# Parallel Computation Of 2D Morse-Smale Complexes

Nithin Shivashankar, Senthilnathan Maadaswamy and Vijay Natarajan

Department Of Computer Science And Automation, IISc, Bangalore

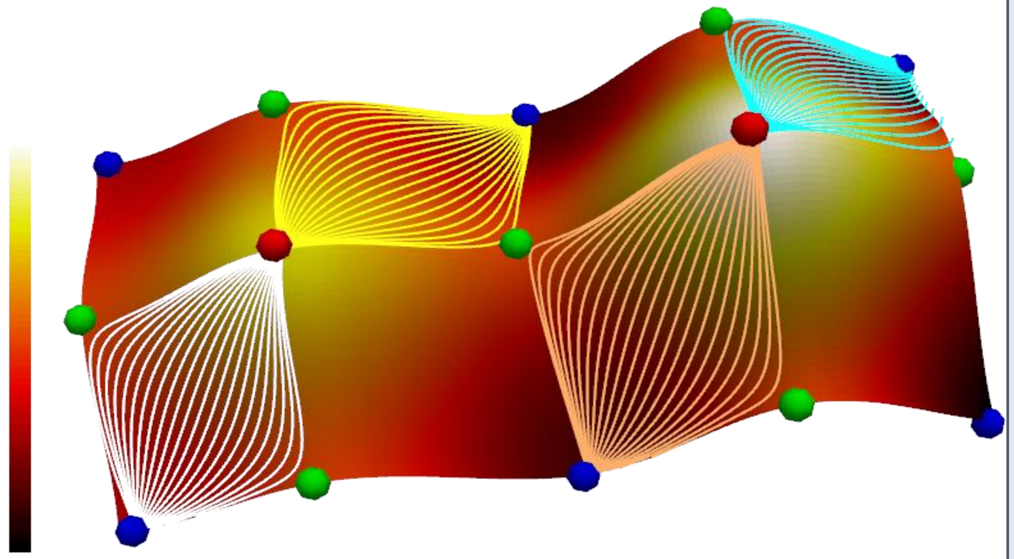
## Objective

Parallel computation of the Morse-Smale complex designed for multi-core and GPU environments.

Computation for datasets that don't fit in Memory.

## The Morse-Smale Complex

**Input:** Scalar (real valued) function Example: 2D plane→ sin(x)+sin(y).



**Gradient curves:** Curves that trace the direction of steepest descent

**Critical Points:** Points of origin/destination of gradient curves

**Morse-Smale complex:** Partition based on origin/destination of gradient curves.

### Reference

Shivashankar, N. Maadaswamy, S. and Natarajan, V., Parallel Computation of 2D Morse-Smale Complexes, *IEEE Transactions on Visualization and Computer Graphics (To Appear).*

### Contact and Acknowledgements

nithin@csa.iisc.ernet.in

senthilnathan.m@intel.com

vijayn@csa.iisc.ernet.in

Work supported by Intel and DST.

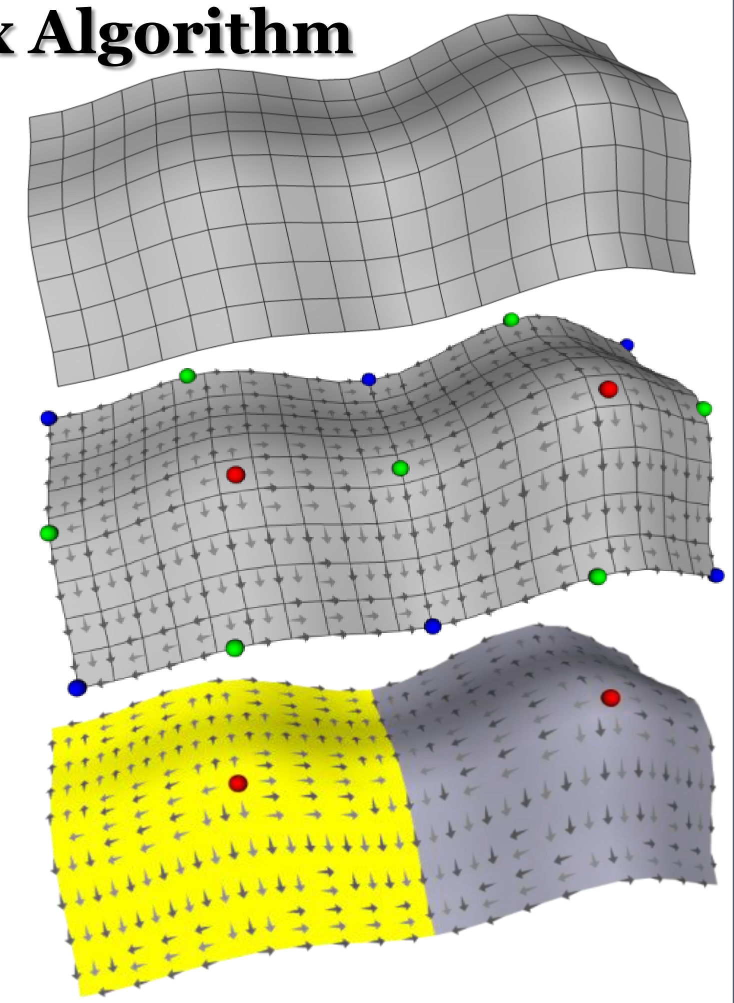
## Parallel Morse-Smale(MS) Complex Algorithm

