

# Scale-Space Approaches to FTLE Ridges

Raphael Fuchs, Benjamin Schindler and Ronald Peikert

**Abstract** The finite-time Lyapunov Exponent (FTLE) is useful for the visualization of time-dependent velocity fields. The ridges of this derived scalar field have been shown to correspond well to attracting or repelling material structures, so-called Lagrangian coherent structures (LCS). There are two issues involved in the computation of FTLE for this purpose. Firstly, it is often not practically possible to refine the grid for sampling the flow map until convergence of FTLE is reached. Slow convergence is mostly caused by gradient underestimation. Secondly, there is a parameter, the integration time, which has to be chosen sensibly. Both of these problems call for an examination in scale-space. We show that a scale-space approach solves the problem of gradient underestimation. We test optimal-scale ridges for their usefulness with FTLE fields, obtaining a negative result. However, we propose an optimization of the time parameter for a given scale of observation. Finally, an incremental method for computing smoothed flow maps is presented.

## 1 Introduction

The idea behind Lagrangian coherent structures (LCS) is to find material surfaces which separate regions of the fluid with different long term particle movement behavior. In other words, we want to find coherent structures based on their influence on material transport and detect transport barriers in the flow. Attracting LCS are structures on which nearby trajectories accumulate. Conversely, repelling LCS locally exhibit the highest rate of repulsion. The FTLE-based approach to LCS was introduced by the seminal work of Haller and Yuan [HY00]. Even though FTLE ridges are not always perfect indicators of the LCS [Hal02], they have been shown to work well in a large number of applications [PD10].

---

Raphael Fuchs, Benjamin Schindler, Ronald Peikert  
ETH Zurich, e-mail: `\{raphael,bschindler,peikert\}@inf.ethz.ch`

For dynamical systems where there is no limit on integration time and spatial resolution of the field it is possible to use very long integration times and very high sampling densities. For practical data and computational resources this can be impossible to do. Often there is only a finite time interval of the flow available, the spatial resolution of the discretization is limited and the sampling density has to be kept as large as possible to reduce computational effort. In this paper we show that scale-space theory [Lin94] offers answers to these problems.

Originally scale-space was developed for detecting features at different scales in digital images. Typical such features are edges and ridges. There are several benefits of extracting ridges in scale-space:

- When extracting features we have to use operators of finite size for gradient estimation and other operations. The scale-space approach inherently solves the question of which size should be selected.
- Selecting the scale where the ridges are most distinctive improves the quality of the result.
- The scale of detection contains valuable information about the feature. For example ridges detected only at very small scales of observation are often noise.

Following the related work section, Section 3 addresses the numerical computation of FTLE. The two standard ways are described in Section 3.1, while Section 3.2 explains our scale-space approach. Scale-space extensions to ridge extraction are presented in Section 4 and a faster method for computing smoothed flow maps is described in Section 5.

## 2 Related Work

There is a wide range of topological methods for flow visualization. For an overview we refer to state-of-the-art reports on this topic in general [LHZP07] and the special case of unsteady flow [PPF\*10].

Haller [Hal00, Hal01] proposed that a repelling LCS appears as a ridge of the forward-time FTLE field, whereas attracting LCS appear as ridges of the backward-time FTLE field. Shadden et al. [SLM05] define an LCS as a ridge of the FTLE field and derive an estimate for the material flux through FTLE ridges. Tang et al. [TCH10] suggest a method to extend a given finite velocity field to an infinite domain to solve the problem of trajectories leaving the domain. Recently, Haller [Hal10] presented examples of dynamical systems where stable and unstable manifolds cannot be extracted as ridges of the FTLE field and suggested a variational approach to LCS extraction.

The computational effort to compute the flow map and for the extraction of FTLE ridges is substantial. Therefore there are several suggestions on how to reduce the required amount of computation. Garth et al. [GGTH07] computed FTLE on planar cross-sections and used direct volume rendering instead of an explicit ridge extraction. In a later work [GWT\*08], FTLE computation is restricted to offset surfaces

at walls. Sadlo et al. [SP07] and Lekien and Ross [LR10] suggested an adaptive refinement strategy for ridge extraction that reduces the number of flow map integrations. Sadlo et al. [SRP10], Lipinski and Mohseni [LM10], and Brunton and Rowley [BR10] exploit temporal coherency for fast FTLE (ridge) computation for multiple time steps. Jimenez [JV09] presented a GPU-based FTLE computation for regular 2D grids.

Scale-space theory has been developed by the computer vision community and received much attention in the 1990s. For a general introduction we refer to the book and tutorial of Lindeberg [Lin94, Lin96]. The main concept is that an image can be extended to a family of smoothed images where the size of the smoothing kernel is parameterized by the scale parameter. Since the detection of features such as ridges is crucially dependent on the size of the gradient estimation stencil, the scale-space representation is a natural approach to select the optimal range for all image operations. Lindeberg [Lin98] discusses a ridge detection approach where an optimal scale is selected at each point based on a strength criterion for a ridge. Florack and Kuiper [FK00] propose to analyze the deep structure of an image, that is, to consider all levels of smoothing simultaneously. Scale-space methods are to be distinguished from multi-resolution methods, where data are represented at lower resolutions, e.g. by applying a wavelet transformation, and where typical applications are data compression and progressive visualizations.

