SmoothViz: An Interactive Visual Analysis System for SPH Data

Vladimir Molchanov
Jacobs University
Bremen, Germany
v.molchanov@jacobs-university.de

Alexey Fofonov
Jacobs University
Bremen, Germany
a.fofonov@jacobs-university.de

Stephan Rosswog
Stockholm University
Stockholm, Sweden
stephan.rosswog@astro.su.se

Paul Rosenthal
TU Chemnitz
Chemnitz, Germany
paul.rosenthal@informatik.tu-chemnitz.de

Lars Linsen
Jacobs University
Bremen, Germany
l.linsen@jacobs-university.de

***Abstract*—The paper describes the software tool SmoothViz designed for visual representation and interactive analysis of data coming from SPH simulations with astrophysics as the driving application. Its main characteristics can be summarized as follows: (1) All visualization methods operate directly on particle data, i.e., any kind of re-sampling or regularization of the original data is avoided. (2) The system supports the analysis of individual scalar and vector fields as well as a simultaneous analysis of multiple fields. (3) The system works with time-varying volumetric physical spaces and respective attribute spaces, where each attribute refers to a field. (4) Interactive analysis is supported by the use of multiple coordinated views, i.e., a number of views of different properties of the data are provided. When the user interacts with one view (e.g., by making a selection) the interactions are immediately reflected in all other views (e.g., by highlighting the selection in all views).**

I. Introduction

Since its invention by Gingold and Monaghan [1] and Lucy [2], SPH simulations produce datasets of permanently increasing size. They involve millions of particles and operate with tens of scalar and vector fields. Data may include thousands consequent time steps of a simulation. Thus, there is a clear need for systems assisting the researchers in exploring such complex datasets. Visualization methods provide intuitive means for data representation, which form a basis for interactive data exploration and analysis. We introduce our system SmoothViz which is targeted at interactive visual analysis of unstructured multi-variate time-dependent volumetric data.

Interactivity when running SmoothViz on an off-the-shelf PC is achieved by means of multi-processor (OpenMP) and GPGPU (CUDA) implementations of the main algorithms, including isosurface extraction, stream- and pathlines tracking, scatterplot generation, etc. The user has access to the information at different levels of abstraction (e.g., particle parameters or characteristics of selected groups) in a textual or visual form. The first main view of the graphical user interface (GUI) of SmoothViz (implemented in Qt) is a 3D physical space visualization widget (implemented in OpenGL) allowing the user to explore individual scalar and vector fields, the range of influence of individual particles, etc. To analyze the multidimensional attribute space of multi-field data, a linear projection method is used in the frame of a star-coordinate widget, which allows the user to understand the attribute values of the particles. It facilitates the intuitive detection of groups of particles with similar characteristics and behavior.

Designed for visual analysis and representation of SPH astrophysical data, SmoothViz can be applied to SPH data from other fields and even to non-SPH point-based unstructured data. In the latter case, Moving Least Squares (MLS) technique is used instead of the standard SPH approximation formula. MLS is able to approximate both values of the sampled field and its derivatives of arbitrary order [3]–[6] with prescribed smoothness degree.

II. Related Work

Several visualization tools exist to work with SPH data, e.g., SPLASH [7], which serves to visualize one-, two- and three-dimensional SPH results consistently with the basic SPH method. SmoothViz was first introduced by Linsen et al. [8] and has been incrementally extended to include state-of-the-art visualization algorithms and developed according to the user feedbacks.

An SPH vector field visualization technique based on a predictor-corrector scheme was proposed by Schindler et al. [9]. The authors were able to extract smooth vortex core lines using the native SPH interpolation procedure.

One of the most popular scalar field visualization approaches is the generation of isosurfaces, i.e., manifolds at which the considered scalar field takes a given value. Isosurface extraction from unstructured point-based volume data was in the focus of papers by Rosenthal and Linsen [10], [11]. These approaches are incorporated in the SmoothViz system.

Since the simulated data typically contain a multitude of physical variables, their interplay is an important part of data analysis. A simple but efficient approach is to investigate the pairwise distribution of fields values using scatterplots and scatterplots matrices [12].

The concept of continuous scatterplots was developed in the series of works by Bachthaler, Heinrich, and Weiskopf [13]–[15]. Their approach eliminates the contradiction between discrete scatterplot representation of fields that are naturally continuous in the object space. The most recent generalizations of continuous representations of attribute spaces to unstructured data, logarithmic scatterplots and linearly projected space were proposed by Molchanov et al. [16].

