Simplification and Repair of Polygonal Models Using Volumetric Techniques

Fakir S. Nooruddin and Greg Turk

presented by
Siyu Song
cs294-7 Spring 2008

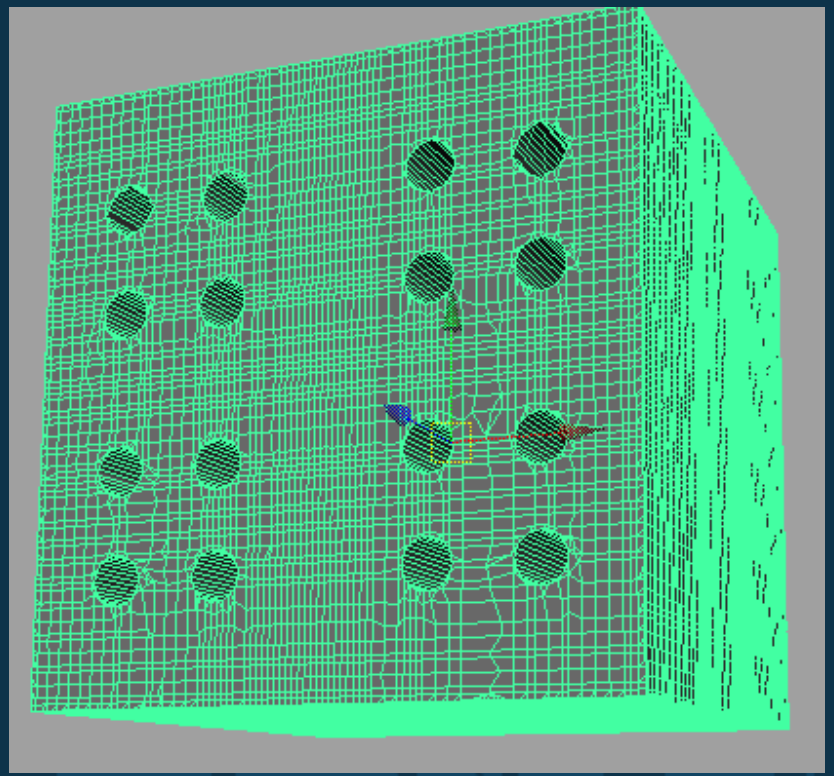
Volumetric Repair/Simplification

Main Goals:

- Repair
- Simplification

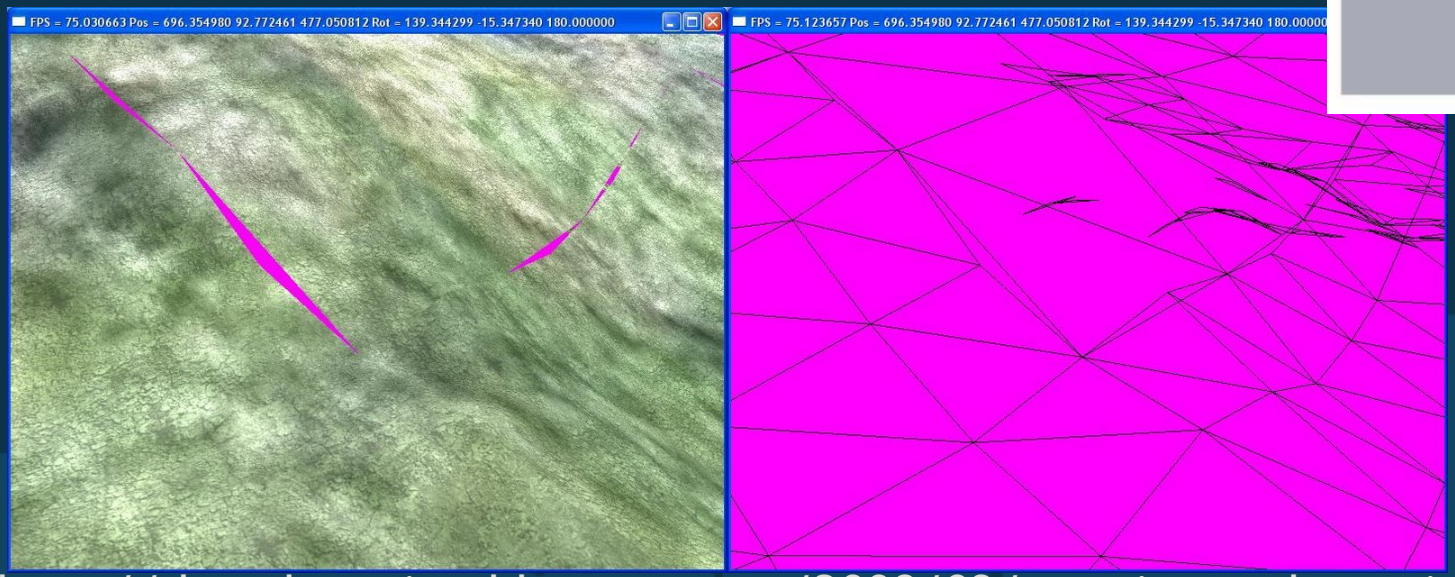
Benefits of Using Volume:

