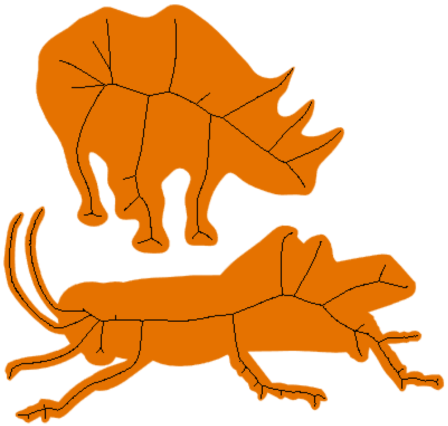
CSE 554

Lecture 3: Shape Analysis (Part II)

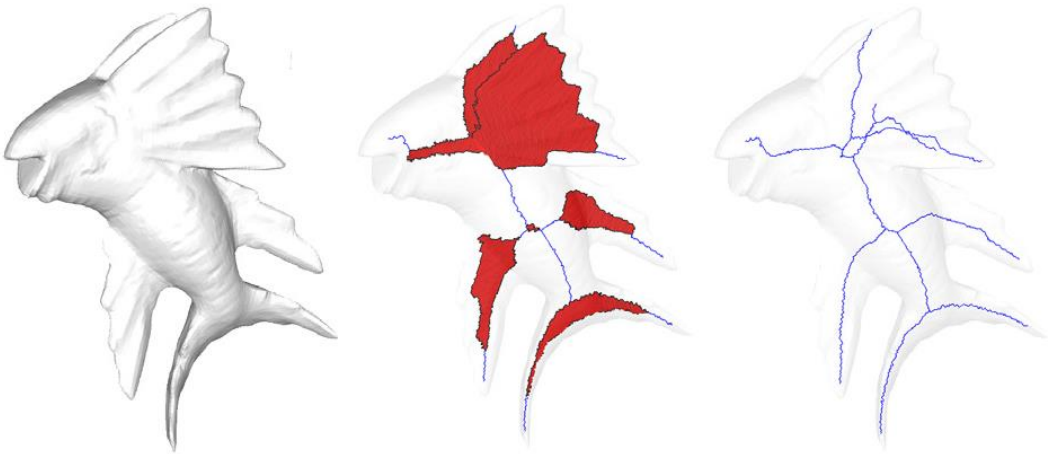
Fall 2018

Review

- Skeletons
  - Centered curves/surfaces
    - Approximations of medial axes
  - Useful for shape analysis



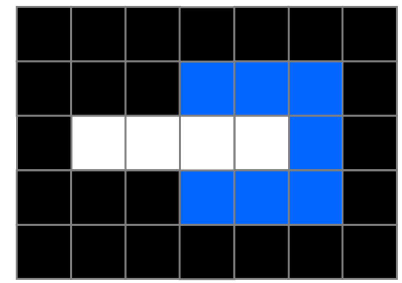
2D skeletons



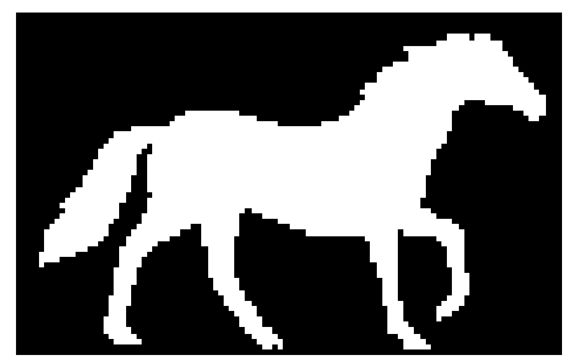
3D surface and curve skeletons

Review

- Thinning on binary pictures
  - Removable pixels (voxels)
    - Whose removal does not alter the object’s shape or topology
    - Border, Simple, and not curve-end
  - Strategies
    - Parallel thinning: topology is lost
    - Serial thinning: topology is preserved
      - But result depends on pixel order



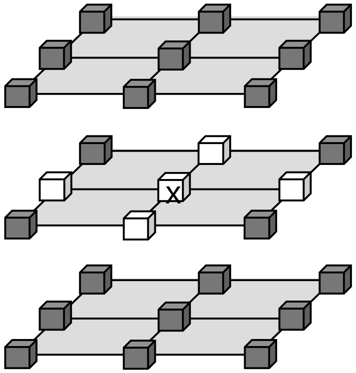
Removal pixels



Thinning

Review

- Issues (with thinning on a binary picture)
  - Difficult to write a 3D thinning algorithm
    - E.g., simple voxel criteria, surface-end voxel criteria
  - Skeletons can be noisy
    - Requires pruning



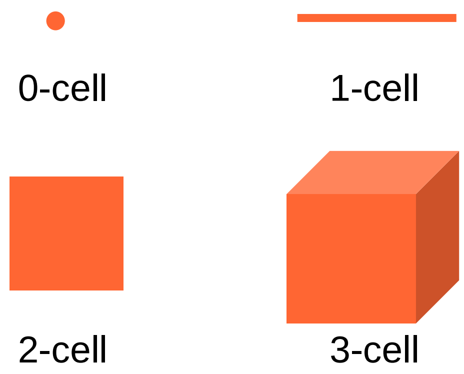
This lecture…

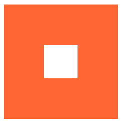
- Thinning on a cell complex
  - One algorithm that works for shapes in any dimensions (2D, 3D, etc.)
  - Integrates pruning with thinning
  - Based on [Liu et al., 2010]

Cells

Washington University in St.Louis
School of Engineering & Applied Science

- Geometric elements with simple topology
  - k-cell: an element at dimension k that can be continuously deformed to a k-dimensional ball
    - 0-cell: point
    - 1-cell: line segment, curve segment, …
    - 2-cell: triangle, quad, …
    - 3-cell: cube, tetrahedra, …



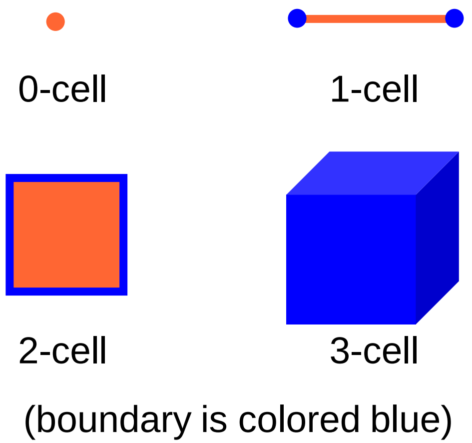


Not a 2-cell

Not a 3-cell

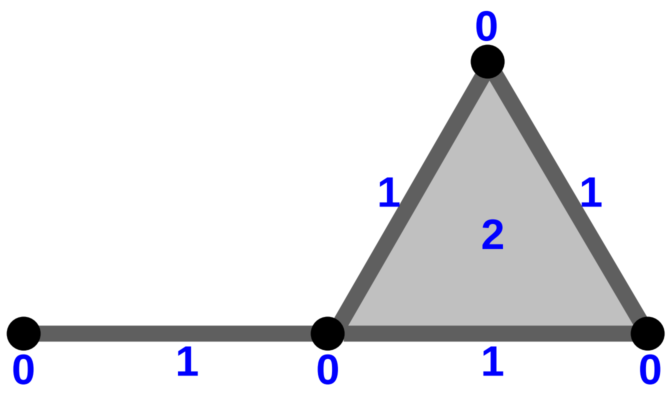
Cells

- The boundary of a k-cell (k>0) has dimension k-1
  - Examples:
    - A 1-cell is bounded by 0-cells
    - A 2-cell is bounded by 1-cells
    - A 3-cell is bounded by 2-cells



Cell Complex

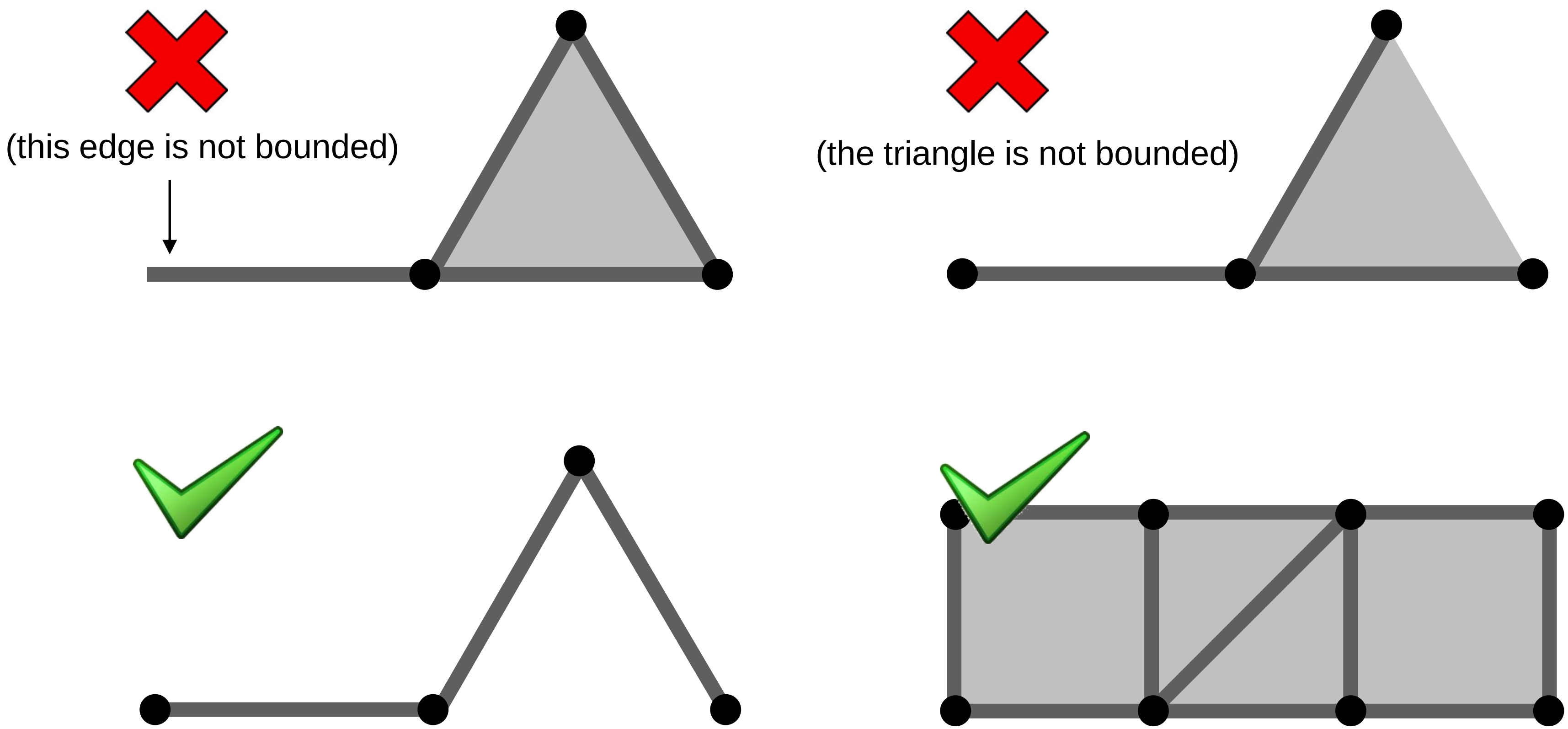
- Union of cells and cells on their boundaries



