

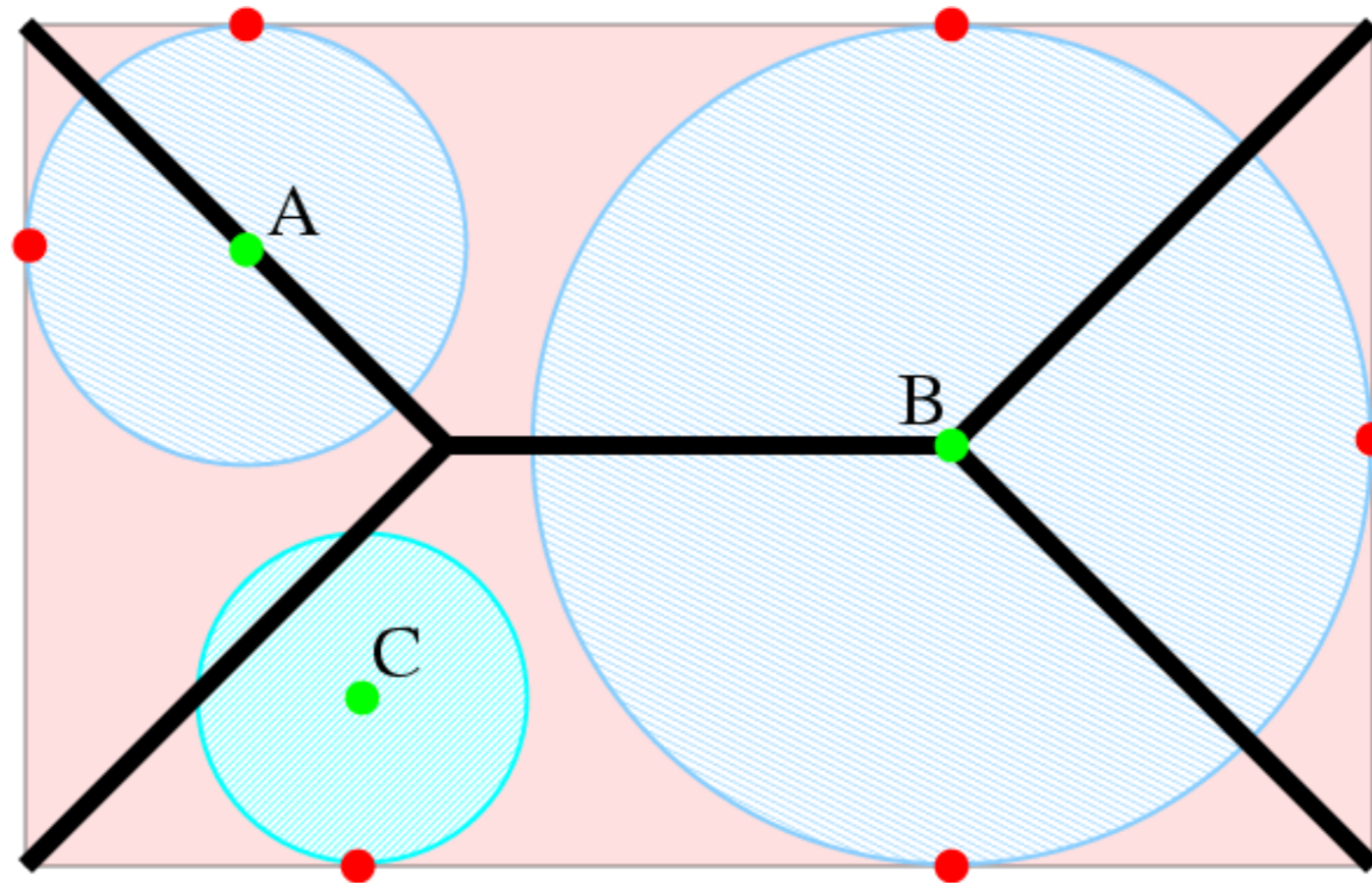
Medial Surface Extraction for 3D Shape Representation

di Rossi Luca

Università Ca' Foscari di Venezia
A.A. 2009/2010

Skeleton: Definition

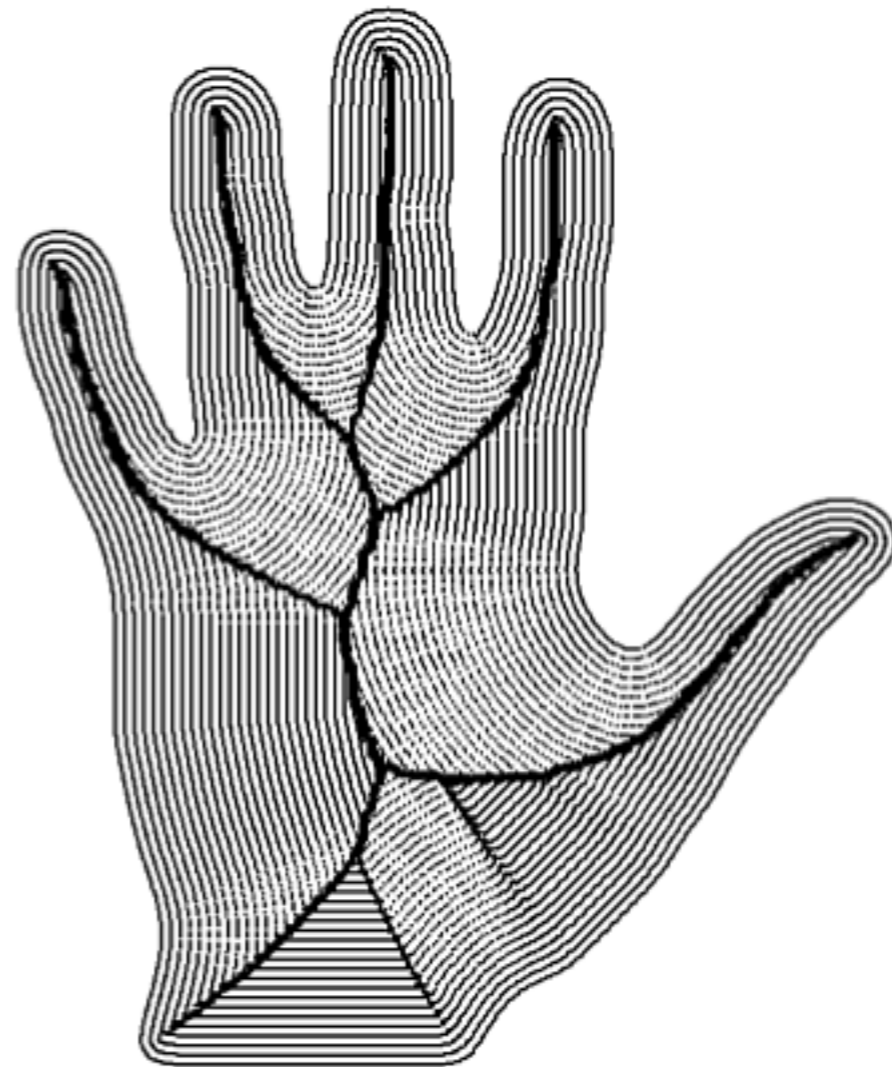
- In the seminal paper of Blum the skeleton is defined as the locus of the center of the maximal inscribed circles



Skeleton: Definition

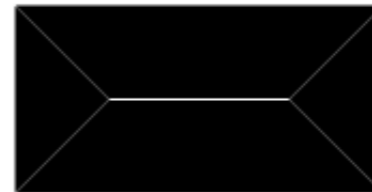
- The skeleton can be also defined as the set of singularities of the inward evolution of the boundary of the shape
- Described by the eikonal equation

$$\frac{\partial}{\partial t} \vec{C}(t) = v \vec{N}(t)$$



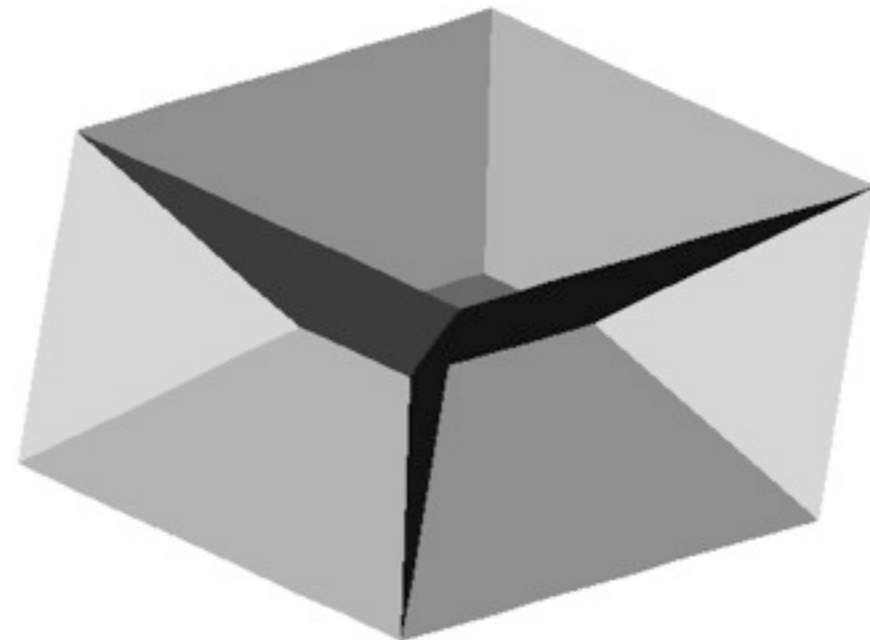
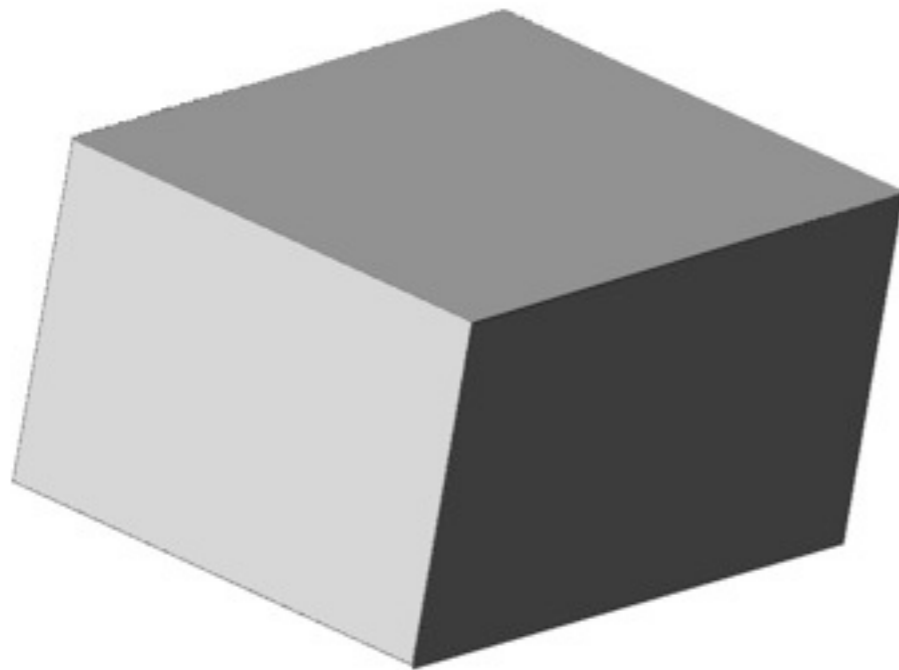
Skeleton: Definition