Dimension reduction approaches like Principal Component Analysis (PCA) [17] or Multidimensional Scaling approaches (MDS) [18] are commonly used for mapping high-dimensional data to a lower-dimensional visual space. Linear projections are often preferred in order to not insert too much distortion and keep computation costs low. It is usually impossible to find one linear projection which would keep all clusters (groups of similar attribute values) separated and show outliers (samples with extreme attribute settings) appropriately. Moreover, it is often useful to apply projection to different subsets of attribute fields and investigate the sensitivity of the projection to a certain parameter. Therefore, it is desirable to have interactive means to analyze the multidimensional attribute space.

SmoothViz provides such means in form of an intuitive star-coordinates interaction widget. Star coordinates [19], [20] arrange the coordinate axes of a multidimensional space on a circle placed on a two-dimensional plane with axes having their origin at the center of the circle and an arrangement exposing equal angles between adjacent axes. A multidimensional point is mapped to a two-dimensional point by summing the unit vectors of each coordinate multiplied with the coordinate of the multidimensional point. Teoh and Ma [21] extended this concept by allowing for interactive change of the coordinate axes (both length and direction) but keeping them being rooted in the origin. Our interaction widget follows the same idea.

A multi-dimensional clustering-based visualization algorithm for multi-field particle volume data was proposed by Linsen et al. [5], [22]. The approach uses an automated multi-dimensional hierarchical density-based clustering method to identify clusters in attribute space and then extracts an enclosing surface of the respective particles in object space.

III. Data Handling

Currently, an internal data format is used in SmoothViz to store and read back particle-based data. A routine to read a specific data format can be easily included to the system upon request.

There are several interpolation techniques for the reconstruction of scalar and vector fields in the entire physical space available for the user. The basis of the SPH interpolation is a "summation interpolant" formula allowing us to approximate the value of a function $f(\mathbf{x})$ sampled at particle positions $\mathbf{x}_i$ at arbitrary spatial location $\mathbf{x}$. The idea is to represent $f(\mathbf{x})$ locally as a superposition of overlapping (spherical) kernels W_i centered at $\mathbf{x}_i$. The radii of these kernels h_i (*smoothing lengths*) are variable for each particle. Then,

$$f(x) \approx \sum_i f_i \frac{m_i}{\rho_i} W_i(\mathbf{x}-\mathbf{x}_i),$$

where m_i, ρ_i, and f_i are mass, density, and function value of the particle with index i, respectively. A normalized version of the result can be derived by dividing it by the interpolation of unity, given by

$$1 \approx \sum_i \frac{m_i}{\rho_i} W_i(\mathbf{x}-\mathbf{x}_i).$$

The most widely used of all SPH kernel functions W_i are the Schoenberg B-splines [23], in particular, the cubic spline $W_i(t) = s(t/h_i)/h_i^d$, where d is the spatial dimensionality and

$$s(q) = C \begin{cases} (2-q)^3/4-(1-q)^3, & \text{if } 0 \le q < 1, \\ (2-q)^3/4, & \text{if } 1 \le q < 2, \\ 0, & \text{if } 2 \le q, \end{cases}$$

where constant C depends on d.

For the formulae above to be applicable, masses, densities, and smoothing lengths need to be known for all particles. Otherwise, we handle unstructured point-based data in a general set-up by applying MLS data interpolation techniques. The MLS method finds its applications in volume ray casting [4], adaptive integration of scanned data [3], statistical implicit representation [6], and in various surface extraction and reconstruction algorithms, e.g., [5], [10]. The interpolated value $f(\mathbf{y})$ at any position $\mathbf{y}$ is found as the solution of the minimization problem

$$f(\mathbf{y}) = \sum_k c_k g_k(\mathbf{y}),$$
$$\{c_k\} = \operatorname{argmin} \sum_i \omega(\mathbf{y}-\mathbf{x}_i, h)\left(f(\mathbf{x}_i)-f_i\right)^2,$$

where g_k denote some basis functions (e.g., polynomials) and ω is a smooth weighting function having support size h.

IV. Physical Space

Various visual representations of data in physical (object) space are available in SmoothViz, among them: Visualization of slices through the volume, isosurface extraction, direct volume rendering, and particle rendering as color-mapped points. The user interaction includes a camera control (rotation, translation, zooming in and out) as well as selections of a single particle or particle groups. Moreover, an interaction range of an interactively selected particle $\mathbf{p}$ can be shown as a semi-transparent sphere centered at $\mathbf{p}$ with the radius equal to the smoothing length. Values of all scalar and vector fields of $\mathbf{p}$ can be provided in text form in an auxiliary window, see Fig. 1.