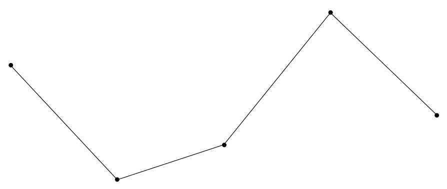
  - The formal name is CW (closure-finite, weak-topology) Complex
    - Precise definition can be found in algebraic topology books

Cell Complex

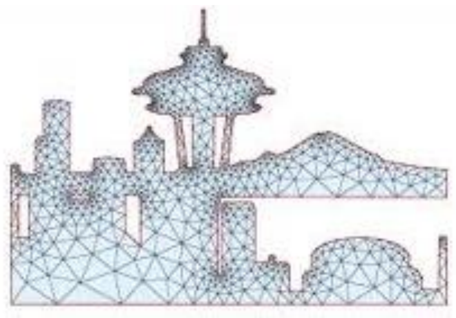
- Are these cell complexes?



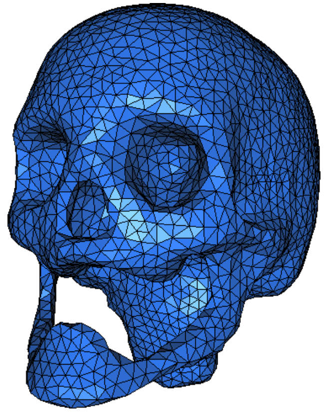
Example Cell Complexes



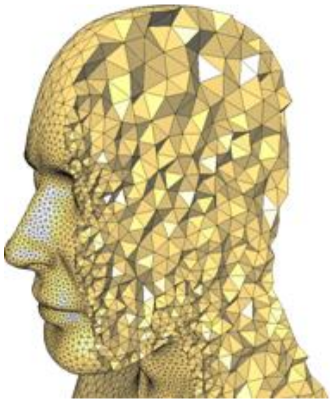
Polyline
(0-,1-cells)



Triangulated polygon
(0-,1-, 2-cells)



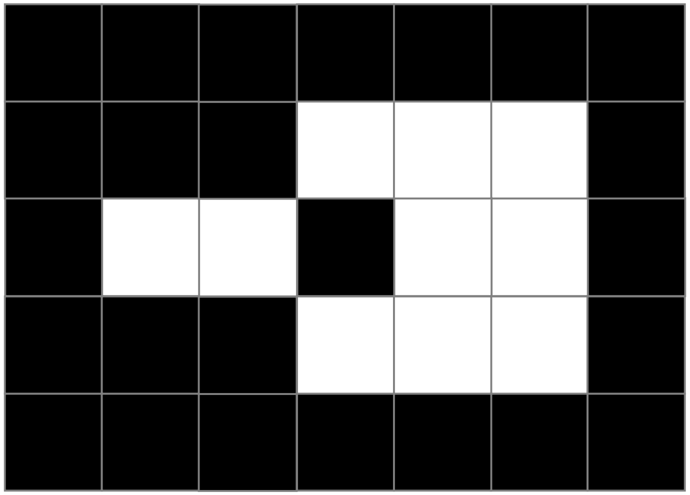
Triangular mesh
(0-,1-, 2-cells)

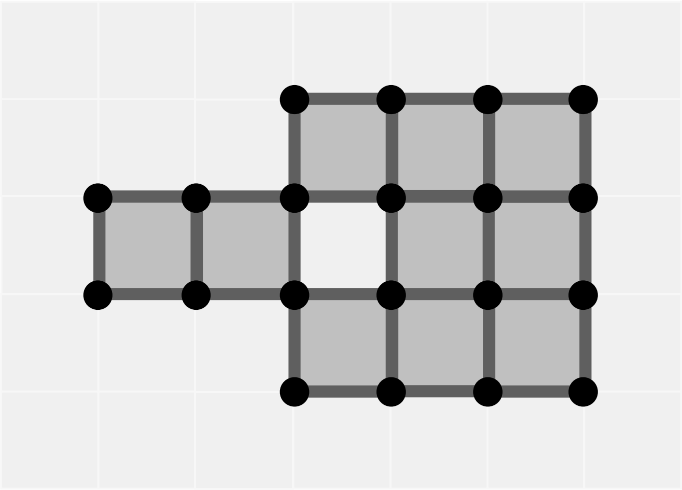


Tetrahedral volume
(0-,1-, 2-, 3-cells)

Cell Complex from Binary Pic

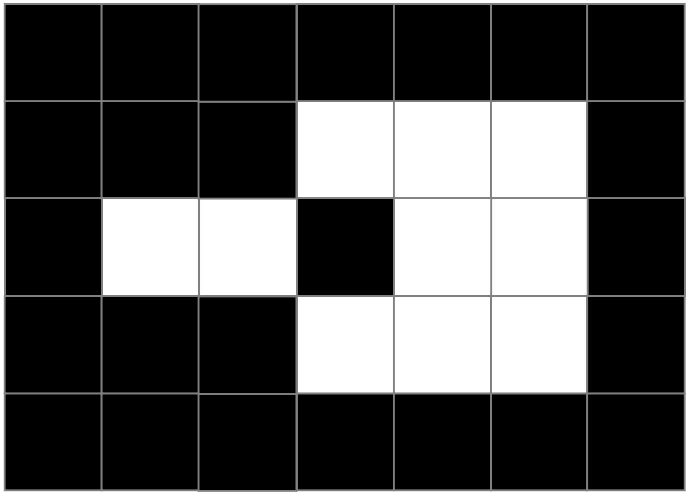
- Representing the object as a cell complex
  - Approach 1: create a 2-cell (3-cell) for each object pixel (voxel), and add all boundary cells
    - Reproducing 8-connectivity in 2D and 26-connectivity in 3D

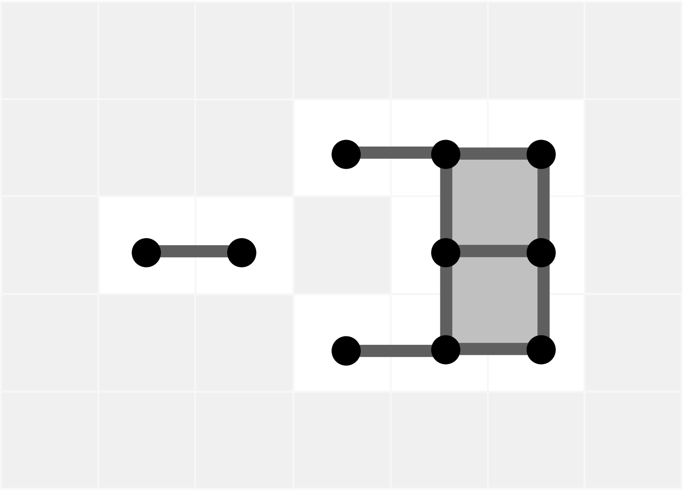




Cell Complex from Binary Pic

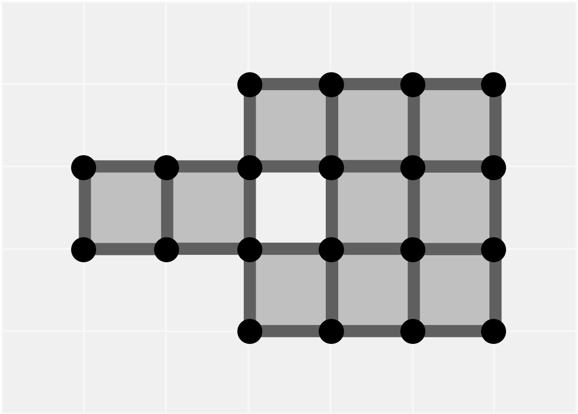
- Representing the object as a cell complex
  - Approach 2: create a 0-cell for each object pixel (voxel), and connect them to form higher dimensional cells.
    - Reproducing 4-connectivity in 2D and 6-connectivity in 3D





Algorithm: Approach 1

- 2D:
  - For each object pixel, create a 2-cell (square), four 1-cells (edges), and four 0-cells (points).
- 3D:
  - For each object voxel, create a 3-cell (cube), six 2-cells (squares), twelve 1-cells (edges), and eight 0-cells (points).



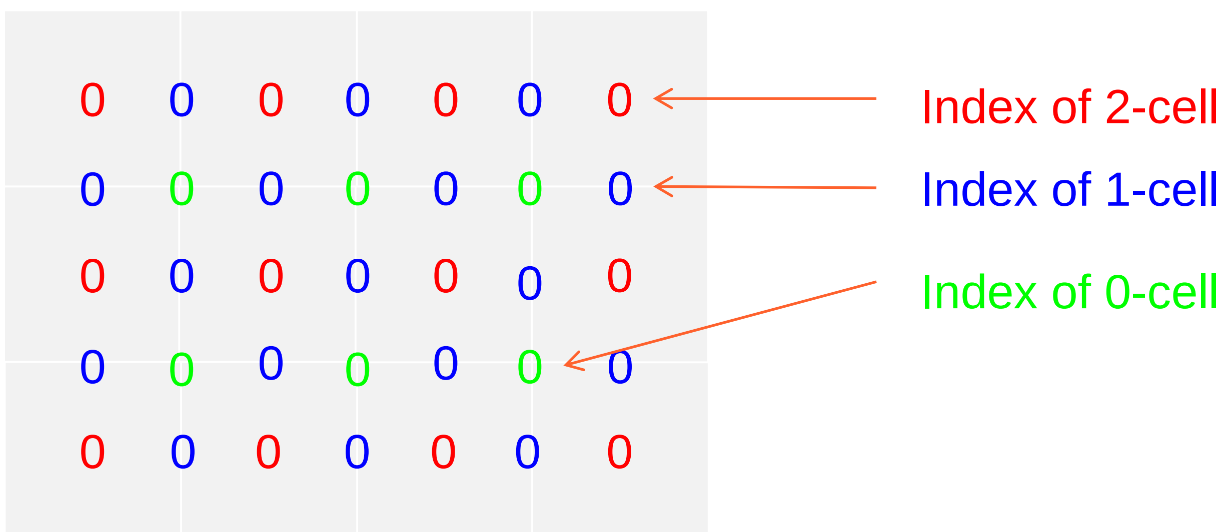
- *Challenge: avoid creating duplicated cells*

