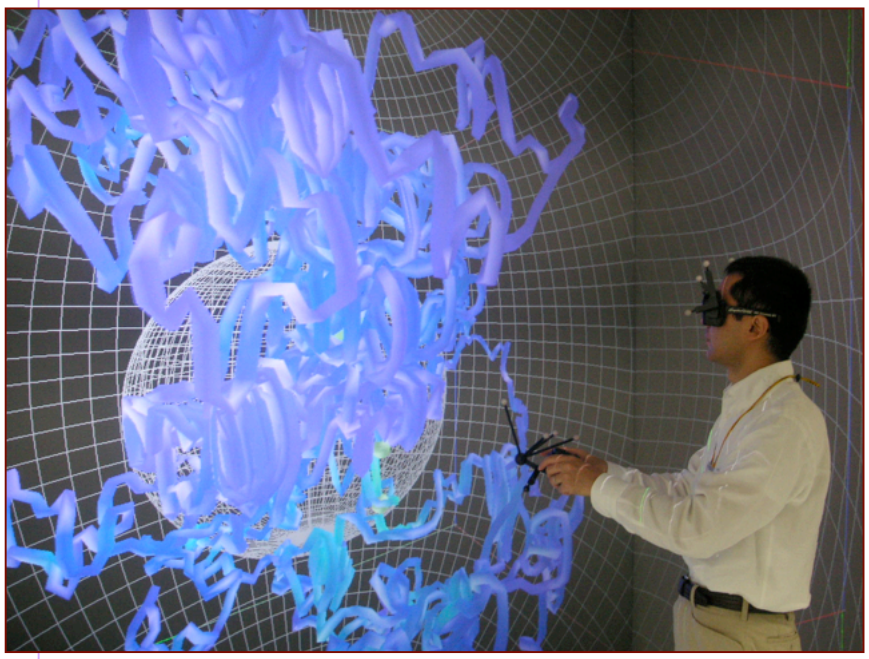
- ROI (Region of Interest) function

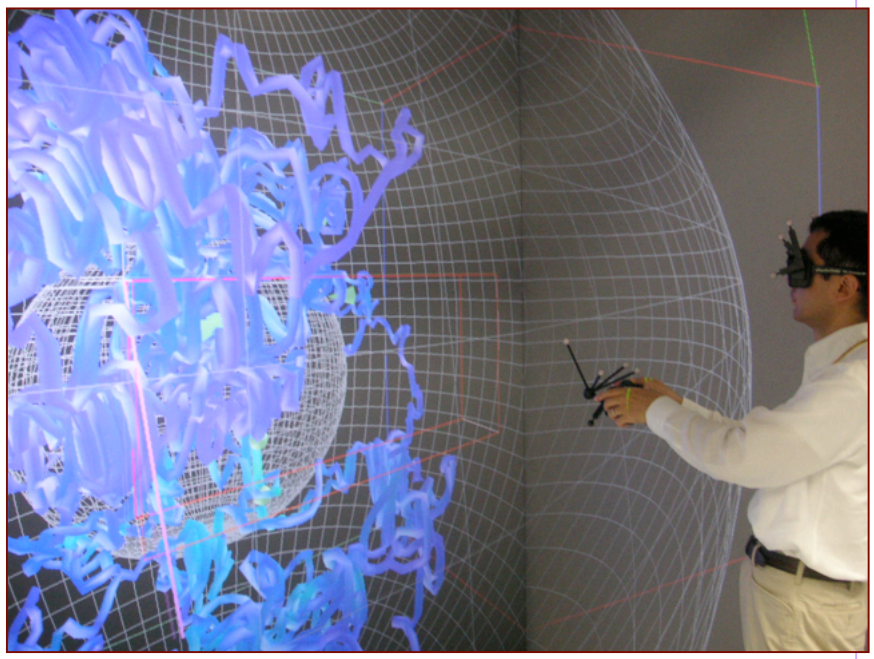




Recent Improvement of VFIVE

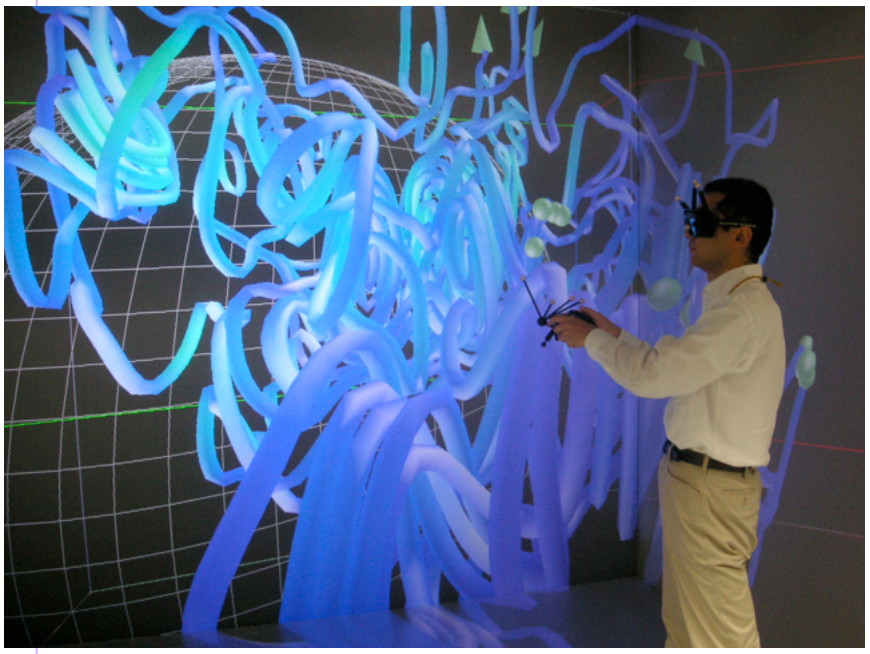
- ROI (Region of Interest) function

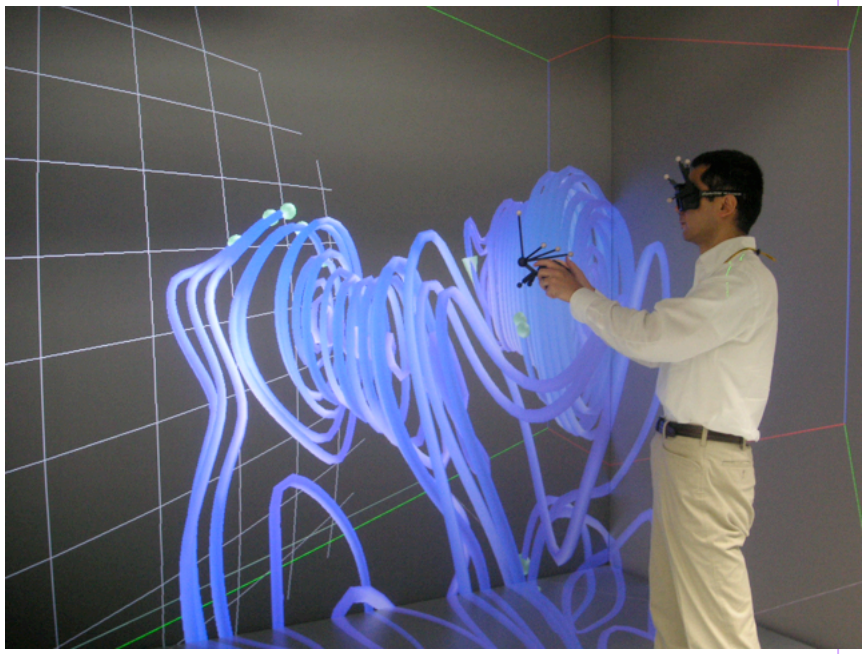




Recent Improvement of VFIVE

- ROI (Region of Interest) function





Summary of VFIVE

- Rich visualization methods
- Interactive visualization
- Parallelization by OpenMP
- Can read 640^3 grid data
- We are analyzing 17GB data (geodynamo)
  - 1GB x 17 variables

From our Experience on VFIVE

- VR visualization by CAVE is really useful!
  - Some people say opposite. But we disagree.
  - It is especially useful when you have:
    - Data of large scale 3D
    - Data with vector fields
    - Data with complex spatial structure
- We've found that ROI is very effective.
  - Without ROI, you need to decrease the grid size. (We cannot show 2000^3 data in CAVE today.)

