Vision, Modeling, and Visualization (2017)
M. Hullin, R. Klein, T. Schultz, A. Yao (Eds.)

Temporal Focus+Context for Clusters in Particle Data

J. Staib, S. Grottel and S. Gumhold

Chair of Computer Graphics & Visualization, TU Dresden

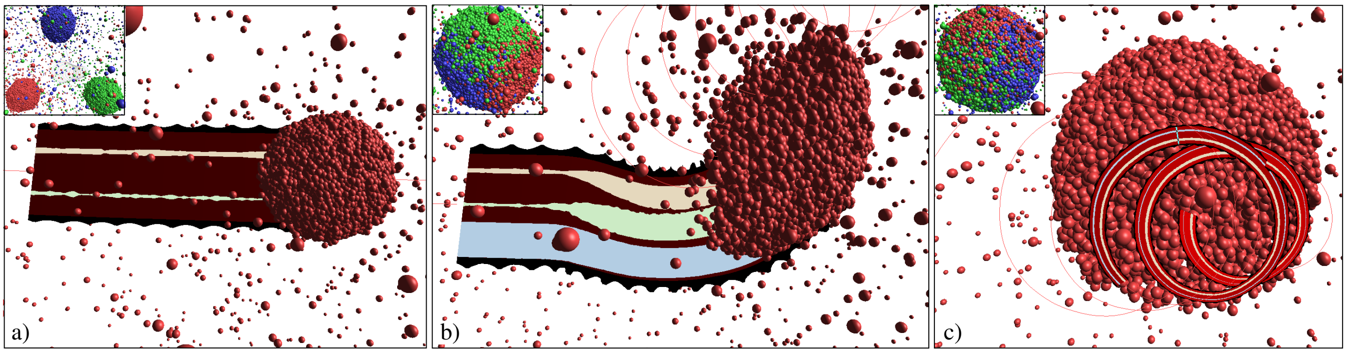

Figure 1: *Development of one of the three particle droplets (from case study 1). One time step is shown in detail while past and future dynamics of the droplet is encoded using ribbons with plotted attributes. (a) The droplet remains in a linear motion in the beginning without spin (yellow) or deformation (green). (b) The droplet is heavily deformed. The trajectory indicates the transition into spiral movement while the attributes show an increase in spin and deformation at the point where linear movement ends. The density (blue) is almost constant. (c) The cluster is subsequently regaining its spherical shape while moving in a spiral, since the spherical anisotropy decreases. For reference, all droplets are shown in the upper left sub images of (a) - (c).*

Abstract

Time-dependent particle-based simulations are typically carried out by direct calculation of interactions between particles over time. The investigation of higher order effects of particle clusters helps understanding the system's dynamic. Existing methods for particle data analysis either rely on animation, where only one time step is visible, or abstraction, which is giving up on visualizing the data in its spatial domain. Inspired from illustrative techniques, we present an interactive focus+context visualization, based on flow ribbons, that combines both approaches. Our method jointly shows one time step in detail, as well as an abstract contextual visualization of past and future dynamics in one image. It allows to assess the time evolution of various cluster attributes around the current temporal focus. We show the usefulness of the approach on two exemplary case studies.

1. Introduction

Particle-based models are the basis for many simulation techniques used for example in thermodynamics, material sciences or molecular biology. Each particle represents an entity, for example an atom or grain. The simulation is typically carried out by directly calculating interactions on the particle level in multiple successive time steps. Clusters of particles form higher order structures, like drops.

In order to assess the time-dependent behavior of these clusters two approaches are commonly used, namely animation and abstraction. Using animation, the data is directly visualized in its spatial domain. This leverages the human capacities to understand higher order structures that would be hard to reliably formulate as algorithms. To interact with time, it is played and paused like a video. The idea is to let the user track related structures and their interaction. Although manipulation of the camera, visualization parameters or filters, is normally allowed, only one time step is visible at a time. Especially for complex scenes, the mental load on the user increases dramatically. This can render animation to be even inferior to static images, since all past information has to be recalled from the user's memory [TMB02]. Also, using animation holds the potential danger of wrong subjective conclusions about movement and especially interaction between data parts [RFF*08]. Thus, a typically used pattern is to repeatedly play, pause and rewind animations while focusing on different parts of the data. Since the human brain can only track a small number of moving objects [AF07], this might become tedious when many objects are present.

DOI: 10.2312/vmv.20171263

The other commonly used approach for analysis, especially in thermodynamics, is based on abstraction, like function plots or graphs. In order to assess and compare changes in time, multiple graphs are presented simultaneously. This visualization, on the one hand, does not rely on the memory of the user and allows for exact depiction of events. On the other hand, by its abstract nature, it gives up on a direct visualization of the data in its domain and thus does not make use of the human's capacity to understand shapes. Methods exist that remedy this problem with linked views (for example [GRVE07]). Generally, designing multi-view solutions with linked graphs is very challenging since the design must be consistent and facilitate the user's understanding of information, that is distributed over many connected plots [WBWK00].

We propose an interactive visualization that jointly shows a direct spatial visualization of particle data for one time step with a condensed abstract view of its temporal context. Our work is inspired by illustrative techniques, especially so called flow ribbons known from comics, that were investigated by Joshi et al. [JR05]. In essence, they are bands that follow a part of the trajectory up to one time step in focus. We extend the idea of showing movement to additionally encode cluster attributes, thus allowing the user to assess past and future development, directly correlated in space and time. We argue that our approach combines the benefits of a spatial, and also an abstract visualization. Since we define one selected time step as focus while having an abstract spatio-temporal context, our approach can be classified as a temporal focus+context method.

Our contributions are:

- We present a focus+context visualization that shows particle clusters of one focused time step in detail and their past and future development as a ribbon in one static scene.
- We discuss and present the calculation of various attributes of clusters, specific to the domain of particle-based simulations.

Throughout the remaining sections, we refer to the state and change of the cluster's attributes as "cluster dynamics".

2. Related Work

Augmenting scientific visualization with abstract graphs is an active field of research (see for example a recent survey by Wang on graph-based techniques [Wan15]). In order to better understand selected data while using a reduced representation of the remaining data, focus+context methods are especially useful. They rely on the definition of a degree-of-interest (doi) function that maps parts of the data either to be in focus or in context, optionally with a smooth boundary [Hau06]. For time-dependent data, this concept is extended to a temporal doi (tdoi) function [CSRC08] by adding a point or range in time to the focus definition.

