Algorithms for Visualization: Constructing Contour Trees [1]

Raghavendra G S

PhD Student,
Visualization and Graphics Lab,
CSA, IISc.

July 6th, 2017

[1] Hamish Carr, Jack Snoeyink, and Ulrike Axen. "Computing contour trees in all dimensions." Computational Geometry 24.2 (2003): 75-94.

Introduction

Scientific Visualization

[2] http://www.cc.gatech.edu/scivis/tutorial/linked/whatisscivis.html

Scientific Visualization

Scientific Visualization (SciVis)[2] is the representation of data graphically as a means of gaining understanding and insight into the scientific data.

[2] http://www.cc.gatech.edu/scivis/tutorial/linked/whatisscivis.html

Why algorithms for SciVis?

Why algorithms for SciVis?

- Automate the process/minimize human intervention.

Why algorithms for SciVis?

- Automate the process/minimize human intervention.
- Spatio-temporal nature of data.

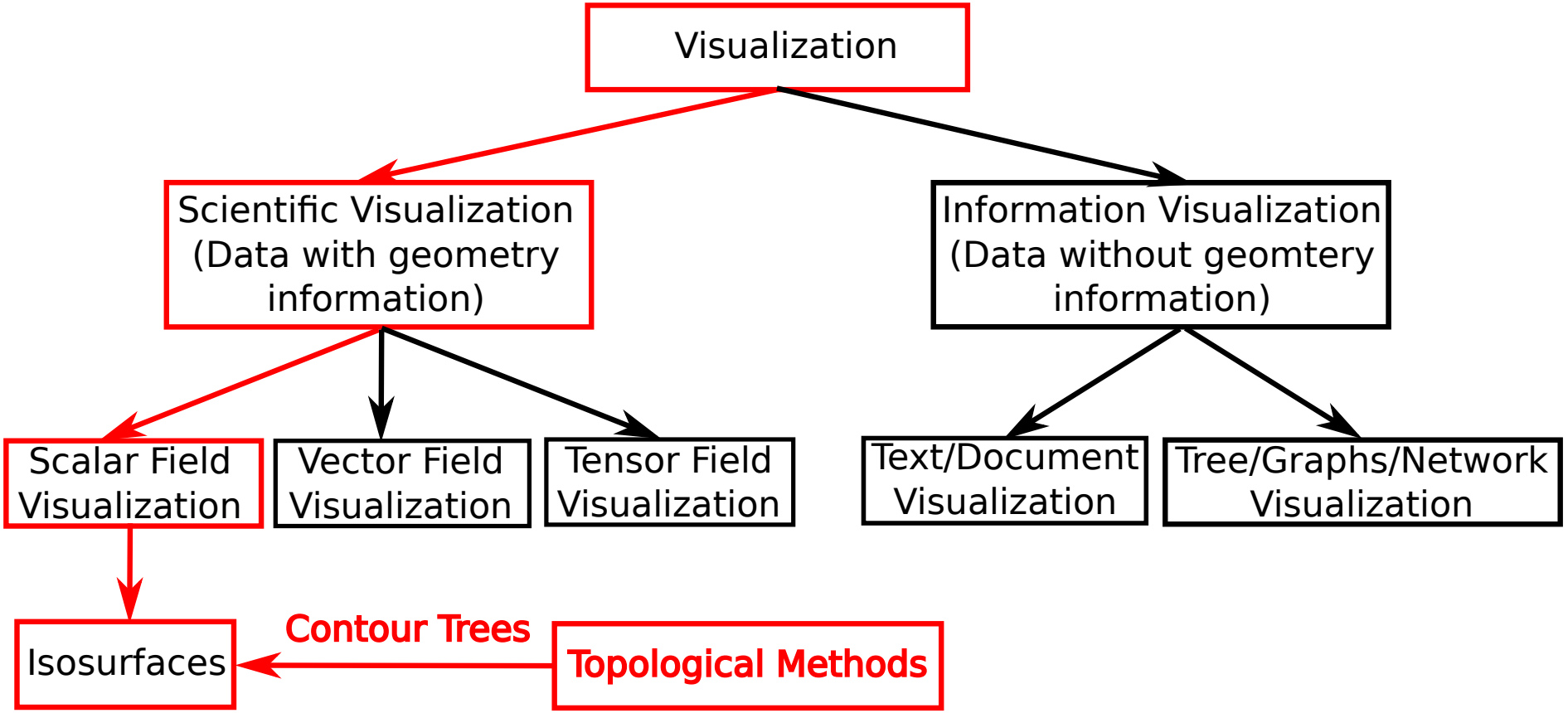
Why algorithms for SciVis?

- Automate the process/minimize human intervention.
- Spatio-temporal nature of data.
  - Huge data size.

Why algorithms for SciVis?

- Automate the process/minimize human intervention.
- Spatio-temporal nature of data.
  - Huge data size.
- Real-time visualization.

Today's Agenda



Background

Some background

- Scalar fields.
  - $f : \mathbb{R}^{2,3} \longrightarrow \mathbb{R}$
- Isovalue.
  - Real value v in the range.
- Level sets.
  - All points $p \in \mathbb{R}^{2,3}$ s.t. $f(p) = v$
  - Can have many components.
- Isosurfaces.
  - When the domain is $\mathbb{R}^3$
- Topological Structures.
  - Stick figures.
  - Succinct representation of the functions.

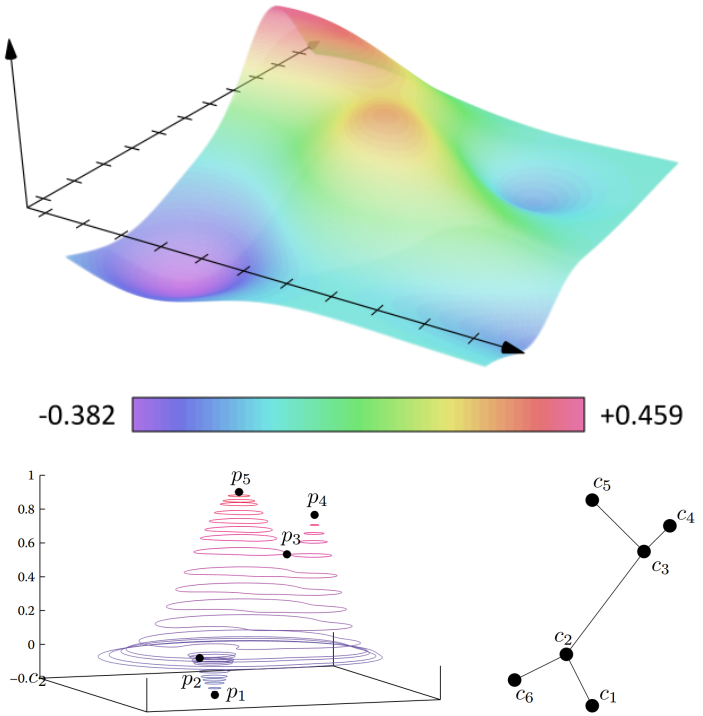


Figure: **Top** Scalar field defined on $\mathbb{R}^2$ [a] **Middle** Isocontours and contour tree[b]

[a] from Wikipedia

[b] from Rephael Wagner, "Isosurfaces: geometry, topology, and algorithms", 2013.

The data

- Continuity?
- Data representation.
- Dividing cubes into tetrahedra.
- Triangulation.
- Linear interpolation.

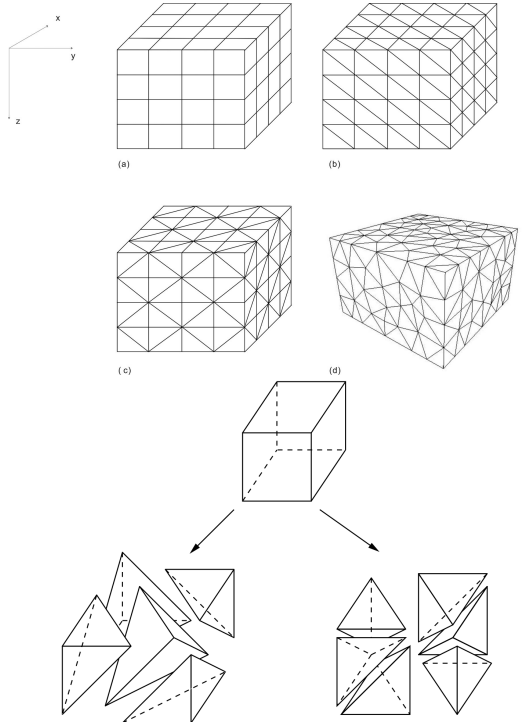


Figure: **Top** Example mesh/grid [a] **Bottom** Dividing a cube[b]

[a] from Wang et al. "Three-dimensional DC anisotropic resistivity modelling using finite elements on unstructured grids". Geophys J Int 2013.

[b] from Rephael Wagner

More about Isosurfaces

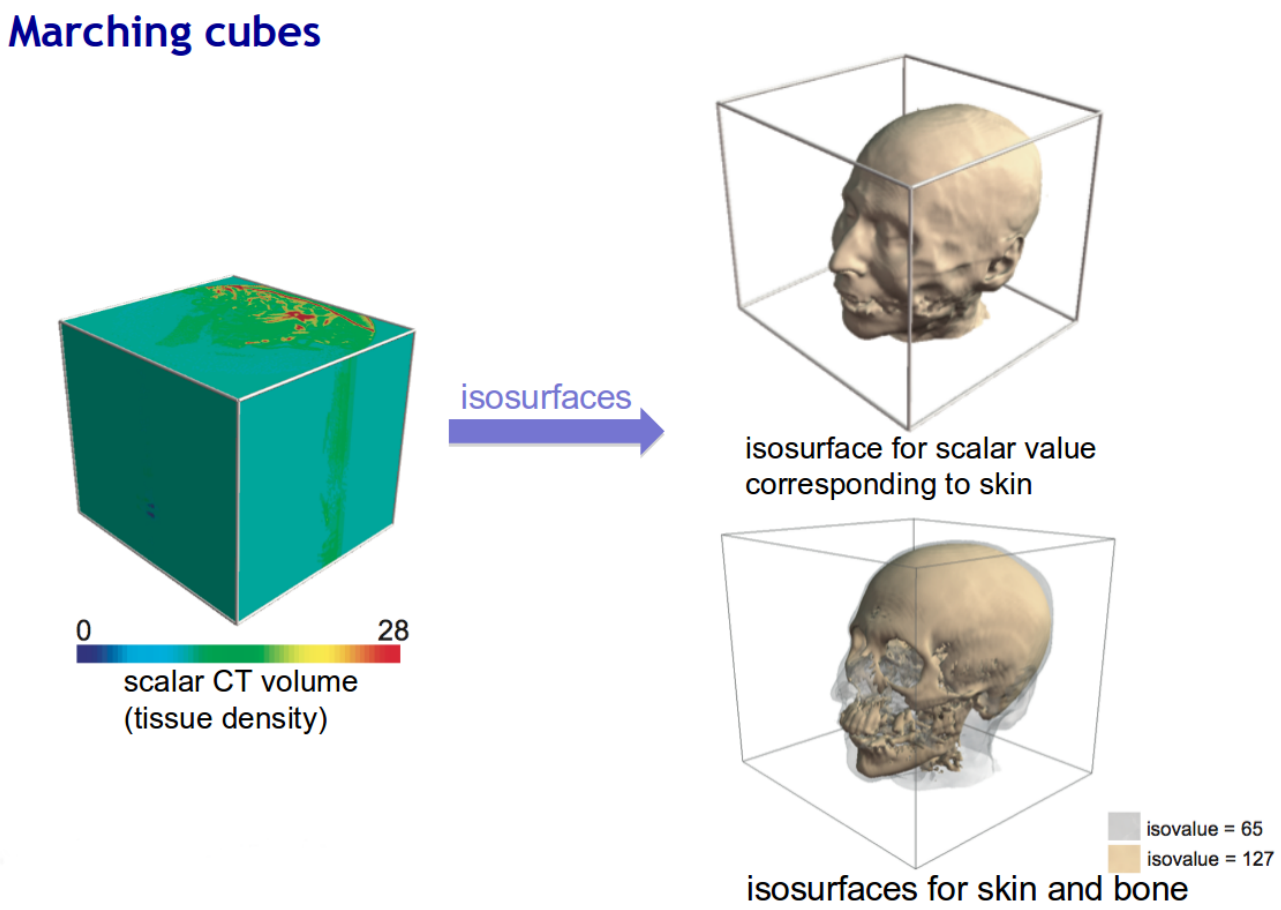


Figure: **Left:** Data set. **Right:**Isosurfaces.[3]

