

Visualization of 4D Vector Field Topology

Lutz Hofmann, Bastian Rieck, and Filip Sadlo

Heidelberg University, Germany

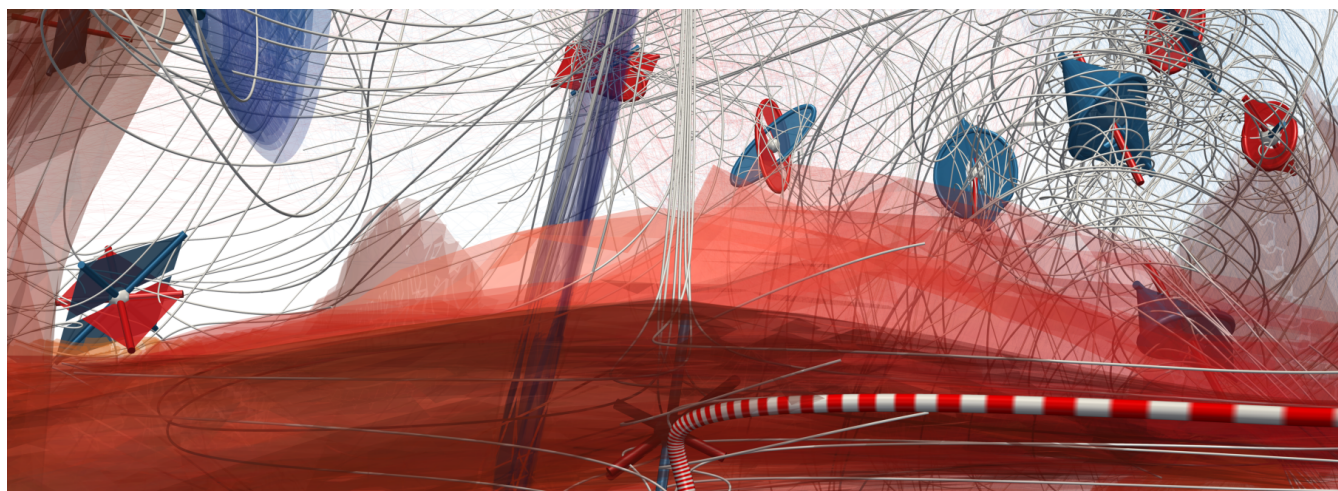


Figure 1: 3D projection of topological structures of a 4D vector field, with critical points depicted by 4D glyphs, their stable and unstable invariant manifolds (blue and red, respectively) by volumes in 4D, together with 4D streamlines for context. The 4D camera moves in 4D space along a selected streamline (striped red–white), while maintaining a view which minimizes the amount of clutter in the 3D projection.

Abstract

In this paper, we present an approach to the topological analysis of four-dimensional vector fields. In analogy to traditional 2D and 3D vector field topology, we provide a classification and visual representation of critical points, together with a technique for extracting their invariant manifolds. For effective exploration of the resulting four-dimensional structures, we present a 4D camera that provides concise representation by exploiting projection degeneracies, and a 4D clipping approach that avoids self-intersection in the 3D projection. We exemplify the properties and the utility of our approach using specific synthetic cases.

CCS Concepts

•Human-centered computing → Visualization techniques; •Applied computing → Mathematics and statistics;

1. Introduction

Many physical phenomena in our everyday life, that involve force or motion, are accessible by the concept of a two-dimensional or three-dimensional time-independent vector field. There are, however, cases, where a fourth dimension is required for appropriate modeling, leading to four-dimensional vector fields. Since vector fields correspond to differential equations [Asi93] and therefore to dynamical systems, any continuous deterministic dynamical system with four-dimensional phase space, and any four-dimensional differential equation, represents a four-dimensional vector field. Besides mathematics, there is a wide range of problems that lead to such four-dimensional dynamical systems, including the change

of concentration of four substances due to chemical reactions, or the evolution of four competing species. It is, however, nowadays often the case that such problems are investigated by only a few simulations instead of dense ensembles, i.e., simulations are obtained only for a few initial conditions or parameter choices. Such sets of simulations result in four-dimensional phase spaces whose domain is, however, less than four-dimensional, and thus do not necessitate four-dimensional analysis. Nevertheless, a prominent source for four-dimensional vector fields is already today the phase space in physics, consisting of position and momentum. It is $2n$ -dimensional for a problem in n -space, and is used to represent the motion of inertial objects due to, e.g., forces.

The concept of vector field topology, i.e., extracting singular structures, such as critical points and periodic orbits, and revealing the invariant manifolds of trajectories that converge to the singular structures in forward or reverse time, has proven very successful in a wide range of research questions and applications over many decades. In particular if such manifolds are of codimension one, i.e., $(n - 1)$ -manifolds in n -dimensional vector fields, they separate the domain into regions of qualitatively different behavior, and thus ease analysis and understanding.

Applying the concept of vector field topology to 4D vector fields, however, leads to a set of challenges, part of which we address in this work. First, the critical points exhibit different types in 4D fields, requiring a respective classification, and more important, appropriate visual representation. Whereas the former is a straightforward mathematical task, we address the latter with a set of glyphs, which, however, are four-dimensional. We handle the visualization of the four-dimensional structures in our approach with projection-based approaches, with a focus on avoiding projection-induced ambiguities and supporting the difficult exploration and imagination in four-dimensional space. Second, the overall navigation in 4D space, populated with the manifolds and glyphs, is of course difficult due to various paradoxes. We therefore complement our approach with a technique that helps in navigating the resulting structures, and 4D space in general. Last but not least, we present a set of vector fields that help introduce our approach, and introduce the reader to 4D space and its navigation.

Our contributions include:

- classification, glyphs, manifold extraction, and projection-based visual representation of critical points in 4D vector fields,
- and manifold-based exploration of 4D vector field topology.

2. Related Work

Vector field topology has first been considered in visualization by Helman and Hesselink [HH89, HH91] in two and three dimensions. An overview of the state of the art in vector field topology is given by Laramée et al. [LHZP07], who focus in particular on the applicability with respect to flow visualization. A recent survey by Heine et al. [HLH*16] takes a broader view and illustrates various concepts that go beyond flow visualization.

In several applications, four-dimensional problems arise from lower-dimensional ones by considering additional aspects, such as time-dependency or inertia. Many of these cases, however, do not lead to true four-dimensional vector fields. Two-parameter-dependent 2D vector fields [WTHS06] only consider vectors in 2-space, while the tracking of 3D critical points [GTS04] over time only considers 3D vector fields at different instances in time. Flow-induced inertial dynamics of 2D systems [GG17], on the other hand, has an underlying 4D phase space, but the underlying flow allows the reduction of the analysis to the two-dimensional spatial domain. Our work focuses on general four-dimensional vector fields. We discuss further related work in this context in Section 8.

Visualization of 4D geometry has been first explored in an early approach [Nol67], where an automatic plotter was used to render frames of a two-dimensional movie showing projections of rotating four-dimensional geometry. This technique was limited to

wireframe depictions, and has later been implemented on a Silicon Graphics workstation [Hol91]. Subsequently, lighting models in four dimensions were proposed [HH92, HC93]. Stereographic projections have been employed for the visualization of contours of complex-valued bivariate functions [WB96] as an alternative. While these techniques are based on the idea of visualizing two-dimensional surfaces in 4-space, i.e., \mathbb{R}^4 , a three-dimensional frame buffer has been utilized for volume rendering [CFHH09]. In the same context, the problem of self-intersections has been addressed with higher-dimensional halos [WYF*13].

Flow in 3D fields can be analyzed using flow volumes, either by calculating explicit geometrical representations of separating volumes by streamline integration [MBC93], or by constructing an implicit flow as a scalar field [XZC04]. Implicit approaches have been shown to be advantageous for characterizing structures in steady vector fields, such as stream surfaces [vW93]. In essence, the distance field that we construct in this paper (measuring the Euclidean distance to invariant manifolds) also falls under this category.

3. Critical Points

In an n -dimensional steady vector field $\mathbf{u}(\mathbf{x})$ with $\mathbf{x} := (x_1, \dots, x_n)^\top$ and $\mathbf{u} := (u_1, \dots, u_n)^\top$, and $\mathbf{x}, \mathbf{u} \in \mathbb{R}^n$, a critical point \mathbf{x}_c is an isolated zero, i.e., $\mathbf{u}(\mathbf{x}_c) = \mathbf{0}$ and $\det \nabla \mathbf{u}(\mathbf{x}_c) \neq 0$. Critical points are typically classified in terms of the linearization of the vector field in their neighborhood, i.e., by means of the eigenvalues of $\nabla \mathbf{u}(\mathbf{x}_c)$.

3.1. Classification in 2D

For $n = 2$, $\nabla \mathbf{u}(\mathbf{x}_c)$ is a real 2×2 matrix, exhibiting either two real or a pair of complex conjugate eigenvalues. In both cases, the real part indicates the inflow/outflow behavior of the vector field along the respective eigenvector, and a nonzero imaginary part indicates rotational behavior. Table 1 summarizes the five types of critical points in 2D vector fields (which can be reduced to the three types repelling node, repelling focus, and saddle, if flow reversal is allowed). Notice that complex eigenvectors represent eigenplanes (Table 1(d) and (e)), which span \mathbb{R}^2 ; moreover, only one of these configurations separates different regions, i.e., that of type saddle (Table 1(c)). We indicate non-separating eigenvectors in these illustrations by dashed representation (leading to circular bands for eigenplanes), whereas separating eigenvectors (and the manifolds they might span) are depicted with solid representation. Throughout this paper, red color indicates repelling (unstable) behavior, i.e., away from the critical point, while blue color indicates attracting (stable) behavior toward the critical point.

