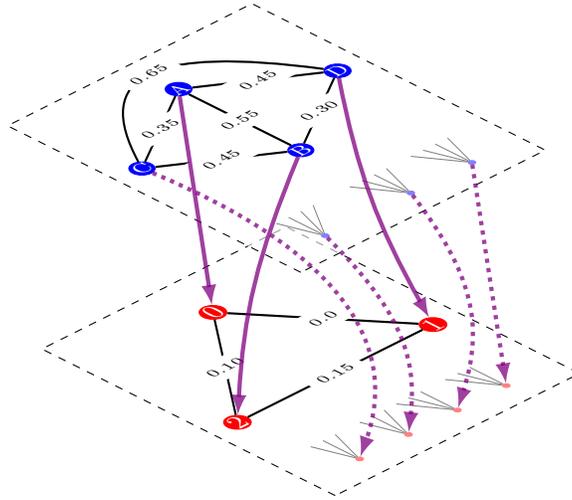


# Distance between Extremum Graphs

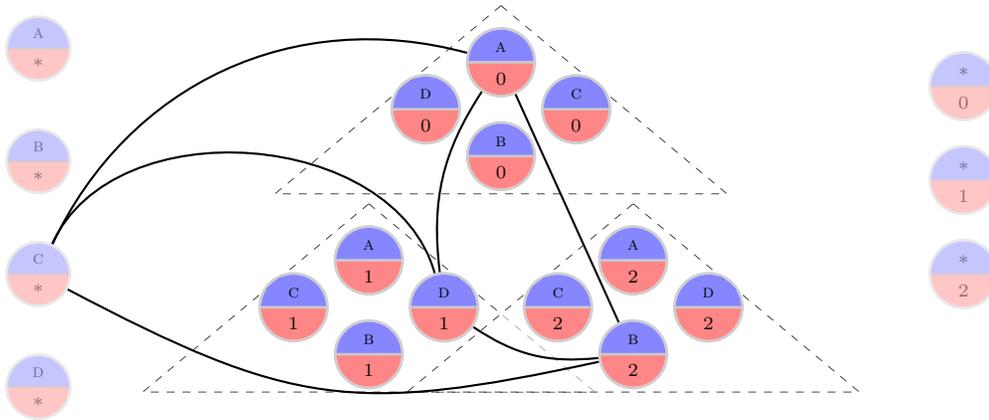
## Supplemental Figures

### 1 Equivalence of maximal cliques and $\rho$ -maps

Section 3.5 discusses distance computation using product graphs. This illustrations shows the equivalence between maximal cliques in the product graph and valid  $\rho$  maps between complete extremum graphs



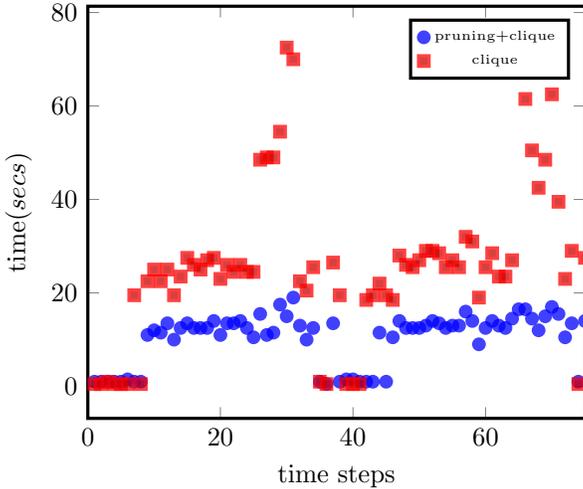
(a) A valid map for  $\rho = 0.45$  is shown above. Dotted edges indicate dummy correspondences.