If several time steps are loaded, it is possible to navigate in time using a simple widget similar to a standard media player. When navigating, contents of all windows are updated simultaneously such that one always has a consistent view on the currently investigated time step. Selections made at one time step are tracked and kept active over all time steps.

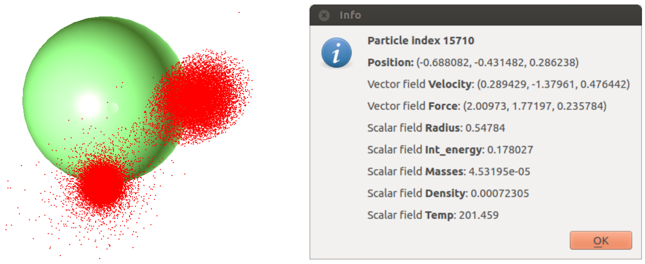


Fig. 1: Range of influence of a selected particle is shown as a sphere with respective radius (left). Fields values at the particle are given in a text form (right).

A. Isosurface Extraction

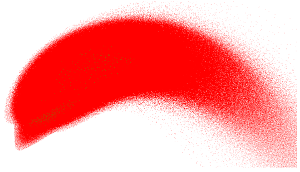
Every point in time of an SPH simulation can be interpreted as a set a of trivariate scalar and vector fields. For any scalar field $f(\mathbf{x})$ and any given real number f_0, the isosurface to f with respect to f_0 is defined as the set of all points $\mathbf{x}$ in space with function values $f(\mathbf{x}) = f_0$. However, explicitly and exactly extracting the isosurface from an SPH scalar field can be seen as impossible. Typical approaches for extracting isosurfaces rely on re-sampling the scalar field onto a regular grid and using standard isosurface extraction techniques. This procedure always introduces interpolation errors which can grow enormously in particular when operating on typical data sets where point densities vary by two or more orders of magnitude.

We rely on extracting the isosurface directly by using the values of the scalar field at the SPH particle positions [10]. Isopoints, i.e. points on the isosurface, are computed by linear interpolation between pairs of particles which are close together and whose function values are respectively below and above the isovalue. An optimal choice for these neighbors would be to use natural neighbors induced by the Voronoi diagram of particle positions. Since the exact detection of these neighbors is computationally intense, we decided to approximate them using a spatial decomposition. A three-dimensional kd-tree is built upon the particle positions and natural neighbors are approximated by evaluating neighbor relations of kd-tree cells.

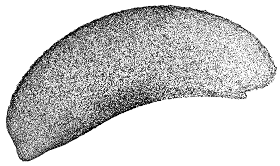
All of these processing steps are very efficiently implemented and can be carried out within seconds even for datasets with several million particles [11]. Furthermore, the actual extraction of isopoints is parallel in nature and can be significantly sped up on multiprocessor systems.

B. Isosurface Representation

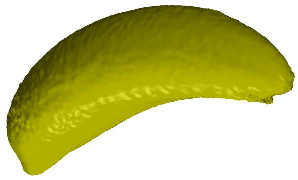
Since only isopoints can be extracted from the unstructured SPH data, the resulting isosurface is represented as a point cloud. Satisfactory renderings of such point clouds require at least the allocation of isosurface normals for a decent shading of points. Here we make use of the fact, that the normal's orientation can be derived from the linear interpolation of each isopoint. More precisely, we save in addition to each isopoint the vector pointing in the direction of the particle with the higher function value. In a post-processing step, the surface normal for each isopoint is approximated by a least-squares method on basis of the surrounding isopoints and oriented according to the stored orientation. The resulting point cloud with surface normals can be displayed using point-cloud rendering techniques [24].



