Eurographics Conference on Visualization (EuroVis), Posters Track (2016)
T. Isenberg and F. Sadlo (Guest Editors)

Tracking Space-Filling Features by Two-Step Optimization

A. Schnorr, S. Freitag, T. W. Kuhlen and B. Hentschel

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

Abstract

We present a novel approach for tracking space-filling features, i.e., a set of features covering the entire domain. The assignment between successive time steps is determined by a two-step, global optimization scheme. First, a maximum-weight, maximal matching on a bi-partite graph is computed to provide one-to-one assignments between features of successive time steps. Second, events are detected in a subsequent step; here the matching step serves to restrict the exponentially large set of potential solutions. To this end, we compute an independent set on a graph representing conflicting event explanations. The method is evaluated by tracking dissipation elements, a structure definition from turbulent flow analysis.

1. Introduction

Feature tracking is a key technique to gain insight into the temporal evolution of objects in time-varying data sets. Existing feature tracking approaches, e.g. [RPS01, SSZC94, SW97], usually operate on the assumption that features only cover a small fraction of the data domain. These approaches test all candidate features from one time step for overlap with a template feature from a neighboring time step and use the best combination of all candidates as solution. First, they check for splitting features from t_i to t_{i+1}, then for merges in the other direction. However, if space-filling features, i.e., structures that cover the entire domain, should be tracked, this approach suffers from two issues. First, the number of possible candidates overlapping with a certain feature grows significantly. Since the number of potential explanations, including continuations, splits, and merges – the latter two of which may include an arbitrarily large subset of overlapping candidates – grows exponentially with the number of overlapping features, the problem quickly becomes intractable in this setting. Second, an assignment of one feature to one or more features in the other time step is usually chosen greedily. This leads to a locally optimal assignment, i.e., the best available solution for the given feature, but it might preclude explanations for other features, which – taken in combination – might be part of a globally optimal event detection. Instead, all explained features are deleted from the search space and are therefore not available for further investigation.

In order to address these issues, we formulate the assignment by means of two graph optimization problems. We first model continuations as a weighted matching problem. We then construct a graph of all conflicting event explanations on which we search for an independent set representing non-conflicting explanations. The bi-partite matching on the features of two consecutive time steps provides one-to-one assignments of features with sufficient overlap. This step significantly reduces the number of possible candidates involved in an event with one feature and with it the possible assignments to be tested. Building a graph containing one node for every possible event explanation and edges to encode conflicts between explanations enables us to consider all possible events and obtain a globally optimal assignment. Additionally, there is no dependency on the ordering of features or events and therefore no preference for continuations, splits, or merges.

2. Method

Our feature tracking approach consists of two steps as illustrated in Figure 1: the bi-partite matching and the subsequent event detection. The first step yields one-to-one assignments for most of the features. Continuations, as well as the largest component of a feature participating in a split or merge between the corresponding time steps are found this way. The event detection step uses the result of the matching as starting point. For all matching edges, an extension with unmatched nodes is tested by building a second graph containing all possible explanations, i.e., splits, merges, and the continuations indicated by the matching, as weighted nodes and conflicts between those events as edges. On the resulting graph, we compute a maximum weight independent set. The selected nodes indicate the features involved in an event like splitting, merging or continuation; all others are assumed as birth or death. These two steps are described in detail in the following sections.

2.1. Weighted Matching

Given two consecutive time steps t_i and t_{i+1}, our approach sets up a weighted, bi-partite graph containing one node for every object in t_i in the set U and one node for every object in t_{i+1} in the set V. The edges between the nodes of U and V are weighted according to

DOI: 10.2312/eurp.20161146

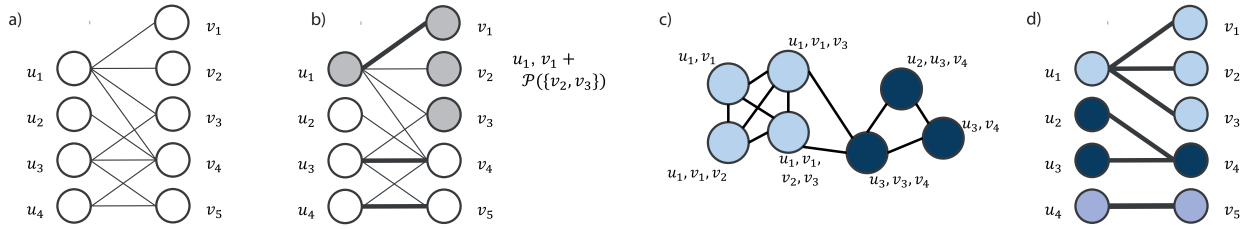

Figure 1: *Illustration of the feature tracking steps. a) shows the bi-partite matching graph for two consecutive time-steps. b) shows the resulting edges of the matching (bold edges) and the nodes considered to paticipate in an event with the first node of U. c) shows the resulting graph on which the independent set is computed, and d) shows the resulting events in the bi-partite graph.*

the similarity between the corresponding objects. While the similarity measure can be chosen arbitrarily in principle, we currently use normalized volume overlap [SW97]. For all combinations of objects which are not overlapping, the edges are not explicitly stored and their weight is assumed to be 0. An example of the resulting graph is illustrated in Figure 1a). On this graph, a maximum-weight, maximal matching is computed using an adapted version of the pseudo-flow algorithm by Goldberg and Kennedy [GK95]. This matching provides one-to-one assignments for most of the features. In Figure 1b) the edges found by the matching are printed in bold. A detailed description of the matching step can be found in [SGKH15].

