Tracking Space-Filling Structures in Turbulent Flows

Andrea Schnorr[1,3*] Jens Henrik Göbbert[1,3] Torsten W. Kuhlen[2,3] Bernd Hentschel[1,3]

[1]Virtual Reality Group, RWTH Aachen University
[2]Jülich Supercomputing Centre, Forschungszentrum Jülich
[3]JARA – High Performance Computing

ABSTRACT

We present a novel approach for tracking space-filling features, i.e. a set of features which covers the entire domain. In contrast to previous work, we determine the assignment between features from successive time steps by computing a globally optimal, maximum-weight, maximal matching on a weighted, bi-partite graph. We demonstrate the method's functionality by tracking *dissipation elements* (DEs), a space-filling structure definition from turbulent flow analysis. The ability to track DEs over time enables researchers from fluid mechanics to extend their analysis beyond the assessment of static flow fields to time-dependent settings.

Keywords: Feature Tracking, Weighted, Bi-Partite Matching, Flow Visualization, Dissipation Elements

1 INTRODUCTION

The analysis of time-varying phenomena is a central aspect of scientific visualization. In this context, time-dependent data is typically given as a series of $T \in \mathbb{N}$ discrete snapshots $\{S(t)\}_{1 \le t \le T}$, each of which captures the underlying system's state at time t. In order to shed light into the temporal evolution of objects, it is essential to automatically identify corresponding objects from successive time steps, a process called *feature tracking*.

While classical feature tracking approaches focus on meaningful structures, which cover only a small fraction of the data domain (cf., e.g. [3]), we concentrate on *space-filling* structures. These partition the entire input domain; every data point inside the domain is assigned to exactly one feature object.

This extension complicates the tracking, since the number of potential matches for every feature object grows and the assignment might be ambiguous: neighboring features might literally "compete" for suitable assignments. To resolve these issues, we model the tracking as a weighted, bi-partite graph matching problem between objects of successive time steps. The edge weights encode feature similarity. The solution is a globally optimal assignment with respect to the chosen similarity metric.

While our work targets space-filling features in general, it is motivated by a close collaboration with domain experts from fluid mechanics, who are interested in a specific type of feature, namely *dissipation elements* (DEs) [4]. We use DEs to evaluate our approach. DEs are tightly related to the notion of the 3D Morse Smale Complex (MSC) [1]; in fact, according to their definition, DEs are equivalent to the 3D Morse Smale cells. Our goal is to extend the DE analysis, which has focused on isolated time steps so far, to a time-varying setting.

2 FEATURE TRACKING BY WEIGHTED MATCHING

Structures in each individual time step t are defined as *feature objects* $O_i^{(t)}$. Temporally cohesive *features* are then defined by matching feature objects in consecutive time steps, the resulting paths are of the form $P = (O_{i_0}^{(t)}, O_{i_1}^{(t+1)}, \ldots, O_{i_{k-1}}^{(t+k-1)})$.

*corresponding author: schnorr@vr.rwth-aachen.de

IEEE Symposium on Large Data Analysis and Visualization 2015
October 25–30, Chicago, Il, USA
978-1-4673-8517-6/15/$31.00 ©2015 IEEE

Given two sets of feature objects from two consecutive time steps, our approach consists of the following steps: First, we build a *matching graph* containing one node per $O_i^{(t)}$ and edges whose weights correspond to the similarity of the two adjacent objects. Second, we compute a maximum-weight, maximal matching providing a globally optimal solution for this bi-partite graph.

2.1 Building a Bi-Partite Graph

To build the matching graph, we consider the feature objects of two consecutive time steps. We set

$$U = \{O_i^{(t)}\}_{1 \le i \le n_t} \text{ and } V = \{O_j^{(t+1)}\}_{1 \le j \le n_{t+1}}.$$

Next, we compute for every pair $(u,v) \in U \times V$ the similarity $c(u,v)$ of the corresponding feature objects, which is used as edge weight in the matching graph. The similarity metric is designed to yield values in $[0 \ldots C_{max}]$, where C_{max} corresponds to the maximum overall similarity. We include only edges with $c(u,v) > 0$. These steps are illustrated in Figure 1.

