FTLE Ridge Lines for Long Integration Times

Thomas Wilde, Christian Rössl and Holger Theisel

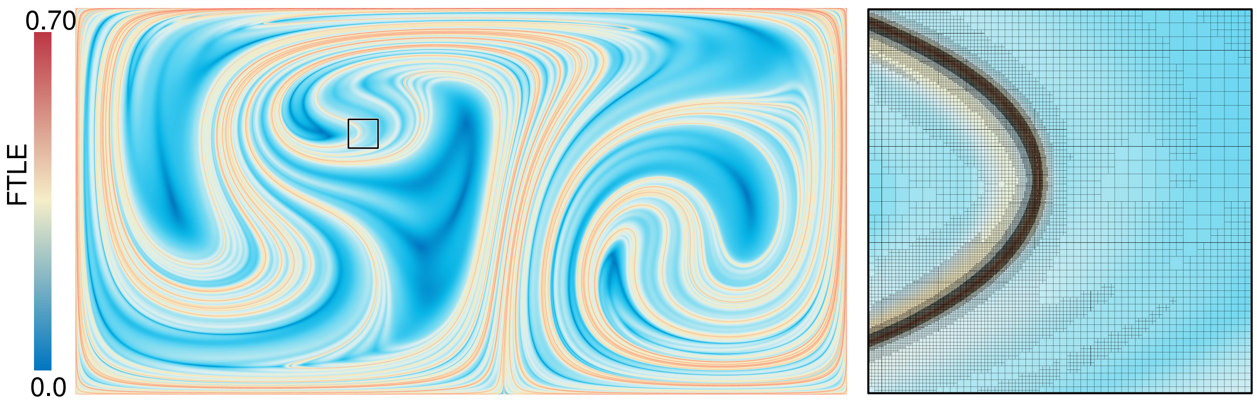


Fig. 1: FTLE for the DOUBLE GYRE for integration time $\tau = 30$. A high number of clearly distinct ridges is visible in the overview (left). The resolution of the underlying sampling grid is dense at positions where ridges are located (detail). The adaptive refinement is steered by information from the whole interval of integration times, to track the development of sharp ridges.

Abstract—We present an approach to the extraction of FTLE ridges for 2D unsteady vector fields under long integration times. This is a hard problem because such FTLE ridges tend to be sharp and close to each other. The main feature of our approach is that it does not only use an FTLE sampling at the desired final integration time but incorporates samples from prior integration times as well. With this additional information, the new method produces more and finer ridge lines than previous approaches. Based on this output, we can consider FTLE ridge statistics. We test the approach on synthetic benchmarks and real-world data sets.

Index Terms—FTLE, Unsteady Vector Field.

1 INTRODUCTION

Lagrangian Coherent Structures (LCS) are prominent and promising approaches for extracting and visualizing the global behavior of time-dependent flow fields. LCS are commonly considered as extremal structures – ridges – of the derivatives of the flow map. Among the various alternatives that have been proposed in recent years, ridges in Finite-Time Lyapunov Exponents (FTLE) are one of the most common representatives of LCS. Such ridges in FTLE fields are structures that separate regions with different flow behavior. In this paper, we present an approach to extracting ridge lines of 2D FTLE fields with a focus on long integration times. At first glance, this seems to be a standard problem: The FTLE field for the desired integration time is a scalar field for which a suitable adaptive sampling has to be computed, and then standard numerical ridge extraction could be applied. It turns out, however, that the problem is much harder for the following reasons: Firstly, FTLE computation is expensive since every sample of the FTLE field requires numerical integration. Secondly, FTLE ridges tend to be thin and sharp for long integration times. Thirdly, adjacent ridges tend to be close to each other for long integration times. In fact, we will argue that a sufficient adaptive sampling of the FTLE field for a fixed, long integration time cannot be computed with reasonable effort. This is the reason why existing approaches either restrict themselves to considering only the FTLE field – without any extraction of the ridge geometry – or to extracting only simple non-sharp ridges for short integration times. The main idea to solve the sampling problem is based on the insight that an adaptive sampling of the FTLE field should not only incorporate samples at the desired integration time but also at intermediate times. This is because sharp FTLE ridges – that are hard to extract – have been non-sharp – that are easier to extract – at shorter integration times. Based on this key insight, we construct an algorithm for adaptive FTLE sampling and FTLE ridge extraction. The algorithm is based on a quadtree subdivision of the domain where the subdivision criterion evaluates FTLE in the whole range of integration times from zero to the desired time.