**INPUT:** Domain as a Cell complex and a scalar function sampled at vertices.

1.Compute discrete gradient

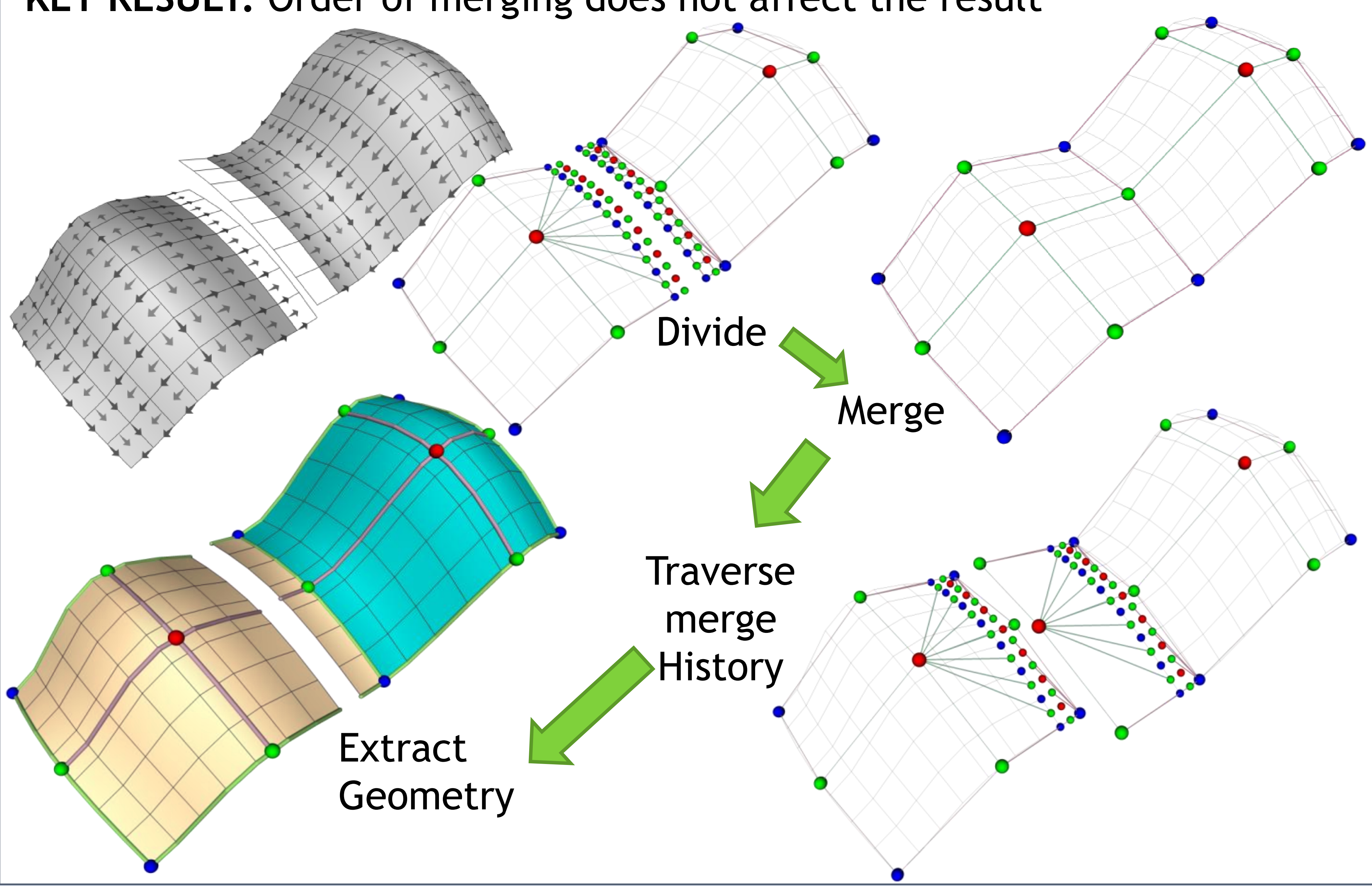
**KEY RESULT**: Computing discrete gradient is independent of order. Can be done in parallel.

2. Traverse the gradient field to compute the MS complex.



## Large Data: Divide and conquer.

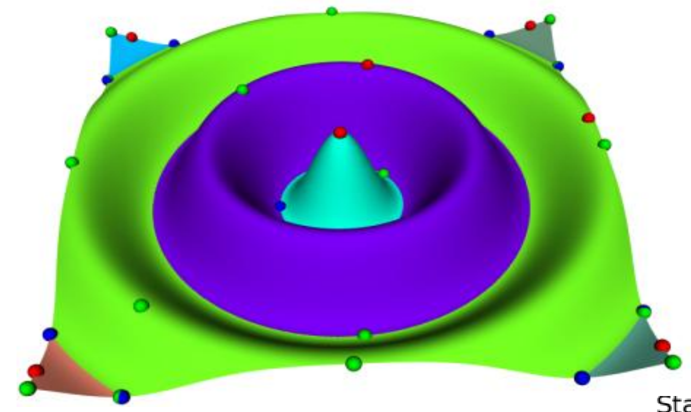
**KEY RESULT:** Order of merging does not affect the result