(a) data points



(b) point-cloud isosurface



(c) splatting of isosurface

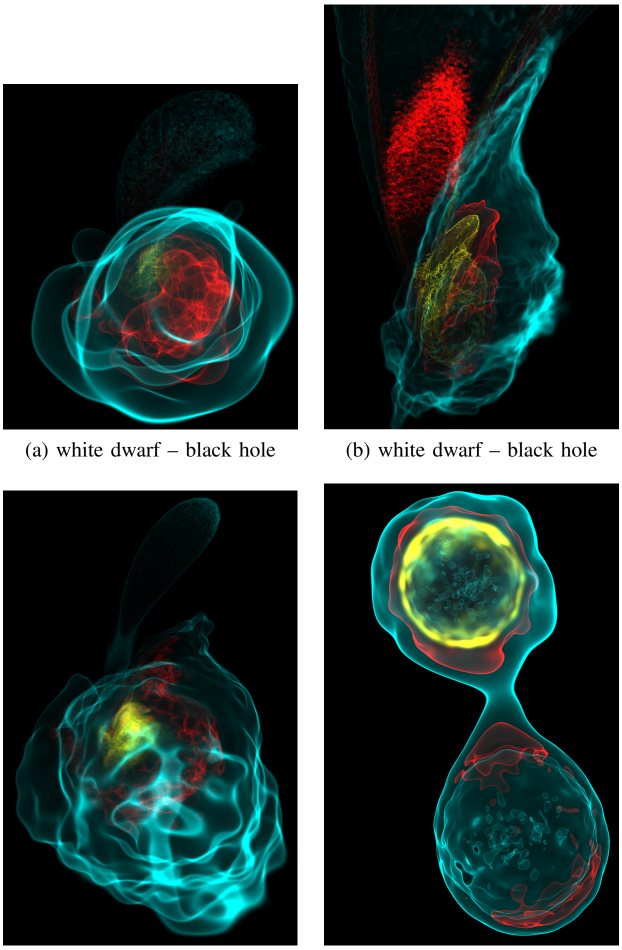
Fig. 2: Extraction of isosurface from SPH data shown in (a). Isosurface is represented as a point-cloud (b) and by means of splatting technique (c).

In Fig. 2 (a) we show the rendering of 7M particles from an SPH data set, color-coded with respect to density, out of a simulation of a white dwarf – black hole encounter. The rendering of the plain isopoints is shown in Fig. 2 (b). Especially for small data sets, the render quality can be unsatisfactory and holes in the surface can remain. This is circumvented when using circular discs (splats) as rendering primitives instead of plain points. The discs can be either rendered as plain OpenGL primitives or using a ray-tracing technique [25]. A rendering of the same isosurface as in Fig. 2 (b) is shown in Fig. 2 (c) as a ray tracing of splats.

C. Direct Volume Rendering

Sometimes it might be beneficial to not geometrically extract isosurfaces to save memory and bandwidth. This can be achieved by using ray casting techniques, shooting a ray through each pixel of the final rendering and detecting hits of the isosurface. Each ray is sampled and for each sample location the value of the scalar field is evaluated. Whenever the value changes from below or above the isovalue to the other side between two sample positions, the isosurface was hit by the ray and the shaded surface color is reported back to the screen. Using this technique, similar renderings like in Fig. 2 (c) are possible without actually extracting any isosurface explicitly. On the downside, such direct volume rendering techniques are view-dependent and the entire ray casting step has to be re-computed each time the viewing position changes.

In general, a direct volume rendering approach allows to



(a) white dwarf – black hole (b) white dwarf – black hole

(c) white dwarf – black hole (d) white dwarf binary

Fig. 3: Direct volume renderings of SPH data.

display not only isosurfaces but whole volumes of interest. Assume that the user is interested in special intervals of values of the scalar field and wants to see the distribution of the respective areas of particles. The intervals can be color-coded and in addition provided with a transparency value, where all uninteresting values get maximum transparency. Such an assignment is called a transfer function. When we now shoot a ray through a pixel, we are not checking the sample values against an isovalue, but we transfer each samples value through the transfer function into a color and transparency value. Those colors are integrated along the ray with respect to the transparency resulting in a color to display in the pixel. We show some sample renderings in Fig. 3 (4M white dwarf black whole, 40k double system).

D. Vector Field Visualization

Vector fields are represented as a set of streamlines starting at selected or randomly chosen particles. The length of streamlines is controlled by the user. For time-varying data, pathlines represent the trajectories of particles during the evolution.

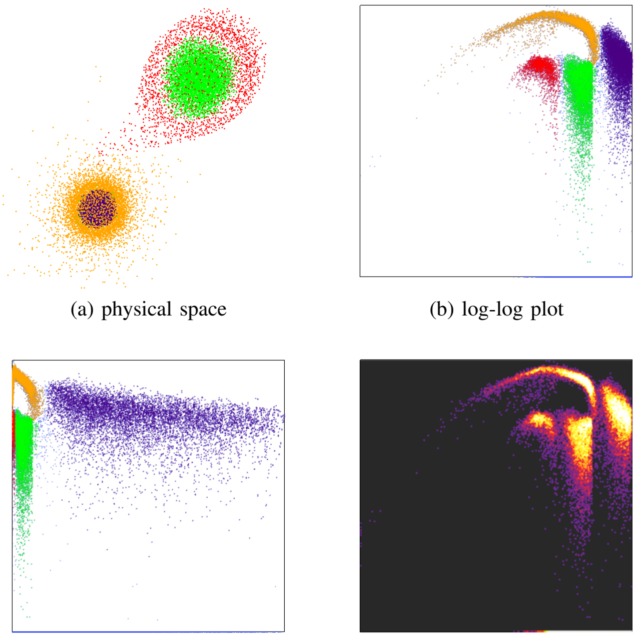
V. Attribute Space

Exploration of the interplay of scalar field distributions (also called *attribute spaces*) is an important part of data analysis.