Algorithm: Approach 1

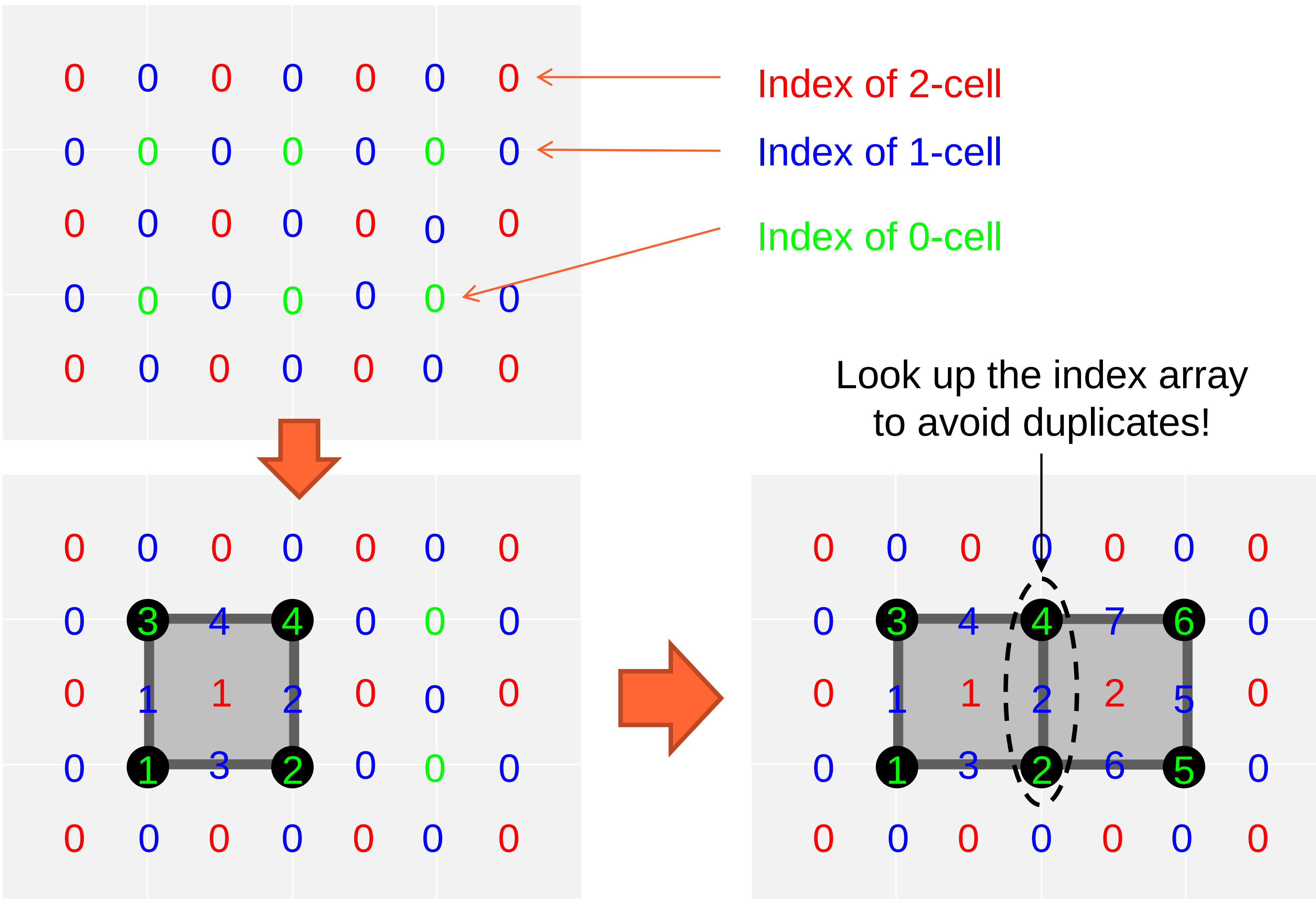
- Avoid duplicates
  - Use a data structure to keep track of the index of cells
    - Look up before creating a new cell, and update if a new cell is created
  - We could use a hash table
    - Indexed by the coordinates of the 0-, 1-, or 2- cells
    - Possible hash collision and/or unused hashing space
  - In our case, an array is more efficient (perfect hashing)

Algorithm: Approach 1

- For a binary image of dimension n by n, this extra array has dimension of 2n-1 by 2n-1
  - Stores the index of 0-cells, 1-cells, 2-cells once they are created
  - Initialized to be all zero (indicating no cells have been created)

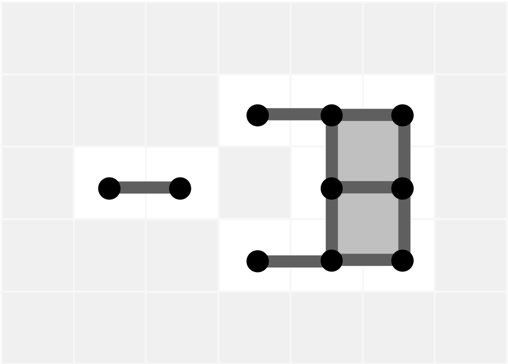


Algorithm: Approach 1



Algorithm: Approach 2

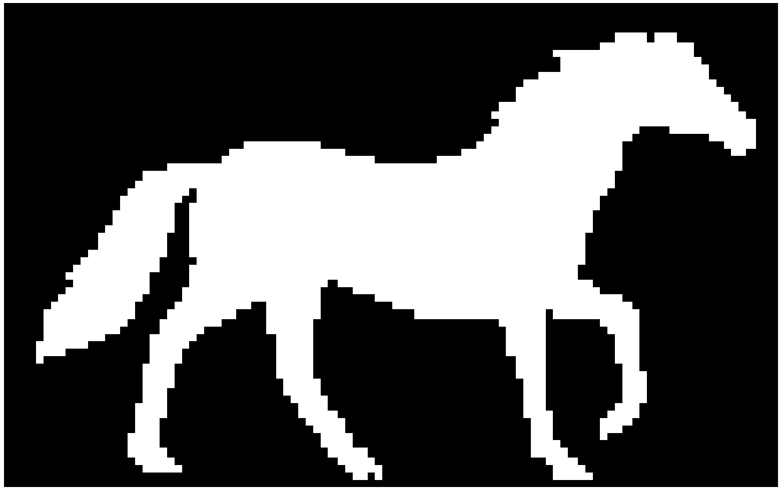
- 2D:
  - Create a 0-cell at each object pixel, a 1-cell for two object pixels sharing a common edge, and a 2-cell for four object pixels sharing a common point



- 3D:
  - Create a 0-cell at each object voxel, a 1-cell for two object voxels sharing a common face, a 2-cell for four object voxels sharing a common edge, and a 3-cell for eight object voxels sharing a common point
- Same strategy as in Approach 1 for storing cell indices

Thinning on Binary Pictures

- Remove simple pixels (voxels)
  - Whose removal does not affect topology
- Protect end pixels (voxels) of skeleton curves and surfaces
  - To prevent shrinking of skeleton



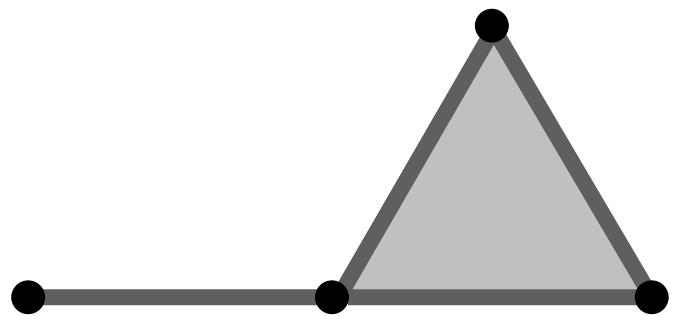
Thinning on Cell Complexes

- Remove simple pairs
  - Whose removal does not affect topology
- Protect medial cells
  - To prevent shrinking of skeleton
  - To prune noise

- Advantages:
  - Easy to detect in 2D and 3D (same code)
  - Robust to noise

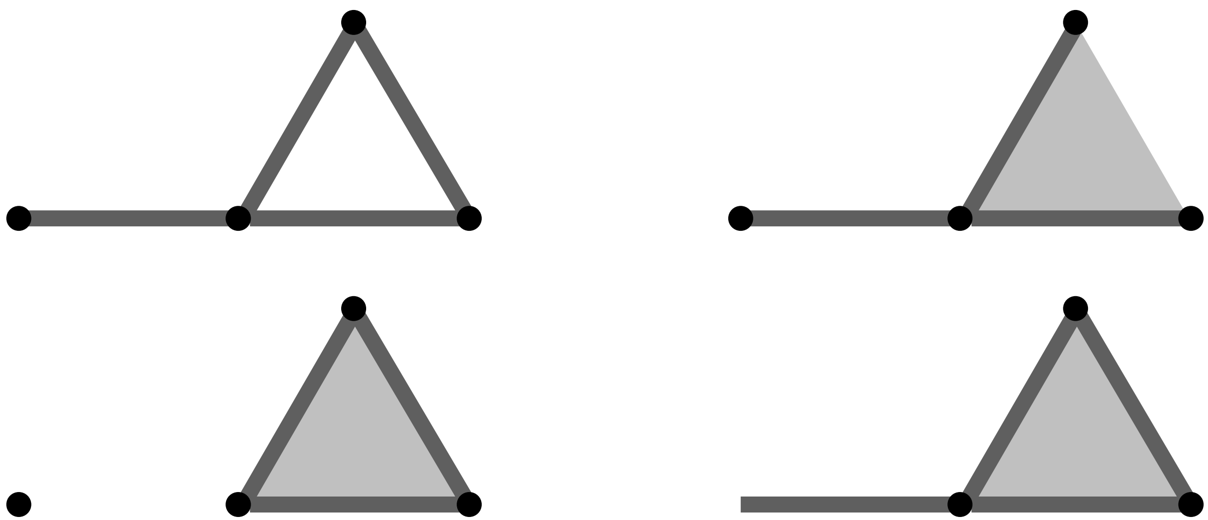
Simple Pairs

- How can we remove cells from a complex so that the result is still a complex and has the same topology?