In the visualization community there are several works exploring the applicability of scale-space theory. Bauer and Peikert [BP02] present a technique to extract vortex core lines in scale-space. Klein and Ertl [KE07] track vector field critical points in scale-space. Kinsner et al. [KCS08] present a GPU implementation for 2D ridge detection. Recently, Kindlmann et al. [KSSW09] proposed a scale-space approach to extract crease surfaces in tensor data. An interesting extension to the approach of Lindeberg is that their approach tries to maintain the spatial continuity of the scale.

### 3 Numerical computation of FTLE

In this section we describe two numerical methods to compute the FTLE field.

#### 3.1 Methods based on discretized flow map and on renormalization

In a given time-dependent velocity field  $\mathbf{u}(\mathbf{x}, t)$  the *trajectory* (pathline) seeded at  $(\mathbf{x}_0, t_0)$  is the solution  $\phi_{\mathbf{x}_0, t_0}(t)$  of the initial value problem

$$\begin{aligned} \frac{\partial}{\partial t} \phi_{\mathbf{x}_0, t_0}(t) &= \mathbf{u}(\phi_{\mathbf{x}_0, t_0}(t), t) \\ \phi_{\mathbf{x}_0, t_0}(t_0) &= \mathbf{x}_0 \end{aligned} \tag{1}$$

By keeping  $t_0$  and  $t$  fixed, but varying  $\mathbf{x}_0$ , we obtain the *flow map*  $\phi_{t_0}^t(\mathbf{x}_0)$ . Its gradient  $\mathbf{F} = \nabla \phi_{t_0}^t(\mathbf{x}_0)$  leads to the right Cauchy-Green deformation tensor  $\mathbf{C} = \mathbf{F}^T \mathbf{F}$  and to the FTLE

$$\text{FTLE}(\mathbf{x}_0) = \frac{1}{t - t_0} \ln \sqrt{\lambda_{\max} \mathbf{C}} \quad (2)$$

Computation of the FTLE field can be done in two ways. In the *flow map method* [Hal01], the flow map is computed on all nodes of a sampling grid, and then the FTLE field is computed using finite differences for the gradients. The *renormalization method* [BGGS80] also starts with a finite-difference stencil (given point plus two neighbor points per dimension). The trajectories of the central point and the neighbor points are computed simultaneously, and after each integration step the neighbor trajectory is renormalized, i.e., the distance of the neighbor trajectory and the central trajectory is restored by moving the current point on the straight line connecting the central trajectory and the neighbor trajectory (see Fig. 1). In other words a multiplier is applied to the distance between the central (fiducial) trajectory and the neighbor trajectory. After integrating over the full time interval  $[t_0, t_0 + T]$ , the product of these multipliers is computed per trajectory, and the inverse of it is applied to its end point. The difference between the two approaches is that renormalization improves the accuracy of the FTLE at the given point, while the first yields a more representative value for the grid cell represented by the point. In a recent comparison, Kasten et al. [KPH\*09] used the terms L-FTLE and F-FTLE for FTLE computed with and without renormalization.

By definition, FTLE converge with increasing integration time  $T$  to Lyapunov exponents (which exist under the conditions of the Oseledec theorem). This can be used to approximate Lyapunov exponents by computing FTLE with sufficiently large  $T$ . However, for this purpose it is crucial to use a method based on renormalization rather than a sampled flow map. The latter approach is easily seen to fail, for example, if the velocity field is fully recirculating, i.e., has no normal component on the domain boundary. In this case the flow map has bounded range, and due to the fixed sampling grid, any estimated flow map gradient is also bounded. Therefore, with growing  $T$ , the FTLE estimate converges to zero.

The practical consequence of this is that for FTLE computed with the flow map method, it does not make sense to compare values for different integration times  $T_1$  and  $T_2$ , e.g. for numerically checking convergence. Also, concepts such as MFTLE [Sad10] inherit this problem.

Sadlo [Sad10] discusses the problem that FTLE computation with the two methods give quite different results in the case of a flow that splits without shear (see Fig. 2). This case can be illustrated by the velocity field

$$\mathbf{u}(x, y) = \begin{pmatrix} 1 \\ v(x, y) \end{pmatrix} \quad \text{where } v(x, y) = \begin{cases} 0 & x < 0 \\ c \operatorname{sgn}(y) & x \geq 0 \end{cases} \quad (3)$$