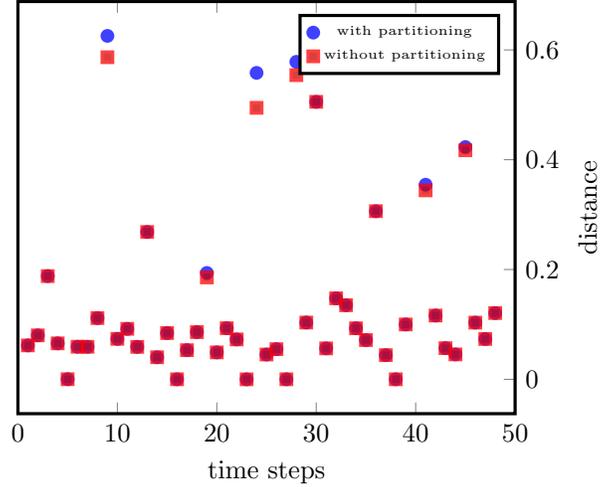
(b) Any valid map appears as a maximal clique in the product graph. Here, the map shown for  $\rho = 0.45$  in figure 1a is shown in the (partial) product graph. It appears as maximal clique. An edge between vertex  $A|0$  and  $B|2$  is included in the product graph since the edge costs of edges  $(A, B)$  and  $(0, 2)$  have a difference  $\leq \rho$ . The vertices representing dummy correspondences(\*) ensure that all extrema appear in the map and the valid map is always maximal. As described in Section 3.5, the weight of a maximal clique indicates the vertex distortion introduced by the map it represents. A maximum weight clique corresponds to minimum vertex distortion and provides an optimal correspondence map between the two complete extremum graphs for a given  $\rho$ . Note that introducing one dummy vertex for each extremum graph is sufficient to compute the appropriate product graph.

## 2 Distance Computation

Section 3.5 discusses pruning and partitioning strategies to sparsify the product graph for efficient clique computation. The following plots show the effect of these strategies for the datasets used in the applications. The table(c) provides graph details and running times for the two applications discussed in Section 4.



(a) This plot shows the effectiveness of pruning in computing the maximum clique for the flow behind a cylinder dataset(Section 4.1). We try to prune 1% of the edges in the product graph before computing the clique. The total time taken to prune the graph and compute the clique(blue) is less than the direct clique computation(red) by a factor of  $2.5\times$  for 75 time steps. The clique computation is done using the Bron-Kerbosch algorithm.



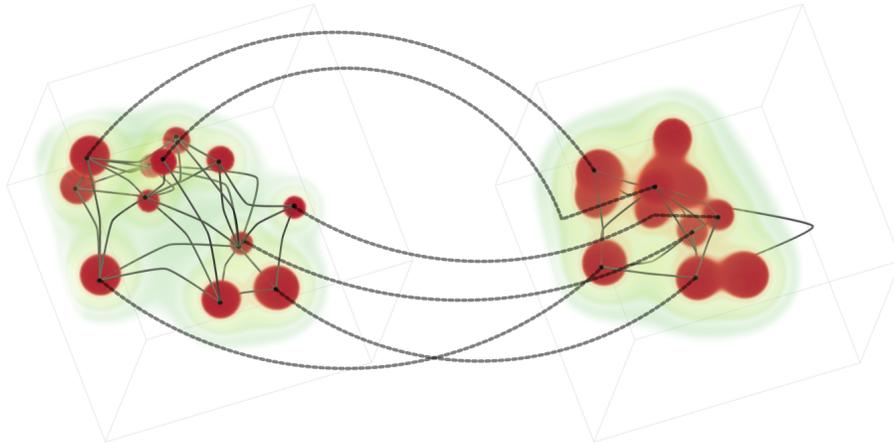
(b) This plot shows the variation in the distance computed for the vortex dataset(Section 4.2) with and without applying partitioning heuristics. For large datasets, where direct clique computation is computationally infeasible, partitioning allows us to compute the clique in a hierarchical fashion. The maximum difference observed between the optimal clique and the hierarchically computed clique for this dataset is  $\leq 10\%$  and typically much smaller.

Dataset	Size	# Ext	CEG Time	#Ext after simp	# vertices product graph	#edges product graph	Product Graph Time	Clique Time	Memory
Flow around cylinder	$400 \times 50$	$\sim 35$	$< 1$	$\sim 35$	$\sim 900$	$\sim 300\text{K}$	$< 5$	$\sim 10$	$\sim 0.2\text{GB}$
Vortex	$128^3$	$\sim 1000$	$\sim 70$	$\sim 60$	$\sim 2000$	$\sim 1500\text{K}$	$\sim 30$	$\sim 50$	$\sim 0.35\text{GB}$

(c) Running time(in seconds) and graph details for distance computations. Experiments were performed on a 2GHz Intel Xeon processor with 16GB RAM. 'Ext after simp' indicates the number of extrema considered for computing the distance. Note that the complete extremum graph (CEG) is computed using all unsimplified extrema, in order to compute correct edge costs. Pruning and partitioning techniques mentioned in the paper have been used to speed up the clique computations.