[3] from Alexandru Telea, "Data visualization: principles and practice". CRC Press, 2014.

More about Isosurfaces

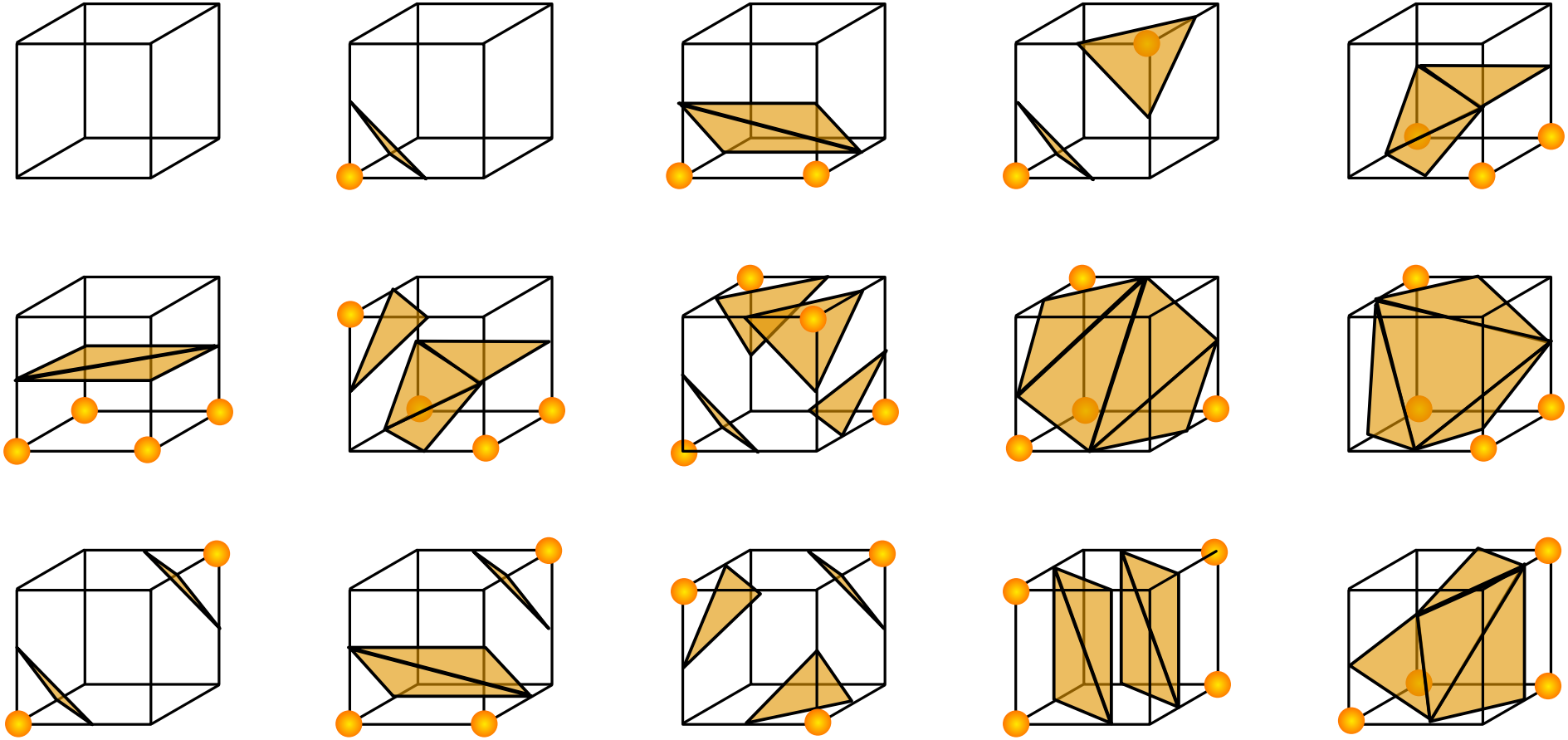
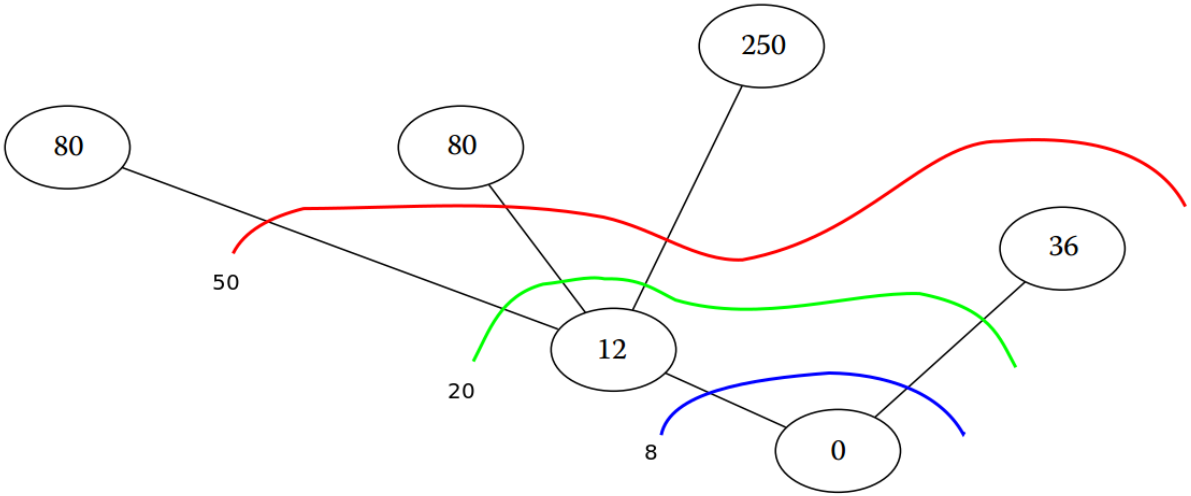


Figure: Marching cubes lookup table.[4]

[4] from Wikipedia

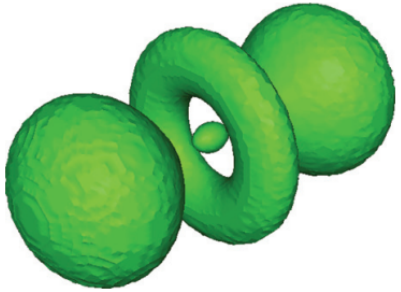
Contour trees

Problems/Why contour trees?

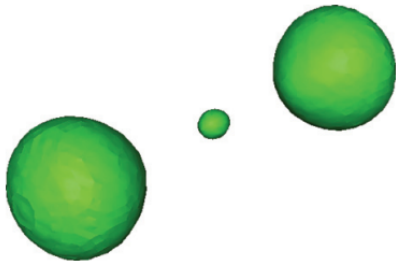


(a) Contour tree.

(b) Isovalue 8.



(c) Isovalue 20.



(c) Isovalue 50.

Figure: Need for contour tree.[5]

[5] from Rephael Wagner, "Isosurfaces: geometry, topology, and algorithms", 2013.

What are contour trees?

We can see how the level sets evolve by changing the isovalues

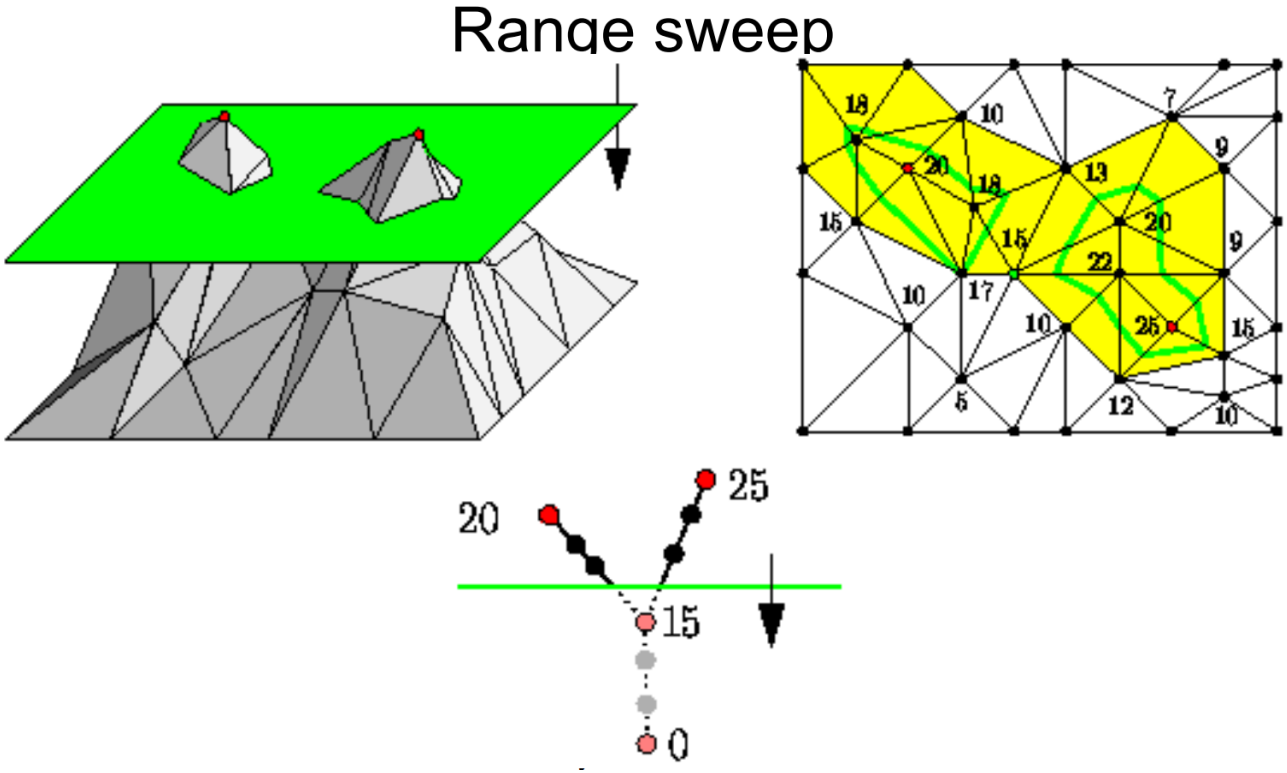


Figure: Evolution of level sets[6]

[6] from J.Snoeyink,C Bajaj

What are contour trees?