## Evaluation

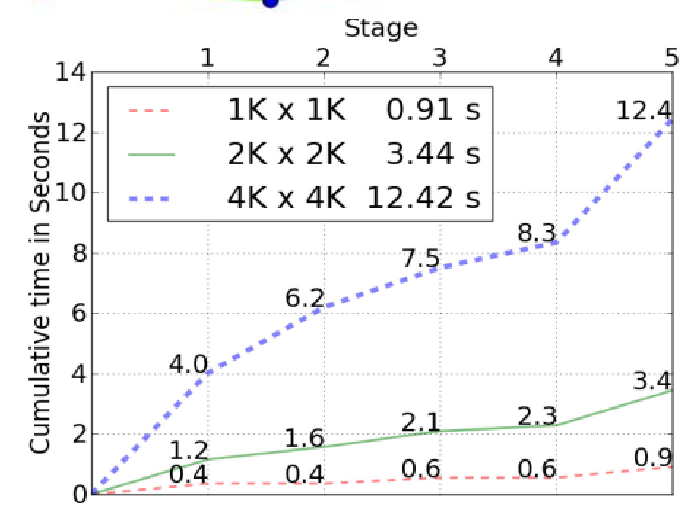
Experiments with synthetic and real world datasets on 2D grids.

Observed near linear scaling with data size and number of cores.

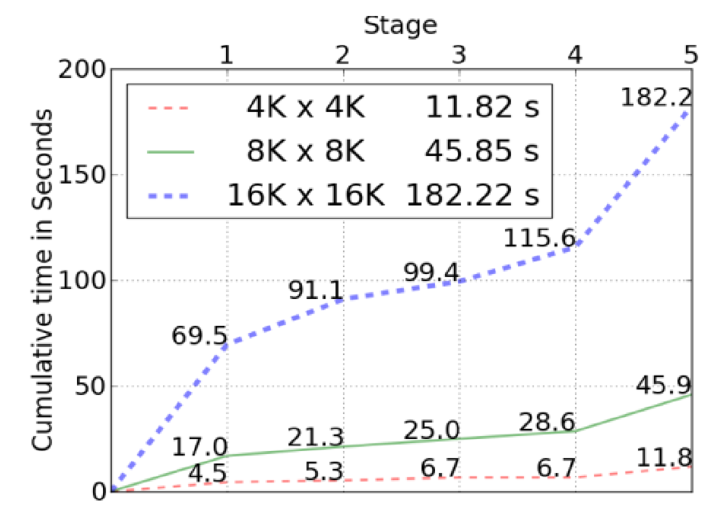


**WGAUSS dataset**

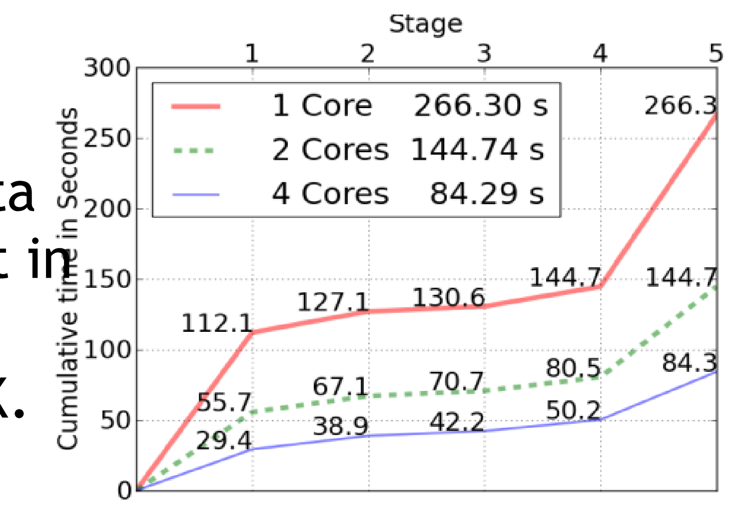
Data fits in CPU but not GPU




Data fits neither in CPU nor GPU.




Computed in CPU. Data does not fit in CPU. Size: 8Kx8K.