2.2. Event Detection

The matching step is followed by an event detection using the matching result as input. First, all possible split events containing one matched node from U and a set of nodes from V are constructed. For each matched node $u \in U$ all currently unmatched nodes $\tilde{v} \in V$ connected by a valid edge to u are enumerated. In Figure 1b) this step is illustrated for u_1. In this example, the matching edge connects u_1 with v_1. Additional possible candidates are v_2 and v_3. The power set of the possible candidates is computed and for every element of the power set an event explanation is stored containing the nodes connected by the matching edge, in this case u_1 and v_1, and the nodes in the subset. Thus, the number of potential explanations grows exponentially with the number of overlapping features. In the example, this results in the four possible explanations "u_1 continues as v_1"; "u_1 splits into $\{v_1, v_2\}$"; "u_1 splits into $\{v_1, v_3\}$"; and "u_1 splits into $\{v_1, v_2, v_3\}$". For each of these explanations, a benefit, in our case an increase in normalized volume overlap, is stored. All explanations which share at least one node from U or V conflict with each other, because they provide different explanations for the same feature. Thus, explanations built from different elements of the same power set are conflicting.

All possible merge events containing one matched node from V and a set of nodes from U are constructed afterwards in the same manner. From this data, we construct a second graph, which contains a node with corresponding weight for every explanation and edges between nodes if the corresponding explanations are in conflict. This results in a graph containing cliques as shown in the Figure 1c).

To choose the best explanation for every node, we determine a maximum weight independent set on this graph using a branch and bound algorithm. While the independent set problem is NP-hard in general, the matching step allows us to reduce problems to a tractable size. The solution is a set of nodes that are mutually unconnected. These correspond to a set of explanations, i.e., splits, merges, and continuations relating features from t_i and t_{i+1}, that do not conflict with each other. After inserting those events in the bi-partite graph, we receive a solution as depicted in Figure 1d).

3. Discussion & Conclusion

We have presented a novel approach for feature tracking and event detection on space-filling structures based on the combination of a bi-partite matching and an event detection using a maximum weight independent set problem. In contrast to previous methods which are designed to solve the correspondence problem on sparse data, we use the matching step to reduce the possible candidates for events. Without the matching, the number of possible events in a data set from a direct numerical simulation with a spatial resolution of 128^3 containing up to 6.000 features per time step would be at least two orders of magnitude higher than the number of potential events when factoring in the matching step. Thus, the matching reduces the complexity of the graph on which the independent set is calculated massively. Our approach can be applied to spatially sparse features as well, however, due to relatively small explanations sets, benefits over existing greedy methods might be limited.

Another important aspect for tracking on space-filling data is the ambiguity of possible assignments. The matching as well as the independent set provide the best possible explanation for a feature which overlaps with more than one other object in the other time step. In previous work, an explanation for an object might be chosen if the rating is locally optimal and the object is not considered to be part of any other event anymore. In our approach, all possible events are enumerated and the best solutions are chosen by the independent set. For the aforementioned data set, our approach selects events with an average score of more than 80% normalized volume overlap.

Our current solution is based on the overlap criterion. In future work we will investigate further options to correlate features. Another aspect is the size of the resulting independent set graph. Since the explanation set is still based on an enumeration of a power set, which grows exponentially, we plan to investigate filtering and thresholding mechanisms to further limit the problem size a priori.

References

[GK95] GOLDBERG A. V., KENNEDY R.: An Efficient Cost Scaling Algorithm for the Assignment Problem. *Mathematical Programming 71*, 2 (1995), 153–177. URL: http://link.springer.com/article/10.1007/BF01585996, doi:10.1007/BF01585996. 2

[RPS01] REINDERS F., POST F. H., SPOELDER H. J.: Visualization of Time-Dependent Data with Feature Tracking and Event Detection. *The Visual Computer 17*, 1 (2001), 55–71. URL: http://dx.doi.org/10.1007/PL00013399, doi:10.1007/PL00013399. 1

[SGKH15] SCHNORR A., GOEBBERT J.-H., KUHLEN T. W., HENTSCHEL B.: Tracking Space-Filling Structures in Turbulent Flows. In *Proceedings of the IEEE Symposium on Large Data Analysis and Visualization* (2015), pp. 143–144. doi:10.1109/LDAV.2015.7348089. 2

[SSZC94] SAMTANEY R., SILVER D., ZABUSKY N., CAO J.: Visualizing Features and Tracking Their Evolution. *IEEE Computer 27*, 7 (1994), 20–27. doi:10.1109/2.299407. 1

[SW97] SILVER D., WANG X.: Tracking and Visualizing Turbulent 3D Features. *IEEE Transactions on Visualization and Computer Graphics 3*, 2 (1997), 129–141. doi:10.1109/2945.597796. 1, 2