- *Thomas Wilde, Visual Computing group at the University of Magdeburg E-mail: thomas@isg.cs.uni-magdeburg.de.*
- *Christian Rössl, Visual Computing group at the University of Magdeburg E-mail: roessl@isg.cs.uni-magdeburg.de.*
- *Holger Theisel, Visual Computing group at the University of Magdeburg E-mail: theisel@isg.cs.uni-magdeburg.de.*

2 RELATED WORK

This section gives a brief overview of previous work on ridges in general and in the field of flow visualization as well as FTLE and ridges in FTLE.

Ridge Concepts: There are a variety of different ridge definitions along with numerical extraction techniques in Scientific Visualization. In particular, ridges serve as a standard extractor for edges in images. We refer to the discussion presented by Eberly et al. [2] for an overview on the most common ridge definitions such as height ridges [15], curvature ridges [22], watershed ridges [8], or medial axes [24]. A discussion about the quality of different ridge extractors is given in [1] and [17]. In Visualization, ridges are known to show relevant structures in many applications and a number of ridge extraction algorithms have been proposed, e.g. for vortex core lines [21, 23, 30], characterstic lines in symmetric second-order tensor fields [35], watershed ridges [31] or ridge surfaces in DT-MRI data [16].

FTLE and FTLE ridges: One of the most prominent approaches to find LCS is the computation of ridge structures in scalar (FTLE) fields, was introduced by Haller [10, 14], see also [12] for an introduction to LCS, their meaning for describing flow dynamics and their extraction via FTLE. FTLE ridges have been used for a variety of applications [11, 19, 34, 36]. Shadden et al. [33] have shown that ridges of FTLE are approximate material structures, i.e., they converge to material structures for increasing integration times. This fact was used in [29] to extract topology-like structures. [20] and [32] introduce methods for tracking FTLE ridges by locally sampling the FTLE field and estimating the ridge direction and location. Due to the discrete sampling used, the accuracy is limited, especially for very sharp ridges. In [3] the minor eigenvector of the Cauchy-Green tensor is integrated to track ridge structures. This approach is, however, prone to accumulating integration errors. Also in the visualization community, different approaches have been proposed to increase performance, accuracy and usefulness of FTLE as a visualization tool [6, 7, 9, 25–28]. Haller and Sapsis [13] additionally explore the smallest FTLE values.

3 Background and Notation

The following notation is used throughout the paper. Given is a continuous unsteady vector field $\mathbf{v}(\mathbf{x},t)$. Its *flow map* $\phi(\mathbf{x},t,\tau)$ maps a particle seeded at $(\mathbf{x},t)$ to its destination after a path line integration of $\mathbf{v}$ over a time interval $[t,t+\tau]$. We assume $\tau > 0$. We write the flow map equivalently as $\phi_\tau = \phi_\tau(\mathbf{x}) = \phi(\mathbf{x},t,\tau)$ whenever position $\mathbf{x}$ and/or starting time t are clear from the context, and we do this similarly for other functions of the flow map. The (spatial) *flow map gradient* $\nabla\phi(\mathbf{x},t,\tau) = \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}}\phi(\mathbf{x},t,\tau)$ encodes the separation of particles seeded near $(\mathbf{x},t)$. Then with $\nabla = \nabla\phi(\mathbf{x},t,\tau)$, $\Delta = \Delta(\mathbf{x},t,\tau) = (\nabla^{\mathrm{T}}\nabla)(\mathbf{x},t,\tau)$ denotes the (right) *Cauchy-Green tensor* of the flow map. As Δ is symmetric and positive definite, its *eigenvalues* $\lambda_i > 0$ are real, and the associated *eigenvectors* are orthogonal. Then the FTLE field is defined as

$$\mathrm{FTLE}(\mathbf{x},t,\tau) = \tfrac{1}{|\tau|}\ln\sqrt{\lambda_{\max}(\Delta)}\ ,$$

where $\lambda_{\max}$ denotes the maximal eigenvalue of Δ. The standard approach to the numerical evaluation of $\mathrm{FTLE}(\mathbf{x},t,\tau)$ uses a discrete approximation of the flow map gradient $\nabla\phi(\mathbf{x},t,\tau)$ from finite differences of ϕ. When using central differences, this requires four evaluations of the flow map, i.e., four times a numerical integration of $\mathbf{v}$. Any discrete difference scheme requires size parameter h, which here denotes the spatial distance of two initial points before advection on the flow. For ridge extraction, in particular, smaller h – or observing a smaller region that is advected – will decrease the "probability" that a ridge is passing this region and is detected due to a high separation and a local maximum of FTLE. In order to find ridges, FTLE has to be evaluated. Throughout the paper we use the term *sampling* for any set of FTLE samples that supports the extraction of ridges. Samples are parametrized by $(\mathbf{x},t,\tau)$ without assumption of a particular distribution (e.g., a uniform grid). Based on the above considerations, we can formulate the problem as follows: Given a fixed starting time t and a fixed, positive – long – integration time τ_e, find a sampling of a scalar field $\mathrm{FTLE}(\mathbf{x},t,\tau_e)$ that is sufficient for extracting all ridges within a maximum resolution that is prescribed by numerical methods. We call this the *sufficient sampling problem.*

