

## Colliding galaxies, rotating neutron stars and merging black holes—visualizing high dimensional datasets on arbitrary meshes

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2008 *New J. Phys.* 10 125004

(<http://iopscience.iop.org/1367-2630/10/12/125004>)

[View the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 80.215.210.117

This content was downloaded on 03/01/2017 at 20:28

Please note that [terms and conditions apply](#).

# New Journal of Physics

The open-access journal for physics

## Colliding galaxies, rotating neutron stars and merging black holes—visualizing high dimensional datasets on arbitrary meshes

**Werner Benger**

Center for Computation and Technology, Louisiana State University,  
Baton Rouge, LA, USA  
E-mail: [werner@cct.lsu.edu](mailto:werner@cct.lsu.edu)

*New Journal of Physics* **10** (2008) 125004 (24pp)

Received 14 April 2008

Published 1 December 2008

Online at <http://www.njp.org/>

doi:10.1088/1367-2630/10/12/125004

**Abstract.** Visualization of datasets stemming from diverse sources is challenged by the large variety of substantial differences in topology, geometry and nature of the associated data fields. Since there is no standard on how to formulate and treat data for scientific visualization, algorithms are frequently implemented in a highly domain-specific way. Here, we explore the potential of point-wise rendering as a generic way to represent single or multiple fields instantaneously on arbitrary mesh types. This approach is discussed within the terminology of fiber bundles as a general mathematical concept to model scalar-, vector- and tensorfields given on topological spaces (with manifolds as a particular case). We give application examples based on datasets originating from astrophysics and show first results of a tensor field visualization of a recently produced complex dataset of colliding black holes in their final orbit. We finally propose a data layout representing the mathematical concept of a ‘field’ generic enough to handle all cases involved.

## Contents

<b>1. Introduction</b>	<b>2</b>
1.1. Outline . . . . .	3
<b>2. The technique</b>	<b>3</b>
2.1. Nomenclature . . . . .	3
2.2. Exemplification . . . . .	4
2.3. High-dimensional fiber space . . . . .	5
2.4. High-dimensional base space . . . . .	7
<b>3. Applications</b>	<b>8</b>
3.1. Particle systems—colliding galaxies . . . . .	9
3.2. Multipatch data—rotating neutron star . . . . .	15
3.3. AMR—colliding black holes . . . . .	16
<b>4. Data organization</b>	<b>20</b>
<b>5. Conclusion</b>	<b>23</b>
<b>Acknowledgments</b>	<b>23</b>

### 1. Introduction

Astrophysical data come in a large variety and tend to be huge. While many numerical simulation algorithms have been developed for a highly specific purpose, they intrinsically share common properties among each other, and with other scientific domains beyond astrophysics. Attempts to conceive a common data model to unify the variety of data are rare. However, they become increasingly mandatory with the complexity of simulations, their resulting data, and the need for interoperability between simulation code as well as eventually with visualization environments. Butler and Pendley (1989) and Butler and Bryson (1992) proposed to consider datasets as fiber bundles. This model was in particular followed within the ASCI project (Cook and Matarazzo 1998). The IBM Data Explorer (Treinish 1997) (now OpenDX) is a widely known implementation of this mathematical model as applied to scientific visualization. Recent modern approaches build on *generic programming* to decouple algorithms from data structures (Berti 2000, Veldhuizen 1995). However, generic programming is not suitable for interactive applications where decisions about data types have to be made at runtime.

In this paper, we discuss the potential of using only the per-vertex information of a mesh as a generic approach covering many data types. Utilizing Gaussian footprints for volume rendering has been widely used for unstructured meshes (Crawfis and Max 1993a, b, Kreylos *et al* 2000, Westover 1990). Structured meshes are covered by texture-based volume rendering in many modern software packages (Kaufman and Mueller 2004, Kitware 2005, Stalling *et al* 2005, Treinish 1997). The handling of adaptive mesh refinement (AMR) grids (constructed from a hierarchy of regular meshes) requires special concentration (Kähler 2005, Weber *et al* 2001). However, these techniques are mostly suitable for rendering single scalar fields. Here, we discuss visualizing multiple fields (such as many scalar fields, or a multi-component tensor field). In particular, the technique to visualize a tensor field via Gabor patches, as described in (Benger and Hege 2004), is very similar to volume rendering via Gaussian footprints and is therefore applicable to arbitrary meshes.

The visualization routines described here are identical for all data types, from particle data to adaptive meshes to multiblock data. The high re-usability of visualization code allows features developed specifically for a certain data type also to be applied to others, possibly leading to new insight. We demonstrate such experimental visualization here, and intend to open new views on how to look at certain data.

### *1.1. Outline*

Section 2 provides a basic introduction to the terms and techniques, and exemplifies visualizing multiple fields based on simple artificial datasets. Section 3 demonstrates the application of these basic concepts to recently generated astrophysical datasets: a particle system modeling colliding galaxies, a multipatch dataset describing a stationary star and an AMR dataset of colliding black holes. Where appropriate, the instantaneous visualization of multiple fields is demonstrated. Finally, section 4 presents the data organization scheme that was used to treat the diverse data fields in a common way.

## **2. The technique**

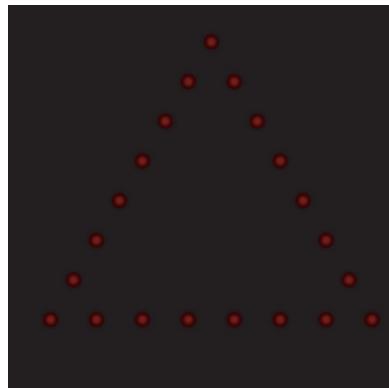
We demonstrate our approach to render a large variety of multidimensional data by means of a very simple artificial dataset. At first, we need to specify what we mean by ‘multidimensional’. In order to do so, we employ the mathematical concept of a fiber bundle, as it can be found in topology. In simple words, a fiber bundle is a space which looks locally like a product space, i.e. small regions of a total space  $E$  look like small regions in the product space  $B \times F$ . Here,  $B$  is called the base space and  $F$  is called the fiber space. The global topological structure of the total space may be different from  $B \times F$ . If so, it is called a nontrivial fiber bundle. The datasets that occur in scientific visualization can be well described as trivial fiber bundles (Butler and Bryson 1992, Butler and Pendley 1989).

### *2.1. Nomenclature*

With respect to dimensionality, we henceforth distinguish between the dimensionality of the base space  $\text{dims}(B)$  and the dimensionality of the fiber space  $\text{dims}(F)$ . Data given on a plane or surface will thus be said to be of base space dimensionality two, data given in a volumetric domain to be of base space dimensionality three, and dynamic volumes of dimensionality four. Higher dimensional base spaces may refer to e.g. parameter studies of simulation sequences, which output evolving volumes of data, each containing a different sequence depending on some initial parameter.

At each point within the base space we will have some given data and these define the fiber space. A pure geometry (such as a mesh) will not have such fiber space data associated. If a scalar field is given on each point, the fiber space will be of dimensionality one, multiple scalar fields, vector or tensor fields will increase the dimensionality. At this point, the properties of the fiber space are not of relevance; it will in certain cases have vector space properties and is then said to be part of a vector bundle.

This scheme in its narrow sense only refers to data given on the vertices of a discretized topological space. In general, we might also have data given on different entities, such as the triangles of a surface or the cells of a mesh. Within the scope of this paper, we limit ourselves



**Figure 1.** Simple point set that will be used for illustrations. At each point, there are multiple datasets defined, and we seek methods to display them instantaneously.

to vertex-based data. In general, these can be constructed from other data, even though it might not be trivial.

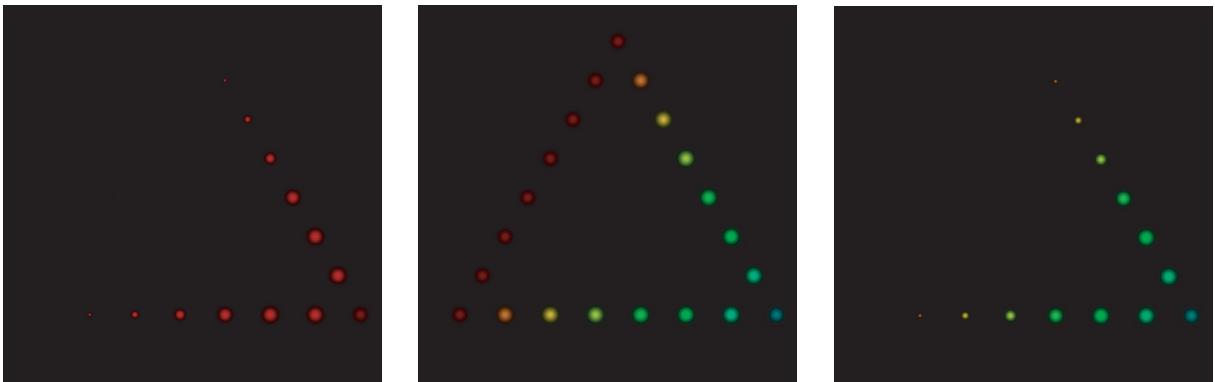
## 2.2. Exemplification

To introduce the ideas of a systematic approach on how to deal with high-dimensional data spaces, we will utilize a very simple artificial dataset. It is a mere set of points arranged along the edges of a triangle—see figure 1, with three scalar fields that are given by the barycentric coordinates  $(r, s, t)$  of each point. These three coordinates fulfill the condition  $r + s + t = 1$  (for a normalized triangle) and are thus interdependent.

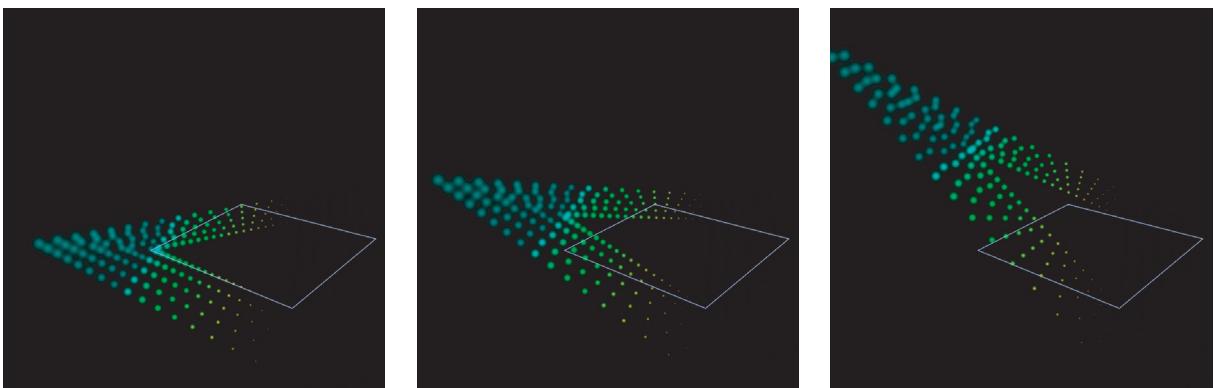
Mapping the values of a scalar field to colors is a common approach and may yield an already three-dimensional (3D) or 4D representation, when the red, green and blue channels are used independently and eventually transparency is taken into account as well. While such a 3D color mapping is an option, it is not necessarily the perceptually best one, because not all colors are perceived equally by the human spectator. Thus, linear data may appear highly nonlinear, and may lead to visual artifacts. In the following, we will use only 1D color mappings, i.e. all colors will refer to the same 1D data field and only represent different values within.