3.2. Classification in 3D

For 3D vector fields, $\nabla \mathbf{u}(\mathbf{x}_c)$ is a real 3×3 matrix, exhibiting either three real, or one real and a pair of complex conjugate eigenvalues. Again, the real part indicates the inflow/outflow behavior of the vector field along the respective eigenvector, and a nonzero imaginary part indicates rotational behavior. Table 2 summarizes the eight resulting types (which can be reduced to the four types source, saddle, spiral source, and spiral saddle, if flow reversal is allowed). Again, we have cases with eigenplanes (Table 2(d), (e),

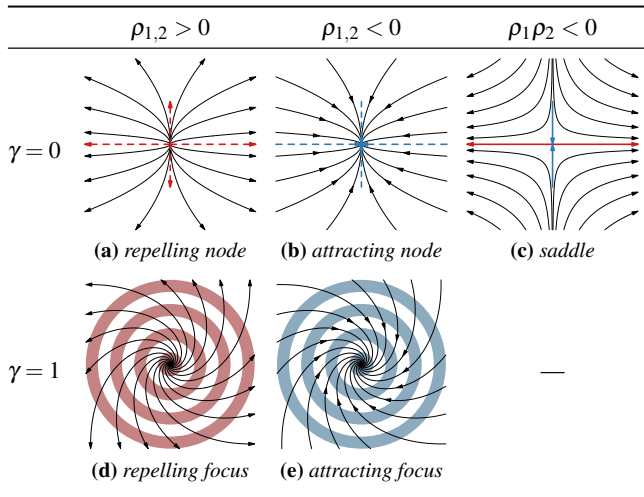


Table 1: Types of critical points in 2D vector fields, by means of $\nabla \mathbf{u}$, with real parts $\rho_{1,2}$, complex pairs γ , and ρ -scaled eigenvectors (repelling in red, attracting in blue, non-separating dashed).

and (f) indicating rotation. In 3D, we have two separating configurations, the saddle and spiral saddle (Table 2(c) and (f)). These saddle-type cases can be further classified with respect to the number of incoming/outgoing directions. Table 2(c) depicts the case with one negative eigenvalue and two positive ones ($\rho_1 \rho_2 < 0$), also denoted 1:2 saddle (1-manifold in, 2-manifold out). If we revert the direction of the vector field, we obtain the 2:1 saddle configuration ($\rho_2 \rho_3 < 0$), exhibiting in our scheme a blue 2-manifold and a red 1-manifold. Accordingly, there are 1:2 spiral saddles, as well as 2:1 spiral saddles, which we propose to abbreviate as 1:s2 and s2:1 saddles, the “s” indicating rotation of the respective manifold (this will become useful for the 4D cases).

Interestingly, we observe the cases from 2D fields in the cases in 3D fields. For example, the vector field on the separating red surface in Table 2(c) (see gray lines) exhibits a 2D repelling node (Table 1(a)), and the red surface in Table 2(f) exhibits a 2D repelling focus (Table 1(d)). Many other sections through the 3D cases exhibit respective 2D cases, in particular in sections spanned by the 3D eigenvectors. For example, the section spanned by the attracting (blue) eigenvector and one of the repelling (red) eigenvectors in Table 2(c) exhibits a 2D saddle (Table 1(c)).

3.3. Classification in 4D

In 4D vector fields, the Jacobian $\nabla \mathbf{u}(\mathbf{x}_c)$ is a real 4×4 matrix, and thus exhibits either four real eigenvalues, two real eigenvalues and a complex conjugate pair, or two complex conjugate pairs. For now, we switch to the complex plane \mathbb{C} for an overview of the cases that this leads to. Due to the difficulty with 4D representation, we do not illustrate the cases in space, but exemplify them subsequently with our glyph-based approach. Table 3 summarizes the resulting 14 cases (which can be reduced to the eight types source, 1:3 saddle, 2:2 saddle, 1-spiral source, 1:spiral-3 saddle, 2:spiral-2 saddle, 2-spiral source, and 2-spiral saddle, if accounting for flow reversal). We keep the naming scheme where the numbers on each side of the

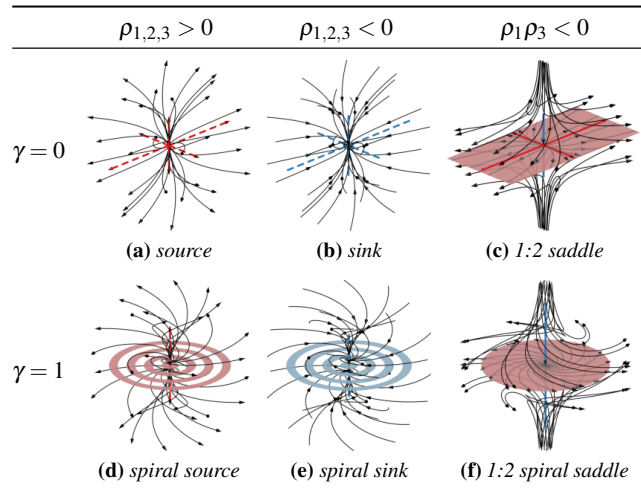


Table 2: Types of critical points in 3D vector fields, based on $\nabla \mathbf{u}$, with real parts $\rho_{1,2,3}$ (ascending sorting), complex pairs γ , and ρ -scaled eigenvectors (repelling red, attracting blue, non-separating dashed). Notice that for the saddles, only $\rho_1 \rho_2 < 0$ is illustrated.

colon denote the dimension of the stable (left) and unstable (right) manifold. Again, the types can be classified in terms of the number of complex conjugate pairs γ .

If there is no such pair ($\gamma = 0$), the cases are simple to classify. If all real parts have the same sign, there are the types source (Table 3(a)) and sink (b). If the real parts have both signs, we obtain saddles. If there is a single real part with opposite sign, we obtain the 1:3 (c) and 3:1 (e) cases, e.g., if one real part is negative (and thus, since $\det \nabla \mathbf{u}(\mathbf{x}_c) \neq 0$, three are positive), we obtain the 1:3 saddle. If there are two of each sign, we identify the 2:2 saddle (d). The 1:3 and 3:1 saddles both exhibit a 1-manifold and a 3-manifold, and since the 3-manifold is of codimension 1, it separates four-dimensional space. In this respect, the 1:3 and 3:1 saddles correspond to a 3D saddle (Table 2(c)). Note also that in Table 2(c) the blue arrow is not dashed, i.e., we classify the 1-manifold of 3D saddles as being separating. We do this because, although a 1-manifold does not separate 3-space, it represents an asymptote for neighboring streamlines, or in other words, it separates streamlines with respect to rotational symmetry. (Another motivation to do so is that the abovementioned section through the case in Table 2(c), with a plane that contains the 1-manifold, results in a 2D saddle (Table 1(c)), and thus a clearly separating structure.) Consequently, we classify the 1-manifold of 1:3 and 3:1 saddles in 4D as being separating, too. This argumentation makes us classify the 2:2 saddle (Table 3(d)) as being separating as well, although 2-manifolds do not separate 4-space. As will be discussed below, we extract invariant manifolds (separatrices) from all critical point manifolds being separating in this sense.

If $\gamma = 1$ and all real parts have the same sign, we have the 1-spiral source (Table 3(f)) and the 1-spiral sink (g). We denote these cases *1-spiral* to indicate that they exhibit one plane of rotation, in contrast to *2-spiral* cases, which exhibit two planes of rotation. (Rotations in 4-space have six degrees of freedom, and a plane of ro-

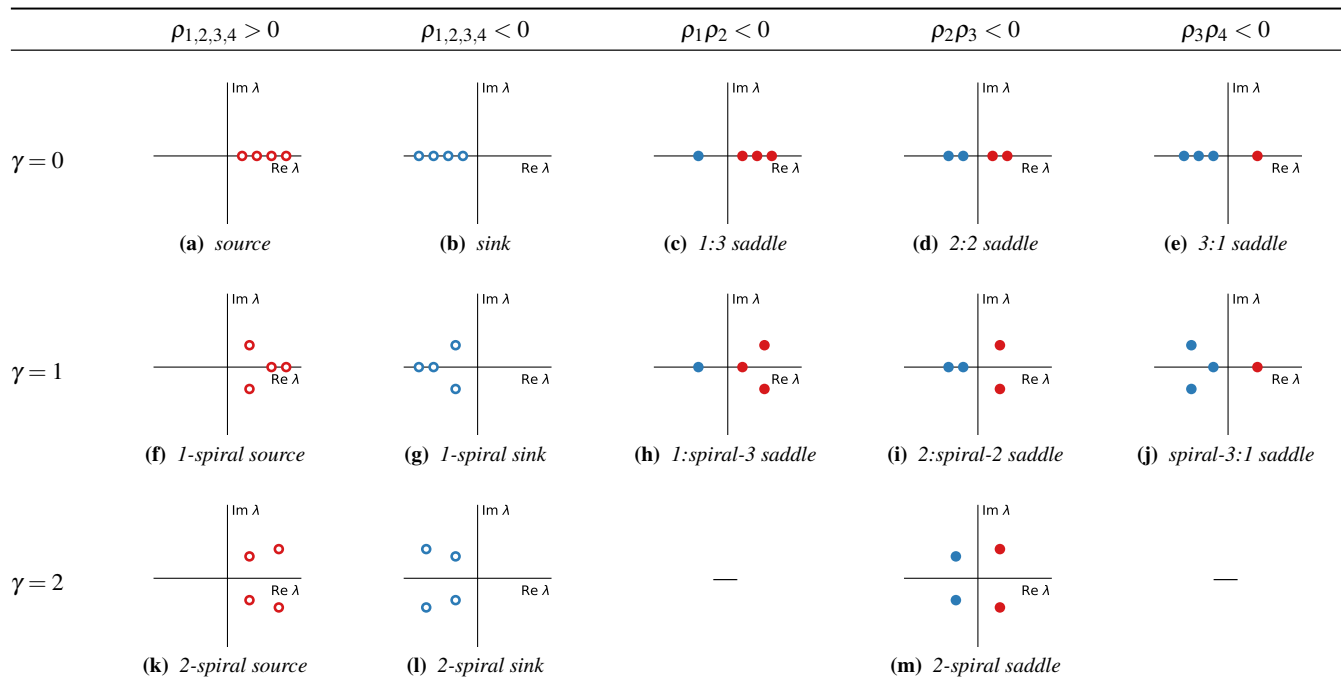


Table 3: Classification of critical point types in 4D vector fields (depicted in \mathbb{C}), by means of $\nabla\mathbf{u}$, with real parts $\rho_{1,2,3,4}$ (ascending sorting) and number of complex pairs γ . No rotation ($\gamma = 0$), rotation in one plane ($\gamma = 1$), and rotation in two planes ($\gamma = 2$). Please note that we did not depict the spiral-2:2 saddle (counterpart to (i)), it is obtained by reverting vector field direction in (i) (mirroring about the imaginary axis). We depict repelling eigenvalues with red, attracting with blue, separating property with a dot, and non-separating with a circle.

