Flow Map Processing by Space-Time Deformation

Thomas Wilde, Christian Rössl, and Holger Theisel

Otto-von-Guericke University Magdeburg, Germany

Abstract. In Flow Visualization, the consideration of flow maps instead of velocity fields has recently moved into the focus of research. We present an approach to transforming standard techniques in vector field processing – like smoothing, modeling, deformation – to flow maps. This requires a solution to the fundamental problem that – contrary to vector fields – a specific modification of the flow map is, in general, not a flow map anymore. We introduce a concept that enables the modification of discrete sampling of a flow map while enforcing the flow map properties. Based on this, we present approaches for flow map deformation that are applied to a 2D time-dependent flow field.

Keywords: Flow Visualization · Flow map · Unsteady vector fields.

1 Introduction

In Flow Visualization, a subfield of Scientific Visualization, vector fields describing the velocity of a flow are usually in the focus of interest. In addition to just visualizing vector fields, there are also established approaches to model, construct, deform, simplify, and compress vector fields. We call these approaches *vector field processing* and consider them as an inevitable part of Flow Visualization. In recent years, Flow Visualization techniques that do not focus on the velocity field but the *flow map* have gained increased interest. This is motivated by the fact that the flow map directly encodes all particle trajectories (path lines) without the necessity of numerical integration. Flow maps are complex data structures: they are computationally expensive, high-dimensional, have highly connected data, and strong gradients [20]. Considering flow maps is challenging but rewarding. This brings up the question for the systematic *processing of flow maps*, similar to vector field processing. The main challenge for flow map processing is the complexity of flow maps and another fundamental problem: a small local modification would destroy inherent flow map properties, and therefore the result is not a flow map anymore. To tackle this problem, we present the following main contributions: first, we give solid foundations to characterize properties of general flow maps. Afterward, we introduce a concept for modification techniques that keep these properties and apply this to a benchmark data set.

2 Related Work

Vector field design: Vector fields have been of research interest over the past decades in various fields. Theisel [17] presents an early approach to designing 2D steady vector fields based on a set of control polygons and extracted topologic features. Weinkauf et al. [19] extend this to a solution based on topological skeletons of vector fields to 3D. Zhang et al. [21, 22] present an interactive design system that uses similar metaphors by placing singularities in the field. Chen et al. [4] go a step further and contribute an element-based framework for designing unsteady vector fields on manifold surfaces. Recently Baeza [1] presented a new approach to the extraction of time-dependent vector field topology. For a more complete overview, we refer to the work of Vaxman[18]. Several authors utilize vector fields to solve other problems, e.g., von Funck et al. [5, 6] present vector field based shape deformations. We formulate our approach for flow map modification similarly.

Flow maps: Several papers use the Lagrangian point of view for flow visualization - they directly or indirectly use flow maps. The majority of state-of-the-art flow visualization techniques are based on advection and therefore utilize at least parts of the flow map [3, 13–15]. Hlawatsch et al. [10] perform a concatenation of short-time flow maps for fast parallel computation of path lines. Hummel et al. [12] give an error estimation for this kind of path line computation on short-time flow maps. Wilde et al. [20] present a distance function based on the full 5D flow map used to extract closed surfaces with recirculation behavior. Hofmann [11] introduces a generic vector operator that is also capable of computing sparse samplings on recirculation surfaces. One of the most prominent tools for the visualization of unsteady flow behavior are FTLE fields, as described by Haller et al. [7, 8]. FTLE indicates the separation of colocated particles and is based on the spatial flow map gradient.

3 Flow Maps and Velocity Fields

Notation: We depict scalar values by lower case letters, e.g., time t, vectorial data is denoted by lower case bold letters, e.g. position $\mathbf{x}$, an uppercase letter describes a set of elements or an area, e.g. S. We use a tilde for modified values, e.g., $\widetilde{\phi}$.

Definitions: We denote a *time-dependent vector field* $\mathbf{v}$, evaluated at position $\mathbf{x}$ and time t as:

$$\mathbf{v} : D \times T_t \rightarrow \mathbb{R}^n \;\; : \;\; (\mathbf{x}, t) \rightarrow \mathbf{v}(\mathbf{x}, t) \tag{1}$$

where $D \subseteq \mathbb{R}^n$ is the spatial domain, and $T_t = [t_s, t_e]$ is the finite time interval in which we define the velocity field. Furthermore, $\mathbf{v} \in C^1$, i.e., $\mathbf{v}$ is sufficiently smooth. In practice, both the spatial and temporal domains are limited to a bounded region and a time interval. The path of a massless particle that is advected by the underlying velocity field is called a *path line*. We compute a

path line starting at $\mathbf{x}_0$ and time t_0 by integrating $\mathbf{v}$ over a time interval $t \in [t_0, t_0 + \tau] \subseteq T$. The corresponding *flow map* for the vector field $\mathbf{v}(\mathbf{x}, t)$ is given by:

