



Alexander Straub · Filip Sadlo · Thomas Ertl

Feature-based deformation for flow visualization

Received: 21 December 2022 / Revised: 21 December 2022 / Accepted: 14 January 2024
© The Author(s) 2024

Abstract We present an approach that supports the analysis of flow dynamics in the neighborhood of curved line-type features, such as vortex core lines, attachment lines, and trajectories. We achieve this with continuous deformation to the flow field to straighten such features. This provides “deformed frames of reference”, within which qualitative flow dynamics are better observable with respect to the feature. Our approach operates at interactive rates on graphics hardware, and supports exploration of large and complex datasets by continuously navigating the additional degree of freedom of deformation. We demonstrate the properties and the utility of our approach using synthetic and simulated flow fields, with a focus on the application to vortex core lines.

Keywords Feature line · Smoothing · Spline skinning · Deformation · Flow visualization

1 Introduction

A wide range of problems in science and engineering involves some kind of transport, modeled prevalently by means of vector fields. Analysis of the resulting fields is, however, nontrivial. While for 2D vector fields, dense visualization using arrows, integral curves, or textures can provide insight, they fail at guiding the attention to relevant structures. Further, they are unfeasible in 3D, due to occlusion and visual clutter. A powerful alternative is to employ feature extraction techniques to extract prominent structures, such as vortex core lines and attachment lines. A vortex core line represents the (typically bent) axis of a vortex, i.e., a curve around which the flow is swirling. Such a core line can be visualized directly to convey the location and overall shape of a vortex. However, it cannot provide insight into the dynamics of a vortex, since that would require to include the flow in its volumetric neighborhood. Therefore, a widely employed approach is to utilize feature lines for seeding integral curves in their vicinity to reveal relevant flow dynamics (see, e.g., Fig. 10b). Although such feature-based visualization provides a good view on the overall dynamics in such a region, it does not do a good job in revealing the dynamics in relation to the feature itself. Imagine, for example, a bent vortex with low “twist” around its core line but with a high tangential velocity along it, as is the case in certain turbines (Roth and Peikert 1996), where streamlines exhibit less than a single

A. Straub (✉) · T. Ertl
Visualization Research Center, University of Stuttgart, Stuttgart, Germany
E-mail: alexander.straub@visus.uni-stuttgart.de

T. Ertl
E-mail: thomas.ertl@visus.uni-stuttgart.de

F. Sadlo
Interdisciplinary Center for Scientific Computing, Heidelberg University, Heidelberg, Germany
E-mail: sadlo@uni-heidelberg.de

Published online: 13 May 2024

revolution around a vortex core. The more bent such a vortex is, the more difficult it becomes to observe the twist, i.e., to mentally separate the rotation around the core line from the overall bent motion (see, e.g., Fig. 12a).

The above observations motivate us to unbend feature lines for an “undistorted view” on the dynamics of interest, i.e., to deform space such that a selected feature becomes straight. To provide guidance and quantitative context, this deformation should be realized as a smooth animation from the original (reference) state to the deformed state. To enable exploration, the animation should be interactive, i.e., one should be able to navigate the deformation process. For continuity in space, our objective is to deform space, such that the vicinity of the feature deforms together with the feature. At the same time, we want to maintain spatial context, by avoiding self-intersection of the deformed space, and by avoiding excessive distortion, which could result if the original feature passed a region multiple times. This necessitates to “tear apart” the space along manifolds during deformation, which we include in our visualization for context (e.g., orange dots in Fig. 10c). All of this leads to the statement of requirements for the deformation process, which guarantee that certain geometric properties are preserved (cf. Sects. 3 and 4), some specific to certain feature types.

While this preservation of properties generally allows for interpretation in deformed space, not all properties can be preserved at the same time. Particularly when dealing with solutions stemming from Navier–Stokes or other differential equations, physical properties are affected and lost. However, important geometric properties are preserved and thus, in a Lagrangian scope, the feature and its relative neighborhood remain intact. Hence, our method still allows for qualitative insight on the deformed feature line and its surrounding, while quantitative information is limited to the preserved properties. Because of this, we see our approach as a complementary strategy and not as a replacement, where visualizations in the original and the deformed space could be analyzed side-by-side. This ensures that physical properties can be observed from the original data, while mostly qualitative information can be retrieved from the deformed vector field. Please note that most importantly, as basis for flow analysis, the feature itself is still provided together with its neighborhood. As a conclusion, explaining flow relative to certain features can be made more intuitive when employing our method, e.g., for observing the windings of streamlines around a vortex core line (–cf. Sect. 4.2). In this regard, our method could, for example, be used for teaching features in flow visualization, or to qualitatively evaluate feature extraction algorithms.

Our contributions include:

- feature-based deformation for flow visualization,
- a spline-based realization of this concept with guarantees, and
- interactive exploration of the resulting deformation space.

2 Related work

In flow visualization, many approaches are based on streamlines and pathlines. Particularly in three-dimensional space, densely seeded, large, and curved integral lines often lead to visual clutter and occlusion. Techniques to improve their usefulness mostly rely on seeding strategies (Li et al. 2008; Spencer et al. 2009; Yu et al. 2012), often combined with similarity measures for filtering (McLoughlin et al. 2013), or opacity optimization (Günther et al. 2014, 2017). The achieved goal is to show only few, representative lines, which still depict the overall flow pattern, or reduce the opacity of less relevant lines. Further, view-dependent methods (Marchesin et al. 2010) can be employed to allow for interactive exploration. By combining ideas from human–computer interaction with flow visualization, the user can be kept in the loop. Here, the user can differentiate into regions of interest and context (Tao et al. 2014; Tong et al. 2016). Many of these approaches use the idea of expanding space to move surrounding lines away from the point of interest but at the same time keeping them visible for context (Correa et al. 2007; Li et al. 2017). While these approaches deform space directly, our method does so in a feature-driven manner.

For the deformation of space, the focus in research is predominantly on mesh deformation and animation. Therefore, methods are often cage-based (Lipman et al. 2008; Ben-Chen et al. 2009), require a surface mesh (Yoshizawa et al. 2003), or are shape-aware (Jacobson et al. 2011). Thus, they are not applicable to feature line-based deformation. Due to their generality, point-deforming methods (Schaefer et al. 2006) better fit our requirements, especially when limited to translational and rotational deformation (Huang et al. 2014). Our deformation approach thus follows a skinning concept (Jacobson et al. 2012), i.e., spline skinning (Forstmann et al. 2007).

Similar to our approach are the papers by Daae Lampe et al. (2009), and Angelelli and Hauser (2011) on curve-centric volume reformation. In their works, they share our goal of straightening a line, and by this deforming its surrounding space. However, the work of Daae Lampe et al. is designed for scalar fields, and is limited to cases where a user-defined up vector is not orthogonal to the center line. Angelelli and Hauser extend this approach, but restrict themselves to straightening tubular domains. There, they rely on explicitly defined planes, i.e., a user-defined sampling along a center line, assuming linearity for the space between two planes. In our method, we overcome these limitations by modeling the feature line as an at least C^1 continuous curve. Because of this, our approach can be seen as an improvement of their work, as well as a generalization applicable to a wider range of feature lines and arbitrarily formed grids. Further, much work has been done on space deformation for comparative visualization in medical applications, e.g., by flattening structures captured by medical imaging, such as colons using a finite element approach (Haker et al. 2000). Other applications include flattening to untangle vascular structures (Kanitsar et al. 2003), or use the third dimension for temporal visualization of flattened mitral valves (Eulzer et al. 2020). An exhaustive overview of such applications is provided by Kreiser et al. (2018).

In other approaches to deform fluid flow, the vector field is manipulated directly (Sato et al. 2014, 2015), instead of deforming the geometry or the underlying grid. However, in contrast to our work, their goal is to alter the fluid simulation, in order to create new artistic fluid animations, while preserving physical behavior. While some works provide approaches for changing frames of reference, e.g., in a Galilean invariant or objective manner (Hadwiger et al. 2019; Rautek et al. 2021), our work focuses on the preservation of (specific) geometric properties. Still, our approach is comparable to their works with respect to interpretability of deformed spaces. Although our main contribution is on space deformation, the deformation itself is governed by the straightening of feature lines, such as vortex core lines (Theisel et al. 2005; Günther and Theisel 2020), bifurcation lines (Roth 2000; Machado et al. 2013), or attachment and separation lines (Kenwright et al. 1999).

3 Method

We seek a mapping $\phi(\mathbf{x}, \tau)$ (our deformation) to transform a point

$$\mathbf{x} = \phi(\mathbf{x}, \tau = 0) \in \Omega \quad (1)$$

of the original domain $\Omega \subseteq \mathbb{R}^n$ to its deformed position

$$\tilde{\mathbf{x}} = \phi(\mathbf{x}, \tau = 1) \in \tilde{\Omega} \quad (2)$$

in the deformed domain $\tilde{\Omega} \subseteq \mathbb{R}^n$, as a function of “deformation animation time” $\tau \in [0, 1]$. That is, we obtain deformation animation by continuously varying τ from 0 to 1 and applying $\phi(\mathbf{x}, \tau)$ to all points \mathbf{x} of our objects of interest. For simplified notation, we write, e.g., ϕ for $\phi(\cdot, \cdot)$, if unambiguous.

Since most flow visualization techniques operate on vector fields discretized into nodes and cells, we could apply ϕ to all nodes of such a grid to obtain its deformed representation, and apply the visualization techniques to the deformed grid. This would, however, introduce inaccuracies, because most visualization algorithms assume straight grid edges, whereas application of ϕ would lead to (higher-order) deformed cells. This would lead to exponential error growth when integrating, for example, streamlines. Therefore, we favor application of visualization techniques to the original grid, and instead deform their geometrical result using ϕ . Techniques that do not generate geometrical results, e.g., texture-based techniques such as line integral convolution (LIC) (Cabral and Leedom 1993), need to be applied on the deformed grid. However, the typical locality of such approaches, e.g., rather short streamlines for LIC or glyph-based visualization, avoids or limits error growth. Both approaches—applying the deformation on the geometry, i.e., on streamlines, or applying the computation of streamlines on the deformed vector field—should lead to results in a Lagrangian scope. This means that the relative neighborhood should be preserved.

Summarizing our requirements, ϕ should:

- R1 deform a feature line given at $\tau = 0$ to a straight line at $\tau = 1$,
- R2 be continuous in τ to enable animations,
- R3 be continuous in space, except for permissible “tears”,
- R4 avoid self-intersection of $\tilde{\Omega}$, to prevent penetration of objects,
- R5 avoid deformation that obscures feature dynamics, and

R6 be efficient to compute, to enable interactive exploration.

Requirement R1 is a hard constraint to define ϕ . Thus, we first need to define a transformation ϑ that deforms the feature line

$$\mathcal{L} = (\mathbf{y}_i)_{i=1,\dots,m}, \quad (3)$$

which in our approach is given as a polyline consisting of m vertices \mathbf{y}_i , to a straight line. To satisfy Requirement R2, this transformation needs to be parametrized with animation time τ . This motivates an iterative formulation of ϑ , which we derive in Sect. 3.1. Based on ϑ , we then obtain the space deformation ϕ in Sect. 3.2.

3.1 Line deformation ϑ

A straightforward approach to straighten a line would be to connect its endpoints to create a straight line, and additionally preserving its arc-length parametrization. However, in configurations where the endpoints of a bent line are close to each other (e.g., Fig. 1j), the resulting line would be very short (Fig. 1k). Furthermore, this would not enable animation of the deformation, as demanded by Requirement R2. Thus, a natural strategy is to gradually smooth the line until it becomes straight, i.e., to apply Laplacian smoothing (Taubin 1995), whose iterative approach lends itself well for this purpose. An issue, however, is that the geometry may shrink during the smoothing process (Fig. 2). Fixing the positions of the line’s endpoints would help in some cases (Fig. 1h), but would fail in others, where the endpoints are already close initially (Fig. 1k). Nevertheless, since curvature is generally reduced by the smoothing, the endpoints tend to move apart during the first phase (Fig. 2a, b) of Laplacian smoothing in such configurations, and they start to approach each other only later (Fig. 2b, c). This motivates us to define our line straightening deformation ϑ as follows: We employ Laplacian smoothing to the entire curve as long as its endpoints move apart. Once they stopped moving apart, we fix the position of the endpoints, i.e., employ Laplacian smoothing only to the inner vertices. For robustness, the endpoints are fixed once their distance drops below 90% of the maximum distance from earlier iterations.