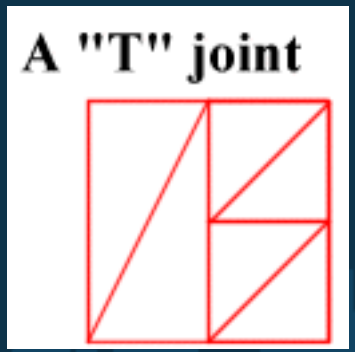
- can change topology
- gives manifold output



Common Mesh Degeneracies

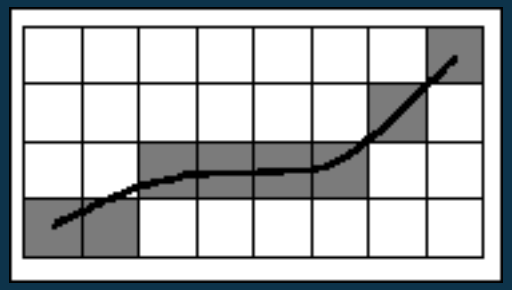
- T-Joints
- Cracks/Holes
- Interpenetrating Surfaces
- Non-manifold edges and vertices





http://decadeengine.blogspot.com/2008/03/terrain-cracks-revisited.html
http://escience.anu.edu.au/lecture/cg/Texture/printNotes.en.html