Beside coloring, when representing singular points we have freedom in how to display each. Instead of a one-pixel representation that is hardly visible, we use a Gaussian intensity distribution as a natural choice here. This can be rendered by modern graphics hardware extremely well using so-called point sprites. The half-width of the Gaussian sprite is now a free parameter that can be used in addition to the coloring, and can be used to represent a 2D fiber space, as in figure 2 (complementary to vector arrow icons, which also represent a 2D fiber space, but are more suitable for vector spaces than general fiber spaces).

Further degrees of freedom are available via the geometric location of the vertices themselves. When representing the vertices of an  $n$ -dimensional base space within an  $m$ -dimensional embedding space, we have  $m$  degrees of freedom to represent data on the vertices by shifting their coordinate locations. In general, this will however lead to ambiguities if the original coordinate locations can not be clearly distinguished from the shifted ones. If the embedding space is higher dimensional than the data base space, using the additional



**Figure 2.** Point set with scalar data, displayed as variation of point size or coloring; both methods combined yield a visualization of two scalar fields.



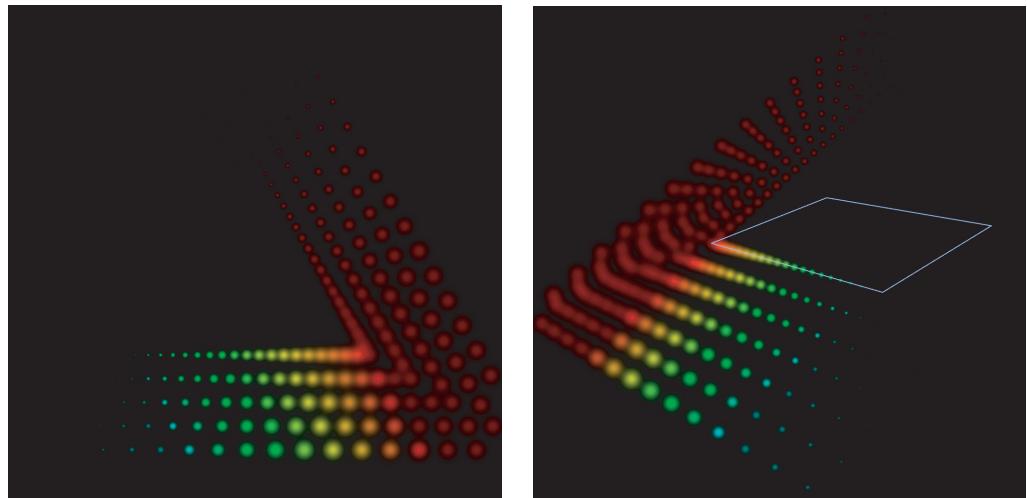
**Figure 3.** Using geometrical variance to display scalar data. The same field is redundantly mapped to color, point size and elevation (with increasing scaling parameter of the elevation for the purposes of illustration).

dimensions is always unique and can be used without ambiguity concerns (for instance, the height field of a 2D dataset in a 3D graphical representation). An example is given in figure 3, where we also have a unique direction of the additional dimension within 3D space, as given by the plane of the triangle that defines the 2D base space.

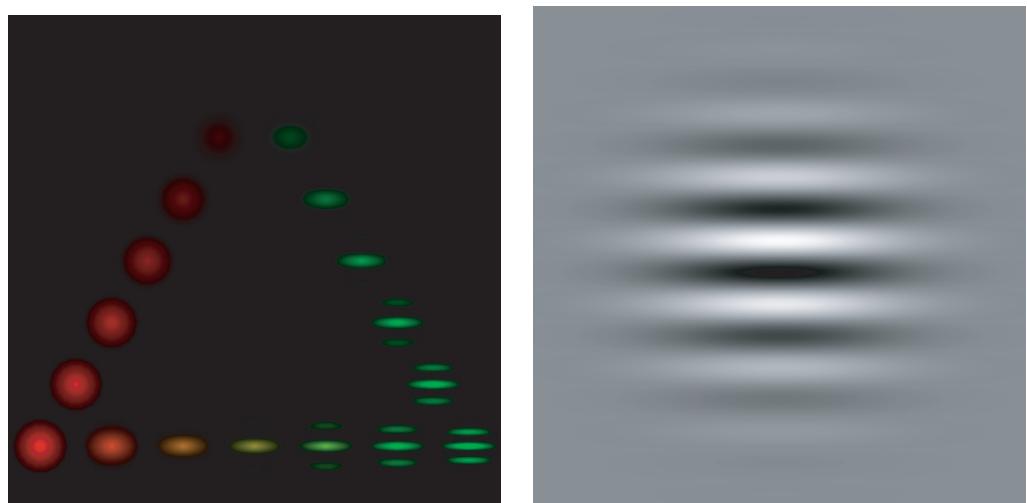
Finally, figure 4 displays all three scalar quantities as given on the triangular test case by independent means of size, color and ‘elevation’ as form of varying the geometric location of the point locations. While in this case the concept of ‘elevation’ is uniquely defined, it will, however, be demonstrated in the text below that the ambiguity of varying a geometric location within the coordinate space of the vertices itself is not as bad as one might believe initially.

### 2.3. High-dimensional fiber space

Varying the size of a spherical Gaussian distribution function represents a 1D fiber space. More complex shape functions may be utilized to represent higher dimensional fiber spaces. For instance, the half-width can also be varied, or an additional radial frequency imposed, such as with an Airy disk (the diffraction pattern cast by a point light source). Non-symmetrical modifications of the point shape are ideally used to represent an intrinsically geometric property



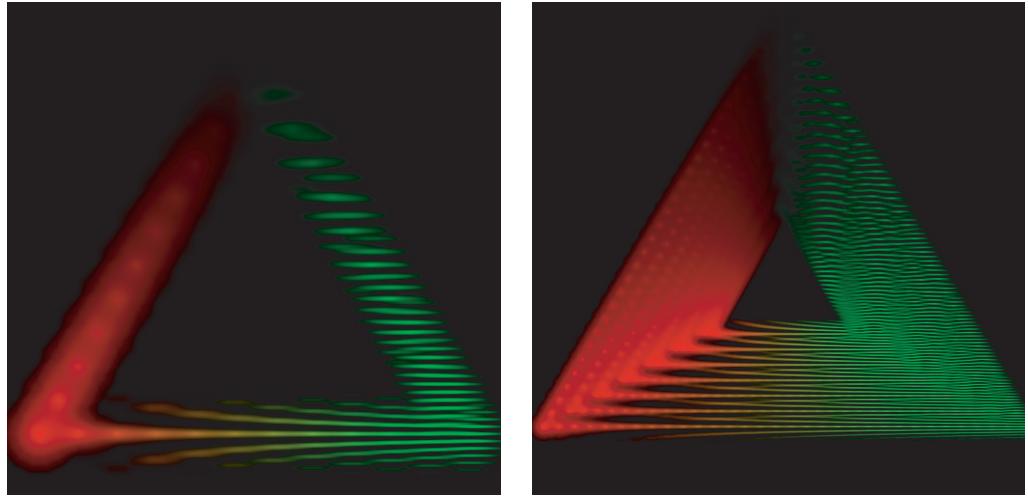
**Figure 4.** Using point size, coloring (left) and geometrical variance (right) to display three independent scalar fields on a set of points, thus eventually yielding a ‘visualization of a 3D fiber space over a 2D base space’.



**Figure 5.** Visualization of a 5D fiber space (a symmetric tensor field of second order, the technique of ‘tensor patterns’—left figure) via parameterized Gabor patches (right).

of the data field in the fiber space. For instance the direction of a vector field at each point is well represented by a vector arrow icon or vector splat (Crawfis and Max 1993*b*), the metric properties of a tensor field of second order are well represented by the cross section of a quadric surface.

One such specific technique was presented in (Benger and Hege 2004) and more thoroughly described in (Benger *et al* 2006) by means of medical datasets. It was substantially inspired by (Laidlaw *et al* 1998). This technique called ‘tensor patterns’ allows representation of six independent, but related quantities (such as the components of a symmetric  $3 \times 3$  tensor field) at each point in space. Instead of a one-parametric radial Gaussian distribution it employs Gabor patches, which are used as an optical stimulus in vision theory—see figure 5, right.



**Figure 6.** Scaling of the vertex primitives allows a nearly continuous impression of the dataset (left: same number of points as in figure 5, right: increasing number of points), suitable for visualization of a 5D dataset, e.g. for data mining purposes and reading of global structure variations.

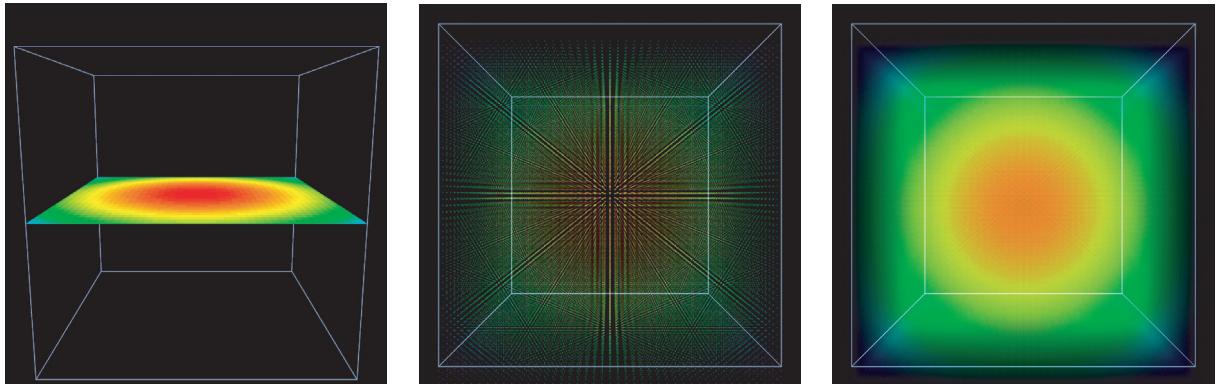
This visual primitive allows encoding more parameters per point than a pure Gaussian blob, as demonstrated in figure 5 (left).

An important benefit of point shapes with smoothly increased transparency at their boundaries, such as Gaussian blobs and Gabor patches, is their property of smoothly blending when scaled large. The result is a visually pleasing smooth impression of a continuous dataset as in figure 6. While information about a specific data point is less comprehensible by such overscaling, the overall global structure comes out more clearly. Such methods are thus suitable for ‘data mining’ purposes and first inspection of yet unknown datasets in order to identify global features and structures.

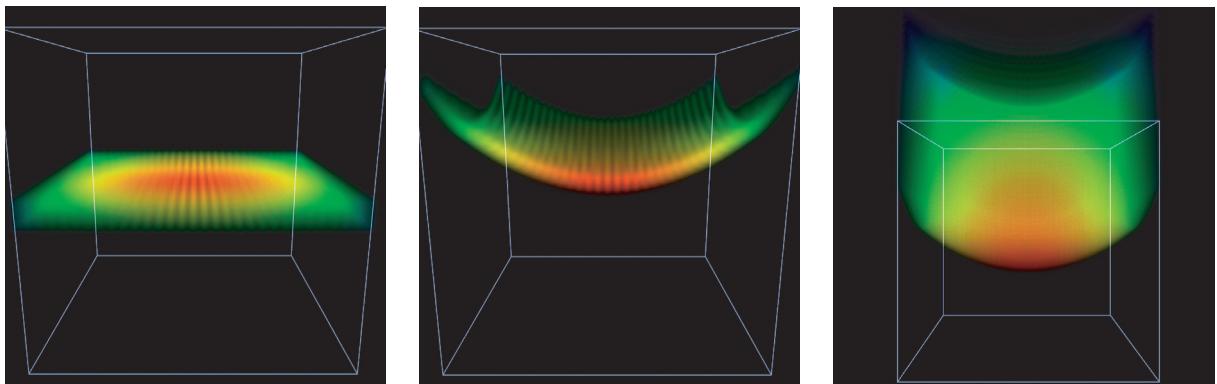
#### 2.4. High-dimensional base space

A natural approach to deal with high-dimensional base spaces is to slice them into lower-dimensional domains and do analysis of the resulting fiber space on this subdomain (figure 7, left). While depicting each data point as a geometric point is not helpful *per se* (figure 7, centre), the ability to employ smooth scaling of the primitives yields an effect known as volume rendering via splatting using Gaussian footprints (Kaufman and Mueller 2004). While volume rendering is a 1:1 representation of a entire scalar-valued dataset, its applicability is strongly dependent on the complexity of the structures found in the datasets. Finding an appropriate color mapping scheme (also known as transfer function) to bring out these features best is a research topic by itself (Kindlmann and Durkin 1998, Kaufman and Mueller 2004).