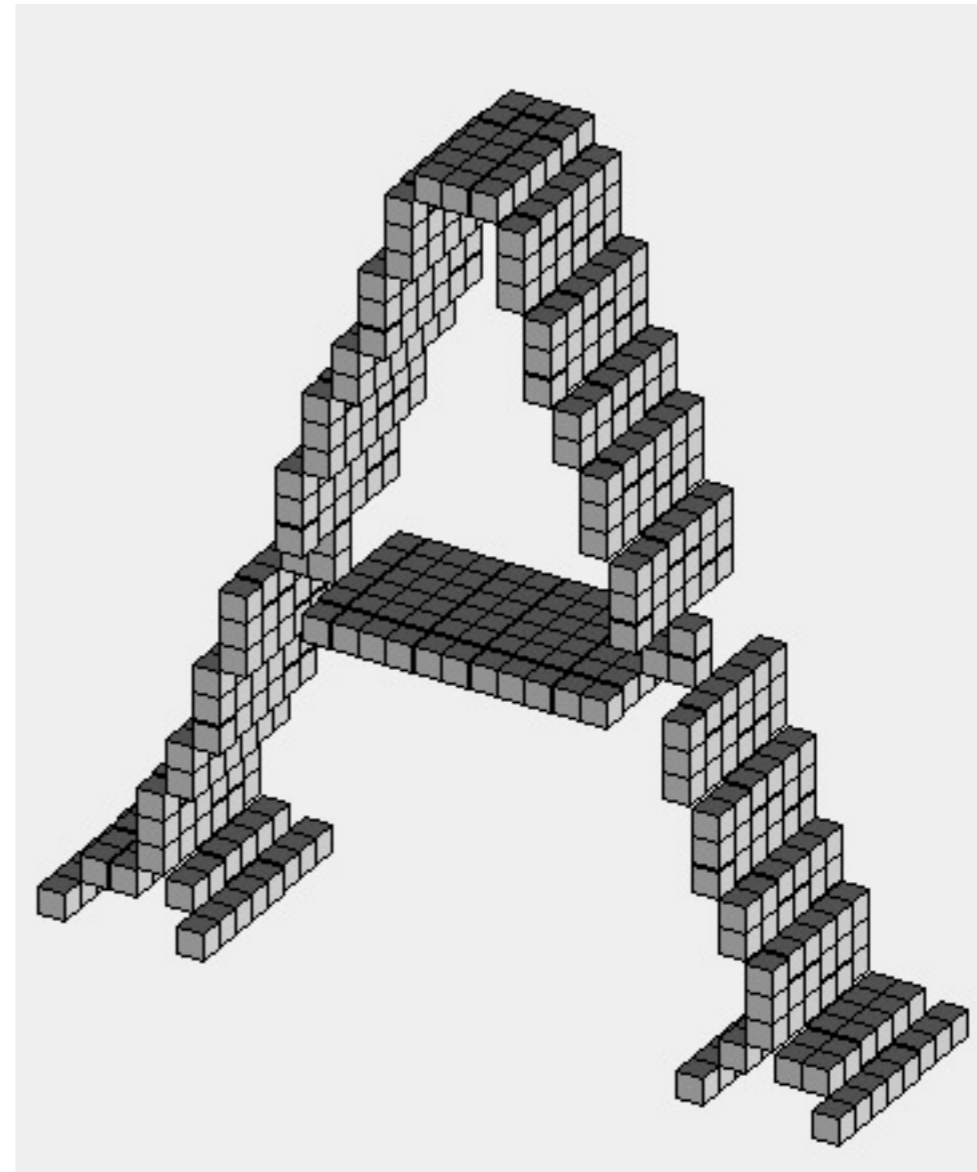
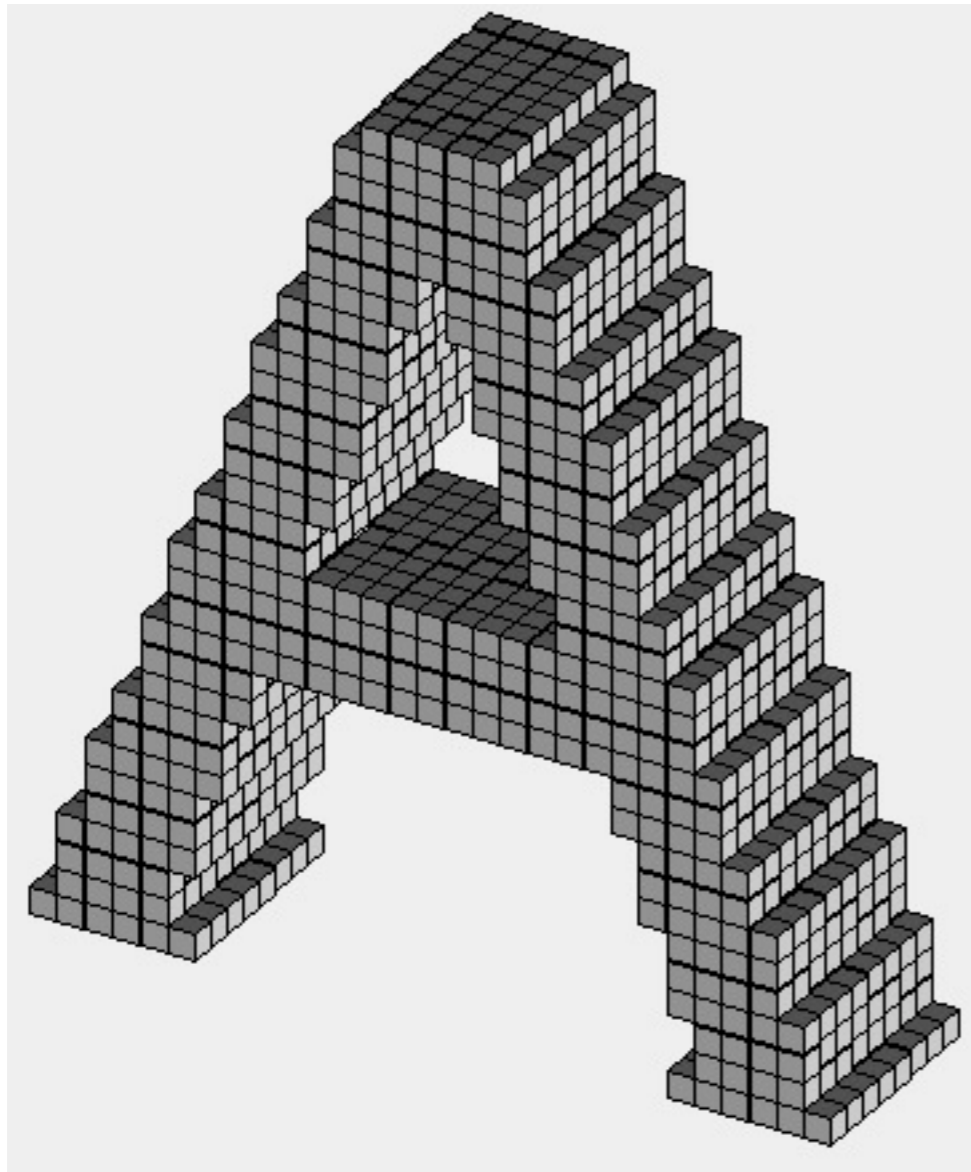
- Alternatively the skeleton can be defined as the set of ridge points of the distance map.
- The distance map assigns to each point in the interior of the shape its distance to the closest boundary points



Medial Surfaces: Definition

- Similarly to the 2D skeleton, the 3D skeleton is defined as the locus of the centers of the maximal inscribed spheres



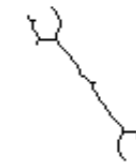
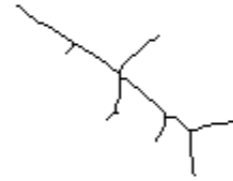
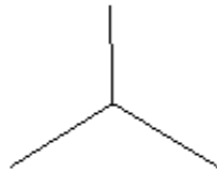
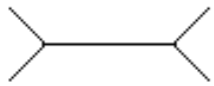


Medial Surfaces

The skeleton of a three-dimensional object

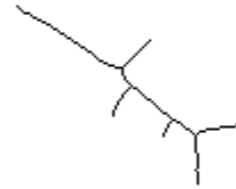
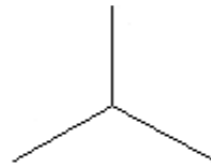
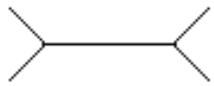
Skeleton: properties

- The properties of the skeleton make it an interesting shape descriptor
 - it is a thin set
 - it is homotopic to the original shape
 - it is invariant under Euclidean transformations
 - it is possible to reconstruct the original shape



Skeleton: Examples

extracted using MATLAB bwmorph operator



Skeleton: Examples

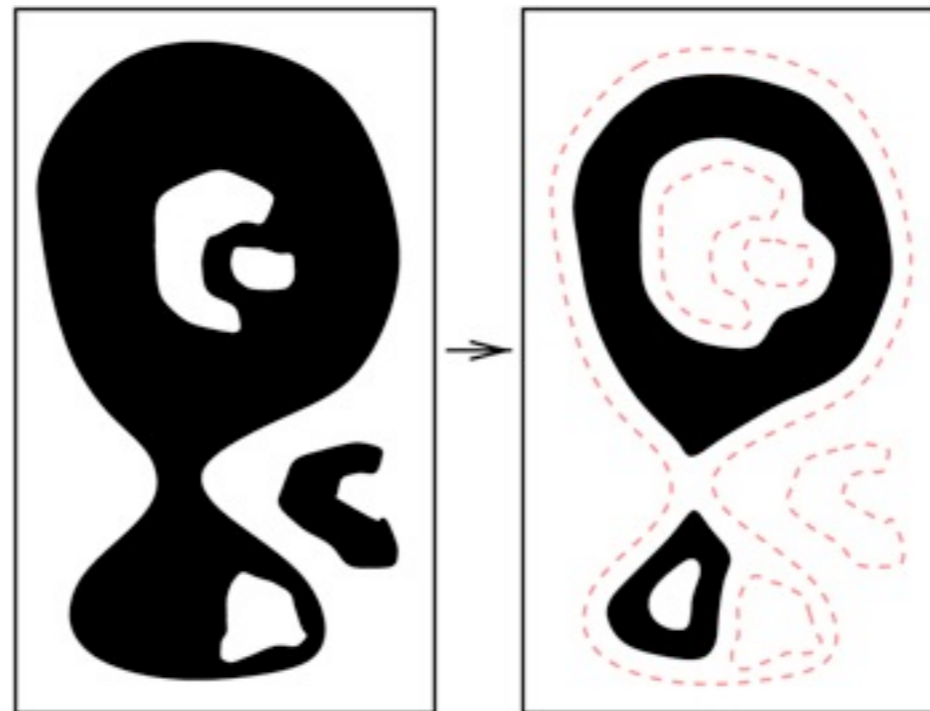
extracted using Torsello and Hancock algorithm