That is, during the first phase, we transform all feature line vertices \mathbf{y}_i^k (with $i \in 1, \dots, m$) from smoothing iteration k to iteration $k + 1$ according to

$$\mathbf{y}_i^{k+1} = \mathbf{y}_i^k + \lambda \frac{\sum_{j \in \mathcal{N}(\mathbf{y}_i^k)} w_{ij} (\mathbf{y}_j^k - \mathbf{y}_i^k)}{\sum_{j \in \mathcal{N}(\mathbf{y}_i^k)} w_{ij}}, \quad (4)$$

with λ being a smoothing parameter, $\mathcal{N}(\mathbf{y}_i^k)$ the polyline vertex neighbors of vertex \mathbf{y}_i^k , and respective weights $w_{ij} = \frac{1}{\|\mathbf{y}_i^k - \mathbf{y}_j^k\|}$. Please note that we employ implicit Laplacian smoothing (Desbrun et al. 1999) to avoid excessive number of iterations, but keep the explicit notation for simplicity. During the second phase, we employ Eq. 4 to all inner points \mathbf{y}_i with $1 < i < m$, but do not update the endpoints, i.e.,

$$\mathbf{y}_1^{k+1} = \mathbf{y}_1^k \quad \text{and} \quad \mathbf{y}_m^{k+1} = \mathbf{y}_m^k. \quad (5)$$

In total, for an original feature line vertex $\mathbf{y}_i = \mathbf{y}_i^0$, our line deformation is

$$\vartheta(\mathbf{y}_i, \tau) := \mathbf{y}_i^{\lfloor \tau / \Delta\tau \rfloor}, \quad (6)$$

with $\Delta\tau = 1/s$, and s the number of smoothing iterations. Notice that depending on the frame rate, not every smoothing iteration results in a frame of the animation. This is necessary to achieve smooth animations, for which we map the frames nonlinearly to τ . This is motivated by the observation that in Laplacian smoothing, early smoothing steps (close to $\tau = 0$) have a larger influence than later steps (close to $\tau = 1$), exhibiting logarithmic decay (see Fig. 3). As demonstrated with our Circular Segment example (Fig. 1), our two-phase approach maintains reasonable line lengths, leading to better ϕ .

Finally, in cases where the input line exhibits “zigzag perturbations”, slight pre-smoothing should be applied for aesthetic reasons and a more smooth resulting ϕ . Such perturbations can arise, e.g., due to inaccuracy of local extraction approaches, such as the parallel vectors operator (Peikert and Roth 1999). We limit these corrections to be smaller than the size of a grid cell, and thus to a permissible range regarding the involved accuracy of line feature extraction algorithms. We employed such pre-smoothing in our vortex core lines examples (cf. Sect. 6.2).

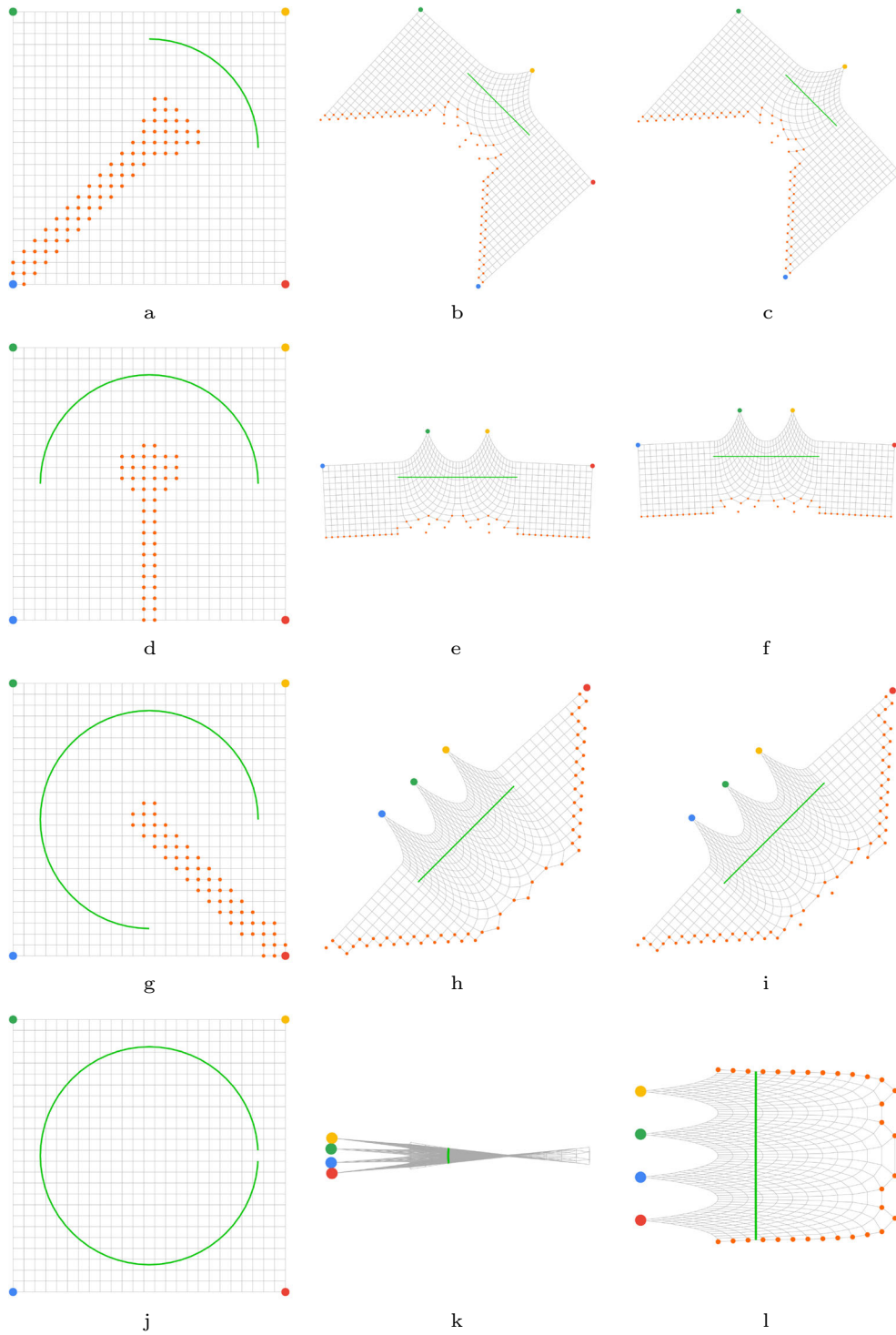


Fig. 1 Left column: original, undeformed grid (with marked corners) and feature line (green). Center column: deformation with fixed endpoints. Right column: deformation allowing endpoints to move apart. Grid lines serve as context, and orange dots indicate tearing of the grid

3.2 Space deformation ϕ

Now that we have defined the line deformation ϑ for the vertices of \mathcal{L} , we need to “transfer” that deformation to the entire domain Ω , considering Requirement R3 (continuity) and Requirement R4.

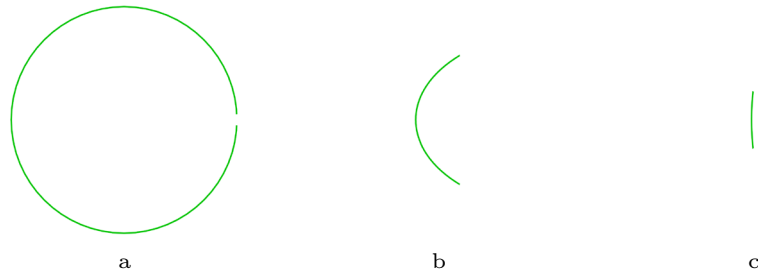


Fig. 2 Open circle in different stages of Laplacian smoothing. **a** Original circle cut open on its right side, **b**, **c** smoothed “circle” after an increasing number of iterations. Observe the initial separation of the endpoints, the reduction of curvature, and the overall shrinking of the smoothed line

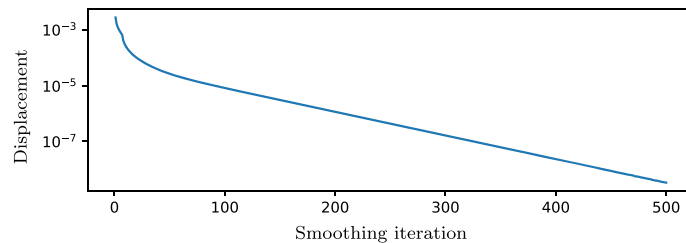


Fig. 3 Displacement from Laplacian smoothing plotted over the iteration, showing logarithmic decay. Please note the logarithmic scale of the y-axis

Particularly to prevent self-intersection of the domain (Requirement R4) and for visual demarcation of areas of strong local deformation, we introduce the concept of “tears”. This means that a large part of necessary strong distortion should be concentrated within thin regions resembling codimension-1 manifolds, along which space is torn apart under the action of ϕ . Those strongly distorted parts are “removed” from the visualization, leading to ruptures (tears). This tearing is implemented by imposing a maximum edge length for the grid cells. If deformation results in a too long edge, incident cells are removed and markers inserted at their nodes for respective depiction (orange regions in Fig. 1 and Fig. 4a, b). If there are several separate components of tearing, we assign each a different color to provide better context (Fig. 4c–g). Essentially, these tears separate regions that are influenced by different parts of \mathcal{L} in analogy to Voronoi cells under a nearest neighbor relation on ϕ .

At the same time, the space deformation ϕ should be continuous in space (Requirement R3), except at the tears. Since ϑ carries over to ϕ , our second basic ingredient is therefore to model the polyline \mathcal{L} as a continuous curve $\bar{\mathcal{L}}$. We achieve this by interpreting \mathcal{L} as the control polygon of a B-spline, which provides us with the smooth feature spline $\bar{\mathcal{L}}$. Thus, by using the (τ -deformed) feature line vertices $\vartheta(\mathbf{y}_i, \tau)$ as de Boor points, we define the (τ -deformed) feature B-spline

$$\bar{\mathcal{L}}^\tau(u) := \sum_{i=1}^m \vartheta(\mathbf{y}_i, \tau) N_i^d(u), \quad (7)$$

with degree d , spline parameter u , and B-spline basis functions

$$N_i^r(u) := \frac{u - u_i}{u_{i+r} - u_i} N_i^{r-1}(u) + \frac{u_{i+1+r} - u}{u_{i+1+r} - u_{i+1}} N_{i+1}^{r-1}(u), \quad (8)$$

for $1 \leq r \leq d$, and

$$N_i^0(u) := \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}. \quad (9)$$

Notice that the B-spline $\bar{\mathcal{L}}^0$ does not strictly follow the polyline \mathcal{L} , but since \mathcal{L} is typically sampled at least at grid resolution, and since we use rather low B-spline degree (we use $d = 3$), $\bar{\mathcal{L}}^0$ is a good approximation to \mathcal{L} (represents a spline interpolation). In our approach, we visualize $\bar{\mathcal{L}}^\tau$ instead of $\vartheta(\mathcal{L}, \tau)$ for consistency during animation, and for better shape perception.