What are contour trees?

Contour tree tracks the changes in the connectivity of level sets of the function.

What are contour trees?

Contour tree tracks the changes in the connectivity of level sets of the function.

- It has set of nodes where connectivity of level sets change.

What are contour trees?

Contour tree tracks the changes in the connectivity of level sets of the function.

- It has set of nodes where connectivity of level sets change.
- It has set of edges which connects these nodes in a meaningful manner.

What are contour trees?

Running Example

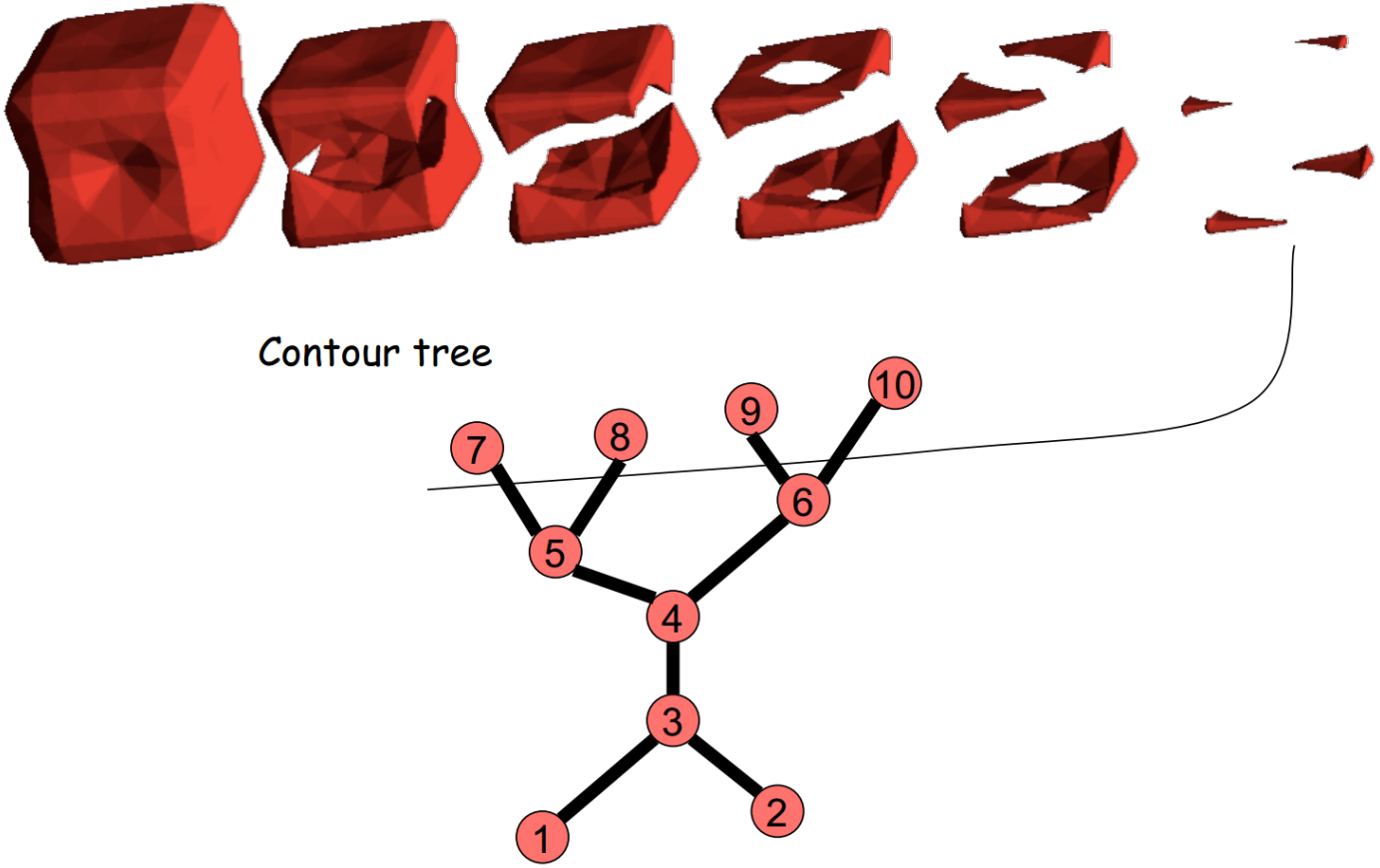


Figure: Evolution of level sets[7]

[7] from Hamish Carr et al.

What are contour trees?

Running Example

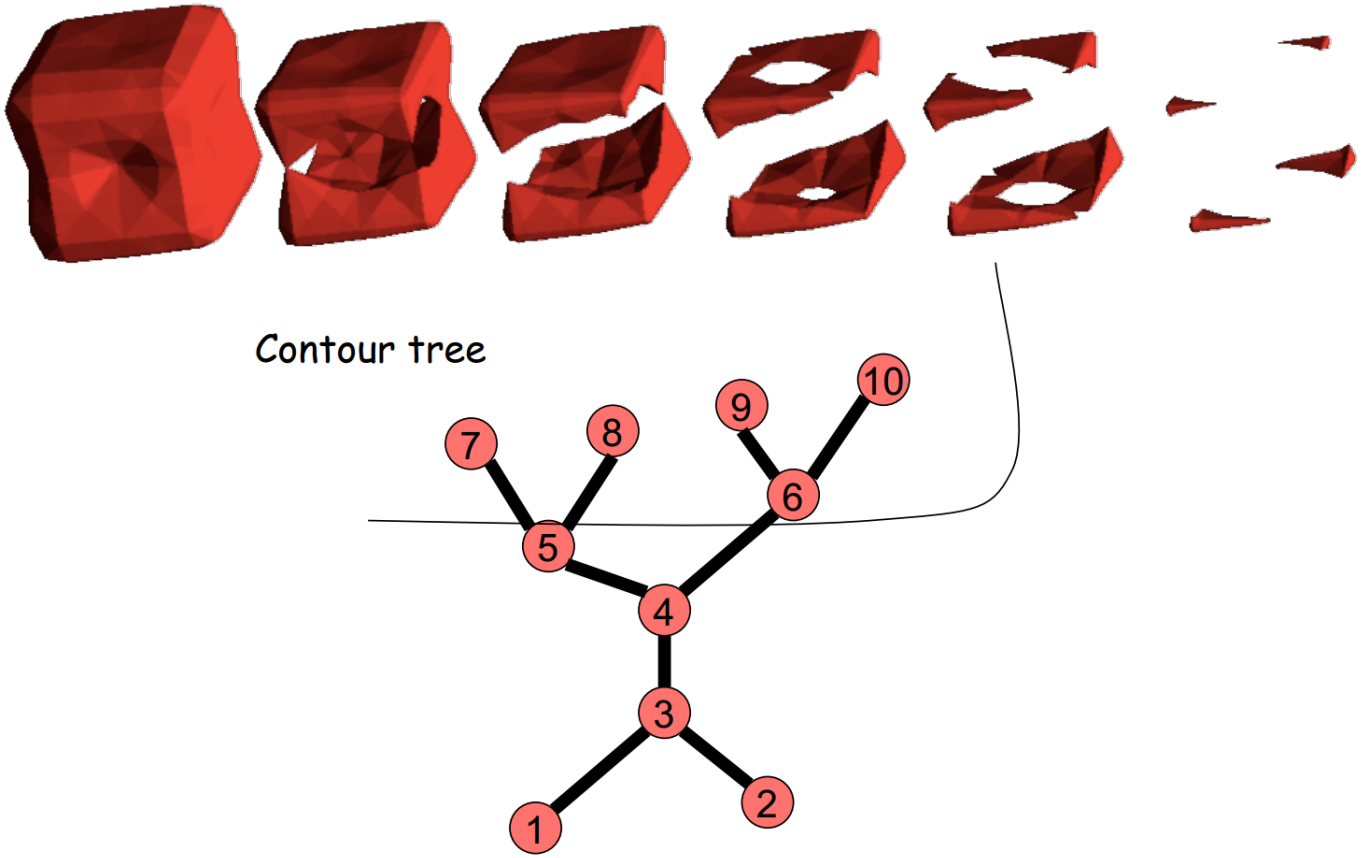


Figure: Evolution of level sets[8]

[8] from Hamish Carr et al.

What are contour trees?