Extraction Algorithms

- There are three main classes of algorithms for skeleton extraction
 - Thinning methods
 - Distance transform methods
 - Voronoi diagram methods

Thinning Methods

- Simulate Blum's grassfire transform
- Iteratively peel away layers from the shape
- **Must preserve object topology!**



Thinning Methods: pros

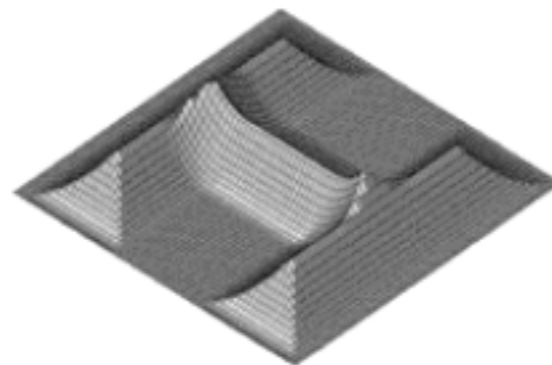
- **Simple** to implement
- **Fast** execution

Thinning Methods: cons

- The result depends on the order in which the pixels are eroded
- The extracted skeleton is **not guaranteed to be geometrically correct**
 - It might not lie in the middle of the shape
- Not invariant under Euclidean transformations!

Distance Transform Methods

- Exploit the definition based on DM
- The skeleton is composed of the ridges of the DM
 - Can be computed in linear time!

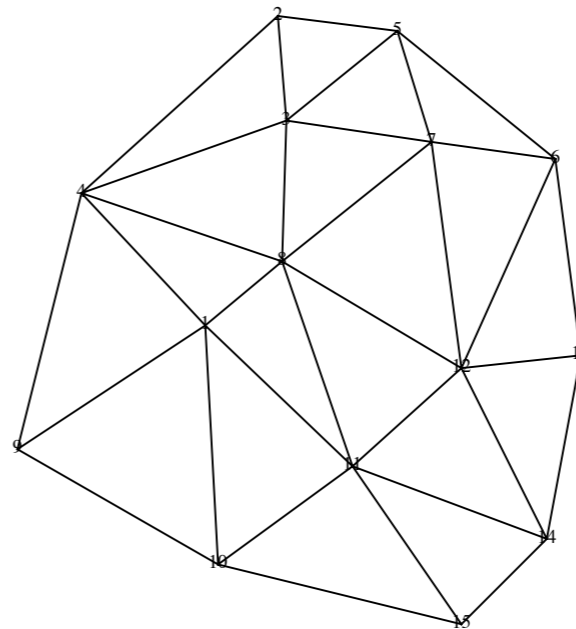
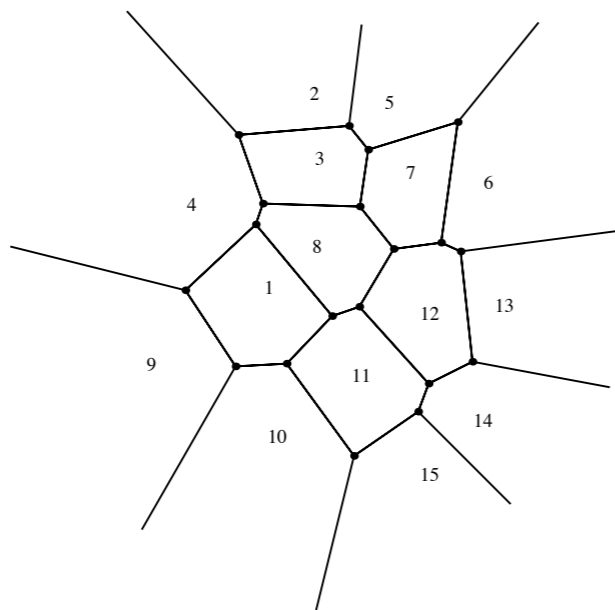


Distance Transform Methods: cons

- No guarantee of topology preservation!
- Locating the ridges is itself a challenging problem...

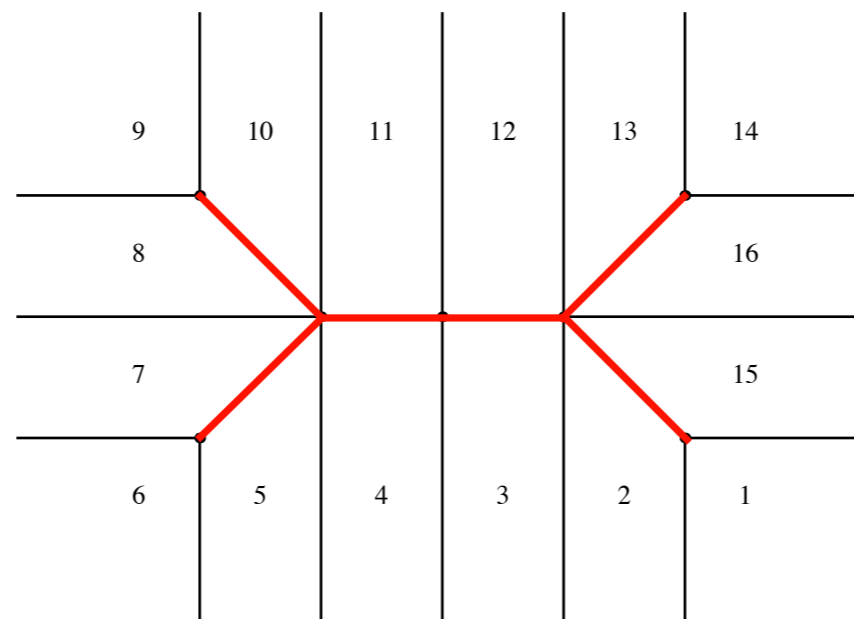
Voronoi Diagram Methods

- Voronoi diagram: a partition of the space into regions so that each region contains a generating point and the set of points which are closer to it than to any other generating points
- The dual of the VD is the Delaunay triangulation



Voronoi Diagram Methods

- The idea is that if enough boundary points are sampled their Voronoi Diagram converges to the skeleton of the shape



Voronoi Diagram Methods: pros

- Ensures topology preservation
- Provides invariance under Euclidean transformations

Voronoi Diagram Methods: cons

- Computing the VD or the DT requires a **lot of time**, if the shape is not a polygon!
- Spurious branches need to be pruned!

Hamilton Jacobi Skeleton

- Siddiqi *et al.* perform a Hamiltonian analysis of the boundary flow dynamics (eikonal equation)
- They propose to identify the points where the divergence of the velocity field is nonzero as the skeletal points
- The divergence measures the tendency of a given point to behave like a source or a sink

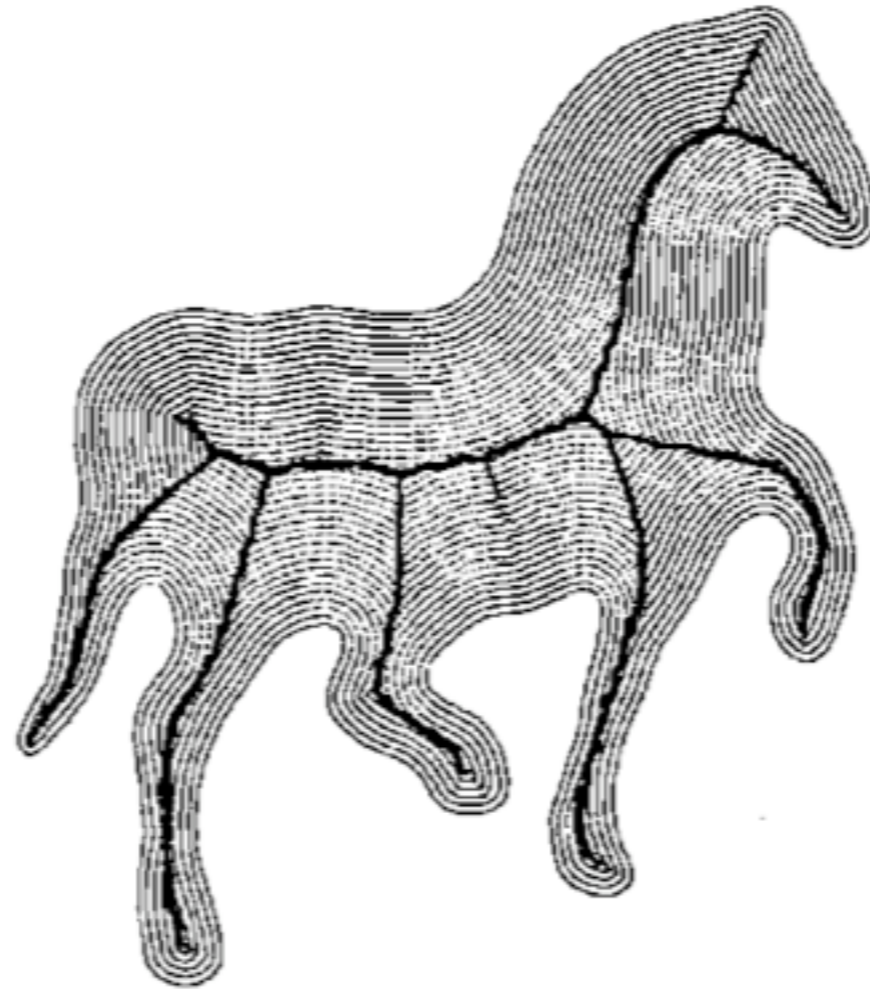
Hamilton Jacobi Skeleton

- Distance Map
 - A function that assigns to each point in the interior of a shape the distance to the closest point on the boundary of the object
- Velocity Field (F)
 - The gradient of the distance map, which governs the trajectory of the boundary points during the boundary evolution