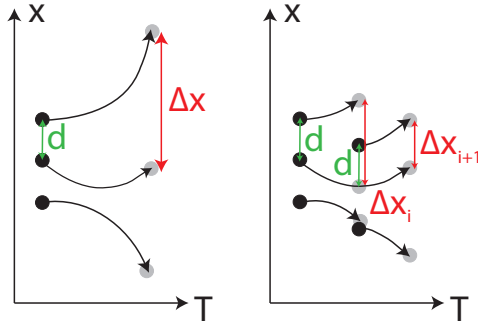
Its flow map, in the interesting region  $-T \leq x \leq 0$ , is

$$\phi_{t_0}^{t_0+T}(x, y) = \phi_0^T(x, y) = \begin{pmatrix} x + T \\ y + c(x + T)\text{sgn}(y) \end{pmatrix} \quad (4)$$

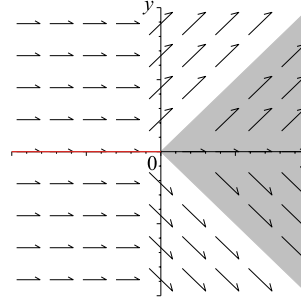
and its gradient is

$$\nabla \phi_{t_0}^{t_0+T}(x, y) = \begin{bmatrix} 1 & 0 \\ c \text{sgn}(y) & 1 + cT \delta(y) \end{bmatrix} \quad (5)$$

where  $\delta(y)$  is the Dirac delta which is infinity at  $y = 0$  and zero elsewhere. The FTLE is then also a Dirac delta, for small  $cT$  it is approximately  $c\delta(y)$ .



**Fig. 1** Flow map computation without and with renormalization.



**Fig. 2** Synthetic example of a flow splitting at an obstacle at the non-negative  $x$ -axis.

If the FTLE is computed with renormalization, this works correctly everywhere except on the  $x$ -axis. However, even though this method is likely to give correct FTLE values everywhere where evaluated, the resulting visualization would miss the only important feature, namely the infinitely thin ridge. For the purpose of visualizing LCS, the renormalization method can therefore not be used. The usual way is to sample FTLE on a grid and to use finite differences for estimating the gradients.

If FTLE is computed with finite differences of the flow map, the infinite gradient at  $y = 0$  gets estimated as a value which is inversely proportional to the sampling density. This causes a problem in adaptive methods, where the computed function should not depend directly on the sampling density. The problem is how to define a stopping criterion, given the fact that estimated FTLE values can depend directly on the mesh width.

Even though velocity fields from CFD simulations do not have discontinuities, they have large gradients especially at boundaries. In principle, gradients are bounded, and therefore by sampling the flow map fine enough, the problem could be solved. However, the sampling would have to be at least as fine as the smallest cells of the CFD grid, and even if adaptive sampling was used [SP07], the refinement cannot be limited to the local density of the CFD grid because the flow map can connect regions of different sampling densities. In practice, this means that the problem of the “flow split without shear” exists, because the necessary sampling

density cannot be afforded. The problem can be alleviated by using spline interpolation in the estimation of flow map gradients [GGTH07]. The approach presented in the current paper allows to use adaptive sampling of the FTLE field during ridge computation, while avoiding the problem of underestimating the gradient during the process.

### 3.2 FTLE computation from smoothed flow map

The proper way of dealing with the above discretization issues is to work in scale-space [Lin94]. Instead of using the original flow map which is possibly sampled too coarsely to resolve all features contained in the underlying numerical velocity field, a smoothed version is used. In the classical scale-space approach, smoothing is done with a Gaussian  $G_\sigma$  with standard deviation  $\sigma$ , which is called the *scale of observation*. The scale of observation can be chosen as fine as the smallest cells of the data grid, or it can be chosen larger to either reduce computation time or focus at larger scale features.

If the flow map from the above example is smoothed with  $G_\sigma$ , then the step function  $\text{sgn}(y)$  becomes  $\text{erf}(y/(\sqrt{2}\sigma))$  and thus the flow map gradient

$$\nabla\phi_0^{t_0+T}(x,y) = \begin{bmatrix} 1 & 0 \\ c \text{sgn}(y) & 1 + 2cTG_\sigma \end{bmatrix} \quad (6)$$

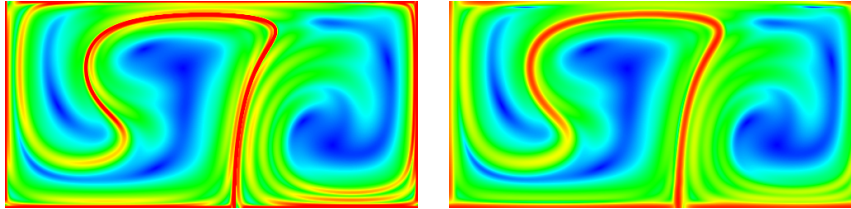
The FTLE is now (for small  $cT$ ) approximately  $2cG_\sigma$ . That means, in this example, the FTLE ridge is a Gaussian with a standard deviation that equals the scale of observation. Here, a sampling density of  $\sigma$  would suffice to reconstruct the FTLE ridge.

In general, the relationship between the scale of observation and the minimal sampling density is more complicated, and in particular, depends on the integration time  $T$ . It is intuitively clear that the frequencies contained in an FTLE field increase if the integration time  $T$  increases, because the longer the integration time is, the longer does the stretching-and-folding effect distort the flow map.

