Visualizing Katrina - Merging Computer Simulations with Observations

Werner Benger, Shalini Venkataraman, Amanda Long, Gabrielle Allen,
Stephen David Beck, Maciej Brodowicz, Jon MacLaren, and Ed Seidel

Center for Computation and Technology, Louisiana State University, Baton Rouge
LA 70803, USA,
`werner@cct.lsu.edu`,
`http://sciviz.cct.lsu.edu/katrina/`

Abstract. Hurricane Katrina has had a devastating impact on the US Gulf Coast, and her effects will be felt for many years. Forecasts of such events, coupled with timely response, can greatly reduce casualties and save billions of dollars. We show how visualizations from storm surge and atmospheric simulations, were used to understand the predictions of how strong, where, and when flooding would occur in the hours leading up to Katrina's landfall. Sophisticated surface, flow and volume visualization techniques show these simulation results interleaved with actual observations, including satellite cloud images, GIS aerial maps and LIDAR showing the 3D terrain of New Orleans. The sheer size and complexity of the data in this application also motivated research in efficient data access mechanisms and rendering algorithms. Our goals were to use the resulting animation as a vehicle for raising awareness in the general populace to the true impact of the event, to create a scientifically accurate representation of the storm and its effects, and to develop a workflow to create similar visualizations for future and simulated hurricanes. Screenings of the animation have been well received, both by the general public and by scientists in the field.

Abstract. Data sets stemming from different sources are frequently incompatible due to domain-specific data layouts and data formats. HPC applications usually provide their own proprietary or domain-specific output mechanisms, and problems often occur not before the visualization stage when multiple datasets are combined into one result. In this article we present a novel approach to combine different data sets describing Hurricane Katrina and its devastating impact on the US Gulf Coast. Forecasts of such events, coupled with timely response, can greatly reduce casualties and save billions of dollars. We show how to create visualizations from storm surge and atmospheric simulations that allow to depict how strong, where, and when flooding would occur in the hours leading up to Katrina's landfall. Sophisticated surface, flow and volume visualization techniques show these simulation results interleaved with actual observations, including satellite cloud images, GIS aerial maps and LIDAR showing the 3D terrain of New Orleans. The sheer size and complexity of the data in this application scenario motivated research in efficient data access mechanisms and rendering algorithms. The resulting animations are suited as a vehicle for raising awareness in the general populace to the true impact of the event. The newly developed technology allows to create a scientifically accurate representation of the storm and its effects, ultimately providing a workflow to create similar visualizations for future and simulated hurricanes. Screenings of the animation have been well received, both by the general public and by scientists in the field.

1 Motivation

The catastrophe of Hurricane Katrina has not only highlighted the need for timely and accurate measurements from instruments and forecasts from numerical simulations but also for meaningful visualizations that draw upon these diverse data sources. In this paper, we highlight one such effort to visualize the events leading to the flooding cased by Hurricane Katrina pulling together models of the hurricane's wind, temperature and pressure fields, the storm surge, 3D terrain views from LIDAR and GIS data, combined with comparisons to what actually happened using time-varying atmospheric imagery from the GOES-12 weather satellite and the actual hurricane tracks. The Center for Computation & Technology at the Louisiana State University (LSU) is a partner in the SURA Coastal Ocean observing and Prediction Program (SCOOP) [1], a interdisciplinary community engaging in distributed coastal modeling across the southeastern US with the goal of building an integrated virtual laboratory for coastal research. Advisories from the National Hurricane Center(NHC) about impending storms automatically trigger automated workflows that use different wind fields to initiate coastal models such as the ADvanced CIRCulation hydrodynamic model[1] (ADCIRC) that require significant parallel computing resources, available at CCT through the 1024 processor cluster *SuperMike*. The wind-fields are generated by the MM5[2] atmospheric model. The SCOOP data archive [2] developed and deployed at LSU aggregates the model outputs from multiple sources across the nation and is the source of data for our visualization efforts.