Simple Pairs

- How can we remove cells from a complex so that the result is still a complex and has the same topology?
  - Removing a single cell will either change topology or not result in a cell complex

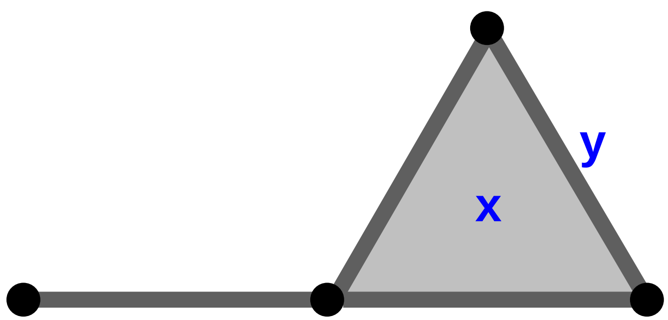


Topology changed

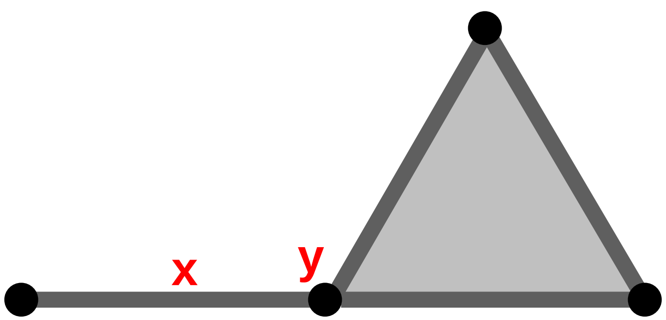
Not a cell complex

Simple Pairs

- Definition
  - A pair {**x**, **y**} such that **y** is on the boundary of **x**, and there is no other cell in the complex with **y** on its boundary.



{x, y} is a simple pair

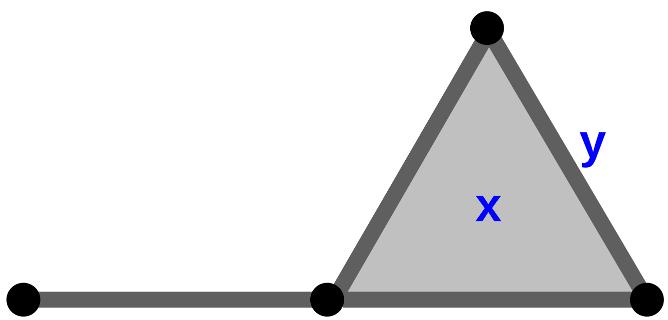


{x,y} is not a simple pair

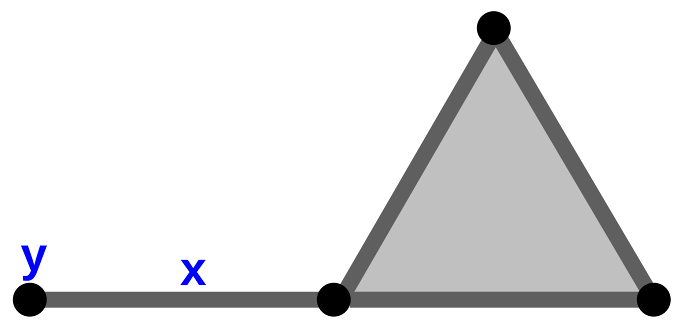
Simple Pairs

Washington University in St. Louis
School of Engineering & Applied Science

- Definition
  - A pair {**x**, **y**} such that **y** is on the boundary of **x**, and there is no other cell in the complex with **y** on its boundary.



{x, y} is a simple pair

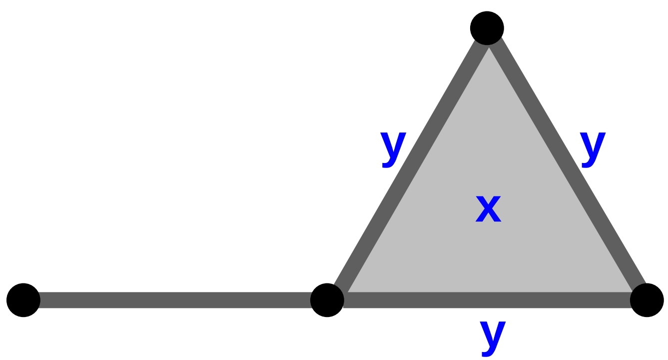


{x,y} is a simple pair

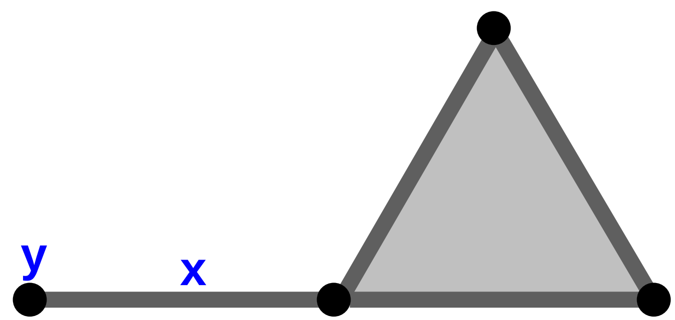
Simple Pairs

Washington University in St. Louis
School of Engineering & Applied Science

- Definition
  - A pair {**x**, **y**} such that **y** is on the boundary of **x**, and there is no other cell in the complex with **y** on its boundary.
  - In a simple pair, **x** is called a simple cell, and **y** is called the witness of **x**.
    - A simple cell can pair up with different witnesses



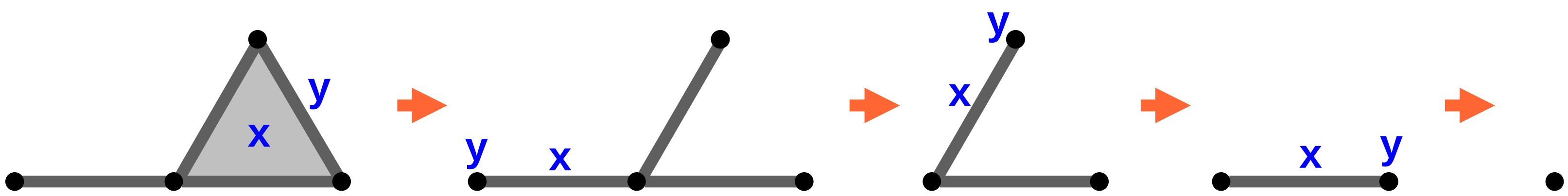
{x, y} is a simple pair



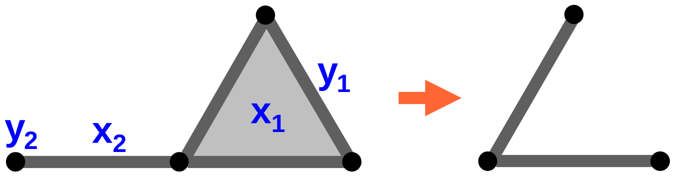
{x,y} is a simple pair

Simple Pairs

- Removing a simple pair does not change topology



  – True even when multiple simple pairs are removed together
    - As long as the pairs are disjoint
    - “Almost” parallel thinning



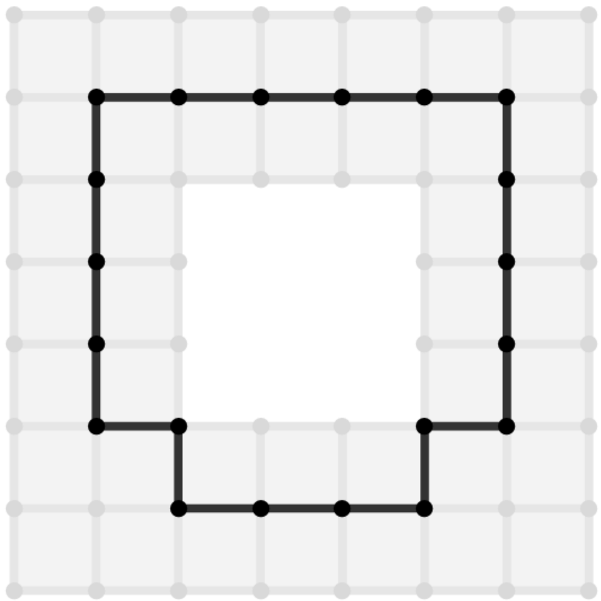
Exhaustive Thinning

- Removing all simple pairs in parallel at each iteration
  - Only the topology of the cell complex is preserved
  - If a simple cell has multiple witnesses, an arbitrary choice is made

```
// Exhaustive thinning on a cell complex C
1. Repeat:
    1. Let S be all disjoint simple pairs in C
    2. If S is empty, Break.
    3. Remove all cells in S from C
2. Output C
```

Exhaustive Thinning

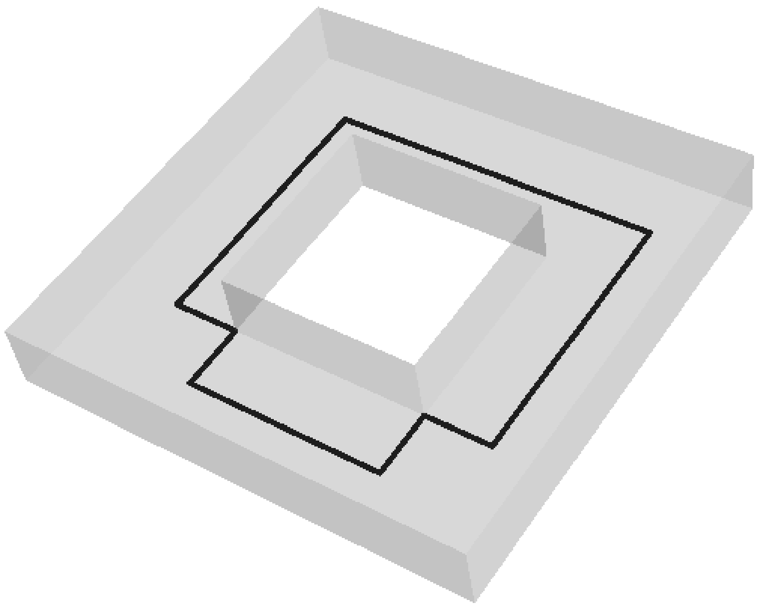
- Removing all simple pairs in parallel at each iteration
  - Only the topology of the cell complex is preserved