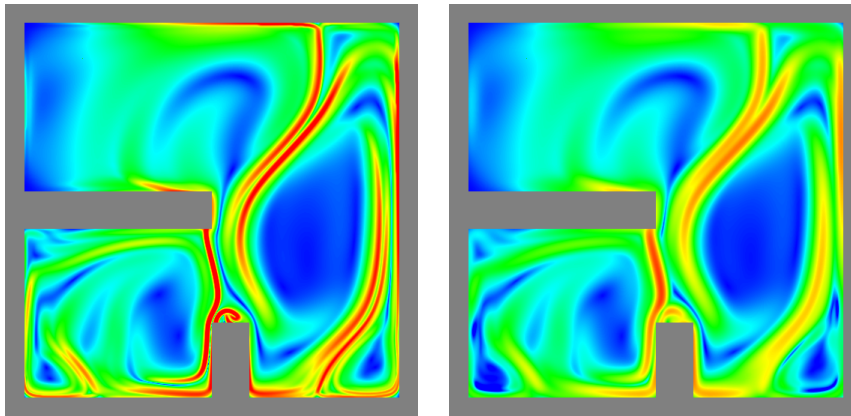
This leads to the assumption that for a given  $T$  there is a minimum sampling density for the flow map such that the FTLE can be reliably computed. By convolving this flow map with a Gaussian, high frequencies can be removed again and the LCS (FTLE ridges) are obtained at a larger scale.

We are not aware of any analytical derivation of the minimum sampling density needed to compute the FTLE of a given velocity field for a given  $T$ . However, the minimum sampling density can be found algorithmically by performing a (local) subdivision as long as the FTLE value changes more than a prescribed error threshold. When comparing two subdivision levels it is important to do this using the same scale of observation. The effect is that high gradients are no longer estimated inversely proportional to the sampling width. It is also important to compute the flow map gradient not with finite differences but by convolution with derivatives of the Gaussian.

This way, scale-space provides the solution for the adaptive refinement problem. If the scale parameter  $s$  is kept fixed, flow map samples can be added adaptively without the aforementioned detrimental effect of the decreasing sampling width. Fig. 3 shows FTLE computed at two different scales from the well-known “double gyre” velocity field [SLM05]. A second example is a simulated 2D air convection flow (Fig. 4).



**Fig. 3** FTLE of the “double gyre” field, computed with  $T = 10$ , and colored from 0 (blue) to 0.4 (red). Scale 0.004 (left) and 0.016 (right). The size of the domain is  $2.0 \times 1.0$ .

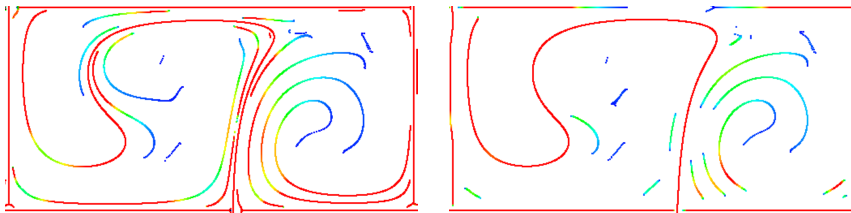


**Fig. 4** FTLE of the convection data, computed with  $T = 0.5$ , and colored from 0 (blue) to 7 (red). Scale 0.0004 (left) and 0.0012 (right). The size of the domain is  $0.1 \times 0.1$ .

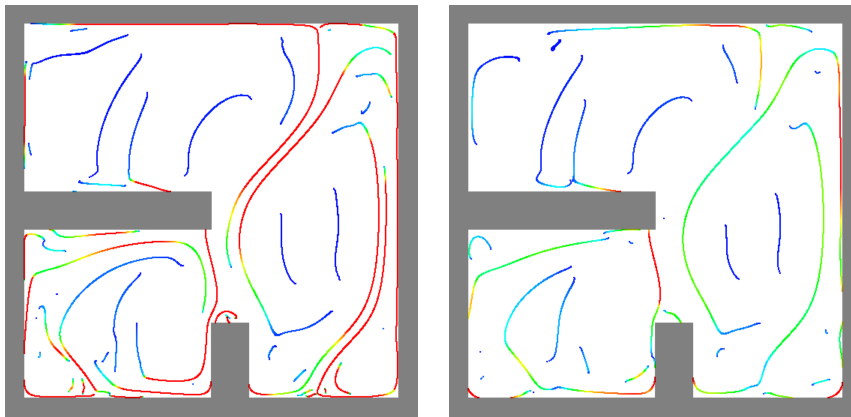
#### 4 Ridge extraction from FTLE fields