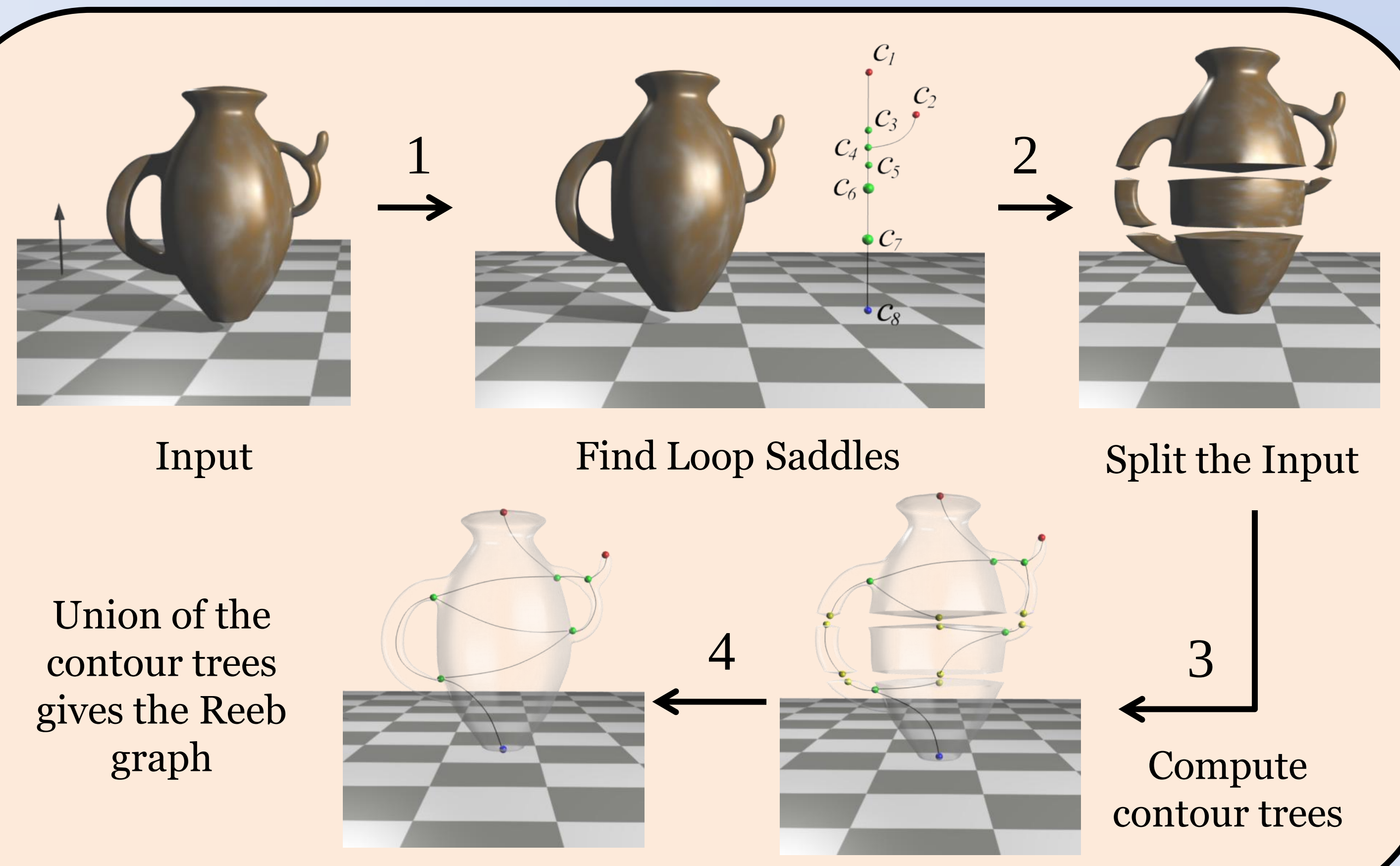


# Computing Reeb Graphs as a Union of Contour Trees

Harish Doraiswamy Vijay Natarajan
Visualization and Graphics Lab, Indian Institute of Science, Bangalore
http://vgl.serc.iisc.ernet.in

The Reeb graph of a scalar function is obtained by mapping each connected component of its level sets to a point

**Applications**
- Topology based shape matching. *Hilaga et al. SIGGRAPH 2001*
- Transfer function design. *Weber et al. TVCG 2007*
- User Interface. *Bajaj et al. Vis 1997*



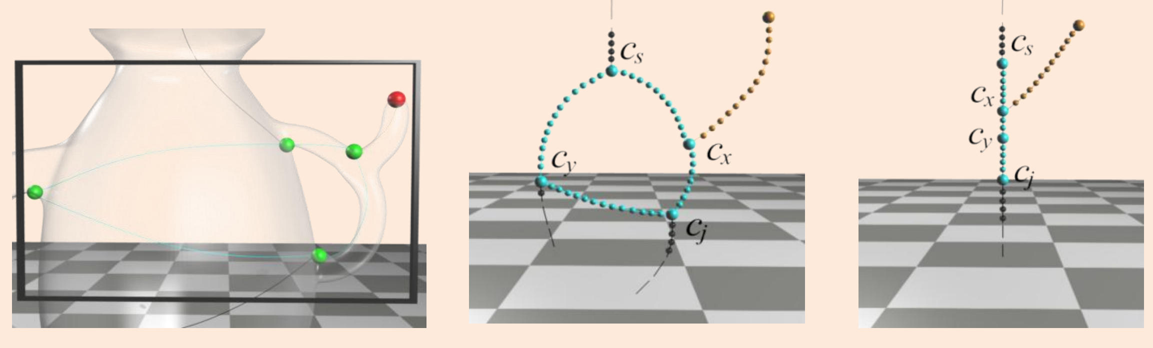

## The ReCon Algorithm

1. Identify the loop saddles of the input
2. Split the input at a function value infinitesimally above that of the loop saddles to obtain a set of simply connected interval volumes
3. Compute the contour trees for each interval volume
4. Construct the Reeb graph by computing the union of these contour trees

## Identifying Loop Saddles

***Lemma***

*Let $G_R$ be the Reeb graph of the given input scalar function **f**. Consider the join tree $T_J$ of **f**. Any join saddle that ends a loop in $G_R$ appears as a degree-2 node in $T_J$.*



Loop in the input Augmented Reeb graph Join Tree

## Properties

1. Efficient: has a running time of $O(n \log n + sn)$
   - $n$ - # triangles
   - $s$ - # saddles
2. Generic: works without any modifications on $d$-manifolds and non-manifolds
3. Easy to implement
4. Handles data that do not fit in memory
5. At least an order of magnitude faster than existing generic algorithms

## Performance for 3D Input

| 3D Model | # Triangles | # Loops | Time taken (sec) | | | |
|---|---|---|---|---|---|---|
| | | | ReCon | LS | OS | Rand |
| Skull | 0.34M | 2 | 0.1 | 0.3 | 3.4 | 2.2 |
| Post | 1.24M | 0 | 0.4 | 0.9 | 13.0 | 14.5 |
| Plasma | 2.64M | 0 | 1.5 | 1.9 | 396.3 | 132.6 |
| SF Earthquake | 4.19M | 0 | 2.4 | 2.8 | 598.1 | 166.9 |

