# Computing Contour Tree for Piecewise Polynomial Functions

Girijanandan Nucha and Vijay Natarajan
Contact : [girijanandan.nucha , vijayn]@csa.iisc.ernet.in
Visualization and Graphics Lab
Department of Computer Science and Automation , IISc , Bangalore

## Problem Statement

Design an algorithm to accurately and efficiently compute the contour tree of a scalar field defined on vertices of a triangle mesh with polynomial interpolants used to extend it within the mesh elements

## Motivation

Contour tree applications :

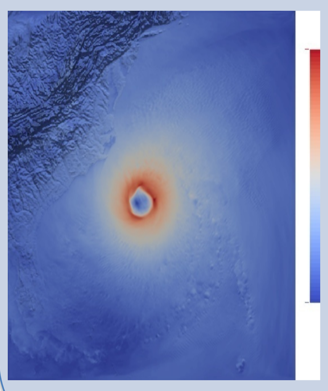
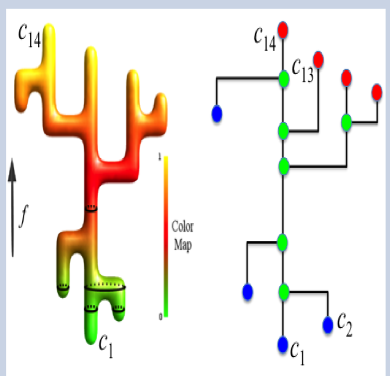
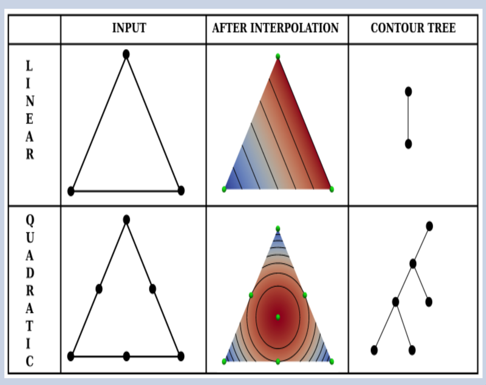
- Computation and exploration of isosurfaces
- Symmetry and similarity detection
- Feature extraction and tracking

Higher order interpolants :

- Suited for modeling smooth functions
- Use of fewer elements

## Definitions

- *Scalar field :* A function that maps a point in a domain to a real value
- *Critical point :* A point where gradient of the scalar field becomes zero
- *Contour tree :* A topological structure which captures changes in the level set of a scalar field
- *Higher order interpolant :* A polynomial function of degree more than one