Running Example

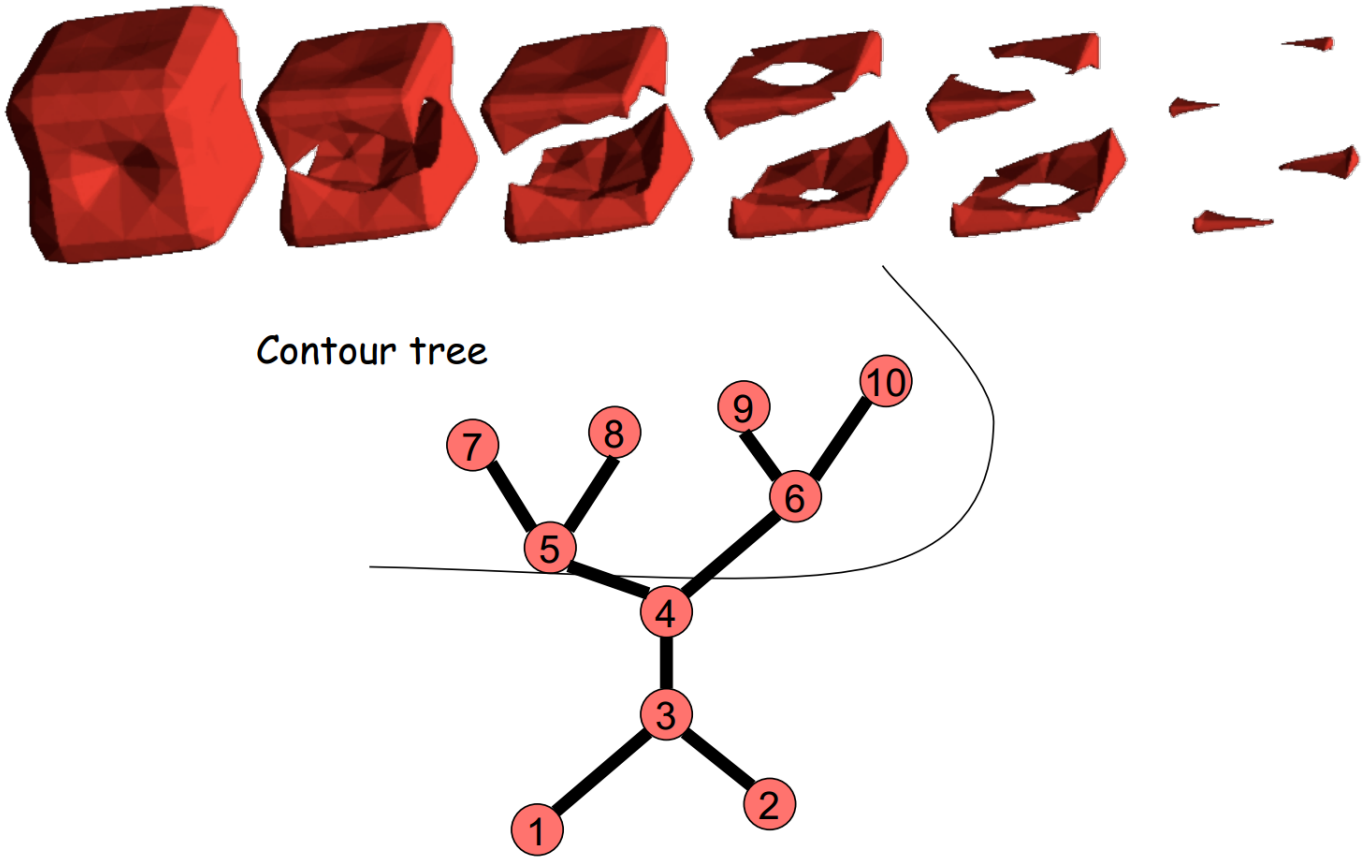


Figure: Evolution of level sets[9]

[9] from Hamish Carr et al.

What are contour trees?

Running Example

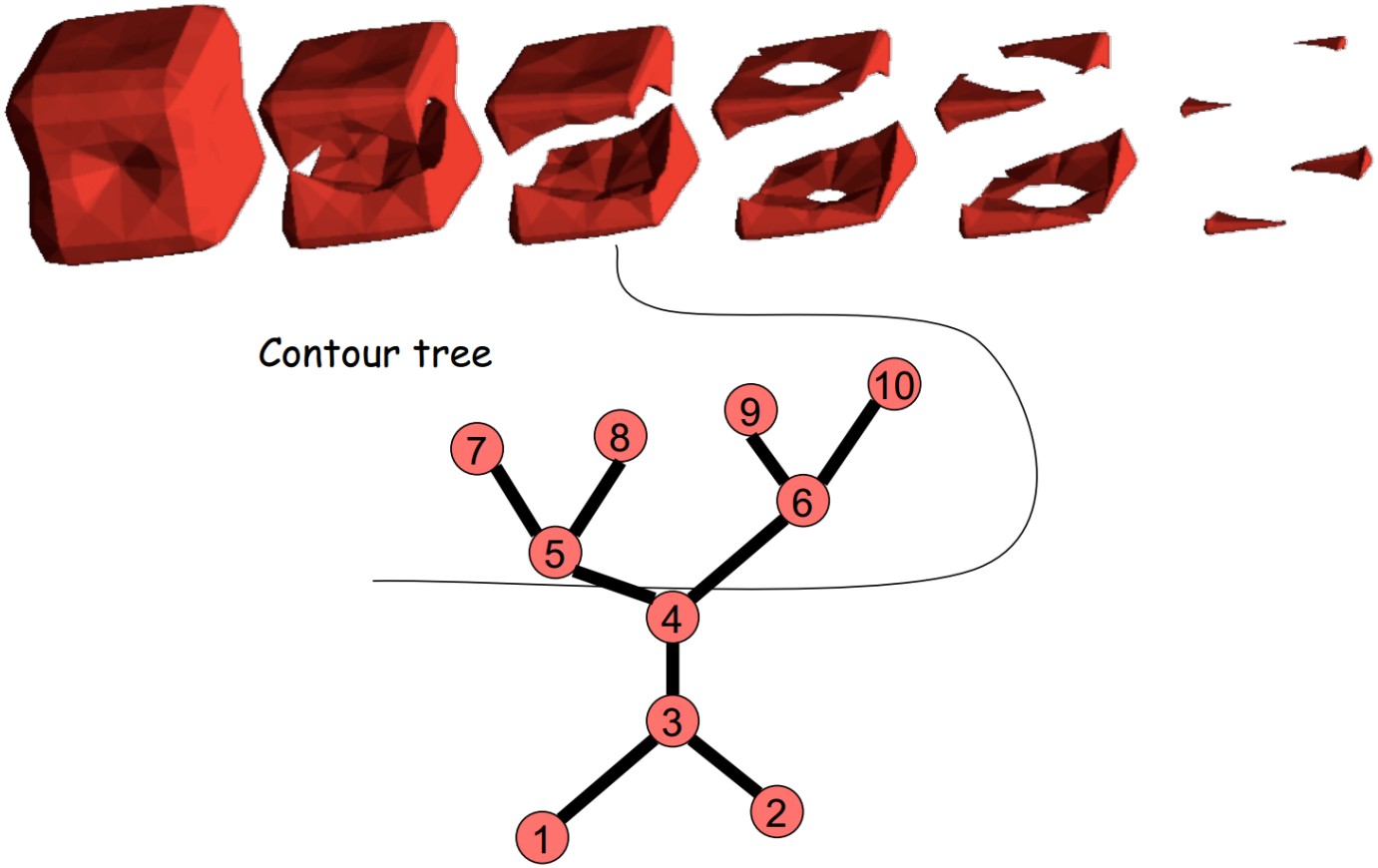


Figure: Evolution of level sets[10]

[10] from Hamish Carr et al.

What are contour trees?

Running Example

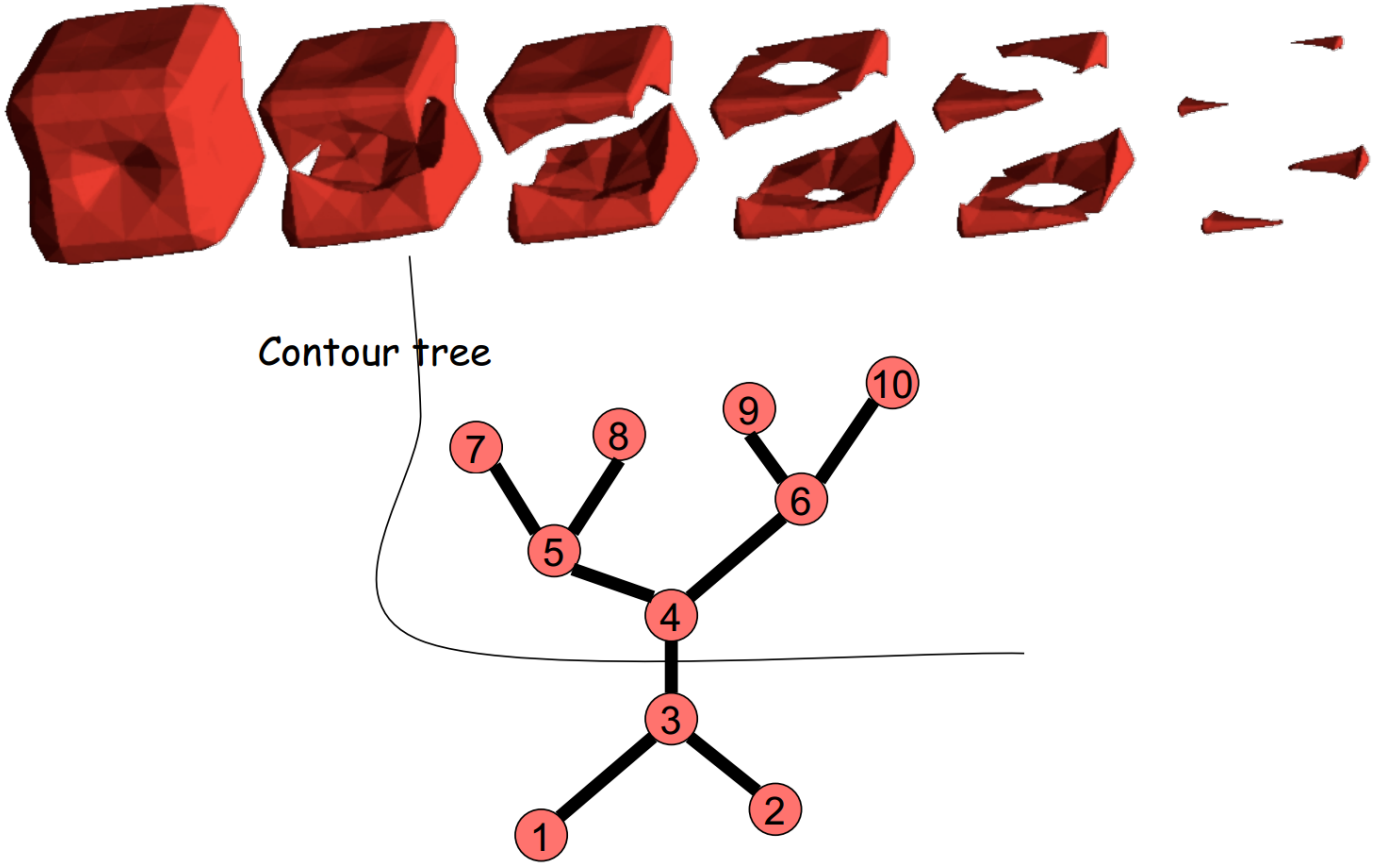


Figure: Evolution of level sets[11]

[11] from Hamish Carr et al.

What are contour trees?