In this study we did all ridge extraction with the C-ridge method [SPFT11] which is computationally easier and produces less noisy results than the (maximum convexity) height ridge [Ebe96]. According to either ridge definition, a point lies on a co-dimension one ridge if the FTLE field assumes a maximum in a specified transver-

sal direction. The transversal direction is an eigenvector of  $\mathbf{C}$  or of the Hessian of the FTLE, respectively, and the corresponding eigenvalue must be negative and the minimum eigenvalue. Haller postulates [Hal10] that this eigenvector of  $\mathbf{C}$  is perpendicular to the intended ridge direction. Our choice of the  $\mathbf{C}$ -ridge is not crucial, the height ridge could as well be chosen. For extracting FTLE ridges we need as pre-computed data the FTLE value and the transversal direction on the points of a (regular or adaptive) sampling grid. The ridge extraction is then done per sampling point and consists in finding a nearby FTLE maximum along the transversal direction, where “nearby” means within a 1-cell neighborhood. Finding the FTLE maximum is done using bisection and by looking for a zero crossing of the FTLE gradient. The FTLE gradient is estimated by convolution with the gradient of a Gaussian, the standard deviation of which is the scale parameter of the ridge extraction.



**Fig. 5** Ridges of the FTLE fields from Fig. 3, colored by ridge strength (red=high). At a higher scale many of the smaller ridges are removed.



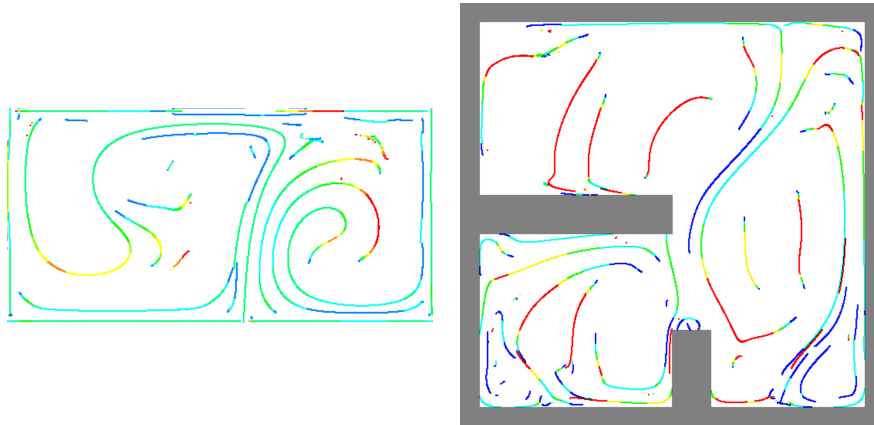
**Fig. 6** Ridges of the FTLE fields from Fig. 4, colored by ridge strength (red=high). At a higher scale, we can see that the ridge separating the large and the small room is the most prominent.

### 4.1 Optimal scale ridges of FTLE

The FTLE field can be interpreted as a 2D or 3D image, and therefore the concept of scale-space ridges [Lin98] is directly applicable. It will extract ridges of different scales by automatically adapting the local scale of observation to the “width” of the ridge. A *scale-space ridge point* is a ridge point with the additional property that it is a maximum of the *ridge strength* in the transversal direction given by the particular ridge definition. Among the various measures for ridge strength, we chose the *normalized curvature*, i.e., the second derivative of FTLE in the transversal direction, multiplied with the normalization factor  $s^{\frac{3}{4}}$ , where  $s$  is the scale parameter.

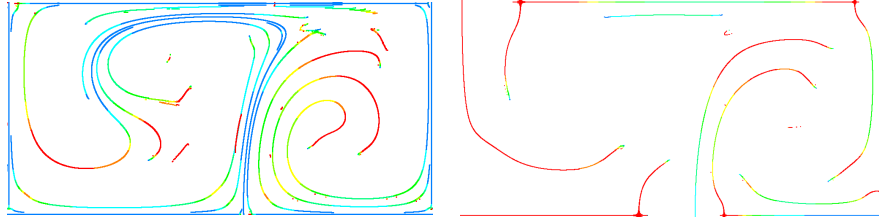
The procedure for extracting scale-space extends the one described in Section 4 by a loop over the scales. In addition, any ridge point obtained at scale  $s_i$  is checked for being also a scale-space ridge point. For this, corresponding ridge points in scales  $s_j$  (with  $j = i - 1, i, i + 1$ ) are searched for on the line given by the transversal direction. Then the ridge strength for the three ridge points is computed. The required second derivative of the FTLE in the transversal direction is computed using the Hessian of FTLE which is obtained by convolution with the Hessian of  $G_{s_j}$ .

When extracting scale-space ridges from our two test data (Fig. 7) it turned out that ridge points were indeed extracted at different scales. However, by comparing the scale-space ridges with a set of fixed-scale ridges (using the same scale for FTLE computation and for ridge extraction), it turned out that the scale-space ridge is very similar to one of the fixed-scale ridges. Surprisingly, however, this fixed scale turned out to be smaller than the  $\sigma$  that was used in the FTLE computation. In both cases, this scale is roughly  $0.75\sigma$ . We do not claim that there exists such a factor, but it seems that it is not optimal to extract ridges at the same scale as was used for FTLE computation. A slightly smaller scale may yield more meaningful detail.