$$\phi_\tau^t : D \times T_t \times T_\tau \to D : (\mathbf{x}, t, \tau) \to \phi(\mathbf{x}, t, \tau) \tag{2}$$

with $T_\tau = [\tau_s, \tau_e]$. ϕ maps a particle seeded at position $\mathbf{x}$ and time t to its destination. τ denotes the integration time, i.e., the time which passes between particle seeding and reaching its destination. We write the flow map equivalently as $\phi = \phi(\mathbf{x}) = \phi_\tau^t$, whenever position $\mathbf{x}$, starting time t, or integration time τ are clear from the context. The flow map encodes path lines. A path line starting at $\mathbf{x}_0$ and t_0 can be computed by an evaluation of $\phi(\mathbf{x}_0, t_0, \tau)$ over a time interval $\tau \in [\tau_0, \tau_1] \subseteq T_\tau$. The time derivative with respect to τ is the direction in which a massless particle currently moves and is defined as:

$$\phi_\tau(\mathbf{x}, t, \tau) = \frac{\delta}{\delta\tau}\phi(\mathbf{x}, t, \tau) \tag{3}$$

Properties and Relations: A map $D \times T_t \times T_\tau \to D$ must fulfill the following properties to be a flow map:

$$\phi(\mathbf{x}, t, 0) = \mathbf{x} \tag{4}$$

$$\phi(\phi(\mathbf{x}, t, \tau_1), t + \tau_1, \tau_2) = \phi(\mathbf{x}, t, \tau_1 + \tau_2) \tag{5}$$

$$\phi(\phi(\mathbf{x}, t, \tau_1), t + \tau_1, -\tau_1) = \mathbf{x} \tag{6}$$

for all $\mathbf{x} \in D, t \in T_t, t + \tau_1 \in T_t, t + \tau_1 + \tau_2 \in T_t$. Eq. (4) denotes the flow map's *identity* for the particular case $\tau = 0$, i.e., a particle is mapped to its starting position if there is no integration time. Eq. (5) describes the flow map's *additivity*, i.e., the particle moves on its path line in several small consecutive steps or one big step. Eq. (6) denotes the *inversion* of the flow map, which is a particular case of additivity. If τ is inverted, a massless particle has to move backward on its path, which also means it will reach its starting position again. Assuming a velocity field $\mathbf{v}(\mathbf{x}, t)$ and the corresponding flow map $\phi(\mathbf{x}, t, \tau)$ are given, both smooth and continuous. The following relations hold:

$$\phi_\tau(\mathbf{x}, t, 0) = \mathbf{v}(\mathbf{x}, t) \tag{7}$$

$$\phi_\tau(\mathbf{x}, t, \tau) = \mathbf{v}(\phi(\mathbf{x}, t, \tau), t + \tau) \tag{8}$$

i.e., the velocity field is the partial derivative of the flow map with respect to the integration time τ. The flow map $\phi_\tau^t(\mathbf{x})$ is a function mapping $\phi : \mathbb{R}^{n+2} \to \mathbb{R}^n$, where n denotes the number of spatial dimensions. The remaining two dimensions are the starting time t and integration time τ. In practice, the use of 2D or 3D unsteady flows is common. In this paper, we also use the 1D case ($\phi : \mathbb{R}^3 \to \mathbb{R}$) to clarify concepts. However, all presented flow map properties and concepts hold for higher dimensions.

4 Flow Map Modification

This section introduces the theoretical background for modifying flow maps based on a space deformation. Afterward, we develop a shape-based method,

which implements this approach and could be the basis for an interactive transformation tool. The flow map explicitly encodes the path lines. We use them to illustrate the presented ideas and concepts for the 1D and 2D cases.

Modification based on space deformation: Given a flow map ϕ, we want to apply a local modification in space-time such that the new map $\widetilde{\phi}$ is a flow map again. The mapping of positions in space, at selected domain locations in space-time, must be modified. An example may be to pick a path line and locally change its shape. Due to continuity reasons, a modification must also affect the region near the modified path line, i.e., we also have to change the shape of adjacent path lines, but in a certain distance (in space-time), all remaining path lines are untouched. We can model such behavior by defining a space deformation. A space deformation $\mathbf{y} : \mathbb{R}^{n+1} \to \mathbb{R}^n$ maps a point $(\mathbf{x}, t)$ in space-time to the new point $\mathbf{y}(\mathbf{x}, t)$. We expect $\mathbf{y}$ to be local, continuous, and invertible:
Local: it affects only a region of the space-time domain, unless $\mathbf{y}$ is the identity
Continuous: $\mathbf{y}$ is at least C^1
Invertible: $\nabla_{\mathbf{x}}\mathbf{y}$ does have full rank, i.e., the inverse map $\mathbf{y}^{-1}$ is well-defined.