As a technique that merely uses the geometrical location of the data points, this variant of volume rendering can directly operate on an arbitrarily discretized base space. In contrast, texture-based approaches of volume rendering require a regular grid structure of the base space, by the benefit of allowing interpolation between the data values to achieve visually higher quality. Independent from the distribution of vertices in coordinates, the data points to be



**Figure 7.** Inspecting a 3D base space: via textured slice, as single points, as Gaussian point sprites scaled to blend with their neighbors (also known as volume rendering via splatting using Gaussian footprints).

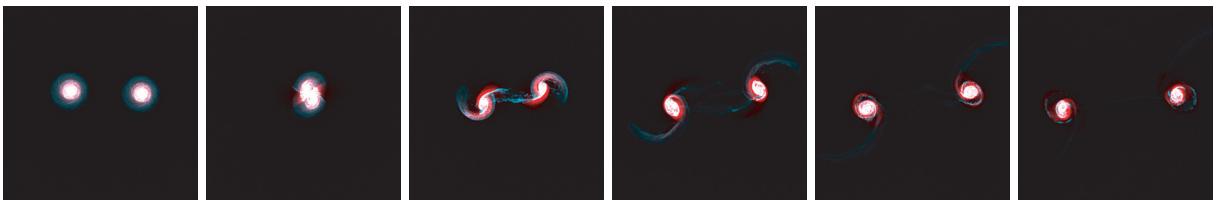


**Figure 8.** Reduction of 3D base space by arbitrary geometric cutting, adding geometrical variance, exploring full volumetric geometrical variance.

displayed can be easily restricted to a subdomain via the graphics hardware, thus achieving an effect of a planar slice but on arbitrary complex base space discretization schemes (more illustrative examples are given in the next section). By defining such a subset of the base space, geometric deviation of the point location can then be used in addition to pure color mapping, yielding the effect of a height field or elevation map. To a certain extent, this approach even works when applied to the entire volume, as illustrated in figure 8.

### 3. Applications

In the previous section, the notions of base space and fiber space with their respective dimensionality have been introduced, along with a systematic approach of how to visualize them via point-wise rendering. This approach is extremely fast, as data can be more or less directly transferred from the hard disk to the graphics card. Preprocessing is thus minimal, and interactive browsing of a high-dimensional parameter space—such as an evolution sequence over time—is achievable when utilizing appropriate caching. While the datasets in the introductory section have been trivial, we will now show the performance of the



**Figure 9.** Evolution of colliding galaxy dataset, SPH simulation based on 1617 286 points, using internal energy  $U$  for coloration.

discussed techniques by means of three intrinsically different datasets originating from practical astrophysical research. It is to be noted that the rendering code for each of these diverse datasets is identical.

### 3.1. Particle systems—colliding galaxies

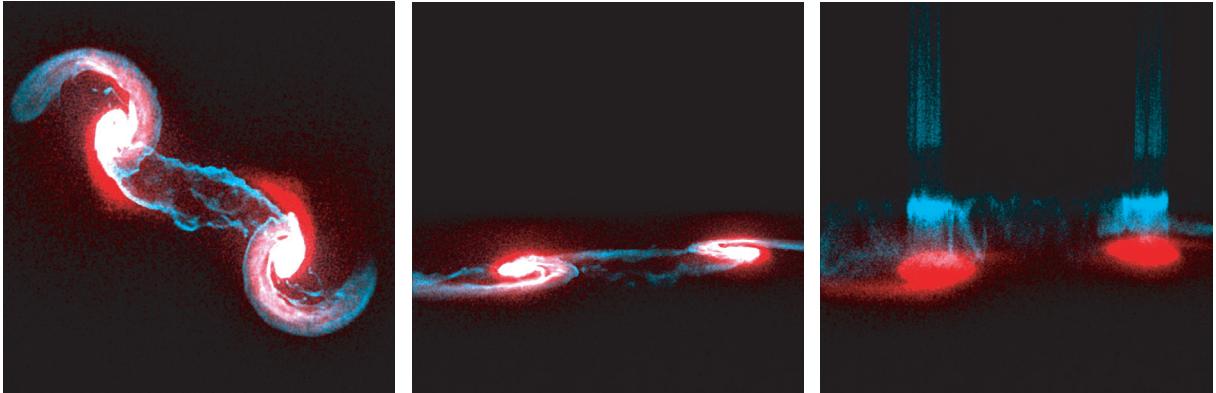
The first dataset explored here models two galaxies on collision course. It has been described by Kapferer *et al* (2005). The entire simulation covers about  $10^9$  years and required one week of computational time on a 40-core Opteron cluster. The simulation code is based on Gadget 2 (Springel 2005) and implements an  $N$ -body gravitational model coupled with smoothed particle hydrodynamics (SPH).

The dataset consists of 1.6 million points without explicit neighborhood information, distributed over 100 time steps. Several scalar fields are given on each point, these are: mass, internal energy ( $U$ ), density ( $\rho$ ), gravitational potential ( $\text{pot}$ ), SPH smoothing length ( $\text{hsml}$ ), resulting in 17 GB when stored in HDF5 (The HDF5 Group 2008). The simulation distinguishes between four kinds of particle types, describing gas, dark matter in the halo of the galaxy, existing stars in the disk of the galaxy and newly generated stars that arise from overdensities in the gas. In the fiber bundle terminology, this is a 4D base space with a 5D fiber space.

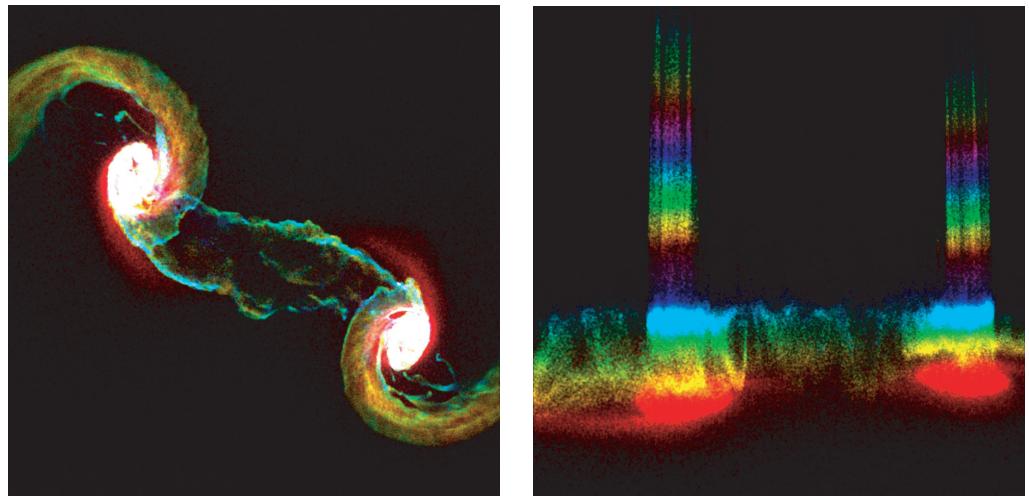
In the visualization shown here, we do not distinguish between these particle types, and seek to display the particles merely based on their numerical scalar fields given on them (though the particle type could, however, be seen as an integer-valued scalar field on the particles). Figure 9 displays six timesteps from the evolution sequence, using the internal energy  $U$  for coloration. These images are considered a ‘first time view’ of the data, without adjustment of color maps to the data range.

While the dataset comes with fully 3D coordinate locations, the physics of this very scenario results in a distribution of the particles dominantly in the same plane, as depicted by the nearly edge-on view in figure 10. This finding allows distributing the particles similar to an elevation field over a plane, and results in a geometrical representation of the scalar field, as in figure 10 (right). While this approach is ambiguous with respect to the point locations, the overall properties of the scalar field are still easier to perceive than via a color map since linearity is better sustained.

A more varying color map as in figure 11 reveals variation of the scalar field among the point locations, yet the geometrical variation (‘particle elevation’) depicts the scalar field more clearly. As in figure 11, right, both coloration and displacement redundantly refer to the same scalar field  $U$ .



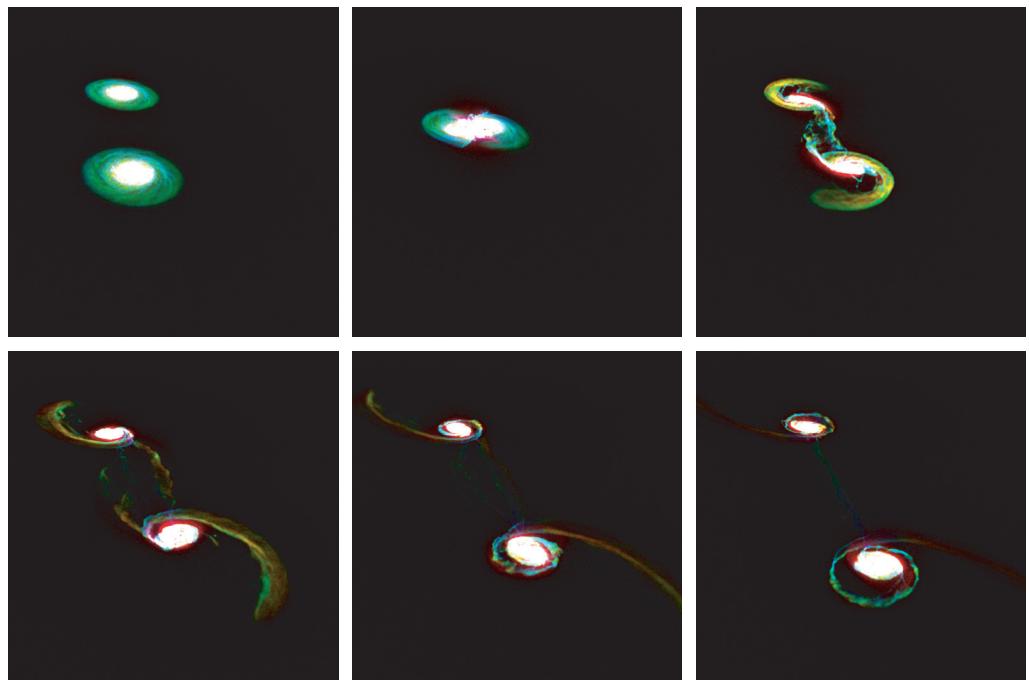
**Figure 10.** Nearly monochrome rendering of the 3D point distribution. It is dominantly concentrated around a common plane, thus we can shift the point locations from this plane based on data values. This is similar to an elevation field on surface data, but here employed upon a particle set.