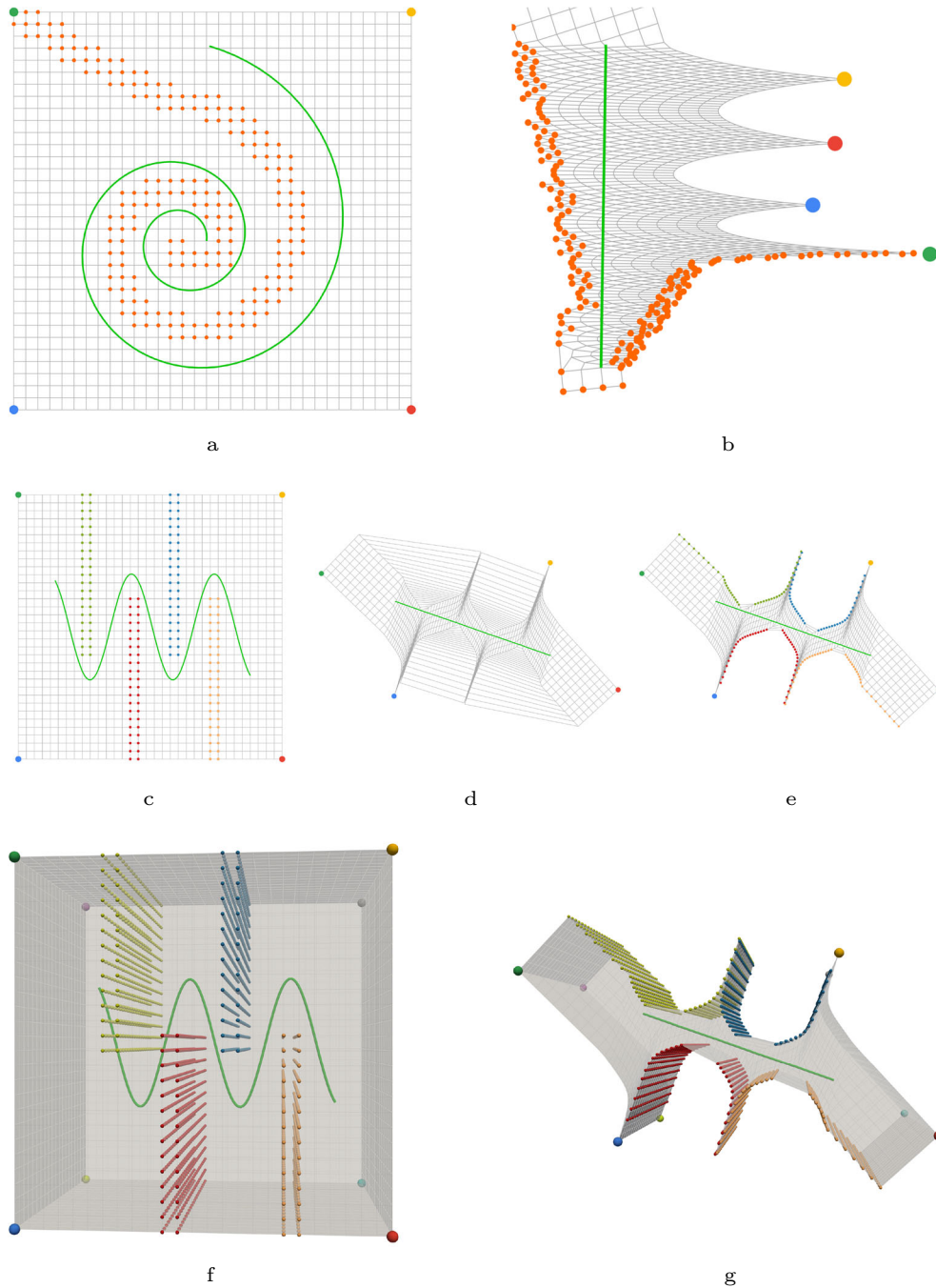


Fig. 4 Deformation process with tearing of the grid. **a** Spiral dataset with tear regions marked by orange dots, and **b** zoom-in on the grid after deformation and tearing. Note that, although much smaller, the individual cells retain a good, convex shape. **c**, **f** 2D and 3D cosine dataset with tear regions in different colors. Each color indicates initially connected grid points, which in the process of deformation move apart. **e**, **g** Grids with torn boundaries or holes, and **d** without. Please see also the supplemental video

To now define the deformation ϕ for a point $\mathbf{x} \in \Omega$, we need to relate it to $\bar{\mathcal{L}}^0$. As our approach follows the idea of spline skinning, the deformation of \mathbf{x} only depends on the transformation at γ_{\min}^0 (red in Fig. 5a), the point on $\bar{\mathcal{L}}^0$ closest to \mathbf{x} :

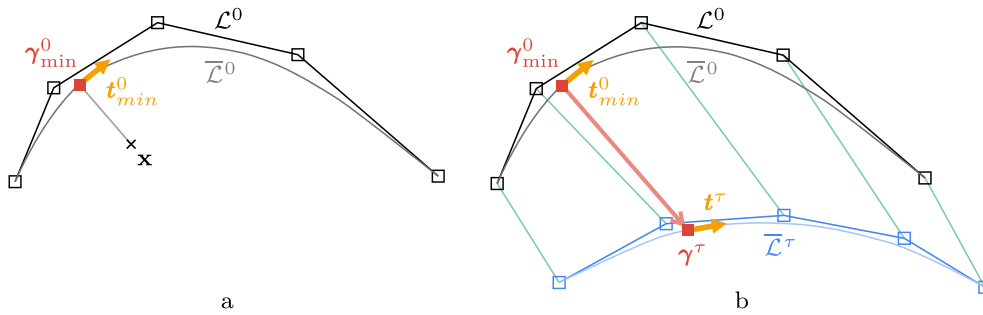


Fig. 5 B-spline approach. **a** First, point \mathbf{x} is mapped onto the closest position γ_{\min}^0 (red square) on the B-spline $\bar{\mathcal{L}}^0$, at $\tau = 0$. **b** The vector (red) from this position to the corresponding position on the B-spline $\bar{\mathcal{L}}^\tau$ at τ provides the translation. The respective tangents (orange arrows) define the rotation

$$\gamma_{\min}^0 := \bar{\mathcal{L}}^0(u_{\min}^0) \quad (10)$$

with

$$u_{\min}^0 := \arg \min_u \left\| \bar{\mathcal{L}}^0(u) - \mathbf{x} \right\|. \quad (11)$$

We obtain this point using binary subdivision on the B-spline, according to Forstmann et al. (2007). If there is more than one such point, it is safe to take the first one. Consequently,

$$\gamma^\tau := \bar{\mathcal{L}}^\tau(u_{\min}^0) \quad (12)$$

provides the point (cf. Fig. 5b) on $\bar{\mathcal{L}}^\tau$, where γ_{\min}^0 is deformed to at time τ . In other words,

$$\phi(\gamma_{\min}^0, \tau) \equiv \gamma^\tau. \quad (13)$$

The only remaining task, is to “transfer” the deformation $\phi(\gamma_{\min}^0, \tau)$ from γ_{\min}^0 to $\mathbf{x} \in \Omega$. For spline skinning methods, this is typically achieved by matching the Frenet–Serret frame of \mathcal{L}^0 at γ_{\min}^0 to the corresponding frame of $\bar{\mathcal{L}}^\tau$ at γ^τ , and applying this transformation to \mathbf{x} . However, this could obscure, e.g., the dynamics of vortex core lines, as the winding number of the surrounding streamlines could be altered, thus violating Requirement R5 (see also Sect. 4.2). Therefore, we constrain the transformation to the translation $\gamma^\tau - \gamma_{\min}^0$ (light red arrow in Fig. 5b), and the minimum rotation required to match the B-spline tangent \mathbf{t}_{\min}^0 to \mathbf{t}^τ (orange arrows in Fig. 5b). Hence, the rotation axis is given by $\mathbf{t}_{\min}^0 \times \mathbf{t}^\tau$, which, together with angle $\angle(\mathbf{t}_{\min}^0, \mathbf{t}^\tau)$, defines the required rotation matrix $R_{\rho_{\min}, \mathbf{t}^\tau}$. Thus, we get for point \mathbf{x} :

$$\begin{aligned} \phi(\mathbf{x}, \tau) &:= \left(R_{\rho_{\min}, \mathbf{t}^\tau} \cdot (\mathbf{x} - \gamma_{\min}^0) + \gamma_{\min}^0 \right) + (\gamma^\tau - \gamma_{\min}^0) \\ &= R_{\rho_{\min}, \mathbf{t}^\tau} \cdot (\mathbf{x} - \gamma_{\min}^0) + \gamma^\tau. \end{aligned} \quad (14)$$

With the resulting deformation ϕ , we can now deform any geometry, such as the geometry of the underlying grid, or the streamlines obtained from the original vector field in Ω . This also includes the straightening of tubular domains in medical applications, as mentioned in Sect. 2. However, as motivated in the introduction, to be able to employ, e.g., texture-based flow visualization, it is necessary to also transform the vector field. This is achieved by multiplying its vectors with $\nabla \phi$, using analytic derivatives of Eq. 14. To preserve divergence, we additionally divide the resulting vector field by the determinant $\det \nabla \phi$.

4 Deformation properties

Before we discuss the geometric properties of our deformation ϕ , let us briefly discuss simpler alternative approaches, which, however, do not meet the identified requirements and hence do provide a motivation for our approach. First, inverse distance weighting, would map each point $\mathbf{x} \in \Omega$ to its deformed state $\tilde{\mathbf{x}} \in \tilde{\Omega}$, by applying a weighted average of the translations that move the feature line vertices \mathbf{y}_i^0 to their deformed positions \mathbf{y}_i^τ . However, the global influence would cause regions close to \mathcal{L} to be affected by

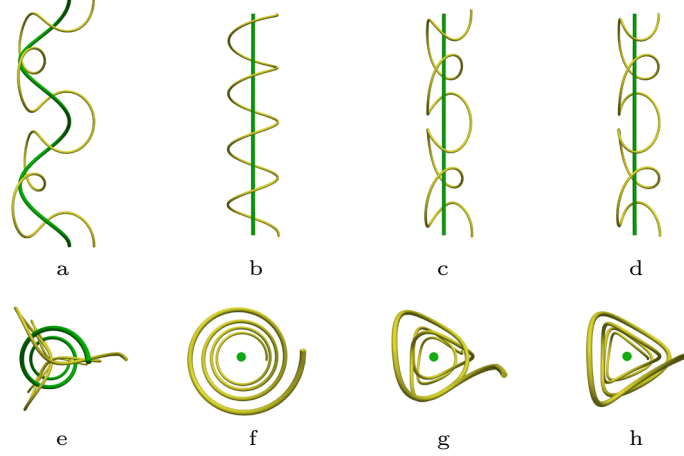


Fig. 6 Swirling Helix example for comparison between our B-spline approach and inverse distance weighting. **a, e** Synthetic dataset of a helix swirling around another helix. **b, f** Result from our B-spline approach, while **c, g** inverse distance weighting, and **d, h** our B-spline approach without considering rotation, lead to unwanted distortion

strong and “contradicting” transformations (Requirement R5, cf. Fig. 6c, g), leading to strong distortions and self-intersection of Ω (Requirement R4). Beyond that, only the vertices of \mathcal{L} would influence the deformation of space, i.e., the segments between the vertices would have no influence. Similarly, transferring only the translation $\gamma^\tau - \gamma_{\min}^0$ in the B-spline approach produces undesired deformations as well (—see Fig. 6d, h).

4.1 Requirements

As per definition of our approach for straightening the feature line, Requirements R1 and R2—straight line at $\tau = 1$, and continuous transition from $\tau = 0$ —are already met. This leaves us with the discussion on the requirements for space deformation (Requirements R3, R4 and R5), and for its efficient computation (Requirement R6). The latter is achieved by moving costly calculations to a preprocessing step, and employing the GPU (cf. Sect. 5).

The continuity of the feature spline ($d > 1$ for $\bar{\mathcal{L}}^\tau$), the continuity of its deformation ϑ , and the continuity of the mapping of a point in space to the closest point on the spline, cause the space deformation ϕ to be also continuous. Therefore, Requirement R3 is met, except at the tearing manifolds, as stated in the requirement. Figure 7 shows the continuity of the closest-point mapping for two examples, and that discontinuities only occur at tearing manifolds.

To show that our method satisfies Requirement R4 (no self-intersection), we could parametrize our three-dimensional space Ω by an infinite number of two-dimensional (possibly bounded) subspaces, each of these planes (black lines in Fig. 8a) being perpendicular to the feature line $\bar{\mathcal{L}}^0$ (green), i.e., their normals are parallel to the feature line tangent t_{\min}^0 at their intersection γ_{\min}^0 . Thus, these subspaces can be defined as

$$\mathcal{P}_{\gamma_{\min}^0} := \left\{ \mathbf{x} \mid t_{\min}^0 \cdot (\mathbf{x} - \gamma_{\min}^0) = 0 \right\}. \quad (15)$$

Every plane $\mathcal{P}_{\gamma_{\min}^0}$ then spans a two-dimensional local coordinate system, bounded by the intersection with other planes (orange and blue tearing markers in Fig. 8a). After deformation ϕ , planes in $\bar{\Omega}$ (black lines in Fig. 8b) are still perpendicular to the feature. This means that (I) a point on a plane $\mathcal{P}_{\gamma_{\min}^0}$ in the original space Ω stays on this plane $\mathcal{P}_{\gamma_{\min}^\tau}$ in deformed space $\bar{\Omega}$ after transformation. Further, (II) any two points on the same plane retain their relative position to each other and to the feature line, i.e., their polar coordinates in plane space do not change. Our approach can therefore be described in terms of rigid body transformations of subspaces. Because all planes are parallel in $\bar{\Omega}$ for $\tau = 1$, i.e., they do not intersect each other, self-intersection of cells, non-convex cells, or degenerate cells cannot occur, hence satisfying Requirement R4. Another property of our approach is that beyond the endpoints of a feature line, all points are transformed in a volumetric rigid body manner, as they all map to feature endpoints (gray arrows in Fig. 8).