We compute the new flow map $\widetilde{\phi}$ from ϕ and $\mathbf{y}$ by:

$$\widetilde{\phi}(\mathbf{x}, t, \tau) = \mathbf{y}(\phi(\mathbf{y}^{-1}(\mathbf{x}, t), t, \tau), t + \tau) \tag{9}$$

Note that $\widetilde{\phi}$ is a flow map as well; we give a proof for this in the appendix. The transformation from ϕ to $\widetilde{\phi}$ is local in space-double-time, i.e., position $\mathbf{x}$, time t, and integration time τ. If both $\mathbf{y}(\mathbf{x}, t)$ and $\mathbf{y}(\phi(\mathbf{y}^{-1}(\mathbf{x}, t), t, \tau), t + \tau)$ are the identity, we have $\widetilde{\phi} = \phi$. A modification must preserve, all flow map defining properties (Sec. 3). Flow maps are high dimensional and have highly connected data, i.e., it is not possible to change the flow map in a small area without breaking the global flow map properties. A small change always has a global impact in the space-time domain. This makes modifications of flow maps a challenging task. In the following, we propose an approach to solving this task, divided into two steps:
1. Define a local area (in space-time) to perform a modification.
2. Globally identify all influenced areas and adapt them.

4.1 Definition of Modification Area

For an interactive manipulation of flow maps, we follow a metaphor using interactive shape modeling by deformations [2, 5] and adapt this to flow map modeling. We define the shape S in space-time, with n dimensions in space and 1 dimension in time; therefore, $S \subset D \times T_{t+\tau}$. All possible $(t + \tau)$-combinations have to be covered, i.e., $T_{t+\tau} = T_t \cup T_\tau = [t_0 - \tau_e, t_0 + 2\tau_e]$. S describes an area of effect, in which the modification is performed. It is placed interactively at a user-defined location. We divide the shape into three regions:
1. an inner region, with a rigid modification
2. an outer region, where no modification takes place

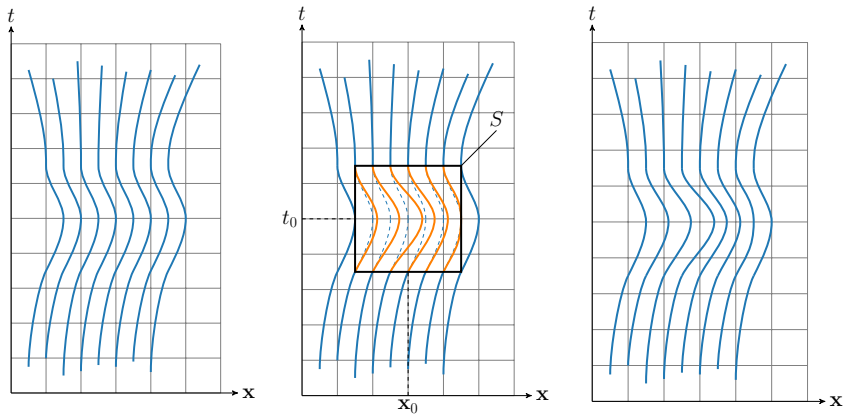

Fig. 1. Left: original path lines. Middle: process of modification, it is strongest in the middle of S. Right: smooth modified path lines.

3. an intermediate region, where the deformation is computed by an energy minimization to ensure continuity and other useful properties.

We define a local space deformation $\mathbf{y}$ by setting the following values for the modification shape S: $\mathbf{x}_0, t_0, r_1, r_2, r_t, \mathbf{d}_0$. Here $(\mathbf{x}_0, t_0)$ denotes the center position of S in space-time, $\mathbf{d}_0$ denotes a transformation vector. Furthermore we demand $0 \leq r_1 < r_2$, $0 < r_t$, $\|\mathbf{d}_0\| < r_2 - r_1$. Then we define a space deformation as

$$\mathbf{y}(\mathbf{x}, t) = \mathbf{x} + \lambda \mathbf{d}_0 \tag{10}$$

$$\lambda = \lambda_s \lambda_t, \qquad \lambda_s = (s_{\mathbf{x}}^3 + 3s_{\mathbf{x}}^2(1 - s_{\mathbf{x}})), \qquad \lambda_t = (s_t^3 + 3s_t^2(1 - s_t)) \tag{11}$$

$$r = \|\mathbf{x} - \mathbf{x}_0\| \tag{12}$$

$$s_{\mathbf{x}} = \begin{cases} 1 & r < r_1 \\ \frac{r - r_2}{r_1 - r_2} & r_1 \leq r \leq r_2 \\ 0 & r > r_2 \end{cases} \qquad s_t = \begin{cases} 1 - \frac{|t - t_0|}{r_t} & |t - t_0| < r_t \\ 0 & \text{else} \end{cases}. \tag{13}$$

