Algorithms for Visualization: Constructing Contour Trees

Raghavendra G S

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

July 6th, 2017

---

Introduction
Scientific Visualization

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

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

Raghavendra G S (VGL CSA, IISc)
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

- Scientific Visualization (Data with geometry information)
  - Scalar Field Visualization
  - Vector Field Visualization
  - Tensor Field Visualization
  - Isosurfaces

- Information Visualization (Data without geometry information)
  - Text/Document Visualization
  - Tree/Graphs/Network Visualization

Visualization

Contour Trees

Topological Methods
Background
Some background

- Scalar fields.
  - $f : \mathbb{R}^{2,3} \rightarrow \mathbb{R}$
- 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 $\mathbb{R}^2$  
*Middle* Isocontours and contour tree

---

*a* from Wikipedia  
*b* from Rephael Wagner,  
The data

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

Figure: Top Example mesh/grid $^a$ Bottom Dividing a cube $^b$


$^b$from Rephael Wagner
More about Isosurfaces

Marching cubes

Figure: Left: Data set. Right: Isosurfaces.\(^3\)

More about Isosurfaces

Figure: Marching cubes lookup table.\textsuperscript{4}

\textsuperscript{4}from Wikipedia
Contour trees
Problems/Why contour trees?

Figure: Need for contour tree.\(^5\)

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?

Contour trees track the changes in the connectivity of level sets of the function. It has a set of nodes where connectivity of level sets change. It has a set of edges which connects these nodes in a meaningful manner.
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

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

\(^9\)from Hamish Carr et al.
What are contour trees?

Running Example

![Contour Tree Diagram]

**Figure:** Evolution of level sets

---

10 from Hamish Carr et al.
What are contour trees?

Running Example

Contour tree

Figure: Evolution of level sets

Superscript 11 from Hamish Carr et al.
What are contour trees?

Running Example

![Contour tree diagram]

Contours appear, merge, split, & vanish

Figure: Evolution of level sets

---

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 Triangulations\(^a\)
Middle/Bottom Classification based on LL\(^b\)

\(^a\)from Wikipedia
\(^b\)from slides by Julian Tierny
Evolution in terms of Critical points

- Minima $\implies$ Level set component is created.
- Maxima $\implies$ Level set component is destroyed.
- Saddle $\implies$ Level set components split or join.
Algorithm
Join tree + Split tree = Contour Tree

- Join tree.
- Split tree.

![Diagram](image)

**Figure**: Join tree + Split tree = Contour Tree

---

14 from Hamish Carr et al.
Algorithm

2. Construct Split tree.
3. Merge them to get Contour tree.

From Hamish Carr et al.

Raghavendra G S (VGL CSA, IISc)
Algorithm

- Construct Join tree.
Algorithm

- Construct Join tree.
- Construct Split tree.

from Hamish Carr et al.
Algorithm

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

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

Figure: Join tree + Split tree = Contour Tree

\[^{15}\text{from Hamish Carr et al.}\]
Some observations and details

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

\[ \text{Join tree} + \text{Split tree} = \text{Contour Tree} \]

\[ \text{Figure: Join tree + Split tree = Contour Tree}^{16} \]
Algorithm for Join/Split Tree

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

\[
\text{JOIN TREE}(V, E)
\]

1. \( JT = \text{NEW TREE}() \)
2. \( UF = \text{NEW UF}() \)
3. \( \text{for } i = 0 \text{ to } n - 1 \)
4. \( \text{ADD NODE}(JT, i) \)
5. \( \text{if } LL(v_i) = \emptyset \)
6. \( \text{NEW SET}(UF, i) \)
7. \( \text{else} \)
8. \( \text{for } v_j \in LL(v_i) \)
9. \( i' = \text{FIND}(UF, i) \)
10. \( j' = \text{FIND}(UF, j) \)
11. \( \text{if } j' \neq i' \)
12. \( \text{ADD EDGE}(JT, i', j') \)
13. \( \text{UNION}(UF, i', j') \)
14. \( \text{return } JT \)

Figure: Algorithm\textsuperscript{17}

---

\textsuperscript{17} from Hamish Carr et al.
The output

Figure: Algorithm

\(^{18}\) from Hamish Carr et al.
Merging trees
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 $\leq 1$ in other tree.

Figure: Observations for merging.

\[19\] from Hamish Carr et al.
How to merge?

- Remember retrieving binary trees using two traversals?
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 $\leq 1$ in other tree.
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 $\leq 1$ in other tree.

Figure: Observations for merging.
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\textsuperscript{20}
Merge Algorithm

Running Example

Figure: Working of Merge algorithm\textsuperscript{21}

\textsuperscript{21}from Hamish Carr et al.
Running Example

Figure: Working of Merge algorithm\textsuperscript{22}

\textsuperscript{22} from Hamish Carr et al.
Merge Algorithm

Running Example

Figure: Working of Merge algorithm

---

from Hamish Carr et al.
Merging trees

Merge Algorithm

Running Example

Figure: Working of Merge algorithm

---

from Hamish Carr et al.
Merge Algorithm

Running Example

Figure: Working of Merge algorithm\textsuperscript{25}

\textsuperscript{25}from Hamish Carr et al.
Merge Algorithm

Running Example

Figure: Working of Merge algorithm

\[ + \quad = \]

---

\(^{26}\)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

from Hamish Carr et al.
Conclusion
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