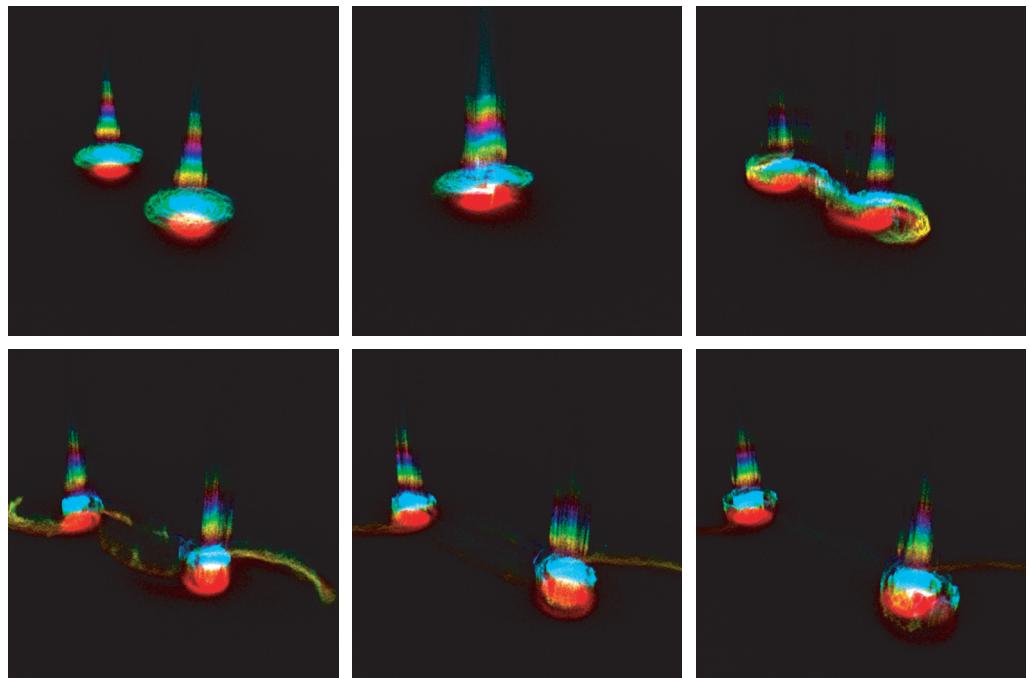
**Figure 11.** SPH dataset carrying internal energy  $U$  rendered with mere color mapping (left), and additional variation of geometry (right).

Finally, we compare an evolution sequence of the galaxy dataset, as visualized by pure color mapping (figure 12) versus adding geometrical displacement as in figure 13. While the pure color mapping results in a more ‘natural’ representation of the data, the elevation method might be superior to appreciate the properties of the data field. It is not within the scope of this paper to provide a physical interpretation here, but to demonstrate some ideas that may be useful for visual data analysis.

As we have two ways to represent a scalar field on a particle set, we may represent two independent fields this way. Depending on which field is mapped to which representation, the results may be easier or more difficult to grasp visually, as demonstrated in figure 14. The geometrical deviation is clearly perceptually overwhelming the coloration. Rendering the gravitational potential by elevation and internal energy  $U$  by coloration (figure 14, lower right) tells us that particles of similar gravitational potential still may have significant variations in



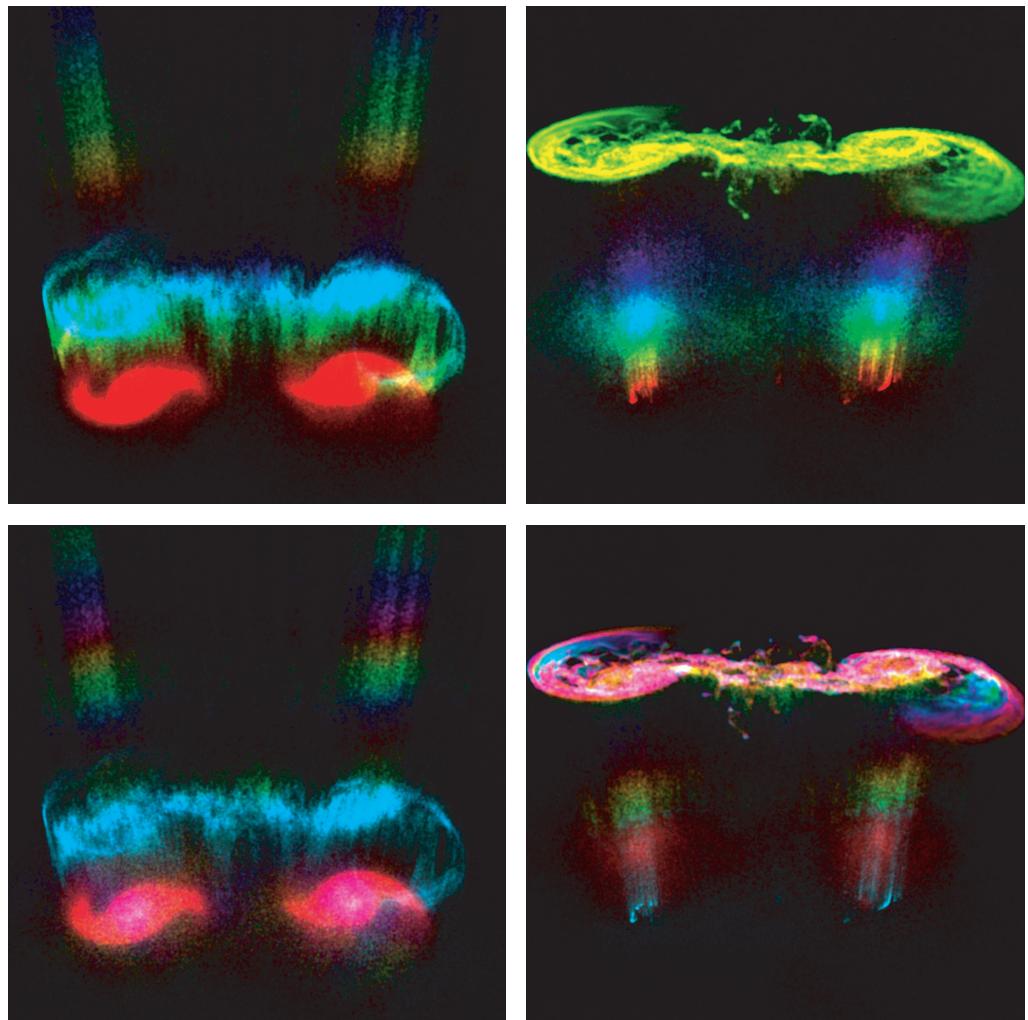
**Figure 12.** Colored evolution of SPH colliding galaxy dataset.



**Figure 13.** Animation sequence of the galaxy dataset using color and geometry variation.

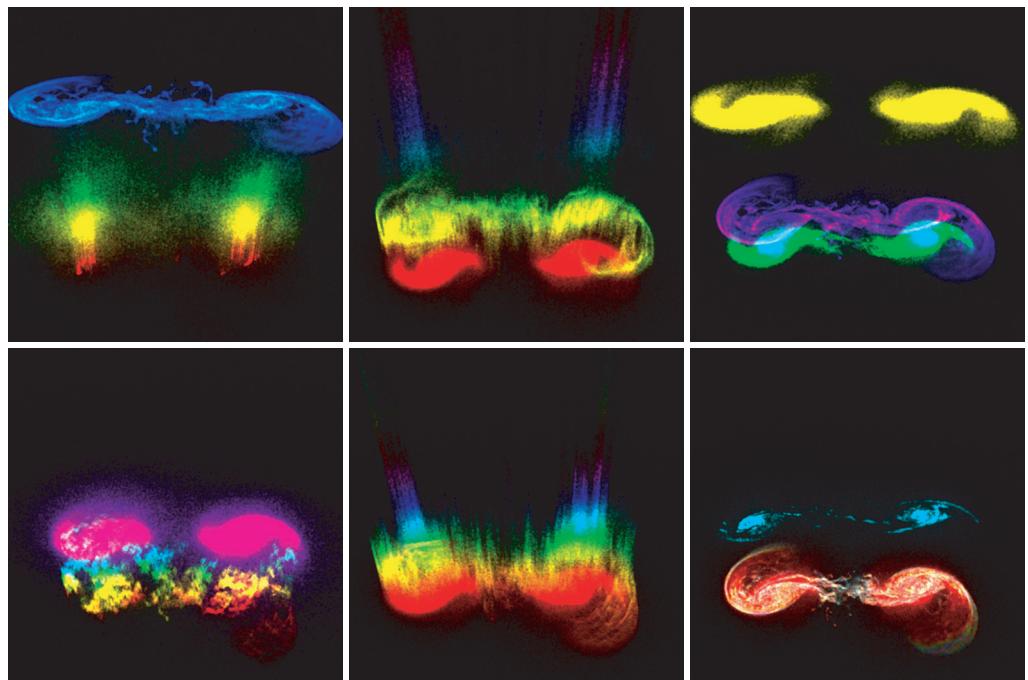
internal energy. This relationship is not obvious from the alternative choice as in figure 14 (lower left).

We may now explore all of the six scalar fields given on the particle set using coloration and displacement redundantly, as shown in figure 15. The displacement yields a visually strong

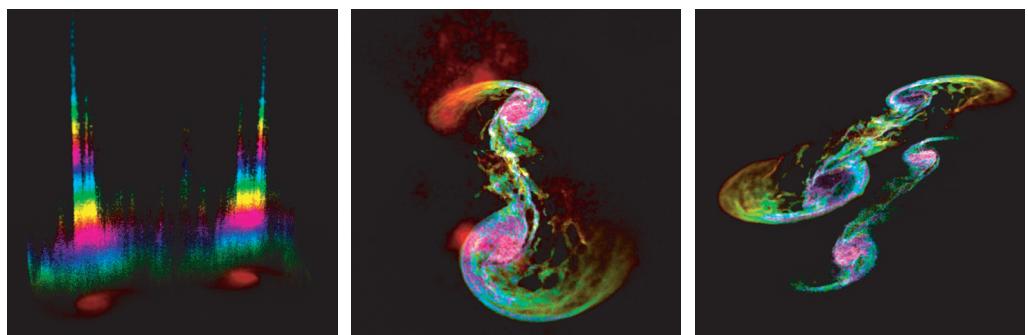


**Figure 14.** SPH dataset, internal energy ( $U$ ), potential energy (pot), elevation by  $U +$  color by pot, elevation by pot + color by  $U$ .

impression and provides an overview of ‘clusters’ of similar data values superior to mere coloration, in particular, since finding perceptually equidistant colors for the color transfer function is nontrivial, see e.g. (Teufel and Wehrhahn 2000). By introducing the point size as an additional visualization quantity as well, we arrive at the possibility to depict three scalar fields instantaneously. The point size is perceptually weak by itself if single points are not distinguishable such as in figure 16(left). However, the size of the splatting elements still contributes essentially to the appearance of the overall image by weighting certain regions. In the setup used in figure 16, we see particles in a strong gravitational potential, while those in weak potential are visually suppressed. Density variations in the particles are depicted in the coloration, whereby the additional geometrical displacement representing their masses shows that we have to deal with mainly two classes of massive particles here. Finally, we inspect an evolution sequence of the data using the setup to visualize the 3D fiber space, as in figure 17. Without *a priori* assumptions of the properties of the particles, it allows us to grasp the different densities of two classes of particles that reside in a large gravitational potential.