λ is a scaling factor for the deformation, defined in the dependency of S. λ is 0 next to the border (no modification) and 1 at the center (full modification) of S. Fig. 2 (left) gives an illustration of S for the 1D case. This definition of the modification shape S leads to a smooth transition from original to modified areas in the ϕ domain. Fig. 1 gives an example of a modification of path lines for the 1D case. We place S at a specific location $(\mathbf{x}_0, t_0)$, e.g., guided by the user. Fig. 1 (left) shows path lines extracted from the flow map. Note that they are defined in space-time and do *not* necessarily start or end at the same time t. Fig. 1 (middle) illustrates a modification in the form of a translation in a positive $\mathbf{x}$ direction. We show the modified path line pieces in orange; the pieces stay inside of S. Outside of the modification shape $\mathbf{y}$ maps to the identity, i.e., $\widetilde{\phi} = \phi$. This also holds for the particular case $\tau = 0$. Fig. 1 (right) shows the final result. Note that by construction, $\mathbf{y}$ is C^1 continuous and invertible. However,

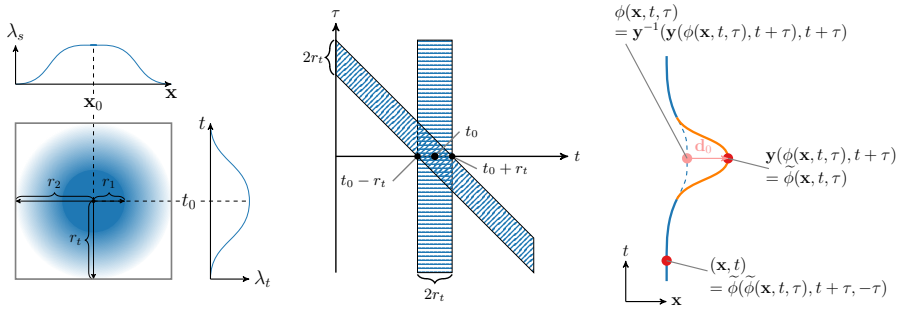

Fig. 2. Left: Illustration of the modification shape S for the 1D case. Middle: Areas with (t, τ)-pairs in which ϕ has to be updated. Right: The effect of $\mathbf{y}$ for two locations on a single path line.

the inverse $\mathbf{y}^{-1}$ does not have a simple formula. Because of this, we numerically precompute $\mathbf{y}^{-1}$ on a uniform grid in a space-time box with the same size as S.

4.2 Local Modification and Global Adaption

Having defined the space deformation $\mathbf{y}$, we have to update the flow map ϕ accordingly by applying Eq. (9). Since $\mathbf{y}$ is local, i.e., only in a specific area, it is not the identity, $\widetilde{\phi}$ is local as well. To ensure the flow map properties, we need to identify the parts, which need to be adapted in the process of a local modification. These parts can be classified into two groups:

1. parts mapping from *inside* S to its *outside*, i.e. $\phi(\mathbf{x}, t, \tau) \notin S$ with $(\mathbf{x}, t) \in S$
2. parts mapping to the *inside* of S, i.e., $\phi(\mathbf{x}, t, \tau) \in S$

Keep in mind that S is defined in space-time. Therefore the terms *inside* and *outside* also refer to space-time. For the spatial part, we have to check if a location $\mathbf{x}$ or its mapping $\phi(\mathbf{x}, t, \tau)$ belongs to S. Regarding the double-time dimensions, only specific (t, τ)-combinations are relevant. The new flow map must be computed only for (t, τ)-pairs with $(t_0 - r_t) \leq (t + \tau) \leq (t_0 + r_t)$. Fig. 2 (middle) illustrates the relevant parts in (t, τ) domain. Regarding a single path line from the example given in Fig. 1, the flow map needs to be adapted for all locations on this path line, where a (t, τ)-pair maps into the modification area S. Fig. 2 (right) illustrates this adaption for two locations on a path line. If $\phi(\mathbf{x}, t, \tau)$ is modified, $\phi(\phi(\mathbf{x}, t, \tau), t + \tau, -\tau)$ has to be adapted as well.

4.3 Discretization of the Flow Map

ϕ is defined as a continuous map, describing particle trajectories in flows from the Lagrangian reference frame. In practice, it is discretized in space and time. We discretize the domain over a regular grid. For the discretization scheme, we require the parameters $h_{\mathbf{x}}$ and h_t. They denote the constant distance between