tation rotates an object in 4D about a plane, not about a line (axis), as would be the case in 3D. Thus in 4D, rotation can take place in two linearly independent planes simultaneously.) In the remaining cases with $\gamma = 1$, the real parts have opposite sign, and thus lead to saddle behavior exhibiting rotation in one plane. If there is a single real part with opposite sign, we obtain the 1:spiral-3 (h) and the spiral-3:1 (j) saddles, which we abbreviate 1:s3 and s3:1 saddles, respectively, to make the distinction to 1-spiral and 2-spiral cases more clear. This notation is an extension from the 3D one, again putting the incoming manifold before the colon, and the outgoing manifold after. Thus, e.g., 1:s3 saddles (h) have inflow along a 1-manifold and outflow along a rotating (spiraling) 3-manifold. The only remaining configuration with $\gamma = 1$ is two real parts of each sign, i.e., the 2:s2 and the s2:2 saddle cases. For improved presentation in Table 3, we show only the 2:s2 saddle (i)—the s2:2 saddle is obtained by reversal of flow direction.

Finally, if $\gamma = 2$, there is simultaneous rotation in two independent planes, which we denote by the term “2-spiral”. We have a 2-spiral source (k) if all real parts are positive, a 2-spiral sink (l) if all are negative, and a 2-spiral saddle (m) in case of opposite sign.

3.4. Extraction and Representation

Several strategies for extracting (detecting) critical points have been proposed so far. One approach is inspired by the marching cubes algorithm [LC87] for isosurface extraction. Since a critical point represents a zero in all n components of the vector field, it

can be interpreted as the intersection of the n zero-level isosurfaces of its components. Technically, an early rejection test skips in these approaches all grid cells that do not exhibit a sign transition in all n components of the vector field. In the remaining candidate cells, critical points are typically searched for by inversion of the interpolation function, or by subdivision. By their nature, these approaches work in any dimension, but the inversion approach can become challenging due to numerics and degenerate cases. In our implementation, we follow the subdivision approach, i.e., we recursively apply the candidate test and subdivide the remaining candidates until a sufficiently high resolution is achieved (or reject the refined cell if it does not span both signs in all n components). In a final clustering step, we remove duplicate candidates, which occur due to limited numerical precision mainly near cell boundaries. For subsequent classification of the determined critical points, we compute the Jacobian from the interpolation function, for consistency with the subdivision-based extraction scheme.

Having described the extraction of 4D critical points, and having classified their properties in terms of inflow, outflow, and rotation, we now can address their representation by means of glyphs. Our approach is inspired by the glyphs for 3D vector field topology by Theisel et al. [TWS03]. Similar to the 3D approach, we build our glyphs in the respective space, i.e., our glyphs are four-dimensional. In our implementation, however, we maintain only the essential information in 4D, in particular the eigenvectors and eigenvalues of the Jacobian. The 3D geometrical representation of the glyph (tubes etc.) is built only in 3D, after projection from 4D

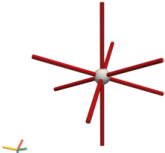
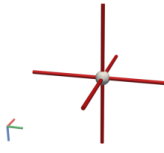

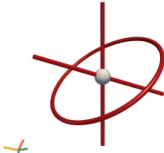
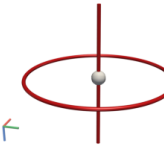
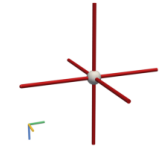
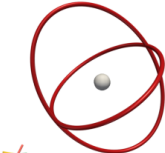
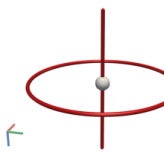
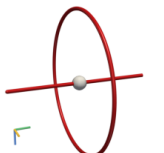
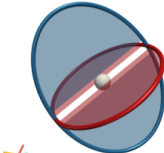
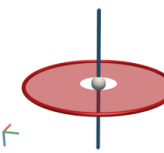
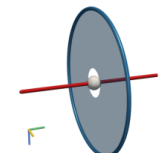
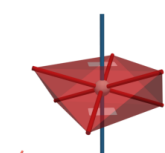
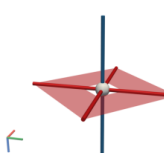
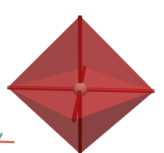
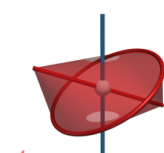
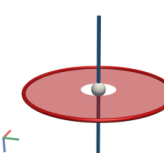
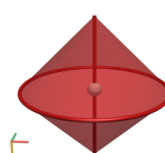
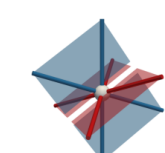
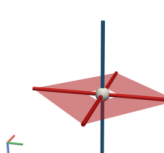
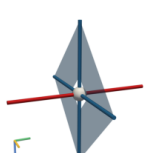

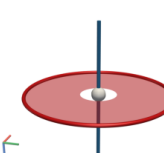
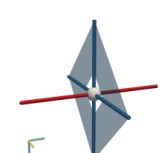
4D view (regular)	4D view (edge case)	4D view (edge case)	4D view (regular)	4D view (edge case)	4D view (edge case)
 (a) <i>source</i>	 (b) <i>source</i>	 (c) <i>source</i>	 (d) <i>1-spiral source</i>	 (e) <i>1-spiral source</i>	 (f) <i>1-spiral source</i>
 (g) <i>2-spiral source</i>	 (h) <i>2-spiral source</i>	 (i) <i>2-spiral source</i>	 (j) <i>2-spiral saddle</i>	 (k) <i>2-spiral saddle</i>	 (l) <i>2-spiral saddle</i>
 (m) <i>1:3 saddle</i>	 (n) <i>1:3 saddle</i>	 (o) <i>1:3 saddle</i>	 (p) <i>1:spiral-3 saddle</i>	 (q) <i>1:spiral-3 saddle</i>	 (r) <i>1:spiral-3 saddle</i>
 (s) <i>2:2 saddle</i>	 (t) <i>2:2 saddle</i>	 (u) <i>2:2 saddle</i>	 (v) <i>2:spiral-2 saddle</i>	 (w) <i>2:spiral-2 saddle</i>	 (x) <i>2:spiral-2 saddle</i>

Table 4: The glyphs for the 8 different basic types ((a), (d), (g), (j), (m), (p), (s), (v)) of critical points in 4D vector fields, with repelling (red) and attracting (blue) behavior (with flow reversal, this provides glyphs for the 14 cases from Table 3). Each type shown in three 4D views, leading to 3D “images”. The regular view (first and fourth column) shows the glyph in non-degenerate projection. The second 4D view (second and fifth column) shows one edge case where manifolds degenerate due to projection, and the third 4D view (third and sixth column) shows the other edge case. Notice that in practice, the degeneracies do not occur (similar to perfectly oblique views in 3D projection), they serve for explanatory purposes. Observe the three-dimensional projections of the 4D axes (bottom left, x-red, y-green, z-blue, and w-orange).

to 3D. Generally, we represent all structures (including the invariant manifolds) in 4D space and observe them with a 4D camera. Essentially, this camera has, similar to the earlier works of Noll and Hollasch [Nol67, Hol91], a 4D view vector and projects the 4D world to a 3D image plane. These 3D “images” can, additionally to the navigation of the 4D camera in 4D space, be navigated in 3D space. In fact, both cameras together represent the 4D camera. Further details on our projection approach are discussed in Section 5.

Glyphs for critical points typically consist of a set of geometric building blocks, one for each manifold type, which are then assembled according to the respective type of a critical point. Most approaches follow directly the linearized structure of the manifolds in vicinity of the critical point, i.e., the structure dictated by the eigenvalues and their eigenvectors. The skeleton of our glyph consists of the eigenvectors in 4D. We represent real eigenvectors with tubes of length proportional to the modulus of the respective eigenvalue, and map its sign to red if positive, and to blue if negative (e.g.,

Table 4(a)). Complex conjugate pairs are represented with circles located in the respective eigenplane, with radius proportional to the modulus of the respective real part, and again positive sign of real part mapped to red, and negative to blue (e.g., Table 4(d)).

When all real parts of the eigenvalues have same sign, the respective eigenvectors span a four-dimensional volume. Projecting this 4D volume to the 3D image plane would lead to a 3D volume, which, however, could be misleading, since there are other critical point types that exhibit a 3D manifold in 4D, which also projects in non-degenerate views to a 3D volume in 3-space. Therefore, we decided to not generate any additional geometry in these cases (Table 4(a), (d), and (g), and Table 3(a), (b), (f), (g), (k), (l)). Notice that this scheme is conceptually consistent with our illustrations for 2D and 3D critical points, where we used dashed representation for non-separating cases, and solid representation for separating cases. Our 4D cases with same real part sign represent non-separating cases, and as can be see in Table 4, they are the only ones that

consist only of lines. All remaining 4D glyphs represent separating cases (saddles), and exhibit 2-manifolds or 3-manifolds.