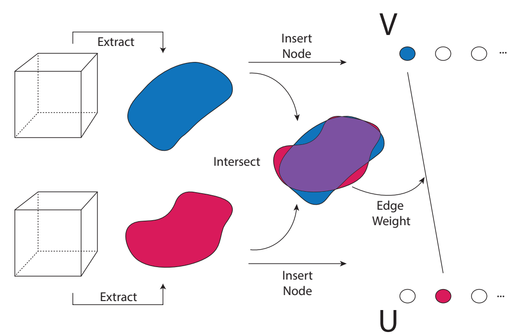

Figure 1: Illustration of the matching graph construction.

Currently, we use the normalized volume overlap of two feature objects to estimate similarity. To this end, we compute the intersection of the point sets associated with the feature objects and normalize it with the size of the larger point set:

$$c(u,v) = \frac{|u \cap v|}{max(|u|,|v|)}.$$

In order to avoid a full-scale comparison for all possible pairs of objects, we separate this process into a broad phase and a narrow phase. First, we check for every combination of feature objects if their bounding boxes overlap. This is done using an efficient box intersection test proposed in [5] and available in the CGAL library. We then compute the similarity of those objects as described above if and only if their bounding boxes overlap. The intersection is done by using the set intersection operation of the C++ STL. If $c(u,v) > 0$, an edge between the corresponding nodes with appropriate weight is inserted; for all other non-overlapping combinations the edges are not explicitly stored and their weight is assumed to be 0.

Due to the fact that most of the algorithms which solve the matching problem work on quadratic matrices, we assume the node sets U and

V to be of the same size. If they are not, we introduce pseudo-nodes to obtain a quadratic matrix. Every feature which could not be assigned to the next or the previous time-step is matched to a pseudo-node and hence is assumed to participate in an event.

The solution of the linear sum assignment problem (LSAP) appropriate to this bi-partite graph indicates the feature correspondence between the given time steps. While the currently used similarity metric of our framework is the normalized volume overlap of two features, the implementation is exchangeable with other metrics which might be more appropriate if another feature definition is employed.

2.2 Computing a Maximum-Weight, Maximal Matching

Determining a maximum-weight, maximal matching in a bi-partite graph means finding a subset of edges such that every node coincides with exactly one edge and the sum of edge weights is maximal. As this is a well-studied problem in combinatorial optimization, there is a large range of approaches for finding a globally optimal solution.

Based on the structure of our domain problems, we chose the pseudo-flow algorithm by Goldberg and Kennedy with a complexity of $O(\sqrt{n} m log(nC_{max}))$ and adapted it to our purpose [2]. The algorithm is based on the idea that a LSAP can be solved by transforming it into a flow network and solving the corresponding maximum-weight flow problem. To transform a bi-partite graph G into a flow network N, a source node and a sink node are introduced, connected by an arc to every node in U and V, respectively. The arc set of N consists of the edges of G maintaining their weight and the arcs connected to the source or the sink obtaining a weight of 0. All arcs are assigned a *capacity* of one, i.e. they could be used only once in the solution. The assignment problem can be solved by finding a maximum flow with cardinality n, which induces a maximum weight, where n denotes the cardinal number of U and V. A valid flow has to fulfill the *flow conservation constraints* and the *capacity constraints* which require that the total outgoing flow of a node must equal its total influx except for the source and the sink and that the flow is non-negative and does not exceed capacity. A pseudo-flow fulfills the capacity constraints but violates the flow conservation constraints. In the employed algorithm, a pseudo-flow is turned into a feasible flow by finding *alternating paths* and interchanging assigned and unassigned edges along this path. An alternating path is a path in the flow network, which alternates between assigned and unassigned arcs. For a detailed description of the pseudo-flow algorithm, we refer to [2].