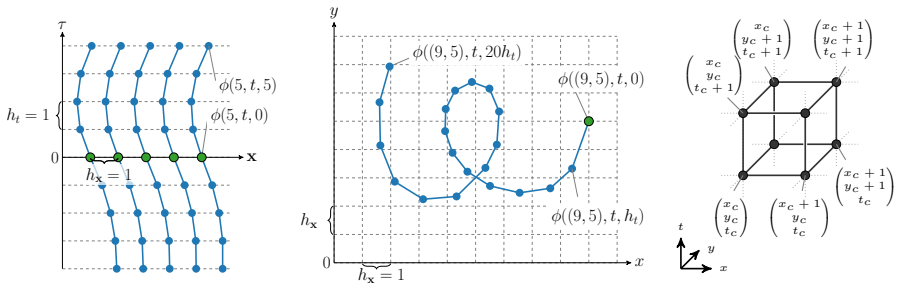

Fig. 3. Left: Sampled path lines for different $\mathbf{x}$, fixed t, and consecutive τ in 1D. Middle: A path line for a fixed $\mathbf{x}$, fixed t, and consecutive τ in 2D. Green circles denote $(\mathbf{x}, t)$-coordinates of samples. Blue circles indicate τ-coordinates and mapping destinations. Right: Illustration of a lookup-cell in (x, y, t)-space.

two samples in the spatial dimensions, e.g. (x, y) and the temporal dimensions t and τ. This discretization of the domain results in a single *slice* for each (t, τ)-combination, containing the spatial grid samples. We perform an integration in the underlying vector field for each discrete $(\mathbf{x}, t, \tau)$-sample. The corresponding trajectory endpoint is stored for this sample and builds a single entry in the discretized flow map. This way, we get a discrete approximation of the flow map in each dimension. Fig. 3 gives an example of the 1D and 2D cases. The access to the flow map entries is given by $(\mathbf{x}, t, \tau)$-coordinates for each sample. Furthermore, each sample gets a unique ID. Flow map values between discrete samples are obtained by linear interpolation of the corresponding neighboring samples. Path lines for a fixed $(\mathbf{x}, t, 0)$-position can be extracted by reading the entries for consecutive τ-coordinates.

Modification of the discrete Flow Map: When we perform the modification on the discrete flow map, we have to adapt the discrete samples' entries. As stated out, we classify the involved samples into two groups:

1. samples located *inside* the modification area S mapping *outside*
2. samples mapping *into* the modification area S .

The first group are the samples covered by the modification area S and are there-

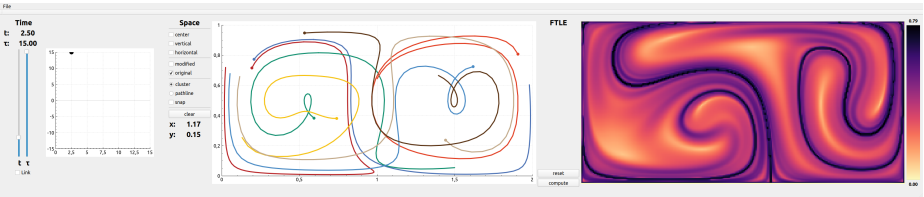

Fig. 4. The path line explorer. Left: (t, τ)-space selection. Middle: random path lines. Right: FTLE visualization. Path lines originate at the dots.

fore easy to identify. The members of the second group are not so easily identifiable. Potentially we have to check all samples with a suitable (t, τ)-combination (see Fig. 2), for each modification. But in general, only a small group of samples maps into S. To cope with this problem, we introduce a further data structure.

Lookup-Cells for fast sample retrieval: Based the flow map sampling, we divide the discretization into *lookup-cells*, defined by $(\mathbf{x}, t)$-coordinates, i.e., we omit one temporal coordinate τ. The adjacent samples of a location in $(\mathbf{x}, t)$-space build the vertices of the surrounding lookup-cell. For the 2D, case a lookup-cell C at the sample $(\mathbf{x}_c, t_c)$ is given by the following coordinates:

$$C_{(\mathbf{x}_c,t_c)} := (x_c, y_c, t_c), \quad (x_c+1, y_c, t_c), \quad (x_c, y_c+1, t_c), \quad (x_c+1, y_c+1, t_c),$$
$$(x_c, y_c, t_c+1), (x_c+1, y_c, t_c+1), (x_c, y_c+1, t_c+1), (x_c+1, y_c+1, t_c+1).$$

Fig. 3 (right) gives an illustration. $2^{(n+1)}$ neighboring samples make up a cell. Vice versa, each discrete sample is part of up to $2^{(n+1)}$ adjacent cells. Each lookup-cell $C_{(\mathbf{x}_c,t_c)}$ holds the *IDs* of samples mapping into its volume in discretized space-time. The set P of these samples for a single cell is given by:

$$P_{C_{(\mathbf{x}_c,t_c)}} := \{(\mathbf{x}, t, \tau) \mid \phi(\mathbf{x}, t, \tau) \in C_{(\mathbf{x}_c,t_c)}\} \quad with \quad t_c \leq (t+\tau) \leq t_c + 1$$

This way, fast retrieval of all parts that need to be adapted is possible. The temporal domain of the lookup-cells covers the whole (t, τ)-space, i.e., $T_t \times T_\tau$.

