Formal Evaluation Strategies for Feature Tracking

Andrea Schnorr* Sebastian Freitag Dirk Helmrich Torsten W. Kuhlen Bernd Hentschel

Visual Computing Institute, RWTH Aachen University
JARA – High-Performance Computing

Abstract

We present an approach for tracking space-filling features based on a two-step algorithm utilizing two graph optimization techniques. First, one-to-one assignments between successive time steps are found by a matching on a weighted, bi-partite graph. Second, events are detected by computing an independent set on potential event explanations. The main objective of this work is investigating options for formal evaluation of complex feature tracking algorithms in the absence of ground truth data.

1 Introduction

Tracking the temporal evolution of features is a key method in visualizing and analyzing time-varying data. Existing feature tracking approaches (e.g., [4]) have focused on structures that cover only a small portion of the data domain, i.e. *sparse* features. In contrast, in our work, we concentrate on the tracking of *space-filling* features, i.e. structures that cover the entire data domain. Due to a close collaboration with domain experts from fluid mechanics, we focus on the tracking of a specific type of feature, namely *dissipation elements* [5], a space-filling structure definition from turbulent flow analysis. This work is motivated by our development of a new tracking approach which is hard to validate due to the lack of ground truth data. This is especially true for cases where even small data sets contain thousands of features. Therefore, we evaluated our method on different flow simulation data sets containing sparse and space-filling features by manual visual inspection of the results. Furthermore, we present a set of options for further validation. Combining them offers a more meaningful evaluation of our approach.

2 Method

Our feature tracking approach is posed as a combination of two graph optimization problems. Assuming that continuations explain the majority of features, we first try to detect these 1 : 1 assignments by computing a maximum-weight, maximum-cardinality matching on a bi-partite graph. To this end, we chose the *pseudo-flow* algorithm by Goldberg and Kennedy [1] to solve the matching. The matching either captures a continuation directly or identifies the largest component of a merge or split event. Second, we construct a graph of potential explanations for all feature objects that have not been explained by the initial 1 : 1 assignment. The connections defined by the matching are assumed to be part of any potential explanation. Those explanations are potentially conflicting because they provide different explanations for at least one shared feature. We compute a valid assignment between successive time steps by searching for a maximum weight, independent set on the graph that is constructed from the potential explanations. Hereby, each node represents a potential event explanation whereas edges between nodes indicate contradictions between those explanations. Thus, a maximum-weight independent set yields a conflict-free selection of event explanations with maximal feature similarity. In our current implementation, we formulate the independent set as an integer linear programming, which we solve using the *CBC* solver of the *COIN-OR* project. The solver features several heuristics. We currently operate with an empirically determined convergence threshold of 0.01, such that the weight of the independent set is within 1% of an estimated upper bound provided by the solver, i.e., the solution found is a 0.99-approximation of the optimal solution. Nodes which are not assigned after these steps are assumed to be the result of a birth or death event. A detailed description of the method can be found in [2, 3].

3 Results

In this section, we evaluate our approach on three data sets resulting from a high-resolution direct numerical simulation (DNS) of homogeneous, isotropic turbulence inside a box. Due to a lack of a ground truth for a formal validation, the presented results are gathered from manual visual inspection which is inline with previous approaches [4]. For the first data set with a spatial resolution of 256^3, dissipation elements are computed in a pre-processing step resulting in a scalar field per time step containing the unique element ID per grid point which is used by the tracking. Each time step has a size of 64 MB and contains an average number of about $29k$ features. Since all feature objects are tracked and their evolution is stored, in our approach single features or a group of features can be visualized to depict their temporal evolution and the events they are involved in.