Running Example

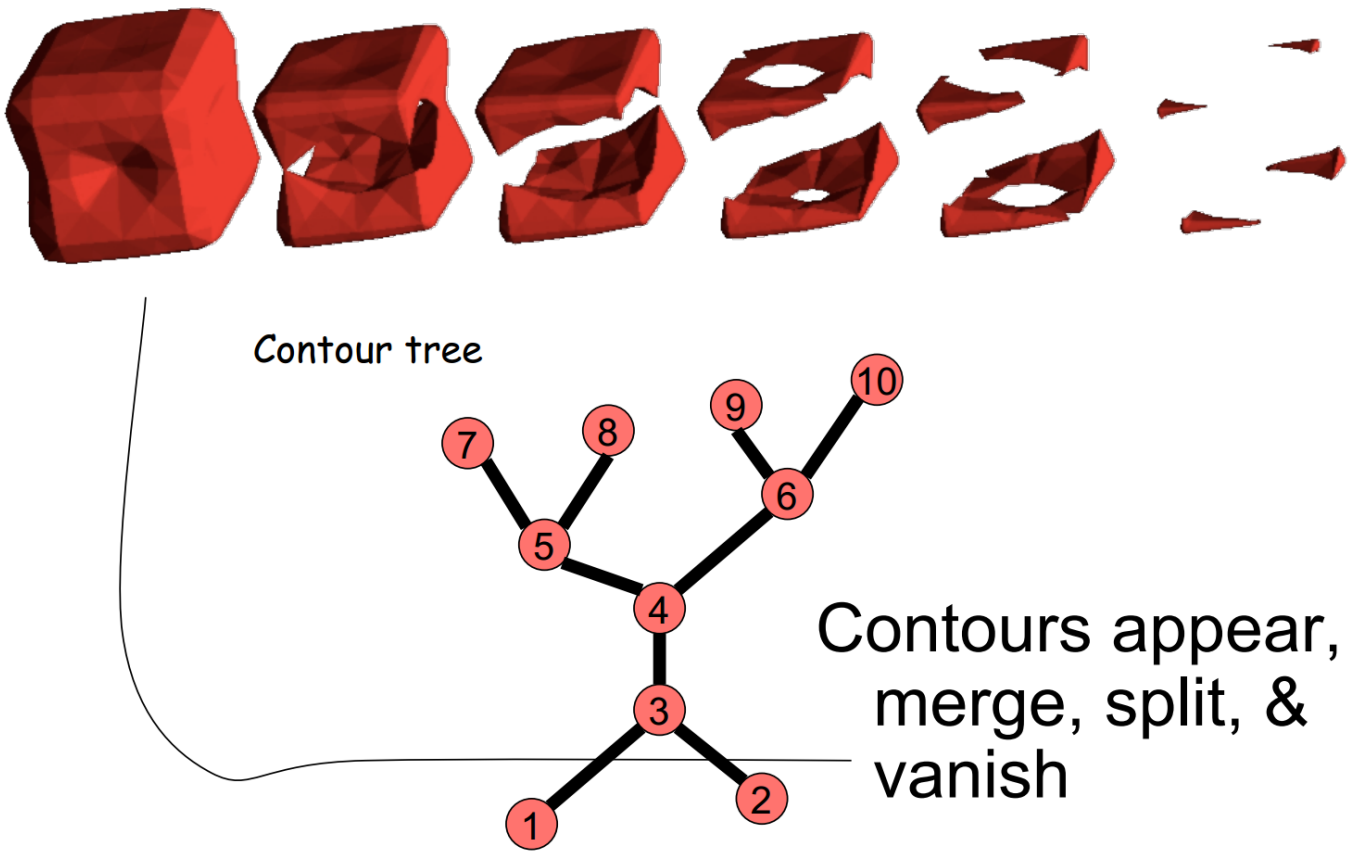


Figure: Evolution of level sets[12]

[12] from Hamish Carr et al.

More background

- Link.
  - Lower link (LL).
  - Upper link (UL).
- Critical points.
  - Maxima (Empty UL).
  - Minima (Empty LL).
  - Saddle (UL/LL not simply connected).
- Identifying by Lower links
- Nodes of Contour tree are all critical points.

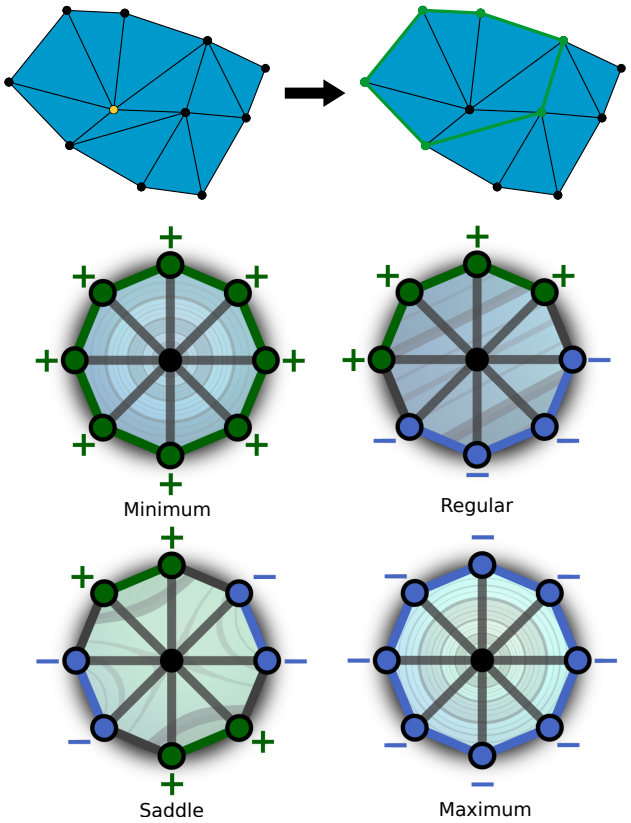


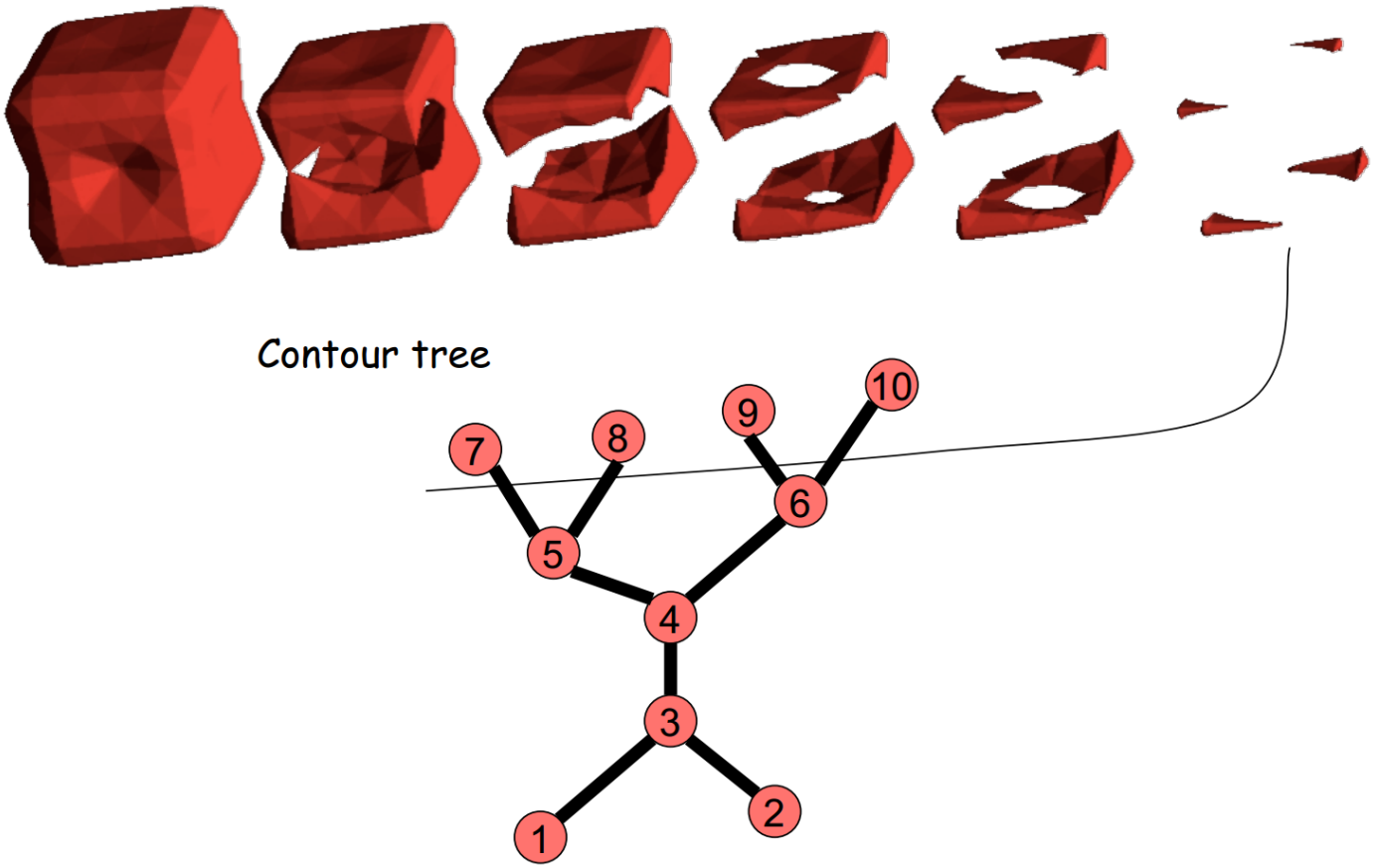
Figure: **Top** Link in Triagulations[a]
Middle/Bottom Classification based on LL[b]

[a] from Wikipedia

[b] from slides by Julian Tierny

Evolution in terms of Critical points

- Minima $\Longrightarrow$ Level set component is created.
- Maxima $\Longrightarrow$ Level set component is destroyed.
- Saddle $\Longrightarrow$ Level set components split or join.



Algorithm

Join tree + Split tree = Contour Tree

- Join tree.
- Split tree.

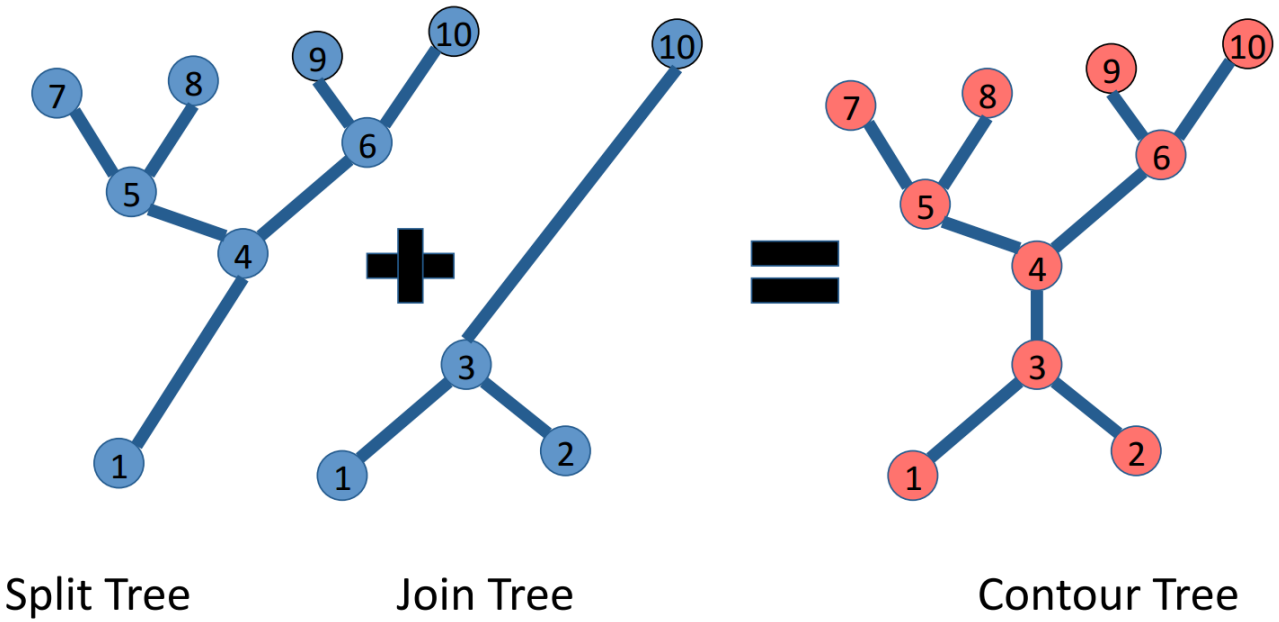


Figure: Join tree + Split tree = Contour Tree[14]

[14] from Hamish Carr et al.