4 Problem Analysis

The solution of the sufficient sampling problem is the main challenge in FTLE ridge extraction. Given a desired final integration time τ_e and a fixed starting time t, a sampling of $\mathrm{FTLE}(\mathbf{x},t,\tau_e)$ in the spatial domain is searched, which sufficiently supports the extraction of all ridges. We analyze this problem in the following.

Why is it important to extract FTLE ridges at all? Explicit ridge geometry gives more information than the scalar FTLE field alone. The relation between an FTLE field and its ridges is similar as between a scalar field and, e.g., isosurface geometry. In particular, the extraction of FTLE ridge geometry allows considering ridge statistics.

Why is it important to separate two FTLE ridges that are close to each other? Adjacent FTLE ridges generally correspond to separating events that occur at different integration times. Imagine a group of spatially close particles traveling with the flow. If after a certain time a separation takes place, the particles divide into two subgroups each showing different further flow behavior. After an even longer integration time, each subgroup could be separated again into new subgroups.

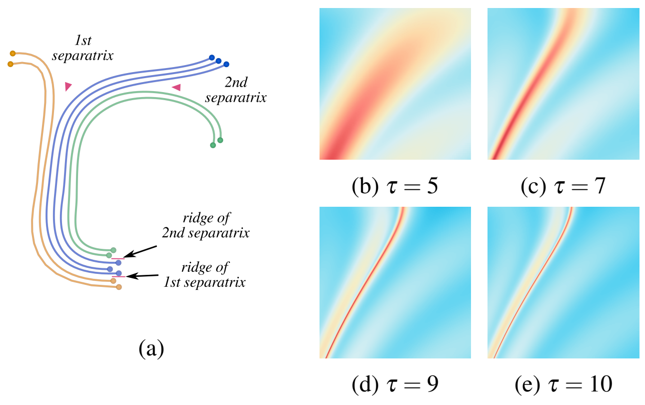


Fig. 2: (a) Spatially close FTLE ridges emerge from separation at different integration times. (b)–(e) Time series of emerging ridges.

This results in spatially close FTLE ridges around the initial locations of the particles. Figure 2(a) gives an illustration.

Why are ridge statistics of interest? FTLE is usually applied to non-turbulent flows. The transition to turbulent flows makes ridges behave "wild", i.e., they become sharp and dense. For these kind of flows, the effect of a single ridge could be hard to interpret. Instead, insights can be gained from statements on the set of all ridges. Therefore we introduce ridge statistics.

Properties of FTLE ridges and sampling density: For linearly increasing the maximal integration time τ_e ridges in FTLE fields tend to show the following two properties:

- **Property 1 (P1):** The sharpness of an FTLE ridge increases at an exponential rate.
- **Property 2 (P2):** The distance to the (spatial) next ridge decreases at an exponential rate.

Both properties were observed and stated similarly by Kuhn et al. [18]. Figure 2(b)–(e) illustrate this behavior by example. Shown is a part of the well-known Double Gyre data set. The strong center ridge appears "early" after few discrete time steps. Shortly after, two smaller ridges to the left and right emerge. With increasing τ they move towards the center ridge. Finding a sufficient sampling for a *fixed* τ_e is a challenging task. Ridges in the FTLE field can only be mapped, if the sampling is dense enough to capture them, otherwise they could be missed completely. Due to **P1** for a large τ_e an extremely dense regular sampling would be necessary to capture a sharp ridge. Because of the high computation cost this is not preferable. A possible solution could be to start with a coarse sampling and adaptively refine it by the following local criteria:

(a) Subdivide a region because a ridge is expected to pass through it.

(b) Stop further subdivision of a region containing a ridge because it is sure that only one ridge is passing through.

A criterion for (a) is crucial because if a ridge falls between two inital samples, it may be completely missed. This problem is central to any sampling. One common way to circumvent this is to increase the size parameter h when estimating the flow map gradient (see section 3). However this will generally not help, to distinguish between detection of a single or multiple ridges. We are not aware of any existing criteria for (b). We believe that such criterion does not exist because of **P2**. This is why we believe that it is *impossible* to get a sufficient sampling by considering FTLE values for the final integration time τ_e only.