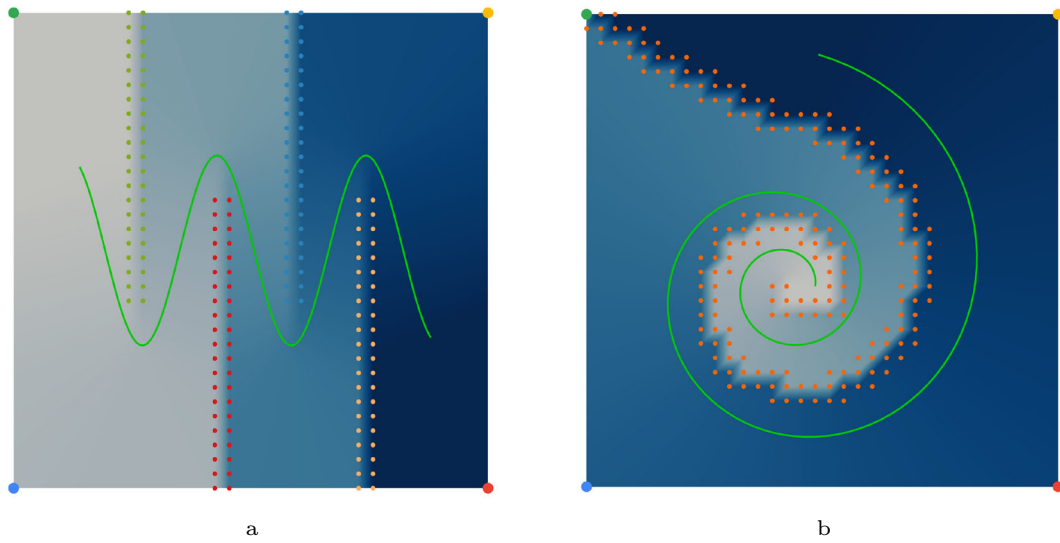


Fig. 7 B-spline parameter u_{\min}^0 from closest-point mapping, with increasing values from gray to blue, for **a** 2D cosine and **b** spiral dataset. Tearing manifolds marked by colored circles coincide with discontinuity in u_{\min}^0 . Note that blending at discontinuities is caused by piecewise linear depiction

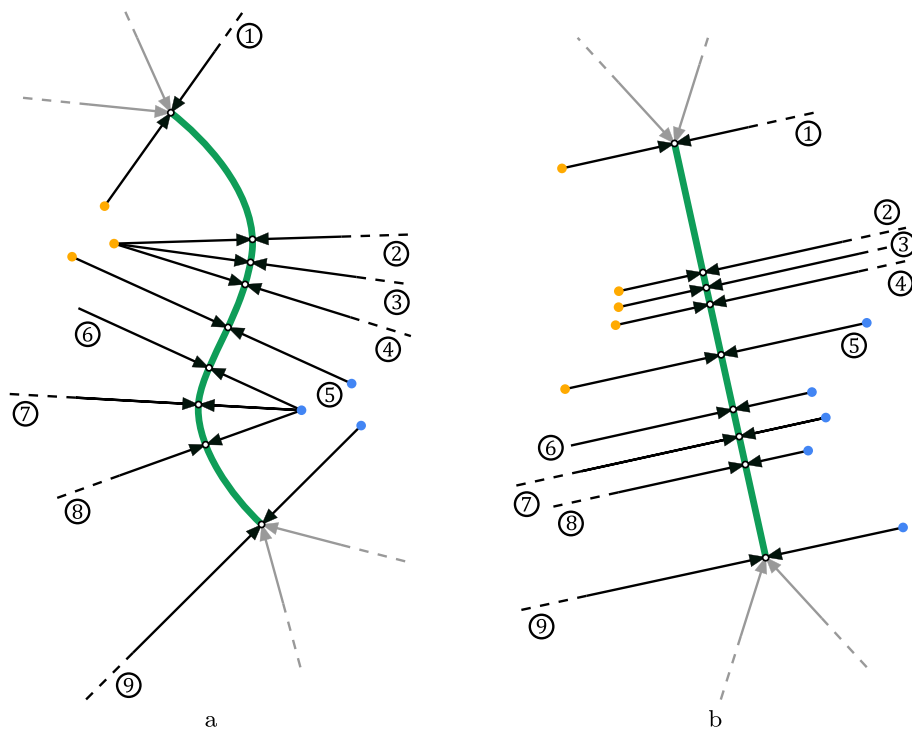


Fig. 8 Parametrization of 2D space by one-dimensional subspaces (black) for **a** original space Ω , and **b** deformed space $\tilde{\Omega}$ with straightened feature line (green). Arrows represent closest-point mapping and show that all points in a subspace are mapped to the same point on the feature line. Dashed lines indicate that subspaces are infinite in the corresponding direction, whereas dots (orange and blue) indicate tearing manifolds. Regions beyond the feature line (gray) are areas where deformation results in rigid body transformation, as all points are mapped to the feature line endpoints. Numbers indicate correspondence between original and deformed space. Note that all subspaces are perpendicular to the feature line

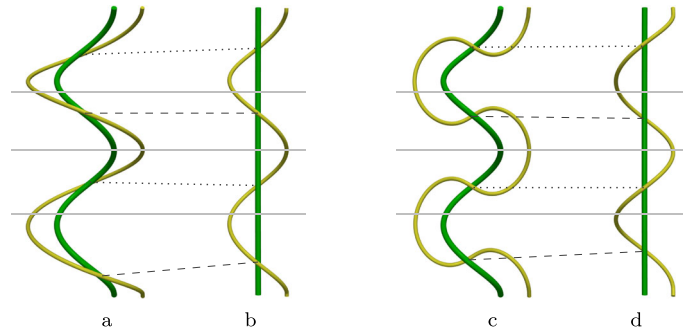


Fig. 9 Feature helix (dark green), surrounded by a second helix (light green) winding around the feature. Winding numbers around original features **a**, **c** are preserved for straightened features **b**, **d**, indicated by crossings in front (dashes) and behind (dots) the feature line

4.2 Application to vortex core lines

In order to meet Requirement R5, that is, to avoid ϕ to contain deformation that would obscure feature-related dynamics, i.e., dynamics related to vortex core lines, we want to keep the deformation small and constrained, and preserve properties that describe the characteristic flow behavior in a vortex neighborhood. Due to the swirling property of a vortex, a straightforward step is thus to additionally preserve the winding number of the flow. This winding number is defined for two-dimensional domains as a number that counts how often a streamline does a full revolution around a specific point in space. Similarly, we can define the winding number in 3D space, akin to Sujudi and Haines (1995), as swirling motion projected onto planes, which are perpendicular to the vortex core line. For this, our approach already preserves the winding number, as the rotation induced by our deformation minimizes to $R_{\min}^{\phi, \epsilon}$, which exhibits no rotation around the feature line tangent. This is illustrated in Fig. 9 for two simple examples that exhibit the same number of “crossings” before and after deformation. Note that for other line features, adjustments to our method could be made in order to preserve their specific properties.

5 Implementation

Our approach is implemented as a ParaView (Ayachit 2015) plugin written in C++. The input of our plugin consists of the feature lines, the vector field grid, and any geometry from the ParaView pipeline, such as streamlines. As our technique should be interactive, we employ a caching strategy for every step of the algorithm. For this, we define every step as a sub-algorithm. Thus—following the leitmotif of VTK algorithms—a step is only repeated if its input has changed, or else returns the cached result. We especially designed sub-algorithms for the precomputation steps that only have to be executed once before interaction. This entails the mapping of points $\mathbf{x} \in \Omega$ onto \mathcal{L}_0 , and identifying tears.

To achieve interactivity, we employ CUDA (Nickolls et al. 2008) for parallel computations on the GPU. Because the deformation ϕ is embarrassingly parallelizable, it is performed on the GPU. Further, uploads to, and downloads from GPU memory are reduced to a minimum. Because (1) Laplacian smoothing is efficient to compute, (2) displacement computation on the GPU is embarrassingly parallelizable, and (3) the analytical solution of a B-spline of fixed degree can be computed efficiently, Requirement R6 is met. Still, the main bottleneck for seamless animation is the output of the computed results, due to the conversion of data to VTK objects. Therefore, rendering the data directly on the GPU without creating explicit geometry could be addressed as future work.

The complete code for the ParaView plugin is available on GitHub under the MIT license (<https://github.com/UniStuttgart-VISUS/feature-deformation>).

6 Results

In the following, we apply our approach to synthetic and simulated flow field examples for different types of feature lines. Our results have been obtained by deforming geometry (not the field), employing third-order

B-splines, using ten binary search iterations for determining the nearest point on the spline, and removing cells if they become two times larger than the original cell diagonal.

6.1 Attracting features

For two-dimensional vector fields, occlusion and visual inspection of streamlines near feature lines is generally not an issue. However, assessing the attracting behavior of such features becomes hard if they are curved. Our method can mitigate this issue and bring insight on the qualitative aspect of attraction.

Figure 10 shows the Attracting 2D Periodic Orbit dataset with a radially symmetric vector field that contains an attracting periodic orbit. First, the closed feature line needs to be opened by cutting a segment. This leads to the tearing manifold marked by the orange dots in Fig. 10b. Because of the strong local deformation caused by the circular shape of the feature line, tearing is unavoidable. The resulting deformed configuration in Fig. 10c exhibits a very regular pattern, with decreasing cell size from left (originally inside the periodic orbit) to right (outside the orbit). Additionally, one can observe that in Fig. 10b, the incident angles of the streamlines onto the periodic orbit appear to be different from its inside and outside. However, our method reveals in Fig. 10c that the dynamics are the same on both sides of the periodic orbit. This is in line with the definition of the synthetic vector field, which was designed to exhibit the same attraction on

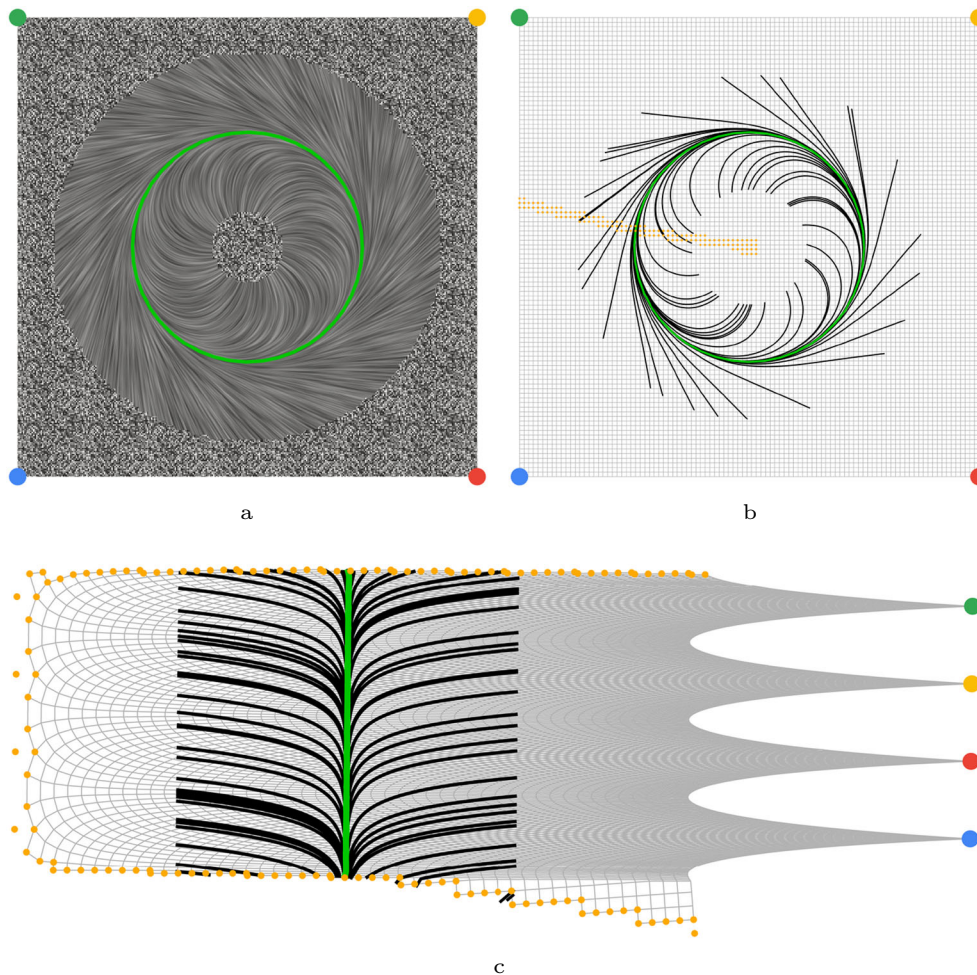


Fig. 10 Feature-based deformation for the Attracting 2D Periodic Orbit example. **a** Line integral convolution with extracted periodic orbit feature line (green curve) shows overall structure. **b** Original (undeformed) data grid and forward streamlines (black) seeded in the vicinity of the periodic orbit show transport and attraction toward it, but make it hard to judge the quality of attraction, i.e., if attraction is stronger from the inside or outside. **c** Deformed grid and streamlines for straightened feature line reveal that attraction is equal on both sides of the orbit. Orange dots indicate tearing, and corner marks help interpreting overall deformation