**Fig. 7** Left: Scale-space ridges of FTLE computed with  $\sigma = 0.08$  (double gyre) and  $\sigma = 0.004$  (convection flow). Ridges are colored by scale. Left: double gyre, blue= $0.25\sigma$ , red= $2\sigma$ . Right: convection, blue= $0.5\sigma$ , red= $1.5\sigma$ .

There exists an alternative way of computing scale-space ridges of FTLE. Rather than using a fixed FTLE “image”, FTLE computation can be done on-the-fly and at the same scale as the subsequent ridge extraction. The results (Fig. 8) again are visually close to ridges extracted at a single scale. In Fig. 8(left) the range of scales was



**Fig. 8** Scale-space ridges where computation of FTLE is done on-the-fly for the double gyre data. Integration time was  $T = 10$  (left) and  $T = 4$  (right). Range of scales: 0.002 (blue) to 0.016 (red).

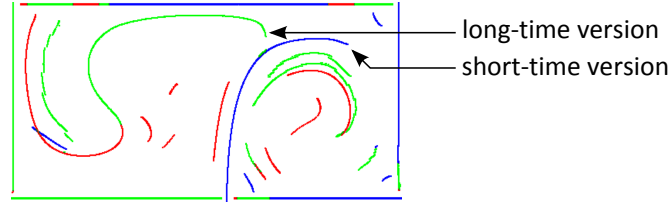
chosen such that even the smallest scale produces a smoothed FTLE image. Consequently, the scale-space ridges are similar to the fixed-scale ridges at this lowest scale. Larger scales do not contribute much besides slightly deforming the ridges. In Fig. 8(right) the flow map has only large-scale features, and the range of scales was chosen such that it extends below the minimum scale of the features. Here, the ridge image is similar to a fixed-scale image at an intermediate scale.

Based on the observation that larger scales have little contribution to FTLE ridges, the only potential advantage of the scale-space ridge method is that it can detect the minimum feature scale. However, this information is also obtained from adaptive FTLE computation. It is certainly more efficient to first compute an FTLE and let it follow by a ridge extraction using the same scale, or possibly a slightly smaller scale, taking the above observation into account. This can be done also if a spatially varying scale is used.

## 4.2 Optimal integration time ridges of FTLE

In Section 4.1 it was assumed that the integration time  $T$  was given and an optimal scale was searched for. However, in the context of FTLE ridges it is even more useful to search for an optimal integration time, given a scale of observation. Computing such optimal integration time ridges is similar to computing scale-space ridges (Section 4.1), with looping over integration times instead of scales. However, when comparing ridge strengths for two different integration times, we have to account for the fact that ridges tend to get stronger with increasing  $T$ . The deformation of a “fluid” element under the flow map is expressed by the Cauchy-Green tensor  $\mathbf{C}$ . The square root of the smallest eigenvalue of this tensor,  $\lambda_{\min} \mathbf{C}$ , expresses contraction in the direction where this is maximized. This direction tends to be aligned with the

FTLE ridges, therefore we propose to modify ridge strength by multiplying it with  $\lambda_{\min} \mathbf{C}$ .



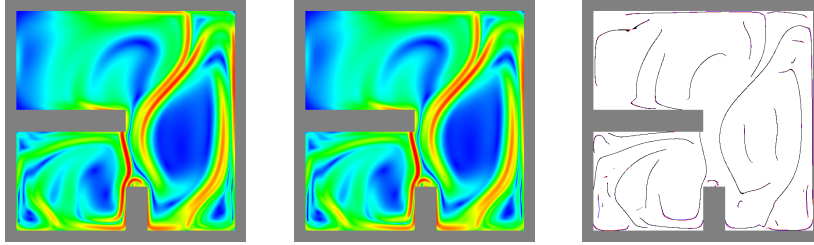
**Fig. 9** Ridge of double gyre at multiple integration times  $T$  ranging from 4 (blue) to 10 (red). We can see how the main separation in the double gyre is detected better as a combination of a long-term FTLE ridge and a short term FTLE ridge.

Such scale-space ridges based on optimal integration time are shown in Fig. 9. When looking at the strong ridge originating near the center of the bottom line, it can be seen that this feature is represented twice. A short-time version of it is curved to the right, while a long-time version is curved to the left. They both represent approximations for material structures, having different life-times. Any ridge extraction at a fixed  $T$  would give at most one of these two manifestations of the ridge.

## 5 Incremental computation of the flow map

If FTLE are to be computed for multiple initial times or multiple integration times, the reuse of computed trajectories can save computing time. In the case of a fixed initial time  $t_0$ , this is easily done by extending trajectories at their end points. If trajectories are seeded on a regular grid, this results in all flow maps sampled on this regular grid. The case of fixed  $T$  but variable  $t_0$ , needed for animated visualization, is more difficult, because shortening trajectories at their beginning results in flow maps sampled on distorted grids. This approach is possible [SRP10], but requires frequent re-initialization to avoid overly distorted grids.