2D example

Exhaustive Thinning

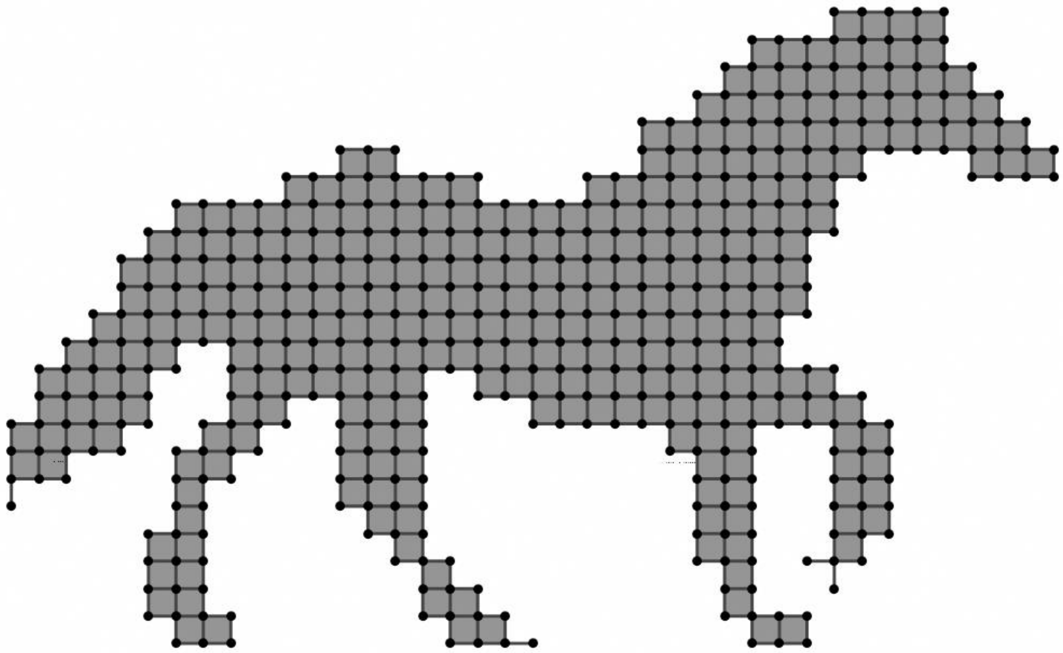
- Removing all simple pairs in parallel at each iteration
  - Only the topology of the cell complex is preserved



3D example

Exhaustive Thinning

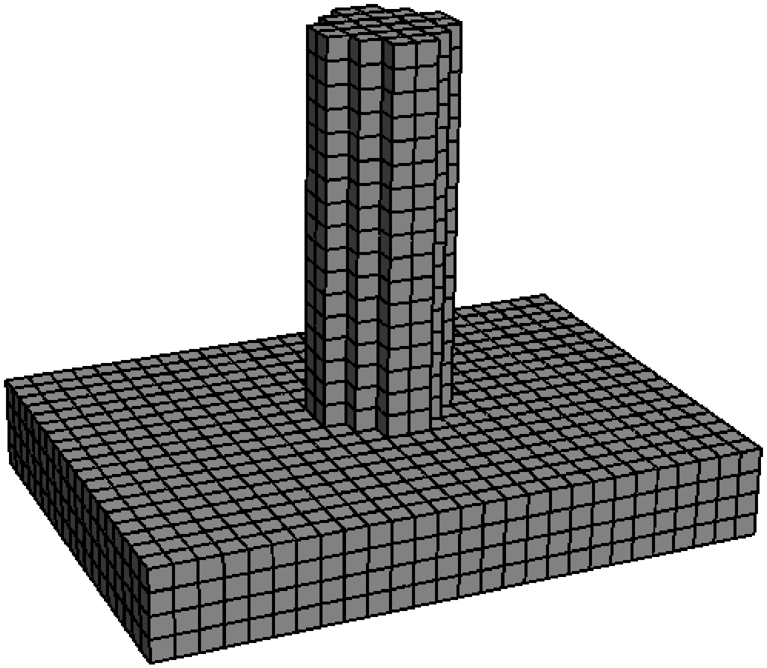
- Removing all simple pairs in parallel at each iteration
  - Only the topology of the cell complex is preserved



A more interesting 2D shape

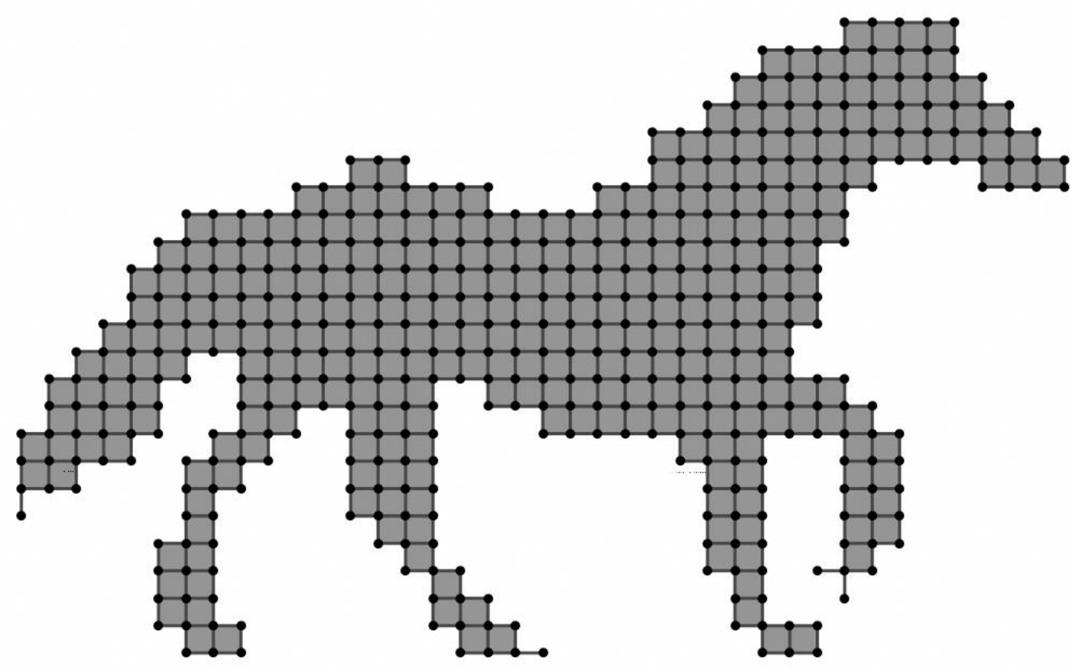
Exhaustive Thinning

- Removing all simple pairs in parallel at each iteration
  - Only the topology of the cell complex is preserved



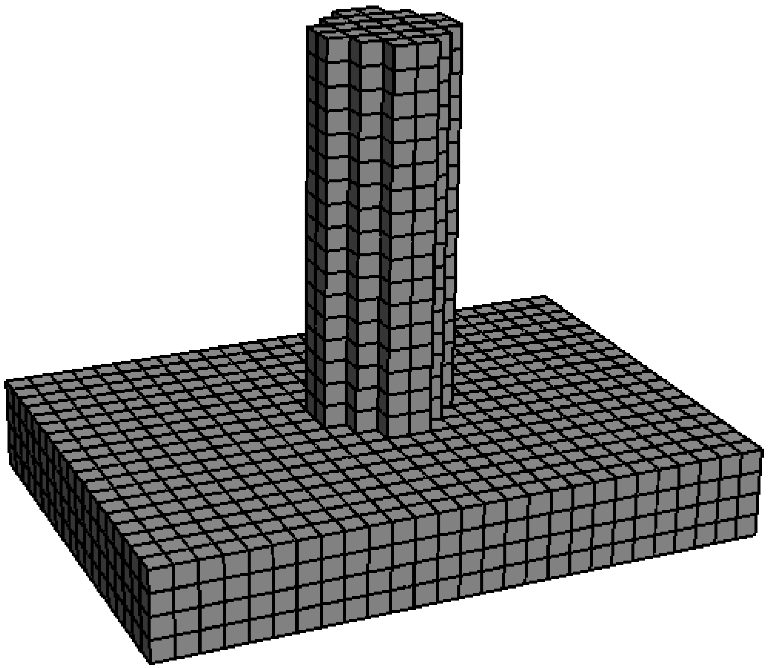
A 3D shape

Medial Cells (2D)



“Meaningful” skeleton edges survive longer during thinning

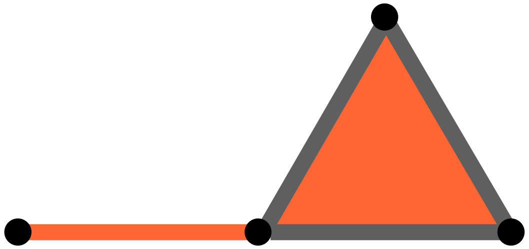
Medial Cells (3D)



“Meaningful” skeleton edges and faces survive longer during thinning

Isolated cells

- A cell **x** is isolated if it is not on the boundary of other cells
  - A k-dimensional skeleton is made up of isolated k-cells

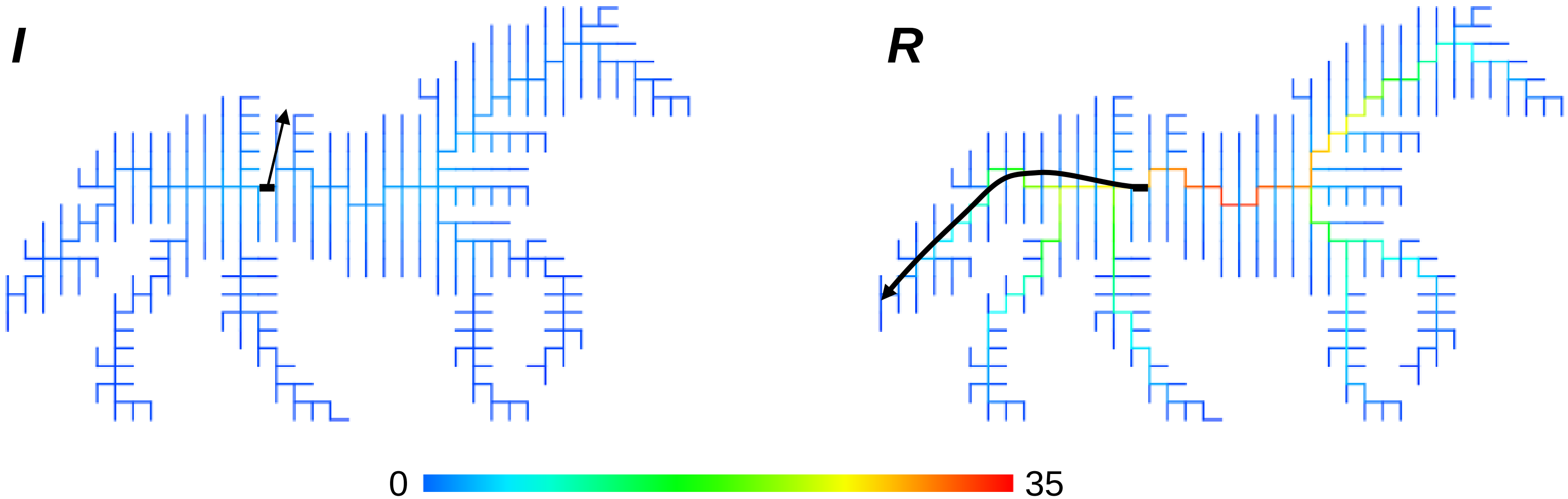


Isolated cells are colored

Medial Cells (2D)