We already started to depict the eigenplanes spanned by complex conjugate eigenvector pairs, i.e., planes of rotation, with *circular discs* (Table 1(d) and (e), and Table 2(d) and (e)). We use this approach also in 4D, where these discs are still 2-manifold, but have a pose in 4D space (Table 4(j) and (v)). On the other hand, we represent non-rotating 2-manifolds, which are spanned by real eigenvectors, with quads in 4D, whose diagonals are the two eigenvector tubes (Table 4(s), and (v)). Non-rotating 3-manifolds are correspondingly represented by octahedra in 4D, whose “diagonals” are the three eigenvector tubes (Table 4(m)). The remaining case of rotating 3-manifolds, which are spanned by a real eigenvector and a complex conjugate pair of eigenvectors, can be considered a combination of a rotating 2-manifold and a non-rotating 1-manifold, leading to an intersection of two elliptic cones (Table 4(p)). Note that we represent both the octahedra and the cone intersections by their (triangulated) surface. To reduce ambiguities in projection due to degenerate projection from 4D to 3D (Table 4), and in particular to indicate that manifolds that intersect in their 3D projection do not intersect in 4D, we cut them out around the critical point in 4D, leading to respective gaps in the 3D projections. Also observe that the degenerate views provided in Table 4 provide additional information about the critical point type itself, but are not needed when analyzing critical points in data (they serve here for complete illustration of their properties). Although such degenerate views will not appear in real-world applications with manual navigation, we make use of them for manifold-based exploration (see below).

4. Invariant Manifolds

Similar to 2D and 3D vector field topology, invariant manifolds in 4D also consist of those streamlines in $\mathbf{u}(\mathbf{x})$ that converge to saddle-type critical points in forward (stable manifolds) and reverse (unstable manifolds) direction. In the 2D and 3D case, it is common practice to build a seeding structure for the streamlines, which is based on the eigenvectors of $\nabla\mathbf{u}$. For 1-manifolds, the seeds consist of two points, each obtained by a small user-defined offset along the respective eigenvector, away from the critical point to escape its zero velocity. For 2-manifolds, one commonly places the seeds on a circle coplanar to the respective two (real parts of the) eigenvectors. Both approaches carry directly over to 4D, and we use them to obtain 1-manifolds and 2-manifolds. For 3-manifolds, finally, we put seeds on a sphere spanned by the respective three (real parts of the) eigenvectors.

Having the seeds, the 1-manifolds are obtained by simple streamline integration, resulting in a 4D polyline for each manifold. For 2-manifolds, we need to compute a streamsurface. We represent the seeding circle by a closed polyline, and employ Hultquist's streamsurface algorithm [Hul92], which can be directly employed in 4D vector fields, providing a mesh of triangles in 4-space. Finally, for the 3-manifolds, we connect the seeds into a triangulated sphere, consisting of triangles in 4-space, and employ the adaptive 3D flow volume approach by Max et al. [MBC93], which again is independent of the dimension of the surrounding space. Max et al. advance the triangular mesh with the flow and subdivide the prisms formed by two triangles at different time steps into tetrahe-

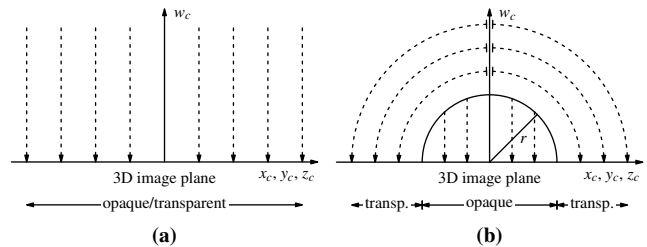


Figure 2: 4D clipping sphere for removal of projection-induced intersection in 3D space. (a) Projection along w_c from 4D space into 3D image plane with camera coordinates x_c , y_c , and z_c (Section 5). (b) Inside sphere (solid-line circle), same projection as in (a). Outside clipping sphere, projection along dashed curves. Both projections provide a single 3D image for interactive exploration.

dra. Whenever the area of a triangle reaches a threshold, they subdivide it and insert additional streamlines. Our resulting tetrahedral mesh is located in 4-space, but can be easily represented by traditional visualization toolkits, because the coordinates can be, if no 4D support is provided, represented as four scalar fields, in which case the toolkits' three coordinates can represent parametrization.

Note that, in analogy to invariant manifolds (also called separatrices) in 2D and 3D fields, they also cannot intersect in 4D, because a vector field defines at each point in space only a single direction. We store our discretized manifolds with their parametrization, i.e., with respect to streamlines and “timesurfaces”. Whenever we need, for example, the normal of a 3D manifold for camera orientation, we compute it using the 4D cross product (taking three vectors) from the three vectors spanning the tetrahedron (recall that in 4D space, 3D structures represent hypersurfaces).

5. 4D Camera

For a more intuitive notation that is consistent with traditional camera coordinates and 4D notations, we denote 4D space with $\mathbf{x} = (x_1, \dots, x_4)^T =: (x, y, z, w)^T$ and 4D vectors, e.g., the vector field, in that space with $\mathbf{u} = (u_1, \dots, u_4)^T =: (u_x, u_y, u_z, u_w)^T$. Since most display devices used for visualization are two-dimensional, we need to reduce the dimensionality from 4D to 2D. Because of its well-defined properties, its intuition, and its simplicity, we build here on projection. The traditional rendering pipeline transforms 3D world coordinates to 3D camera coordinates, and those then by projection to 2D image coordinates. The extension to our 4D setup adds one degree of freedom (DOF) for the position of the camera, and three DOFs for its orientation (we have four DOFs for position, and six DOFs for orientation in 4D). Managing these additional four DOFs is nontrivial, in particular because 4D space is difficult to interpret, and navigate. We therefore handle these degrees of freedom with two cameras: a 4D one, and a 3D one.

What we call the *4D camera* is a camera with a 4D position \mathbf{c} , a 4D view direction $\hat{\mathbf{c}}$ (a unit vector defining the optical axis of the camera in 4D space), and a “3D image plane”, where the scene is projected to. Hence, this camera produces three-dimensional scenes. Since humans are not used to projections in 4D space, we recommend using orthographic projection within the

4D camera, leading to the setup in Figure 2(a). The camera coordinates of this camera are $\mathbf{x}_c := (x_c, y_c, z_c, w_c)^\top =: (x, y, z, w)_c^\top$, with $(0, 0, 0, 1)_c^\top = \hat{\mathbf{c}}$. A point $(x_c, y_c, z_c, w_c)^\top$ in 4D camera coordinates is thus projected to 3D image coordinates $(x_c, y_c, z_c)^\top$. Notice that a 4D unit vector has three DOFs, so if we define the 4D camera by \mathbf{c} and $\hat{\mathbf{c}}$, three DOFs remain for orientation of the camera, which fits the three degrees of orientation of a 3D camera. Thus, we do not have to accomplish the difficult definition and handling of two 4D “up vectors”, as in other approaches for 4D rendering, but simply define \mathbf{c} and $\hat{\mathbf{c}}$, with subsequent traditional 3D rendering *and navigation* of the “three-dimensional image” provided by the 4D camera. In other words, we treat $(x, y, z)^\top = (x_c, y_c, z_c)^\top$ for visualization of the 3D image using a traditional 3D camera. To support orientation in the original 4D space, we provide context by four additional “thumbnail 3D images” defined by $(x, y, z)^\top$, $(x, y, w)^\top$, $(x, z, w)^\top$, $(y, z, w)^\top$, that represent simple 4D projections along the w -, z -, y -, and x -axis, i.e., that are independent of the 4D camera and show its position \mathbf{c} and orientation $\hat{\mathbf{c}}$ by a black sphere and an arrow, respectively. An example is shown in Figure 3(a). Here, the top row shows the thumbnail 3D images, which the user can explore independently. Additionally, we also show the axes of the camera coordinates in the main 3D image, i.e., we show the x_c -, y_c -, z_c -, and w_c -axis in red, green, blue, and yellow, respectively (see Figure 3, for example). Notice that conceptually, when the 3D camera is rotated relative to the 3D image, this defines the missing DOFs of rotation of the 4D camera but does not change $\hat{\mathbf{c}}$.

6. Exploration Techniques

We have seen in Table 4 (second, third, fifth, and sixth column) that certain 4D views of the 4D camera can cause degeneracies in the resulting 3D image, leading to lower-dimensional images of manifolds. Moreover, we have seen that (i) 1-manifolds can disappear (e.g., Table 4(e)), (ii) 2-manifolds can appear as 1-manifolds (e.g., Table 4(t)), and (iii) 3-manifolds can appear as 2-manifolds (e.g., Table 4(n)). These degeneracies happen when the view direction $\hat{\mathbf{c}}$ of the 4D camera is parallel (i), coplanar (ii), or covolumetric (iii) with the manifold at the respective point (as invariant manifolds tend to be curved in practice, this property only holds locally).

There are two main difficulties with the visualization of 3D manifolds in 4D topology: (I) visual clutter due to their volumetric appearance, and (II) (self-)intersection with manifolds due to projection; although manifolds cannot intersect in 4D space because they consist of streamlines there, their projection in 3D can intersect and tends to do so, leading to 3D images that are hard to interpret and explore. This is problematic because topological structure is often conveyed by configurations where stable and unstable manifolds meet, e.g., at critical points and saddle connectors [TWS03].

6.1. Manifold-Based Exploration

Fortunately, there is a simple approach to bring 3D manifolds into oblique 4D projection, which renders them planar at least locally in the 3D image, and thus helps solve (I). Since all manifolds consist of streamlines in 4D, their 4D normal(s) is perpendicular to the streamlines, so a projection along the tangent of a streamline will bring the respective region of the manifold in degenerate 4D

view, locally rendering 3D manifolds as surfaces in the 3D image, 2D manifolds as curves, and making 1D manifolds disappear. We enable the user to adjust $\hat{\mathbf{c}}$ to the vector \mathbf{u} of the vector field at any time, to pick manifolds using backprojection, which moves the camera center \mathbf{c} to the picked 4D point and sets $\hat{\mathbf{c}}$ to the respective precomputed streamline tangent, and also to navigate (and generate animations by moving) the camera along a selected streamline of the manifold. That way, the camera can follow a precomputed 4D streamline of a 3-manifold, keeping $\hat{\mathbf{c}}$ tangential to the streamline to show the 3-manifold locally as a plane in the 3D image. We provide a default configuration of the 3D camera by defining one “up vector” of the 4D camera to be the 4D normal vector of the manifold at \mathbf{c} . As the manifold is triangulated into tetrahedra, we first compute normals for each vertex of the tetrahedron that contains \mathbf{c} , and finally obtain the normal at \mathbf{c} by barycentric interpolation. The normal of a vertex is computed as the average of the normals of all tetrahedra adjacent to it. Given a tetrahedron $(\mathbf{t}_1, \dots, \mathbf{t}_4)$ in 4-space, its normal \mathbf{c} is obtained component-wise as $\mathbf{c}_i = \det(\mathbf{t}_2 - \mathbf{t}_1, \mathbf{t}_3 - \mathbf{t}_1, \mathbf{t}_4 - \mathbf{t}_1, \mathbf{e}_i)$, $i = 1, \dots, 4$, where \mathbf{e}_i denotes the i -th standard basis vector. This is a 4D generalization of the three-dimensional cross product. As a consequence of this choice of the 4D “up vector”, the default view of the explorative 3D camera is oblique to the manifold too, i.e., its 3D “up vector” is the projection of the 4D “up vector”, leading to a 2D projection that shows the manifold locally as a curve. This minimizes local occlusion and clutter, and provides a starting point for exploration by navigation of the 3D image.