Some Terms



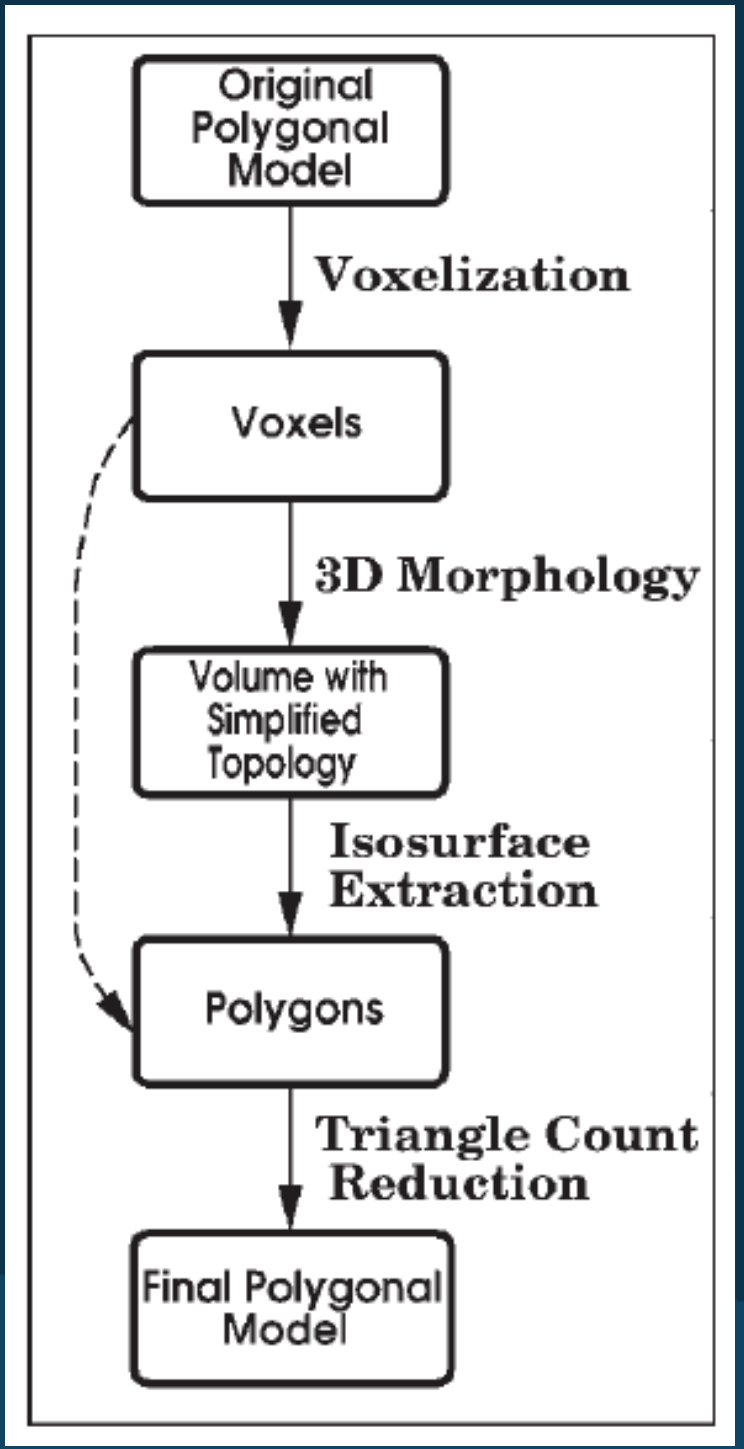
Voxelization - convert continuous geometric representation
to an approximation of a set of voxels
2D voxel = square 3D voxel = cube

Morphology - manipulating shapes in image processing

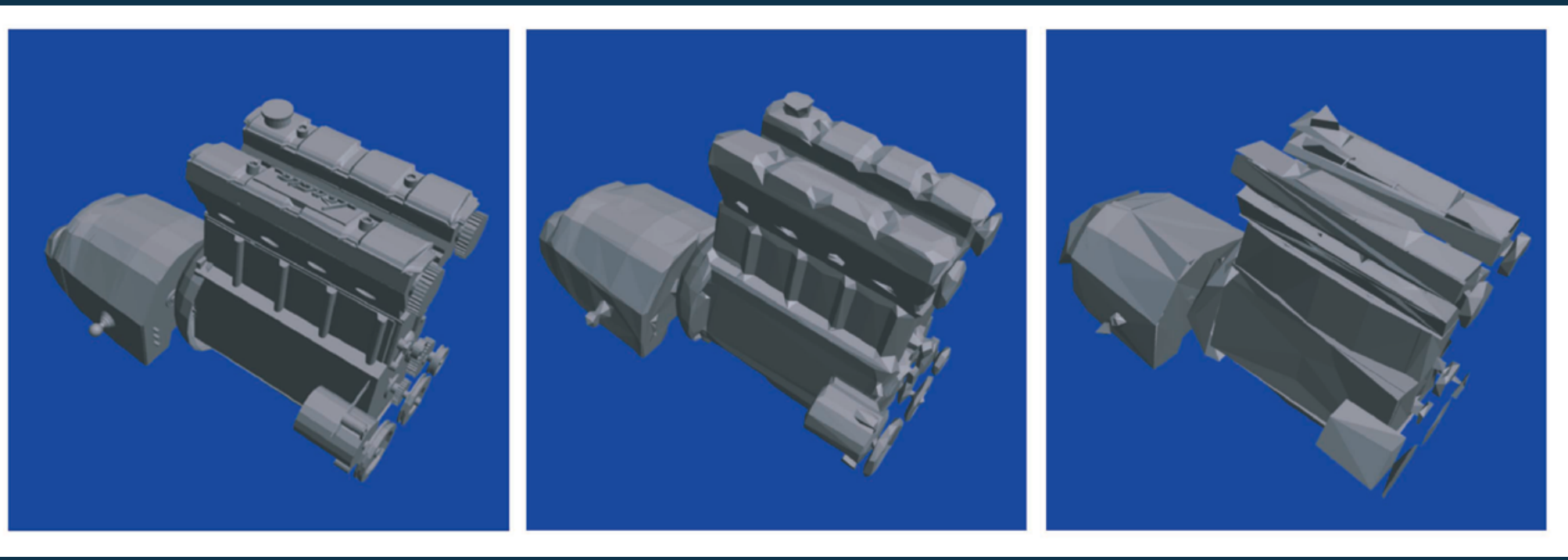
Garland and Heckbert (Qslim) - error quadrics