Washington University in St.Louis
School of Engineering & Applied Science

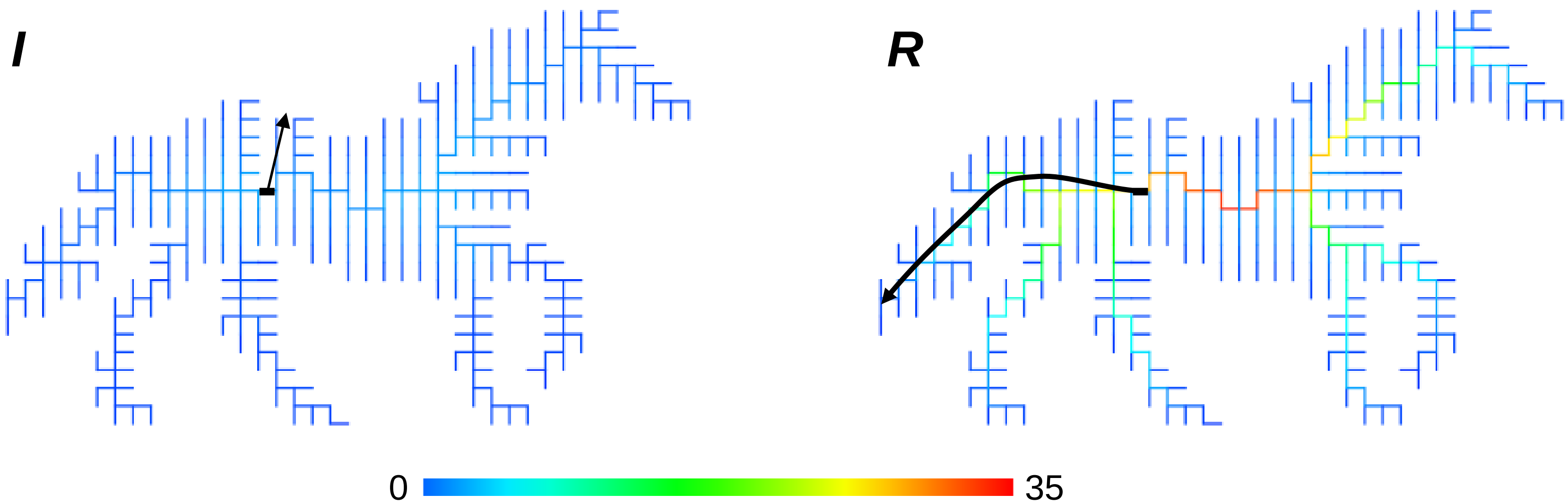
- Isolation iteration (**I(x)**): # thinning iterations before cell is isolated
  - Measures "thickness" of shape
- Removal iteration (**R(x)**): # thinning iterations before cell is removed
  - Measures "length" of shape



Medial Cells (2D)

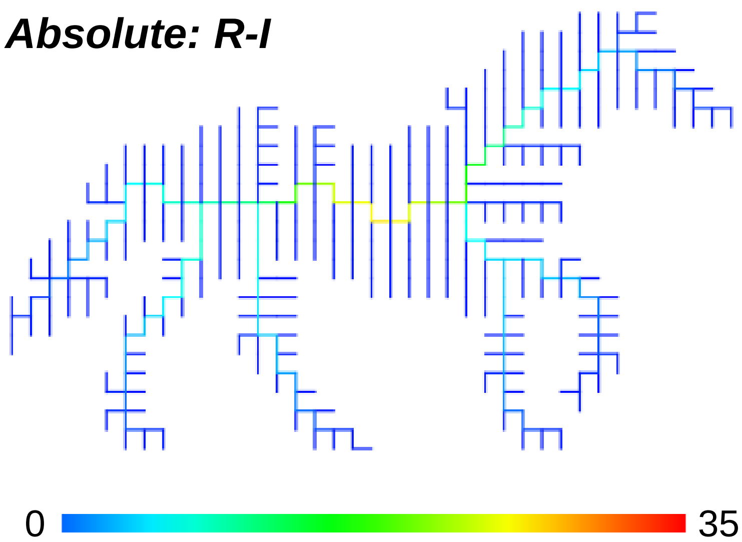
Washington University in St.Louis
School of Engineering & Applied Science

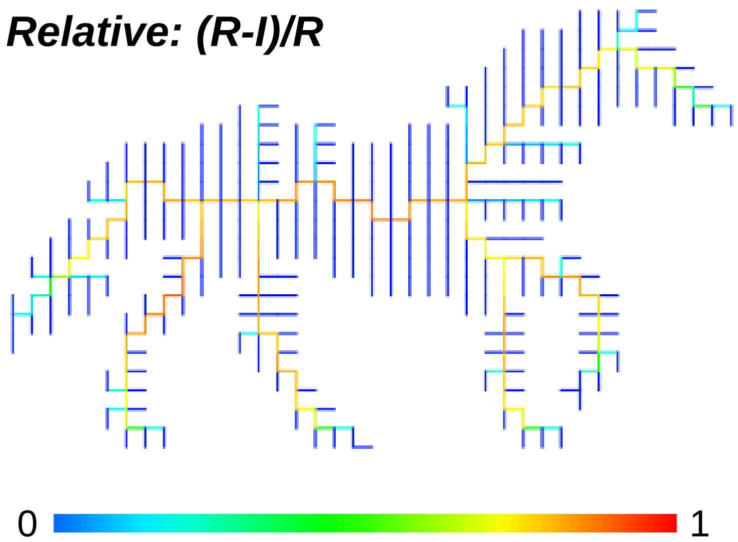
- Medial-ness: difference between **R**(x) and **I**(x)
  - A greater difference means the shape around x is more tubular



Medial Cells (2D)

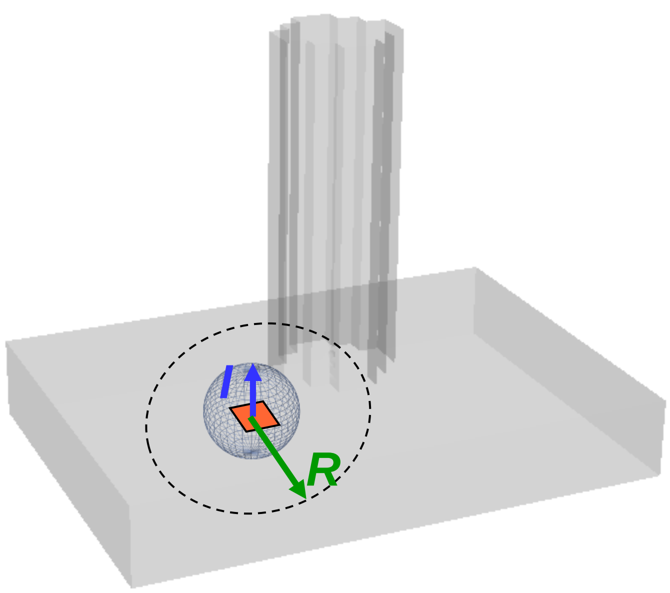
- Medial-ness: difference between **R**(x) and **I**(x)
  - A greater difference means the shape around x is more tubular

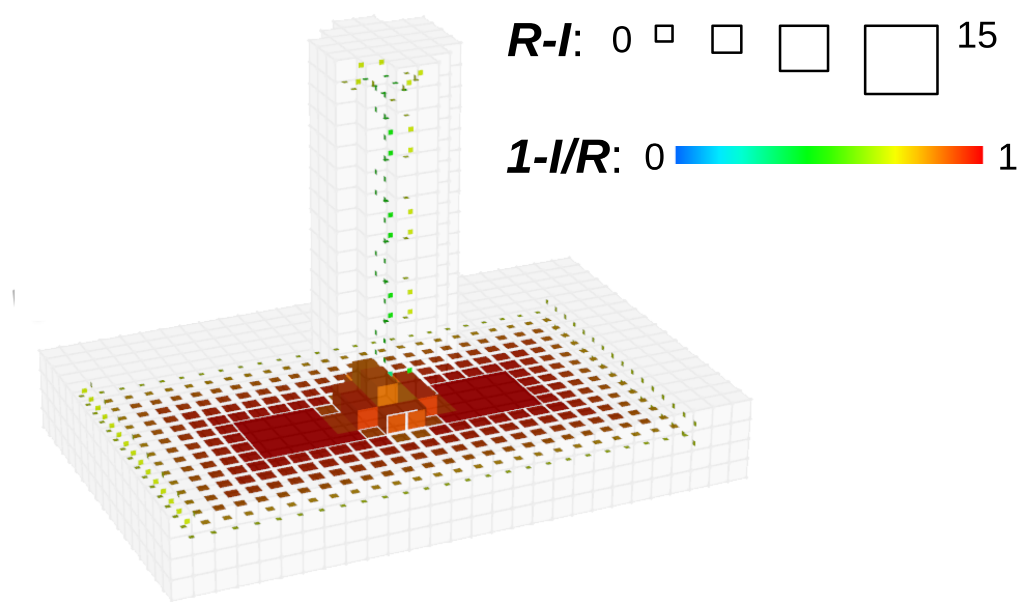




Medial Cells (3D)

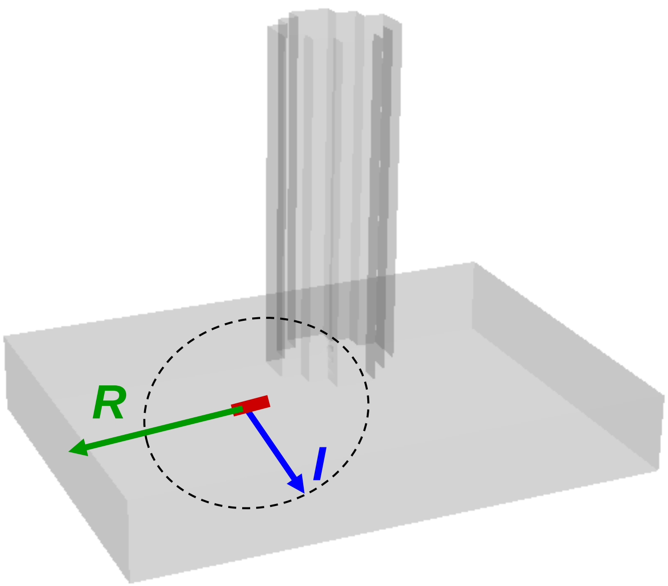
- For a 2-cell **x** that is isolated during thinning:
  - **I(x)**, **R(x)** measures the “thickness” and “width” of shape
  - A greater difference means the local shape is more “plate-like”