Open Souce VFIVE

- VFIVE code will be an open source, very soon.
- Please try VFIVE and enjoy interactive VR visualization in your CAVE.

Outline

1. Introduction
2. Our projects
   - MovieMaker
   - VFIVE
   - others
3. Future

CUI / GUI Interface to MovieMaker

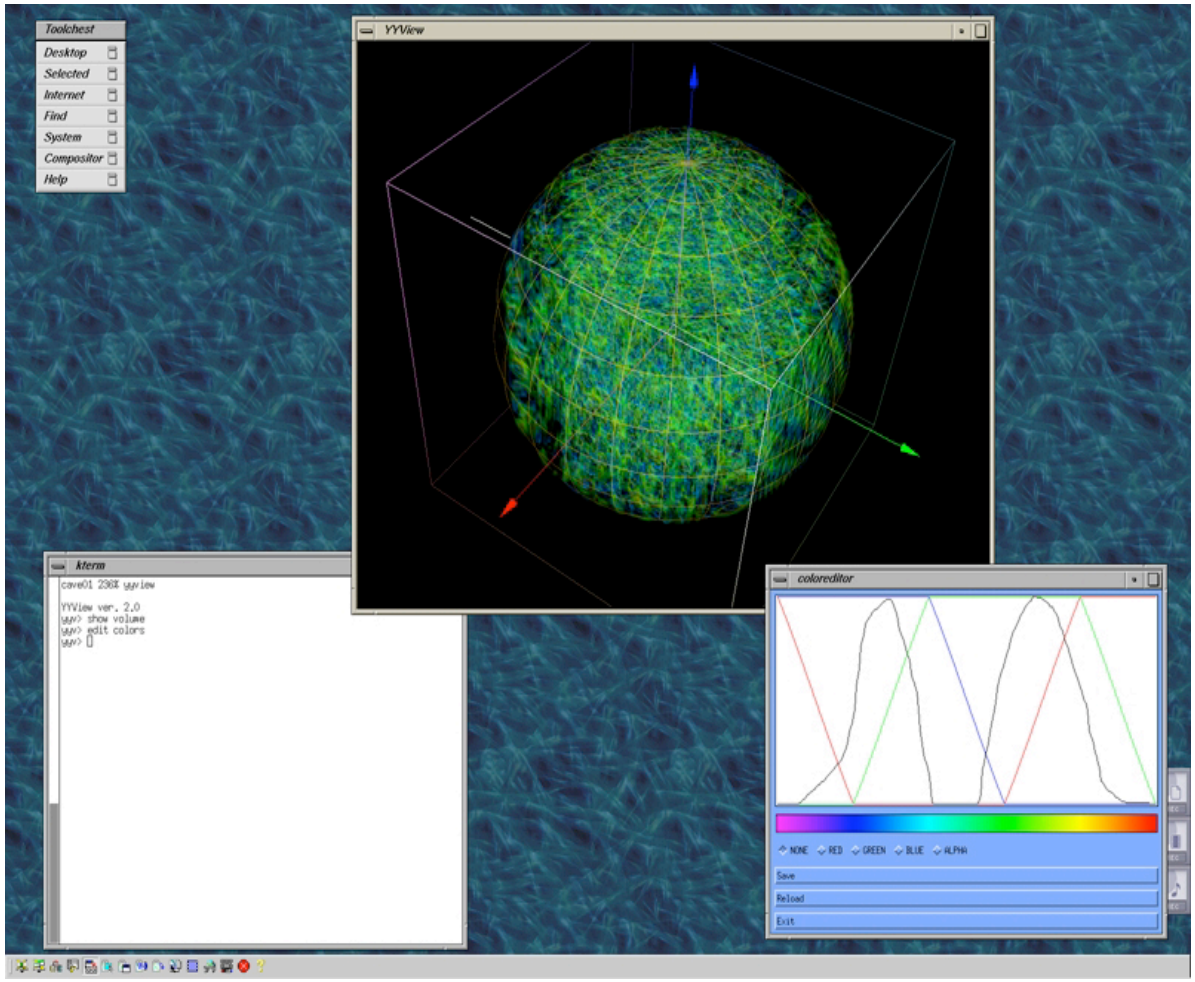
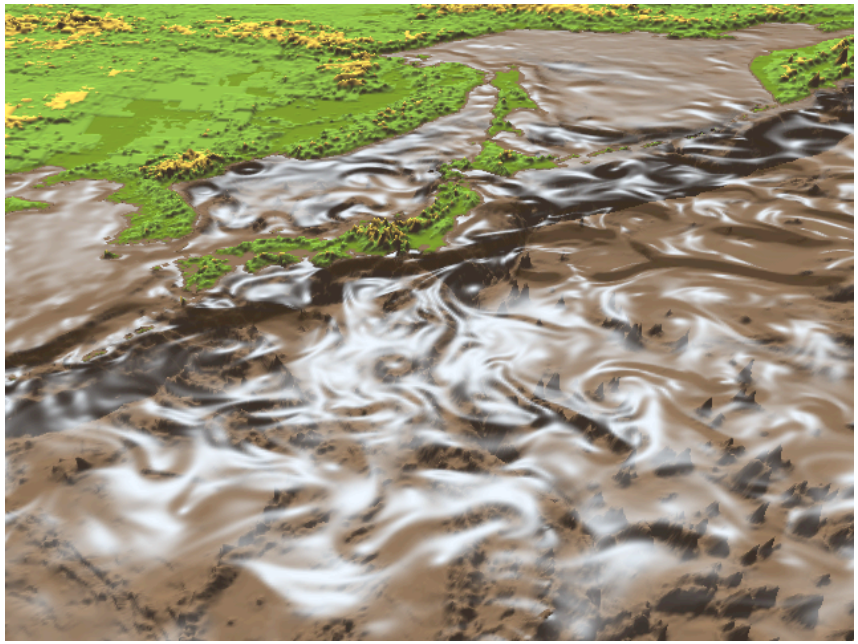
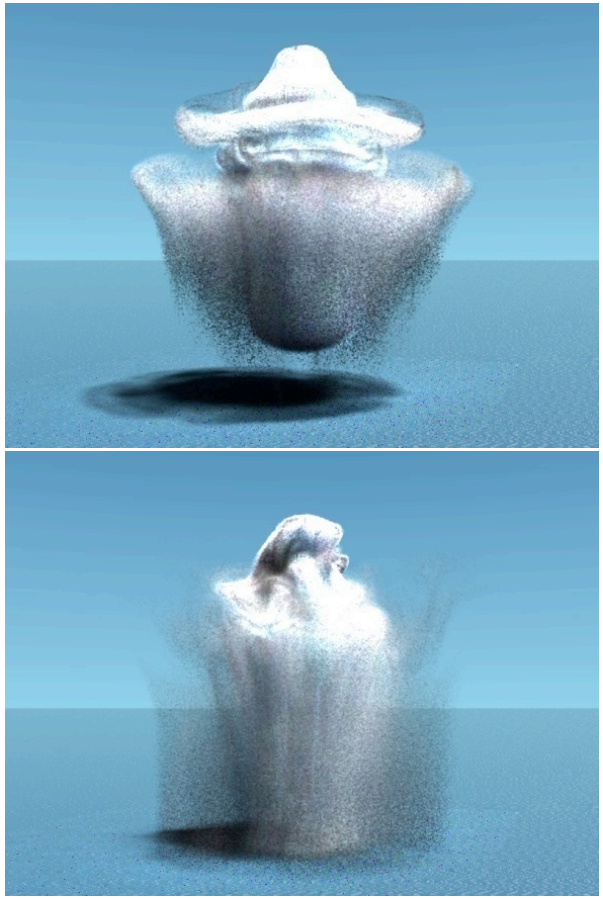