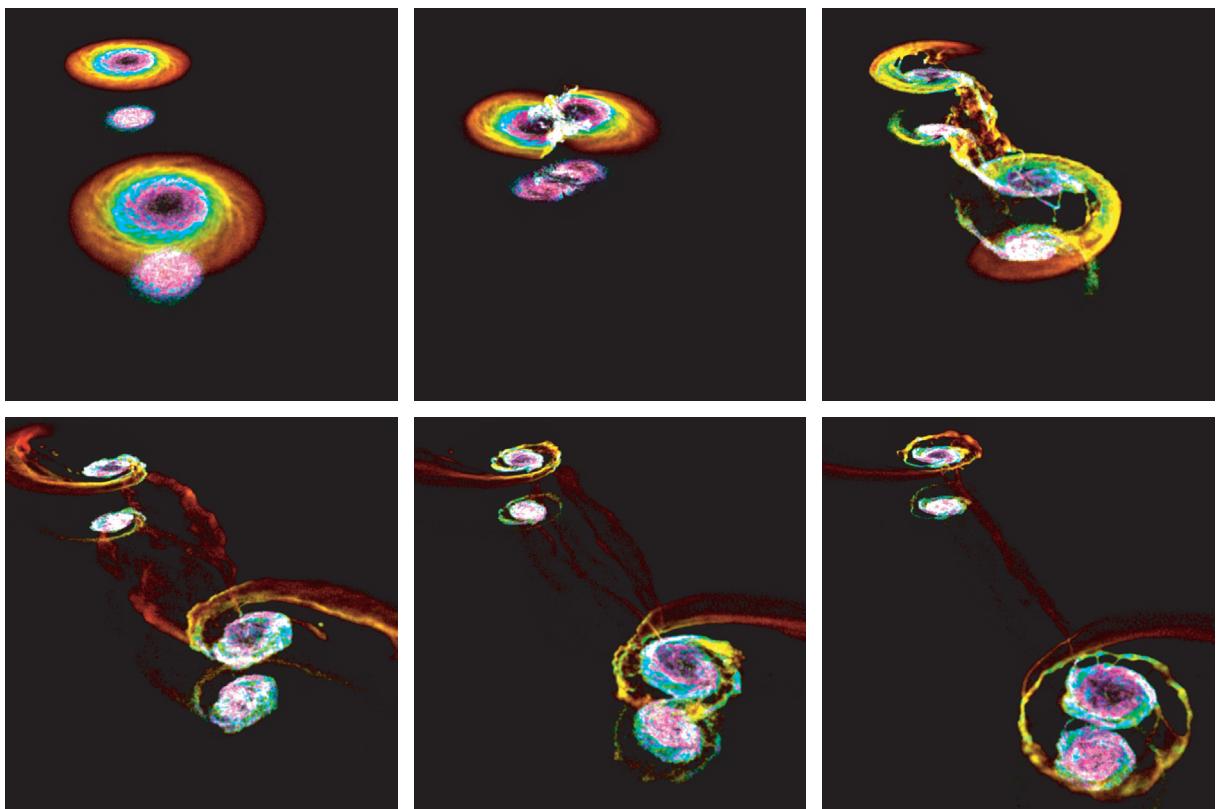


**Figure 15.** Display of gravitational potential (pot), internal energy, mass, ne, rho and hsml. Coloration and elevation displacement are used redundantly for each scalar field.



**Figure 16.** Exploring the 3D fiberspace of a 3D particle set: single field (rho), set triply redundant as point size, geometry, and color; two scalar fields using point size (pot) and color (rho); three scalar fields using size (pot), color (rho) and geometry (mass).

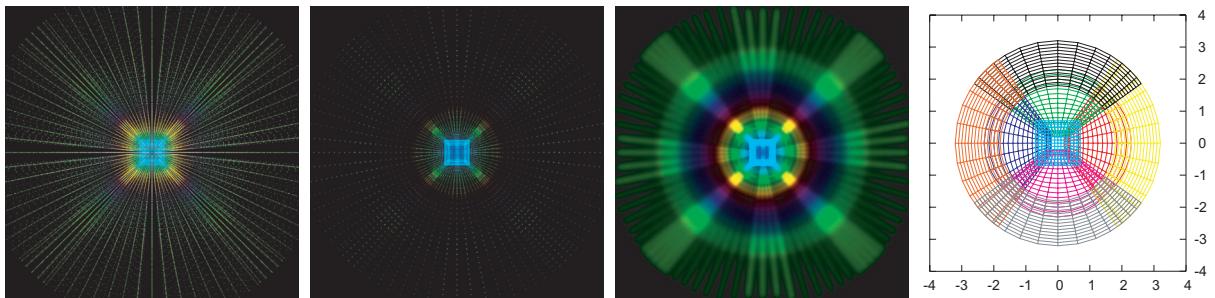
With appropriate caching mechanisms, rendering of the entire evolution can be done interactively and in real time. For instance, the dataset used here consisting of 1.6 million particles is rendered easily with 30 fps on a NVidia GeForce 7900 GTX graphics card. Rendering times are only impacted if the Gaussian point splats are scaled to be unrealistically large, such as about  $64 \times 64$  pixels each and beyond. Otherwise, the only limiting factor is given by the reading speed of the hard disk, i.e. visualizing the entire 17 GB dataset takes approximately the same time as copying a file of this size. However, in order to interactively



**Figure 17.** Animation sequence of galaxy dataset including color ( $\rho$ ), size ( $pot$ ) and geometry variation ( $mass$ ).

navigate in time, it is not required to keep the entire dataset of 17 GB in RAM. Rather, once data have been loaded and processed, they can be stored as vertex buffer objects (VBO) as a data representation that is optimal for the graphics processing unit (GPU), with one such VBO for each timestep. On replaying an animation sequence, no data need to be accessed from disk once a VBO already exists for a given time step.

The technique can be enhanced by also specifying threshold parameters that allow to configure for which data range of a given field points need to be rendered at all. Such a threshold allows visual clutter to be reduced significantly, as unimportant information can be suppressed. For instance, all data points outside the galactic regions in the demonstrated dataset can be visually omitted, same as data in the very dense centers, provided there is a data field that expresses such properties on the particles. Also, by scaling of the Gaussian splats via a given field, the splats may be smaller in dense regions than in thin regions, thus preserving details in the datasets. Which field is suitable for such setting is a question of arithmetic operations on the available data. Using the smoothing length of the SPH simulation itself appears to be a natural choice, but has not yielded optimal results so far. It might be an option to employ other smoothing kernels rather than Gaussian splats here, such as those used directly in the numerical simulations. Such explorations will be the subject of further investigation.



**Figure 18.** Multipatch volume, constructed from 13 distinct regular patches. Left image: depicting each point of the volume; centre left image: reduction to an equatorial slice; centre right image: scaling of point size by radial distance; right image: patch layout by coordinates (Zink *et al* 2007).

### 3.2. Multipatch data—rotating neutron star

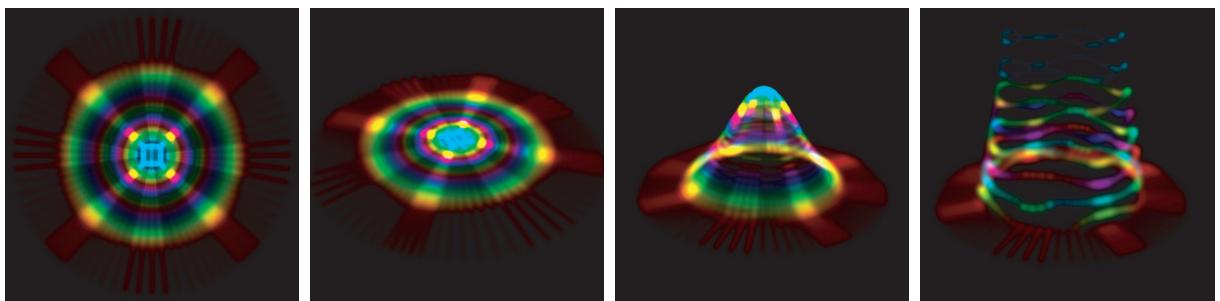
Many systems of interest in computational fluid dynamics as well as general relativistic astrophysics and other domains benefit from using non-Cartesian coordinates for computational grids. This allows continued use of topologically simple discretization schemes, i.e. regular (structured) meshes in the form of multidimensional arrays, thereby avoiding the complex topology of e.g. triangular or tetrahedral meshes.

It is however not the best choice to cover an entire domain with the same coordinate system, due to the need to avoid coordinate singularities and degenerated grid points (multiple vertices in the grid referring to the same point in physical space). In general relativity, a global coordinate system might not even exist at all.

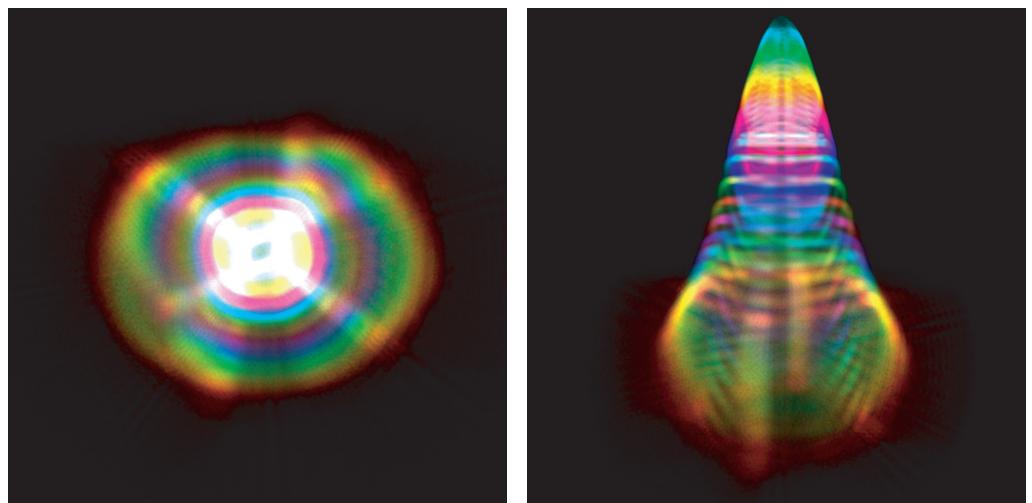
Spherical polar coordinates have singularities at the poles. A sphere cannot be covered by a single regular mesh without resulting in degenerated points. Addressing this issue requires at least two regular meshes, one for the northern and one for the southern hemisphere—with arbitrary overlap at the equator. Each of those meshes covering a coordinate domain is called a *patch*, and a dataset consisting of many of them is then called a *multipatch* dataset. While the topology of each patch is trivial, the global topology is no longer trivial, which troubles visualization algorithms.

In a 3D setup there may easily be much more than just two patches required, and here we demonstrate a visualization based on a dataset built from 13 separate patches, as described in (Zink *et al* 2007). This specific dataset describes a stationary rotating neutron star, resulting in 350 time steps of a single scalar field (density). About 300 000 data points are distributed over 13 patches with four blocks each (due to parallelization issues). Figure 18(left) depicts all points in the volume, which can be easily restrained to just a plane in world coordinates. In order to achieve a smooth appearance, the size of each Gaussian footprint needs to be related to the distance of a point to its neighbors; in this specific case, this distance is coarsely related to the radial coordinates. The regions of patches overlapping each other appear brighter. In the fiber bundle terminology, this is a 4D base space (provided in multiple coordinate systems and patches) with a 1D fiber space.

Restricting the rendering of points to a plane is efficiently done by graphics hardware and yields an impression similar to a 2D slice of the multipatch volume. However, no knowledge of the complex topology is required and this approach is very fast and thus suitable for rendering



**Figure 19.** Rendering of the vertex plane rendered as Gaussian point sprites (left), perspective rendering (centre left), geometry variation (centre right) and high sensitive colors and geometry variation (right).