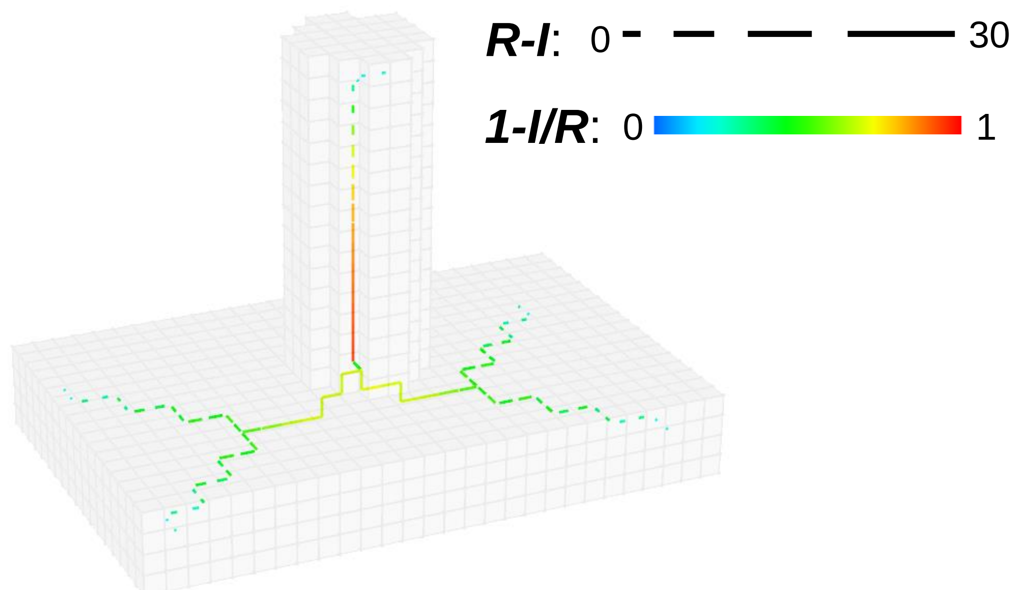




Medial Cells (3D)

- For a 1-cell **x** that is isolated during thinning:
  - **I(x)**, **R(x)** measures the “width” and “length” of shape
  - A greater difference means the local shape is more “tubular”





Medial Cells and Thinning

- A cell **x** is a medial cell if it is isolated and the difference between R(x) and I(x) exceeds given thresholds
  - A pair of absolute/relative difference thresholds is needed for medial cells at each dimension
    - 2D: thresholds for medial 1-cells
      - $t1_{abs}$, $t1_{rel}$
    - 3D: thresholds for both medial 1-cells and 2-cells
      - $t1_{abs}$, $t1_{rel}$
      - $t2_{abs}$, $t2_{rel}$
- Thinning: removing simple pairs that are not medial cells
  - Note: only need to check the simple cell in a pair (the witness is never isolated)

Thinning Algorithm (2D)


```
// Thinning on a 2D cell complex C
// Thresholds t1_abs and t1_rel for medial 1-cells
1. k = 1
2. For all x in C, set I(x) be 0 if x is isolated, NULL otherwise
3. Repeat and increment k:
     1. Let S be all disjoint simple pairs in C
     2. Repeat for each pair {x,y} in S:
          1. If x is 1-cell and ( k-I(x)>t1_abs and 1-I(x)/k>t1_rel ),
             exclude {x,y} from S.
     3. If S is empty, Break.
     4. Remove all cells in S from C
     5. Set I(x) be k for newly isolated cells x in C
4. Output C
```

Current iteration

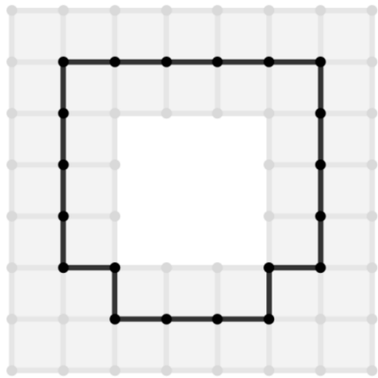
Thinning Algorithm (3D)

```
// Thinning on a 3D cell complex C
// Thresholds t1_abs and t1_rel for medial 1-cells
// Thresholds t2_abs and t2_rel for medial 2-cells
1. k = 1
2. For all x in C, set I(x) be 0 if x is isolated, NULL otherwise
3. Repeat and increment k:
    1. Let S be all disjoint simple pairs in C
    2. Repeat for each pair {x,y} in S:
        1. If x is 1-cell and ( k-I(x)>t1_abs and 1-I(x)/k>t1_rel ),
           exclude {x,y} from S.
        2. If x is 2-cell and ( k-I(x)>t2_abs and 1-I(x)/k>t2_rel ),
           exclude {x,y} from S.
    3. If S is empty, Break.
    4. Remove all cells in S from C
    5. Set I(x) be k for newly isolated cells x in C
4. Output C
```

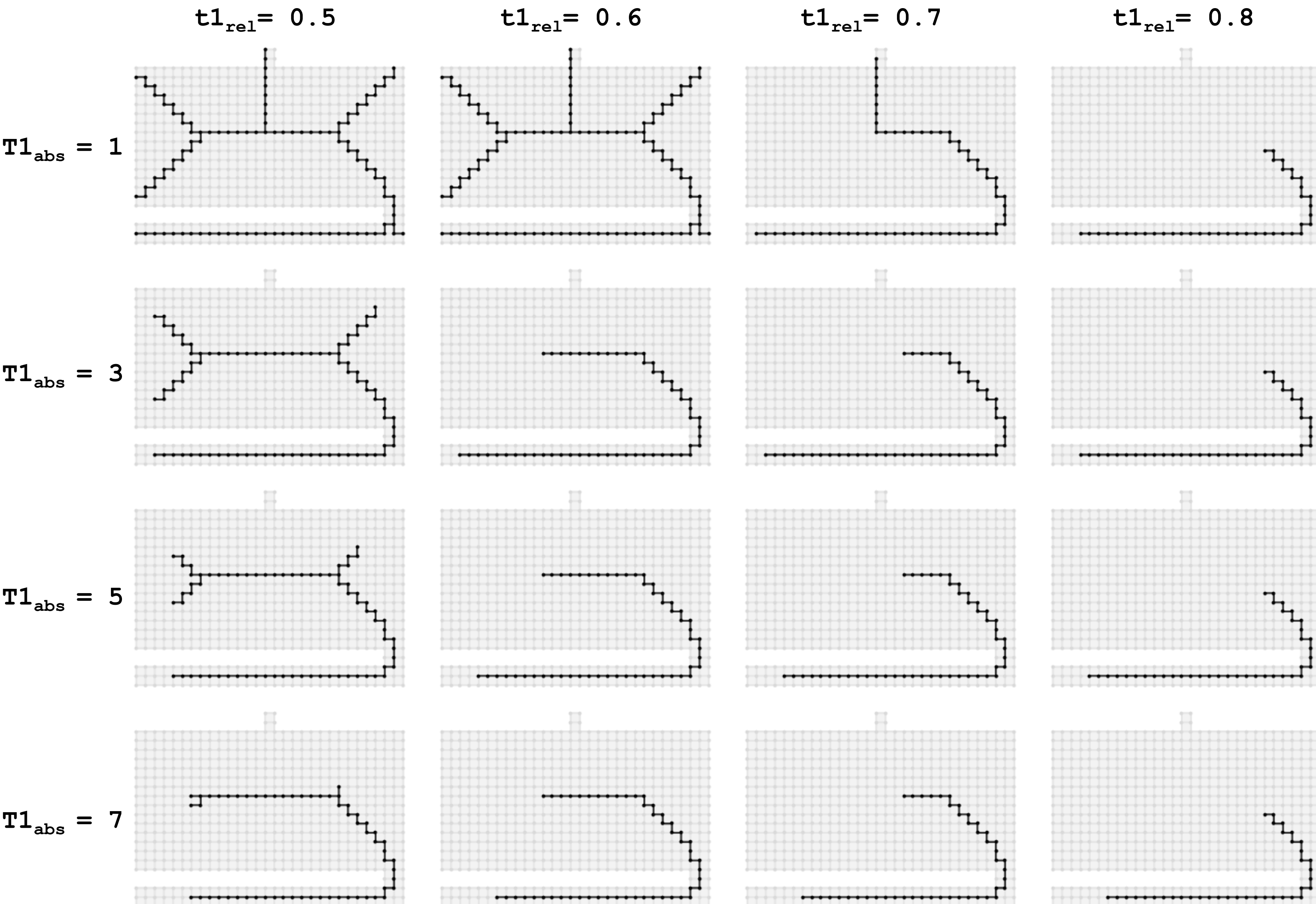
Current iteration

Choosing Thresholds

- Higher thresholds result in smaller skeleton
  - Threshold the absolute difference at ∞ will generally purge all cells at that dimension
    - Except those for keeping the topology
  - Absolute threshold has more impact on features at small scales (e.g., noise)
  - Relative threshold has more impact on rounded features (e.g., blunt corners)