A. Scatterplot

Joined distribution of two given scalar fields can be efficiently visualized in a scatterplot. Each sample is represented as a point with horizontal and vertical coordinates equal to the value of the first and the second attribute (or field), correspondingly. Logarithmically scaled plots are very useful for data exploration, since physical real-world or simulated scalar fields often have logarithmically distributed values. The use of logarithmic scatterplots creates a layout with more evenly distributed samples and simplifies the identification and selection of clusters, i.e., groups of particles with similar attribute values within the group and dissimilar attribute values to particles outside the group, see Fig. 4. The datasets presented in this paper are taken from Dan et al. [26].

Due to the finite resolution of the screen, many samples may be mapped to the same pixel, which may lead to misinterpretation of the resulting plot. To distinguish between pixels with low and large number of accumulated points, a transfer



(a) physical space (b) log-log plot

(c) lin-log plot (d) log-log plot

Fig. 4: Comparison of linearly and logarithmically scaled scatterplots. The dataset shown in object space (a) represents a merger of a binary system. Scatterplots (b)–(d) have dimensions density (horizontal) and temperature (vertical). The group of points marked as red can be easily identified and selected on the logarithmically scaled plot (b), whereas it is indistinguishable from the green group in linearly scaled scatterplot (c). Transfer function highlights sample density in (d).

Fig. 5: An interactive widget for transfer function manipulation. The color bar represents a range of the selected scalar field. User-defined colors are assigned to a number of moving nodes and linearly interpolated in-between.

function can be applied, see Fig. 4 (d). The transfer function selection widget is implemented as a variable number of nodes on a color bar representing the full range of considered values. The nodes are assigned with colors. Nodes can be added or removed, may interactively change their positions, and may get any new user-specified colors. Colors are interpolated linearly in hue between the nodes, see Fig. 5.

A system of SPH particles provide a discretization of a modeled object in physical space. The fields sampled at particle positions usually serve to provide discrete information about continuous simulated phenomena. Because of the continuity assumption, it is common practice to reconstruct a continuous field from the values at the discrete samples for visualization purposes. Such reconstruction is done using a radial SPH kernel in the physical space resulting in a smooth approximation of the field. A similar idea is applicable to the attribute spaces producing continuous scatterplots.

In classical scatterplot approach, each sample $\mathbf{p}$ in the object space is mapped to the scatterplot according to its attribute values and appears there as a single point. Since the value (and higher derivatives) of a field at the sample determine the field behavior in a small vicinity of $\mathbf{p}$, it is possible to map the sample together with this neighborhood U to the scatterplot. The resulting image is called a *footprint*. The geometry of the footprint can be computed. When restricting to the first-order local approximation of scalar fields in a spherical U, the footprint takes an elliptical shape. To agree with the SPH approximation method, it is natural to take the radius of U to be equal to the local smoothing length and weight it with the SPH interpolation kernel. Then, the resulting continuous scatterplot results after blending of elliptical footprints of all particles from the dataset. It is possible to construct continuous scatterplots which are scaled logarithmically, see Fig. 6.

B. Projected View

Scatterplots and scatterplot matrices can be efficiently used if the number of attributes is moderate. For n scalar fields, there are $(n^2-n)/2$ different scatterplots, which are hard to observe for large n. Since SPH data are usually time-varying multi-field data, other approaches are needed, e.g., parallel coordinates or dimension reduction methods. The use of parallel coordinates has some difficulties related to the ordering of axes and representation of samples as polylines, which are non-intuitive to compare. For our system, we developed projected view widgets to handle multi-attribute data.

For the class of linear maps, we offer the user the possibility to interactively define the parameters of a projection from the multidimensional attribute space to a 2D visual domain. The current state of the projection matrix is graphically encoded in a star-coordinate widget. A disadvantage of such a data representation is that the mapping is not invertible in general, i.e., it is not possible to uniquely determine all attributes of the sample based on its image in the visual domain. However, provided the user interaction is supported, it is possible to detect meaningful groups of samples, which are not visible otherwise. Finally, usual scatterplots are just a particular case of a linear projection, in which only two dimensions are active. Deactivation of a certain attribute in our star-coordinate representation is done by placing the end-point of the respective axis at the origin. Then, the respective column of the projection matrix has only zero entries and the disabled attribute field does not affect the final projection.

