Applying and Extending Scientific Visualization Algorithms for Astrophysical and CFD Data using the Vish Environment
Fernando E. Berrios Cádiz, Edwin Mathews2, Dr. Werner Benger2

1. University of Puerto Rico, Humacao
2. Louisiana State University

Abstract

In the world of computational science, the ability to visualize computational results becomes invaluable as the scale of the data grows to sizes that are unable to be interpreted by observing numerical outputs. These visualizations can provide insights into the results that are otherwise lost in large amounts of data. On this poster, we describe the methods we used to effectively visualize large astrophysical datasets and the development of new modules for the analysis and visualization of computational fluid dynamics datasets in the Vish visualization environment. Astrophysical images and movies were created in the hope of producing effective visualizations that can be appreciated by researchers and casual observers. CFD modules were developed specifically for data that is using a triangular surface mesh to represent a virtual "bubble". All programming was developed specifically for Vish and done using C++.

Astrophysical Data Visualization

These visualizations of astrophysical data computations model natural phenomena and provide a visual method for researchers to understand them. Here are two examples of datasets: formation of the cosmos and part of its evolution and colliding galaxies. The study of these phenomena is easier and clearer when we can observe the details of every time step of their evolution. The Vish environment is useful in the visualization of large datasets because it can quickly render and load data with a high dynamic range. For example, the evolution of the cosmos is 284.7 GB large and has 16 million data points and the colliding galaxies is 17 GB large with 1.6 million data points. Each has multiple fields of information that can be extracted using Vish.

Data File

Visualization Method

- Vish can be used to visualize Vertex, Scalar, Vector, and Tensor fields with seven rendering methods
- Images of rendered data can be taken in Vish – can capture single images or a series of images over a certain amount of time steps. Vish also has image post-processing capabilities built into its recorder.
- Series of images can be converted to movies using ffmpeg development software. Some movies we created are available on Youtube - search for VISHVisualization
- For the Cosmos and Colliding Galaxies, Velocity Vectors were extracted from the grid of particles and was rendered using the Vish module Vector Speckles. Vector speckles are useful because they are rendered quickly even in very large visualizations and because they have a translucent quality, areas dense in particle vectors appear to be glowing. They can also be colored according to their position from the point of the viewer. More information on Vector Speckles is available in [1].

Visualization Examples

Cosmos

Colliding Galaxies

Acknowledgements

We would like to thank our collaborators and the University of Puerto Rico for their support.

Dr. Werner Benger
Center for Computational and Technology Visualization Services Center (ITS)
National Science Foundation

Future Work

The next step for our visualization work would be to refine our current modules and to continue developing helpful methods for CFD data visualization and analysis:
- The curvature tensor code is still in the development phase and needs to be verified for accuracy. It also has points on triangular grids where the eigenvectors are displaying incorrect values and needs to be addressed
- Other than the corrected, Gaussian Curvature can be visualized by implementing a simple formula
- Convolve the total volume surrounded by the surface
- Develop a module in Vish that uses Line Integral Convolution to display vector fields on a surface and look into developing a similar method for tensor fields. This also can be used to visualize the principle directions from the curvature tensor directly on to the 2-D surface
- Work with visualization group on creating a module for streak surfaces in Vish

References & Datasets


Datasets visualized in this poster were created by:
Dr. Wolfgang Kapferer, University of Innsbruck, Austria
Dr. Sumanta Acharya, Louisiana State University

Method

To compute both the total surface area and the area of triangles around each vertex we used a simple algorithm that first found the area of each individual triangle. These were summed up for all triangles to find the total surface area. For the area around each vertex, the triangular area was added to a scalar array for each vertex that was a member of the triangle. It was then visualized using a color scale based on the value at each vertex. To calculate the curvature tensor, a method described in [3] was used to approximate it at each vertex:

$K_{ij} = \frac{1}{2} \sum_{j} \left( \frac{w_{ij}N_{i} \cdot \hat{T}_{j}}{w_{ij}N_{i} \cdot \hat{T}_{j}} \right) - \frac{1}{2} \sum_{j} \left( \frac{w_{ij}N_{j} \cdot \hat{T}_{i}}{w_{ij}N_{j} \cdot \hat{T}_{i}} \right)$

Where $i$ represents the center vertex and $j$ represents the outer surrounding vertices. $N_i$ is the unit normal vector at each vertex, $w_i$ is the curvature in the $ij$ direction, $\hat{T}_i$ is the unit projected vector of the edge $ij$ on the tangent plane, and $w_i$ is a weighted value dependent on the area. We can visualize this tensor as an ellipsoid on the vertex. The shape and size of the ellipsoids represent the directions and magnitude of the curvature in the principal directions, providing a visual of how much and which way the Gaussian curvature can also be determined for each point using a simple function of the non-zero eigenvalues of the curvature tensor. Curvature is an important characteristic of the CFD surface because it represents the amount that a surface is deviating from a flat surface and is indicative of the turbulence in that area.

Results

- Computed cumulative surface area of all triangles
- Visualized the area of triangular surfaces around each vertex (represents "stretching" or "compressing" of bubble surface at vertices)
- Visualized the tensor of curvature at each vertex

Cone Surface with Vertex Area displayed - blue areas depict large stretching, red areas compression
Cone Surface with eigenvectors displayed as vector speckles on each vertex – vectors are in the direction of the principle curvatures and the area
Cone Surface with Tensor Patterns displayed – Note: Tensor Patterns is a 3D visualization method and is used on a 2-D surface here. See Future Work for next steps for 2-D tensor visualization