http://www.cs.sunysb.edu/~vislab/projects/volume/Papers/Voxel/index.html

The Pipeline



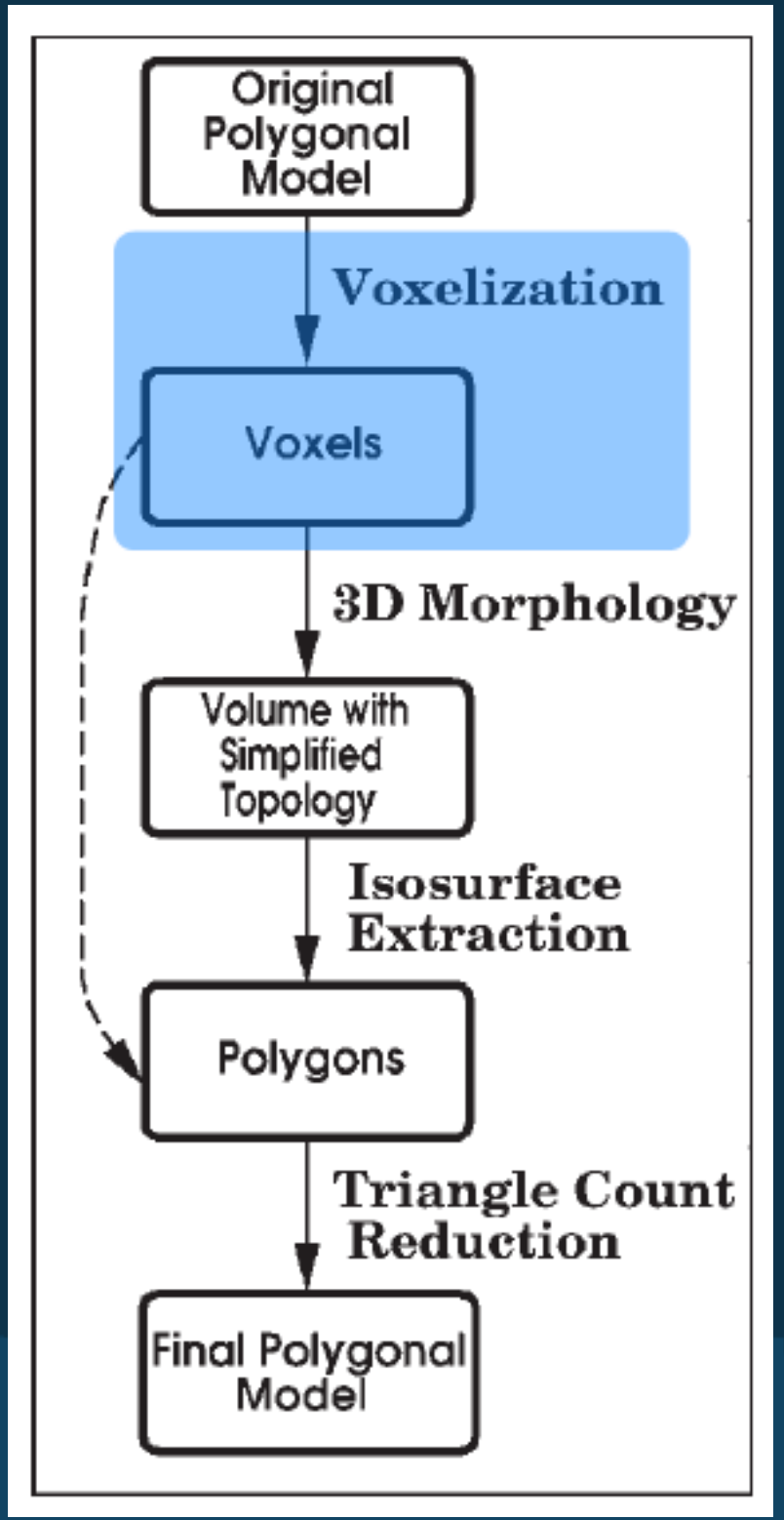
Benefits



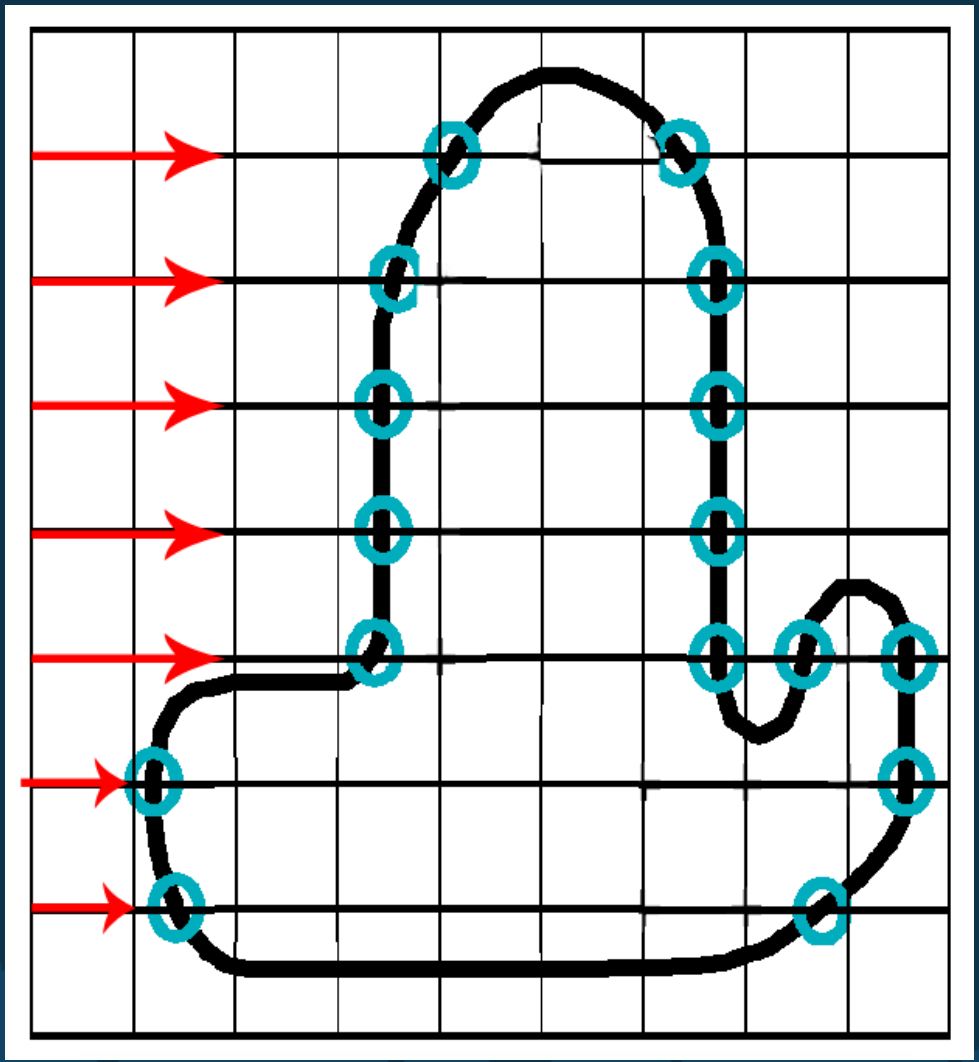