Conveying movement in static images is one key element in illustration and comics [McC94]. Joshi et al. investigated the applicability of various comic elements for scientific visualization [JR05]. Following the same idea, Nienhaus and Döllner presented a system to analyze a dynamic scene and automatically assign art-inspired techniques where appropriate [ND05]. Our work is based on the idea of simplified, temporally localized and easy to grasp glyphs, as well.

A variety of methods exist to visualize trajectories and their associated attributes. They are employed in various fields, such as biology [KERC09] and especially traffic analysis [CGW15]. Works in these areas can roughly be subdivided into aggregation methods and direct visualizations, which in turn can be divided into approaches that either work on 2D or 3D trajectory data. Aggregation methods, for example density maps are best suited for large numbers of trajectories [SWvdWvW12]. For 2D trajectories like car or ship movement data, the remaining dimension of a 3D scene can be used to either encode time or trajectory dynamics. Tominski et al. presented a visualization of 2D trajectories by using colored bands, stacked in the z-axis to show temporal dynamics [TSAA12]. Häb et al. built on this idea, but used the stacking approach to show multiple attributes [HMRH15].

In contrast to 2D data, 3D trajectories pose additional challenges, since all axes of a 3D scene are occupied for the spatial attributes of the trajectory itself. Especially occlusion and clutter are problems, foremost for large and dense trajectory data. Buschmann et al. proposed an interactive system that combines direct and aggregated visualizations of massive airplane trajectory data [BTD14]. Dense data is selected and filtered using various querying techniques. Also, they include a means of temporal focus+context by allowing the user to compare subsets of trajectories for different time spans by color coding.

For interactive analyses of sparse trajectories, as in our approach, more screen space and rendering effort can be dedicated to single trajectories. Ware et al. visualized sparse movement data of humpback whales [WAPW06] using thick ribbons. They are twisted around their central axis according to the rolling of the whales. To show the direction of movement, chevrons are used. Inspired from their works, Schroeder et al presented a system to visualize surgical training data [SKW*12]. Participants were given a surgical device whose position was tracked. Beside other representations, the authors included a 3D visualization using ribbons. Various attributes are encoded with texture, orientation and color. A related approach was presented by Rakita et al. to show complex trajectories of robot arms [RMG16]. In addition to trajectories, they extract meaningful keyframes and superimpose them in one static image. Recently, Karch et al. presented a system to visualize the dynamics in two-phase flow fields [KBE*17]. Their visualization approach includes a ribbon that conveys movement and twists of droplets, integrated into a spatial visualization.

Several works exist that visualize paths or trajectories in particle data. Lindow et al. proposed a method for molecular dynamics data to estimate and show potential molecular binding paths [LBH11]. In order to show deeply occluded paths, they use advanced shading techniques and semantically meaningful clipping. Fraedrich and Westermann presented a visualization in order to assess motion in astrophysical particle simulations [FW12]. They render distinct particle trajectories on multiple scales and combine tube rendering and volume rendering. Recently, Scharnowski et al. presented a distributed method to efficiently encode and simplify a large number of particle trajectories using Hermite splines [SFRE17].

Note, that rendering techniques for flow ribbons also share similarities with methods in vector field visualization. But since they differ in their motivation (analysis of flow vs. direct depiction of single movements), they are out of scope for this work [JR05].

3. Data Preprocessing

Our method is laid out with a focus in molecular dynamics and thermodynamics. Data sets in these areas typically consist of particles that move and interact, but do not vanish. Higher order structures, represented as clusters, show complex dynamics, like deformation, splitting, merging or evaporation. Please note that the identification of clusters is not part of this work. We rely on pre-clustered data, provided by our application domain scientist partners.

The input is a time-dependent particle dataset with multiple time steps. In the following we denote a certain time step t with a superscript. Each time step consists of a set of particles P^t, each defined by their position $\underline{\boldsymbol{p}}_i^t \in \mathbb{R}^3$ and an optional radius. Per time step, the particles are partitioned into a set of clusters C^t, where each element $C_i^t \subseteq P^t$ itself is a set of its corresponding particles. Particles may join of leave a cluster, which means, that a clusters may appear or vanish.

3.1. Determination of evaporated particles

Especially in molecular dynamics simulations, evaporation and its implications is an important research topic (see for example [HVF16] and related works). Parts of the simulated matter evaporate by detaching from the cluster. While those particles tend to maintain the momentum of the original cluster for still some time, they are successively less affected from the cluster's dynamics. Evaporated particles are identified in the very first step, and removed from subsequent calculations.

In our approach, a particle is considered to be in a gas phase when its approximate local density falls below a simulation specific threshold. For particle datasets, the density can be approximated for each particle by counting the number of elements in a neighborhood with a constant radius. Note that neighbors are sought independently of the cluster partitioning. Formally, a particle at position $\underline{\boldsymbol{p}}_i^t \in P^t$ is in gas phase if

$$\left| \left\{ \underline{\boldsymbol{q}} \in P^t \mid \| \underline{\boldsymbol{q}} - \underline{\boldsymbol{p}}_i^t \| < r \right\} \right| < n_g \tag{1}$$

holds, where r is the search radius and n_g is the lower threshold. Using this criterion, the particles in a cluster C_i^t form two new sets. $C_{g,i}^t$ denote the particles of cluster t in time step i that are in a gas phase and the set $C_{\overline{g},i}^t$ contains all other particles.

3.2. Calculation of the centroid

The cluster's particles that are not in gas phase are used to calculate a centroid $\underline{\boldsymbol{c}}_i^t$. In this work, we treat the particles as elements with constant mass which allows us to calculate the center of mass by calculating their arithmetic mean.

Averaging is done for each coordinate independently to account for cyclic border conditions as shown in [BB08]. First, the coordinates of all particles in $C_{\overline{g},i}^t$ are rescaled to range from 0 - 2π and used as angle for a point on a unit circle. All points on the circle are averaged to yield a new center point inside the circle, which is then projected back to the unit circle. Finally, the angle is calculated and rescaled to the dimension's simulation space. In case the average point is in the origin, all points are uniformly distributed and no angle can be calculated. Since all positions are equally valid, we use the direct arithmetic mean of this coordinate, then. In order to ensure consistent centroids over time, the closest point $\underline{\boldsymbol{c}}_i^t$ in the cyclic simulation space to the centroid $\underline{\boldsymbol{c}}_i^{t-1}$ is calculated.