Photo-realistic presentations

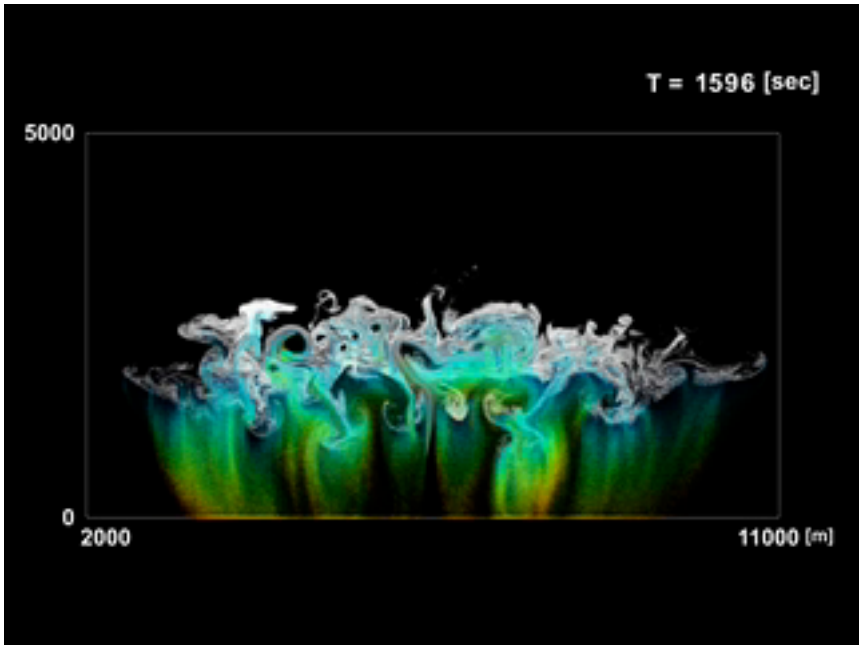




Visualized by F. Araki

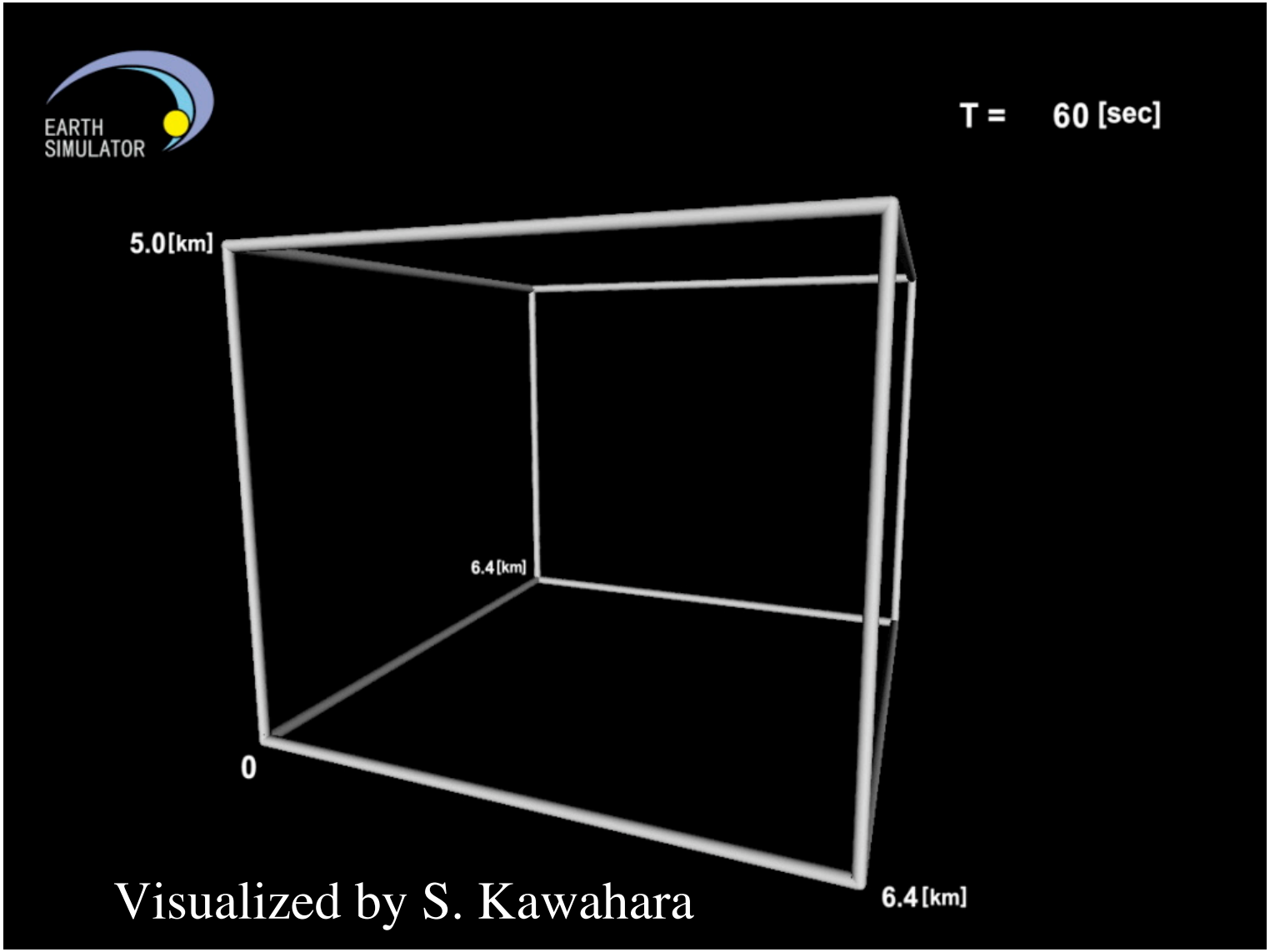
“Pointillism”