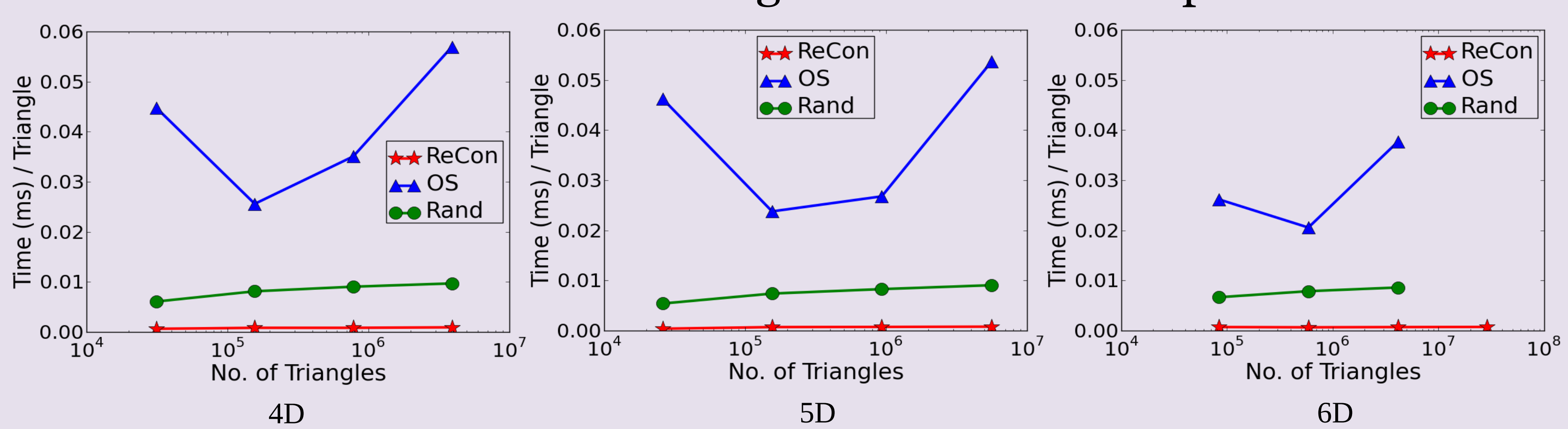
## Performance for 2D Input

| 2D Model | # Triangles | # Loops | Time taken (sec) | | | |
|---|---|---|---|---|---|---|
| | | | ReCon | Online | OS | Rand |
| Awakening | 4M | 1643 | 2.4 | 6.72 | 41.4 | 51.8 |
| Day | 6M | 2161 | 4.3 | 10.3 | 91.3 | 67.5 |
| Dawn | 6.6M | 757 | 4.5 | 11.5 | 73.2 | 71.8 |
| Lucy | 28M | 15 | 34.2 | 60.1 | *Mem* | *Mem* |

## Scalability with increasing loop size

| Model | # Triangles | # Loops | Time taken (sec) |
|---|---|---|---|
| s4d-7 | 0.78M | 1 X $10^5$ | 0.7 |
| s5d-6 | 0.93M | 1 X $10^5$ | 0.7 |
| s4d-8 | 3.9M | 5.8 X $10^5$ | 3.6 |
| s5d-7 | 5.6M | 5.6 X $10^5$ | 4.4 |
| s6d-6 | 4.1M | 2.9 X $10^5$ | 2.9 |
| Lucy | 28.0M | 15 | 34.2 |
| s6d-7 | 28.8M | 2 X $10^6$ | 21.2 |

## Performance for higher dimensional input




Input – 4D, 5D, and 6D Sierpinski simplexes of various sizes

## Handling Large data

| Model | # Triangles | Function | Time taken | |
|---|---|---|---|---|
| | | | ReCon | Online |
| David | 56M | x | 3.6 m | 4.7 m |
| | | y | 3.8 m | 4.8 m |
| | | z | 3.2 m | 16.6 m |
| St. Matthew | 372M | x | 26.9 m | 40 m |
| | | y | 26.7 m | 4.2 hrs |
| | | z | 25.2 m | 41 m |
| Atlas | 507M | x | 41.5 m | * |
| | | y | 38.5 m | * |
| | | z | 42.6 m | * |

## References

- Pascucci et al. *ACM Trans. Graph 2007* (Online)
- Doraiswamy et al. *IEEE TVCG 2011* (OS)
- Tierny et al. *IEEE TVCG 2009* (LS)
- Harvey et al. *SCG 2010* (Rand)

## Contact

Harish Doraiswamy: harishd@csa.iisc.ernet.in
Vijay Natarajan: vijayn@csa.iisc.ernet.in

## Acknowledgements

Harish Doraiswamy was supported by Microsoft Corporation and Microsoft Research India under the Microsoft Research India PhD Fellowship Award. This work was supported by the Department of Science and Technology, India, under Grant SR/S3/EECE/048/2007