2 Previous Work

The geoscience community has concentrated on visualizing, generally in 2D, data from remote sensing and GIS mapping sources. In contrast, within the atmospheric sciences, much work is confined to scientific visualization methods such as isosurfaces or volume rendering for 3D atmospheric model outputs. Nativi et al. [3] clearly highlighted the differences between the data models in the GIS and atmospheric sciences. While GIS is concerned with 2D georeferenced spatial data in multiple layers, atmospheric science deals with hyperspatial (3D, 4D and

---

[1] `http://www.nd.edu/~adcirc/`

[2] `http://www.mmm.ucar.edu/mm5/`

beyond) where geo-referencing is not critical. Moreover, the temporal scales for GIS is orders of magnitude more than in the atmospheric sciences (years vs. minutes).

Examples of atmospheric visualization include work from National Oceanic and Atmospheric Administration (NOAA) Geophysical Fluid Dynamic Laboratory (GFDL)[3], and more recently that of Hurricane Isabel motivated from the IEEE Visualization 2004 contest. The above efforts deal with effectively rendering [4] or data mining and feature extraction [5] of time-varying multdimensional scalar and vector fields and so, do not incorporate other data like GIS or storm-surge models. Recent work in storm-surge and GIS visualization from Zhang et al[6] focuses on geo-information processing such as extracting buildings from LIDAR and interactive animation of flooding. The NOAA Satellite and Information service also have some height-field visualizations of weather satellite data including GOES-12 [4]. What makes our contribution unique is the integrated visualization of all the above diverse data sources.

3 Data Management

3.1 Data Sources

ADCIRC accurately models a wind-driven storm surge - its formation, movement across the ocean and morphology as it impacts land. The underlying computational mesh is built upon topographic or bathymetric information given on an adaptively refined unstructured grid ranging from the atlantic ocean into the canals of New Orleans, where the physical resolution approaches just 100m. The numerical simulation outputs water elevation, given as a scalar quantity on each surface vertex, plus wind and water flow directions, provided as a 2D vector field, on each vertex as well. In our dataset, this output responds to a physical time interval of every 30 mins during Aug 15^{th} until Sep 1^{st}, 2005, just after Katrina had made landfall.

To show the atmospheric conditions that lead to the hurricane formation and the resulting surge, we make use of wind, pressure and temperature fields from the MM5 atmospheric model simulation covering the same time period. Only the domain-2 from this hierarchical data set was used. Here, each time-step is a structured 3D grid with dimensions 150x140x48 storing the wind velocity, pressure and temperature values, output for every hour of simulation domain.

The satellite imagery is based on observational data captured every 15 min from the GOES-12 satellite. GOES[5] is an acronym for Geostationary Operational Environmental Satellite, they are the American counterparts to the European METEOSAT weather satellites. Here, we focus on the "longwave" infrared channel, $10.7\mu m$ measured at 4km resolution.

The dates for all data sets were available on the same temporal domain. In the MM5 case, the available atmospheric simulation results covered the time span of

---

[3] `http://www.gfdl.noaa.gov/research/weather/hurricane.html`

[4] `http://www.nnvl.noaa.gov/`

[5] `http://www.oso.noaa.gov/goes/`

up to 72 hours starting at midnight each day. Since this yielded multiple datasets corresponding to the same time coordinates, we decided to restrict the analysis to data representing the first 24 hour period of each simulation time frame. While this might result in a slight discontinuity of animations when crossing the date boundary, it also ensures that the input datasets contain only the simulation results based on the most recently acquired meteorological measurements.

A 5m resolution elevation grid of the New Orleans area is provided by a LIDAR[6] data set, obtained from the State of Louisiana. In addition, we used satellite imagery of the terrain from the MODIS and LANDSAT instruments at 500m and 250m resolution respectively.