Main idea: As a solution to this problem, we propose to steer adaptive subdivision by also incorporating FTLE sampling for shorter integration times. This is justified by the following observations: a ridge that is very sharp and close to its neighbors at the final integration time τ_e was "better behaved" at a shorter integration time $\tau < \tau_e$. This means, ridges emerge (slowly) over time. In fact, **P1** and **P2** show that the

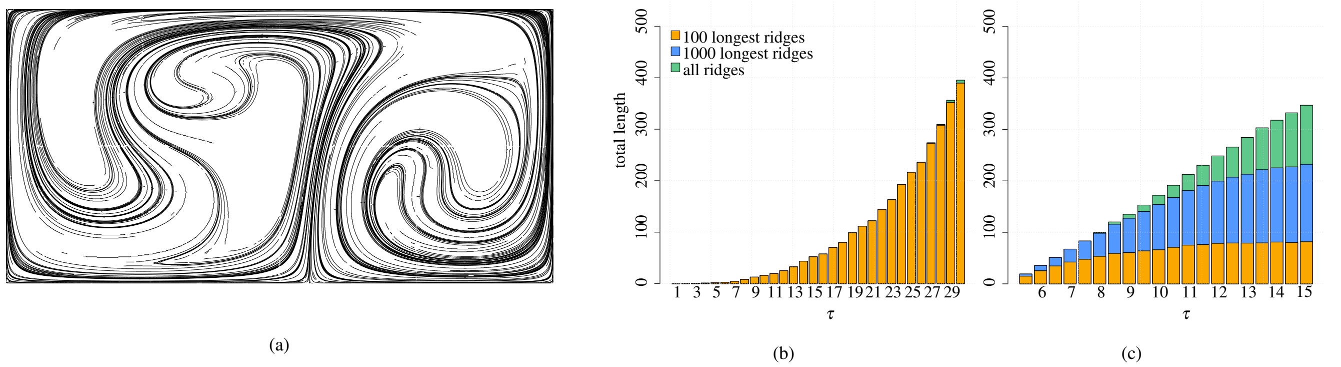


Fig. 3: Ridge geometry as vector graphic for the DOUBLE GYRE for $t = 0$ and $\tau = 30$ (a). Total ridge length over integration time of the 100 longest, 1000 longest and all ridges for the DOUBLE GYRE (b) and the BOUSSINESQ (c).

ridges can stably be extracted for a shorter integration time. Increasing the integration time from τ to τ_e makes a further subdivision necessary due to the increasing sharpness as well as due to a spatial motion of the ridge. However, ridge motion decreases with ongoing integration time which corresponds to the fact that FTLE ridges converge to material separation structures for increasing integration time (see [33]). This way we can make sure that the ridges of $\mathrm{FTLE}(\mathbf{x}, t, \tau_e)$ are resolved.

5 Finding a Sufficient Sampling and Extract Ridges

Algorithm specification: The *input* is a flow field $\mathbf{v}(\mathbf{x},t)$ in a domain $\mathscr{D}$ and an integration time τ_e. The period τ_e is high such that we expect sharp ridges, which are close to each other. It is obvious that two ridges cannot get arbitrarily close to each other and still be robustly separated: at some point numerical errors, e.g., from interpolation and integration, or floating point precision will put a practical limit on computing ridge locations precisely enough to distinguish the two different ridges. For this reason, the algorithm takes a final input parameter, the maximum resolution ε, which limits the minimal spatial distance between two samples. We assume that there will be no gain of information, if samples get closer than ε. The *output* of the algorithm is a set of samples $(\mathbf{x}, t, \tau)$ that supports the extraction of ridges within the maximum resolution ε. For each sample, $\tau \leq \tau_e$ denotes the maximum integration time until either the desired τ_e was reached ($\tau = \tau_e$) or the maximum spatial resolution ε was reached ($\tau < \tau_e$) and would have been exceeded at τ_e. In the first case, the sampling locally supports the extraction of ridge geometry. In the second case, a reliable ridge extraction is possible only for the returned $\tau < \tau_e$.

Domain discretization and initialization: We discretize the domain by a coarse grid of square cells, where each grid cell is refined by a quadtree subdivision. The algorithm then finds the minimal amount of subdivisions for all quadtrees and yields samples defined by leaf cells of quadtrees. We decided to use a quadtree [4] for adaptive refinement of the domain, because it is a fairly simple data structure with mature and efficient operations for traversal, finding neighbors, etc. (see, e.g., [5]). The regular structure of the quadtree fits our needs best.