Voxelization Methods

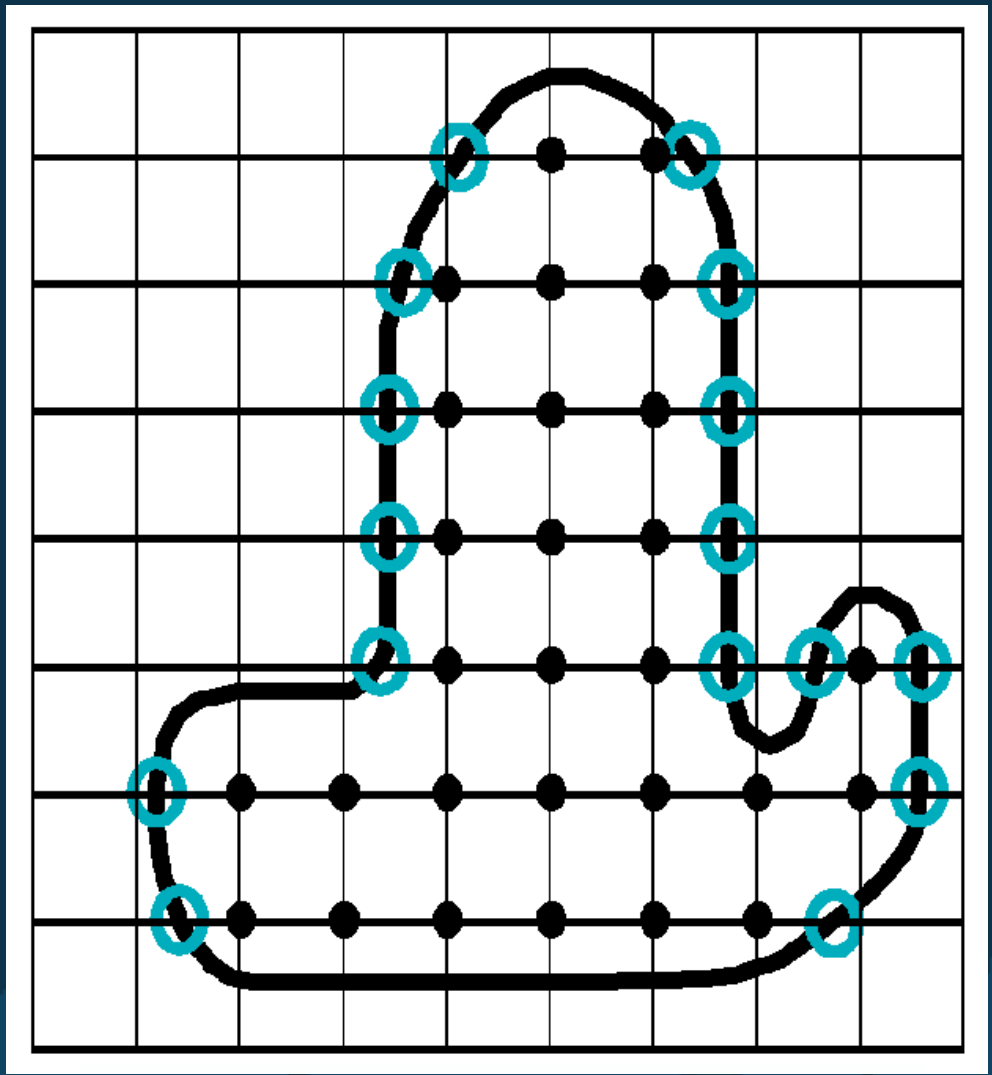
Parity counting

Ray stabbing