3.2 Data Management Challenges

Satellite images and GIS data are well representable in common image file formats such as GEOtiff [7]. In contrast, MM5 and ADCIRC data are more complex, and no standard format exists for these kind of data types. A huge number of file formats compete, each with particular features for each application. Mostly file formats are optimized for a certain data type, and consequently become mutually exclusive. For instance, a file format being able to cover MM5 data can not necessarily handle ADCIRC data as well and vice versa. Even for each specific, allegedly simple class of data, such as a triangular surface, there co-exist myriads of file formats. Supporting each of these is a tedious work wasting time of application developers. In a scenario where n various independent applications with no a-priori mutual knowledge need to interact, supporting each other application's file format becomes a major n^2 implementation effort.

Ideally, we would like to use a common file format which covers all cases of types of scientific data and thus achieves maximum synergy effects. To find such a unified description, a common denominator is essential, which, following D. Butler [7, 8], is naturally provided by the language of mathematics for the domain of scientific data. D. Butler proposed to use the mathematical concept of vector and fiber bundles to layout data, a concept which is successfully implemented in the IBM DataExplorer, now available as OpenDX [8]. Within the classification scheme of the fiber bundle data model, MM5 outputs are dynamic scalar and vector data on three-dimensional regular domain, while ADCIRC data are described by a a dynamic scalar field given on a static triangular surface.

3.3 The "F5" Approach

We do not necessarily need to introduce a new file format from scratch. The Hierarchical Data Format V.5 [9] is a widely used I/O library developed at NCSA with a corresponding file format, known as HDF5. The HDF5 API provides many

---

[6] LIght Detection And Ranging `http://www.lidarmapping.com`

[7] `http://www.remotesensing.org/geotiff/geotiff.html`

[8] `http://www.research.ibm.com/people/l/lloydt/dm/DM.htm`

[9] Hierarchical Data Format version 5 `http://hdf.ncsa.uiuc.edu/HDF5/`

unique features, which are particularly valuable in the context of Grid computing [9]. However, while HDF5 provides a syntax for the efficient representation of scientific data, there still remains ambiguity in how to formulate a certain type of scientific data. The layout in the concept of a fiber bundle provides a direction toward narrowing down such ambiguities, and at the same time defining generic operations. Such a layout is still not unique per se; our version [10] has been shown to be able of covering a wide range of data types. Among other features, it intrinsically supports the notion of time and handles scalar, vector, tensor and other multivalued fields of arbitrary dimensions on regular and irregular mesh triangulation schemes.

The fiber bundle HDF5 formulation according to [10] ("F5") casts data into a non-cyclic graph of five levels, called the `Slice`, `Grid`, `Topology`, `Representation` and `Field` levels, with two additional invisible levels describing internal memory layout. This graph maps well to the hierarchical grouping scheme of HDF5 by identifying the nodes of the graph with HDF5 groups.

Writing custom file converters to transform the time-varying surge surfaces, wind, pressure and temperature volumes into the `F5` format was a one-time effort. The conversion of MM5 data was done via translation into the intermediate NetCDF format[10] generated by the utilities available from the MM5 site[11]. Even though NetCDF shares some similarities with the HDF5 (self-description, platform independence), it is missing several of its crucial features, such as the capability of organization of datasets in named hierarchies as well as allowing their cross-referencing at metadata level. The second step of the translation focused on the extraction of the desired data volumes from the NetCDF files, reorganizing them in memory and storing the resulting buffers in correctly annotated F5 hierarchy. We simplified this approach by treating cell-related quantities as given on vertices and σ-level as height coordinates .

A file-system like listing of the 5-levels of the F5 structure of MM5 data (regular uniform grid with three fields) will then appear as

```
/T=1.0/MM5/Points/Cartesian Group
/T=1.0/MM5/Points/Cartesian/Positions Group
/T=1.0/MM5/Points/Cartesian/wind Dataset {43, 135, 174}
/T=1.0/MM5/Points/Cartesian/temperature Dataset {43, 135, 174}
/T=1.0/MM5/Points/Cartesian/pressure_perturbation Dataset {43, 135, 174}
```

whereby the fifth level contains the actual data, displayed here with their shared dimensionality.