Refinement of the sampling: We associate with each quadtree cell a sample. The spatial position $\mathbf{x}$ is the cell's center. Assuming a fixed start time t, we store with each cell the current integration time τ and $\mathrm{FTLE}(\mathbf{x}, t, \tau)$, which is computed from finite differences of ϕ_τ evaluated at the four cell corners. The refinement iteratively increases the integration time τ by a step $\Delta\tau$. The *step size* $\Delta\tau$ determines the sampling density of the integration time τ up to the target time τ_e. Note: $\Delta\tau$ is not the step size that is used for the numerical integration of the flow map ϕ. The flow map is sampled at each time step for computing finite differences. In each step of the iteration the following two operations are performed:

1. Update FTLE_τ and τ for all *leaf* cells.
2. Mark all leaf cells that must be split, then apply the splits.

Here, *leaf cells* refers to all leaf nodes of *all* quadtrees. The first step is straightforward. The second step eventually splits a quadtree at its center and refines the sampling locally, if one of the following conditions is violated for this cell and any of its quadtree neighbors:

(a) The difference of their quad tree depth must not exceed 1.

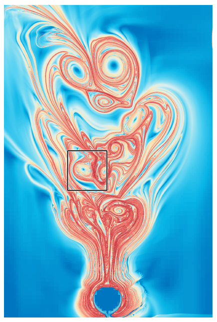
(b) The difference of their FTLE_τ must not exceed a threshold θ.

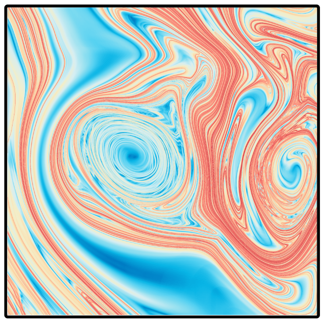
Condition (a) accounts for a regular and "smooth" refinement of the quadtree. This way, information from comparison of FTLE_τ values is reliable for neighboring cells. Condition (b) is the essential one, and it is motivated as follows. Ridges gain sharpness over integration time. They are "thicker" and "lower" for smaller τ and become "thinner" and "higher" with increasing τ. Whenever we observe an FTLE_τ value that gets locally maximal ("higher") for some τ, the cell (or one of its neighbors) is a candidate for supporting a ridge. While this reads trivial, the only reason why this refinement rule can work reliably is that the algorithm tracks ridges and their development over time. At observation time, the ridge is not yet too sharp, it may become sharper in the future. The refinement takes care of this and adapts the resolution locally – and exponentially – such that the ridge can be tracked further and such that the event that new, close ridges develop can be detected. The maximum resolution ε may stop further refinement locally. The splitting process eventually is influenced by two parameters: $\Delta\tau$ and θ. We recommend a choice in the order $\Delta\tau = 1\%\tau_e$. The output of the algorithm is not sensitive to this choice unless a much larger time step is used, in this case split events may be missed. In contrast, the algorithm *is* sensitive to the choice of the *threshold* θ, which steers the local refinement. If this threshold is too large, there is not enough adaptation and the generated sampling is not sufficient as ridges may be missed. If it is too low, the refinement will be too aggressive. We recommend a value of 1% of the maximum FTLE ($FTLE_{max}$) value at τ_e that is estimated from a coarse sampling.

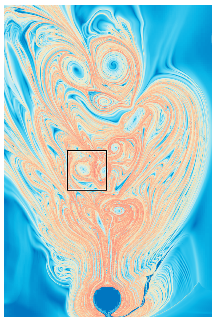
Ridge Extraction: The adaptive quadtree refinement provides a sufficient sampling for the extraction of ridges that can be reliably observed within the maximal resolution. A three-step process of filtering, clustering and post-processing is used for extracting ridges from quadtree cells.

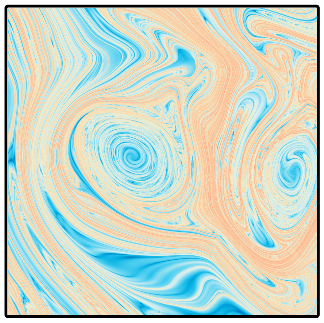
1. Find ridge candidates: a leaf node of the quadtree is considered a candidate if its FTLE value is a local maximum in co-gradient direction and one of its neighbours in gradient direction has the same property. This rule rejects isolated candidates and filters out those which clearly support a ridge.
2. Cluster ridge candidates: neighbouring candidates are partitioned into groups of connected "chains" such that each group supports a ridge.
3. Connect ridge clusters: consecutive clusters are connected, if they are separated by a single cell. Effectively, this stage closes small gaps that may arise from step 1.