3.3. Estimation of consistent frame

Additionally, a consistent direction $\vec{\boldsymbol{n}}_i^t$ is estimated associated with each centroid, which is used during visualization to properly orient the ribbons. Its estimation is based on an iterative plane fit to obtain a smooth direction, even in the presence of noisy and degenerated data. Around a centroid, an interval of preceding and successive centroids on the trajectory is used for an eigenvalue decomposition of its covariance matrix. The vector $\vec{\boldsymbol{n}}_i^t$ is obtained from the normalized eigenvector of the smallest eigenvalue, if the plane fit is stable. To assess the stability, the planar isotropy $c_p \in [0,1]$ is used [WPG*97]. Let the eigenvalues be $\lambda_1^c \geq \lambda_2^c \geq \lambda_3^c \geq 0$, then the planar isotropy is calculated as:

$$c_p = \frac{2(\lambda_1^c - \lambda_2^c)}{\lambda_1^c + \lambda_2^c + \lambda_3^c}$$

The bigger the value of c_p the more the points spread into a plane. As long as c_p is too small (i.e. the points spread on a line), the process is repeated with an increased neighborhood. In our experiments, the results were best with a threshold of 0.2.

3.4. Cluster attribute estimation

For the sake of clarity, the cluster id i and time step t is omitted in the following. All calculations refer to particles of one cluster in one time step. Estimated attributes include the angular velocity magnitude ω, the cluster's density ρ, the ratio f of particles to all particles in a time step, the spherical anisotropy σ and the fraction of evaporated particles γ. Also, from the trajectory, i.e. the list of all centroids, we estimate the arc length s, as well as the curvature κ and torsion τ. Note, that also derivatives of these attributes can be used for visualization. This covers for example linear velocity and acceleration as derivatives of the arc length.

Fraction of evaporated particles Since all particles in gas phase are marked, the calculation of the fraction, relative to all particles of a cluster, is very simple. It is given by

$$\gamma = \frac{|C_g|}{|C|} \tag{2}$$

where C_g is the set of particles in a gas phase of a cluster.

Cluster density Given the centroid, a measure that is approximately proportional to the density of the cluster, is calculated. The examination of the density can provide useful information on the expansion or contraction of a cluster. We define this scalar as the ratio of the number of particles to the volume of their enclosing ellipsoid. Assuming an ellipsoidal shape is a good choice for media in thermodynamics that tends to distribute homogeneously, thus generating a spherical shape. In the case of applied translational or rotational forces, the sphere is distorted to be elliptical. First, we estimate the principal axes of the enclosing ellipsoid with a principal component analysis. Then, all particles are projected on each axis. To estimate the length of the axes while accounting for outliers, we then use the interval from the 10-percentile to the 90-percentile of projected lengths from the cluster center on each axis.

Let $\vec{\boldsymbol{v}}_1, \vec{\boldsymbol{v}}_2, \vec{\boldsymbol{v}}_3$ be the eigenvectors of the covariance matrix of all particles in $C_{\overline{g}}$ and $L^{\lambda}_{1/2/3} = \left\{ (\underline{\boldsymbol{p}} - \underline{\boldsymbol{c}}) \cdot \vec{\boldsymbol{v}}_{1/2/3} \mid \underline{\boldsymbol{p}} \in C_{\overline{g}} \right\}$ the projected lengths, then the axis half lengths are defined by the 10th and 90th percentile function $R_{10}(S)$ and $R_{90}(S)$ as

$$\vec{\boldsymbol{w}} = \frac{1}{2} \left| \begin{pmatrix} R_{90}(L^{\lambda}_1) - R_{10}(L^{\lambda}_1) \\ R_{90}(L^{\lambda}_2) - R_{10}(L^{\lambda}_2) \\ R_{90}(L^{\lambda}_3) - R_{10}(L^{\lambda}_3) \end{pmatrix} \right| .$$

The density then reads:

$$\rho = \frac{3|C_o|}{4\pi w_x w_y w_z}, \tag{3}$$

where $C_o \in C_{\overline{g}}$ denotes the set of particles who's projected lengths fall into the interval between the 10th and 90th percentile.

Cluster's angular velocity The angular speed, i.e. the speed of the rotation around the cluster center, is a useful property to characterize applied and inherent forces. We define this measure as the average angular velocity of its particles around the centroid. For particle i, it is approximately proportional to the magnitude of a perpendicular vector to the direction vector $\vec{\boldsymbol{r}}_i$ (from the centroid to the particle's position) and the relative direction vector $\vec{\boldsymbol{d}}_i$ of the particle's movement. The latter can only be obtained for particles that are part of the cluster in the current and last time step. Let C_ω be the intersection of C^t_g and $C^{t-1}_{\overline{g}}$ and points denoted as $\underline{\boldsymbol{p}}_k \in C_\omega$. Let further be $\vec{\boldsymbol{r}}_k = \underline{\boldsymbol{p}}^t_k - \underline{\boldsymbol{c}}^t$ and $\vec{\boldsymbol{d}}^t_k = (\underline{\boldsymbol{p}}^t_k - \underline{\boldsymbol{c}}^t) - (\underline{\boldsymbol{p}}^{t-1}_k - \underline{\boldsymbol{c}}^{t-1})$, then the cluster's angular velocity magnitude is:

$$\omega = \frac{1}{|C_\omega|} \cdot \sum_k \frac{\|\vec{\boldsymbol{r}}^t_k \times \vec{\boldsymbol{d}}^t_k\|}{\|\vec{\boldsymbol{r}}^t_k\|^2} \tag{4}$$

Cluster's spherical anisotropy The spherical anisotropy of a cluster is a measure for its deviation from a spherical shape. In the context of our work, it quantifies the change of the cluster's shape in all spatial dimensions. Bigger values indicate a more anisotropic shape. In particle simulations, the deformation of the cluster's shape is typically the effect of applied internal or external forces, like collisions or bounces [ZJL16]. Given the eigenvalues of the particle's covariance matrix $\lambda^p_1 \geq \lambda^p_2 \geq \lambda^p_3 \geq 0$, we define the spherical anisotropy σ similar to tensor analysis [WPG*97] as:

$$\sigma = \frac{3\lambda^p_3}{\lambda^p_1 + \lambda^p_2 + \lambda^p_3} \tag{5}$$

Trajectory's Curvature and Torsions In order to quantitatively assess changes in the trajectory, its curvature magnitude κ and torsion magnitude τ is estimated. The points on the trajectory are typically noisy which hinders the use of direct numerical derivatives. In order to obtain smooth curvature and torsion values and also to account for non-uniformly spaced sample points, we iteratively fit the trajectory with a B-spline [LGLC05]. In the first step, control points are equally distributed over the (numerically estimated) arc length of the curve. Unless the approximation error between curve points and their closest points on the trajectory falls below a threshold, the control points are moved. The curvature and torsion associated with each centroid is estimated by first locating the curve parameter that produces the closest curve point, given an approximate similar arc length. We then use the first and second derivative to calculate the curvature and torsion. Let $c(s)$ be the fitted curve, then the curvature $\kappa(s)$ and torsion $\tau(s)$ is calculated as

$$\begin{aligned} \kappa(s) &= \frac{c'(s) \times c''(s)}{\|c'(s)\|^3} \\ \tau(s) &= \frac{\left(c'(s) \times c''(s)\right) \cdot c(s)'''}{\|c(s)' \times c(s)''\|^2} \end{aligned}$$

4. Visualization and Interaction

Our visualization is based on the concept of (temporal) focus+context. All time steps of the simulation are jointly visualized in one static scene that can be interactively explored. We map the temporal dynamics to three distinct rendering styles, defined by one time step in focus and a (not necessarily symmetrical) user definable time span before and after this point.