5 Implementation

After precomputing the complete flow map in a particular resolution, we can reformulate standard visualization approaches based on path line integration as a simple array lookup. Based on this, we implemented an interactive exploration tool for unsteady 2D flows. Since the path line starting at $(\mathbf{x}, t)$ in a parametrization of τ is just $\phi(\mathbf{x}, t, \tau)$, we can compute every point by a quadrilinear interpolation in the sampled flow map. Note that this way, the path line's accuracy does *not* depend on the path line's sampling density: even for a sparse sampling, the points on the path line are correct (up to the accuracy of the initial flow map sampling). This is contrary to a numerical integration of path lines where the integration's step size influences the integration error. Besides the exploration via path lines, we also implemented a visualization via FTLE fields. The FTLE is a widely used scalar value that characterizes the separation of neighboring particles over time. For further information about FTLE, we refer to the work of Haller et al. [7–9] and Shadden et al. [16]. The FTLE field as well can be computed by a straightforward lookup in the discretized flow map. Given t and τ, we determine the FTLE field at the same spatial resolution as the sampled flow map. For this, we compute the flow map derivatives by central differences. Fig. 4 shows a snapshot of our interactive viewer. The values t and τ are modified either by sliders or by interactively moving around in the (t, τ)-space. According to this, we update the path lines and the FTLE fields in interactive real-time. We implemented a toolset based on the concepts presented in this paper for interactive flow map modification.

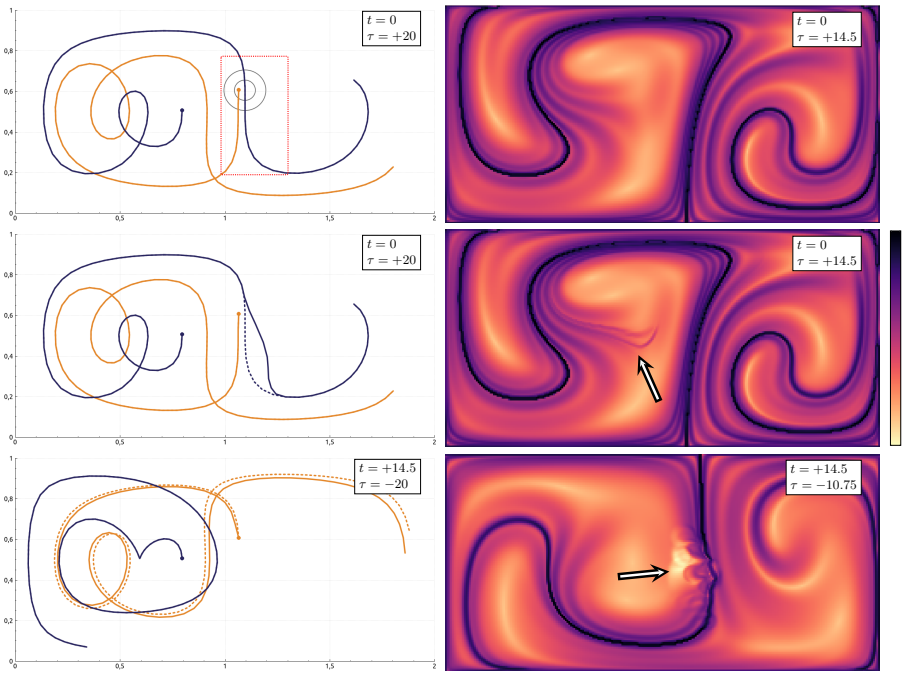

Fig. 5. Path lines and FTLE fields after modification with the translation tool.

6 Results and Discussion

We show the results as plots of selected path lines and FTLE fields. Timings were taken on an Intel Core i7-8700K CPU running at 3.70GHz with 32GB of RAM. We executed all algorithms on a single core. The data set we use is the Double Gyre, a synthetic periodic 2D unsteady vector field introduced by Shadden et al. [16]. The Double Gyre is one of the most used benchmarks in Flow Visualization. Fig. 4 shows path lines and the FTLE field for $t = 2.5$ and $\tau = 15$ for the Double Gyre. To obtain the discretized flow map, we sampled the domain with a resolution of 256×128 for $(x, y) \in [0.0, 2.0] \times [0.0, 1.0]$. We discretized the temporal dimension $t \in [0.0, 20.0]$ by 80 samples and $\tau \in [-20.0, 20.0]$ by 160 samples. This sums up to approximately $4.2 \cdot 10^8$ samples, leading to a memory usage of 6.5GB for the discretized flow map. In a pre-computation step, we perform a second-order Runge-Kutta integration for each sample; this took about 70 *minutes*. In another pre-computation step, we compute the lookup-cells; this takes approx. 30 *seconds* and costs additional 1.6GB of RAM. We show results for two different interactive modification tools.