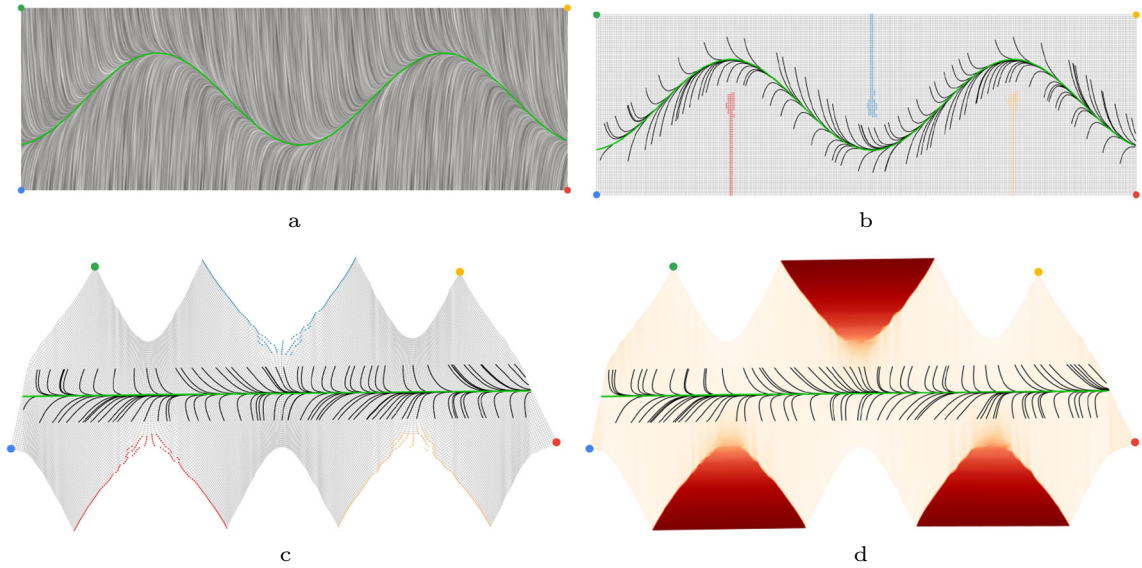


Fig. 11 Oscillating Attachment Line example. **a** LIC reveals sine-like oscillations of an attachment line (green). Flow in its vicinity visualized by streamlines (black) for the original **b** and deformed vector field **c**. Colored manifolds indicate separate tearing regions. **d** Logarithmic mapping to magnitude of deformation by encoding the Frobenius norm of $\nabla\phi$ (beige to red)

both sides. Please also see our user study in Appendix A, assessing the visual perception of convergence to a straightened or circular feature line.

Unlike in the previous example, looking at the streamlines in Fig. 11b for the synthetic Oscillating Attachment Line example, one assumes that their incident angles are different on either side. This time, the impression is confirmed when consulting the straightened configuration (Fig. 11c). We identify a recurring pattern on both sides, which seems to correlate with the curvature of the original attachment line. While this observation can be made already from Fig. 11b, our approach confirms it and provides an explicit view on its angular configuration. Because the grid is not deformed as strongly as in the previous example, due to the shape of the feature line, there are only small tears. To further highlight that by tearing the grid (and thus removing strongly deformed cells) we gain an indication for where deformed streamlines are reliable, we show the magnitude of deformation in Fig. 11d. For this, we map $\|\nabla\phi\|$ to color. It is clearly observable that, with our grid tearing approach, regions of strong local deformation are removed (cf. Fig. 11c). Thus, such color coding is not necessary, as the tearing of the grid already provides us with this information.

6.2 Vortex core lines

As discussed in Sect. 4.2, deforming vortex core lines is a target application of our approach, in the sense that essential geometric properties, as well as the winding number of streamlines around the core lines are preserved. Thus, it enables dedicated visual analysis of datasets exhibiting curved vortex cores.

In the introduction, we argued that the winding of the streamlines around a bent vortex with low “twist” around its core but with high tangential velocity is difficult to observe. Figure 12 shows such flow of the synthetic Spiral Vortex dataset. There, the streamlines close to the vortex core line exhibit no apparent swirling motion about the core line. Only at the bottom part of the spiral, such swirling is observable from the original configuration (Fig. 12a). Straightening the feature line and deforming its surrounding streamlines (Fig. 12b), however, clearly reveals the swirling motion around the vortex core line. To further highlight this, a streamline is highlighted (red). Notice that the number of crossings—the streamline either crosses in front or behind the feature line—matches for the original and the deformed space. While this shows that the winding number is preserved, it is easier to observe in the deformed space, especially when visualizing multiple streamlines. Thus, this enables visual denomination of the winding number and provides better insight into the flow dynamics close to the vortex core line. Particularly, the change in swirling motion along the vortex stands out, with stronger swirling motion at the bottom of the domain and reduced swirling at the top. While some parts of the grid are torn, the feature line and its surroundings are well preserved.

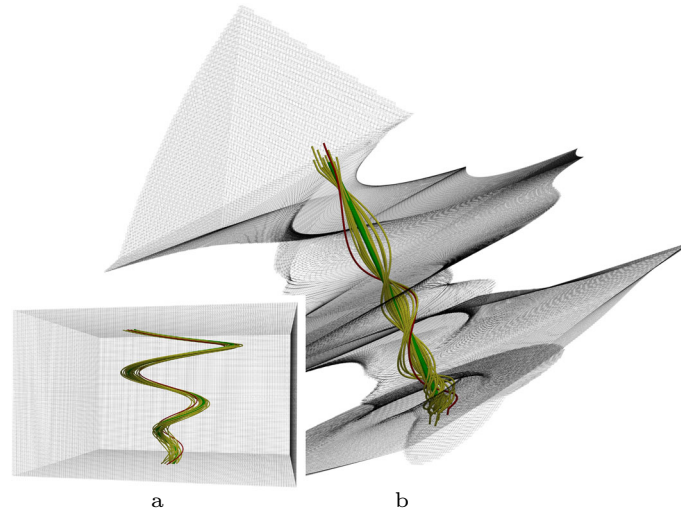


Fig. 12 Synthetic Spiral Vortex example. Vortex core line (dark green) and streamlines nearby (light green) for the original **a**, and for the deformed space **b**. The latter was partly torn apart, vortex core line and streamlines remaining enclosed by the grid. Please also notice the windings of the red streamline around the feature

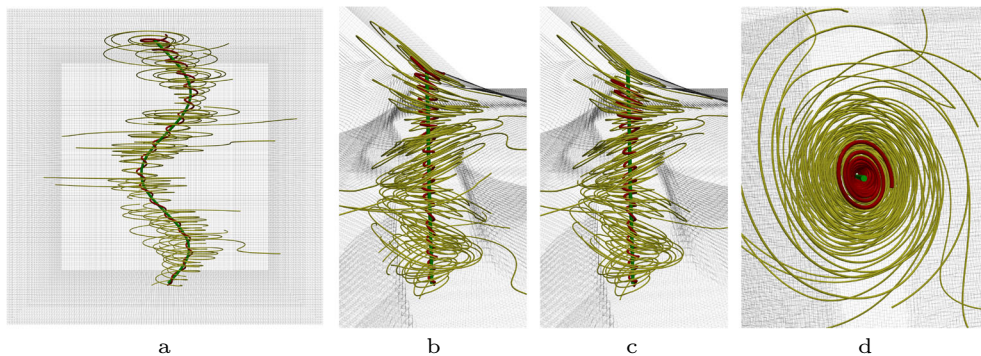


Fig. 13 Flow in the Tornado (Crawfis 2003) dataset with vortex core line (dark green) and streamlines (light green). Additionally, a streamline close to the vortex core line is highlighted (red) to show the difference in deformation between close streamlines and the ones farther away. **a** Original space, **b**, **c** deformed space, and **d** deformed space with view along the straightened feature line. Comparison between **b** deformed streamlines, i.e., deformed geometry, and **c** streamlines seeded in the deformed vector field

The synthetic Tornado dataset (Crawfis 2003) shows a near opposite behavior as the Spiral Vortex dataset, with low tangential velocity and a large winding number due to strong swirling motion. Here, the extracted vortex core line can be interpreted as the “eye of the storm”. A very peculiar behavior of the tornado is the apparent motion visualized by the streamlines in Fig. 13a, which is mainly horizontal. Hence, the streamlines seem to behave independently from the curvature of the feature line. Further, an upward motion can barely be seen for the streamlines farther away from the feature. Only for the streamline very close to the vortex core (red in Fig. 13), such a trend can be observed. The horizontal motion of the streamlines, together with the curved vortex core line, therefore leads to askew-looking streamlines in the deformed configuration (Fig. 13b). This is an interesting behavior, which originates from the used mathematical model to create the vector field, whereas in physical models, motion relative to the vortex core line is typically regular and dependent of the shape of the vortex core line. To additionally show that there is no significant difference between deformed streamlines (Fig. 13b) and streamlines computed on the deformed vector field (Fig. 13c), we visualize both for comparison. One can observe that close to the feature line there is no apparent difference. Farther outside, however, the accumulated error increases, as cells far away from the feature are generally more deformed, and thus more prone to numerical inaccuracy.

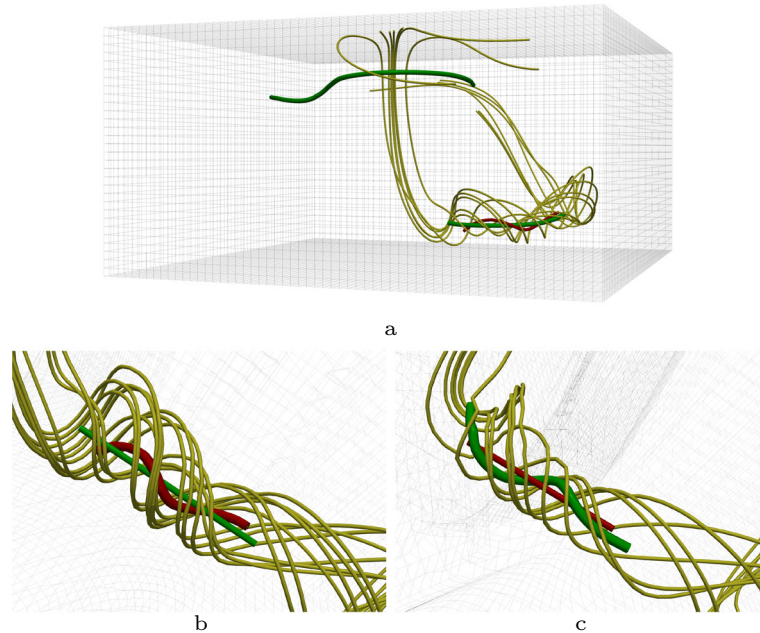


Fig. 14 Convective Heat Flow with two vortex core lines (dark green), surrounding streamlines (light green) and selected streamline (red) in the undeformed configuration **a**. Zoom-in on the deformed configuration after **b** straightening the lower vortex core line, and **c** straightening the selected streamline, respectively. Note that after deformation in **c**, other streamlines and the lower vortex core line are swirling around the straightened streamline

Our approach is not only able to straighten locally defined feature lines, but lines in general. To exploit this for further analysis of the flow in a vortex region, we can, e.g., straighten arbitrary streamlines as well. In Fig. 14, showing a time step from a convective heat flow simulation with two vortex core lines, the space is deformed by straightening the lower vortex core line (Fig. 14b) or an arbitrary streamline (Fig. 14c). While for the deformation of the vortex core line, we get again insight into the flow relative to it, we can see that by deforming the streamline instead, the lower vortex core line now swirls around that streamline. This is expected, as by defining the respective other line as reference, the swirling still has to be present. However, other neighboring streamlines also rotate around our feature streamline. This is an interesting observation, which is not obvious, and not visible in the original dataset, demonstrating that considering streamlines as features can provide insight into flow dynamics, meriting further investigation.

Another benefit, by generally deforming the space and not only the vector field, is that we can deform arbitrary geometry. The Centrifugal Pump dataset (Lucius and Brenner 2010), provided in the scope of the 2011 SciVis Contest, consists of the flow field and the geometry of a pump. The original, undeformed geometry, together with a vortex core line and surrounding streamlines, is depicted in Fig. 15a, b. It shows not only the flow lines, but also the casing of the pump and its blades. Thus, additional to the streamlines, the pump itself is deformed using our approach. This way, the pump can still be visualized for context, as shown in Fig. 15c, d. Straightening the vortex core line (Fig. 15c) shows again the swirling motion of the streamlines around it. Similarly, vortex cores in turbines often exhibit low numbers of revolution (cf. Roth and Peikert 1996), which constitutes a good application case for our method. Again straightening a selected streamline allows to observe the motion of the vortex core line and the other streamlines in its vicinity (Fig. 15d). Like in the previous example, the other streamlines exhibit swirling motion around our feature streamline.