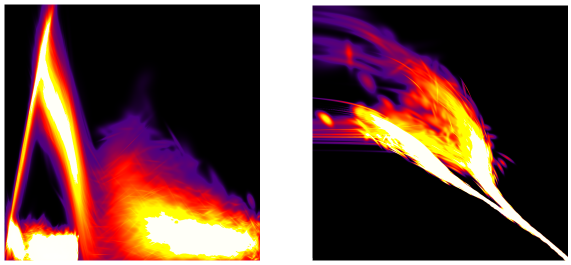


Fig. 6: Continuous representation of scatterplots. For two white dwarfs datasets, linear (internal energy vs. temperature fields) and logarithmic (internal energy vs. smoothing length) scatterplots are shown in the left and right figure, correspondingly.

We provide an option to generate a continuous representation of projected attribute spaces (CoRPAS). Similar to continuous scatterplots, footprints of samples are ellipses with geometry defined by values and first derivatives of all active fields at the samples. In order to eliminate the impact of background on the resulting plot, various weighting methods are proposed and discussed in [16]. The key idea is to scale the intensity of the footprints by an arbitrary non-negative function vanishing outside the modeled object. This prevents the footprints of the particles with extremely large smoothing length to appear as the main contributors to CoRPAS. The use of CoRPAS with different weighting schemes is illustrated in the subsequent example.

We explore an SPH simulation of a binary system, see Fig. 7. Two white dwarfs are bound together by gravity. The donor star gradually loses its mass, which flows to the heaviest white dwarf. The delivered material is heated when reaching the surface of the accreting star. The system consists of $40k$ particles with a number of scalar fields, including radii and masses of particles, temperature, density, internal energy, and chemical components. For the projection in Fig. 7 (a), we constructed a discrete plot (point size is 4 pixels) and several continuous representations weighted differently. They guided us when selecting five groups of particles as shown in Fig. 7 (g). The corresponding configuration in physical domain is presented in Fig. 7 (h): The detected groups of samples represent the disrupting star (green), the particles being transferred (blue), the heated particles that reached the surface of the accretor (red), the surface of the heaviest white dwarf (orange), and its core (yellow).

A clear advantage of continuous plots when compared to the discrete one is that the orientations of splats follow the paths of particles, i.e., they are aligned with an imaginary path over green-blue-red-orange groups. Thus, the CoRPAS reflect the underlying simulated process. In addition, it is now possible to see a border between the green and the orange groups, which is not visible on the discrete plot. The temperature-weighted plot helps to distinguish between the orange and the yellow clusters. Different weighting allows the user to draw conclusions about properties of particles in each group. The typically used volume weights cannot reveal all the groups because of the issue discussed in the previous section. Note that the relatively small blue group is dominant in Fig. 7 (c), since the particles building the envelopes of the stars have large radii. The particles that reached the second star (red group) are heated and therefore have bright footprints in (f).