1. For the time step in focus, a detailed spatial visualization is shown. All particles of a cluster are considered and rendered as small spheres that are locally lighted. We employ current state-of-the-art ray casting of simple particle spheres [FGKR16].
2. The trajectory for temporal intervals that are neither in focus nor inside the user defined time span, we show a thin line.
3. Most interesting is the region in the user defined time span, which we visualize using a ribbon-like glyph.

As mentioned, our ribbon rendering work is inspired by illustrative techniques, especially flow ribbons as investigated by Joshi et al. [JR05]. The authors differentiate various types of flow ribbons with increasing complexity. The most complex type consists of a parallel pair of lines with an opaque interior. To hint at occluded features, small line segments are rendered on top. We alter this approach in the following ways as shown in Figure 2 to not only convey movement, but the dynamics of a cluster:

- Instead of placing lines to indicate occluded objects, we use lines with varying thickness to encode a user selectable set of cluster attributes. While the depiction of absolute values is not possible, quantitative changes can be assessed.
- In order to also assess future changse of clusters, we allow the ribbon not only to cover past time, but also future time steps.
- Additionally, we map a fixed selection of attributes to the ribbon's appearance itself. We encode the cluster id, passed time and particle fraction as color in the inner of the ribbon and the degree of evaporation on the ribbon's border.

Each cluster and its ribbon is rendered using a distinct color that is chosen equally on the hue circle in the HSV color space.

4.1. Ribbon Visualization

For each cluster i, a temporal interval $[t_p, t_f]$ that spans a set of discrete time steps is selected by the user that defines a ribbon. The ribbon is a triangle strip that follows the trajectory of the clusters,

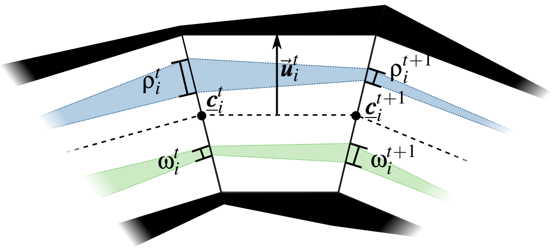
ρ_i^t
$\vec{\boldsymbol{u}}_i^t$
ρ_i^{t+1}
$\underline{\boldsymbol{c}}_i^t$
$\underline{\boldsymbol{c}}_i^{t+1}$
ω_i^t
ω_i^{t+1}

Figure 2: *One face of the ribbon is created from two consecutive points $\underline{\boldsymbol{c}}_i^t$ and $\underline{\boldsymbol{c}}_i^{t+1}$, and an up direction $\vec{\boldsymbol{u}}_i^t$ that is chosen to face the camera. The border is deformed to reflect the cluster's evaporation. A user selectable number of attributes (here exemplary the density ρ and angular velocity ω) is mapped as line plots.*

i.e. the ordered list of centroids $\underline{\boldsymbol{c}}_i^{t_p} ... \underline{\boldsymbol{c}}_i^{t_f}$. It is oriented such that it faces the interactive camera, while minimizing visual distortion. Billboard techniques, i.e. simply enforcing the ribbon's face normal to be directed towards the camera, introduce numerical instabilities when the normal approaches the tangent. Also, this causes twists in the faces that are connected to neighboring faces where the sign is flipped in at least one dimension. By binding the view dependent orientation to a quarter circle, flipping is prohibited.

The viewers position and orientation is defined by a camera viewing matrix V. In order to face the ribbon to the camera, a consistent up direction is calculated view-dependently for every centroid $\underline{\boldsymbol{c}}_i^t$. It is obtained through spherical blending of the preprocessed frame direction $\vec{\boldsymbol{n}}_i^t$ in camera space and another consistently chosen vector $\vec{\boldsymbol{m}}_i^t$. The vector $\vec{\boldsymbol{m}}_i^t$ is perpendicular to both, $\vec{\boldsymbol{n}}_i^t$ and the tangent at $\underline{\boldsymbol{c}}_i^t$. As blend factor, the projection of $\vec{\boldsymbol{n}}_i^t$ to the view direction of the camera is used. Let $\vec{\boldsymbol{d}}_i^t = V \cdot \left(\underline{\boldsymbol{c}}_i^{t+1} - \underline{\boldsymbol{c}}_i^t\right)$ be the direction vector and and $\vec{\boldsymbol{m}}_i^t = \vec{\boldsymbol{d}}_i^t \times V\vec{\boldsymbol{n}}_i^t$, then the sought up direction $\vec{\boldsymbol{u}}_i^t$ is calculated as

$$\vec{\boldsymbol{u}}_i^t = \mathrm{slerp}(V\vec{\boldsymbol{n}}_i^t, \vec{\boldsymbol{m}}_i^t, |(V\vec{\boldsymbol{n}}_i^t)_z|) \qquad (6)$$

The vector $\vec{\boldsymbol{u}}_i^t$ is then normalized.

Equation 6 states, that the bigger the z component of $\vec{\boldsymbol{n}}_i^t$ in camera space, the less it affects $\vec{\boldsymbol{u}}_i^t$, which leads to a vector that has a minimal z component. Blending between two orthogonal vectors with a blend factor between 0 and 1 generates a vector that is always between $\vec{\boldsymbol{n}}_i^t$ and $\vec{\boldsymbol{d}}_i^t$, thus flipping or twisting cannot appear as long as $\vec{\boldsymbol{n}}_i^t$ is consistent, which is practically always the case. Figure 3 (a) and (b) show the effect of our orientation approach, compared to billboard techniques. Relative camera angles exist where the ribbon is facing orthogonally to the camera and thus is invisible. This is a cause of constraining rotation towards the camera to a quarter circle. In order to obtain a seamless consecutive ribbon, the up direction is averaged with the up direction from the subsequent line segment and scaled to the user selectable ribbon thickness.