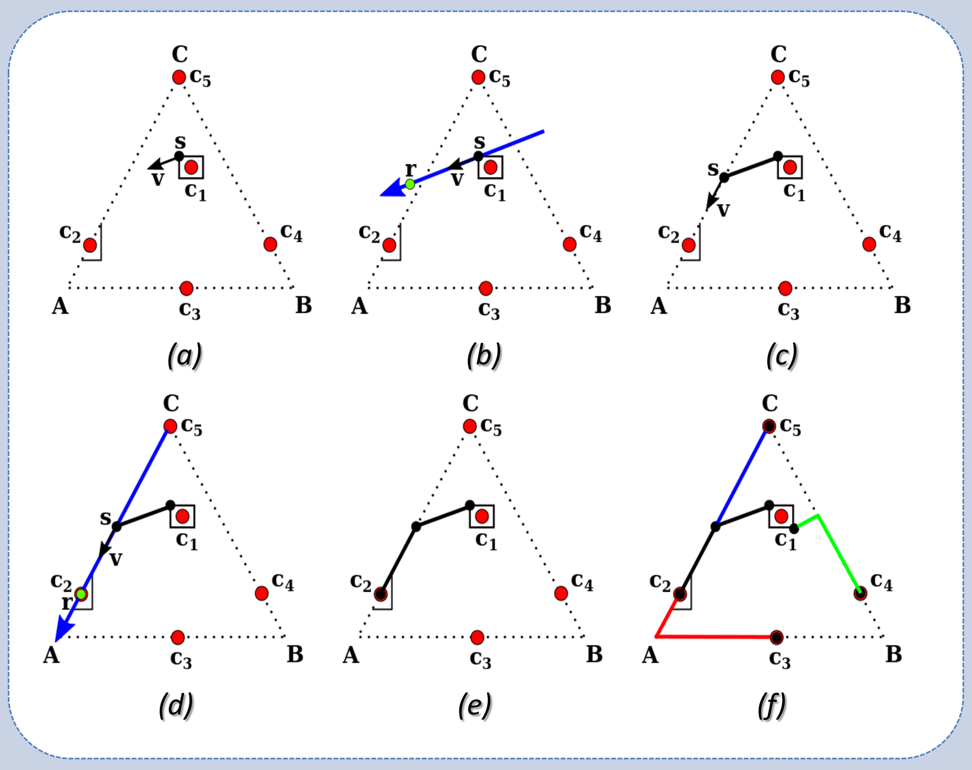
## Challenges

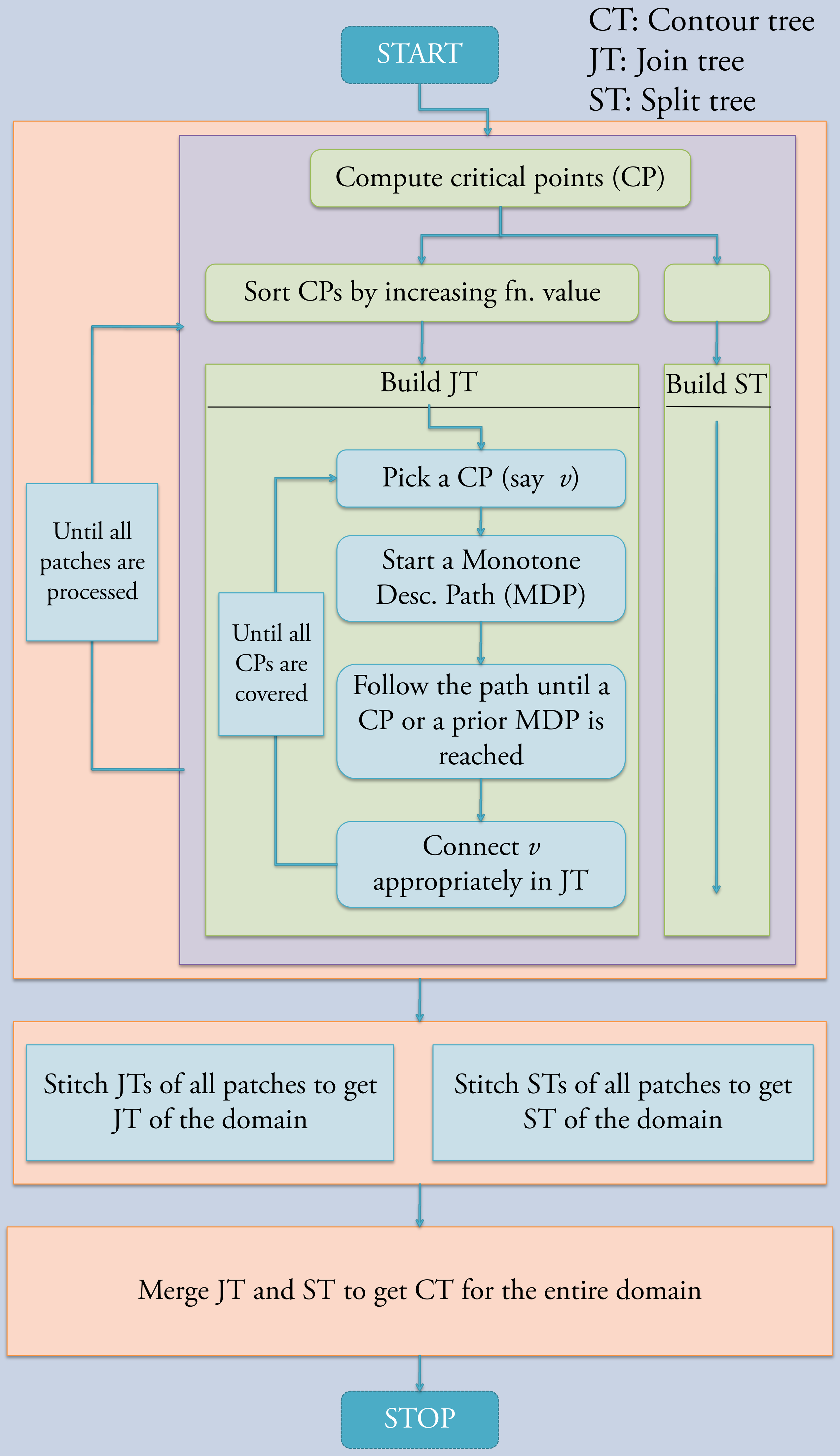
| | Linear | Higher Order |
|---|---|---|
| **Critical point computation** | Simple. CPs are located at mesh vertices | Difficult. CPs may lie inside a mesh element |
| **Neighbourhood** | Clearly defined | Needs further analysis |
| **Monotone path tracing** | Comparison based | Requires finding roots of univariate polynomials |