Algorithm

[15] from Hamish Carr et al.

Algorithm

- Construct Join tree.

[15] from Hamish Carr et al.

Algorithm

- Construct Join tree.
- Construct Split tree.

[15] from Hamish Carr et al.

Algorithm

- Construct Join tree.
- Construct Split tree.
- Merge them to get Contour tree.

[15] from Hamish Carr et al.

Algorithm

- Construct Join tree.
- Construct Split tree.
- Merge them to get Contour tree.

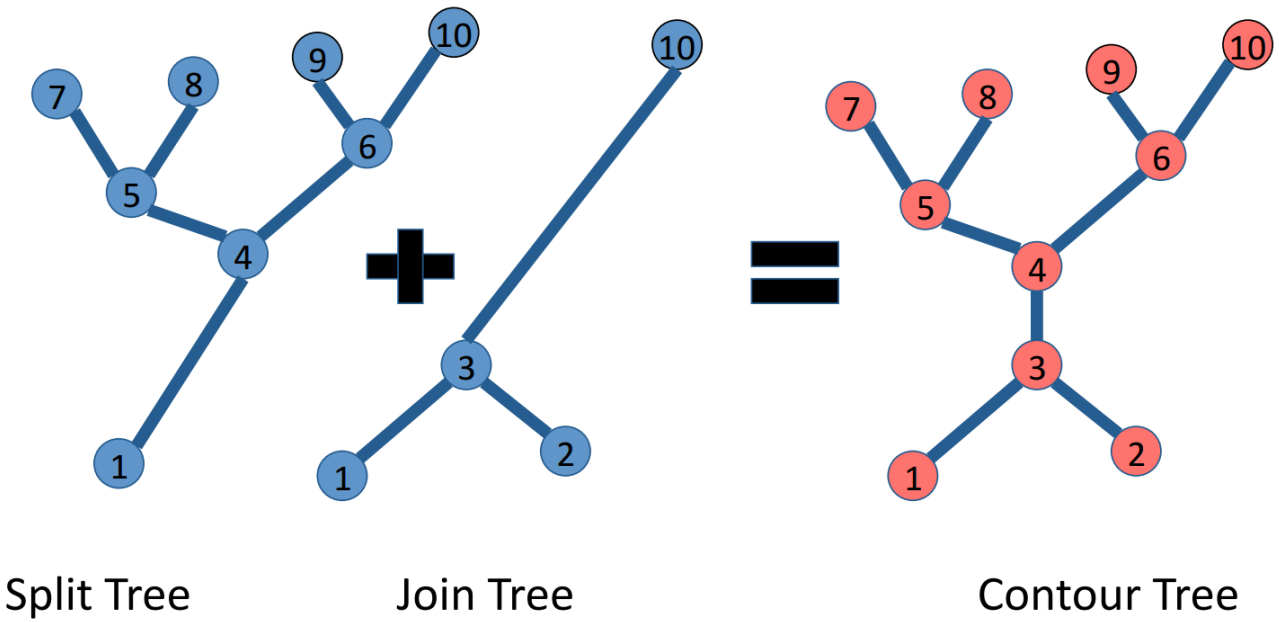


Figure: Join tree + Split tree = Contour Tree[15]

[15] from Hamish Carr et al.

Some observations and details

- Joins are easy to track but splits aren't. Why/how?

[16]

Some observations and details

- Joins are easy to track but splits aren't. Why/how?
- Notice that Split tree is actually Join tree for the function swept in opposite direction.

[16]

Some observations and details

- Joins are easy to track but splits aren't. Why/how?
- Notice that Split tree is actually Join tree for the function swept in opposite direction.
- Use Union-Find (UF) to track joins.

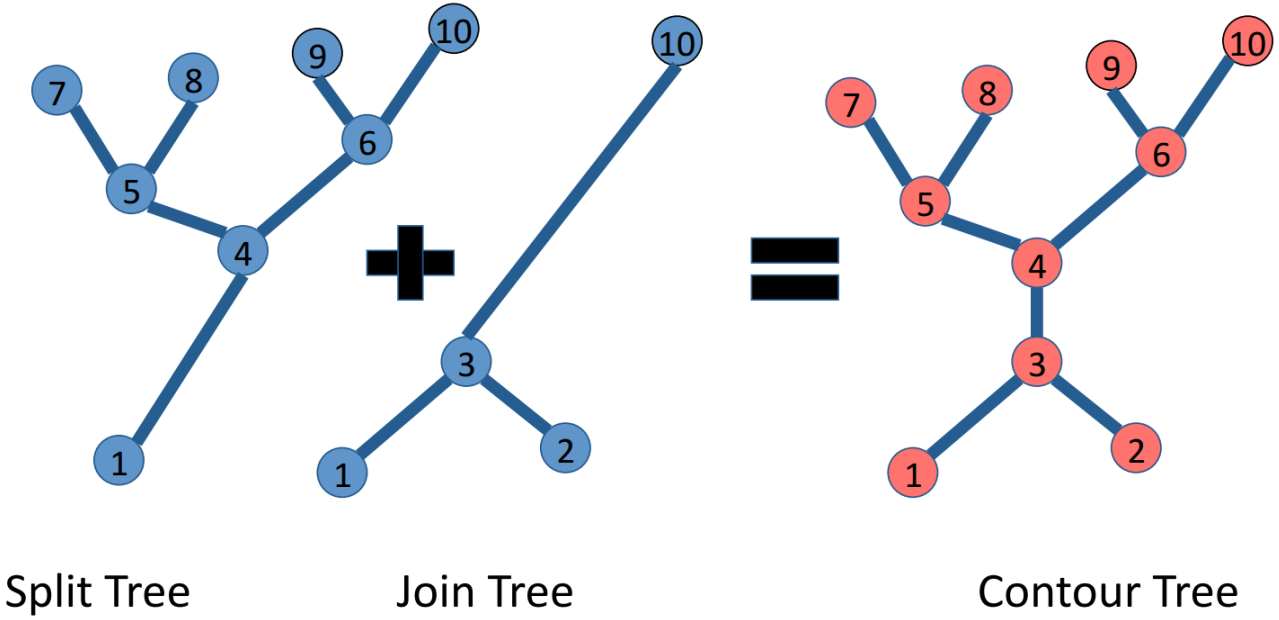


Figure: Join tree + Split tree = Contour Tree[16]

[16]

Algorithm for Join/Split Tree

- Sort all the vertices in the domain based on function values.

```
JoinTree(V, E)
JT = NewTree()
UF = NewUF()
for i = 0 to n − 1
    AddNode(JT, i)
    if LL(v_i) = ∅
        NewSet(UF, i)
    else
        for v_j ∈ LL(v_i)
            i' = Find(UF, i)
            j' = Find(UF, j)
            if j' ≠ i'
                AddEdge(JT, i', j')
                Union(UF, i', j')
    return JT
```

Figure: Algorithm[17]

[17] from Hamish Carr et al.

The output

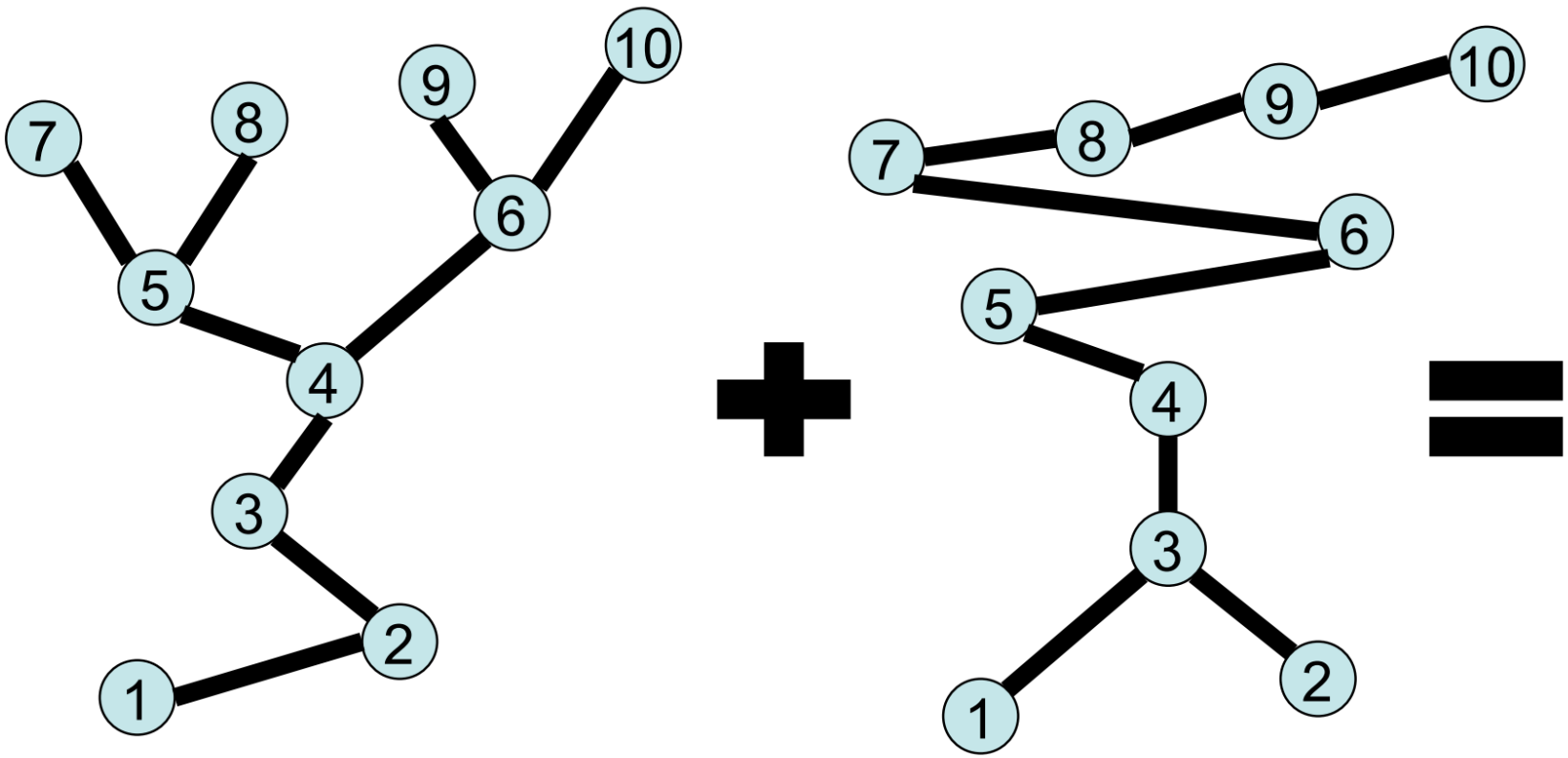


Figure: Algorithm[18]

[18] from Hamish Carr et al.