VI. Linked Views

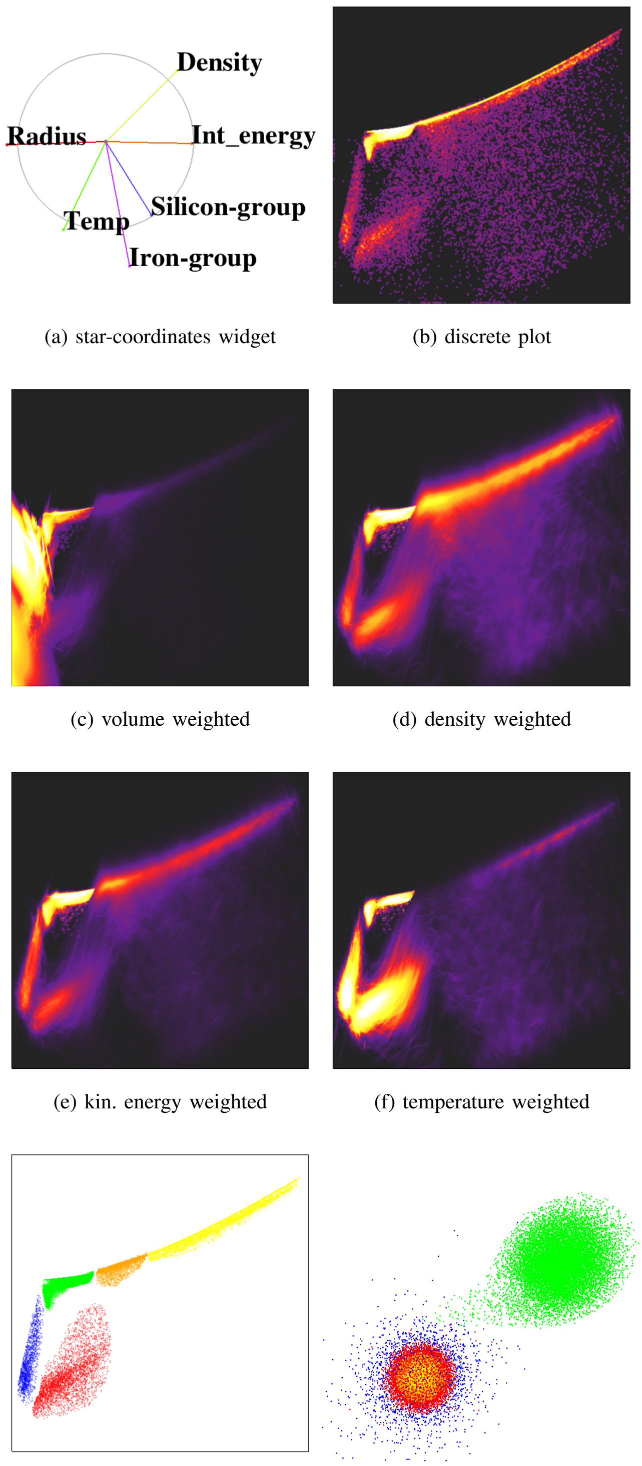
User interactions on a certain widget are not only tracked when navigating forward or backward in time within the same simulation, but also affect the current data representations in all other related widgets. For instance, selection of a group of particles and assigning a color to the selected group in a scatterplot leads to marking the same particles with this color in the object space and in the projected views.

It is possible to store the contents of all views into files by selecting a corresponding item in the menu bar. The user-defined settings as well as the current status of the open project can also be saved into and loaded from a file.

VII. Conclusion and Future Work

SmoothViz, presented in this paper, is an interactive tool for visual analysis and representation of unstructured (or particle-based) volumetric data with a focus on astrophysical SPH simulation outputs. The user can choose between the usage of SPH or MLS approximation schemes. The former requires mass, density, and smoothing length of all particles to be provided in the dataset. The latter extends the applicability of the system to arbitrary point-based volumetric data.

Compared to SPLASH, SmoothViz is currently limited to producing standard appropriately annotated screenshots for the use in research papers as it uses OpenGL instead of graphics libraries such as PGPLOT. However, this is balanced by the speed-up we get for handling complex 3D graphics and will definitely be improved in the later release of the system. Besides the speed-up, an OpenGL-based system offers a much better tool for user interaction, which is extensively exploited in SmoothViz, in particular, for the creation multiple linked views. Thus, SmoothViz is currently more suited for interactive visual data analysis.



(a) star-coordinates widget
(b) discrete plot
(c) volume weighted
(d) density weighted
(e) kin. energy weighted
(f) temperature weighted
(g) selected groups
(h) physical space

Fig. 7: Projections of two white dwarfs dataset. Orientation of splats in CoRPAS reflect the actual evolution of particles on their way between two stars. It also allows to visually separate particles corresponding to the cores of different stars (green and orange groups in (g)). Different weighting helps to analyze the properties of particles in each group. Note that the volume-weighted CoRPAS (c) is not helpful for the analysis since it is strongly affected by outliers.

Future work will be directed to speed-up the algorithms, which are not yet implemented on GPU, and to extend the toolbox for vector data representation. Since no unified SPH data format exists to our best knowledge, we would like to adapt the data reading procedures for the SPH output forms which are used most. As SmoothViz is mainly developed under Ubuntu and tested on Windows machines, the goal is to maximally extend the list of compatible platforms including Mac OS and different Linux derivatives.

ACKNOWLEDGMENT

This research was funded by DFG via grant LI 1530/6-2. The authors would like to thank Marius Dan for productive discussions and providing us datasets.

REFERENCES

[1] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics — theory and application to non-spherical stars," *Mon. Not. Roy. Astron. Soc.*, vol. 181, pp. 375–389, November 1977. [Online]. Available: http://adsabs.harvard.edu/cgi-bin/nph-bib_query?bibcode=1977MNRAS.181..375G

[2] L. B. Lucy, "A numerical approach to the testing of the fission hypothesis," *The Astronomical Journal*, vol. 82, pp. 1013–1024, Dec. 1977.