The ADCIRC data set provides topological information about the connectivity of vertex points, explicit vertex coordinates and scalar values denoting surge elevation on each vertex. The F5 structure listing appears as:

```
/T=1.0/ADCIRC/Connectivity Group
/T=1.0/ADCIRC/Connectivity/Points Group
/T=1.0/ADCIRC/Connectivity/Points/Positions Dataset {1190404}
```

---

[10] `http://www.unidata.ucar.edu/software/netcdf`

[11] `ftp://ftp.ucar.edu/mesouser/user-contrib/mm5tonetcdf_1.2.tar.gz`

```
/T=1.0/ADCIRC/Points Group
/T=1.0/ADCIRC/Points/Cartesian Group
/T=1.0/ADCIRC/Points/Cartesian/Positions Dataset {598240}
/T=1.0/ADCIRC/Points/Cartesian/elevation Dataset {598240}
```

Going via F5 reduced the loading time of the large time-varying datasets provided originally as text files from several minutes to a fraction of a second. Moreover, the integrated caching algorithms in the HDF5 library itself eases loading of data on demand, both for ADCIRC as well as for MM5, as both can be accessed through the same interface. They may even be stored in the same file, thereby allowing to specify relationships among both simulations types and ensuring consistency (e.g., with respect to same timescale).

3.4 Data Import For Visualization

We used the Amira visualization tool [11] for rendering. It does not have an intrinsic notion of time-dependent objects and supports only static geometries well. Each `Grid` node in the fiber bundle hierarchy describes a geometry at a certain time step and can thus be mapped into a static geometry. The enveloping `Slice` level provides a sequence of `Grid` objects. Thus we extended the Amira class hierarchy by deriving dynamic objects from their static pendants. This recipe is straightforward to implement and scales well to the diverse data types. As drawback, this approach does not allow to inspect more than one time step at once (e.g. in different viewers) except by copying the entire visualization network. Another implementation issue is that not all of the Amira base classes allow easy modification of their properties once created.

The mapping of `Grid` objects to static objects works fine for entirely time-varying objects (as long as the topological type, e.g. of being a triangular surface, remains the same). However, some components of the dynamic `Grid` may well remain constant. In particular, the connectivity and vertex location of the ADCIRC grid does not change through time, only the data values (surge elevation, wind velocity) evolve. We can address this issue by utilizing symbolic links among HDF5 datasets - a feature provided by HDF5 similar to a Unix filesystem. For instance, we make a symbolic link of the `Grid`'s connectivity information at time 1290.0 to the connectivity information of time 0.0 to indicate that it did not change, resulting a structure as follows:

```
/T=1290/ADCIRC/Connectivity Group
/T=1290/ADCIRC/Connectivity/Points Group
/T=1290/ADCIRC/Connectivity/Points/Positions Dataset,
⟶        same as /T=0/ADCIRC/Connectivity/Points/Positions
/T=1290/ADCIRC/Points Group
/T=1290/ADCIRC/Points/Cartesian Group
/T=1290/ADCIRC/Points/Cartesian/Positions Dataset,
⟶        same as /T=0/ADCIRC/Points/Cartesian/Positions
/T=1290/ADCIRC/Points/Cartesian/elevation Dataset {598240}
```

This way we can easily specify any property of an evolving `Grid` to remain constant, equally referring to the entire time range, just a time interval or even intermittent. This feature can well be utilized as certain ADCIRC runs are also performed on a mesh that is modified once during the simulation in order to cope with levee failure. We are not aware about any other file format which supports a comparable mechanism to express partial time-dependency.