For our evaluation, we picked out a number of dissipation elements which are located close to each other and inspected their evolution for plausibility. All features are color-coded by their ID with a random selected color. In case of a continuation, the feature maintains its color. All features participating in an event – e.g., merge or split – receive a new color to ease the distinction of different features which might be densely packed because of the space-filling characteristics and, hence, not distinguishable if they receive the same color. This is in contrast to existing approaches where all features or at least the largest component in an event maintains its color. In Figure 1 we show an exemplary tracking of five dissipation elements which contact each other. The dense packing of the selected elements illustrates the space-filling characteristics. While the amber, dark green, turquoise, and cyan element in the left portion of the image continue and remain as single features, the violet one in the lower right portion of the image first splits into two parts (red and magenta) and then immediately reconnects by a merge-event to the green feature in the third time step.

Additionally, we validated our approach on a second data set with dissipation elements which has a spatial resolution of 512^3 and on a vorticity data set with a spatial resolution of 256^3. The latter one which contains sparse features is included since similar settings have previously been used for evaluating tracking algorithms which concentrated on sparse features (cf. [4]). In this case, features are extracted by a threshold on the scalar vorticity magnitude. Each time step of the 512^3 dissipation element data set contains an average number of about $110k$ features, each time step of the vorticity data set an average number of 209 features.

*e-mail: schnorr@vr.rwth-aachen.de

2016 IEEE 6th Symposium on Large Data Analysis and Visualization (LDAV)
23-28 October, Baltimore, Maryland, USA
978-1-5090-5659-0/16/$31.00 ©2016 IEEE

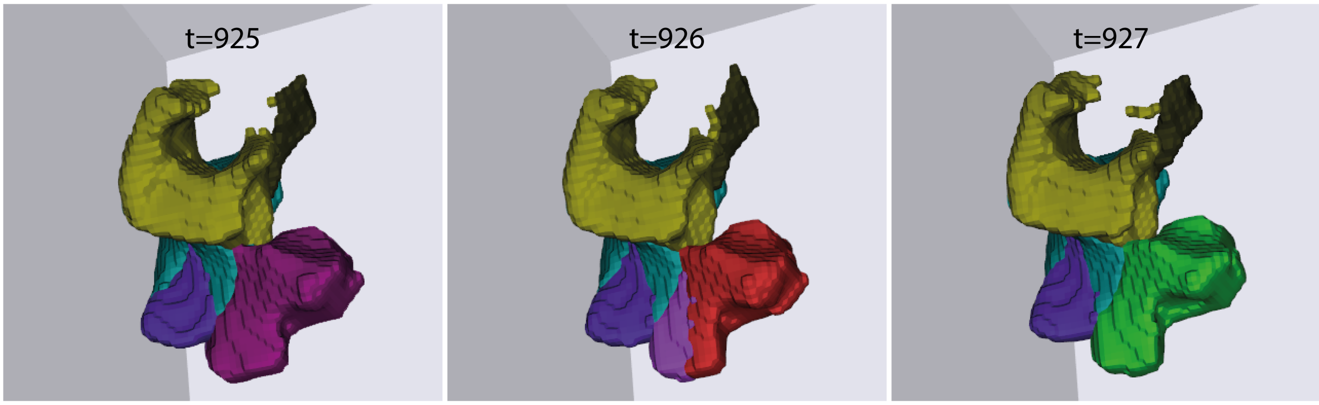

Figure 1: Depiction of a fast splitting and merging sequence of selected dissipation elements in the 256^3 isotropic turbulence data set. While the amber, dark green, turquoise, and cyan elements in the left portion of the image remain largely stable, the violet one first undergoes a split followed by an immediate reconnection. The dense packing of elements illustrates the challenges of tracking space-filling structures.

We measured the runtime performance of our approach on these data sets on a dual socket server featuring two Intel® XEON® E5-2695 v3 CPUs (24 cores 2.3 GHz each) and 512 GB of RAM. However, the algorithm does not (yet) run in parallel, thus it does not benefit from the available cores. On the given hardware, a full tracking step – including data loading, object extraction, matching and event detection – for the vorticity data set takes on average $9.3s$. Most of the time is consumed by I/O, the matching takes $2.9ms$ and the independent set calculation $15.5ms$ on average. The runtimes for the space-filling dissipation element data sets are $151.4s$ and $1,269.2s$ for the 256 and the 512 case, respectively.