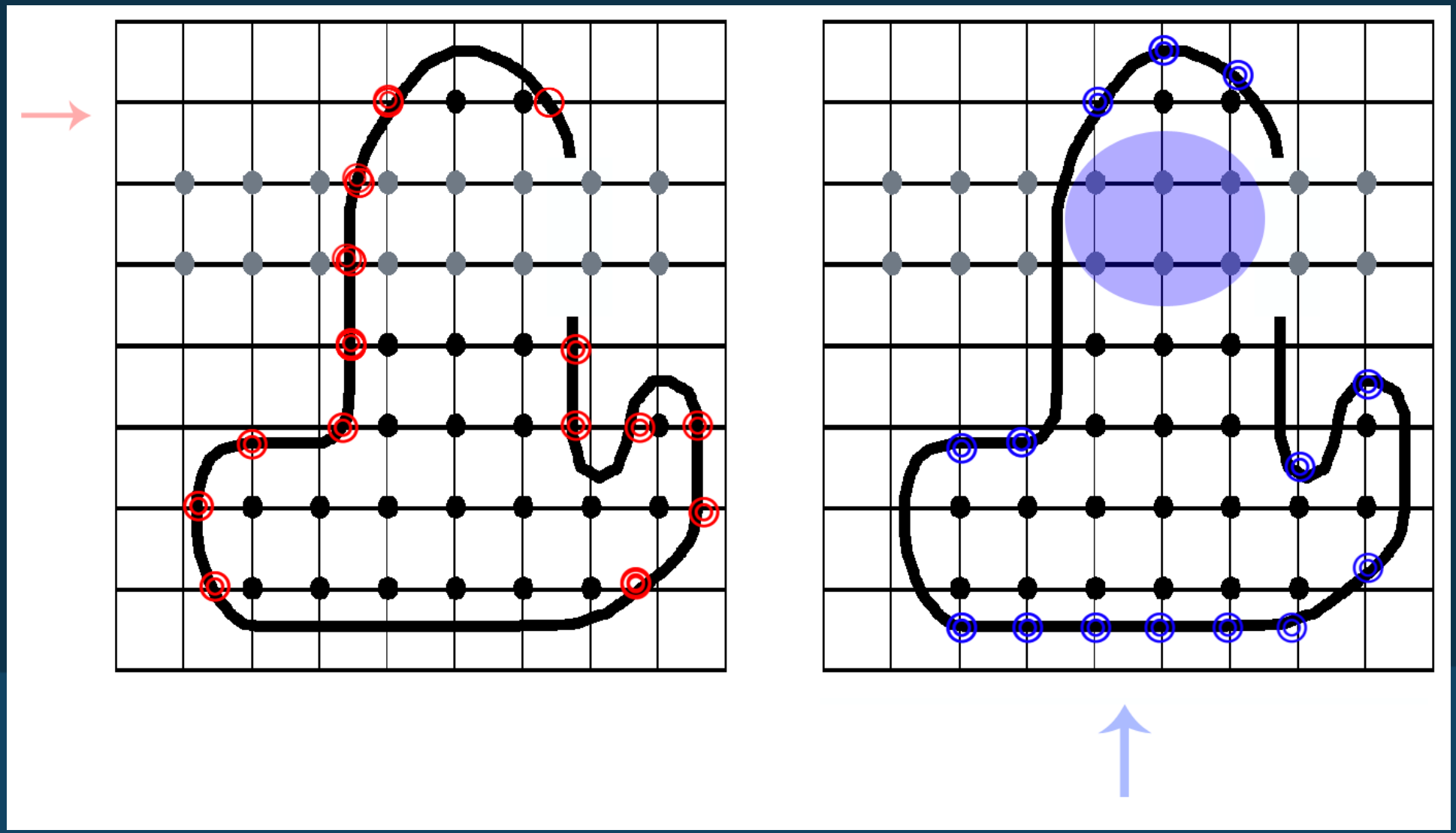
2D Parity Count On Watertight Shape