Hamilton Jacobi Skeleton

- **Assumption:** this dynamic system is hamiltonian everywhere except on the skeleton
 - \Rightarrow the flux of F is conservative everywhere except on skeletal locations, i.e. the net flux is zero
- **Idea:** find the *sinks* by looking at the divergence of the velocity field

$$\nabla \cdot \vec{F} = \lim_{|A| \rightarrow 0} \frac{\int_{\partial A} \vec{F} \cdot \vec{n} \, dl}{|A|} = \lim_{|A| \rightarrow 0} \frac{\phi_A(\vec{F})}{|A|}$$



The boundary flow

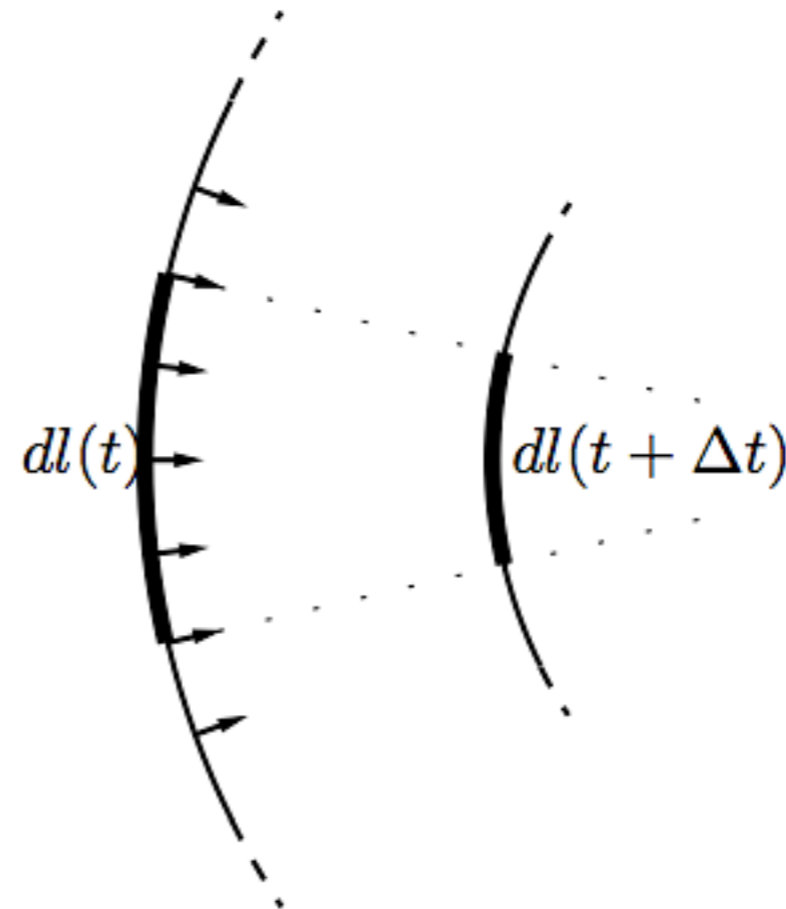
Singularities correspond to skeletal locations

Hamilton Jacobi Skeleton

- Unfortunately the velocity field is not conservative
- Analogy with compressible fluid
- Siddiqi *et al.* introduce the normalized flux...

$$N\phi_A(\vec{F}) = \lim_{r \rightarrow 0} \frac{\phi_A(\vec{F})}{2\pi r}$$

- ...but it's still not enough.



Correcting the Curvature-Density

- The problem is that we're not taking into account the effect of the **changes in density**
- Torsello and Hancock solve this problem by defining the **Momentum Field**

$$\vec{M} = \rho \vec{F}$$

- This is indeed a conservative field, i.e.

$$\nabla \cdot (\rho \vec{F}) = 0$$

Computing the Momentum Field

- From $\nabla \cdot (\rho \vec{F}) = 0$ we obtain $\nabla \sigma \cdot \vec{F} = -\nabla \cdot \vec{F}$, which is equivalent to

$$\begin{cases} \frac{\partial}{\partial t} \sigma(s(t)) = -\nabla \cdot \vec{F}(s(t)) \\ \frac{\partial}{\partial t} s(t) = \vec{F}(s(t)) \end{cases}$$

- As we're working in a discrete setting, we need to find a discrete solution to the above equations
- But we need to be careful...

Computing the Momentum Field

- Since the skeleton is the set of the singularities of the momentum field, we expect the density to have different values at opposite sides of the skeleton
- Thus we cannot solve the previous system of ODE by means of a system of difference equations!
- ...what to do?

- **Never cross the skeleton!**



Computing the Momentum Field

- **Solution:** integrate the equation in the time domain!
- We use a second order Crank-Nicolson approximation to obtain

$$\sigma(\vec{C}(t, s)) = \sigma(\vec{C}(t-1, s)) - \frac{1}{2}[\nabla \cdot \vec{F}(\vec{C}(t, s)) + \nabla \cdot \vec{F}(\vec{C}(t-1, s))]$$

- ...what if we want to go back to the space domain?

Computing the Momentum Field

- We can approximate the position of a boundary point at time $t-1$ as

$$\vec{C}(t-1, s) = (x, y, z) - \vec{F}(x, y, z) = (x - F_x, y - F_y, z - F_z)$$

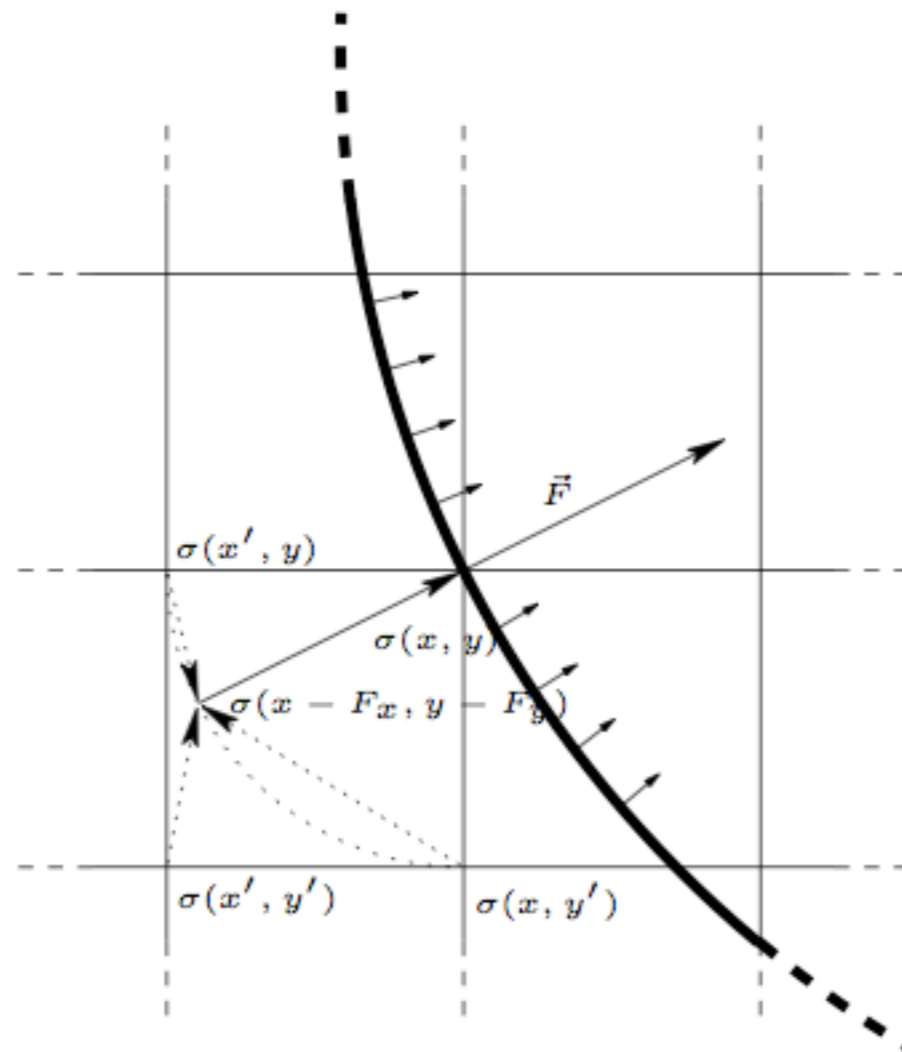
- Thus we obtain

$$\begin{aligned} \sigma(x, y, z) &= \sigma(x - F_x, y - F_y, z - F_z) \\ &\quad - \frac{1}{2}[\nabla \cdot \vec{F}(x, y, z) + \nabla \cdot \vec{F}(x - F_x, y - F_y, z - F_z)] \end{aligned}$$

- But what is $(x - F_x, y - F_y, z - F_z)$?

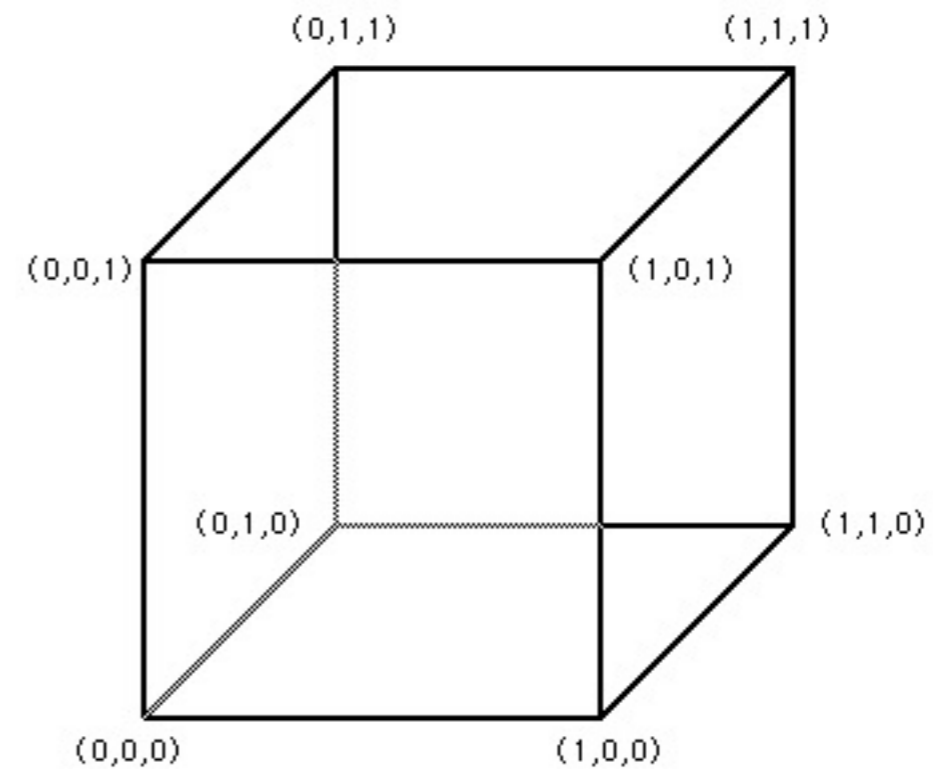
Computing the Momentum Field

- **Warning:** this point is not guaranteed to belong to the image lattice!
- We must interpolate it using the values of the vertices of the cube that contains it