Merging trees

How to merge?

[19] from Hamish Carr et al.

How to merge?

- Remember retrieving binary trees using two traversals?

[19] from Hamish Carr et al.

How to merge?

- Remember retrieving binary trees using two traversals?
- Nodes have correct up degree in split and down degree in the join tree.

[19] from Hamish Carr et al.

How to merge?

- Remember retrieving binary trees using two traversals?
- Nodes have correct up degree in split and down degree in the join tree.
- Leaf in one tree has up/down degree ≤ 1 in other tree.

[19] from Hamish Carr et al.

How to merge?

- Remember retrieving binary trees using two traversals?
- Nodes have correct up degree in split and down degree in the join tree.
- Leaf in one tree has up/down degree ≤ 1 in other tree.

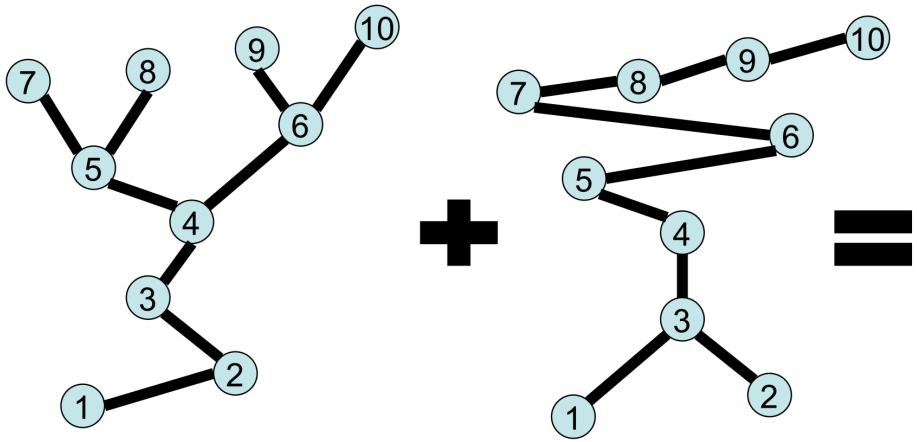


Figure: Observations for merging.[19]

[19] from Hamish Carr et al.

Merge Algorithm

- Initialize CT to be an empty graph on vertices of JT.
- Construct a queue of leaves.
  - if $updegree(v_i)$ in $ST + lowdegree(v_i)$ in $JT \leq 1$ enqueue.
- While $QueueSize > 1$
  - Dequeue first vertex v_i from the Queue.
  - If v_i is the upper leaf, find incident arc $(v_i, v_j) \in ST$
  - Else find incident arc $(v_i, v_j) \in JT$
  - Add (v_i, v_j) to CT
  - Remove v_i from both JT and ST.
  - If v_j is a leaf then Enqueue v_j.

Merge Algorithm

Running Example

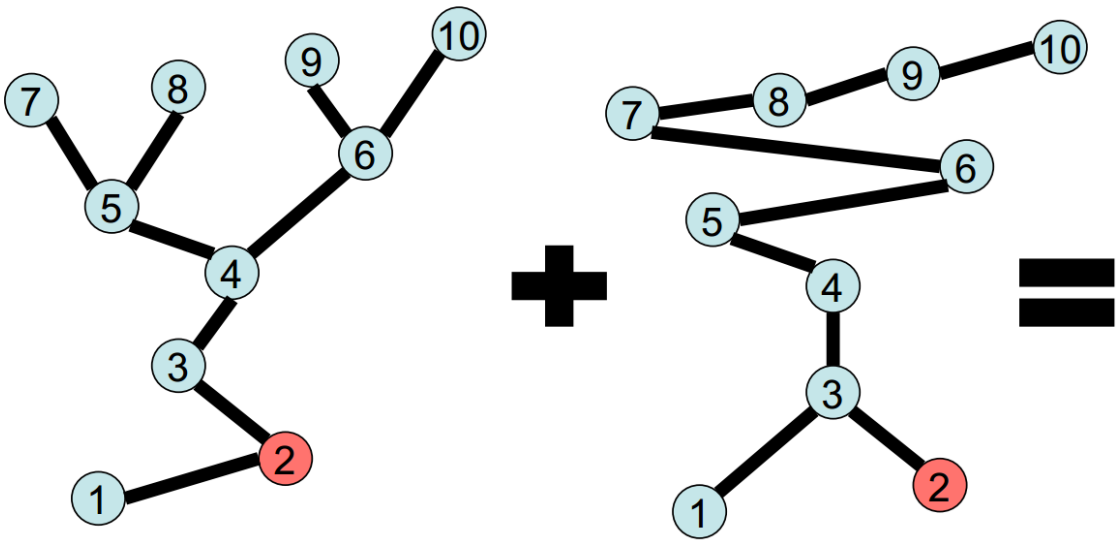


Figure: Working of Merge algorithm[20]

[20] from Hamish Carr et al.

Merge Algorithm

Running Example

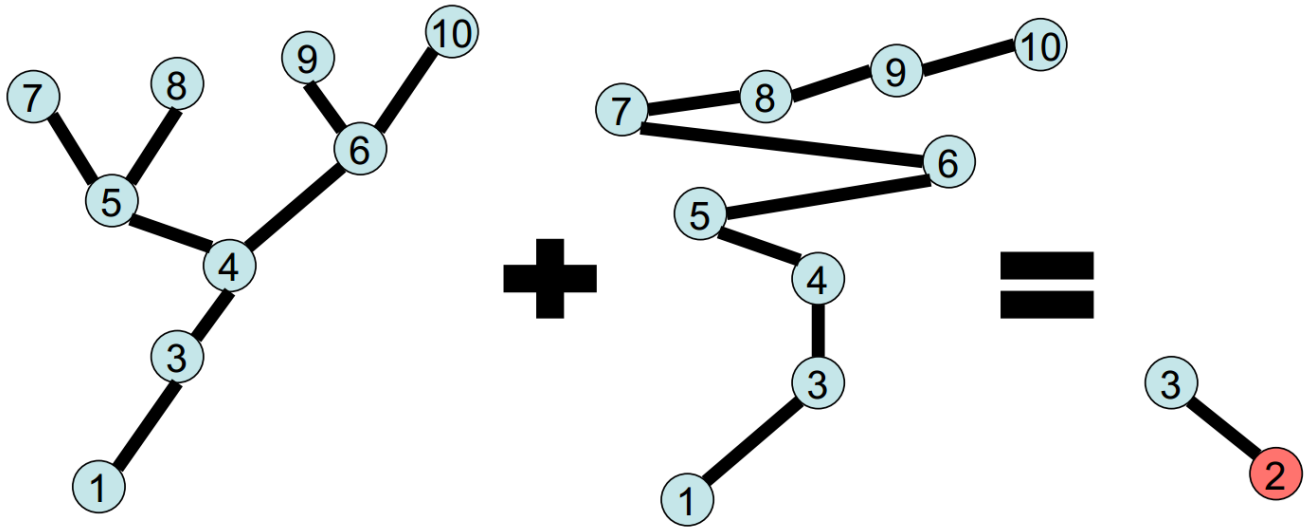


Figure: Working of Merge algorithm[21]

[21] from Hamish Carr et al.

Merge Algorithm

Running Example

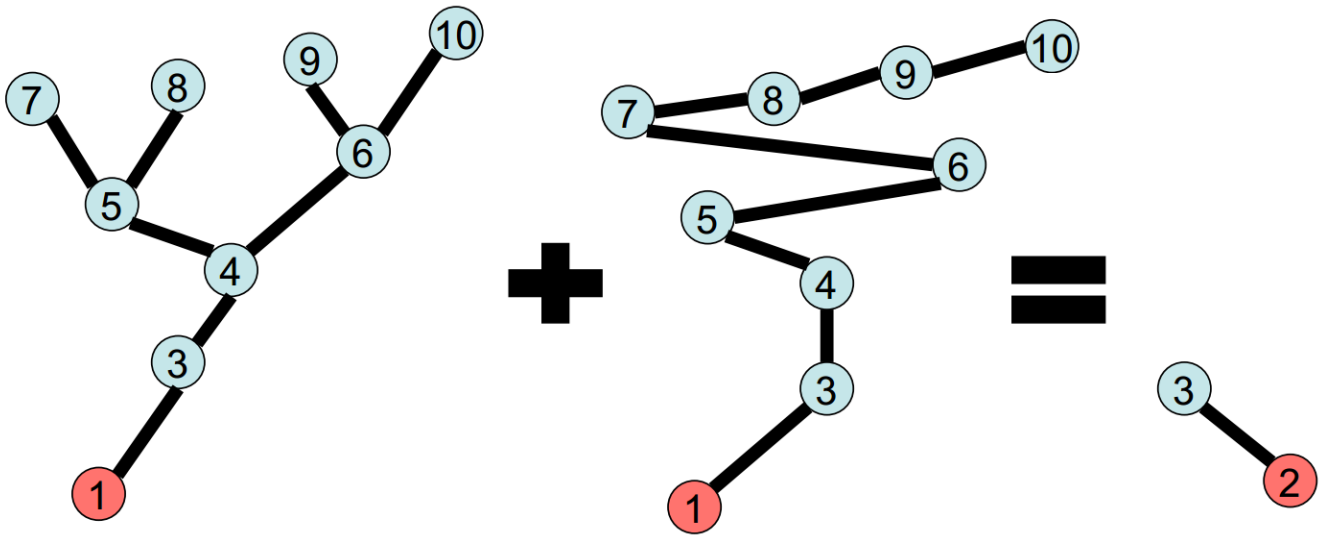


Figure: Working of Merge algorithm[22]

[22] from Hamish Carr et al.

Merge Algorithm

Running Example

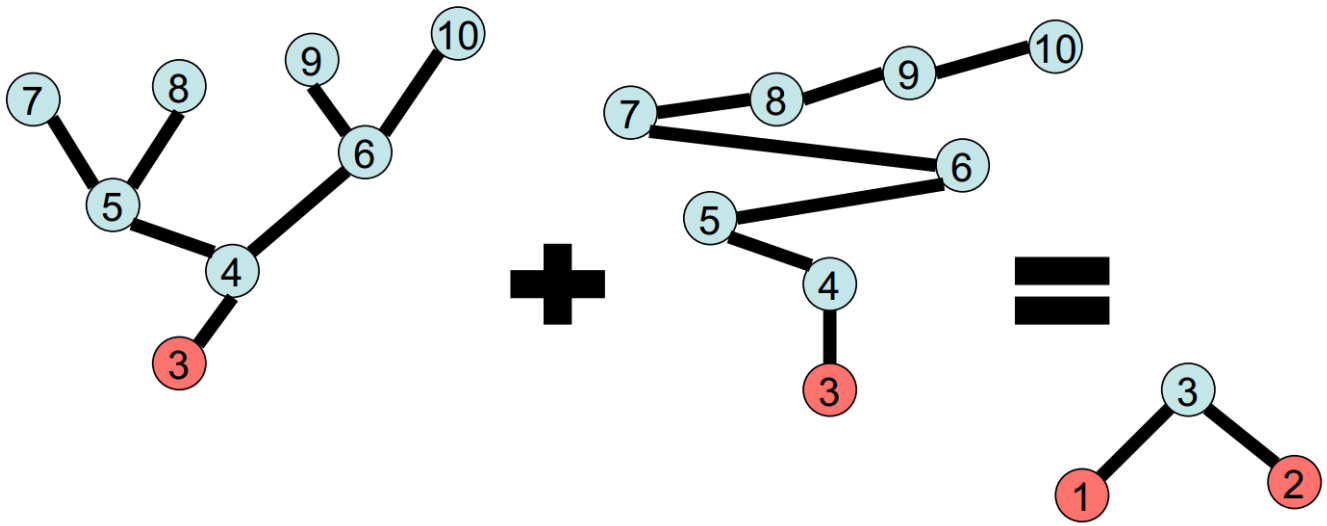


Figure: Working of Merge algorithm[23]

[23] from Hamish Carr et al.