Translation Tool: The first one performs a simple translation of flow map entries and is presented in Fig. 5. Top row shows path lines and FTLE field before modification. Top left also shows the modification tool with its inner and

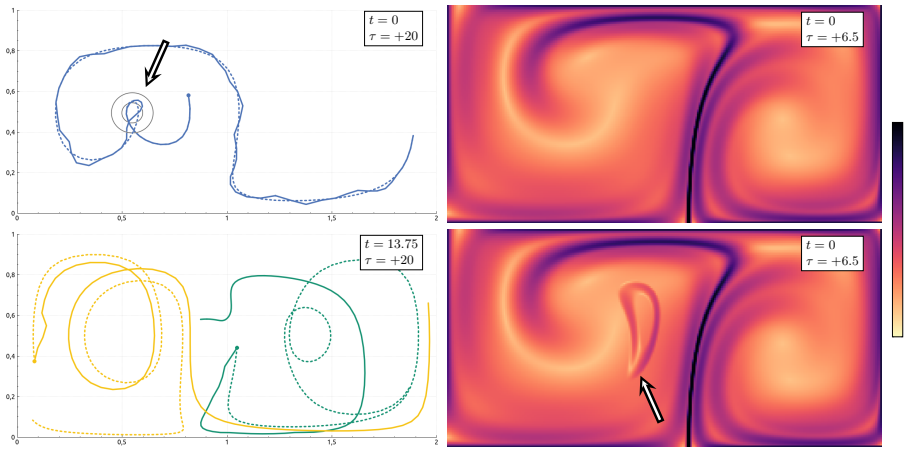

Fig. 6. Path lines and FTLE fields after modification with the rotation tool.

outer region by two black circles and the modified domain parts' spatial extent by a red rectangle. The modification was manually performed in 56 repeated small translation steps in the area of $1.0 \leq x \leq 1.3$, $0.2 \leq y \leq 0.8$, $14.25 \leq t+\tau \leq 15.5$. In each step, about $5.2 \cdot 10^5$ entries were adapted, which took 0.13 *seconds* in average. After the process, $5.9 \cdot 10^6$ flow map entries were modified in total. The overall interactive modification took about 1.5 *minutes*. Fig. 5 middle left shows a blue path line, which passes the modified area and an orange one, that is not influenced. Dashed lines indicate the original trajectories before modification. Middle right shows an FTLE field that differs from the original one. Path lines starting in the region marked with a white arrow at $t = 0$ will pass the modified area after $\tau = 14.5$. The global influence of the modification is even more evident in the images in the bottom row. Path lines starting in the modified area, with respect to space-time, take a different path over the whole integration time. This effect is also visible in the FTLE field. Please note, we decided to show only two carefully selected path lines to avoid visual clutter. The selected ones and the shown (t, τ)-combinations give a good impression of the global influence of (pseudo-)local modifications.

Rotation Tool: The second tool performs a clockwise rotation of entries around the center of the modification area. Fig. 6 shows the modification performed on consecutive locations along the whole trajectory in space-time for a fixed $t = 0$ of the blue path line shown top left. We modified the flow map in the vicinity of the whole blue path line. This can also be seen with the green and yellow path lines shown bottom left, which start next to the blue line in space-time and take rather different paths. The arrow marks the modification tool, which is currently placed at time $t + \tau = 6.5$. The FTLE fields in the right column show the effect of modification at this specific location. All points starting in the area marked with a white arrow with $t = 0$ and $tau = 6.5$ map into the modified

region. The modification was manually performed in 66 consecutive steps at different locations in space-time. On average, in each step $3.3 \cdot 10^6$ entries were adapted, which took an average of 0.71 *seconds*. The time needed for adaption is noticeable but still sufficient for an interactive process. The reason for the higher processing time is the larger area that is covered by the rotation tool. After the process, $3.9 \cdot 10^7$ flow map entries were modified in total. This means almost 10% of all samples had to be adapted, which clearly shows the high connectivity and complexity of the underlying flow map. The overall interactive modification with the rotation tool took about 3.0 *minutes*.

7 Conclusion and Future Work

Flow maps are a highly complex but powerful tool for flow representation. We expect flow map processing to be of more interest in future research. This will lead to more knowledge, a better understanding, new algorithms, and new applications considering the flow map. With this work, we want to go a step further in this direction. Flow map processing opens up new possibilities for efficient and effective Flow Visualization. In Sec. 3, we gave a compact overview of flow map properties and their relations to vector fields. In Sec. 4, we introduced an approach for flow map modifications based on space deformation. Although we stick to the 1D and 2D cases to explain, all presented concepts are valid for any dimension. We presented an implementation in Sec. 5 and 6. Future research could be related to more and better (interactive) tools for flow map processing, e.g., for smoothing, deformation, construction or simplification. Furthermore, better techniques to visualize the global behavior of flow maps in space-double-time are necessary. In the presented results, we modify the flow map of a 2D unsteady flow. Effectively this modification takes place in a highly connected 4D domain (x, y, t, τ). Due to the data's high connectivity, this can lead to unexpected results in areas currently not visible. This will even be a bigger problem for 3D flows and needs to be addressed in future research.