A ribbon consists of a border and an interior. We encode the change in evaporation, i.e. the ratio of particles that detach from the cluster per time step, in the border by using a jigsaw like pattern. The higher its amplitude, the stronger the evaporation. The color of the inner part is obtained from the cluster id, that defines the hue.

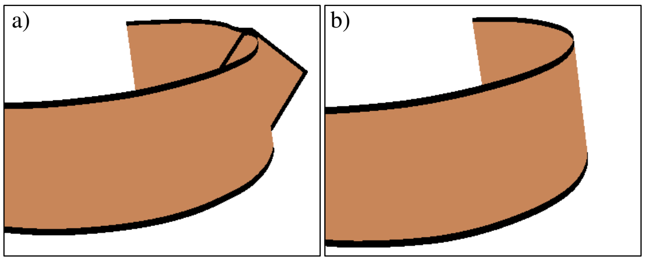

Figure 3: *Instabilities that arise with billboard techniques (a) are eliminated with our method (b).*

The current point in time defines the saturation and the brightness is set to the particle fraction $f_{i,t}$. In the foreground we quantitatively encode user selectable attributes as lines with varying thickness according to the attributes values, normalized to the range $[0,1]$. The colors of the lines are uniquely chosen for each attribute from the "pastel1" qualitative color set from ColorBrewer [HB03].

4.2. User Interface Design and Interaction

The spatial navigation is carried out directly in the 3D visualization using a standard orbit camera. The time step in focus is set globally through a slider. The number of time steps in the future or past that are represented by a ribbon, as well as its thickness can be set for each cluster independently. Clusters and ribbons can be enabled or disabled for visualization. The ribbon's appearance can be manipulated independently for each ribbon or a selection of ribbons. Global parameters allow to enable or disable the mapping of cluster characteristics to properties of the ribbon, i.e. the jigsaw border and color. Furthermore, the user can enable or disable attributes to be shown on the ribbons as line plots. Regarding the focus visualization, the global radius can be manipulated. Also, independently per cluster, a scaling relative to the cluster center can be defined.

To efficiently select clusters, the 2D user interface contains a table of all clusters. Each row represents one cluster and consists of a color field, a checkbox to control its visibility and an abstract overview visualization of the trajectory's approximate curve types, as shown for example in Figure 5 (a).

To generate the overview, the complete simulation time is equally subdivided into 20 intervals. Each subset is represented by a square, with the color mapped to the trajectory's approximate type. Note that these colors are only existent in the user interface and thus do not interfere with the attribute colors on the ribbons in the 3d scene. The curve types include linear, circular, spiral (circular in the plane with decreasing radius), helical, helical with increasing or decreasing curvature and arbitrary. It is obtained by a linear regression of the subset's curvature and torsion values. Depending on combinations of the slope m and y-intercept n the types are assigned according to Table 1. The curvature, resp. torsion is approximately zero if $m = n = 0$, constant if $m = 0, n \neq 0$ and linear if $m \neq 0$. If the maximum error of the curvature, resp. torsion values from the regression exceeds a threshold, which is chosen to be smaller than the slope, then the curve type is assumed to be arbitrary.

	$\kappa = 0$	κ const	κ linear
$\tau = 0$	🟩 line	🟪 circle	🟨 spiral
τ konst	🟦 arbitrary	🟧 helix	🟫 helix +/-
τ linear	🟦 arbitrary	🟦 arbitrary	🟦 arbitrary

⬛ no movement

Table 1: *Local curve types are assigned depending on the local curvature and torsion. The colored squares indicate their colors in the overview plot.*

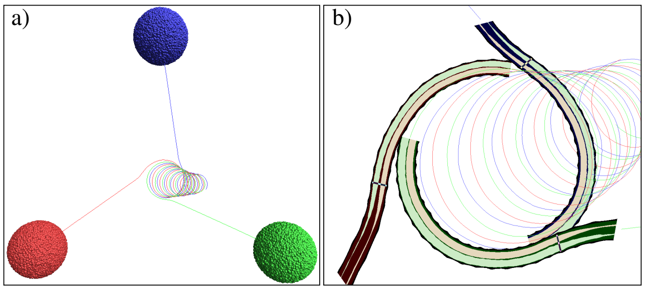

Figure 4: *Three colliding droplets (case study 1). (a) The complete trajectory as line strip shows an initial linear movement, followed by a spiral like pattern. (b) Ribbons of past and future development of all three droplets similarly show a similar development.*

5. Results

We show the usefulness of our approach on two exemplary case studies from molecular dynamics and thermodynamics. Several time steps for both datasets are focused and the past and future dynamics is explained.

5.1. Case Study 1 - Three Colliding Droplets

This dataset is a molecular dynamics simulation of three liquid droplets of similar Lennard-Jones material model that approach linearly, collide and then merge. During the complete simulation, the droplets evaporate into the simulation space. The dataset consists of 3,000 time steps with 79,509 particles per time step, i.e. 26,503 particles per droplet. Figure 4 (a) shows the trajectory of all three droplets as line strips. The droplets approach and then collide approximately at the center of the simulation space, slightly in an angle which induces linear movement and rotation.

Investigating only one droplet, the motion can be split into a linear part and a spiral-like movement. Figure 1 (a) shows a point in time in the middle of the linear part with a ribbon showing past motion for 50 frames. On the ribbon, the angular velocity (upper line, light red) and spherical anisotropy (lower line, light green) are rendered as line plots. The movement pattern is very simple with low rotational velocity and low spherical anisotropy. The ribbons show that this has been a stable process in the past. Also shown is the evaporation by the jagged border.

Figure 1 (b) shows a later point in time of the development, after the droplet begins moving in a spiral. Here, a third attribute, the density, is shown at the bottom of the ribbon (light blue). The cluster in the focused time step is heavily deformed, caused by the collision with other clusters. On the ribbon of past movement the rotational velocity and the spherical anisotropy increase, indicating on the deformation and an emerging spin. It can be seen that the cause of this drastic change is in the past, right at the moment where the droplet also begins the spiral movement pattern. Also, it does not affect the density or evaporation, as the thickness of the light blue line, as well as the amplitude of the jigsaw-pattern is almost constant.

Figure 1 (c) shows a point in time where the droplet has completed three revolutions. Apparently, it returns back to a spherical shape. A ribbon of past and future movement is shown with two line plots encoding the density (light blue) and the spherical anisotropy (light green). To improve visibility, it is superimposed on the particle visualization and the focused time step is marked on the ribbon with a bar. What can be seen is, that the anisotropy decreased in the past (i.e. in direction of an increasingly darker ribbon color) while the cluster's movement is a spiral. The same happens with the density. After collision, the clusters diffuse into each other and thus slowly return to their spherical shape. The process, that started the reformation of the cluster into a sphere started further in the past. Also, it can be seen, that this decrease in spherical anisotropy and density continues in the future, which means, that it is a long-term process.