6.3 Attachment and separation lines

The Blunt Fin dataset shown in Fig. 16 represents an air flow simulation around a fin, defined on a curvilinear grid. The fin causes the formation of attachment and separation lines along the surface, between which swirling motion is present. In the original space in Fig. 16b, the overall impression is that the streamlines “pass by” the two feature lines. In our deformed representation in Fig. 16d, which was obtained

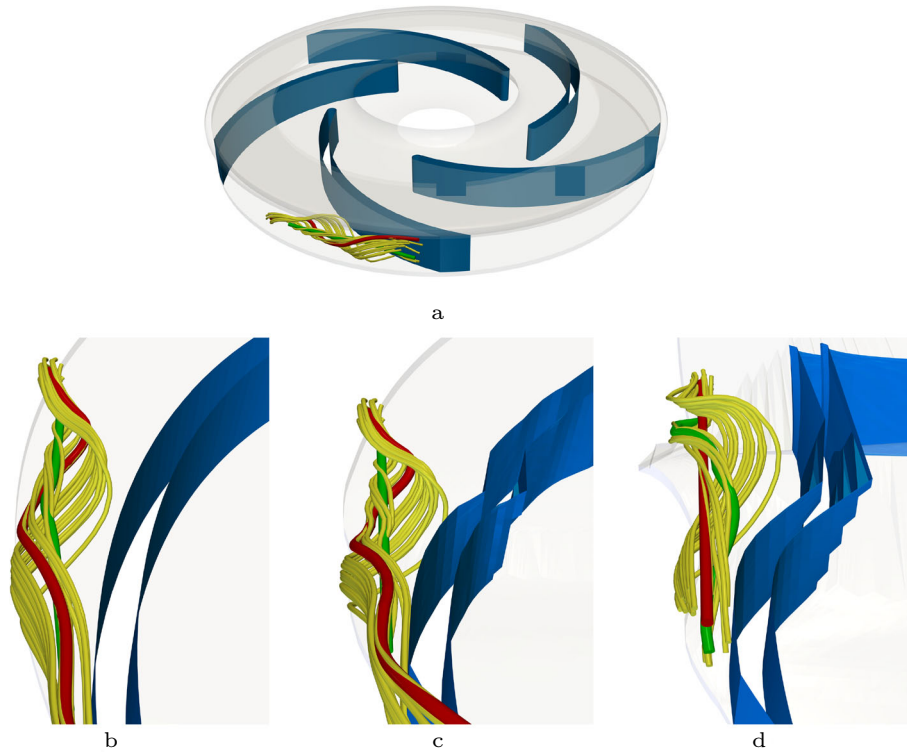


Fig. 15 Flow within the Centrifugal Pump, whose geometry—casing (transparent gray) and blades (blue)—is shown. **a**, **b** Original space and undeformed geometry, containing a vortex core line (dark green), a selected feature streamline (red), and surrounding streamlines (light green). Deformed space and geometry after straightening the vortex core line **c**, or the feature streamline **d**, respectively

by straightening the lower separation line, attachment and separation flow behavior can be seen much clearer. Additionally, the swirling behavior between the two feature lines can now be better observed by looking along its swirling path. As, in this case, there is flow on a surface, line integral convolution can be employed for visualization. For this, we deform the grid and corresponding vector field, demonstrating that our method can readily be used in such situations. Here, the original (Fig. 16a) and the deformed configurations (Fig. 16c) are shown, respectively.

6.4 Bifurcation lines

Taylor–Couette flow is induced by two nested cylinders rotating at different velocities. This leads to oscillating bifurcation lines (Roth 2000), as can be seen in Fig. 17a. Visualizing the flow close to the bifurcation lines, as in Fig. 17b and for a zoomed-in region in Fig. 17d, does not reveal much detail about separating flow behavior. Deforming one of the feature lines, however, reveals the expected saddle-like behavior. This is shown in Fig. 17c, and in Fig. 17e for a view along the straightened bifurcation line. As before, preserving certain geometric properties is of vital interest. Among those is the “plane property”, as described in Sect. 4.2. Contrary to vortex core lines, however, here we are not interested in preserving the winding number. To observe the flow relative to a bifurcation line, we would rather restrict the flow close to the feature to single flat (stream-)planes. This alignment would then provide us with a local view on the saddle behavior. To this end, an approach could be to additionally “twist” the space along the feature by rotating the already transformed planes \mathcal{P}_{γ^f} around their normal, i.e., around the feature line tangent \mathbf{t}^f .

7 Discussion

In the previous section, we showed that our method can be readily applied to vector fields for flow analysis. Qualitative interpretation of such vector fields is directly possible when visualizing the deformed geometry,

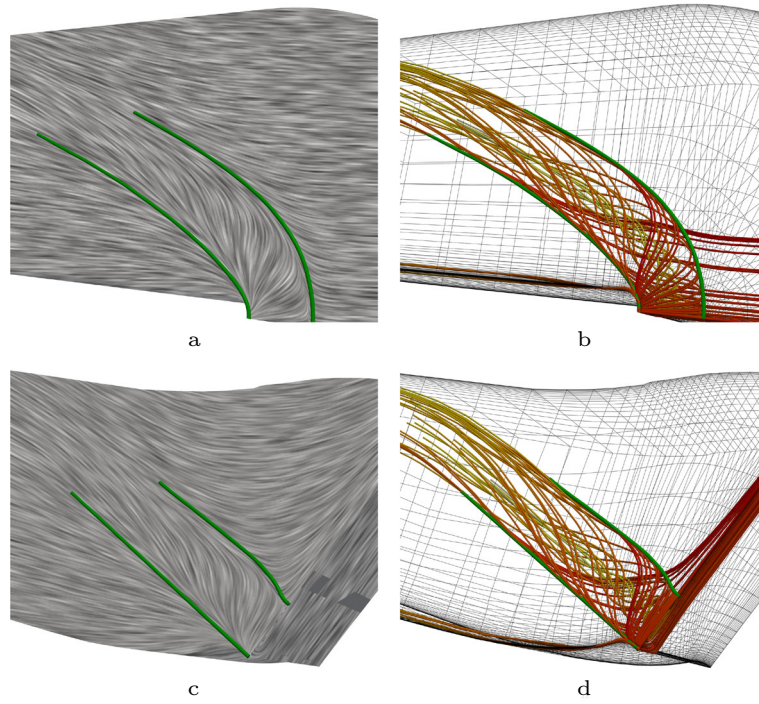


Fig. 16 Blunt Fin dataset with attachment and separation lines (dark green). **a** LIC and **b** streamlines between the two feature lines in the undeformed space. **c** LIC for the deformed grid and **d** respective streamlines. Integration time is color-coded to provide the flow direction (from red to yellow)

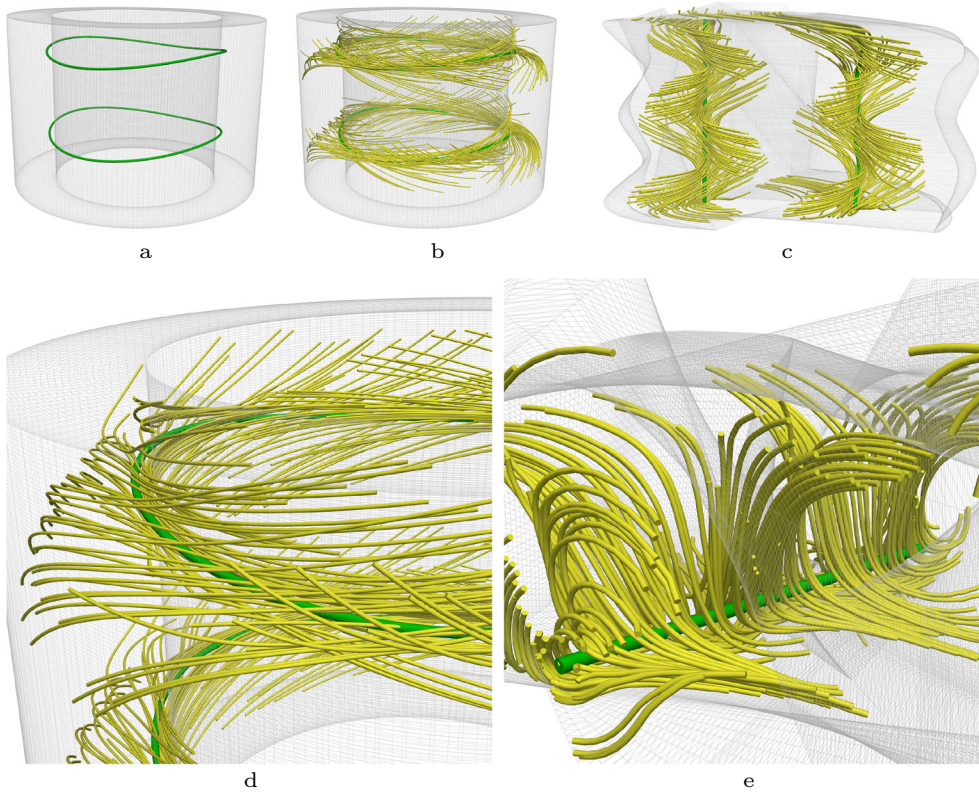


Fig. 17 Taylor-Couette flow between two rotating cylinders, with **a** two oscillating bifurcation lines (dark green) near the wall of the inner cylinder. **b** Original streamlines (light green), **d** close-up, and **c**, **e** deformed

i.e., the straightened feature line and its deformed neighborhood. Additionally, for preserved geometric properties, quantitative interpretation is applicable as well. Examples presented are the number of windings for vortex core lines, but also more generally the distances of the streamlines to the feature, and relative point positions in subspaces $\mathcal{P}_{r_{\min}}^o$. In cases where properties are to be analyzed that are not preserved, such as physical properties, we recommend to employ visualizations of the original and the deformed space for complimentary analysis.

While closed feature lines, such as periodic orbits (cf. Sect. 6.1), are natural limitations of our line straightening approach, we encourage to just “cut open” the line for straightening. Similarly, applying appropriate preprocessing to the feature line, it is possible to, e.g., only straighten parts of the feature. This can be done by cropping the line and feeding our algorithm only the relevant part as input. Finally, due to the fact that line straightening and space deformation parts can be easily exchanged in our modular approach, easy adaptation to other feature line types is possible, or to map an input feature line to a predefined target feature line instead of applying regular smoothing.

In general, only cases where the user is interested in regions that are affected by strong distortion and possibly tearing, our approach has its limitation. Such features might include, but are not limited to, knots. Because of their narrow winding, only very limited space is not affected by said distortion and thus may not yield useful results. This is due to the Voronoi-like property of our point mapping. Otherwise, the complexity of a feature line does not negatively affect the capabilities and usefulness of our method.

Because smoothing and deformation is currently not constrained to the spatial domain in unsteady flow, our method would result in space-time deformation. While this impedes interpretation, our method could be applied to time-dependent datasets to analyze the temporal evolution of vortex core lines by applying line straightening and deformation to the core line and its surrounding streamlines (neglecting time). This way, by mapping the vortex core line always to the same straight target line, similar to the idea of registration, streamlines could easily be compared between time steps side-by-side.

8 Conclusion

We presented the concept of interactive straightening of line-type flow features for improved inspection of the dynamics that cause them. For that, we introduced a new spline-based deformation scheme, which employs “tears” to enable strong local deformations but keep distortion low. We discussed the properties of the deformation, with emphasis on the deformation of vortex core lines for which our approach is well suited in the sense that it preserves the characteristics (winding) of vortical flow. To enable interactive rates, we implemented our approach on the GPU, and integrated it into the visualization platform ParaView for improved usability and availability. We evaluated our approach using various synthetic and simulated fields, as well as for different line features, such as attachment/separation lines, vortex core lines, and bifurcation lines. The overall approach allows one to observe and assess the flow—its behavior and patterns, and related phenomena—relative to the feature line. As such, it can be used in scenarios where mostly qualitative analysis is required, e.g., when teaching features in flow visualization to students or non-experts.

As future work, we propose to conduct a study to evaluate and discuss our method with domain experts. Further, we would like to provide specialized deformations for other feature lines, and extend the approach to surface features. This would enable the investigation of flow relative to separation surfaces and stream surfaces, but also to three-dimensional Lagrangian coherent structures. Because the application to unsteady flow leads to space-time deformation, we would like to constrain our approach to spatial deformation.

Appendix A: User study

To evaluate the concept of visualizing flow relative to a straightened feature line, we conducted a small online user study. In this study, the goal is to verify three hypotheses, stating that evaluating the convergence of streamlines approaching a feature line from both sides is (I) more reliable, i.e., less error prone, for a straightened feature than for a curved one, (II) it requires less convergence difference for a correct assessment, and (III) less time is needed to assess the convergence. These three hypotheses are formulated into our research questions:

RQ 1: Is the assessment more reliable?

RQ 2: Does the assessment take less time?