## References


- H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. CGTA 2003.
- Y.-J. Chiang, T. Lenz, X. Lu, and G. Rote. Simple and optimal output sensitive construction of contour trees using monotone paths. CGTA 2005.
- S. Dillard, V. Natarajan, G.Weber, V. Pascucci, B.Hamann. Topology guided tessellation of quadratic elements, ISAAC 2006
- A. Acharya, V. Natarajan. A parallel and memory efficient algorithm for constructing the contour tree, PacificVis, 2015


## Contour Tree Computation Algorithm

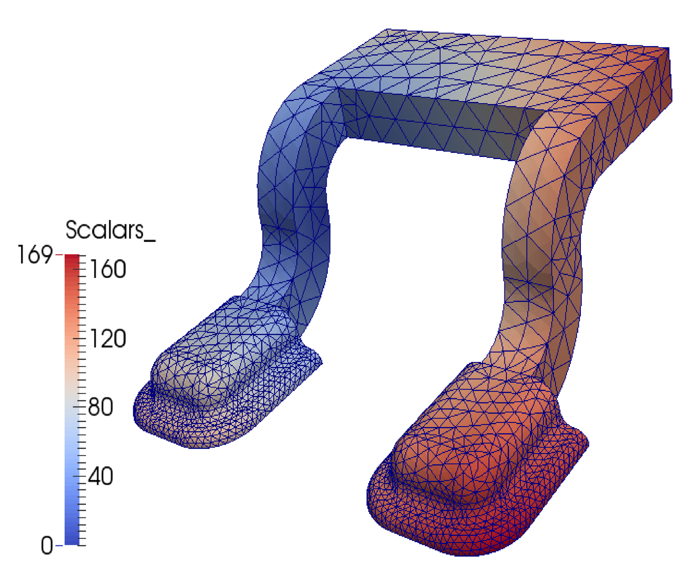


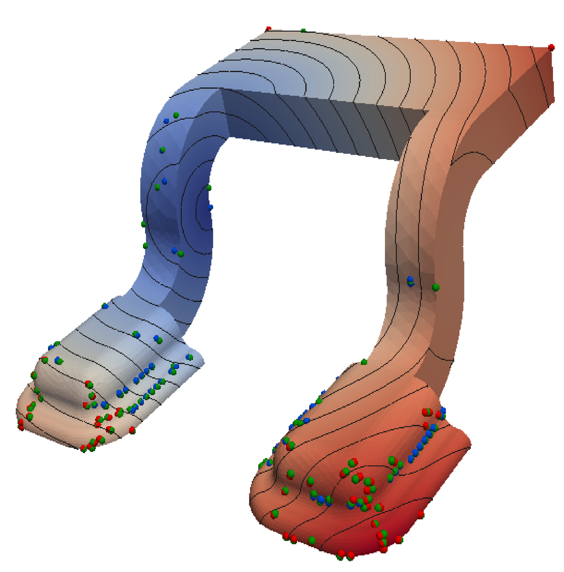

### Properties

- The algorithm computes contour tree for 2D scalar fields with any degree of interpolation. Earlier methods were applicable to linear and quadratic interpolants.
- The algorithm is intrinsically parallelizable.
- Running time for CT computation of a single patch is $O(c^2d^2)$. Where $c$ is number of critical points in the patch and $d$ is degree of the interpolant.