We enable further exploration of the manifold, by providing the user with the option to gradually switch to neighboring streamlines of the manifold, thereby choosing the travel direction. The streamline that the camera is located on in this mode, is shown by a striped tube in the 3D image (Figure 1), providing additional context and a notion of motion, which otherwise might be missing (both translation and rotation in 4D do not necessarily appear as translation and rotation in the resulting 3D image, and rotations typically involve some deformation of the 4D image, as can be seen, e.g., in Table 4). Of course, we enable the observer to interactively deviate $\hat{\mathbf{c}}$ from \mathbf{u} to, e.g., obtain overview and context.

6.2. 4D Clipping Sphere and Manifold Distance

To address the problem of projection-induced intersection (II) of structures in the 3D image, that do not intersect in 4D space, we use two complementing approaches. On the one hand, we provide a “4D clipping sphere” that suppresses projection-induced intersection by modifying the projection from 4D to 3D. On the other hand, we visualize on each manifold the shortest 4D distance to all other manifolds, to convey proximity in 4D space.

Our 4D clipping sphere is centered at the camera center \mathbf{c} , and has an interactively defined radius r . Whereas the 4D space within the sphere is projected to the 3D image plane with orthographic projection, the outer part is projected along annular sectors to the 3D image plane (Figure 2(b)). As can be seen in this illustration, this has the effect that the outer part in 4D is also located outside the respective 3D sphere in the 3D image plane, and thus it cannot intersect with the 3D image part inside the sphere. The smaller r is chosen, the more projection-induced intersections are prevented

inside the sphere, but the less regions are projected to the central part, where high-opacity rendering proved useful. The part outside the sphere is distorted in 4D, but still provides consistent context in 3D, supporting navigation. Rendering the part outside the sphere with transparency follows a “shadows on the wall” metaphor in 4D (see, e.g., Figure 8(c)).

A drawback of the clipping sphere is that it constrains focus. We thus complement it by computing for each vertex of each manifold the shortest 4D distance to all vertices of all *other* manifolds. We take the logarithm of this distance field and map values close to one to high contribution in the green channel and high opacity (see, e.g., Figure 4(d)). One can see that this technique successfully reveals regions where manifolds meet in 4D; in particular it reveals saddle connectors [TWH03] in 4D. We leave the extraction of saddle connectors in 4D, however, as future work. Self-intersections of a manifold are not indicated, but these are always projection-induced because a manifold cannot self-intersect in 4D.

7. Results

In the following, we introduce datasets of increasing complexity (see Table 5), and use them for exemplifying our approach. The first five datasets were constructed for explanatory purposes, whereas the sixth one exemplifies the utility of our approach for analyzing the physical phase space of inertial motion. The last dataset was constructed randomly to provide a good coverage regarding complexity and variability. All of them were sampled on regular four-dimensional grids consisting of quadlinearly interpolated hypercube cells. Streamlines were integrated using the fourth-order Runge–Kutta scheme (RK4).

7.1. Linear Fields

Linear fields $\mathbf{u}(\mathbf{x}) = A\mathbf{x}$ are consistent with the classification of critical points, making them an ideal entry for demonstration. A matrix that exhibits a given set of eigenvalues can be constructed using a block-diagonal matrix A with a 1×1 block for each real eigenvalue, and a 2×2 block $(c, s, -s, c)$ for each complex conjugate pair $s + ic$.

Recall that we showed the glyphs of every type of critical points using one regular projection and two degenerate ones (Table 4). Since our glyphs are aligned with the invariant manifolds of the linearized field, the same projections that are degenerate for the glyphs, are degenerate for the entire separatrices. In the subsequent examples, we have chosen the 4D view direction $\hat{\mathbf{c}} = (1, 1, 1, 1)^\top / 2$

for depicting regular views, and the directions $\hat{\mathbf{c}} = (0, 0, 0, 1)^\top$ and $\hat{\mathbf{c}} = (1, 0, 0, 0)^\top$, resulting in degenerate views. These are explored in the accompanying video for the following two examples.

2:S2 Saddle We obtain a 2:s2 saddle by taking the eigenvalues $1 + i, 1 - i, -1, -1$. Such a saddle has a two-dimensional stable and a two-dimensional unstable rotating manifold. Both structures can be readily observed in regular views (Figure 3(a)). However, the manifolds intersect in such a view, whereas in 4D space, they only intersect in the critical point. This is revealed by the degenerate views (Figure 3(b), (c)), which show that each manifold can be projected to a line that intersects the other manifold only in a point.

3:1 Saddle We obtain a 3:1 saddle by choosing the eigenvalues $-1, -1, -1, 1$. Here, the regular view (Figure 3(d)) is fully occluded by the three-dimensional stable manifold. Furthermore, we find a degenerate projection (Figure 3(f)) that fully occludes the one-dimensional unstable manifold (red) as well. Projecting tangentially to the stable manifold, i.e., in direction of one of its eigenvectors, yields the least amount of clutter (Figure 3(e)).

7.2. Saddle Connectors

More complex configurations are obtained by placing two linear saddle points near each other. If stable and unstable directions of the respective saddles are aligned in a certain manner, it is possible to obtain saddle connectors [TWH03]. As the superposition of two linear fields is again a linear field, we introduce non-linearity by multiplication with Gaussian functions $w_i(\mathbf{x})$. Given matrices A_i , positions $\boldsymbol{\mu}_i$, and standard deviations σ_i , the vector field

$$\mathbf{u}(\mathbf{x}) = \sum_i w_i(\mathbf{x}) A_i (\mathbf{x} - \boldsymbol{\mu}_i), \quad w_i(\mathbf{x}) = \exp\left(-\sum_{j=1}^4 \frac{(x_j - \mu_{ij})^2}{2\sigma_i^2}\right),$$

permits the construction of arbitrary fields with specified configurations of critical points. We leave certain parameters fixed, i.e., $\boldsymbol{\mu}_1 = (0.5, 0, 0, 0)^\top$, $\boldsymbol{\mu}_2 = (-0.5, 0, 0, 0)^\top$, and $\sigma_i = 0.5$, and vary only the matrices.

Setting $A_1 = \text{diag}(-1, -1, -1, 1)$ creates a 3:1 saddle-type critical point at $\boldsymbol{\mu}_1$. We then set $A_2 = \text{diag}(1, -1, -1, 1)$, or $A_2 = \text{diag}(1, 1 - 1, 1)$. The former results in a saddle connector between a 3:1 saddle and a 2:2 saddle (Figure 5), while the latter yields a connector between a 3:1 saddle and a 1:3 saddle (Figure 4). Following the argumentation above, we again may find projections that show degenerate structures (Figures 4(a), 4(e), 5(a), 5(e)). Again, the non-degenerate view is occluded by three-dimensional manifolds, a problem that does not exist in the degenerate views of our linear fields, where volumes appear as surfaces (Figures 4(c), 5(c)).

None of these views, however, clearly shows the intersection of the manifolds in 4D space. Furthermore, some of the intersections in the projection are artifacts of the projection and do not exist in 4D space. Examining the distance field (Section 6.2), which is mapped to opacity and the green color channel, reveals proximity in 4D space (Figures 4(b), 5(b)). Regions shown in yellow and cyan belong to an intersection (hyper-)surface between two manifolds. We see that the saddle connector between a 3:1 and a 2:2 saddle is a curve (Figure 5(b)), while the one between a 3:1 and a 1:3 saddle is

Table 5: Sizes of the obtained manifolds for the different datasets.

Dataset	Vertices	Triangles	Tetrahedra
2:S2 Saddle	31976	61244	—
3:1 Saddle	47466	—	166548
Saddle Conn. 3:1–1:3	439972	—	2000148
Saddle Conn. 3:1–2:2	272979	87540	1016418
Five Critical Points	737955	538542	2101293
Acceleration Field	138016	—	449481
Random 4D Field	208961	340371	1236603