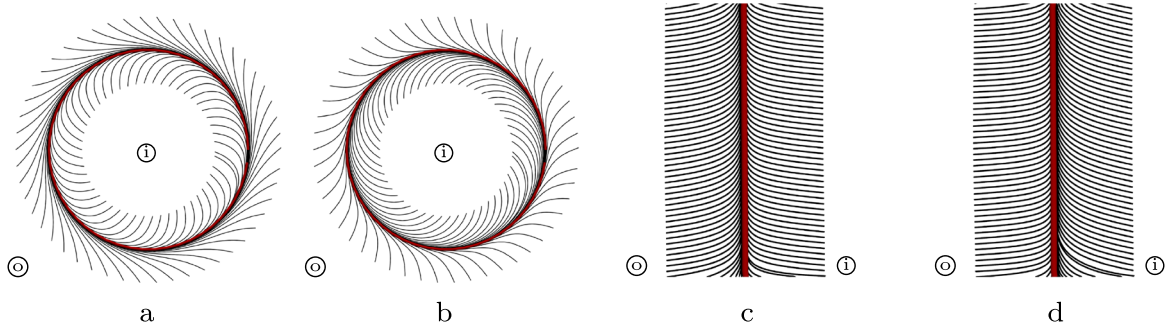


Fig. 18 Experiment setup with streamlines (black) converging to a feature line (red). The streamlines in the original field show stronger convergence on the inside **a**, and stronger convergence on the outside **b**. After straightening the feature line, stronger convergence can be observed on the right **c**, and on the left side **d**, respectively. The inside of the circle corresponds to the right side after deformation (i), the outside corresponds to the left (o)

RQ 3: Does a smaller convergence difference suffice?

Please note that the research questions are formulated from the point of view of the straightened feature.

Experiment design

Our experiment setup resembles a two-alternative forced choice (2AFC) approach, where the participant is presented an image containing two stimuli, i.e., the convergence rate of the streamlines on either side of the feature line (cf. Fig. 18). However, additional to the two choices, stronger convergence on the left side (outside) and stronger convergence on the right side (inside), equal convergence was added as a possible choice. This decision was motivated by RQ 3, to assess the minimum parameter difference needed for recognition. Therefore, for each parameter combination, the participant was required to choose between the three given options.

The experiment was grouped into two parts, each comprised of a set of images with either a straightened or circular feature line. Each group started with a short introduction to the task, followed by several examples. At the end of the instructions, the participants had to finish a task of obvious stimuli as an exercise, which they needed to answer correctly before proceeding. Within the group, all questions were put in a random order to prevent learning effects from introducing bias to the results. Additionally, the participants started randomly with one of the groups, following the same reasoning. Each group consisted of 49 questions, as a combination of the parameters $\kappa_{1,2} \in \{3.3, 4.0, 4.55, 5.0, 5.5, 6.25, 7.5\}$, indicating the convergence rate. The following equation was used to model the flow convergence to the periodic orbit:

$$f(\mathbf{x}) = \frac{\alpha}{|r(\mathbf{x}) - r|^\beta} \cdot \begin{cases} \kappa_1 & r(\mathbf{x}) - r > 0 \\ \kappa_2 & r(\mathbf{x}) - r < 0 \end{cases} \quad (16)$$

where $r(\mathbf{x})$ is the distance of \mathbf{x} to the center of the periodic orbit of radius r , and α and β are arbitrarily chosen constants. An example vector field is visualized using streamlines and showing the feature line (periodic orbit) in Fig. 18a for $\kappa_1 = 3.3$ (outside) and $\kappa_2 = 7.5$ (inside), and in Fig. 18b for $\kappa_1 = 7.5$ and $\kappa_2 = 3.3$. Likewise, the deformed vector field, showing the straightened feature line, is depicted in Fig. 18c for the same parameters as in Fig. 18a, and in Fig. 18d for the same parameters as in Fig. 18b. Every participant was asked to finish *all* tasks. For simplicity and to avoid confusion, the possible answers were kept in the same order.

Participants

The population for the study consisted of 20 participants, five of which are female and 15 are male. Their age ranged between 20 years and 49 years of age, where age was collected in ranges 20 to 29, 30 to 39, and 40 to 49. Five participants indicated to suffer from myopia, two stated to suffer from astigmatism. However, no correlation between impairment, age, or gender and the quality of results could be identified.

Few of the participants have a background in visualization and computer science. Most, however, did not have any domain-specific experience. Further, only one of the participants had expertise in flow

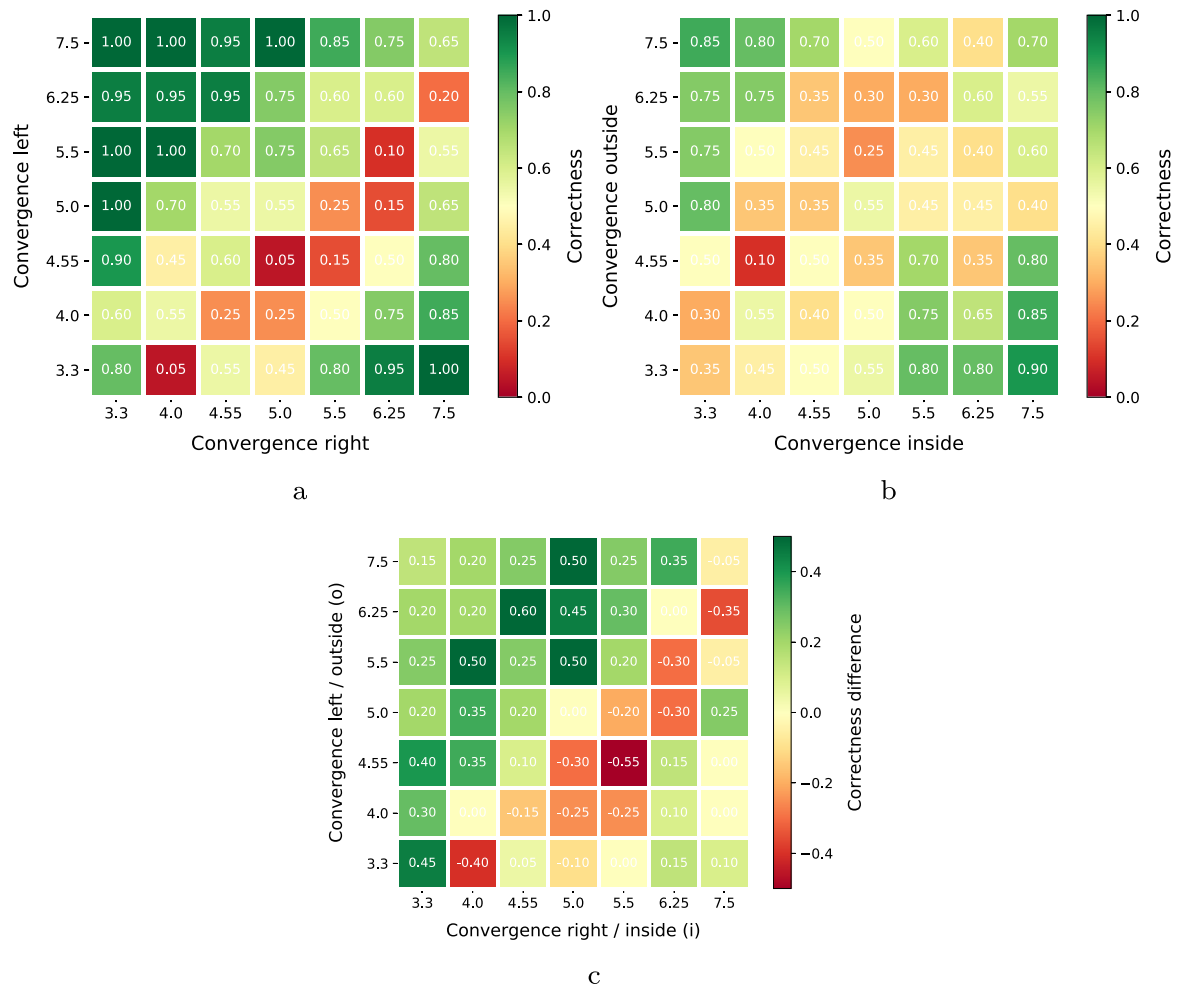


Fig. 19 Correctness for the different parameter combinations for the straight experiments **a** and the circular ones **b**. The combined plot shows the correctness difference between the straight and the circular experiments **c**, with positive values (green) indicating better results for the straight experiments, and negative values (red) better results for the circular ones. Larger parameters indicate stronger convergence on the respective side

visualization. From the data, it is not possible to establish a link from domain knowledge to the results, as the knowledge was not explicitly asked in the questionnaire. However, no outliers could be identified that would suggest any correlation of knowledge and quality of results. All of the participants have finished all of the 98 tasks, amounting to 1,960 responses in total.

Results

The results for all parameter combinations are plotted in Fig. 19, separately for the tasks involving the straightened feature line in Fig. 19a, and for the circular feature in Fig. 19b. While the calculated p -value of 0.5% clearly indicates statistical significance, the difference in correctness between both experiments in Fig. 19c highlights that the assessment of the convergence using the straightened feature line generally performs better.

However, in the region below the diagonal in Fig. 19a, the assessment of the participants differ strongly from the assumed ground truth. This is due to the fact that the streamline length depends on the distance of its seed to the periodic orbit. Therefore, determining the rate of convergence from the streamline length is influenced by this distance, and introduces this error. It should have been stated clearly in the instructions that one should deduct the convergence from the incident angle of the streamline onto the feature line. This “bias”, however, is present in both sets, depicting the circular and the straightened feature line. That it is

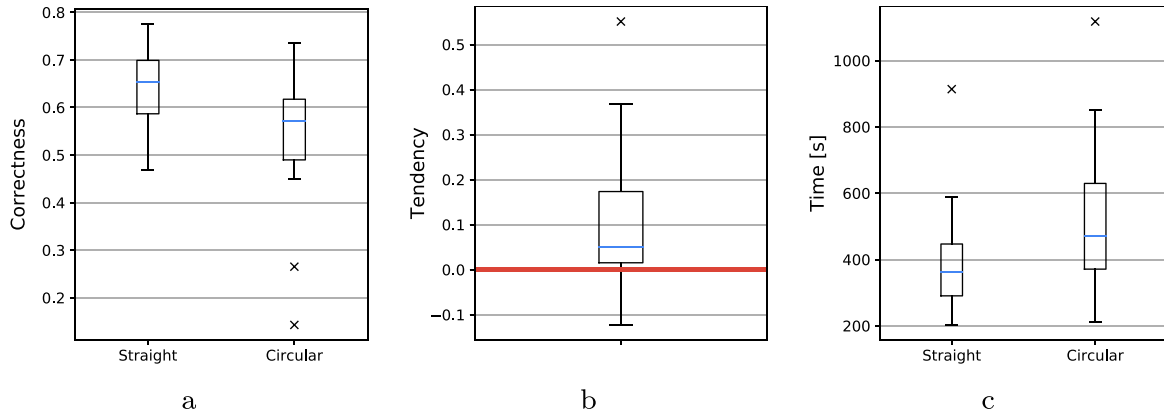


Fig. 20 Evaluation per participant. **a** Correctness shows that in general, the participants could provide more correct answers for the straightened feature line than for the circular. Particularly, two participants performed exceedingly bad assessing the convergence in the circular experiments. **b** The tendency indicates that most participants performed better on the straight images (above zero) than participants performing better on the circular ones (below zero). The value depicted is the difference in correctness, evaluated per participant. **c** Timings show that the participants took considerably more time in assessing the circular images. Note that the plotted time starts at 200 s

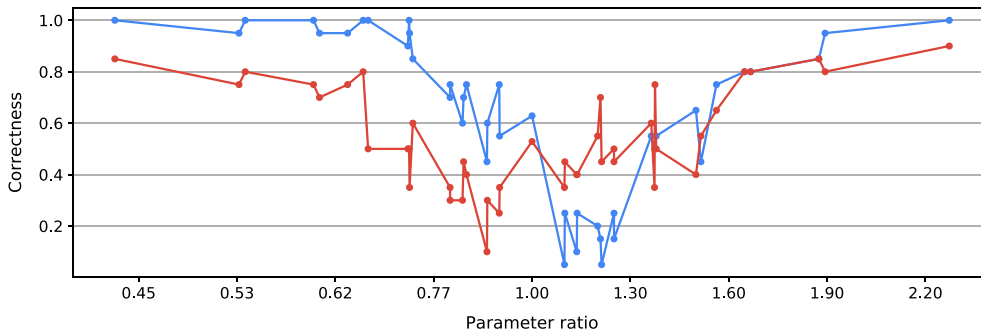


Fig. 21 Correctness of the participants' assessment for varying parameter ratios for the straightened feature line (blue) and the circular one (red). A parameter difference ratio far from 1 corresponds to a large difference in convergence rate on either side, while a ratio of 1 means that the convergence on both sides is equally strong. Small values correspond to larger convergence on the left side or on the outside, large values to larger convergence on the right side or on the inside, respectively. Please note the scaling of the x-axis, where values below 1 are mapped to $2 - \frac{1}{x}$ for symmetry

stronger recognized for the straightened feature, indeed shows its usefulness, as the results of the circular feature show a broader range of low values of correctness.