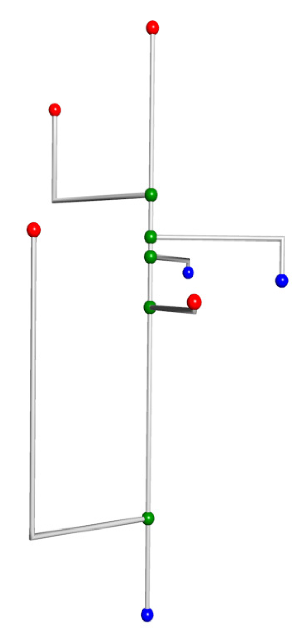
## Results



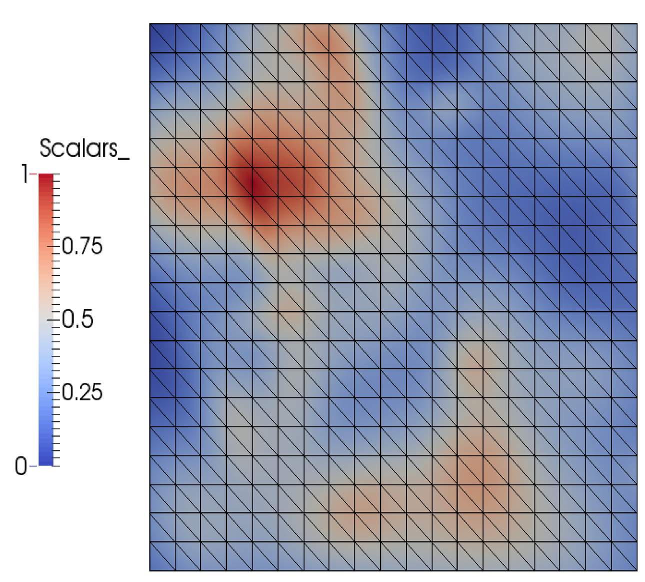
A simple scalar field on a thermal conductor



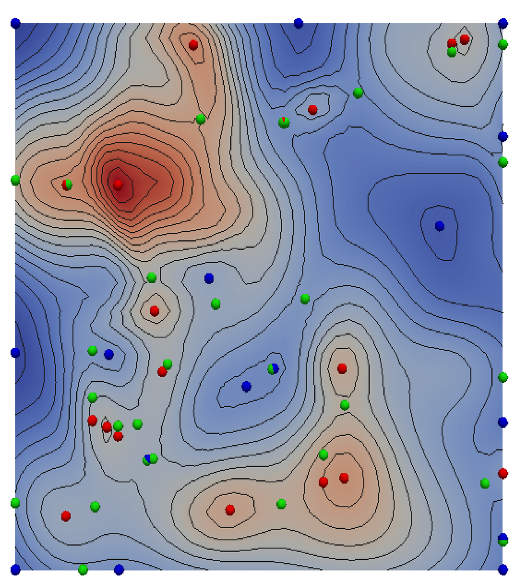
Critical points and contour lines



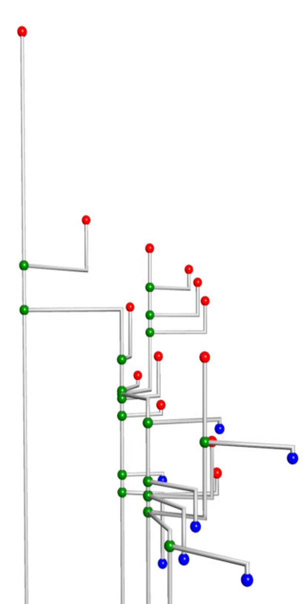
Contour Tree



A sum of Gaussian field defined on a unit square



Critical points and contour lines



Contour Tree

## Future Work

- Parallelize the implementation of the algorithm
- Extend the algorithm to 3D meshes
- Explore applications of the contour tree