Figure 4 (b) shows the interplay of all droplets using ribbons that show past dynamics before the collision took place and future dynamics for the time needed for a quarter revolution. For this visualization, focus rendering was disabled. The current time point is again indicated by the bar on the ribbons. All three droplets show similar movement, as well as deformation and beginning spin. This indicates an identical setup for all three droplets and supports the assumption, that the linear movement and rotation after collision is an effect of the collision angles.

5.2. Case Study 2 - Laser Ablation

In this dataset, a laser ablation is simulated. Here, an aluminum block, simulated as a regular grid of particles, is hit by two lasers that both move linearly along the x, resp. y axis, such that they form a cross. The laser impact injects high amounts of energy into the particles, thus stimulating the electrons. Driven by their kinetic energy, they move away from the block. This simulation has 400 time steps with 562,500 particles per time step. Since the majority of particles remain in the metal block throughout the simulation, the emerging clusters only consist of a small number of elements in the order of 100. The clustering was reconstructed in a preprocessing step using local density analysis and tracking of coherent structures, leading to 32 clusters in total.

Figure 5 (b) shows the dataset focused on the first time step with ribbons indicating future development. On the ribbons, the linear acceleration is plotted. In the image, one laser moves from the right to the left and another from the front to the back on the metal block. Due to the linear laser movement, the cluster trajectories all aim into a diagonal direction. Although all ribbons represent the same time span, their lengths differ, indicating that the clusters move with different speed. Clusters near the intersection point of the two lasers move much faster. Interestingly, the later a cluster emerges,

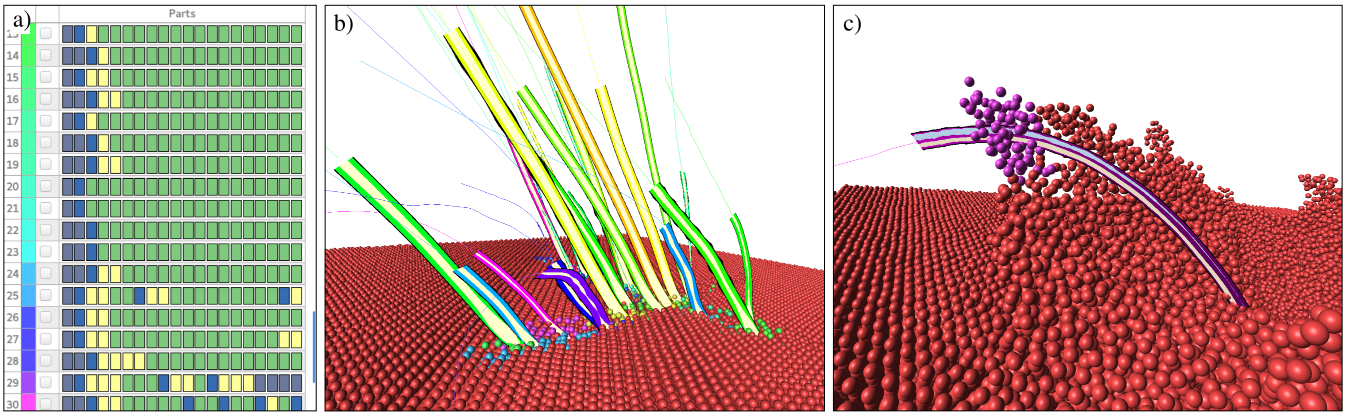

Figure 5: *Laser ablation dataset (case study 2). (a) Excerpt of the trajectory overview. (b) Visualization of all trajectories with acceleration shown as line plots on the ribbons. (c) One trajectory shows an increase in density where it detaches from the wave on the metal block.*

the faster it moves. This might be an effect of the heating of the metal block due to the laser's energy, facilitating the detachment of particles. The linear acceleration increases fast in almost all trajectories and then sinks, hinting on the applied forces that initiate the movement. Note, that due to the fact that the acceleration is below zero at the end of the trajectory, the value of zero, which is mostly present, is mapped to a visible line.

Some trajectories, show a special behavior. Figure 5 (a) shows an excerpt of the overview trajectory list. As mentioned , one column consists of colored blocks that roughly describe the trajectory's shape. Most trajectories are stationary in the beginning, then move in an arbitrary pattern, then in a spiral pattern and finally show linear movement. The trajectories with the ordering number 27 to 30 show other movement patterns, since they do not move linearly after being set in motion. In fact, their angle is too shallow to escape the metal block. One of the clusters is even falling back on the surface. In Figure 5 (b) they are located in the front left of the image.

Figure 5 (c) contains a detailed view of trajectory 30, rendered as a ribbon of future and past movement with the density (top) and the spherical anisotropy plotted. While the density rises, the spherical anisotropy is almost constant. Focusing on the point of the highest density, it becomes apparent that this is the point in time, where the cluster separates from the wave. Here, the cluster becomes compact, but is still heavily distorted.

5.3. Performance and Scalability

The preprocessing step is CPU-intensive. For example, in order to determine evaporated particles, a neighborhood graph has to be built for every frame and nearest-neighbor queries must be performed for every particle. In our current implementation, the preprocessing is completely done on the CPU, but parallelized using OpenMP. While the determination of evaporated particles is parallelized over different time steps, all other steps can be trivially be parallelized over different clusters. Figure 6 shows the contribution of the several steps to the overall computation time. On a consumer test system (Intel Core i7-3770 with 3.4 GHz), the preprocessing took 1.6 min for the colliding droplets dataset and 1.8 min for the laser ablation dataset. Note, however, that this task has to be carried out only once for a dataset, since the results can be stored on the harddisk and loaded for subsequent analysis tasks.

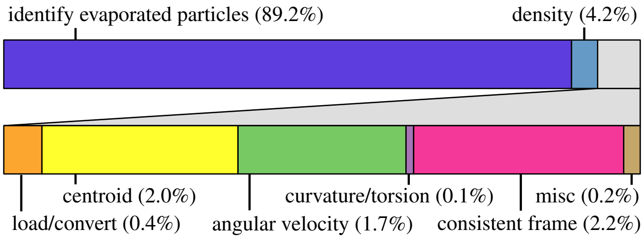

Figure 6: *Contribution of the several steps to preprocessing time.*

The performance of the visualization itself is defined by the used particle rendering method. We use a direct visualization that is based on ray casting. Optimized implementations can achieve realtime framerates for particle counts in the order of 10^7 [FGKR16]. Our contribution, the ribbon visualization, has a comparably low computational cost. A ribbon is a triangle strip that is rendered in a single draw call and generated on-the-fly using the geometry shader. The line graphs are computed in the fragment shader based on vertex buffers and thus does not require any additional texture lookups. Typically, a ribbon consists of triangles in the order of hundreds or thousands.