**Figure 20.** Rendering of the entire multipatch volume, setting points in place (left) and imposing additional geometry variation (right).

of time-dependent data (4D base space). By shifting the render points in coordinate space, the appearance of a height field is achieved, as demonstrated in figure 19. Proper scaling enhances even small variations, showing tiny instabilities in the otherwise stationary rotating neutron star. This topology-independent point-wise rendering intrinsically morphs to volume rendering, as the thickness of the rendering domain is a free parameter. Imposing a geometrical shift on each data point may even work when applied to the full volumetric volume, as depicted in figure 20. At the moment of writing this paper, only one scalar field was available in the data, so the application of a 2D fiber space could not be demonstrated.

### 3.3. AMR—colliding black holes

Orbiting and coalescing binary black hole systems emit gravitational waves, which will be detected by gravitational wave observatories such as LIGO (LIGO 2008) and GEO600 (GEO600 2008). Both are coming on-line now. In the past 3 years, advances in numerical methods made it possible to simulate such systems numerically in a stable manner

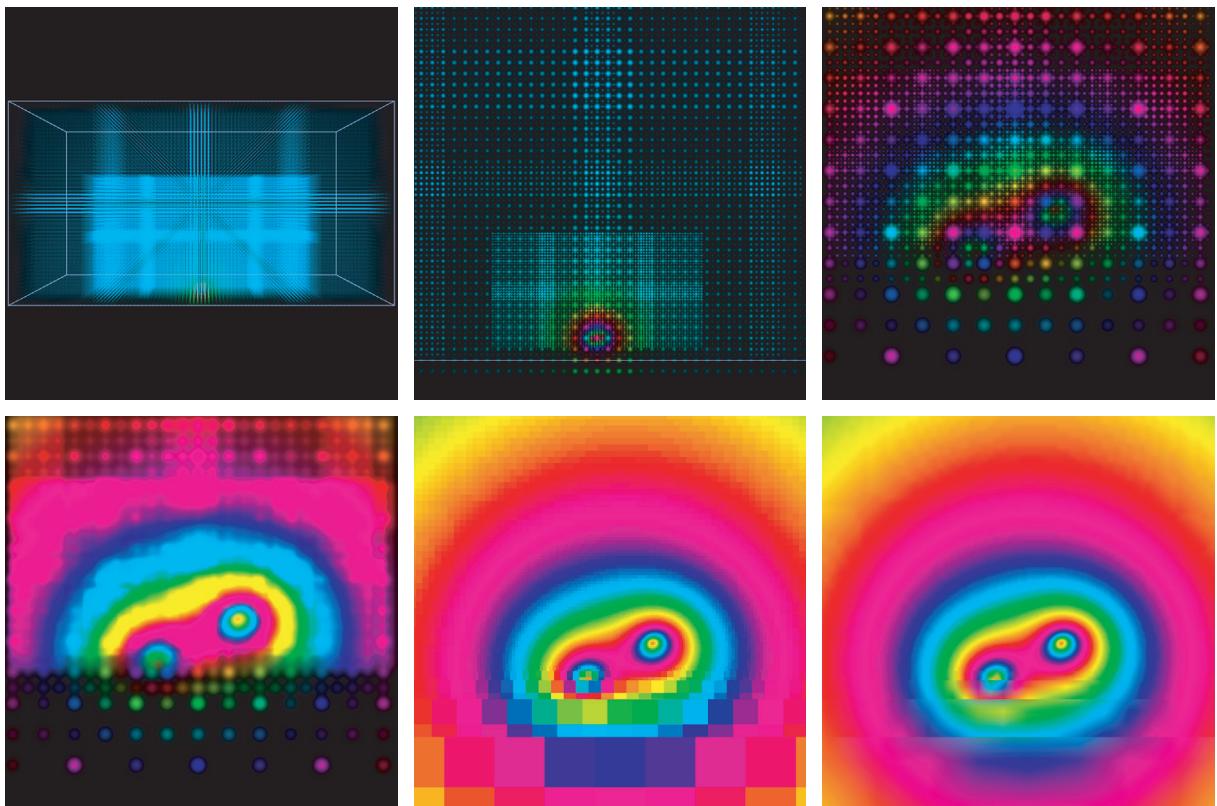
and with unprecedented accuracy (Baker *et al* 2006, Campanelli *et al* 2006, Diener *et al* 2006). One prototypical setup of such a binary system is *QC-0*: this setup performs approximately one orbit, then merges, and finally settles down to a single rotating black hole, emitting a burst of gravitational radiation in the process. Current state-of-the-art simulations of black hole binaries also include several orbits before merger, spinning black holes, spin/orbit interaction and recoil from asymmetric gravitational wave emission. These simulations were impossible without the AMR scheme; we demonstrate visualization of a *QC-0* dataset.

AMR is a technique to automatically adapt the resolution of structured grids to a given function by recursively overlaying fine grids over a coarser grid where necessary. The Berger–Oliger AMR algorithm (Berger and Oliger 1984) is a landmark achievement without which many numerical calculations would be plainly impossible. The so-called *refinement criteria* dictate where fine grids should be placed, and as these criteria change during a simulation, fine grids are moved, created, or destroyed. The main advantage of AMR over unstructured grids is that structured grids can be handled much more efficiently on current hardware, and that numerical stability may be easier to achieve. AMR was introduced to 3D calculations in numerical relativity e.g. in (Schnetter *et al* 2004), which also gives more details on the time stepping algorithm.

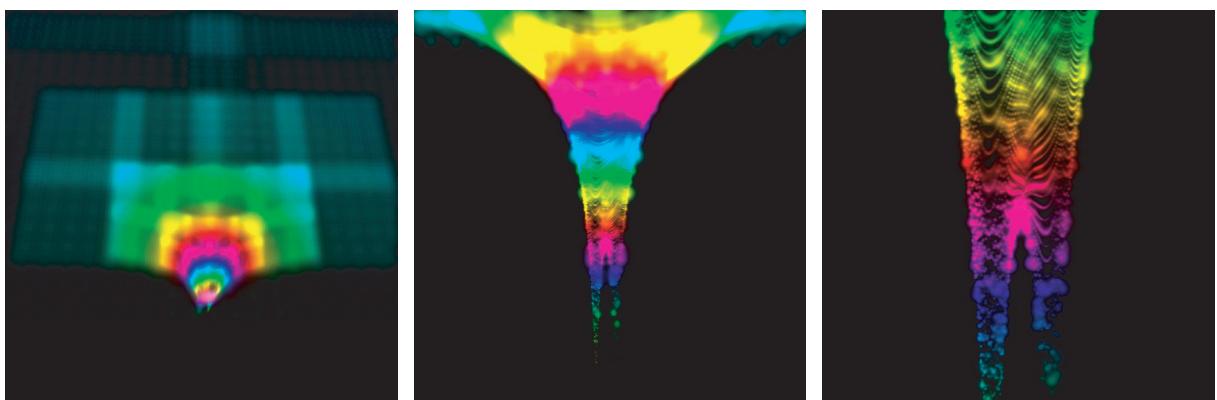
Numerical stability requirements make it necessary to timestep the solution on finer grids in smaller steps than on coarser grids, leading to a rather complex, recursive time evolution algorithm where not all grids exist at all times. This in turn complicates post-processing and visualization, which generally requires interpolation (in time) between successive coarse grid points if no fine grid points exist at a certain time. AMR was introduced to 3D calculations in numerical relativity, e.g. in (Schnetter *et al* 2004), which also gives more details on the time stepping algorithm.

While each structured grid by itself is topologically simple and comes with a linear mapping to coordinate space, the global structure is no longer trivial, which impedes efficient visualization algorithms such as texture-based volume rendering (Kähler 2005, Weber *et al* 2007). An AMR dataset may, however, always be treated as an unstructured grid and volume rendering of scalar fields may be achieved with Gaussian footprints about the size of the voxels of each refinement level. Reduction of the rendering domain to a thin plane yields results similar to texture-based rendering approaches (figure 21). With the point size being fixed in the case of rendering AMR datasets, geometric variation of point locations may still be used to enhance the properties of the dataset (figure 22).

The primary quantity in general relativity to consider is the metric tensor field. Here, we have to deal with a 4D base space (the spacetime) and a 10D fiber space (if no matter is involved) when the full 4D metric tensor is used, or a 6D fiber space when only the spatial part of the metric is investigated. However, the direct visualization of such tensor fields is not a common practice, firstly due to the practical availability of tensor field visualization tools to end-users, and secondly due to the unfamiliarity of possibly end-users with the complex visible result. Visualization of tensor fields is mostly investigated in medical imaging based on magneto-resonance data acquisition methods (Benger *et al* 2006). Such techniques tend to be highly domain-specific and do not necessarily transport over to astrophysics, but provide good source of inspiration, see Benger (2004) for an extensive discussion. Dealing with a base space in the layout AMR however, adds another hurdle, and early results of tensor field visualization on AMR are presented here for the first time.

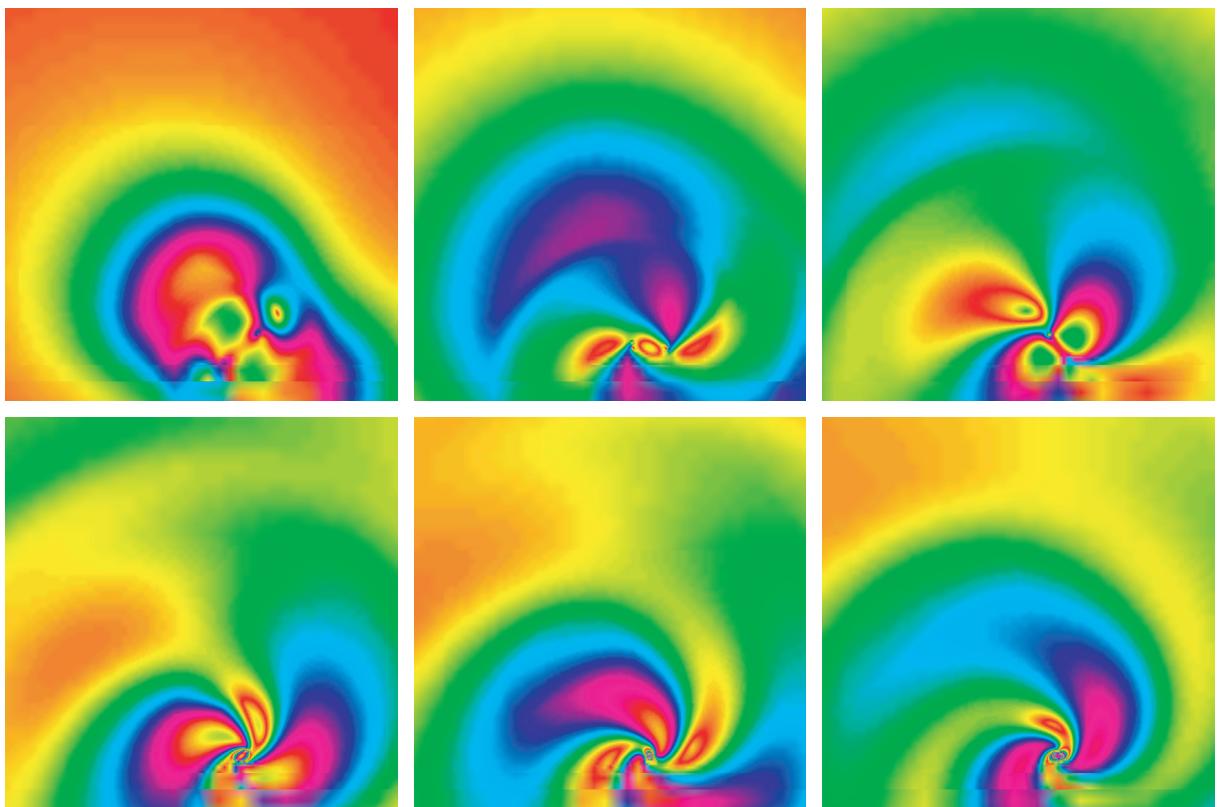


**Figure 21.** Rendering of a 3D AMR dataset: upper row: full volume, reduction to a 2D slice, detail depicting refinement levels; lower row: proper scaling of Gaussian splats, texture-based alternative without and with interpolation.