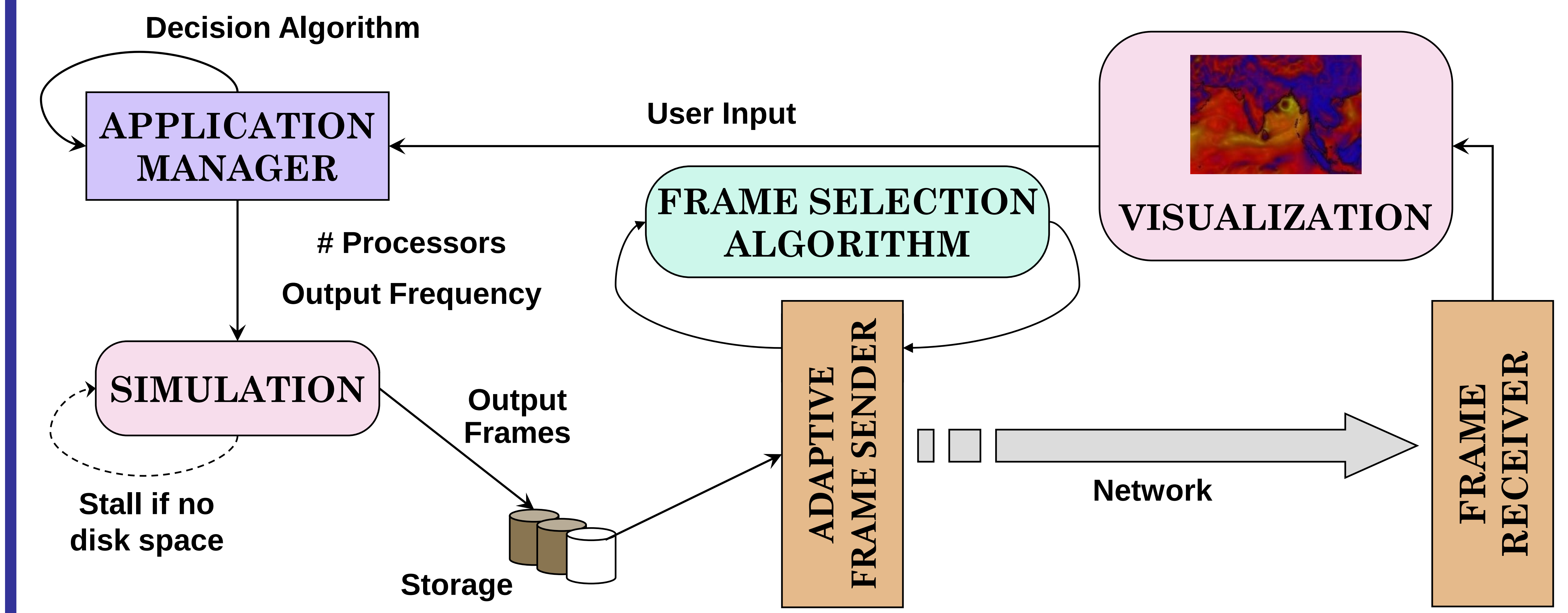
# Efficient Online Visualization and Simulations for Large-scale Applications

## Problem Statement

Develop an adaptive integrated steering framework that

- allows simultaneous simulation and online visualization
- spawns high-resolution simulation dynamically over desired region-of-interest
- supports optimal processor allocation for simulation
- supports optimal frequency of output for visualization

## Framework for Adaptive Simulation and Online Visualization




## Simulation-Visualization Lag

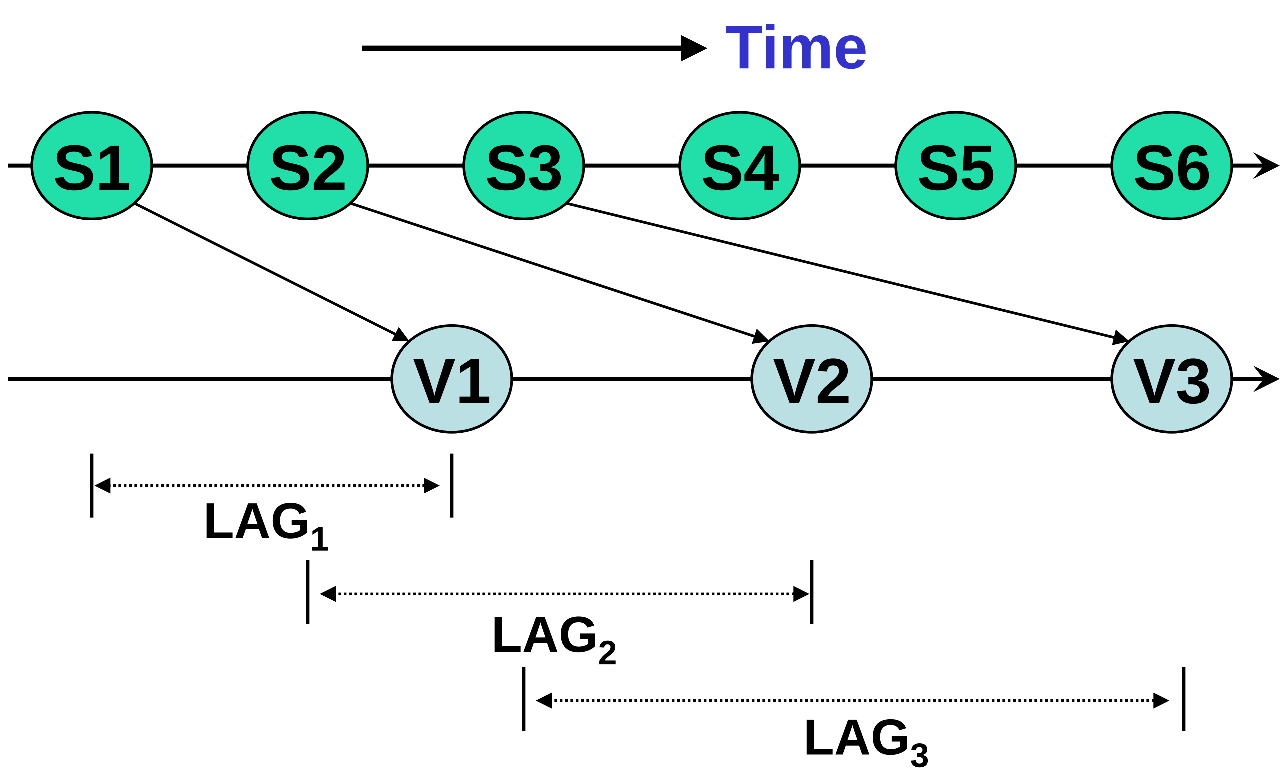
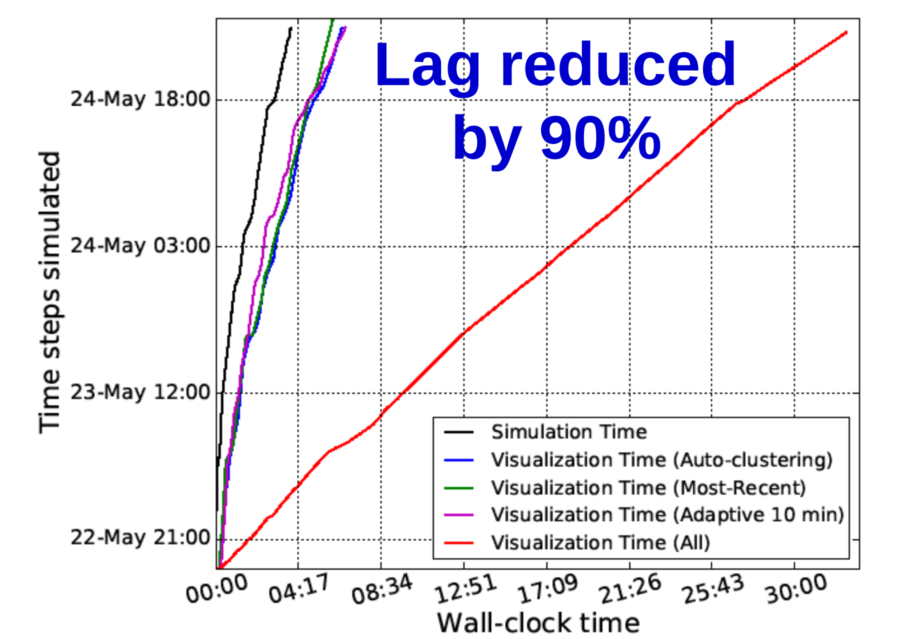