Our approach is scalable in terms of the dataset size. For the preprocessing, not more that two complete time steps need to be loaded into memory at the same time. The data that is generated is only a small fraction of the original data. For example, the extracted attributes have a size of 540.7 KB for the droplet dataset. Also for the visualization, only one frame, as well as the attribute data, must be present in GPU memory at one time. For the droplet dataset, GPU usage can thus be reduced to not more than 2 MB. Of course, keeping more frames in memory would allow for faster playback of the time steps as an animation, but this is of minor importance in our method which is based on the analysis of a static scene.

6. Discussion

Existing methods for sparse trajectory visualization aim at showing the trajectory as a whole in order to identify interesting behaviors throughout the movement. The goal of our focus+context method differs, as it aims to help explaining the cause or the effect of a certain cluster state over a limited time span.

The direct visualization of particles as distinct spheres allows the user to assess and understand complex cluster shapes and constellations. Especially when function plots show an unexpected behavior, spatial visualizations are required. Abstracted visualizations allow to reduce complex dynamics of a cluster to a small number of scalars. The case studies show that a joint visualization indeed conveys complex structures while also hinting at their cause, as well as their further development in one static image. For example, in the droplets dataset, the moment in time where the deformation started is clearly visible. More importantly, dynamics that lead to a certain cluster state at a focused time can be correlated to a spatial region. In the droplets dataset, the beginning of the deformation can clearly be correlated with the beginning of the spiral movement, induced by the collision of the other two similar droplets. As shown in the laser ablation study, the cause of the shape of at least one of the clusters can be ascribed to a region of the wave on the metal block.

The ribbons help to filter from a larger number of clusters directly in the spatial visualization, judged not only by the movement but also other cluster attributes. For example, some of the trajectories in the laser ablation dataset show a differing behavior in their movement and their acceleration. These trajectories lie in close proximity. Additionally, the overview table, as shown in Figure 5 (a), allows to assess the approximate movement of each cluster in order to identify prominent trajectories. In the laser ablation case study, three trajectories could be easily identified.

Trajectory visualizations that show the complete movement typically suffer from the problem, that points on the trajectory cannot be mapped to time. Visual hints must be added, for example by arrows or color. Although we also use color to indicate time, the temporal context is much clearer in our scenario, since it is explicitly defined by the user and only affects a comparably short time span. Also, shorter ribbons allow to dedicate more screen space and computational effort to their rendering. By facing the ribbon to the camera, without introducing twists or distortions, the effective plotting area is enlarged. We use this area to additionally qualitatively map attribute values, like evaporation.

Since the ribbon serves as a context for the particle visualization in focus, a trade off concerning visibility and occlusion has to be made. On the one hand, the ribbons should occlude as less of the focused particle cluster as possible. On the other hand, the ribbon itself must be sufficiently visible in order to allow assessment plotted values. We alleviate this issue with various techniques like superposition, as well as interactive control, such as filtering or manipulation of ribbon and cluster parameters.

Another challenge concerning the ribbons is the distortion of plotted attribute values. Although we aim at minimizing distortion, it is naturally induced simply by plotting information on a curved band. It is further induced by perspective and occlusions. A much bigger issue is the differing arc length of the lower ribbon border and the upper ribbon border. While the direct depiction of absolute values is not possible, the user can compare values and put them into order, even to assess ratios.

7. Conclusion and future work

In this work we propose a temporal focus+context method to assess past and future development of clusters in time-dependent particle data sets. We have presented a variety of cluster attributes and their estimation. Our visualization is based on the idea of flow ribbons, augmented to not only encode movement but additional attributes of a selectable time span. The ribbon's placement and layout is designed to minimize distortions while at the same time remain maximum visibility. Our case studies have shown, that past and future development can be conveyed in one static scene. Furthermore we have shown that our approach helps in selecting interesting elements from a larger number of clusters.

For future work, we identified four challenging areas: attribute estimation, mapping, overplotting and selection. Deriving precise attributes from discrete data is a complex task. Finding more robust or more general estimation methods, for example to better approximate the density, would help to make values more expressive and general. We also want to experiment with more sophisticated mappings for attributes. This includes special glyphs to indicate no movement, as well as methods to convey dynamics even when clusters are not spatially locatable or separable. Also, more sophisticated choices for colors improve intelligibility.

Regarding overplotting, advanced methods might be helpful, from using transparency to interactive tools, like cutout views. Selection is a challenge on the semantic scale (i.e. which trajectory to select) and on the temporal scale (i.e. which time point to select). We presented our solutions to these problems. More advanced methods for semantic selection are imaginable, for example by coselecting similar trajectories. Automatic methods for time span selection can help to preselect areas of interesting attribute changes. Overall, we believe, that this method is a valuable addition to existing tools to help understanding the dynamics of particle clusters.

Acknowledgements

This work was partially founded by BMBF Project No. F-002093-521-001-1110407 (ScaDS).

References

[AF07] Alvarez G. A., Franconeri S. L.: How many objects can you track?: Evidence for a resource-limited attentive tracking mechanism. *Journal of vision 7*, 13 (2007), 14–14. 1

[BB08] Bai L., Breen D.: Calculating center of mass in an unbounded 2d environment. *Journal of Graphics Tools 13*, 4 (2008), 53–60. doi:10.1080/2151237X.2008.10129266. 3

[BTD14] Buschmann S., Trapp M., Dollner J.: Real-time animated visualization of massive air-traffic trajectories. *2014 International Conference on Cyberworlds (CW) 00* (2014), 174–181. doi:doi.ieeecomputersociety.org/10.1109/CW.2014.32. 2

[CGW15] Chen W., Guo F., Wang F. Y.: A survey of traffic data visualization. *IEEE Transactions on Intelligent Transportation Systems 16*, 6 (Dec 2015), 2970–2984. doi:10.1109/TITS.2015.2436897. 2

[CSRC08] CARVALHO A., SOUSA A. A. D., RIBEIRO C., COSTA E.: A temporal focus + context visualization model for handling valid-time spatial information. *Information Visualization 7*, 3-4 (2008), 265–274. doi:10.1057/PALGRAVE.IVS.9500188. 2