Trilinear Interpolation

$$\begin{aligned} V_{xyz} = & V_{000} (1 - x) (1 - y) (1 - z) + \\ & V_{100} x (1 - y) (1 - z) + \\ & V_{010} (1 - x) y (1 - z) + \\ & V_{001} (1 - x) (1 - y) z + \\ & V_{101} x (1 - y) z + \\ & V_{011} (1 - x) y z + \\ & V_{110} x y (1 - z) + \\ & V_{111} x y z \end{aligned}$$



Trilinear Interpolation

$$\begin{aligned}\sigma(x - F_x, y - F_y, z - F_z) = & \\ & (1 - |F_x|)(1 - |F_y|)(1 - |F_z|)\sigma(x, y, z) \\ & + |F_x|(1 - |F_y|)(1 - |F_z|)\sigma(x', y, z) \\ & + |F_y|(1 - |F_x|)(1 - |F_z|)\sigma(x, y', z) \\ & + |F_z|(1 - |F_x|)(1 - |F_y|)\sigma(x, y, z') \\ & + |F_x||F_y|(1 - |F_z|)\sigma(x', y', z) \\ & + |F_x||F_z|(1 - |F_y|)\sigma(x', y, z') \\ & + |F_y||F_z|(1 - |F_x|)\sigma(x, y', z') \\ & + |F_x||F_y||F_z|\sigma(x', y', z')\end{aligned}$$

Computing the Momentum Field

$$\begin{aligned} & \sigma(x, y, z)(1 - (1 - |F_x|)(1 - |F_y|)(1 - |F_z|)) = \\ & \quad |F_x|(1 - |F_y|)(1 - |F_z|)\sigma(x', y, z) \\ & \quad + |F_y|(1 - |F_x|)(1 - |F_z|)\sigma(x, y', z) \\ & \quad + |F_z|(1 - |F_x|)(1 - |F_y|)\sigma(x, y, z') \\ & \quad + |F_x||F_y|(1 - |F_z|)\sigma(x', y', z) \\ & \quad + |F_x||F_z|(1 - |F_y|)\sigma(x', y, z') \\ & \quad + |F_y||F_z|(1 - |F_x|)\sigma(x, y', z') \\ & \quad + |F_x||F_y||F_z|\sigma(x', y', z') \\ & \quad - \frac{1}{2}[\nabla \cdot \vec{F}(x, y, z) + \nabla \cdot \vec{F}(x - F_x, y - F_y, z - F_z)] \end{aligned}$$

Divergence of the Momentum Field

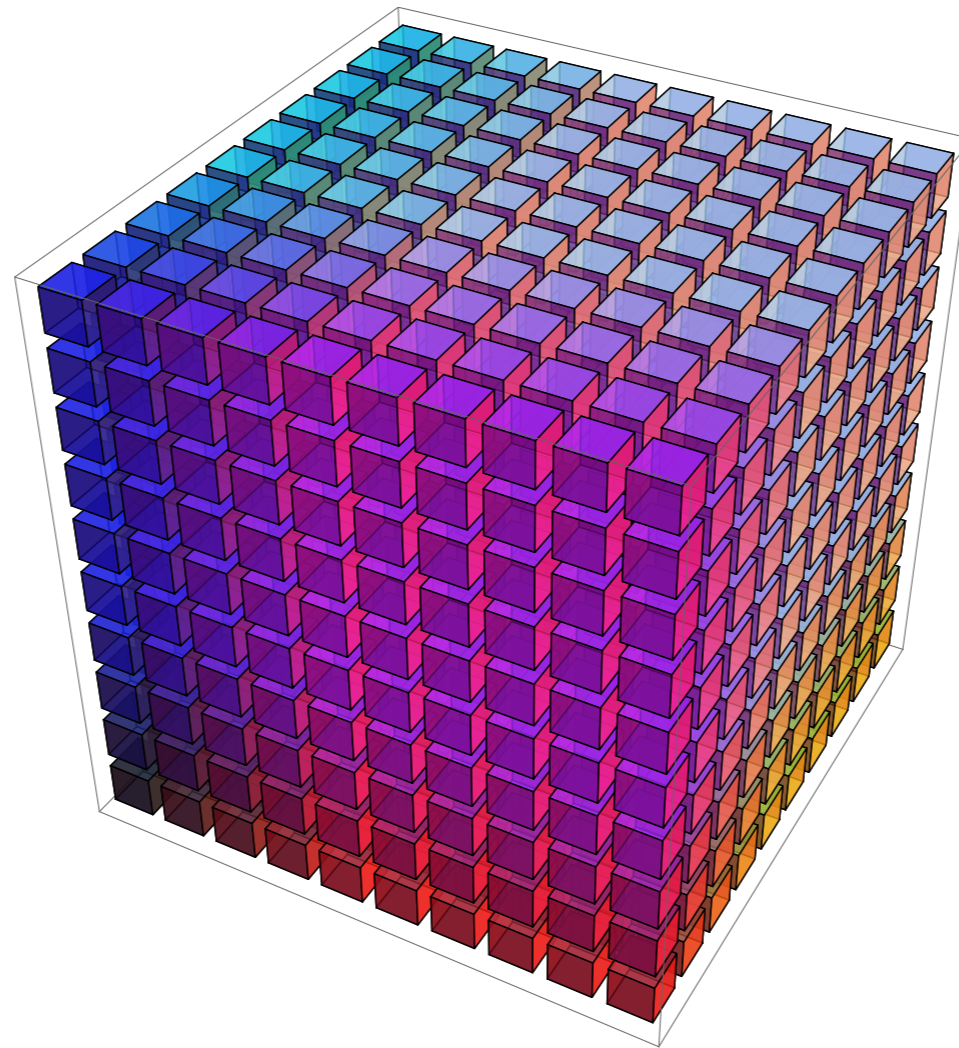
$$\begin{aligned}\nabla \cdot (\rho \vec{F})(x, y, z) = & \\ & [\sigma(x, y, z) - \sigma(x - F_x, y - F_y, z - F_z)]e^{\sigma(x, y, z) - \frac{1}{2}\Delta\sigma} \\ & + \frac{1}{2}[\nabla \cdot \vec{F}(x - F_x, y - F_y, z - F_z)e^{\sigma(x - F_x, y - F_y, z - F_z)} \\ & + \nabla \cdot \vec{F}(x, y, z)e^{\sigma(x, y, z)}]\end{aligned}$$

Divergence of the Momentum Field

- All we need to do is to iterate over the interior of the shape, starting from its boundary and computing first the density field and then the divergence of the momentum field.
- We can now proceed to the thinning step!
- But first....

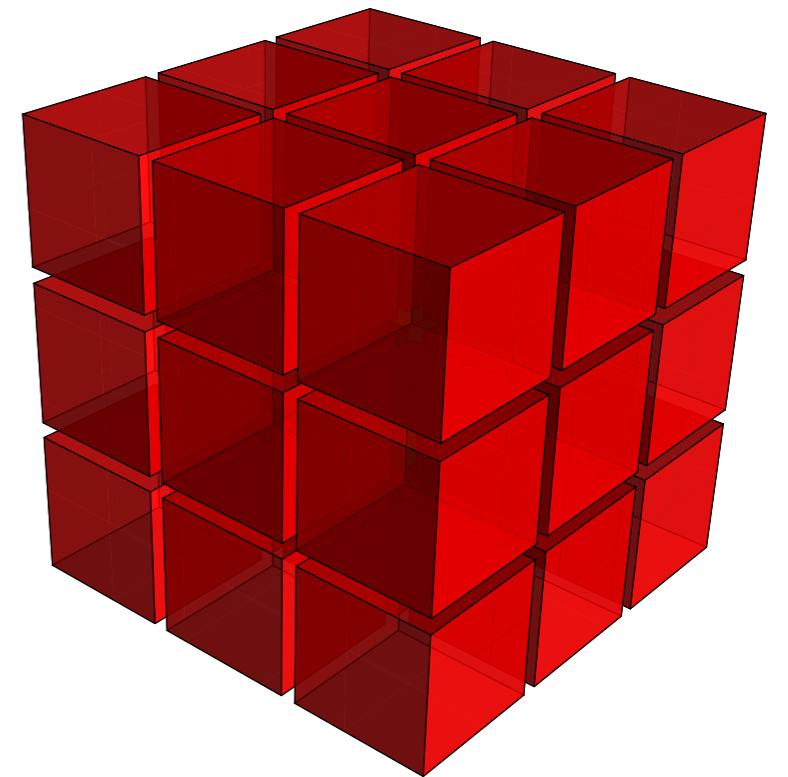
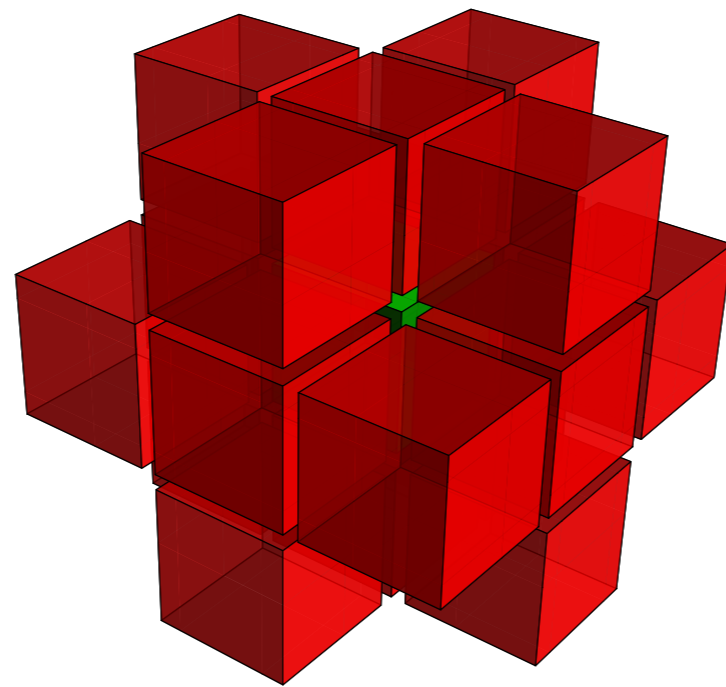
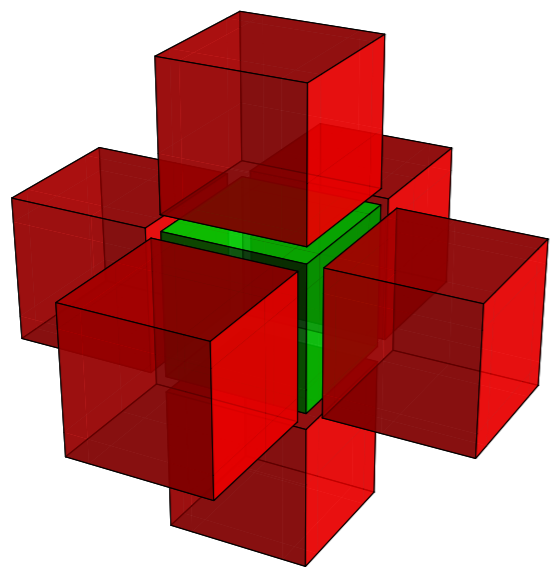
Digital Topology: Basic Concepts

- Consider a 3D object stored in a 3D binary array: 0 for background, 1 for the object.