[3] V. Fiorin, P. Cignoni, and R. Scopigno, "Practical and robust mls-based integration of scanned data," in *Sixth Eurographics Italian Chapter Conference*, 2008, pp. 57–64. [Online]. Available: http://vcg.isti.cnr.it/Publications/2008/FCS08

[4] C. Ledergerber, G. Guennebaud, M. Meyer, M. Bacher, and H. Pfister, "Volume MLS ray casting," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1372–1379, 2008.

[5] L. Linsen, T. V. Long, P. Rosenthal, and S. Rosswog, "Surface extraction from multi-field particle volume data using multi-dimensional cluster visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1483–1490, 2008.

[6] C. Öztireli, G. Guennebaud, and M. Gross, "Feature preserving point set surfaces based on non-linear kernel regression," *Computer Graphics Forum*, vol. 28, no. 2, 2009. [Online]. Available: http://vcg.isti.cnr.it/Publications/2009/OGG09

[7] D. Price, "SPLASH: An interactive visualisation tool for smoothed particle hydrodynamics simulations," *Publications of the Astronomical Society of Australia*, vol. 24, pp. 159–173, 2007.

[8] L. Linsen, V. Molchanov, P. Dobrev, S. Rosswog, P. Rosenthal, and T. V. Long, "Smoothviz: Visualization of smoothed particle hydrodynamics data," in *Hydrodynamics — Optimizing Methods and Tools*, H. E. Schulz, A. L. A. S. es, and R. J. Lobosco, Eds. InTech, October 2011, pp. 3–28.

[9] B. Schindler, R. Fuchs, J. Biddiscombe, and R. Peikert, "Predictor-corrector schemes for visualization of smoothed particle hydrodynamics data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1243–1250, 2009.

[10] P. Rosenthal and L. Linsen, "Direct isosurface extraction from scattered volume data," in *Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization*, 2006, pp. 99–106.

[11] ——, "Smooth surface extraction from unstructured point-based volume data using PDEs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1531–1546, 2008.

[12] N. Elmqvist, P. Dragicevic, and J.-D. Fekete, "Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, pp. 1141–1148, 2008.

[13] S. Bachthaler and D. Weiskopf, "Continuous scatterplots," *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization / Information Visualization 2008)*, vol. 14, no. 6, pp. 1428–1435, Dec. 2008.

[14] ——, "Efficient and adaptive rendering of 2-d continuous scatterplots," *Computer Graphics Forum (Proc. Eurovis 09)*, vol. 28, no. 3, pp. 743 – 750, 2009.

[15] J. Heinrich and D. Weiskopf, "Continuous parallel coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1531–1538, nov 2009. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2009.131

[16] V. Molchanov, A. Fofonov, and L. Linsen, "Continuous representation of projected attribute spaces of multifields over any spatial sampling," 2013.

[17] I. T. Jolliffe, *Pincipal Component Analysis*. Springer-Verlag, 1986.

[18] I. Borg and P. J. F. Groenen, *Modern Multidimensional Scaling Theory and Applications*, 2nd ed., ser. Springer Series in Statistics. Springer, 2010.

[19] E. Kandogan, "Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions," in *Proceedings of IEEE Information Visualization Symposium (Hot Topics)*, 2000, pp. 4–8.

[20] ——, "Visualizing multi-dimensional clusters, trends, and outliers using star coordinates," *Proc. ACM Int. Conf. Knowledge Discovery and Data Mining*, pp. 107–116, 2001.

[21] S. T. Teoh and K.-L. Ma, "Starclass: Interactive visual classification using star coordinates," in *SDM*, 2003.

[22] L. Linsen, T. V. Long, and P. Rosenthal, "Linking multi-dimensional feature space cluster visualization to surface extraction from multi-field volume data," *IEEE Computer Graphics and Applications*, vol. 29, no. 3, pp. 85–89, 2009.

[23] D. J. Price, "Smoothed particle hydrodynamics and magnetohydrodynamics," *J. Comput. Phys.*, vol. 231, no. 3, pp. 759–794, Feb. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.jcp.2010.12.011

[24] P. Rosenthal and L. Linsen, "Image-space point cloud rendering," in *Proceedings of Computer Graphics International*, 2008, pp. 136–143.

[25] L. Linsen, K. Müller, and P. Rosenthal, "Splat-based ray tracing of point clouds," *Journal of WSCG*, vol. 15, pp. 51–58, 2007.

[26] M. Dan, S. Rosswog, J. Guillochon, and E. Ramirez-Ruiz, "How the merger of two white dwarfs depends on their mass ratio: orbital stability and detonations at contact," *MNRAS*, vol. 422, pp. 2417–2428, May 2012.