4 Specific Visualization Algorithms

4.1 Atmospheric Data (MM5)

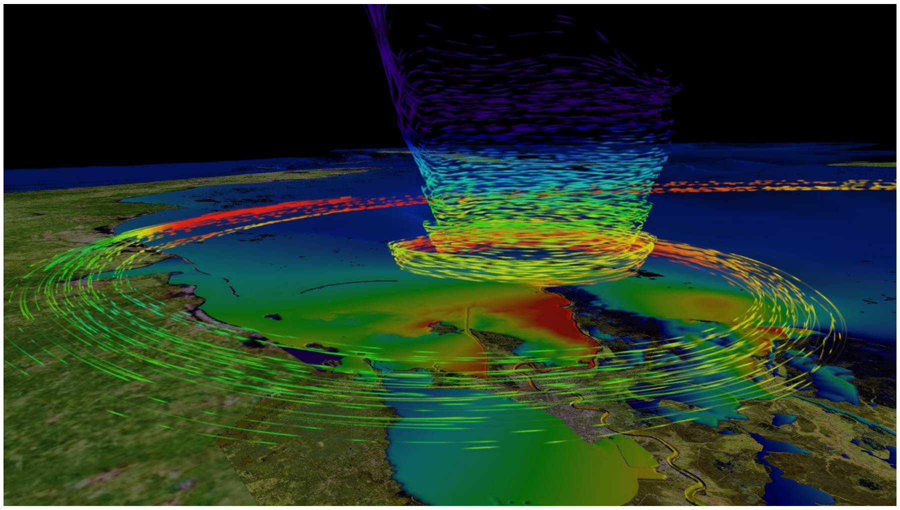

Fig. 1. Streamlines of the hurricane wind vector field at landfall. The streamlines are color-coded by temperature showing higher temperatures above sea surface than at land indicating loss of energy after landfall, while at the same time depicting the push onto the Lake Ponchartrain causing the flood in the city of New Orleans.

Amira provides many means of visualizing vector fields such as LIC, streamsurfaces and streamlines [12]. The non-commercial research version also includes advanced algorithms for extracting and displaying topological features [13, 14] However, for our purposes of communicating the results of hurricane simulations to the public and scientists unfamiliar with vector field topologies, we found the the technique of illuminated stream lines [15] most intuitive beside simple vector arrows icons. While vector arrows are frequently used as a first step and easily convey the values of a vector field, they do not scale well to display its global structure. Streamlines are superior to depict features such as the vortex of a

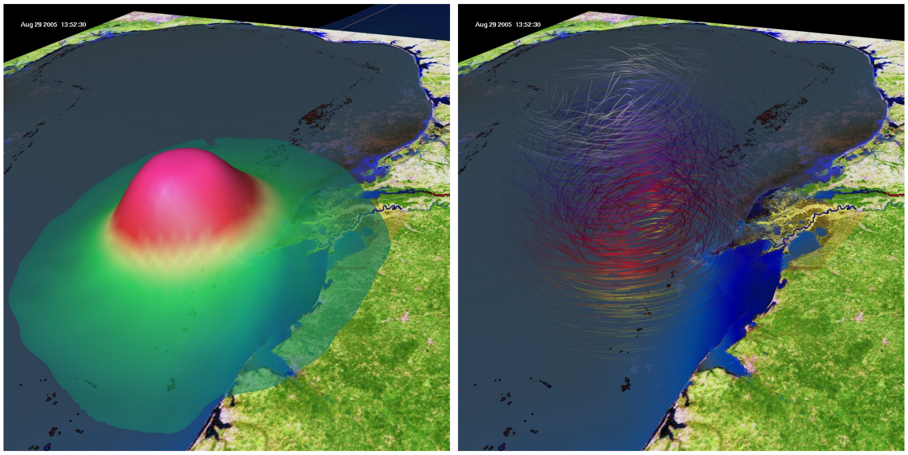

Fig. 2. The pressure scalar field indicates the location of the eye of the hurricane (left). We use it to set the transparency of the streamlines (right), thereby emphasizing the hurricane's eye in an automated way which is suitable for animation.

hurricane. The *seeding* of streamlines is a critical issue affecting the overall appearance. For a static view, we can manually seed the streamlines within region of interest such as the city of New Orleans (Fig. 1). This approach does not extend to a dynamic vector field, where streamlines are no longer appropriate at all due to their vastly changing character. However, we can reduce the length of the streamlines radically such they only depict local variations of the vector field, as these are more likely to be temporally smooth than global features. We need to compensate the smaller line fragments by increasing the density of lines. Consequently this increases the visual clutter again and requires suppressing of regions in the volume where the wind is of minor relevance. Such regions are indicated by the *pressure*, see Fig. 2. We therefore map this scalar field to the transparency of the stream lines and get wind field indicators limited to the vicinity of the eye of the hurricane.