- Cloud formation simulation by “super-droplet” method
- Visualization by pointillism with OpenGL point.



Visualized by S. Kawahara

3D Pointillism by OpenGL Point Sprite



Outline

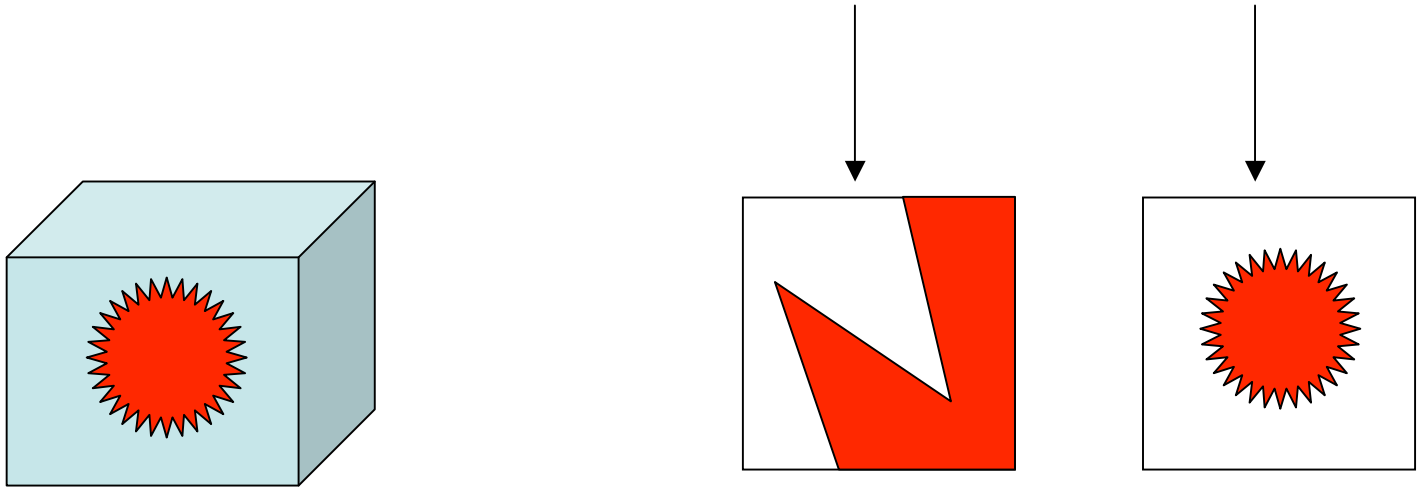
1. Introduction
2. Our projects
   - MovieMaker
   - VFIVE
   - others
3. Future

Future

- End of post processing
  - Size of 3-D data for post processing
  - Transfer time of the 3-D data
- 8000^3 mesh simulation, 1000 snapshots of 3D data
  - 4Byte*(8000^3)*1000 = 2 PB

