# Summary

IMT2021014 Karan Raj Pandey IMT2021104 Prasad Jore

**Topic:** Output-Sensitive Construction of Reeb Graphs

A Reeb graph is a mathematical structure that represents the topological properties of level sets in a scalar function. It provides a compact and simplified representation of the function's topology by capturing the connectivity and changes in the topology as the function value varies. The Reeb graph is constructed by identifying critical points in the scalar function, which are points where the gradient is zero. These critical points correspond to locations where the level sets of the function intersect or change their topology.

In a Reeb graph, the critical points are represented as nodes, and the edges capture the transitions between different critical points. The edges connect critical points that correspond to adjacent level sets or connected components of the function. The main idea behind the Reeb graph is to collapse each connected component of a level set into a single node, representing the topology of that level set. This collapsing process simplifies the representation of the function's topology and provides an abstract overview of how the level sets are connected

The Doraiswamy and Natarajan paper describes a near-optimal output-sensitive algorithm for computing the Reeb graph of scalar functions defined over manifolds or non-manifolds in any dimension. Key to the simplicity and efficiency of the algorithm is an alternate definition of the Reeb graph that considers equivalence classes of level sets instead of individual level sets. The algorithm works in **two steps**. The first step **locates all critical points** of the function in the domain. Critical points correspond to nodes in the Reeb graph. **Arcs connecting the nodes are computed** in the second step by a simple search procedure that works on a small subset of the domain that corresponds to a pair of critical points. The algorithm stores the connected components of a level sets using dynamic connectivity data structures

The paper also describes a scheme for controlled simplification of the Reeb graph and two different graph layout schemes that help in the effective presentation of Reeb graphs for visual analysis of scalar fields. Finally, the Reeb graph is employed in four different applications – surface segmentation, spatially-aware transfer function design, visualization of interval volumes, and interactive exploration of time-varying data.

The algorithm's time complexity is $O(n + l + t \log t)$, where n is the number of triangles in the input mesh, t is the number of critical points of the function, and l is the total size (number of edges) of all critical level sets.

This algorithm possesses several desirable characteristics. Firstly, it is output-sensitive, meaning its running time depends on the number of critical points and the size of critical level sets, which reflects the significance of features in the data. Secondly, it is nearly optimal, as the size of critical level sets is typically $O(n)$ in practical scenarios, resulting in a running time close to the lower bound of $W(n + t \log t)$. Thirdly, it is applicable to a wide range of scenarios, including functions defined on d-manifolds and non-manifolds, as well as data sampled on both unstructured and structured grids. Lastly, the algorithm is straightforward to implement, involving a sorting operation followed by a series of tree search operations.

This algorithm efficiently computes the Reeb graph for piecewise-linear functions, considering the number of critical points and the size of critical level sets. It is versatile, simple to implement, and achieves near-optimal performance, making it suitable for various applications.