Digital Topology: Basic Concepts

- The cube has 6 faces, 12 edges, 8 vertices
- two **6-neighbors** share a face
- two **18-neighbors** share a face or an edge
- two **26-neighbors** share a face or an edge or a vertex



6-neighborhood, 18-neighborhood and 26-neighborhood

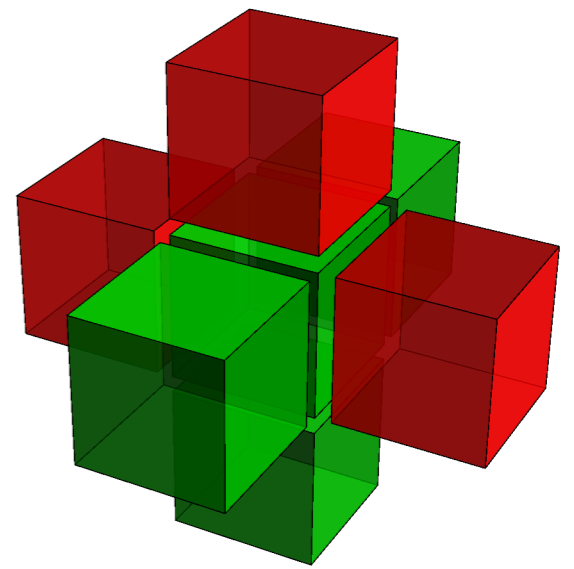
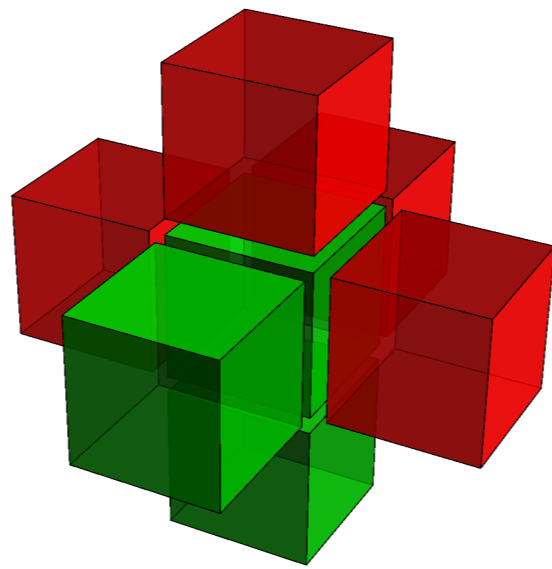
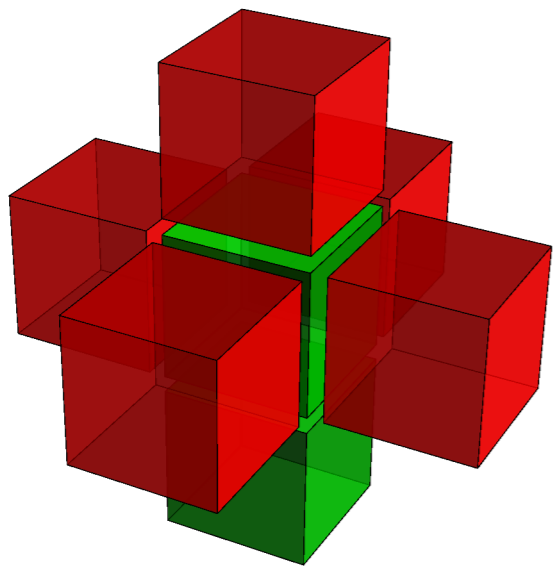
Digital Topology: Basic Concepts

- More definitions...
 - n-adjacent object
 - n-path between two points
 - n-connected object

Topological Classification

- Malandain et al. provide a topological classification of voxels based on:
 - C^* the number of 26-connected components in $B_n N^{*26}$
 - C^- the number of 6-connected components 6-adjacent to x in $W_n N_{18}$

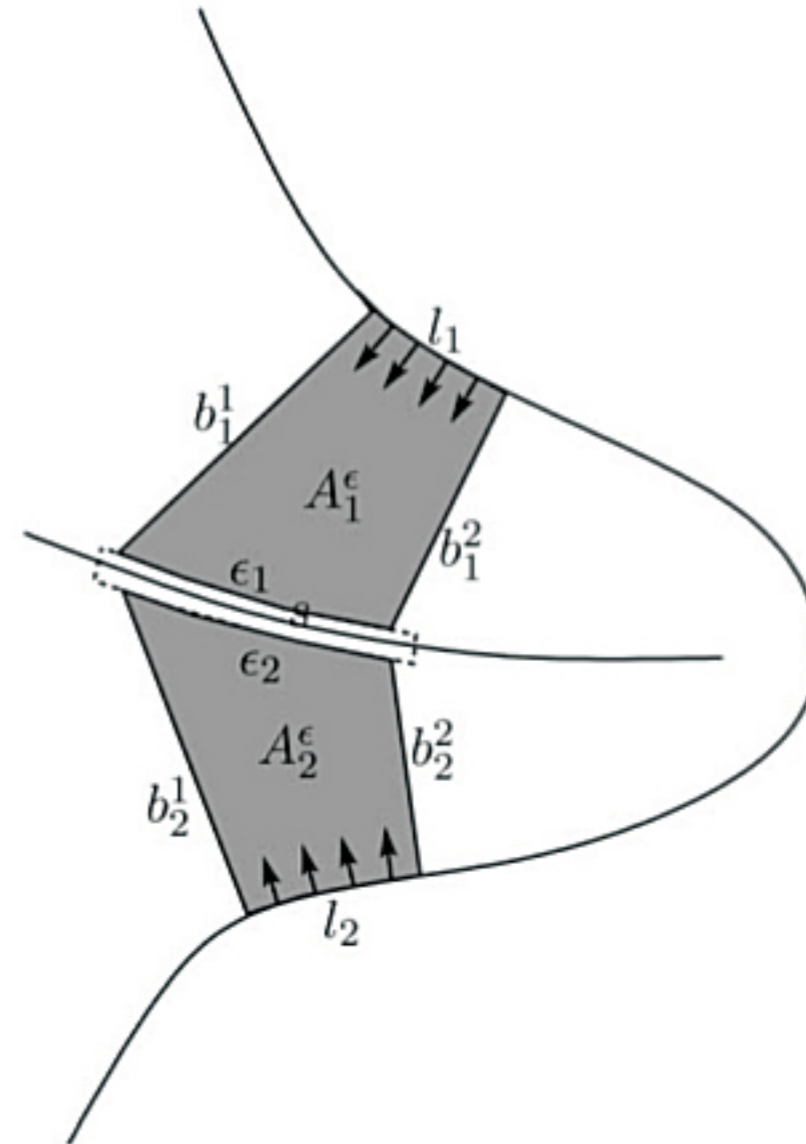
\bar{C}	C^*	TYPE
0	any	interior point
any	0	isolated point
1	1	border (simple) point
1	2	curve point
1	> 2	curves junction
2	1	surface point
2	> 2	surface-curve(s) junction
> 2	1	surfaces junction
> 2	≥ 2	surfaces-curves junction



Curve endpoint, Corner and Surface Endpoint

Divergence as a Skeletal Measure

- Divergence is negative at sinks points and it is proportional to the amount of the energy absorbed
- Moreover the divergence at skeletal location s can be shown to be proportional to dl/ds
- **IDEA**: thin the shape by removing points with decreasing divergence (down to a threshold), ensuring topology preservation