Finally, ridge lines are extracted from cells. We do not aim at finding the "exact" ridge locations but instead output a polyline that has the ridge cells as vertices. This gives lines that are suitable for ridge statistics and avoids pretending exactness by a, e.g., marching cubes like, interpolation or a potentially error-prone optimization near the maximum resolution.


(a) $t = 5\ \tau = 10$


(b) $t = 5\ \tau = 10$


(c) $t = 5\ \tau = 15$


(d) $t = 5\ \tau = 15$

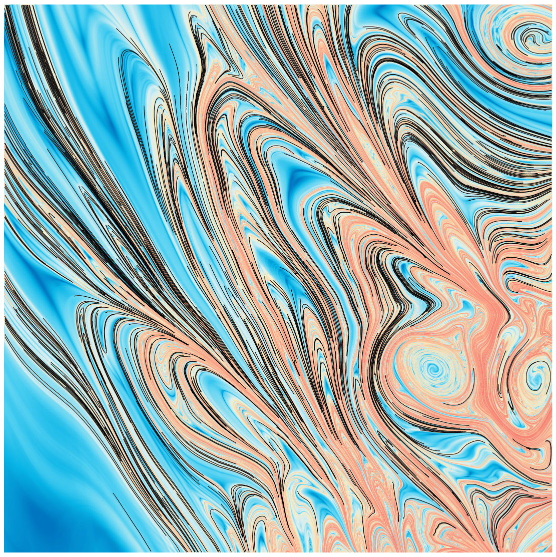

(e) $t = 5\ \tau = 15$

Fig. 4: BOUSSINESQ flow: FTLE field for different integration times with close-up views (a)–(d). Close-up with overlaid ridge geometry of the 1000 longest ridges (e)

6 Results

We show the results of two challenging data sets we used to test our algorithm: the DOUBLE GYRE is a synthetic data set, which can be evaluated at arbitrary resolution. The BOUSSINESQ data set stems from simulation and is given as discrete samples on uniform grid.

Double Gyre: The DOUBLE GYRE is a synthetic data set, which was introduced by Shadden et al. [33]. It represents a periodic 2D unsteady vector field

$$\mathbf{v}(x,y,t) = \begin{pmatrix} -\frac{\pi}{10}\sin(\pi f(x,t))\cos(\pi y) \\ \frac{\pi}{10}\cos(\pi f(x,t))\sin(\pi y)\,\frac{\mathrm{d}}{\mathrm{d}x}f(x,t) \end{pmatrix} \quad \text{with}$$

$$f(x,t) = a(t)x^2 + b(t)x\ ,\quad a(t) = \tfrac{1}{4}\sin\tfrac{\pi}{5}t\ ,\quad b(t) = 1 - \tfrac{1}{2}\sin\tfrac{\pi}{5}t\ .$$

Figure 1 shows the result of our algorithm for $\tau = 30$ and a close-up with the quadtree. Figure 3(a) shows the corresponding ridge geometry for $\tau = 30$ (as scalable vector graphics). Due to parallel computation the domain was split into eight parts, which leads to the visible missing ridge connections in the middle of the image. The time series in Figure 2(b)–(e) shows a close-up of a ridge that emerges over integration time in the middle of the domain. Figure 3(b) provides some ridge statistics on the DOUBLE GYRE: for given integration time τ, we sum the length of all, the 1000 longest and the 100 longest ridges. The DOUBLE GYRE was processed in the domain $[0,2]\times[0,1]$ and with $t = 0$, $\tau_e = 30$, $\Delta\tau = \frac{1}{100}\tau_e$, $\theta = \frac{1}{100}FTLE_{max}$.

Boussinesq: The BOUSSINESQ data set represents the natural convection generated by a heated cylinder: a stagnant fluid is heated and after short time develops highly turbulent plume above the cylinder. The data is given as a series of 1.600 time steps for $t \in [0,20]$ of 100×300 spatial grids for $[-0.5,0.5]\times[-0.5,2.5]$. The flow field is reconstructed by trilinear interpolation. Figure 4 shows FTLE for $\tau = 10$ and $\tau = 15$ for the region $[-0.45,0.45]\times[-0.45,0.90]$ of the domain. Figure 4(e) shows a close-up view for $\tau = 15$ with overlaid ridge geometry. Please note that we only show the 1000 longest extracted ridges, meaning the ones that are most relevant for the description of the flow. Figure 3(c) show statistics for the accumulated ridge length with respect to the integration time τ. The BOUSSINESQ flow was processed with $t = 5$, $\tau_e = 15$, $\Delta\tau = \frac{1}{100}\tau_e$, $\theta = \frac{1}{100}FTLE_{max}$.