Evaluated per participant, Fig. 20a plots the results for the straightened and the circular feature line. It is apparent that most participants performed better on the straight dataset, with two outlying participants performing exceedingly bad on the circular set. These two participants were among the average for the experiments on the straightened feature line. Unfortunately, it is not possible to further investigate the reason for these particular outliers, as our collected, anonymized data does not allow for the identification of individuals, and the study did not provide the possibility for feedback. Plotting the difference in correctness per participant, Fig. 20b confirms the overall impression, and certainly proves the hypothesis behind RQ 1.

For RQ 2, we evaluated and plotted the timings per participant in Fig. 20c. This plot shows that it took considerable more time to assess the convergence rate to the circular feature line. With a p -value of 0.05% indicating statistical significance, this proves the hypothesis of RQ 2. Interestingly, we could not find a correlation between the time to finish the tasks and the correctness of the participants' performance. While the participant corresponding to the time outlier on the circular dataset indeed was one of the three participants to perform better than on the straight dataset, the second longest in finishing the tasks performed much better in the experiments of the straightened feature line.

In order to prove the hypothesis behind RQ 3, the correctness around the parameter ratio of 1 would need to be larger for the straightened feature line. Observing the correctness in Fig. 21, this is true for parameter

ratios below 1, but not for ratios above. Therefore, our study does not allow the conclusion that smaller convergence difference suffices for the straightened feature line. While we cannot prove the hypothesis, the data does not disprove it, either. To follow up, a new study trying to prove or disprove this research question would need to account for the issue of the streamline length used to determine the convergence rate, as described above.

Acknowledgements We want to thank Lutz Hofmann und Egzon Miftari for their support on new deformation schemes. This work was partially funded by Deutsche Forschungsgemeinschaft (DFG) as part of the Clusters of Excellence EXC 2075 “SimTech” (390740016) and EXC 2181 “STRUCTURES” (390900948), International Research Training Group GRK 2160 “DROPIT” (270852890), Collaborative Research Centre SFB 1313 (327154368), and Transregional Collaborative Research Centre SFB-TRR 191 (281071066).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

- Angelelli P, Hauser H (2011) Straightening tubular flow for side-by-side visualization. *IEEE Trans Vis Comput Graph* 17(12):2063–2070. <https://doi.org/10.1109/TVCG.2011.235>
- Ayachit U (2015) The ParaView guide: a parallel visualization application. Kitware, Inc
- Ben-Chen M, Weber O, Gotsman C (2009) Variational harmonic maps for space deformation. *ACM Trans Gr* 28(3):34. <https://doi.org/10.1145/1531326.1531340>
- Cabral B, Leedom LC (1993) Imaging vector fields using line integral convolution. In: *Proceedings of SIGGRAPH 93*, pp 263–270. <https://doi.org/10.1145/166117.166151>
- Correa CD, Silver D, Chen M (2007) Illustrative deformation for data exploration. *IEEE Trans Vis Comput Graph* 13(6):1320–1327. <https://doi.org/10.1109/TVCG.2007.70565>
- Crawfis R (2003) Tornado data set generator. <http://web.cse.ohio-state.edu/~crawfis.3/Data/Tornado>
- Daae Lampe O, Correa CD, Ma KL, Hauser H (2009) Curve-centric volume reformation for comparative visualization. *IEEE Trans Vis Comput Graph* 15(6):1235–1242. <https://doi.org/10.1109/TVCG.2009.136>
- Desbrun M, Meyer M, Schröder P, Barr AH (1999) Implicit fairing of irregular meshes using diffusion and curvature flow. In: *Proceedings of SIGGRAPH 99*, pp 317–324. <https://doi.org/10.1145/311535.311576>
- Eulzer P, Engelhardt S, Lichtenberg N, De Simone R, Lawonn K (2020) Temporal views of flattened mitral valve geometries. *IEEE Trans Vis Comput Graph* 26(1):971–980. <https://doi.org/10.1109/TVCG.2019.2934337>
- Forstmann S, Ohya J, Krohn-Grimberghe A, McDougall R (2007) Deformation styles for spline-based skeletal animation. In: *Eurographics/SIGGRAPH symposium on computer animation, SCA 2007*, pp 141–150. <https://doi.org/10.2312/SCA/SCA07/141-150>
- Günther T, Rössl C, Theisel H (2014) Hierarchical opacity optimization for sets of 3D line fields. *Comput Graph Forum* 33(2):507–516. <https://doi.org/10.1111/cgf.12336>
- Günther T, Theisel H (2020) Hyper-objective vortices. *IEEE Trans Vis Comput Graph* 26(3):1532–1547. <https://doi.org/10.1109/TVCG.2018.2868760>
- Günther T, Theisel H, Gross MH (2017) Decoupled opacity optimization for points, lines and surfaces. *Comput Graph Forum* 36(2):153–162. <https://doi.org/10.1111/cgf.13115>
- Hadwiger M, Mlejnek M, Theußl T, Rautek P (2019) Time-dependent flow seen through approximate observer Killing fields. *IEEE Trans Vis Comput Graph* 25(1):1257–1266. <https://doi.org/10.1109/TVCG.2018.2864839>
- Haker S, Angenent SB, Tannenbaum AR, Kikinis R (2000) Non-distorting flattening for virtual colonoscopy. In: *Medical image computing and computer-assisted intervention, MICCAI 2000*, pp 358–366. https://doi.org/10.1007/978-3-540-40899-4_36
- Huang J, Gao Z, Wang C (2014) A new grid deformation technology with high quality and robustness based on quaternion. *Chin J Aeronaut* 27(5):1078–1085. <https://doi.org/10.1016/j.cja.2014.09.003>
- Jacobson A, Baran I, Kavan L, Popovic J, Sorkine O (2012) Fast automatic skinning transformations. *ACM Trans Graph* 31(4):77:1–77:10. <https://doi.org/10.1145/2185520.2185573>
- Jacobson A, Baran I, Popovic J, Sorkine O (2011) Bounded biharmonic weights for real-time deformation. *ACM Trans Graph* 30(4):78. <https://doi.org/10.1145/2010324.1964973>
- Kanitsar A, Wegenkittl R, Fleischmann D, Gröller ME (2003) Advanced curved planar reformation: flattening of vascular structures. In: *Proceedings of IEEE visualization 2003*, pp 43–50. <https://doi.org/10.1109/VISUAL.2003.1250353>
- Kenwright DN, Henze C, Levit C (1999) Feature extraction of separation and attachment lines. *IEEE Trans Vis Comput Graph* 5(2):135–144. <https://doi.org/10.1109/2945.773805>

- Kreiser J, Meuschke M, Mistelbauer G, Preim B, Ropinski T (2018) A survey of flattening-based medical visualization techniques. *Comput Graph Forum* 37(3):597–624. <https://doi.org/10.1111/cgf.13445>
- Li C, Tong X, Shen HW (2017) Virtual retractor: an interactive data exploration system using physically based deformation. In: *IEEE Pacific Visualization Symposium, PacificVis 2017*, pp 51–60. <https://doi.org/10.1109/PACIFICVIS.2017.8031579>
- Li L, Hsieh HH, Shen HW (2008) Illustrative streamline placement and visualization. In: *IEEE Pacific Visualization Symposium, PacificVis 2008*, pp 79–86. <https://doi.org/10.1109/PACIFICVIS.2008.4475462>
- Lipman Y, Levin D, Cohen-Or D (2008) Green coordinates. *ACM Trans Graph* 27(3):78. <https://doi.org/10.1145/1360612.1360677>
- Lucius A, Brenner G (2010) Unsteady CFD simulations of a pump in part load conditions using scale-adaptive simulation. *Int J Heat Fluid Flow* 31(6):1113–1118. <https://doi.org/10.1016/j.ijheatfluidflow.2010.06.005>
- Machado GM, Sadlo F, Ertl T (2013) Local extraction of bifurcation lines. In: *Vision, Modeling and Visualization, VMV 2013*, pp 17–24. <https://doi.org/10.2312/PE.VMV.VMV13.017-024>
- Marchesin S, Chen CK, Ho C, Ma KL (2010) View-dependent streamlines for 3D vector fields. *IEEE Trans Vis Comput Graph* 16(6):1578–1586. <https://doi.org/10.1109/TVCG.2010.212>
- McLoughlin T, Jones MW, Laramée RS, Malki R, Masters I, Hansen CD (2013) Similarity measures for enhancing interactive streamline seeding. *IEEE Trans Vis Comput Graph* 19(8):1342–1353. <https://doi.org/10.1109/TVCG.2012.150>
- Nickolls J, Buck I, Garland M, Skadron K (2008) Scalable parallel programming with CUDA. *Queue* 6(2):40–53. <https://doi.org/10.1145/1365490.1365500>
- Peikert R, Roth M (1999) The “parallel vectors” operator—a vector field visualization primitive. In: *Proceedings of IEEE Visualization '99*, pp 263–270. <https://doi.org/10.1109/VISUAL.1999.809896>
- Rautek P, Mlejnek M, Beyer J, Troidl J, Pfister H, Theußl T, Hadwiger M (2021) Objective observer-relative flow visualization in curved spaces for unsteady 2D geophysical flows. *IEEE Trans Vis Comput Graph* 27(2):283–293. <https://doi.org/10.1109/TVCG.2020.3030454>
- Roth M (2000) Automatic extraction of vortex core lines and other line-type features for scientific visualization. PhD Thesis, ETH Zürich
- Roth M, Peikert R (1996) Flow visualization for turbomachinery design. In: *Proceedings of IEEE Visualization '96*, pp 381–384. <https://doi.org/10.1109/VISUAL.1996.568137>
- Sato S, Dobashi Y, Iwasaki K, Yamamoto T, Nishita T (2014) Deformation of 2D flow fields using stream functions. In: *SIGGRAPH Asia 2014 technical briefs*, pp 4:1–4:4. <https://doi.org/10.1145/2669024.2669039>
- Sato S, Dobashi Y, Yue Y, Iwasaki K, Nishita T (2015) Incompressibility-preserving deformation for fluid flows using vector potentials. *Vis Comput* 31(6):959–965. <https://doi.org/10.1007/s00371-015-1122-y>
- Schaefer S, McPhail T, Warren JD (2006) Image deformation using moving least squares. *ACM Trans Graph* 25(3):533–540. <https://doi.org/10.1145/1141911.1141920>
- Spencer B, Laramée RS, Chen G, Zhang E (2009) Evenly spaced streamlines for surfaces: an image-based approach. *Comput Graph Forum* 28(6):1618–1631. <https://doi.org/10.1111/j.1467-8659.2009.01352.x>
- Sujudi D, Haimes R (1995) Identification of swirling flow in 3-D vector fields. In: *12th Computational Fluid Dynamics Conference*. <https://doi.org/10.2514/6.1995-1715>
- Tao J, Wang C, Shene CK, Kim SH (2014) A deformation framework for focus+context flow visualization. *IEEE Trans Vis Comput Graph* 20(1):42–55. <https://doi.org/10.1109/TVCG.2013.100>
- Taubin G (1995) Curve and surface smoothing without shrinkage. In: *Proceedings of the 5th International Conference on Computer Vision, ICCV 95*, pp 852–857. <https://doi.org/10.1109/ICCV.1995.466848>
- Theisel H, Sahner J, Weinkauff T, Hege HC, Seidel HP (2005) Extraction of parallel vector surfaces in 3D time-dependent fields and application to vortex core line tracking. In: *Proceedings of IEEE Visualization 2005*, pp 631–638. <https://doi.org/10.1109/VISUAL.2005.1532851>
- Tong X, Edwards J, Chen CM, Shen HW, Johnson CR, Wong PC (2016) View-dependent streamline deformation and exploration. *IEEE Trans Vis Comput Graph* 22(7):1788–1801. <https://doi.org/10.1109/TVCG.2015.2502583>
- Yoshizawa S, Belyaev AG, Seidel HP (2003) Free-form skeleton-driven mesh deformations. In: *Proceedings of the 8th ACM Symposium on Solid Modeling and Applications*, pp 247–253. <https://doi.org/10.1145/781606.781643>
- Yu H, Wang C, Shene CK, Chen JH (2012) Hierarchical streamline bundles. *IEEE Trans Vis Comput Graph* 18(8):1353–1367. <https://doi.org/10.1109/TVCG.2011.155>