Divergence Based Thinning

For each point p in distance order

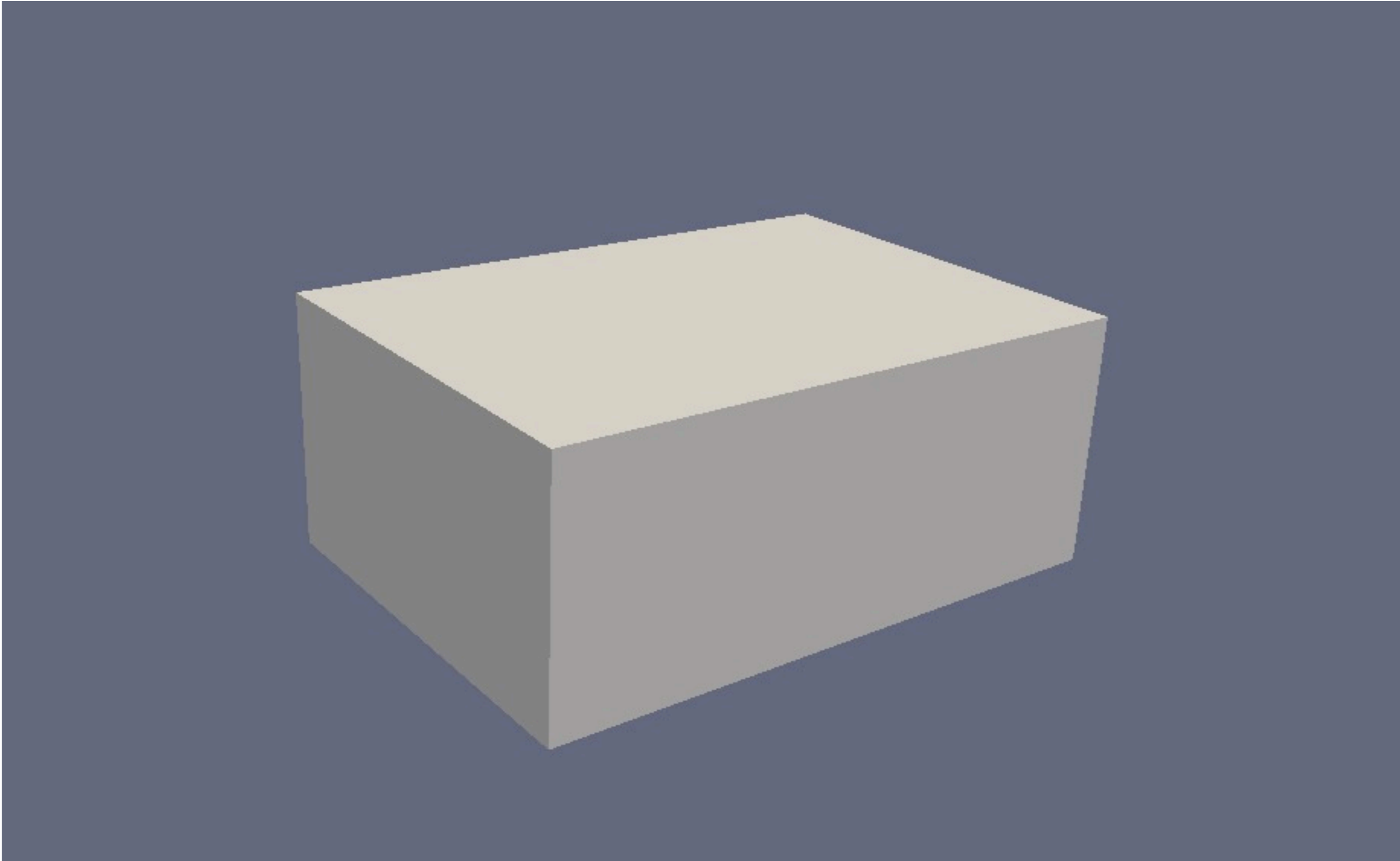
if `is_simple($S \setminus p$)` and $-\nabla \cdot \rho \vec{F}(p) < \epsilon$

then $S = S \setminus p$

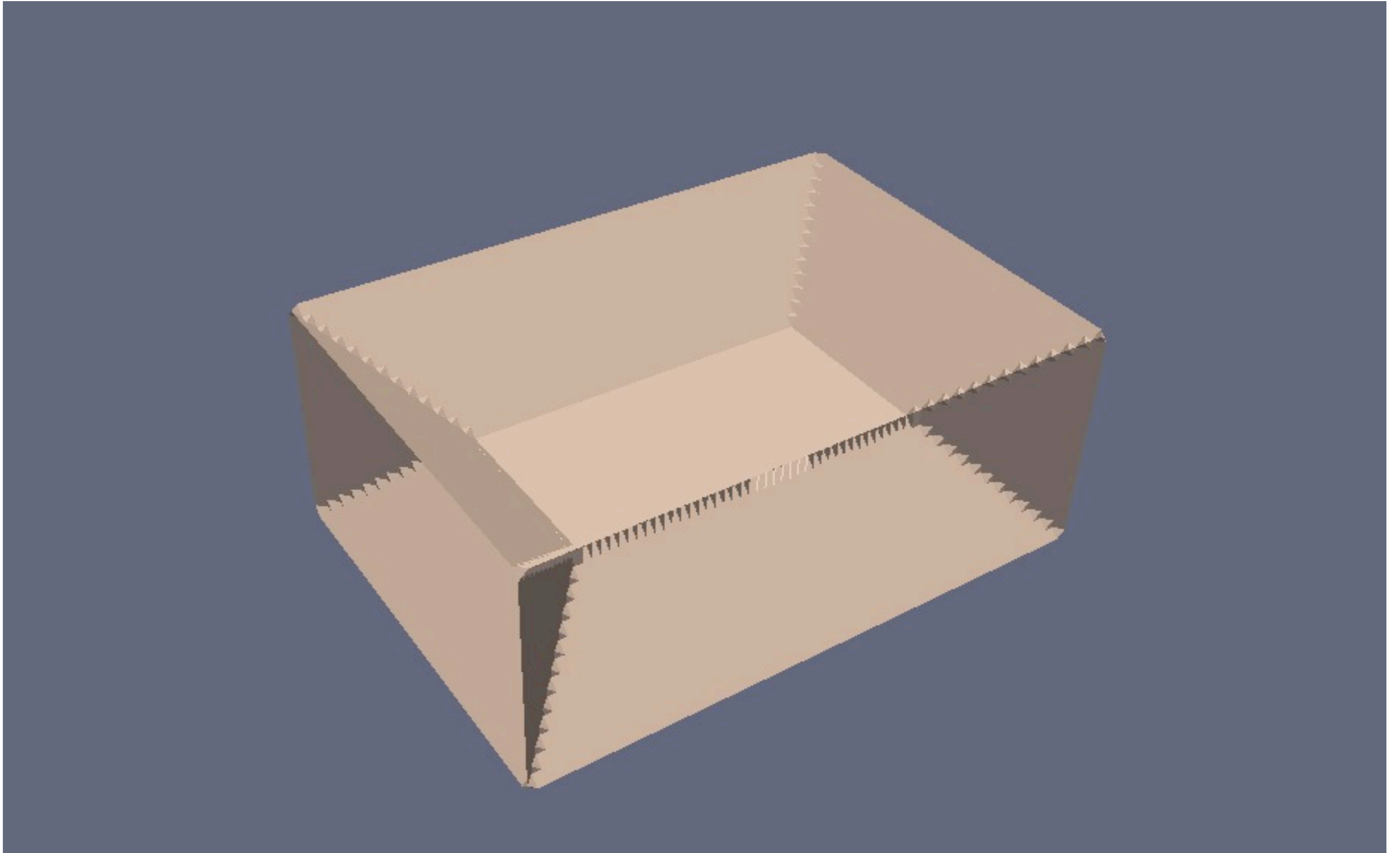
For each remaining point p in distance order