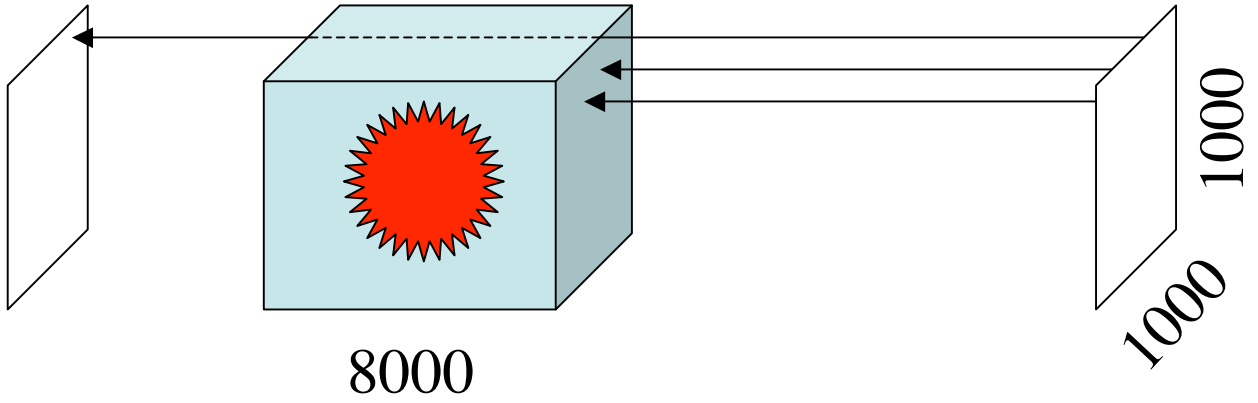
Overtaking of Resolution

- Resolution of visualization image 1000x1000 pixels.
- Resolution of simulation mesh 8000^3
- Visualization will be partial or reduced.



Overtaking of Computational Cost

Simulation: 8000 x 8000 x 8000
Ray casting: 1000 x 1000 x 8000 (isosurface, volume rendering, slices)



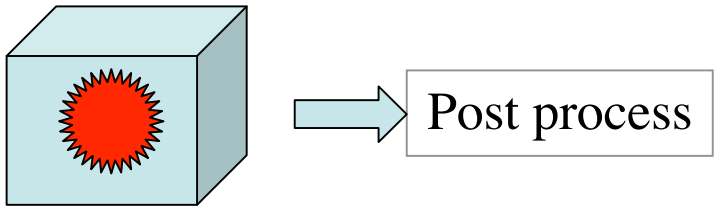
For each simulation time step,

Cost for a visualization image << Cost for simulation

In-situ Movie Making with Thousands of Cameras

With different view points, different variables, different color maps…

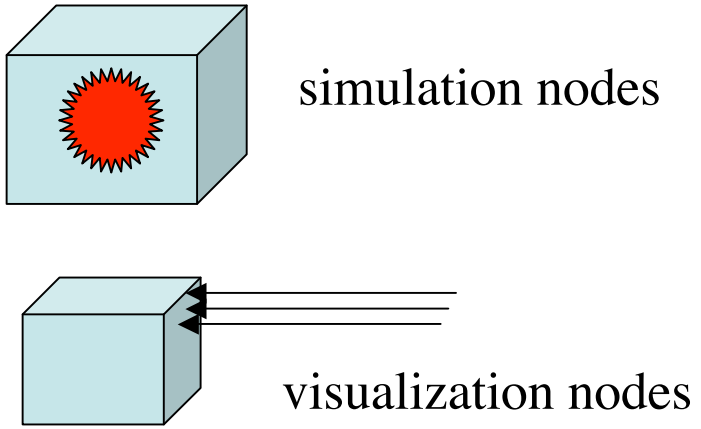
In-situ Movie Making with Thousands of Cameras



Output: 3D numerical data

8000^3 mesh x 1000 snapshots

= 4Byte*(8000^3)*1000 = 2 PB



Output: Thousands of movies

- a still image: 1000x1000 pixels
- a movie: 10000 frames
- thousand movies

= 4Byte*(1000^2)*10000 = 40 TB

Summary

- Data analysis in experimental style
  - Repeat the “experiment”, if necessary.
  - But thousands of movies will be enough…
- Information mining from thousands of movies