4.2 Surge Data (ADCIRC)

The ADCIRC data set consists of more than one million triangles plus time-varying scalar for surge elevation and a vector field for wind information. In order to achieve good rendering performance we utilize OpenGL extensions such as Vertex Buffer Objects. Surge elevation is most intuitively represented by modifying the vertex locations like a height field. For a triangular surface given in 3D, this is not straightforward because there is a freedom of choice in which direction to extrude the surface at each vertex. For the special case here we may just concentrate on the z (height) direction of the surface, which denotes the bathymetry of the sea ground. We *blend* this bathymetric value and the surge elevation, as this provides a visually appealing mean to display structure within otherwise homogeneously watered regions.

4.3 Elevation Data (LIDAR)

A LIDAR data set is canonically visualized as a height-field. The extremely high-resolution of GIS elevation models, 11Kx7K in our case renders 154 million triangles using a brute-force triangulation method. Interactive rendering is virtually impossible with this approach. We have implemented Continuous Level-of-Detail (CLOD) techniques to dynamically simplify the mesh at run-time depending on the view point. Several algorithms already exist in this area. We chose the ROAM [16] algorithm due to its inherent simplicity and low memory overhead. At every frame, ROAM recursively tessellates the terrain generating triangles depending on the distance to the viewer criteria (or one could also use surface roughness). One nice feature of the recursive method is that we are not storing any per-vertex data but just generating them on the fly for the drawing, freeing up huge amounts of memory. We use the automatic texture coordinate generation functionality in OpenGL, mapping texture coordinates to the vertices. The drawback of this approach is heavy computation on the CPU and only using the GPU for drawing triangles.

4.4 Cloud Data (GOES)

The channel from the GOES satellites do not correspond to visual colors, so they cannot be used to create a true-color image. Five channels are beyond the capabilites of our trichromatic color perception anyway, so we face the challenge of appropriate representation. As we also require integrated display with atmospheric, surge, LIDAR and GIS data, the cloud representation needs to be minimalistic and we refrain from displaying all channels at once. The visible or IR channels are appropriately displayed as a transparent 2D gray-scale layer; after geospatial alignment their evolution allows depicting the match with atmospheric

Fig. 3. Match (left) and mismatch (right) of atmospheric simulation and satellite data.

data (see Fig. 3), where the discontinuity in the MM5 model data mentioned above appears more or less prominently. Alternatively it is also reasonable to represent the long-wave IR channel as height field, because this channel which is directly related to the physical height of the cloud cover through their temperature. Thus we can employ again the height field rendering algorithm described earlier, using the GOES-12 visible channel as texture.

5 Conclusions

We have shown how various data sources ranging from computational models of storm-surge and wind fields to observational data from satellites and sensors can be integrated into a holistic, compelling and interactive visualization of Hurricane Katrina. The selected methods illustrate the interaction between topographical and surge data (LIDAR/ADCIRC), the development of the surge as predicted from the atmospheric model (ADCIRC/MM5) and allow to assess the deviation of the atmospheric model from observation (MM5/GOES), all within the geospatial context provided by GIS reference images. We have developed efficient data layout mechanisms to ensure fast and uniform access to the multiple time-varying datasets. Existing rendering techniques were also applied and extended to better understand the phenomenon and her effects. All these above efforts required new partnerships between coastal modelers, engineers and computer scientists.

6 Acknowledgements

We thank our colleagues Shreekanth Balasubramanian, Dewitt Braud, Chirag Dekate, Carola Jesch and Wei He and LSU collaborators - Earth Scan Laboratory, Coastal Studies Institute, CLEAR and Hurricane Center, SURA SCOOP, UNC Department of Marine Science and TAMU for invaluable assistance with the visualization and data access.