### 3 Applications

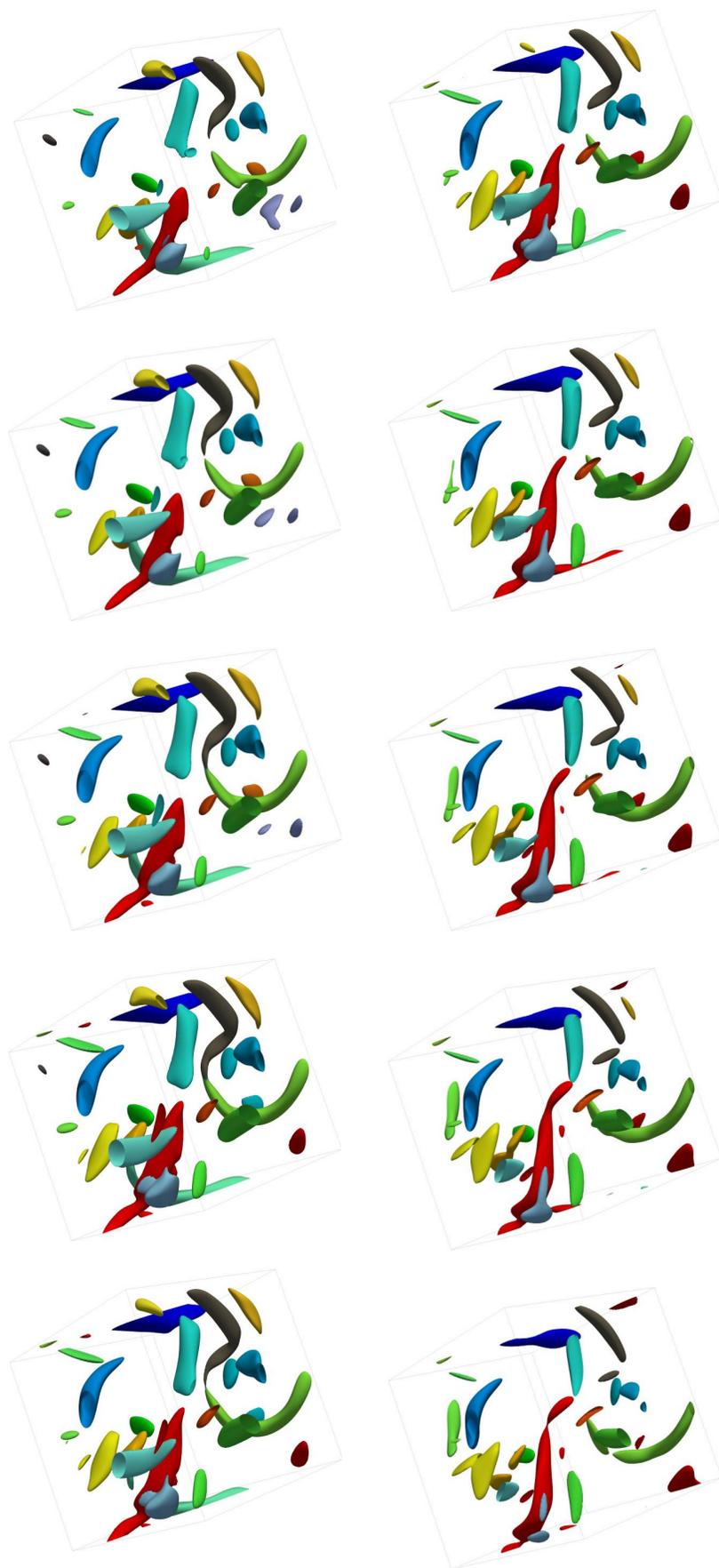
Figure (a) shows the extremum graph and correspondence overlaid for two frames of the synthetic volume dataset. Figures (b) and (c) provide additional results for vortex tracking (Section 4.2)



(a) The extremum graph captures adjacencies between extrema. In the synthetic dataset shown above, The maxima are shown with black spheres. The grey edges of the extremum graph indicate adjacencies. The complete extremum graph captures pairwise relationships between all pairs of extrema based on the extremum graph. The dashed edges indicate correspondences between the extrema of the two data sets. The distance measure captures the cost required to merge unmatched extrema and the quality of the matched extrema in terms of their persistence.



(b) Comparison between two frames of the vortex data with a difference of 5 time steps between them. Extremum graph and morse decomposition based measures can capture correspondence in the presence of changes in shape and degree of overlap. For the extremum associated with the green isosurface, adjacencies are indicated by the gray edges. The extremum graph is proximity based and the similarity in the adjacency relationships is used to establish correspondence. Though some adjacencies are missing due to variations in the field, the complete extremum graph can capture these distortions and establish appropriate correspondences.



(c) Additional frames showing correspondence for the vortex data. Time steps 6-10 (column 1) and Time steps 11-15 (column 2).