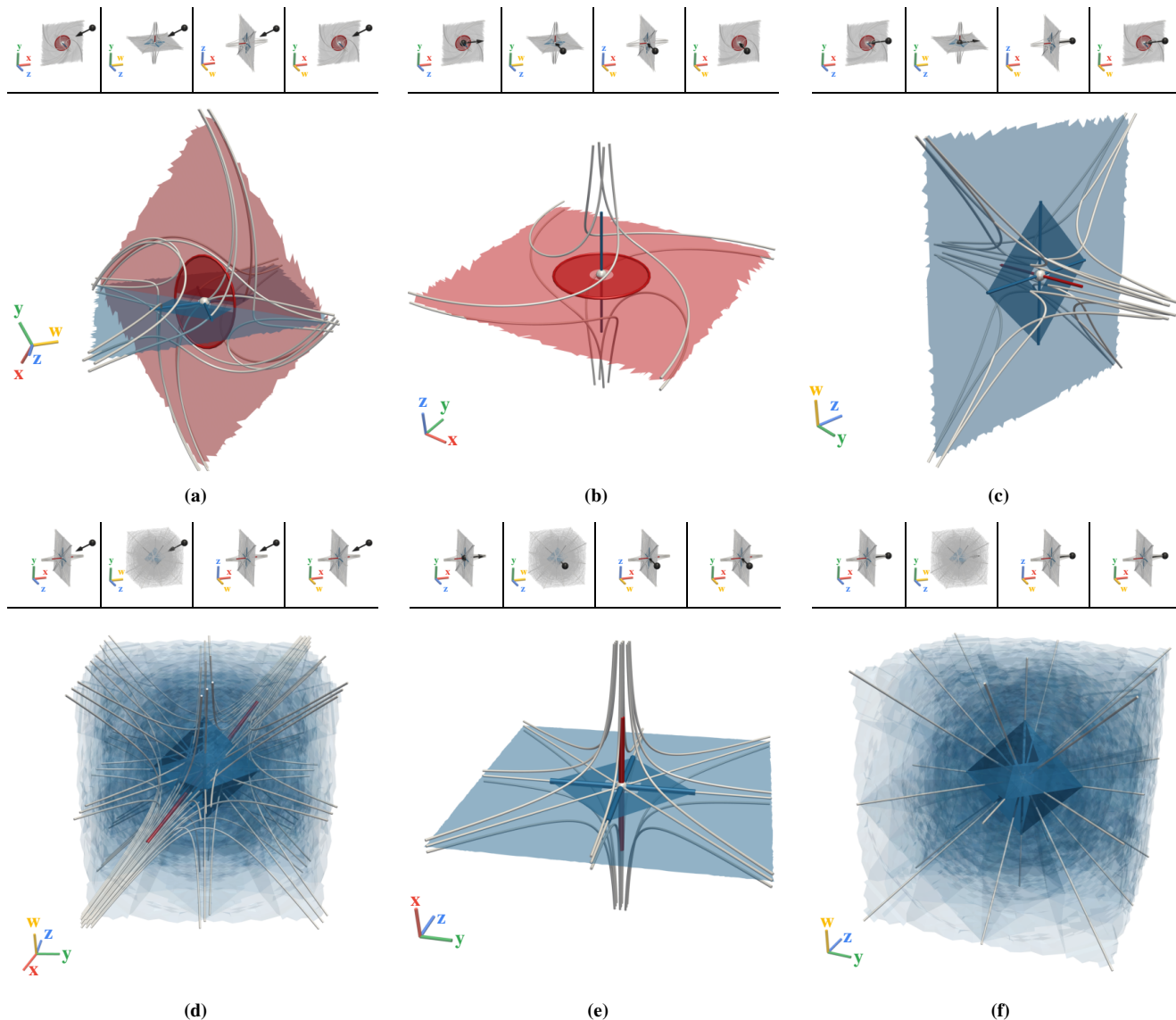


Figure 3: Two linear vector fields with different saddle-type critical points, a 2:s2 saddle (a)–(c) and a 3:1 saddle (d)–(f). Each case is shown in a regular view (a), (d), as well as two degenerate views (b), (c), and (e), (f). While the manifolds of the 2:s2 saddle cause no occlusion in all views, the three-dimensional manifold of the 3:1 saddle causes the least occlusion in the degenerate view (e).

a surface (Figure 4(b)). Finally, we observe that the distance field shows a spherical gradient near every critical point. This demonstrates that stable and unstable manifolds of the same critical point only intersect at a single point.

7.3. Five Critical Points

We construct a more complex example by placing five critical points using the technique described in Section 7.2. Here, we set all standard deviations to 0.75 and place five critical points in our vector field, a source at $(-1, -1, -1, -1)^T$, a 1:3 saddle at $(0, 0, 0, 0)^T$, a sink at $(1, 1, 1, 1)^T$, a 3:1 saddle at $(-1, -1, 1, 1)^T$, and a 2:2 saddle at $(1, 1, -1, -1)^T$. We first examine the glyphs of the critical points together with streamlines seeded near each of the critical

points (Figure 6(a)). This yields a rough overview of the vector field, with each type clearly visible. Next, we examine the same view together with all separatrices (Figure 6(b)). This projection exhibits many intersections and heavy occlusion near all critical points. The intersections in 4D space are revealed by the distance field (Figure 6(c)), which shows a saddle connector (yellow) between the 3:1 and 1:3 saddle. As opposed to the previously discussed datasets, this dataset is too complex to be examined using only the full view. As an example, we want to further examine the 1:3 saddle. Moving the 4D camera toward it leads to a cluttered view (Figure 6(d)). Using the clipping sphere, we are able to focus on the 4D neighborhood of the saddle (Figure 6(e)), however the view is still obstructed by the three-dimensional manifolds. Align-

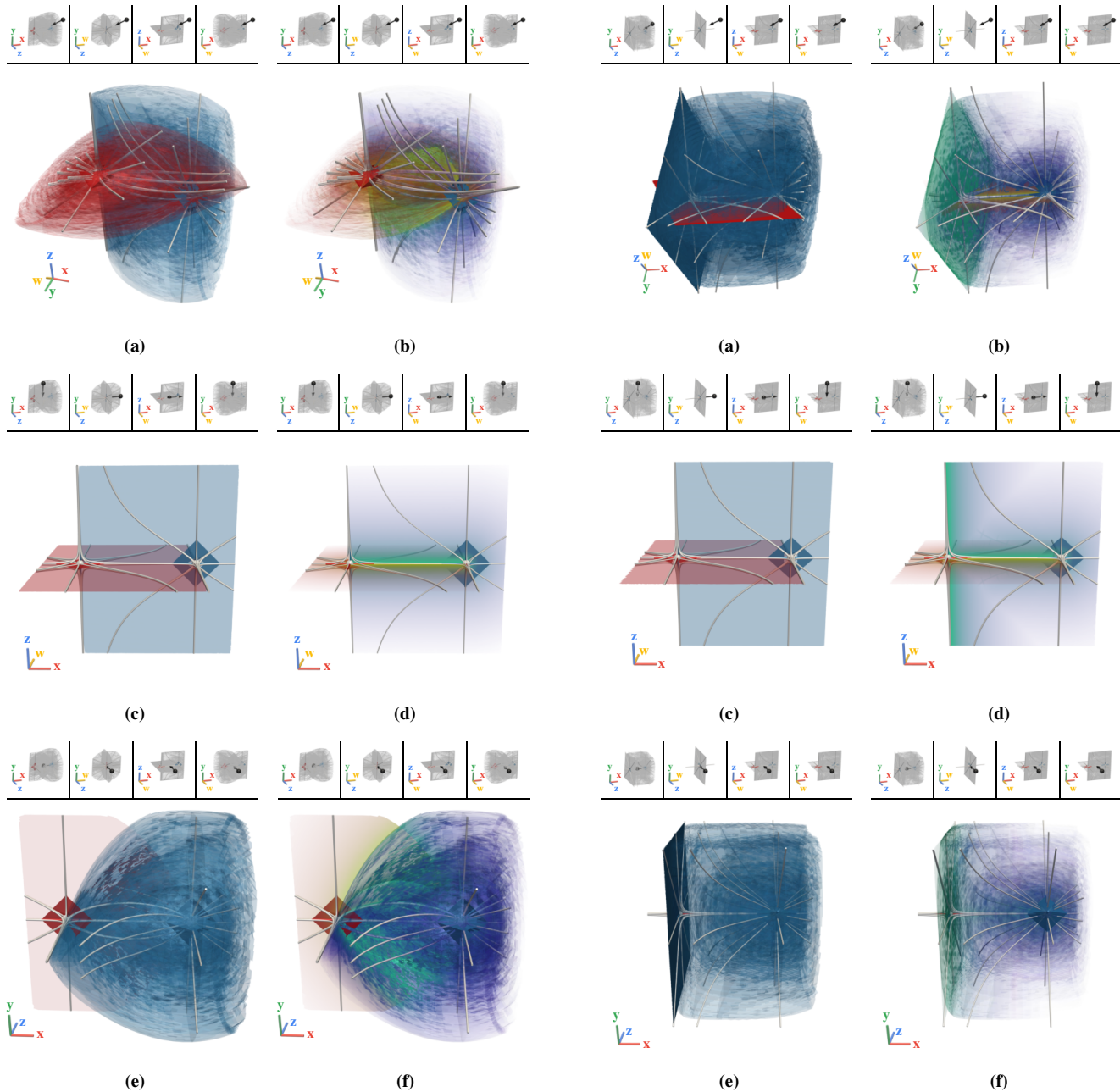


Figure 4: Surface-type saddle connector between a 3:1 saddle and a 1:3 saddle, without (left column) and with (right column) distance field (green) in regular (a), (b), and degenerate (c)–(f) views.

Figure 5: Line-type saddle connector between a 3:1 saddle and a 2:2 saddle. Same views as Figure 4, including distance field.

ing the camera with the manifold (Figure 6(f)) lets the viewer observe an intersection of the two manifolds near the critical point. Choosing a larger radius for the clipping sphere (Figure 6(g)), allows more context to be visible, but at the same time introduces an intersection of an additional two-dimensional manifold with the three-dimensional manifold of the critical point. We may additionally use the distance field to distinguish intersections caused by the projection, from actual intersections in 4D space (Figure 6(h)).