Illustration of increasing visualization lag

## Frame Selection Algorithms

- Most-recent
  - Transfer the most-recently simulated frame
- Auto-clustering
  - Modified k-means for temporal clustering to select the most-representative frames
- Adaptive
  - Transfer full or reduced frames within acceptable lag bound

## Results




Preeti Malakar (preeti@csa.iisc.ernet.in)
Vijay Natarajan (vijayn@csa.iisc.ernet.in)
Sathish S. Vadhiyar (vss@serc.iisc.ernet.in)

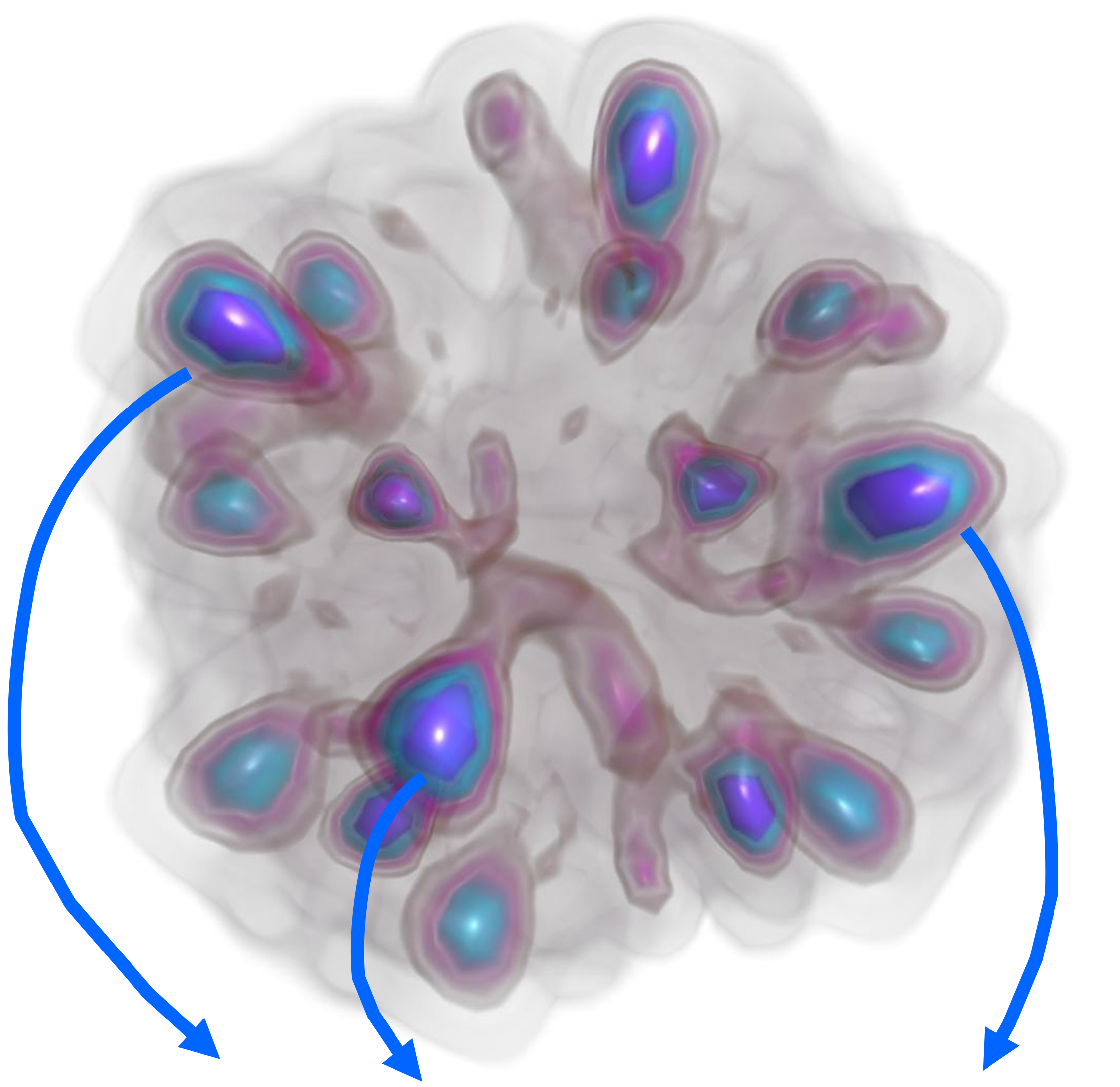
http://garl.serc.iisc.ernet.in/
http://vgl.serc.iisc.ernet.in/