7 Discussion

For both data sets our approach was able to compute high amounts of ridge geometries. The DOUBLE GYRE and the BOUSSINESQ were considered in a comparable domain with side lengths in the order of 1-2 length units. For these two data sets we were able to extract 350 to 400 length units of ridge lines. To the best of our knowledge, no other approach comes even close to extracting such a rich ridge line geometry. This is mainly made possible by a sampling that is tuned to the particular problem: It is based on sampling of FTLE not only at the final integration time but at intermediate times as well.

Based on the extraction of the line geometry, ridge statistics for increasing integration times became possible. We do not know any previous publications that compute ridge statistics in this or a similar way. In our opinion the statistics have the potential to characterize the global behavior of FTLE ridges over time. Hence they could be a future tool for the transition from long integration times to even longer ones. Here the test data sets show a significantly different behavior. The DOUBLE GYRE exhibits an exponential behavior in the total amount of ridges. It has a few ridges, that become longer over time – rapidly but smoothly. This reflects the behavior of the underlying flow field, which changes constantly but smoothly over time. The length of the ridge lines grows approximately linearly for the BOUSSINESQ flow. The plot reveals that there are only a few long ridges (yellow bars), that do not become longer over time. For further integration times we expect the number of the short ridges (green bars) to increase. The BOUSSINESQ flow becomes turbulent after short integration times, which could explain stagnation of long ridges.

8 Limitations and Future Work

If the flow $\mathbf{v}$ is smooth and differentiable, FTLE is smooth and differentiable as well for a finite integration time. This means that there is only a finite number of ridges, and ridges have finite length. It would be desirable to give guarantees that our algorithm finds all of them. However, this cannot be guaranteed because the algorithm depends on numerical integration which introduces errors especially for the estimation of the gradient of the flow map. While these numerical artifacts occur to particular lines, they do not have an impact on the ridge statistics. For future research, an extension to 3D ridge surfaces is desirable. There is no conceptional reason that prohibits the extension: also in 3D, ridge surfaces become sharper and closer to each other with increasing integration times. To compute statistics the FTLE surfaces need to be extracted. We expect this to be the most challenging task for the step to 3D.

Acknowledgments: The work on this paper was partially supported by DFG grant TH 692/8-1.

References

[1] H. Blum and R. N. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10(3):167–180, 1978.

[2] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach. Ridges for image analysis. *Journal of Mathematical Imaging and Vision*, 4(4):353–373, 1994.

[3] M. Farazmand and G. Haller. Computing Lagrangian coherent structures from their variational theory. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(1):013128, 2012.

[4] R. A. Finkel and J. L. Bentley. Quad trees: A data structure for retrieval on composite keys. *Acta Inf.*, 4:1–9, 1974.

[5] S. F. Frisken and R. N. Perry. Simple and efficient traversal methods for quadtrees and octrees. *Journal of Graphics Tools*, 7:2002, 2002.

[6] C. Garth, F. Gerhardt, X. Tricoche, and H. Hagen. Efficient computation and visualization of coherent structures in fluid flow applications. *IEEE Transactions on Visualization and Computer Graphics*, 13(6), 2007.

[7] C. Garth, G. Li, X. Tricoche, C. Hansen, and H. Hagen. Visualization of coherent structures in transient 2d flows. In *Proceedings of TopoInVis*, 2007.

[8] J. Gauch. Image segmentation and analysis via multiscale gradient watershed hierarchies. *Image Processing, IEEE Transactions on*, 8(1):69 –79, 1999.

[9] T. Germer, M. Otto, R. Peikert, and H. Theisel. Lagrangian coherent structures with guaranteed material separation. *Computer Graphics Forum (Proc. EuroVis)*, 30(3):761–770, 2011.

[10] G. Haller. Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D*, 149:248–277, 2001.

[11] G. Haller. Lagrangian coherent structures from approximate velocity data. *Physics of Fluids*, 14(6):1851–1861, 2002.

[12] G. Haller. Lagrangian Coherent Structures. *Annual Review of Fluid Mechanics*, 47:137–62, 2015.

[13] G. Haller and T. Sapsis. Lagrangian coherent structures and the smallest finite-time Lyapunov exponent. *Chaos*, 21(2):023115, 2011.

[14] G. Haller and G. Yuan. Lagrangian coherent structures and mixing in two-dimensional turbulence. *Physica D*, 147(3-4), 2000.

[15] R. Haralick. Ridges and valleys on digital images. *Computer Vision, Graphics, and Image Processing*, 22:28–38, 1983.

[16] G. Kindlmann, X. Tricoche, and C.-F. Westin. Delineating white matter structure in diffusion tensor MRI with anisotropy creases. *Medical Image Analysis*, 11(5):492–502, October 2007.