7.4. Acceleration Field

The dynamics of inertial objects due to acceleration fields is a prominent class of problems studied in physics. There, the concept of the physical phase space is employed, consisting of position and momentum/velocity. Our example mimics the dynamics of an inertial object in a planetary system consisting of five bodies with different masses, each effecting a gravitational field, and each exhibiting an atmosphere. There are configurations where such a system of bodies has a frame of reference within which their positions

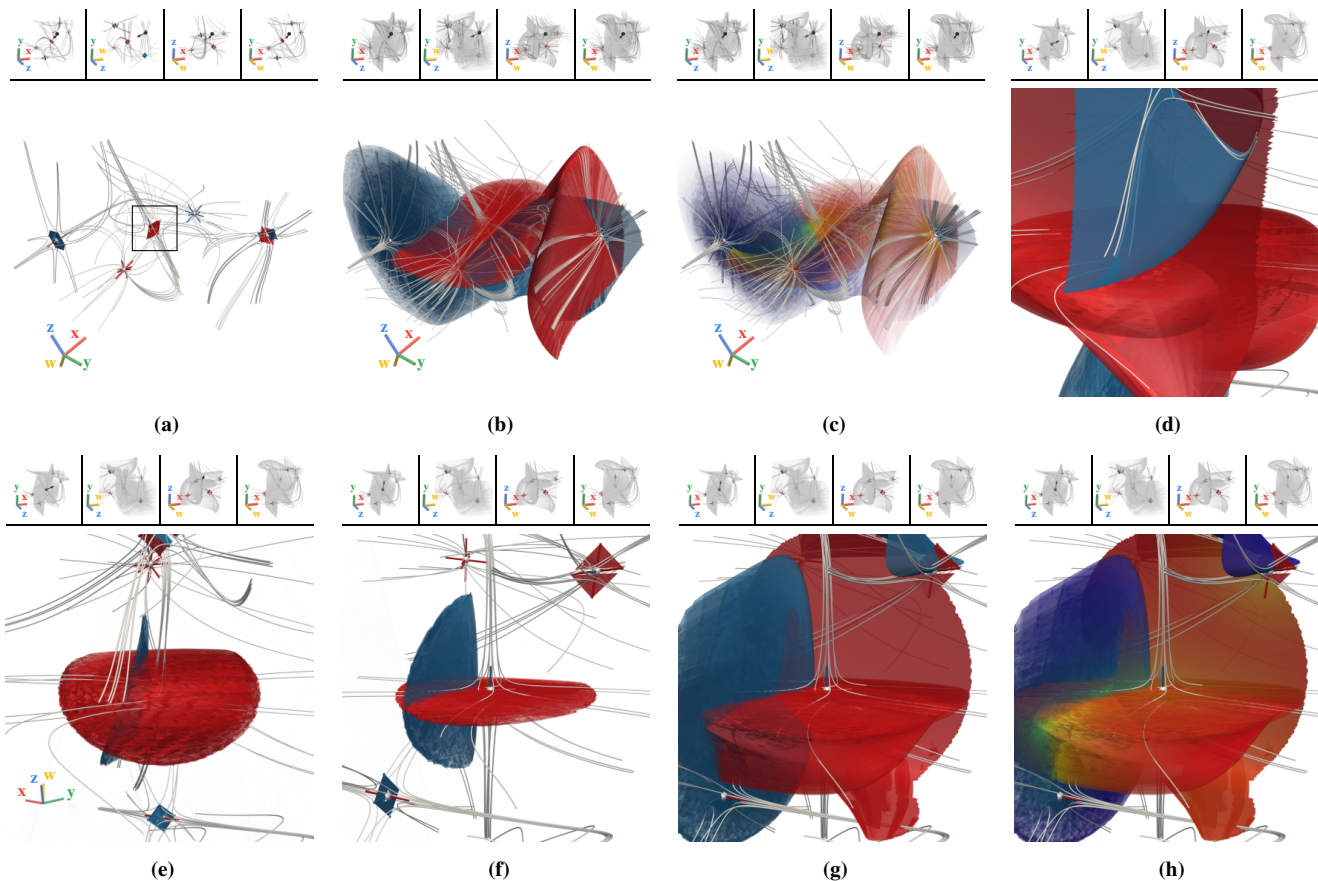


Figure 6: Five Critical Points dataset, overview using glyphs (a), separatrices (b), and the distance field mapped to green (c). Initially, no clipping of the 1:3 saddle (box in (a)) is performed (d). Using a small clipping sphere (e), and aligning the view to the manifold (f) yields the least cluttered view. The saddle connectors in the dataset can be revealed using either a larger clipping sphere (g) or the distance field (h).

do not vary, and where motion of the bodies is negligible compared to the influence of the object under consideration, also enabling a stationary model of the gravitational field. Thus, the acceleration of such an object depends solely on its position, and is defined by gravitational attraction and friction with the atmospheres.

We randomly place five point masses, i.e., the bodies, at positions $\xi_i \in \mathbb{R}^2$ with mass $m_i \in [1, 2]$ on the domain $[-5, 5]^2$ (see Figure 7(a)). We assume a gravitational constant $G = 1$, and consider the force due to the gravitational field and friction ($c = 10^{-3}$) on a particle with mass m_T at position $\xi \in \mathbb{R}^2$ and velocity $\dot{\xi} \in \mathbb{R}^2$, which finally yields a 4D phase space vector field:

$$\mathbf{F}(\xi, \dot{\xi}) = m_T \sum_{i=1}^5 \left[\frac{m_i (\xi_i - \xi)}{\|\xi_i - \xi\|^3} - \frac{c \|\dot{\xi}\| \dot{\xi}}{\|\xi_i - \xi\|^2} \right],$$

$$\mathbf{u}(\mathbf{x}) = \mathbf{u} \left(\begin{pmatrix} \xi \\ \dot{\xi} \end{pmatrix} \right) = \begin{pmatrix} \dot{\xi} \\ \mathbf{F}(\xi, \dot{\xi}) / m_T \end{pmatrix}.$$

Due to the friction term, the vector field has 2-spiral sink critical points at each of the mass points and s3:1 saddle points in between them (see Figure 7(b)). Because in this special case, the x - and y -components (red and green axes) of the 4D vector field describe spatial location, while the z - and w -components (blue and

yellow axes) describe velocity, we project, by default, the first two 3D thumbnail images onto the xy - and zw -planes, respectively (see Figure 7(b) top left corner). Even though this field describes a two-dimensional phenomenon, separatrices span all four dimensions (Figure 7(c)). The clutter in the projection is removed by aligning our 4D camera with one of the separatrices and using a small clipping sphere. Seeding particles on each side of a separatrix in 4D phase space demonstrates their separating property (Figure 7(d)).

7.5. Random 4D Field

Finally, we generate a dataset by choosing random values on a coarse $4 \times 4 \times 4 \times 3$ grid. It contains one 1-spiral sink, two 1-spiral sources, three 1:3 saddles, seven 1:s3 saddles, six s3:1 saddles, three 2:2 saddles, and one 2:s2, s2:2, and 2-spiral saddle each. An overview showing only the glyphs and streamlines seeded at the critical points shows its overall structure (Figure 8(a)). We choose a saddle-type critical point, align the camera with the corresponding manifold (Figure 8(c)), and avoid clutter in the image space, while providing context projected onto the background, using a small clipping sphere. Increasing the radius of the clipping sphere reveals more of the nearby structures (Figures 8(d) and 1).

L. Hofmann, B. Rieck, F. Sadlo / Visualization of 4D Vector Field Topology

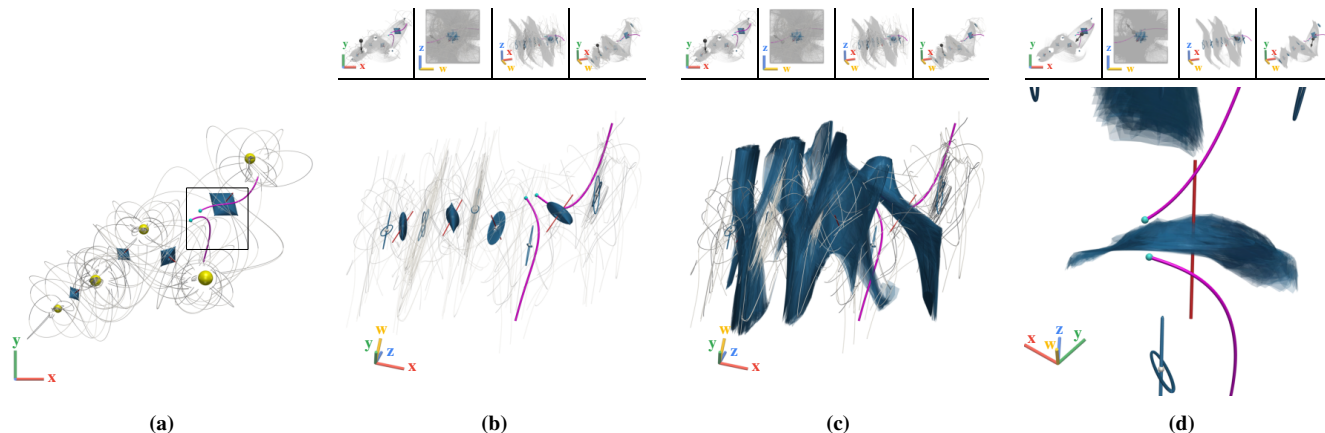


Figure 7: Acceleration Field dataset with five point masses (yellow spheres, scaled proportionally to mass) distributed on the 2D space domain (a), overview of the phase space using glyphs (b), and separatrices (c). Using a small clipping sphere, and seeding two trajectories (seeds cyan, trajectories magenta) near a saddle point (box in (a)) shows separating property of the manifold in phase space (d).

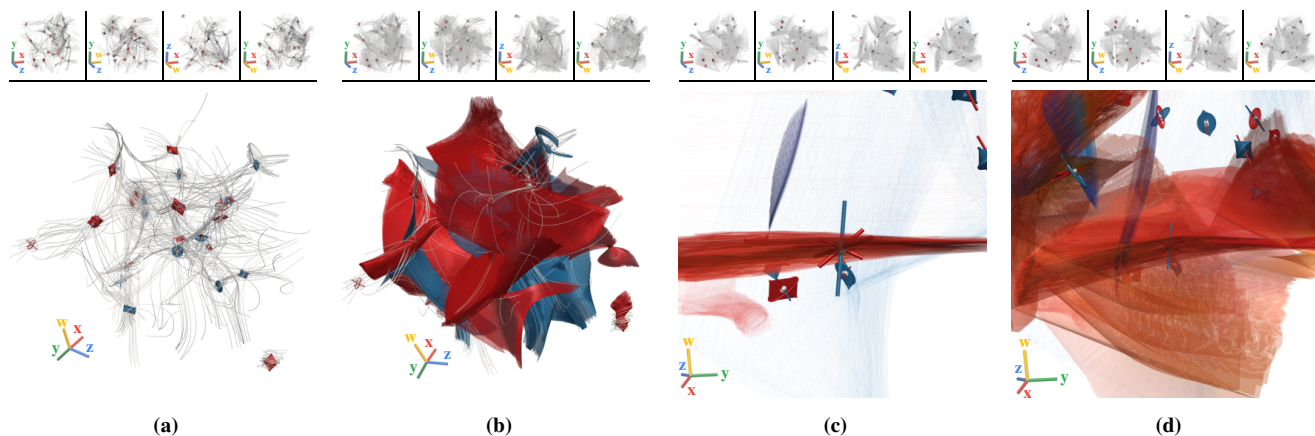


Figure 8: Random 4D Field dataset, with global structure shown using glyphs (a), and separatrices (b). The dataset is explored by aligning the view to a manifold (c) near a saddle point, providing a planar 3D projection, with further context projected in the background. Increasing the radius of the clipping sphere reveals more structures that are nearby (d). The last view with streamlines is shown in Figure 1.