if `is_simple($S \setminus p$)` and `not is_endpoint(S, p)`

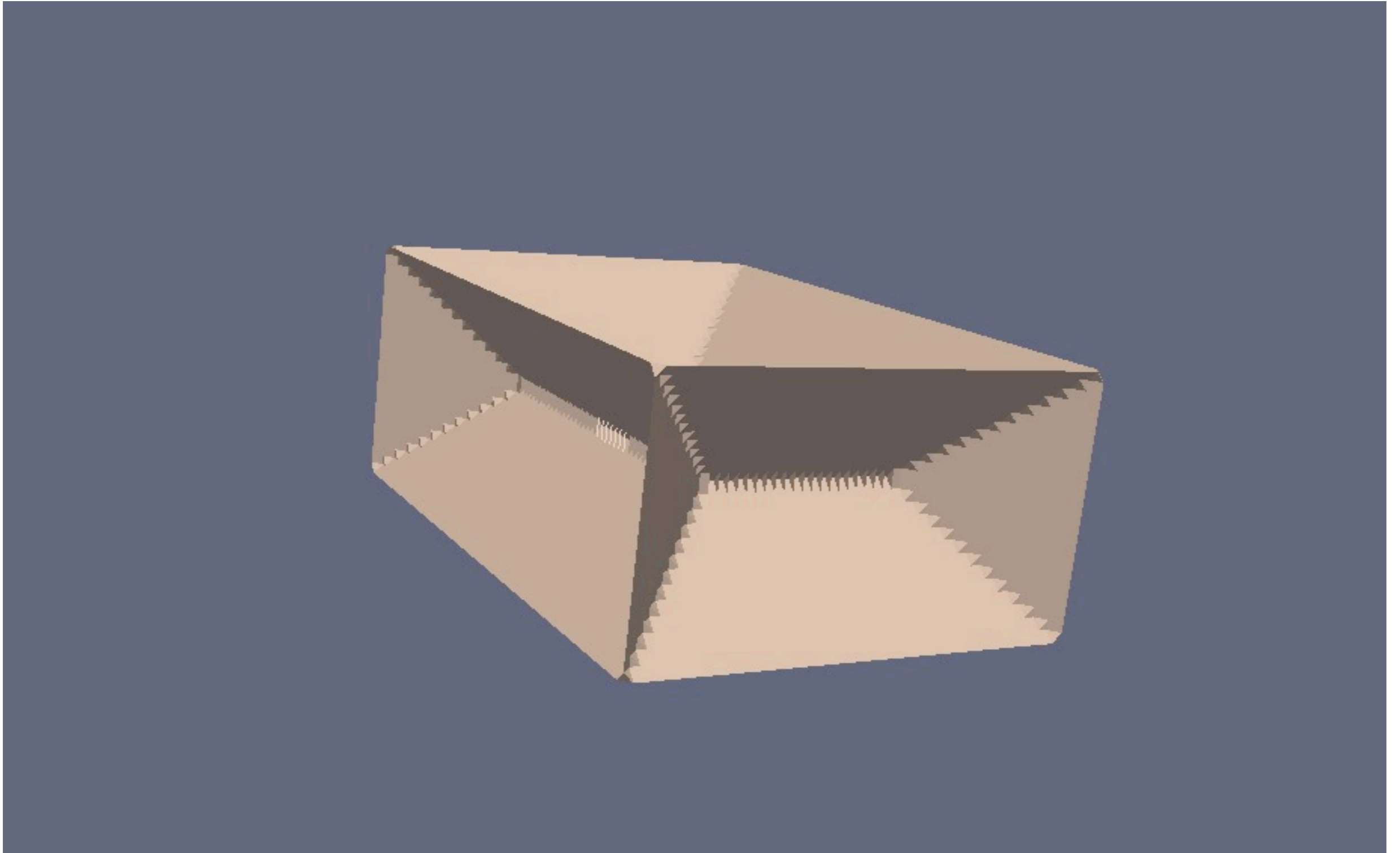
then $S = S \setminus p$



Original Shape



Extracted Skeleton



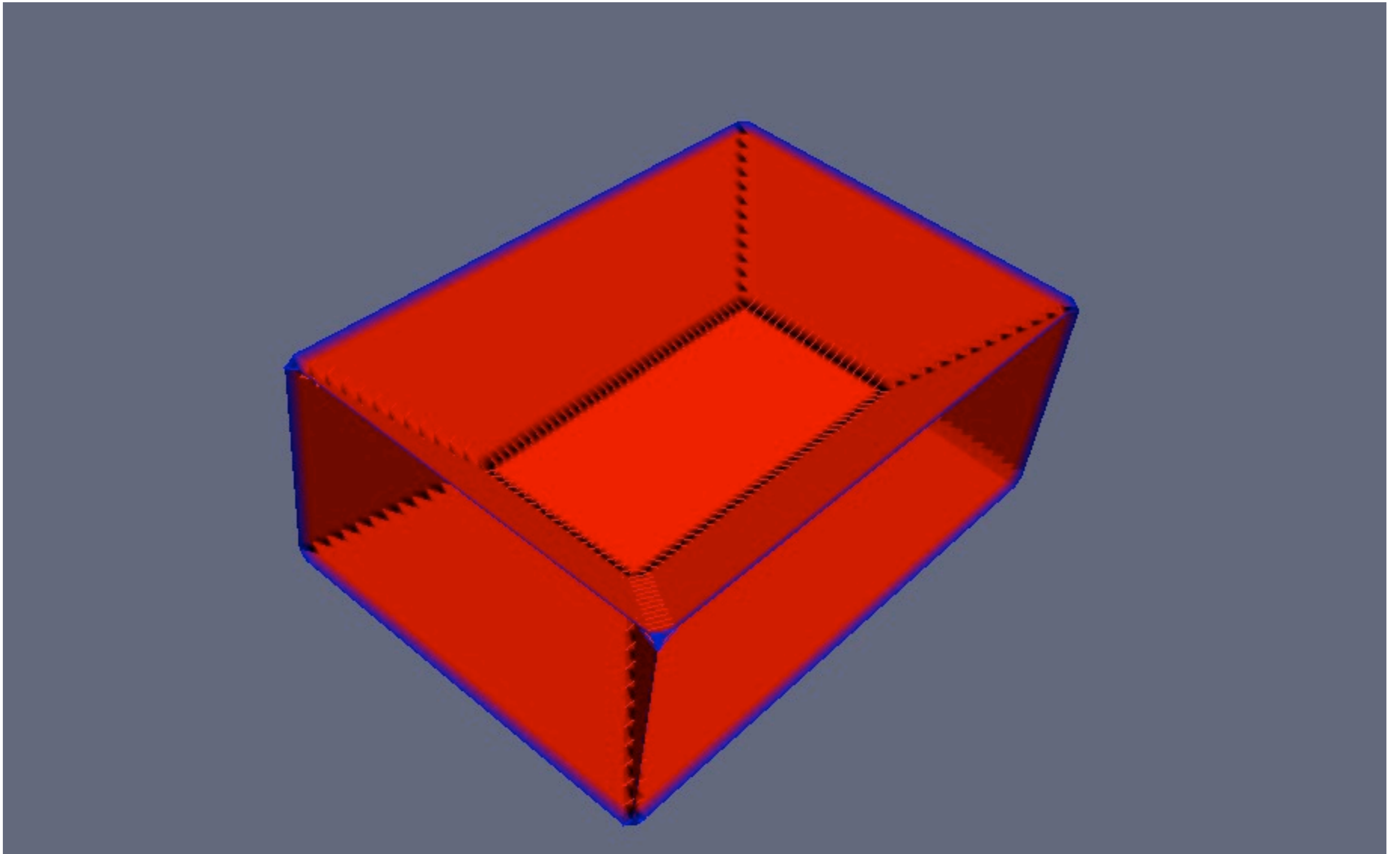
Extracted Skeleton (alternative view)

Labeling the Medial Surfaces

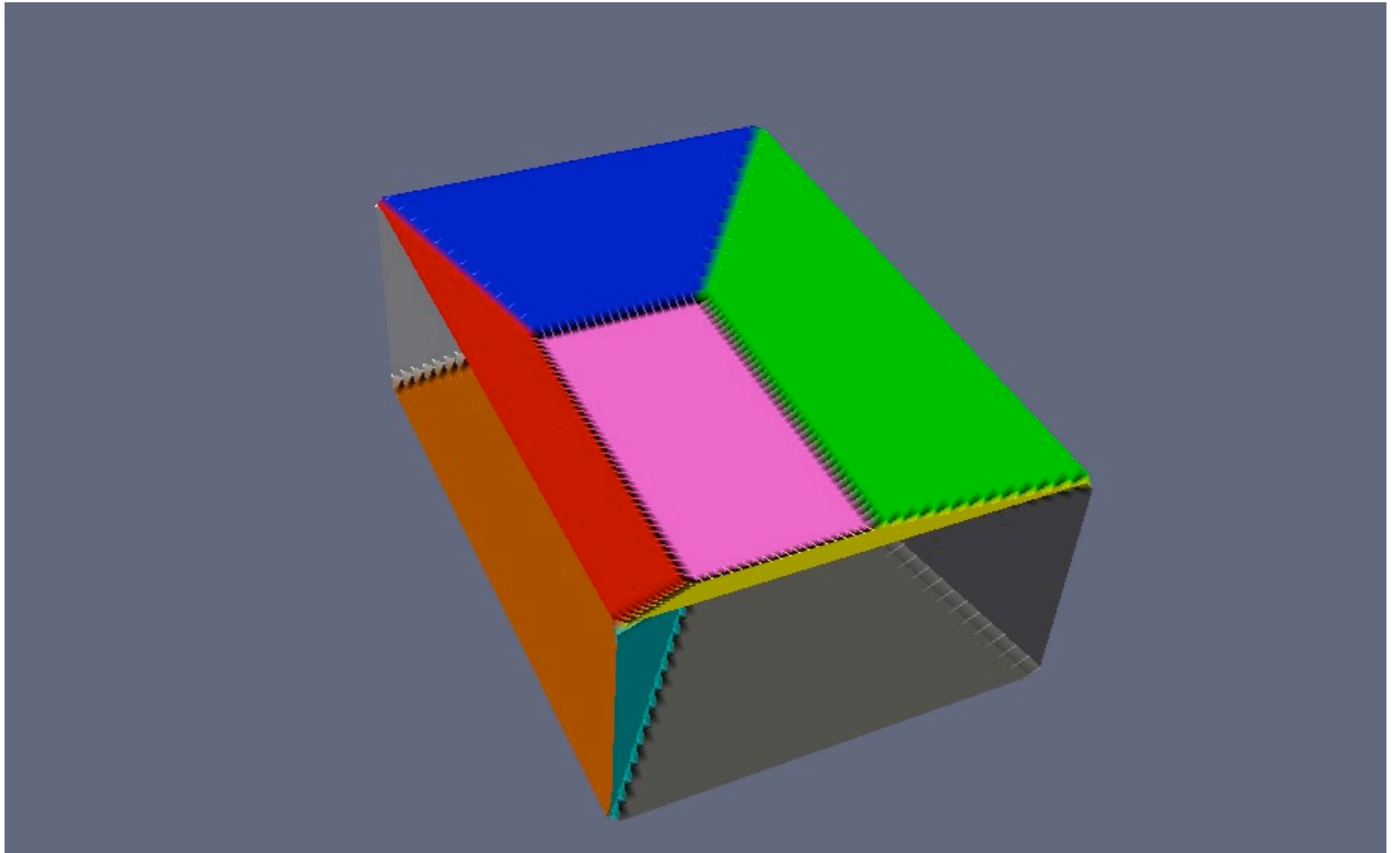
- With the 3D skeleton to hand, the next step is the segmentation of the surfaces composing the skeleton
- We'd like to take a region growing approach:
 - Start from a random seed and propagate the information (ID) to all the neighbors until you can't go on any further
 - When?

Labeling the Medial Surfaces

- Mark all the junctions using the topological classification of Malandain
- Identify the missed junction (curve-surface) with the following criterion
 - If p is adjacent to both a surface point and a curve point, then p is a junction point



Topological Classification



Segmented Skeleton

Improvements

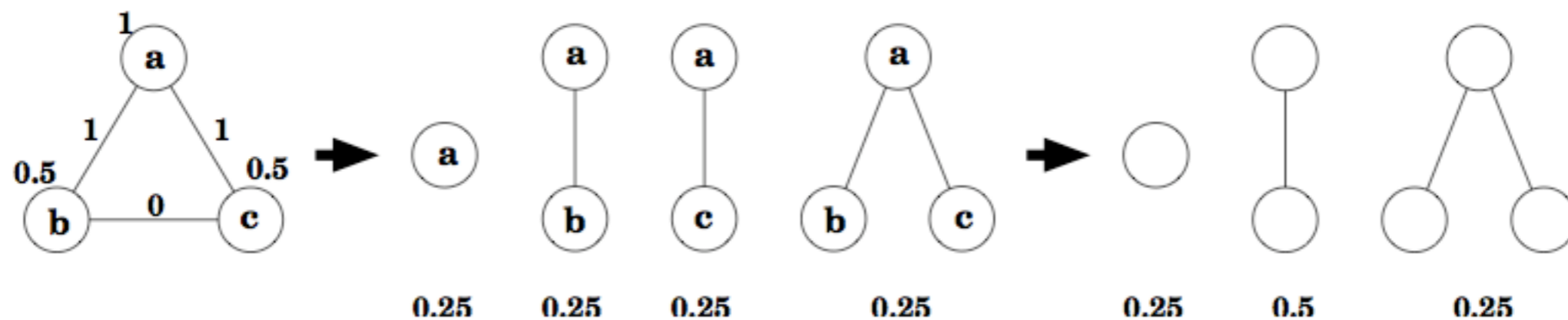
- Better dataset
- Better voxellization
- Subpixel precision
- Closure?

Extracting an Attributed Graph

- Once the skeleton is segmented, we can construct a graph such that each node corresponds to a separated surface (or curve) and two nodes share an edge if the corresponding surfaces (curves) are separated by a junction.
- Then we can assign each node a weight which is the sum of the value of the divergence at all points comprising the surface (curve). The idea is that the more important the surface, the more weight we'll put on the corresponding node

Learning the Graph Structure

- Now we can learn shape models
 - Take N attributed graphs obtained from the skeletonization of similar shapes (e.g. chairs)
 - Try to infer an archetype of the shape “chair”



Bibliography

H. Blum. A Transformation for Extracting New Descriptors of Shape. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, 1967.

J. Crank and P. Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type. In *In Proc, Cambridge Philos. Soc.*, vol. 43, pages 50–67, 1947.

G. Malandain, G. Bertrand, and N. Ayache. Topological segmentation of discrete surfaces. *International Journal of Computer Vision*, 10(2):183–197, 1993.

A. Torsello and D. L. Dowe. Learning a generative model for structural representations. In *AI '08: Proceedings of the 21st Australasian Joint Conference on Artificial Intelligence*, pages 573–583, Berlin, Heidelberg, 2008. Springer-Verlag.

A. Torsello and E. Hancock. Correcting curvature-density effects in the hamilton-jacobi skeleton. 15(4):877–891, April 2006.

A. Torsello and E. Hancock. A skeletal measure of 2d shape similarity. 95(1):1–29, July 2004.

A. Torsello and E. R. Hancock. Learning shape-classes using a mixture of tree-unions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(6):954–967, 2006.