**Figure 22.** Geometric elevation of vertices within an AMR volume, based on the ‘lapse’ scalar function representing time.

It is common in numerical relativity to decompose the 4D metric tensor of the spacetime (10 independent quantities) into the lapse scalar  $\alpha$ , the shift vector  $\beta$  and a spatial metric tensor  $\gamma$ . In figures 21 and 22, we display the lapse function as a scalar field, as it represents the amount of time between two subsequent time steps of the simulation. This lapse function is an indication

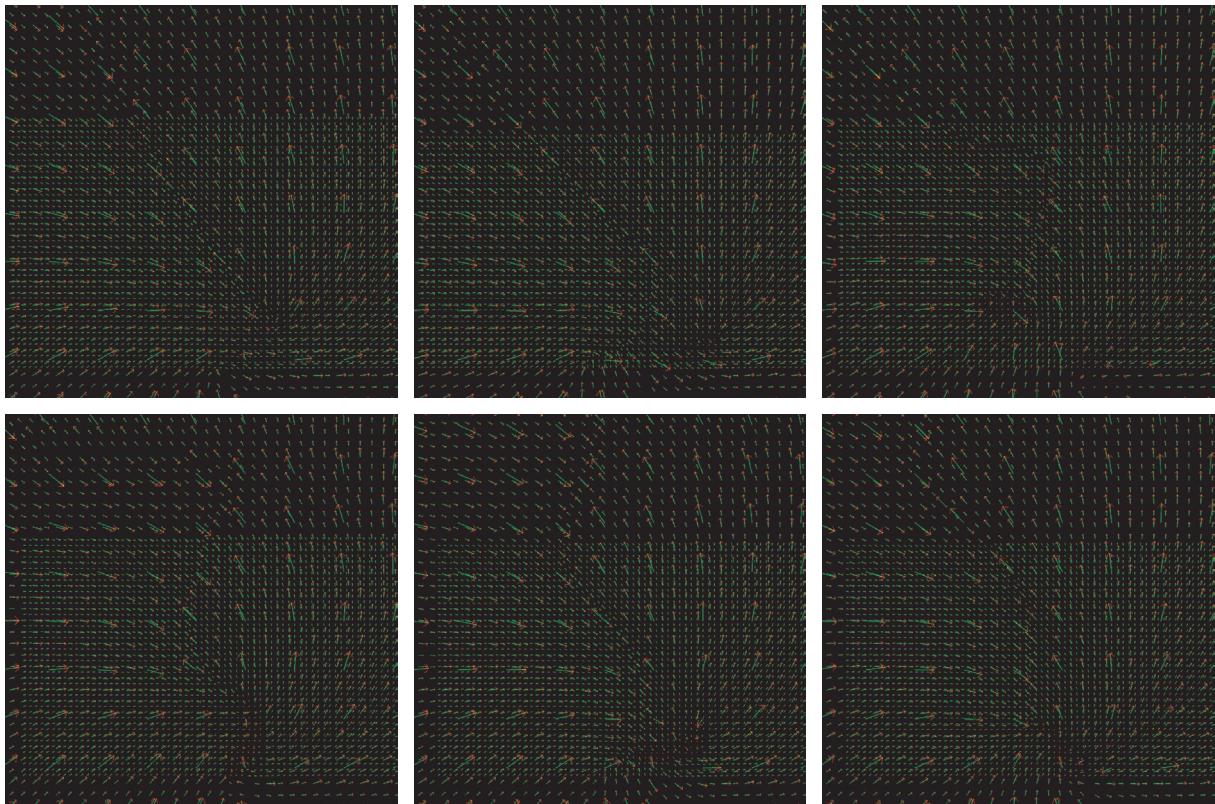


**Figure 23.** The linear shape factor of the metric tensor field of a black hole merger, as it depicts the creation and evolution of spiral structure indicating a gravitational wave.

of the gravitational potential. Gravitational redshift can be intuitively explained as light losing energy when escaping from a gravitational potential, but also as time progressing slower in regions of extreme gravity. In the extreme case of a black hole, time comes to an apparent hold at the horizons of the black holes, which is hereby represented by the lapse function dropping to zero—no advancement of the time coordinate at the location of the black hole horizons in the numerical grid.

The full complexity of the spacetime however requires inspection of the metric tensor field. Inspecting the components of the metric tensor as scalar fields is highly misleading, since they depend on the coordinate system, and are thus view-dependent (i.e. not necessarily of physical relevance by themselves). In order to apply scalar field visualization techniques, we need to extract certain invariant quantities. Westin *et al* (1997) introduced geometrical measures for the local properties of a tensor field, the so-called shape factors. They describe the relationship of the eigenvalues of a tensor field: the linear shape factor is large when one eigenvector is dominant, the planar shape factor indicates two dominant eigenvectors and the spherical shape factor is a measure for isotropic regions (all three eigenvectors of equal weight).

Figure 23 displays the linear shape factor—a concept originating from medical imaging—of the astrophysical dataset of a black hole merger. In contrast to visualizing the metric components or the lapse function, the creation and evolution of a spiral structure indicating a gravitational wave becomes evident immediately. The nature of the gravitational wave can

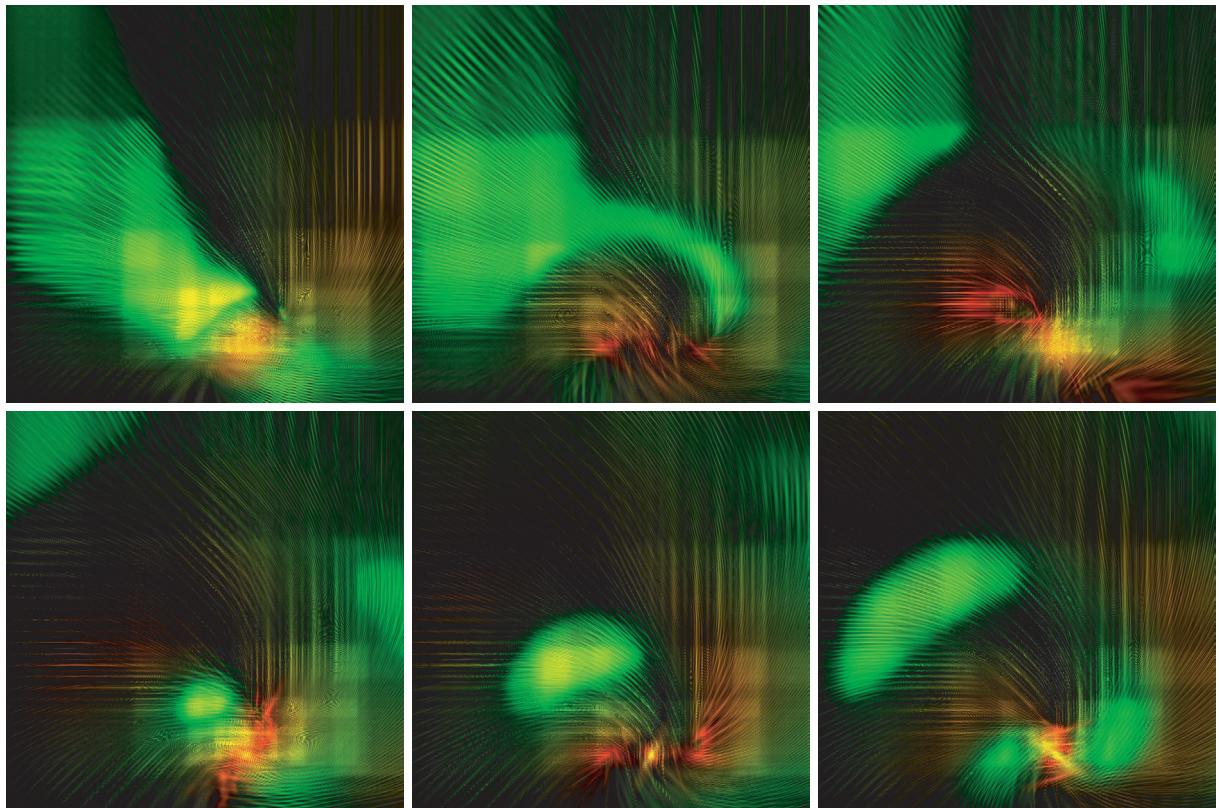


**Figure 24.** Eigenvectors of the metric tensor field, same time steps as in figure 23, shown here for illustrative purposes; three independent values given per point. However, rendering via vector arrow icons is not appropriate.

be interpreted as a region in space that is predominantly ‘stretched’ in one direction. While the linear shape factor indicates where such linear stretching happens, it does not depict in which direction it occurs. Such information is available from the major eigenvector of the metric tensor field, displayed in figure 24 using the standard approach of vector arrows at each point (as a visualization of a 3D fiber space). The visual results are not very impressive and shown here just for illustrative purposes. Actually of interest is to completely convey all properties of the metric tensor field, which also includes information about the second eigenvector and the shape factor information. These are depicted in figure 25 using the technique of tensor patterns. Linear regions (one dominant eigenvector) appear green, red regions indicate two dominant eigenvectors. The red/green structures are thus similar to figure 23, but here we also see the orientation of the eigenvectors. Visually, the two black holes appear to drag a coma of a planar region behind them and seem to plough through the spacetime, thereby emitting a gravitational wave of highly linear stretching. This visualization covers a 6D fiber space over a 4D base space.

#### 4. Data organization

In the previous sections, we were considering the datasets as fibers given on a base space, i.e. each point in base space has the same amount of data attached. An unstructured grid or particle set with fields implements this mathematical model most generically. However, it is

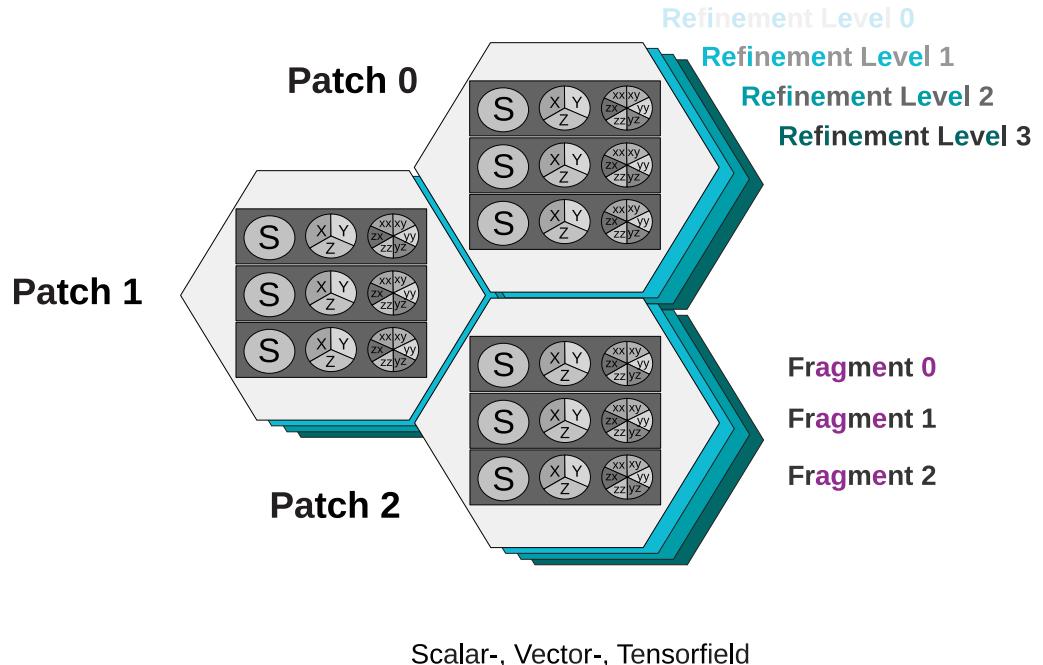


**Figure 25.** Metric tensor field, six independent quantities per point, rendered by the technique of tensor patterns. The two orbiting black holes appear to be ploughing through the spacetime.

not desirable to *convert* all structured data types into unstructured ones and lose the additional information thereby. Rather, we need to have the visualization algorithm operate directly on the given data, structured or unstructured, in whatever form they exist for a given case.