3 Results

We evaluated our method on data sets resulting from a direct numerical simulation (DNS) of a homogeneous, isotropic turbulence inside a box; DEs are computed in a pre-processing step. We considered four data sets: *128hr*, *128lr*, *256*, *512* with a spatial resolution of 128^3, 128^3, 256^3, and 512^3, respectively. Each of the first three data sets contains 2,000 time steps. The *512* case contains 100 time steps. For all cases exept the *128hr* every 10-th time step has been written to disk. The *128hr* case is written out for every simulation step.

The correlation between the node and edge counts for the constructed matching graphs of these data sets suggests a strong linear relationship. The median of edges per node in the four cases is 2, 4, 4, and 5 for the *128hr*, *128lr*, *256*, and *512* cases, respectively. Thus, the matrix representing the matching graph is sparse.

Tracking performance is very hard to measure exactly. To aggravate the situation, there is no ground truth against which to compare the results. At the same time, a test on synthetic data would have limited expressiveness since it is hard to model similar dynamics. Hence, we use the fraction of tracked volume data as a proxy for tracking performance. To this end, we relate the number of grid points which have been assigned to the next time step to the overall number of points contained in DEs. In a perfect setting, where every feature could be correlated to one or multiple features in the other time step, one would expect the tracked fraction to be 100%. If a birth or death event occurs, the tracked fraction should be slightly below 100%.

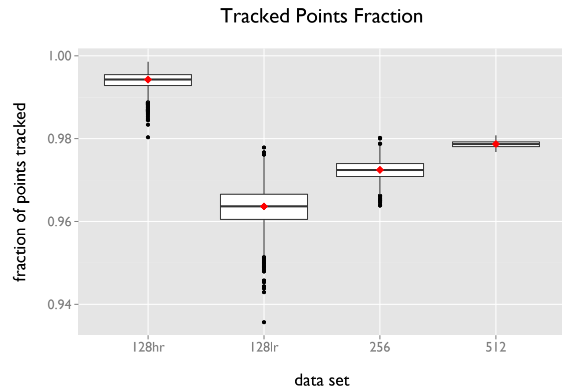

Figure 2: Overview of global tracking quality measured by the fraction of tracked points per data set.

Figure. 2 shows the resulting fraction of tracked points. The mean tracked fraction over the entire temporal range is 99.40%, 96.33%, 97.24%, and 97.86% for the *128hr*, *128lr*, *256*, and *512* case, respectively. In all cases, the missing percentage is attributed to sudden topological changes in the underlying gradient field which lead to discontinuous changes in structure of dissipation elements. In its current implementation, our framework does not include a dedicated event detection, which should be resolved in future work. However, the tracking is able to establish connections between successive time steps for more than 96% of the grid points. The differences in performance between the *128hr* and the other cases can be explained with their temporal resolution.

4 Conclusion and Future Work

We have presented a novel approach for feature tracking which is based on solving a matching problem between successive time steps. Compared to classical methods which have been based on greedy strategies so far, our approach facilitates the tracking of space-filling structures and yields a globally optimal solution for the correspondence of two consecutive timesteps. The proposed method enables fluid mechanics researchers to monitor and assess time-dependent characteristics of DEs.

Our current, purely matching-based approach is not able to detect events between multiple features; assignments have to be one-to-one. However, the frequency of events, such as splits and merges, and the complete temporal evolution of every feature, are of central interest to our domain experts. Thus, we are investigating options to resolve the identification of these events while keeping the matching-based approach.

Another aspect is the workflow's overall performance. The machting algorithm and the entire tracking code are not yet running in parallel. This should also be solved to scale the tracking to larger data sets. Additionally, we plan to investigate other similarity metrics appropriate to DEs or other space-filling structures.

References

[1] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Morse-smale Complexes for Piecewise Linear 3-Manifolds. In *Proceedings of the Annual Symposium on Computational Geometry*, pages 361–370, 2003.

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

[3] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch. The State of the Art in Flow Visualisation: Feature Extraction and Tracking. *Computer Graphics Forum*, 22(4):775–792, 2003.

[4] 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.

[5] A. Zomorodian and H. Edelsbrunner. Fast Software for Box Intersections. *International Journal of Computational Geometry & Applications*, 12(01n02):143–172, 2002.