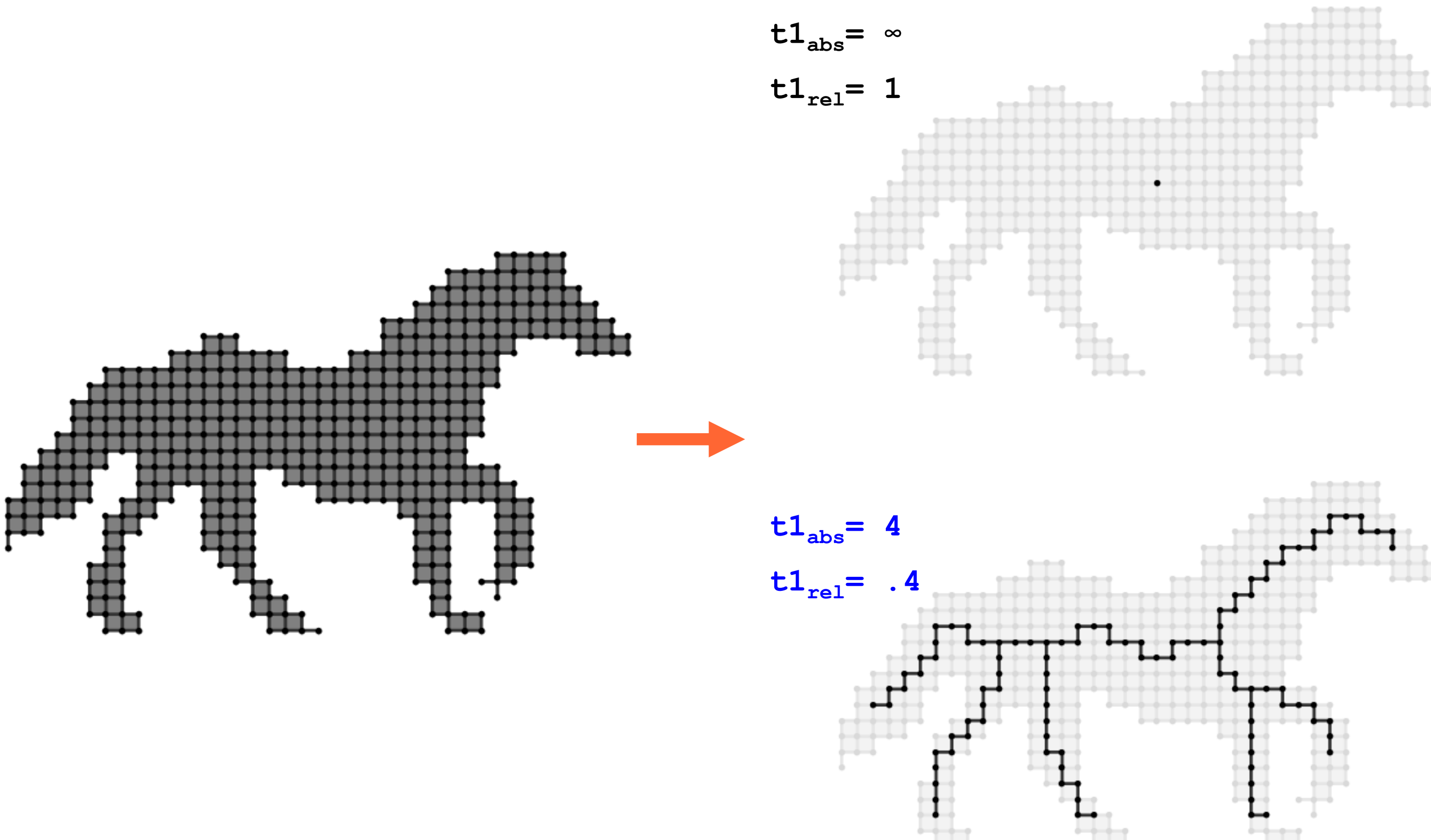


Skeletons computed at threshold $\mathtt{t1}_{\mathtt{abs}} = \infty$


t1rel= 0.5
t1rel= 0.6
t1rel= 0.7
t1rel= 0.8
T1abs = 1
T1abs = 3
T1abs = 5
T1abs = 7

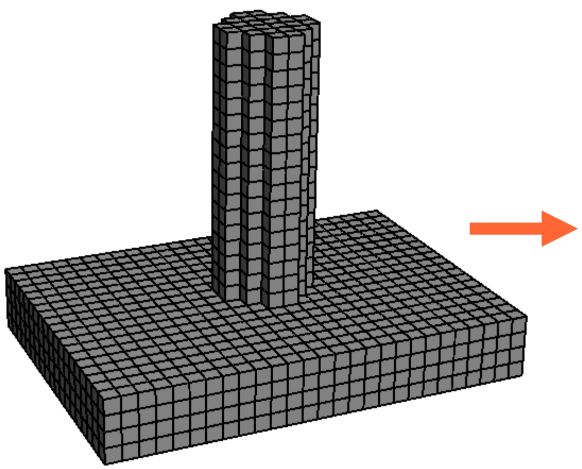
More Examples

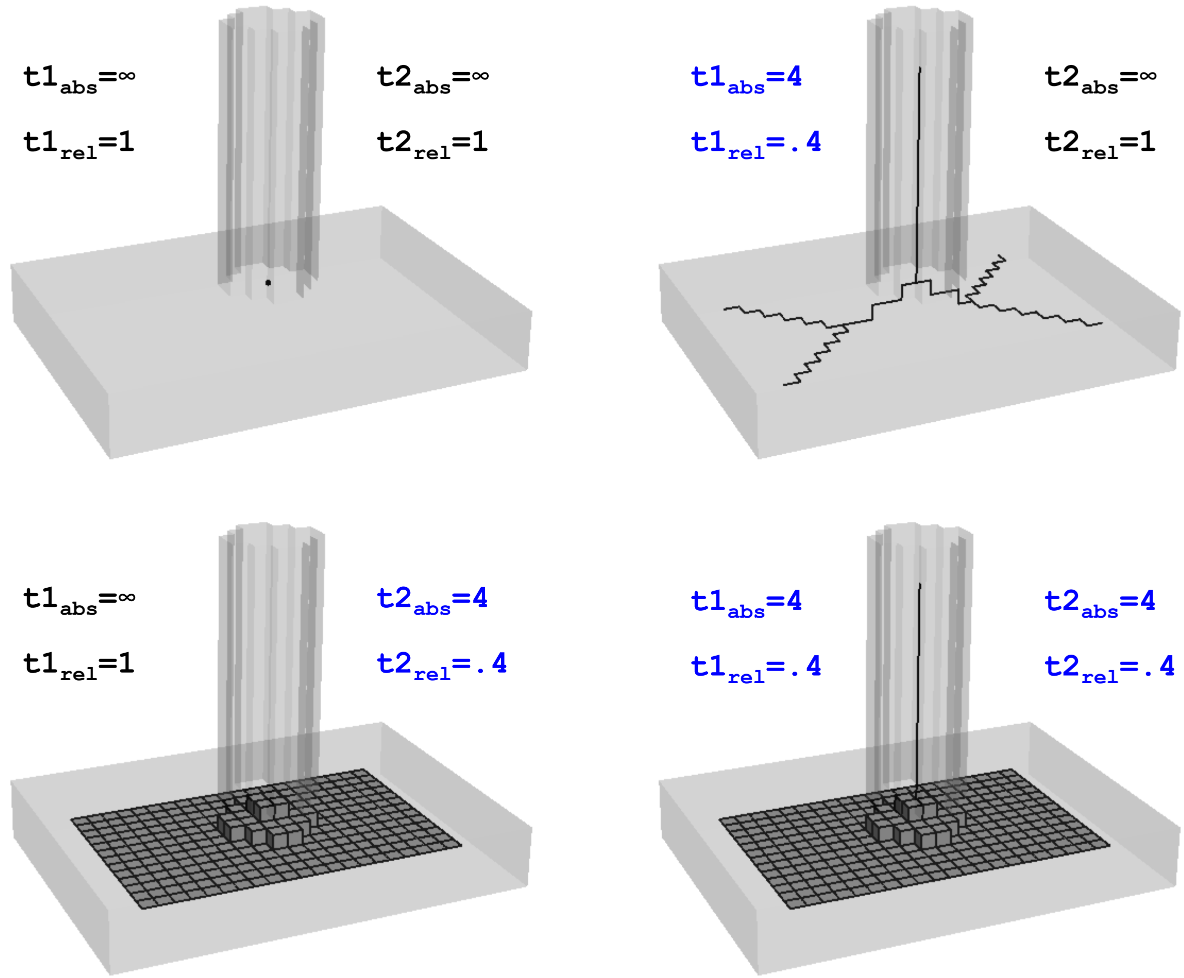
- 2D



More Examples

- 3D

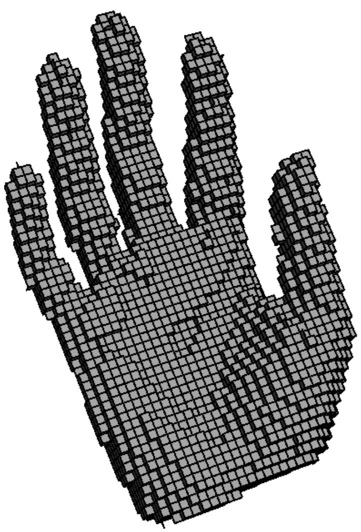




More Examples

Washington University in St. Louis
School of Engineering & Applied Science

- 3D

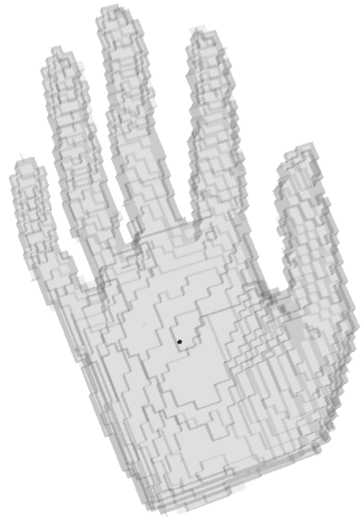


$t1_{abs}=\infty$

$t1_{rel}=1$

$t2_{abs}=\infty$

$t2_{rel}=1$

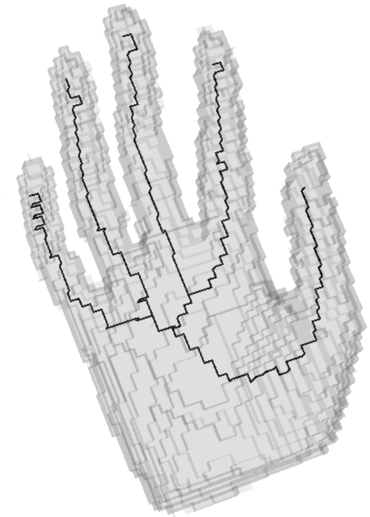


$t1_{abs}=5$

$t1_{rel}=.4$

$t2_{abs}=\infty$

$t2_{rel}=1$

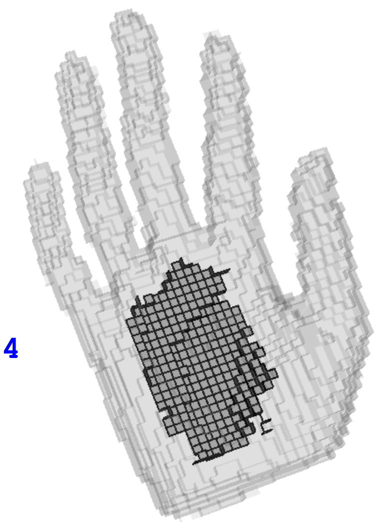


$t1_{abs}=\infty$

$t1_{rel}=1$

$t2_{abs}=5$

$t2_{rel}=.4$



$t1_{abs}=5$

$t1_{rel}=.4$

$t2_{abs}=5$

$t2_{rel}=.4$