4 Discussion & Future Validation

The preliminary results presented in Section 3 show that our algorithm performed as expected for specific data sets. Nevertheless, these promising results do not allow the assumption of the algorithm's correctness in general scenarios. Therefore, a broader validation of the method is required. Unfortunately, a ground truth which would be necessary for a formal validation is missing. Thus, in this section, we discuss a set of options to further evaluate our approach despite the lack of a ground truth.

One aspect to formally prove the tracking correctness of our approach is the test on synthetic data which inclose all possible motions and events such as continuation, split, merge, birth, and death. In this regard, we plan to compose data sets with a known temporal evolution and predefined events which exhibit as similar dynamics as possible to our data sets resulting from DNS. A feasible technique to generate such synthetic data is, e.g., to take a single time step of the simulation data and move the elements across the domain. Additionally, events could be included by defining a feature in a subsequent time step as background to simulate a death event or by dividing one element in two ore more parts with a new unique ID to simulate a split. Birth and merge events could be constructed in the same manner going backwards in time. However, a test on such artificially constructed data sets would have only limited expressiveness since it is hard to model similar temporal dynamics as in real data sets.

For this reason, another facility to evaluate the method is a thorough visual inspection of the resulting temporal evolutions of tracked features by domain scientists. For example, the plausibility of different tracks of one feature or a group of features could be retrieved by experts. This could be done by generating videos of a visualization of the evolution of single features and events as in Figure 1 which could be inspected by the domain experts. Moreover, single features and events could be tracked manually by domain scientists and the results could be compared to the results of the algorithm. Though, a visual inspection of whole data sets – in case of the dissipation element data sets even for one time step – would be to extensive, i.e. this method is limited to a small amount of feature tracks. Nevertheless, this is covered by the tests on synthetic data sets as mentioned above.

Validation using synthetic data and expert reviews only offer internal validity of the algorithm's functionality. Therefore, at least a comparison with the results of existing approaches on the same data sets could be taken into account for external validity. To this end, we plan to investigate the consistency of the tracking results for different approaches and their runtime performance on the synthetic as well as on well-known sparse and dense real data sets.

Independent from this internal and external validation, we determined a dependency of tracking quality on the temporal sampling of the data as previously stated in [4]. In this context, we further plan to investigate the *temporal stability* of our approach in comparison to existing approaches. An increased temporal resolution implies smaller changes between time steps, which drastically reduces the search space during event detection. A reduction in candidate features would allow us to ease the artificial cutoff thresholds of our approach or to omit them entirely.

In summary, we presented an approach for the tracking of sparse and space-filling features and its evaluation by manual visual inspection. Additionally, we depicted various options for further validation which will be addressed in future work and offer a more solid evaluation.

References

[1] A. V. Goldberg and R. Kennedy. An Efficient Cost Scaling Algorithm for the Assignment Problem. *Mathematical Programming*, 71(2):153–177, 1995.

[2] A. Schnorr, S. Freitag, T. W. Kuhlen, and B. Hentschel. Tracking Space-Filling Features by Two-Step Optimization. In T. Isenberg and F. Sadlo, editors, *EuroVis 2016 - Posters*, pages 77–79. The Eurographics Association, 2016.

[3] A. Schnorr, J.-H. Goebbert, T. W. Kuhlen, and B. Hentschel. Tracking Space-Filling Structures in Turbulent Flows. In *Proceedings of the IEEE Symposium on Large Data Analysis and Visualization*, pages 143–144, 2015.

[4] D. Silver and X. Wang. Tracking and Visualizing Turbulent 3D Features. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):129–141, 1997.

[5] L. Wang and N. Peters. The Length-Scale Distribution Function of the Distance Between Extremal Points in Passive Scalar Turbulence. *Journal of Fluid Mechanics*, 554:457–475, 2006.