A better alternative is to compute partial flow maps for a sequence of initial times  $t_0 + i\Delta t$  and integration time  $\Delta t$ . A contiguous subset of these are then composited to the flow map for the desired sliding time window. Composition of flow maps requires their values at arbitrary points, which are obtained by convolution with a Gaussian kernel. Of course, such a repeated smoothing requires the choice of a smaller  $\sigma$ . Convolution with  $G_\sigma$  repeated  $N$  times is equivalent to convolution with  $G_{\sqrt{N}\sigma}$ . In the proposed incremental approach convolution alternates with application of partial flow maps. Therefore, it is hard to determine the standard deviation for the single flow map method that exactly corresponds to the partial flow map method with  $G_\sigma$ . Nevertheless, our experiments showed that  $\sqrt{N}\sigma$  works in a qualitative sense.



**Fig. 10** (left) FTLE computed with single flow map and (middle) composition of partial flow maps, colored from 0 (blue) to 7 (red). The faster, approximate method (middle) yields visually almost identical results. (right) Ridges (at scale 0.0008) of the above two FTLE fields computed with single (red) and partial (blue) flow map method, colored black where identical within pixel precision.

With this setting, the partial flow map method produced FTLE values with a mean absolute error of 0.025, at a data range of  $[-1.16, 7.36]$ , which are barely noticeable in Fig. 10. In the extracted ridges, the error is mostly below the sampling width of the grid (Fig. 10 (right)). In the “double gyre” example, the mean absolute error is only 0.0004, at a data range of  $[0.005, 0.424]$  and the extracted ridges are visually indistinguishable.

Often, an exact scale value is not required, e.g., when computing optimal-scale ridges (Section 4.1). It has to be pointed out that the number of partial flow maps must be kept relatively small because the standard deviation cannot be chosen arbitrarily small. We observed that for sufficient approximation quality this should be about 2.0 times the sampling width.

The incremental computation reduces the integration time by a factor  $1/N$ , assuming that a sequence of flow maps for successive times is computed. Gaussian smoothing has to be done  $N$  times instead of just once. If done efficiently, computing time for one such smoothing is dominated by the forward and inverse FFT operation. This means that a speedup of close to  $N$  can be achieved by using the partial flow map method.

## 6 Conclusion

In this paper we showed that the scale-space offers an elegant way to cope with undersampling problems that can occur due to limited space and computing time availability. This way, smoothed FTLE fields can be correctly computed, even in adaptive methods, where existing methods produced results that were biased by the sampling width. Also, a series of smoothed FTLE in a sliding time window can be computed efficiently using a set of partial flow maps.

An often criticized problem of FTLE is its dependence on a parameter, the integration time. While FTLE itself is not parameter-free, FTLE ridges become parameter-free with our proposed definition of optimal-integration time ridges.

Finally, a topic that deserves further study is how to handle trajectories that prematurely leave the domain. Given the strong dependence of FTLE on the choice of  $T$ , it seems problematic to include such trajectories into FTLE visualizations and into further processing such as ridge extraction. On the other hand, excluding them means to exclude large parts of the domain from visualization. Fully recirculating flow, as we used in our examples, is the exception rather than the rule in application-related velocity fields.

## Acknowledgments

We thank Filip Sadlo for the simulation of the air convection. The project Sem-Seg acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number 226042.

## References

- [BGGSS80] BENETTIN G., GALGANI L., GIORGILLI A., STRELCYN J.-M.: Lyapunov Characteristic Exponents for smooth dynamical systems and for Hamiltonian systems; a method for computing all of them. Part 1: Theory. *Meccanica* 15 (1980), 9–20.
- [BP02] BAUER D., PEIKERT R.: Vortex tracking in scale space. In *Data Visualization 2002: Proc. of the 4th Joint EUROGRAPHICS – IEEE TCVG Symp. on Visualization (Vis-Sym 2002)*, Ebert, Brunet, Navazo, (Eds.). Eurographics, 2002, pp. 233–240.
- [BR10] BRUNTON S. L., ROWLEY C. W.: Fast computation of finite-time Lyapunov exponent fields for unsteady flows. *Chaos* 20, 1 (2010), 017503.1–017503.9.
- [Ebe96] EBERLY D.: *Ridges in Image and Data Analysis*. Computational Imaging and Vision. Kluwer Academic Publishers, 1996.
- [FK00] FLORACK L., KUIJPER A.: The topological structure of scale-space images. *J. Math. Imaging and Vision* 11, 1 (2000), 365–79.
- [GGTH07] GARTH C., GERHARDT F., TRICOCHÉ X., HAGEN H.: Efficient computation and visualization of coherent structures in fluid flow applications. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1464–1471.
- [GWT\*08] GARTH C., WIEBEL A., TRICOCHÉ X., JOY K., SCHEUERMANN G.: Lagrangian Visualization of Flow-Embedded Surface Structures. *Computer Graphics Forum* 27, 3 (May 2008), 1007–1014.
- [Hal00] HALLER G.: Finding finite-time invariant manifolds in two-dimensional velocity fields. *Chaos* 10(1) (2000), 99–108.
- [Hal01] HALLER G.: Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D* 149 (2001), 248–277.
- [Hal02] HALLER G.: Lagrangian coherent structures from approximate velocity data. *Physics of Fluids* 14 (2002), 1851–1861.
- [Hal10] HALLER G.: A variational theory of hyperbolic Lagrangian Coherent Structures. *Physica D: Nonlinear Phenomena* 240, 7 (2010), 574–598.
- [HY00] HALLER G., YUAN G.: Lagrangian coherent structures and mixing in two-dimensional turbulence. *Physica D* 147(3) (2000), 352–370.
- [JV09] JIMENEZ R., VANKERSCHAVER J.: *Optimization of FTLE Calculations Using nVidia's CUDA*. Tech. rep., California Institute of Technology, 2009.