It is however not possible to employ generic programming here (Veldhuizen 1995) because the graphics hardware requires a specific layout of the data and the type of the dataset is only known at runtime when loading data from within an interactive application. Thus, in order to allow a generic visualization algorithm to operate *in situ* on the different data types, we need to impose a certain structure on the concept of a *field* over a set of points. We identified these four properties that the implementation of a field as a data structure needs to support:

- (i) Hierarchical ordering: for a certain point in space, there exist multiple data values, one for each *refinement* level. This property describes the topological structure of the base space.
- (ii) Multiple coordinate systems: one spatial point may have multiple data representations relating to different coordinate systems. This property describes the geometrical structure of the base space.
- (iii) Fragmentation: data stems from multiple sources, such as a distributed multiprocess simulation. The field then consists of multiple data blocks, each of them covering a subdomain of the field's base space. Such field fragments may also overlap.



**Figure 26.** Hierarchical structure of the data layout of the concept of a *field* in computer memory: (i) organization by multiple resolutions for the same spatial domain; (ii) multiple coordinate systems covering different spatial domains (arbitrary overlap possible); (iii) fragmentation of fields into blocks (recombination from parallel data sources); (iv) layout of compound fields as components for performance reasons, indicated as *S* (scalar field), {*x*, *y*, *z*} for vector fields and {*xx*, *xy*, *yy*, *yz*, *zz*, *zx*} for tensor fields.

- (iv) Separated compounds: a compound data type such as a vector or tensor, may be stored in different data layouts since applications have their own preferences. An array of tensors may also be stored as a tensor of arrays, e.g. XYZXYZXYZXYZ as XXXXYYYYZZZZ . This property describes the internal structure of the fiber space.

All these structure components are optional. In the most simple case, a field is just represented by an array of native data types; however, in the most general case (which the visualization algorithm must always support), the data are distributed over several such property elements and built from many arrays. With respect to quick transfer to the GPU, only the ability to handle multiple arrays per dataset is of relevance.

In our implementation, these data arrays describing the entity of a *field* are organized in a hierarchy of four levels, as illustrated in figure 26. The ordering of these levels is done merely based on their semantic importance, with the uppermost level (i) embracing multiple resolutions of the spatial domain being the most visible one to the end-user. Each of these resolution levels may come with different topological properties, but all arrays within the same resolution are required to be topologically compatible (i.e. share the same number of points). There might still be multiple coordinate representations required for each resolution, which constitutes the second hierarchy level, (ii) of multiple coordinate patches. Data per patch may well be distributed over various fragments (iii), which is considered an internal structure of each

patch, due to parallelization or numerical issues, but not fundamental to the physical setup. Last but not least fields of multiple components such as vector or tensor fields may be separated into distinct arrays themselves (iv). This property, merely a performance issue of in-memory data representation, is not what the end-user usually wants to be bothered with, and is thus set as the lowest level among these four entries.

## 5. Conclusion

In this paper, we have discussed a systematic approach on how to visualize multiple fields on a spatio-temporal domain, based on the properties of Gaussian footprints representing the various fields. Such properties can be color, size, or geometric displacement as well as texturing and additional structure of these footprints. This approach is intrinsically independent from the topological structure of the underlying mesh, and an appropriate data structure has been presented that allows specific meshes to be directly represented without the need to convert these datasets. The methods have been demonstrated upon recently generated datasets stemming from practical applications.

## Acknowledgments

I thank Georg Ritter, University of Innsbruck, and Imitiaz Hossain, LSU for their contributions to the paper; furthermore Dr Wolfgang Kapferer, University of Innsbruck, for providing the SPH dataset of the colliding galaxies; Dr Burkhard Zink, LSU, for providing the multipatch dataset of the neutron star; Dr Erik Schnetter, LSU, for providing the AMR dataset of the merging black holes and his contributions to this paper. This research also employed resources of the Center for Computation and Technology at LSU, which is supported by funding from the Louisiana Legislature's Information Technology Initiative. Portions of this work were supported by NSF/EPSCoR award no. EPS-0701491 (CyberTools).

## References

- Baker J G, Centrella J, Choi D I, Koppitz M and van Meter J 2006 *Phys. Rev. Lett.* **96** 111102  
 Bender W 2004 Visualization of general relativistic tensor fields via a fiber bundle data model *PhD Thesis* FU Berlin  
 Bender W and Hege H C 2004 *Conference on Visualization and Data Analysis 2004* (Proc. SPIE vol 5295) ed R F Erbacher, P Chen, J C Roberts, M Gröhn and K Börner (IS& T/SPIE Electronic Imaging Symposium in San Jose, CA) pp 151–62. Online at <http://www.zib.de/visual/publications/sources/VDA2004.pdf>  
 Bender W, Bartsch H, Hege H C, Kitzler H, Shumilina A and Werner A 2006 *Int. J. Neurosci.* **116** 461–514  
 Berger M J and Oliger J 1984 *J. Comput. Phys.* **53** 484–512  
 Berti G 2000 Generic software components for scientific computing *PhD Thesis* University of Cottbus. Online at <http://www.math.tu-cottbus.de/~berti/diss/>  
 Butler D M and Bryson S 1992 *Comput. Phys.* **6** 576–84  
 Butler D M and Pendley M H 1989 *Comput. Phys.* **3** 45–51  
 Campanelli M, Lousto C O, Marronetti P and Zlochower Y 2006 *Phys. Rev. Lett.* **96** 111101  
 Cook L and Matarazzo C 1998 *1998 Nuclear Explosives Development Conf., Las Vegas, NV*  
 Crawfis R A and Max N 1993a Texture splats for 3D scalar and vector visualization *Proc. IEEE Visualization 1993 (October 1993)* pp 91–8

- Crawfis R and Max N 1993b *Visualization '93, San Jose* ed G M Nielson and D Bergeron (Los Alamitos, CA: IEEE Computer Society Press) pp 261–6
- Diener P, Herrmann F, Pollney D, Schnetter E, Seidel E, Takahashi R, Thornburg J and Ventrella J 2006 *Phys. Rev. Lett.* **96** 121101 (arXiv:[gr-qc/0512108](#))
- GEO600 2008 GEO 600. Online at <http://www.geo600.uni-hannover.de/>
- Kähler R 2005 Accelerated volume rendering on structured adaptive meshes *PhD Thesis* Freie Universität Berlin, Fachbereich Mathematik und Informatik (P Deuflhard). Online at <http://www.zib.de/visual/publications/thesis/DissKaehler.pdf>
- Kapferer W, Knapp A, Schindler S, Kimeswenger S and van Kampen E 2005 Star formation rates and mass distributions in interacting galaxies arXiv:[astro-ph/0503559v1](#)
- Kaufman A and Mueller K 2004 Overview of volume rendering *The Visualization Handbook* ed C D Hansen and C Johnson (New York: Academic) Online at <http://www.cs.sunysb.edu/~mueller/papers/volvisOverview.pdf>
- Kindlmann G and Durkin J W 1998 Semi-automatic generation of transfer functions for direct volume rendering *Proc. of the 1998 IEEE Symp. on Volume Visualization (VVS '98)* pp 79–86 doi:[10.1145/288126.288167](#)
- Kitware 2005 *Visualization Toolkit*. Online at <http://www.kitware.org/> (Last accessed 25 April 2007)
- Kreylos O, Ma K L and Hamann B 2000 A multi-resolution interactive previewer for volumetric data on arbitrary meshes *Proc. Workshop on Computer Graphics and Virtual Reality, 2000 Int. Computer Symp. National Chung Cheng University, Chiayi, Taiwan* ed M Outhyoung and Z-C Shih pp 74–81
- Laidlaw D, Ahrens E, Kremers D, Avalos M J, Jacobs R E and Readhead C 1998 *Proc. Viz. '98* pp 127–34. Online at <http://www.cs.brown.edu/people/dhl/pdf/vis98dti.pdf>
- LIGO 2008. LIGO: Laser Interferometer Gravitational Wave Observatory. Online at <http://www.ligo.caltech.edu/>
- Schnetter E, Hawley S H and Hawke I 2004 *Class. Quantum Grav.* **21** 1465–88
- Springel V 2005 *Mon. Not. R. Astron. Soc.* **364** 1105–34. Online at <http://www.mpa-garching.mpg.de/gadget>
- Stalling D, Westerhoff M and Hege H C 2005 *Visualization Handbook* ed C R Johnson and C D Hansen (New York: Academic) Online at <http://www.amiravis.com>
- Teufel H J and Wehrhahn C 2000 *J. Opt. Soc. Am. A* **17** 994–1006
- The HDF5 Group 2008 *Hierarchical Data Format Version 5* Online at <http://hdf5.ncsa.uiuc.edu/hdf5/>
- Treinish L A 1997 *Data Explorer Data Model* Online at [http://www.research.ibm.com/people/l/lloyd/tm/dm/dx\\_dx\\_dm.htm](http://www.research.ibm.com/people/l/lloyd/tm/dm/dx_dx_dm.htm)
- Veldhuizen T L 1995 *C++ Rep.* **7**(4) 36–43 reprinted in Lippman S B (ed) 1998 *C++ Gems* (Cambridge: Cambridge University Press)
- Weber G H, Beckner V E, Childs H, Ligocki T J, Miller M C, Straalen B V and Bethel E W 2007 *4th High-End Visualization Workshop, Obergurgl, Tyrol, Austria, 18–21 June 2007* (Berlin: Lehmanns Media-LOB.de) pp 12–25
- Weber G H, Kreylos O, Ligocki T J, Shalf J M, Hagen H, Hamann B, Joy K I and Ma K L 2001 *Vision, Modeling, and Visualization 2001* (Berlin: Akademische Verlagsgesellschaft Aka GmbH) pp 121–8, 522
- Westin C, Peled S, Gudbjartsson H, Kikinis R and Jolesz F A 1997. *Proc. ISMRM, Fifth Meeting, Vancouver, Canada* p 1742. Online at <http://spl.bwh.harvard.edu:8000/pages/ppl/westin/papers/smr97/htmlversion.html>
- Westover L 1990 *Proc. 17th Annu. Conf. on Computer Graphics and Interactive Techniques* (New York: ACM Press) pp 367–76
- Zink B, Schnetter E and Tiglio M 2007 Multi-patch methods in general relativistic astrophysics - I. Hydrodynamical flows on fixed backgrounds arXiv:[0712.0353](#)