Visualization and Graphics Lab
Indian Institute of Science, Bangalore
http://vgl.serc.iisc.ernet.in

# Symmetry in Scalar Field Topology

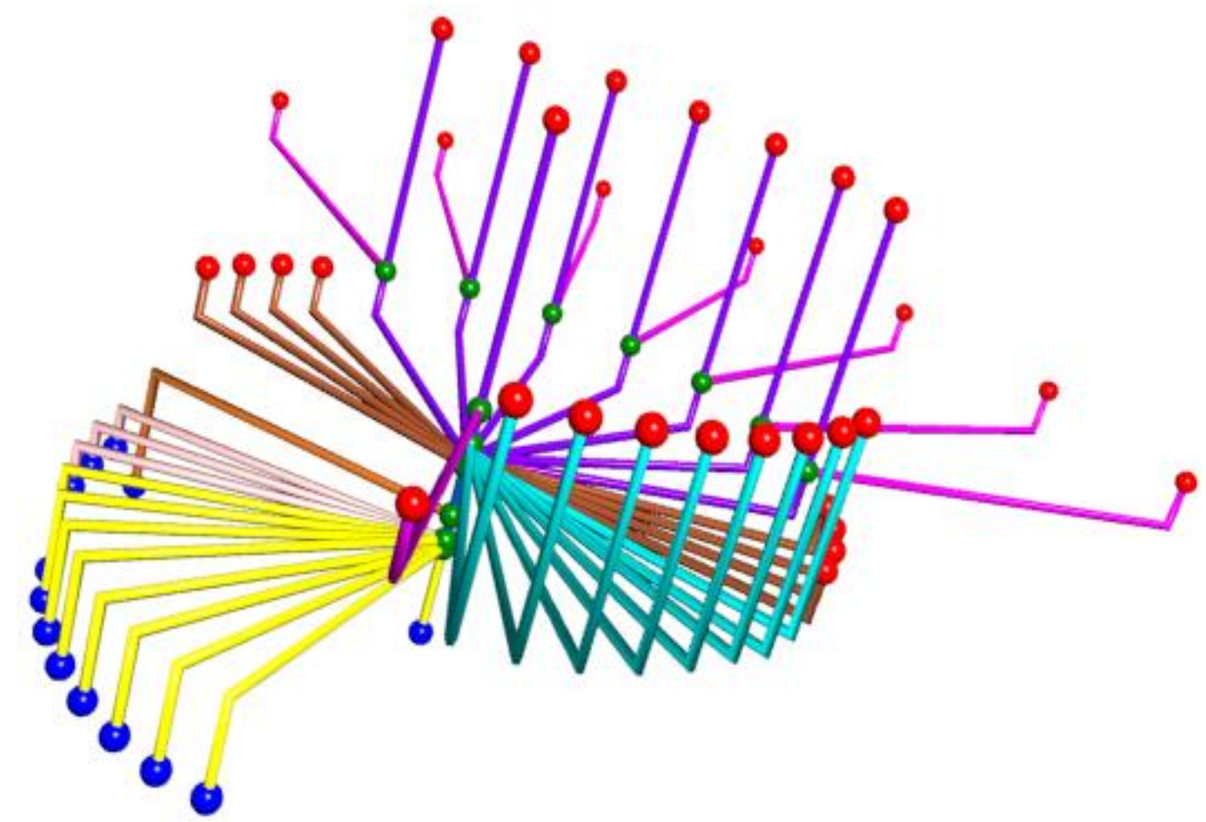
Dilip Thomas
dilip@csa.iisc.ernet.in

Vijay Natarajan
vijayn@csa.iisc.ernet.in



Symmetric structures in EM image of RuBisCO molecule

Four types of symmetric regions identified



Classify subtrees of the contour tree

## Motivation

Scalar fields contain repeating patterns

Provide insights on scientific phenomena

## Technique

Classify subtrees of the contour tree

Similarity measure to compare subtrees

## Applications

Symmetry-aware isosurface extraction

Symmetry-aware transfer function design