8. Discussion

Our approach focuses on critical points and their invariant manifolds. While these are able to provide qualitative insight into a 4D vector field, for a more complete description, several building blocks are still missing. These include periodic orbits [KRRS14, WS02, WS01], saddle connectors [TWHS03], boundary switch curves [WTHS04], Cantori, as well as invariant and KAM tori, and bifurcation lines [MBES16, MSE13], some of which we aim to extend to the fourth dimension as future work.

There are some special cases, where 4D vector fields are obtained from spatially lower-dimensional phenomena. The first is regarding the space-time domain of a 3D time-dependent vector field as a 4D steady vector field. Such fields do not contain any critical points, since time moves constantly in forward direction (i.e., the last component is constant 1), and thus different approaches need to be employed, such as tracking of critical points [GTS04] or time-dependent topology [SW10, USE13, MBES16]. The 4D space-time domain has further been employed for the extraction

of swirling particle cores [WSTH07] and the computation of streak lines [WT10]. Second, the physical phase space of 2D steady inertial dynamics is a four-dimensional vector field, where two components correspond to space and two components to momentum. While, as presented by Günther and Gross [GG17], dynamics of flow-induced inertial dynamics can be reduced to the spatial dimension, general inertial dynamics need to be considered in the full $2n$ -dimensional phase space. Further, being restricted to the n -dimensional spatial domain, Günther and Gross cannot show separatrices in the $2n$ -dimensional phase space.

The case of *time-dependent* inertial dynamics adds another dimension, and is covered by Sagristà et al. [SJJ⁺17]. Our work can be applied to 2D steady inertial dynamics; however, we currently do not distinguish the different units of position and momentum. For a quantitative analysis, a more specific adaption would be needed. We show such an example in Section 7.4. Since our method relies on separatrices, and thus focuses on the qualitative analysis of the system, mixing scales and units is not an issue.

Unfortunately, we were unable to find complex (simulated) real-world datasets. Even though many problems with, e.g., a three-dimensional spatial domain depend on a fourth dimension such as pressure or concentration, most numerical simulations treat these as attribute only, instead of computing dense ensembles. Such simulations thus do not output the full four-dimensional state space, but a three-dimensional submanifold for, say, isolated initial conditions. We hope to see such data in the future.

9. Conclusion

In this work, we classified critical point types in 4D vector fields, developed a novel set of glyphs, and presented an approach for the interactive visualization of 4D scenes by means of a 4D and a 3D camera involving 3D images. We also developed a manifold-based exploration technique by exploiting degeneracies that enable us to visualize the structure of 3D manifolds without massive occlusion, and presented two strategies for avoiding projection-induced intersection of 4D manifolds in 3D projection. The first one uses a modified projection scheme, while the second one involves 4D distance fields between invariant manifolds. We demonstrated the utility of our approach using vector fields that showcase properties of 4D topology. As future work, we plan to develop techniques for the visualization of periodic orbits and saddle connectors.

References

- [Asi93] ASIMOV D.: *Notes on the Topology of Vector Fields and Flows*. Tech. Rep. RNR-93-003, NASA Ames Research Center, 1993. 1
- [CFHH09] CHU A., FU C.-W., HANSON A., HENG P.-A.: GL4D: A GPU-based architecture for interactive 4D visualization. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1587–1594. 2
- [GG17] GÜNTHER T., GROSS M.: Flow-induced inertial steady vector field topology. *Computer Graphics Forum* 36, 2 (2017), 143–152. 2, 12
- [GTS04] GARTH C., TRICOCHÉ X., SCHEUERMANN G.: Tracking of vector field singularities in unstructured 3D time-dependent datasets. In *Proc. IEEE Conference on Visualization* (2004), pp. 329–336. 2, 12
- [HC93] HANSON A. J., CROSS R. A.: Interactive visualization methods for four dimensions. In *Proc. IEEE Conference on Visualization* (1993), pp. 196–203. 2
- [HH89] HELMAN J., HESSELINK L.: Representation and display of vector field topology in fluid flow data sets. *IEEE Computer* 22, 8 (1989), 27–36. 2
- [HH91] HELMAN J. L., HESSELINK L.: Visualizing vector field topology in fluid flows. *IEEE Computer Graphics and Applications* 11, 3 (1991), 36–46. 2
- [HH92] HANSON A. J., HENG P. A.: Illuminating the fourth dimension. *IEEE Computer Graphics and Applications* 12, 4 (1992), 54–62. 2
- [HLH*16] HEINE C., LEITTE H., HLAWITSCHKA M., IURICICH F., DE FLORIANI L., SCHEUERMANN G., HAGEN H., GARTH C.: A survey of topology-based methods in visualization. *Computer Graphics Forum* 35, 3 (2016), 643–667. 2
- [Hol91] HOLLASCH S. R.: *Four-space Visualization of 4D Objects*. Master's thesis, Arizona State University, 1991. 2, 5
- [Hul92] HULTQUIST J. P.: Constructing stream surfaces in steady 3D vector fields. In *Proc. IEEE Conference on Visualization* (1992), pp. 171–178. 6
- [KRRS14] KASTEN J., REININGHAUS J., REICH W., SCHEUERMANN G.: Toward the extraction of saddle periodic orbits. In *Topological Methods in Data Analysis and Visualization III*. Springer, 2014, pp. 55–69. 12
- [LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH Computer Graphics* 21, 4 (1987), 163–169. 4
- [LHZP07] LARAMEE R. S., HAUSER H., ZHAO L., POST F. H.: Topology-based flow visualization: The state of the art. In *Topology-based Methods in Visualization*. Springer, 2007, pp. 1–19. 2
- [MBC93] MAX N., BECKER B., CRAWFIS R.: Flow volumes for interactive vector field visualization. In *Proc. IEEE Conference on Visualization* (1993), pp. 19–24. 2, 6
- [MBES16] MACHADO G. M., BOBLEST S., ERTL T., SADLO F.: Space-time bifurcation lines for extraction of 2D Lagrangian coherent structures. *Computer Graphics Forum* 35, 3 (2016), 91–100. 12
- [MSE13] MACHADO G. M., SADLO F., ERTL T.: Local extraction of bifurcation lines. In *Proc. International Workshop on Vision, Modeling and Visualization* (2013), pp. 17–24. 12
- [Nol67] NOLL A. M.: A computer technique for displaying n -dimensional hyperobjects. *Communications of the ACM* 10, 8 (1967), 469–473. 2, 5
- [SJJ*17] SAGRISTÀ A., JORDAN S., JUST A., DIAS F., NONATO L. G., SADLO F.: Topological analysis of inertial dynamics. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 950–959. 12
- [SW10] SADLO F., WEISKOPF D.: Time-dependent 2-D vector field topology: An approach inspired by Lagrangian coherent structures. *Computer Graphics Forum* 29, 1 (2010), 88–100. 12
- [TWHS03] THEISEL H., WEINKAUF T., HEGE H.-C., SEIDEL H.-P.: Saddle connectors - an approach to visualizing the topological skeleton of complex 3D vector fields. In *Proc. IEEE Conference on Visualization* (2003), pp. 225–232. 4, 7, 8, 12
- [USE13] ÜFFINGER M., SADLO F., ERTL T.: A time-dependent vector field topology based on streak surfaces. *IEEE Transactions on Visualization and Computer Graphics* 19, 3 (2013), 379–392. 12
- [vW93] VAN WIJK J. J.: Implicit stream surfaces. In *Proc. IEEE Conference on Visualization* (1993), pp. 245–252. 2
- [WB96] WEIGLE C., BANKS D. C.: Complex-valued contour meshing. In *Proc. IEEE Conference on Visualization* (1996), pp. 173–180. 2
- [WS01] WISCHGOLL T., SCHEUERMANN G.: Detection and visualization of closed streamlines in planar flows. *IEEE Transactions on Visualization and Computer Graphics* 7, 2 (2001), 165–172. 12
- [WS02] WISCHGOLL T., SCHEUERMANN G.: Locating closed streamlines in 3D vector fields. In *Proc. Symposium on Data Visualisation 2002* (2002), pp. 227–232. 12
- [WSTH07] WEINKAUF T., SAHNER J., THEISEL H., HEGE H.-C.: Cores of swirling particle motion in unsteady flows. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1759–1766. 12
- [WT10] WEINKAUF T., THEISEL H.: Streak lines as tangent curves of a derived vector field. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (2010), 1225–1234. 12
- [WTHS04] WEINKAUF T., THEISEL H., HEGE H.-C., SEIDEL H.-P.: Boundary switch connectors for topological visualization of complex 3D vector fields. In *Proc. Eurographics Conference on Visualization* (2004), pp. 183–192. 12
- [WTHS06] WEINKAUF T., THEISEL H., HEGE H.-C., SEIDEL H.-P.: Topological structures in two-parameter-dependent 2D vector fields. *Computer Graphics Forum* 25, 3 (2006), 607–616. 2
- [WYF*13] WANG W., YAN X., FU C., HANSON A., HENG P.: Interactive exploration of 4D geometry with volumetric halos. In *Short Papers Proc. Pacific Conference on Computer Graphics and Applications* (2013), pp. 1–6. 2
- [XZC04] XUE D., ZHANG C., CRAWFIS R.: Rendering implicit flow volumes. In *Proc. IEEE Conference on Visualization* (2004), pp. 99–106. 2