[FGKR16] FALK M., GROTTEL S., KRONE M., REINA G.: *Interactive GPU-based Visualization of Large Dynamic Particle Data*, vol. 4 of *Synthesis Lectures on Visualization*. Morgan & Claypool Publishers, San Rafael, CA, 2016. doi:10.2200/S00731ED1V01Y201608VIS008. 4, 7

[FW12] FRAEDRICH R., WESTERMANN R.: Motion visualization in large particle simulations. vol. 8294, pp. 82940Q–82940Q–12. doi:10.1117/12.904668. 2

[GRVE07] GROTTEL S., REINA G., VRABEC J., ERTL T.: Visual verification and analysis of cluster detection for molecular dynamics. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (Nov 2007), 1624–1631. doi:10.1109/TVCG.2007.70614. 2

[Hau06] HAUSER H.: Generalizing focus+context visualization. In *Scientific visualization: The visual extraction of knowledge from data*. Springer, 2006, pp. 305–327. 2

[HB03] HARROWER M., BREWER C. A.: ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. *The Cartographic Journal 40*, 1 (2003), 27–37. doi:10.1179/000870403235002042. 5

[HMRH15] HÄB K., MIDDEL A., RUDDELL B. L., HAGEN H.: Travis - a visualization framework for mobile transect data sets in an urban microclimate context. In *2015 IEEE Pacific Visualization Symposium (PacificVis)* (April 2015), pp. 167–174. doi:10.1109/PACIFICVIS.2015.7156374. 2

[HVF16] HEINEN M., VRABEC J., FISCHER J.: Communication: Evaporation: Influence of heat transport in the liquid on the interface temperature and the particle flux. *The Journal of Chemical Physics 145*, 8 (2016), 081101. doi:10.1063/1.4961542. 3

[JR05] JOSHI A., RHEINGANS P.: Illustration-inspired techniques for visualizing time-varying data. In *VIS 05. IEEE Visualization, 2005.* (Oct 2005), pp. 679–686. doi:10.1109/VISUAL.2005.1532857. 2, 3, 4

[KBE*17] KARCH G., BECK F., ERTL M., MEISTER C., SCHULTE K., WEIGAND B., ERTL T., SADLO F.: Visual analysis of inclusion dynamics in two-phase flow. *IEEE Transactions on Visualization and Computer Graphics PP*, 99 (2017), 1–1. doi:10.1109/TVCG.2017.2692781. 2

[KERC09] KEEFE D., EWERT M., RIBARSKY W., CHANG R.: Interactive coordinated multiple-view visualization of biomechanical motion data. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (Nov 2009), 1383–1390. doi:10.1109/TVCG.2009.152. 2

[LBH11] LINDOW N., BAUM D., HEGE H. C.: Voronoi-based extraction and visualization of molecular paths. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (Dec 2011), 2025–2034. doi:10.1109/TVCG.2011.259. 2

[LGLC05] LEWINER T., GOMES J. D., LOPES H., CRAIZER M.: Curvature and torsion estimators based on parametric curve fitting. *Computers & Graphics 29*, 5 (2005), 641 – 655. doi:http://dx.doi.org/10.1016/j.cag.2005.08.004. 4

[McC94] MCCLOUD S.: *Understanding Comics*. A Kitchen Sink book. HarperCollins, 1994. 2

[ND05] NIENHAUS M., DOLLNER J.: Depicting dynamics using principles of visual art and narrations. *IEEE Comp. Graph. & Appl. 25*, 3 (May 2005), 40–51. doi:10.1109/MCG.2005.53. 2

[RFF*08] ROBERTSON G., FERNANDEZ R., FISHER D., LEE B., STASKO J.: Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (Nov 2008), 1325 – 1332. doi:10.1109/TVCG.2008.125. 1

[RMG16] RAKITA D., MUTLU B., GLEICHER M.: Motion synopsis for robot arm trajectories. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (Aug 2016), pp. 281–287. doi:10.1109/ROMAN.2016.7745143. 2

[SFRE17] SCHARNOWSKI K., FREY S., RAFFIN B., ERTL T.: Spline-based Decomposition of Streamed Particle Trajectories for Efficient Transfer and Analysis. In *EG 2017 - Short Papers* (2017), Peytavie A., Bosch C., (Eds.), The Eurographics Association. doi:10.2312/egsh.20171010. 2

[SKW*12] SCHROEDER D., KOWALEWSKI T., WHITE L., CARLIS J., SANTOS E., SWEET R., LENDVAY T. S., REIHSEN T., KEEFE D. F.: Exploratory visualization of surgical training databases for improving skill acquisition. *IEEE Computer Graphics and Applications 32*, 6 (Nov 2012), 71–81. doi:10.1109/MCG.2012.67. 2

[SWvdWvW12] SCHEEPENS R., WILLEMS N., VAN DE WETERING H., VAN WIJK J.: Interactive density maps for moving objects. *IEEE Computer Graphics and Applications 32*, 1 (Jan 2012), 56–66. doi:10.1109/MCG.2011.88. 2

[TMB02] TVERSKY B., MORRISON J. B., BETRANCOURT M.: Animation: can it facilitate? *International Journal of Human-Computer Studies 57*, 4 (2002), 247 – 262. doi:10.1006/ijhc.2002.1017. 1

[TSAA12] TOMINSKI C., SCHUMANN H., ANDRIENKO G., ANDRIENKO N.: Stacking-based visualization of trajectory attribute data. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (Dec 2012), 2565–2574. doi:10.1109/TVCG.2012.265. 2

[Wan15] WANG C.: A Survey of Graph-Based Representations and Techniques for Scientific Visualization. In *Eurographics Conference on Visualization (EuroVis) - STARs* (2015), Borgo R., Ganovelli F., Viola I., (Eds.), The Eurographics Association. doi:10.2312/eurovisstar.20151111. 2

[WAPW06] WARE C., ARSENAULT R., PLUMLEE M., WILEY D.: Visualizing the underwater behavior of humpback whales. *IEEE Computer Graphics and Applications 26*, 4 (July 2006), 14–18. doi:10.1109/MCG.2006.93. 2

[WBWK00] WANG BALDONADO M. Q., WOODRUFF A., KUCHINSKY A.: Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (New York, NY, USA, 2000), AVI '00, ACM, pp. 110–119. doi:10.1145/345513.345271. 2

[WPG*97] WESTIN C.-F., PELED S., GUDBJARTSSON H., KIKINIS R., JOLESZ F. A.: Geometrical diffusion measures for MRI from tensor basis analysis. In *ISMRM '97* (April 1997), p. 1742. 3, 4

[ZJL16] ZHANG Y. R., JIANG X. Z., LUO K. H.: Bounce regime of droplet collisions: A molecular dynamics study. *Journal of Computational Science 17* (2016), 457–462. 4