[17] J. J. Koenderink and A. J. van Doorn. Local features of smooth shapes: ridges and courses. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 2031, pp. 2–13, June 1993.

[18] A. Kuhn, C. Rössl, T. Weinkauf, and H. Theisel. A benchmark for evaluating FTLE computations. In *Proc. IEEE Pacific Visualization Symposium*, PacificVis, pp. 121–128, 2012. g.

[19] F. Lekien, C. Coulliette, A. J. Mariano, E. H. Ryan, L. K. Shay, G. Haller, and J. Marsden. Pollution release tied to invariant manifolds: A case study for the coast of Florida. *Physica D: Nonlinear Phenomena*, 210(1-2):1–20, 2005.

[20] D. Lipinski and K. Mohseni. A ridge tracking algorithm and error estimate for efficient computation of Lagrangian coherent structures. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(1), 2010.

[21] H. Miura and S. Kida. Identification of Tubular Vortices in Turbulence. *Journal of the Physical Society of Japan*, 66:1331–1334, May 1997.

[22] S. Musuvathy, E. Cohen, J. Damon, and J.-K. Seong. Principal curvature ridges and geometrically salient regions of parametric b-spline surfaces. *Comput. Aided Des.*, 43:756–770, July 2011.

[23] R. Peikert and M. Roth. The parallel vectors operator - a vector field visualization primitive. In *Proc. IEEE Visualization*, pp. 263–270, 1999.

[24] S. M. Pizer, C. A. Burbeck, J. M. Coggins, D. S. Fritsch, and B. S. Morse. Object shape before boundary shape: Scale-space medial axes. *Journal of Mathematical Imaging and Vision*, 4(3):303–313, 1994.

[25] A. Pobitzer, R. Peikert, R. Fuchs, H. Theisel, and H. Hauser. Filtering of FTLE for visualizing spatial separation in unsteady 3d flow. In *Proc. TopoInVis*, 2011.

[26] F. Sadlo and R. Peikert. Efficient visualization of Lagrangian coherent structures by filtered AMR ridge extraction. *IEEE Transactions on Visualization and Computer Graphics (Proc. Visualization)*, 13(6), 2007.

[27] F. Sadlo and R. Peikert. Visualizing Lagrangian coherent structures and comparison to vector field topology. In *Proceedings of the 2007 Workshop on Topology-Based Method in Visualization (TopoInVis)*, 2007.

[28] F. Sadlo, A. Rigazzi, and R. Peikert. Time-Dependent Visualization of Lagrangian Coherent Structures by Grid Advection. In V. Pascucci, X. Tricoche, H. Hagen, and J. Tierny, eds., *Topological Methods in Data Analysis and Visualization*, Mathematics and Visualization, pp. 151–165. Springer, 2009.

[29] F. Sadlo and D. Weiskopf. Time-dependent 2d vector field topology: An approach inspired by Lagrangian coherent structures. *Computer Graphics Forum*, 29(1):88–100, 2010.

[30] J. Sahner, T. Weinkauf, and H.-C. Hege. Galilean invariant extraction and iconic representation of vortex core lines. In *Proc. Eurographics/IEEE VGTC Symposium on Visualization (EuroVis)*, pp. 151–160, 2005.

[31] J. Sahner, T. Weinkauf, N. Teuber, and H.-C. Hege. Vortex and strain skeletons in eulerian and Lagrangian frames. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):980–990, 2007.

[32] B. Schindler, R. Peikert, R. Fuchs, and H. Theisel. Ridge concepts for the visualization of Lagrangian Coherent Structures. In R. Peikert, H. Hauser, H. Carr, and R. Fuchs, eds., *Topological Methods in Data Analysis and Visualization II*, Mathematics and Visualization, pp. 221–235. Springer, 2012.

[33] S. C. Shadden, F. Lekien, and J. E. Marsden. Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. *Physica D*, 212(7), 2005.

[34] S. C. Shadden, F. Lekien, J. D. Paduan, F. P. Chavez, and J. E. Marsden. The correlation between surface drifters and coherent structures based on high-frequency radar data in Monterey Bay. *Deep Sea Research Part II: Topical Studies in Oceanography*, 56(3-5):161 – 172, 2009.

[35] X. Tricoche, G. Kindlmann, and C.-F. Westin. Invariant crease lines for topological and structural analysis of tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1627–1634, 2008.

[36] M. Weldon, T. Peacock, G. B. Jacobs, M. Helu, and G. Haller. Experimental and numerical investigation of the kinematic theory of unsteady separation. *Journal of Fluid Mechanics*, 611, 2008.