Merge Algorithm

Running Example

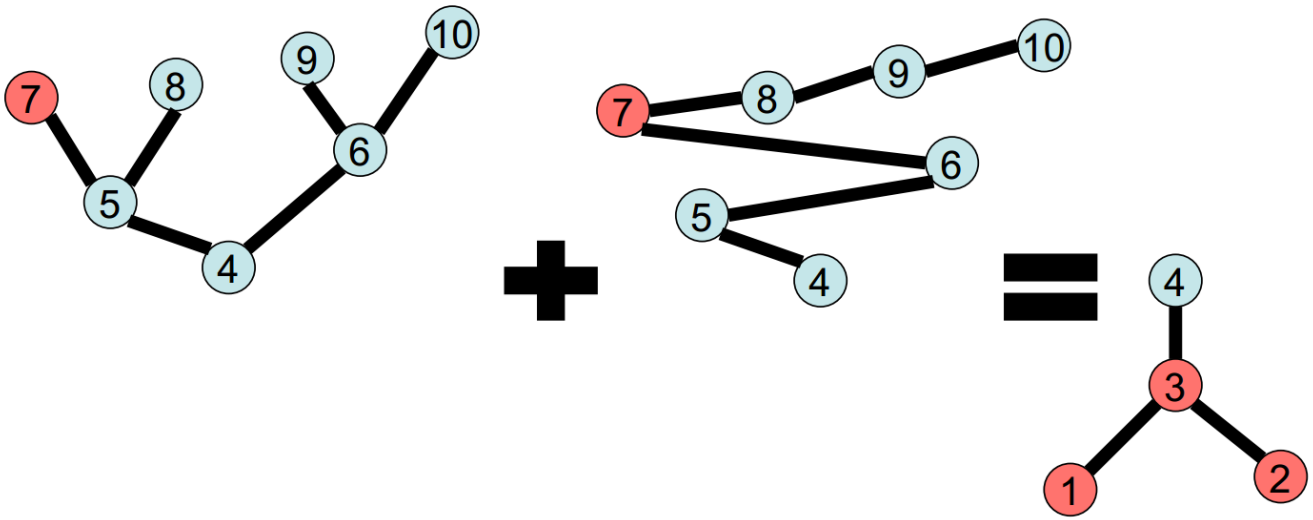


Figure: Working of Merge algorithm[24]

[24] from Hamish Carr et al.

Merge Algorithm

Running Example

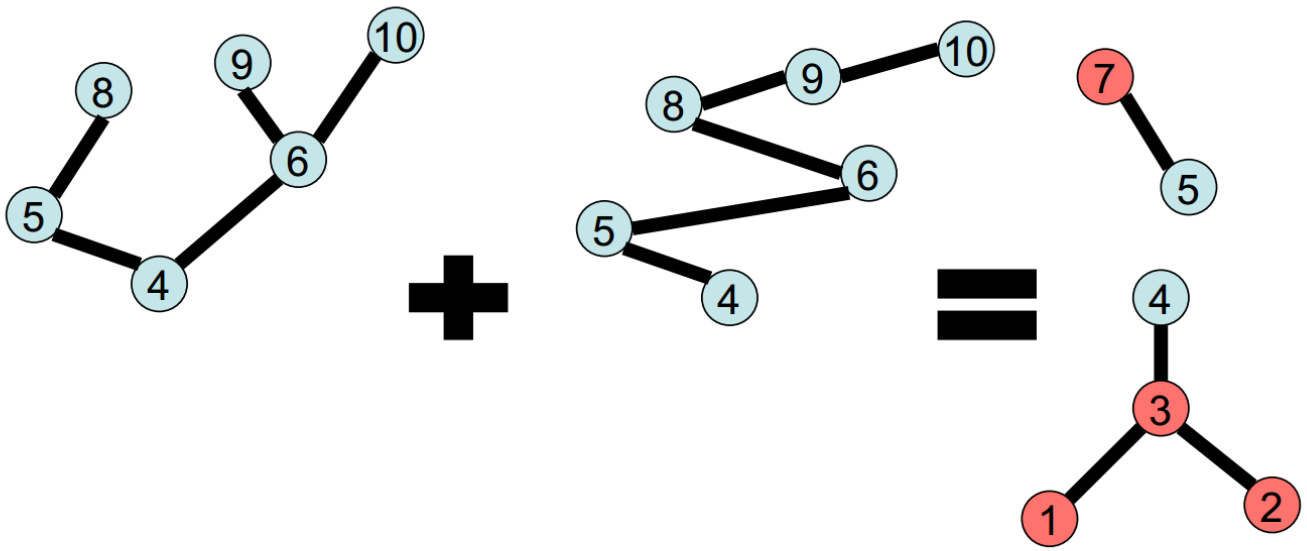


Figure: Working of Merge algorithm[25]

[25] from Hamish Carr et al.

Merge Algorithm

Running Example

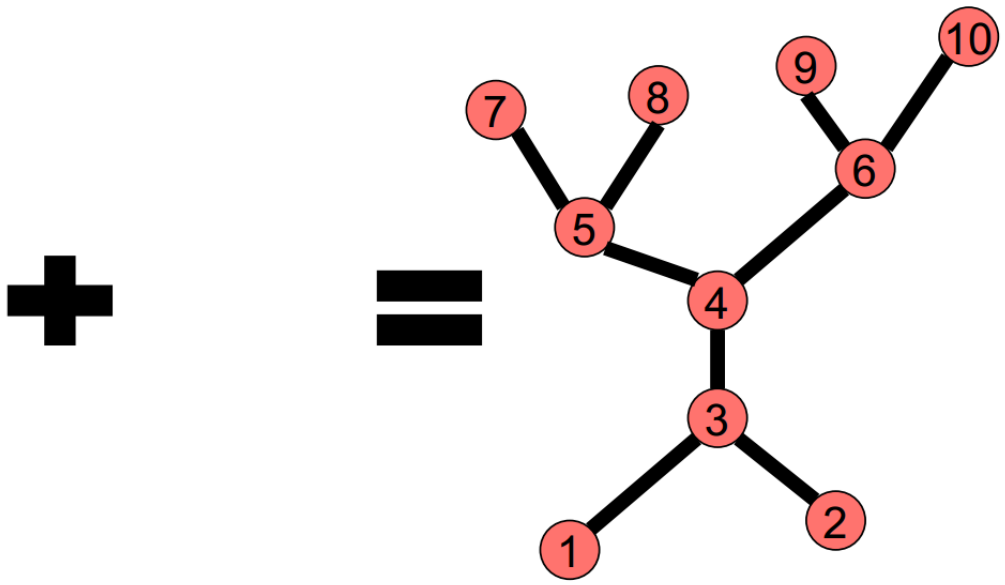


Figure: Working of Merge algorithm[26]

[26] from Hamish Carr et al.

Analysis

Runtime Analysis

Running time for data containing n vertices and N simplices/triangles, is given by $O(n \log n + N\alpha(N))$.

- $O(n \log n)$ for the initial sorting of the vertices based on function values.
- $O(N\alpha(N))$ is for the union find algorithm.

What about correctness?

- Skipped due to time constraint.

Application

One application

Flexible Isosurfaces

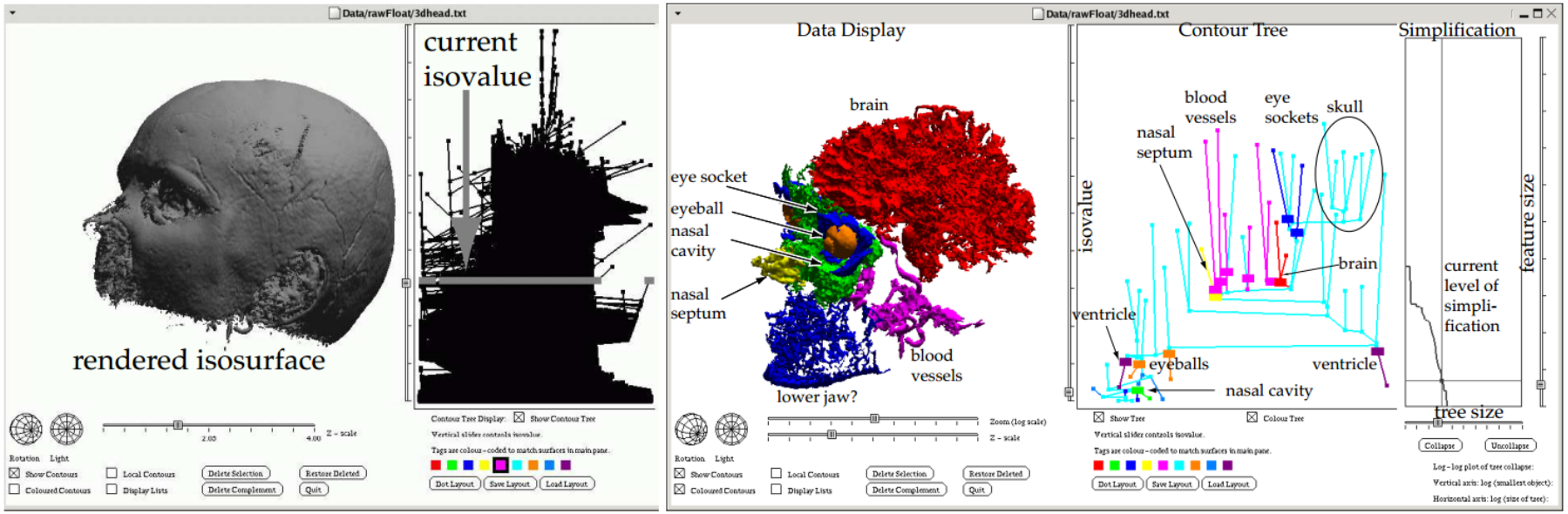


Figure: Flexible Isosurfaces[27]

[27] from Hamish Carr et al.

Conclusion

Conclusions

Conclusions

- We have a simple algorithm for constructing contour trees.
- Works for all dimensions.
- based on split/join trees followed by a merging algorithm.

Thank You