Acknowledgement:
This work was partially supported by DFG grant TH 692/17-1

Appendix

Lemma. *If ϕ is flow map, then $\widetilde{\phi}$ defined by Eq. 9 is a flow map as well.*

Proof. To show that $\widetilde{\phi}$ is a glow map, we need to show the identity (Eq. 4) and additivity (Eq. 5) of $\widetilde{\phi}$:

$$\widetilde{\phi}(\mathbf{x}, t, 0) = \mathbf{y}(\phi(\mathbf{y}^{-1}(\mathbf{x}, t), t, 0), t) = \mathbf{y}(\mathbf{y}^{-1}(\mathbf{x}, t), t) = \mathbf{x}$$

$$\widetilde{\phi}(\widetilde{\phi}(\mathbf{x}, t, \tau_1), t + \tau_1, \tau_2)$$

$$= \mathbf{y}(\phi(\mathbf{y}^{-1}(\mathbf{y}(\phi(\mathbf{y}^{-1}(\mathbf{x}, t), t, \tau_1), t + \tau_1), t + \tau_1), t + \tau_1, \tau_2), t + \tau_1 + \tau_2)$$

$$= \mathbf{y}(\phi(\mathbf{y}^{-1}(\mathbf{x}, t), t, \tau_1 + \tau_2), t + \tau_1 + \tau_2) = \widetilde{\phi}(\mathbf{x}, t, \tau_1 + \tau_2)$$

□

References

1. Baeza, I., Günther, T.: Vector field topology of time-dependent flows in a steady reference frame. IEEE Transactions on Visualization and Computer Graphics (2019)
2. Botsch, M., Kobbelt, L.: An intuitive framework for real-time freeform modeling. ACM Trans. Graph. (2004)
3. Bujack, R., Middel, A.: State of the art in flow visualization in the environmental sciences. Environmental Earth Sciences **79**(2) (2020)
4. Chen, G., Kwatra, V., Wei, L., Hansen, C.D., Zhang, E.: Design of 2d time-varying vector fields. IEEE Transactions on Visualization and Computer Graphics (2012)
5. Funck, W.v., Theisel, H., Seidel, H.P.: Vector field based shape deformations. Transactions on Graphics (Proc. ACM SIGGRAPH) **25**(3) (2006)
6. Funck, W.v., Theisel, H., Seidel, H.P.: Explicit control of vector field based shape deformations. In: Proc. Pacific Graphics 2007 (2007)
7. Haller, G., Yuan, G.: Lagrangian coherent structures and mixing in 2d turbulence. Physica D **147**(3-4) (2000)
8. Haller, G.: Distinguished material surfaces and coherent structures in three-dimensional fluid flows. Physica D **149**, 248–277 (2001)
9. Haller, G.: Lagrangian Coherent Structures. An. Review of Fluid Mechanics (2015)
10. Hlawatsch, M., Sadlo, F., Weiskopf, D.: Hierarchical Line Integration. IEEE Transactions on Visualization and Computer Graphics **17**(8) (2011)
11. Hofmann, L., Sadlo, F.: The dependent vectors operator. CG Forum (2019)
12. Hummel, M., Bujack, R., Joy, K.I., Garth, C.: Error estimates for lagrangian flow field representations. In: 18th Eurographics Conference on Visualization (2016)
13. Laramee, R.S., Hauser, H., Doleisch, H., Vrolijk, B., Post, F.H., Weiskopf, D.: The State of the Art in Flow Visualization: Dense and Texture-Based Techniques. Computer Graphics Forum (2004)
14. 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 (2007)
15. Salzbrunn, T., Wischgoll, T., Jnicke, H., Scheuermann, G.: The state of the art in flow visualization: Partition-based techniques. In: In Simulation and Visualization 2008 Proceedings (2008)
16. Shadden, S.C., Lekien, F., Marsden, J.E.: Definition and properties of lcs from ftle in 2d aperiodic flows. Physica D (2005)
17. Theisel, H.: Designing 2d vector fields of arbitrary topology. Computer Graphics Forum (Proc. Eurographics 2002) **21** (2002)
18. Vaxman, A., Campen, M., Diamanti, O., Panozzo, D., Bommes, D., Hildebrandt, K., Ben-Chen, M.: Directional Field Synthesis, Design, and Processing. Computer Graphics Forum (2016)
19. Weinkauf, T., Theisel, H., Hege, H.C., Seidel, H.P.: Topological construction and visualization of higher order 3d vector fields. Computer Graphics Forum (2004)
20. Wilde, T., Rössl, C., Theisel, H.: Recirculation surfaces for flow visualization. IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE SciVis) (2018)
21. Zhang, E., Hays, J., Turk, G.: Interactive tensor field design and visualization on surfaces. IEEE transactions on visualization and computer graphics (2007)
22. Zhang, E., Mischaikow, K., Turk, G.: Vector field design on surfaces. ACM Trans. Graph. **25**(4) (2006)