- [KCS08] KINSNER M., CAPSON D., SPENCE A.: Scale-space ridge detection with GPU acceleration. In *Electrical and Computer Engineering, 2008. CCECE 2008. Canadian Conference on* (May 2008), pp. 1527–1530.
- [KE07] KLEIN T., ERTL T.: Scale-space tracking of critical points in 3d vector fields. In *Topology-Based Methods in Visualization* (2007), Hauser, Hagen, Theisel, (Eds.), pp. 35–49.
- [KPH\*09] KASTEN J., PETZ C., HOTZ I., NOACK B., HEGE H.-C.: Localized finite-time lyapunov exponent for unsteady flow analysis. In *Vision Modeling and Visualization* (2009), Magnor M., Rosenhahn B., Theisel H., (Eds.), vol. 1, Universität Magdeburg, Inst. f. Simulation u. Graph., pp. 265–274.
- [KSSW09] KINDLMANN G. L., SAN JOSE ESTEPAR R., SMITH S. M., WESTIN C.-F.: Sampling and visualizing creases with scale-space particles. *IEEE Trans. Visualization and Computer Graphics* 15, 6 (Nov/Dec 2009), 1415–1424.
- [LHZP07] LARAMEE R., HAUSER H., ZHAO L., POST F.: Topology-based flow visualization, the state of the art. In *Topology-Based Methods in Visualization* (2007), Hauser, Hagen, Theisel, (Eds.), pp. 1–20.
- [Lin94] LINDBERG T.: *Scale-Space Theory in Computer Vision*. The Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, 1994.
- [Lin96] LINDBERG T.: Scale-space: A framework for handling image structures at multiple scales. In *Proc. CERN School of Computing* (1996).
- [Lin98] LINDBERG T.: Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision* 30, 2 (1998), 117–154.
- [LM10] LIPINSKI D., MOHSENI K.: A ridge tracking algorithm and error estimate for efficient computation of Lagrangian coherent structures. *Chaos* 20, 1 (2010), 017504.1–017504.9.
- [LR10] LEKIEN F., ROSS S. D.: The computation of finite-time Lyapunov exponents on unstructured meshes and for non-Euclidean manifolds. *Chaos* 20, 1 (2010), 017505.1–017505.20.
- [PD10] PEACOCK T., DABIRI J.: Introduction to focus issue: Lagrangian coherent structures. *Chaos* 20, 1 (2010), 017501.1–017501.3.
- [PPF\*10] POBITZER A., PEIKERT R., FUCHS R., SCHINDLER B., KUHN A., THEISEL H., MATKOVIC K., HAUSER H.: On the Way Towards Topology-Based Visualization of Unsteady Flow—the State of the Art. In *State of the Art Reports* (May 2010), Hauser H., Reinhard E., (Eds.), Eurographics Association, pp. 137–154.
- [Sad10] SADLO F.: *Computational Visualization of Physics and Topology in Unsteady Flow*. PhD Dissertation No. 19284, ETH Zurich, 2010.
- [SLM05] SHADDEN S., LEKIEN F., MARSDEN J.: Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. *Physica D Nonlinear Phenomena* 212 (Dec. 2005), 271–304.
- [SP07] SADLO F., PEIKERT R.: Efficient Visualization of Lagrangian Coherent Structures by Filtered AMR Ridge Extraction. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov 2007), 1456–1463.
- [SPFT11] SCHINDLER B., PEIKERT R., FUCHS R., THEISEL H.: Ridge Concepts for the Visualization of Lagrangian Coherent Structures. In *submitted to TopoInVis 2011* (2011), pp. 1–14.
- [SRP10] SADLO F., RIGAZZI A., PEIKERT R.: Time-Dependent Visualization of Lagrangian Coherent Structures by Grid Advection. In *Topological Data Analysis and Visualization: Theory, Algorithms and Applications*, Pascucci, Tricoche, Hagen, Tierny, (Eds.). Springer, 2010, pp. 151–166.
- [TCH10] TANG W., CHAN P. W., HALLER G.: Accurate extraction of LCS over finite domains, with application to flight safety analysis over Hong Kong International Airport. *Chaos* 20, 1 (2010), 017502.1–017502.8.