References

1. Allen, G., Bogden, P., Creager, G., Dekate, C., Jesch, C., Kaiser, H., MacLaren, J., Perrie, W., Stone, G., Zhang, X.: Gis and integrated coastal ocean forecasting. Submitted to Concurrency and Computation: Practice and Experience (2006)
2. MacLaren, J., Allen, G., Dekate, C., Huang, D., Hutanu, A., Zhang, C.: Shelter from the storm: Building a safe archive in a hostile world. OTM 2005 Workshops, LNCS 3762 (2005) pp. 294–303
3. Nativi, S., Blumenthal, B., Habermann, T., Hertzmann, D., Raskin, R., Caron, J., Domenico, B., Ho, Y., Weber, J.: Differences among the data models used by the geographic information systems and atmospheric science communities. In: Proceedings American Meteorological Society - 20th Interactive Image Processing Systems Conference. (2004)

4. Helgeland, A., Elboth, T.: High-quality and interactive animations of 3d time-varying vector fields. IEEE Transactions on Visualization and Computer Graphics (2006)
5. Doleisch, H., Gasser, M., Hauser, H.: Interactive feature specification for focus+context visualization of complex simulation data. In: VISSYM '03: Proceedings of the symposium on Data visualisation 2003, Aire-la-Ville, Switzerland, Switzerland, Eurographics Association (2003) 239–248
6. Zhang, K., Chen, S.C., Singh, P., Saleem, K., Zhao, N.: A 3d visualization system for hurricane storm-surge flooding. IEEE Computer Graphics and Applications **26**(1) (2006) 18–25
7. Butler, D.M., Pendley, M.H.: A visualization model based on the mathematics of fiber bundles. Computers in Physics **3**(5) (1989) 45–51
8. Butler, D.M., Bryson, S.: Vector bundle classes from powerful tool for scientific visualization. Computers in Physics **6** (1992) 576–584
9. Benger, W., Hege, H.C., Merzky, A., Radke, T., Seidel, E.: Efficient Distributed File I/O for Visualization in Grid Environments. In Engquist, B., Johnsson, L., Hammill, M., Short, F., eds.: Simulation and Visualization on the Grid. Volume 13 of Lecture Notes in Comp. Science and Engineering. Springer Verlag (2000) p. 1–6
10. Benger, W.: Visualization of General Relativistic Tensor Fields via a Fiber Bundle Data Model. PhD thesis, Free University Berlin (2004)
11. Stalling, D., Westerhoff, M., Hege, H.C.: Amira - an object oriented system for visual data analysis. In Johnson, C.R., Hansen, C.D., eds.: Visualization Handbook, Academic Press (2005)
12. Stalling, D.: Fast Texture-Based Algorithms for Vector Field Visualization. PhD thesis, Free University Berlin (1998)
13. Theisel, H., Weinkauf, T., Hege, H.C., Seidel, H.P.: Saddle connectors - an approach to visualizing the topological skeleton of complex 3d vector fields. In Turk, G., van Wijk, J.J., Moorhead, R., eds.: Proc. IEEE Visualization 2003, Seattle, U.S.A. (2003) 225–232
14. Weinkauf, T., Theisel, H., Shi, K., Hege, H.C., Seidel, H.P.: Extracting higher order critical points and topological simplification of 3D vector fields. In: Proc. IEEE Visualization 2005, Minneapolis, U.S.A. (2005) 559–566
15. Zöckler, M., Stalling, D., Hege, H.C.: Interactive visualization of 3d-vector fields using illuminated streamlines. In: Visualization '96. (1996) 107–113
16. Duchaineau, M., Wolinsky, M., Sigeti, D., Miller, M., Aldrich, C., Mineed-Weinstein, M.: Roaming terrain: Real-time optimally adapting meshes. In: Proceedings of IEEE Visualization '97, IEEE (1997) 81–88