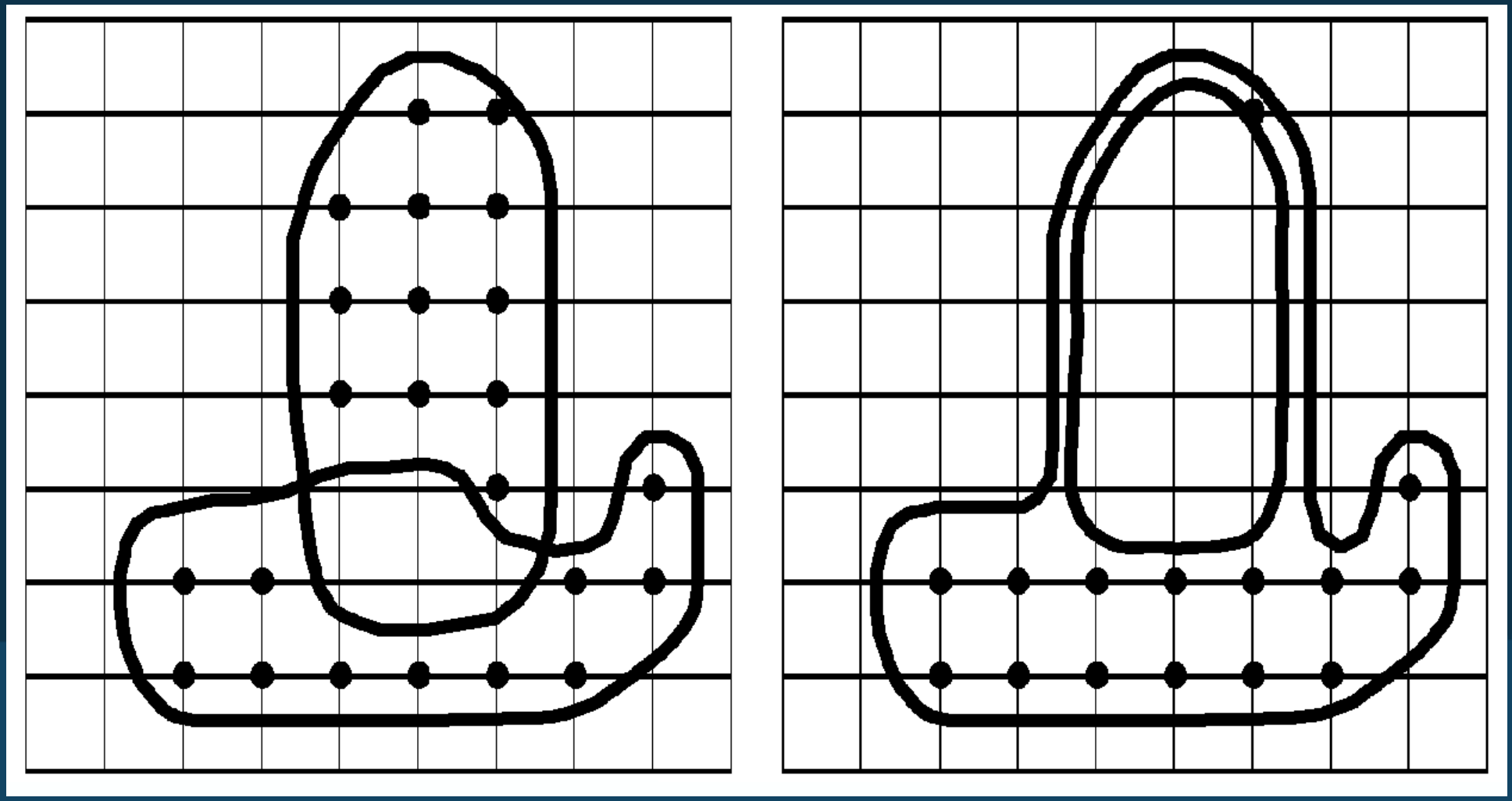
Dealing with Cracks/Holes

Intersections in multiple directions

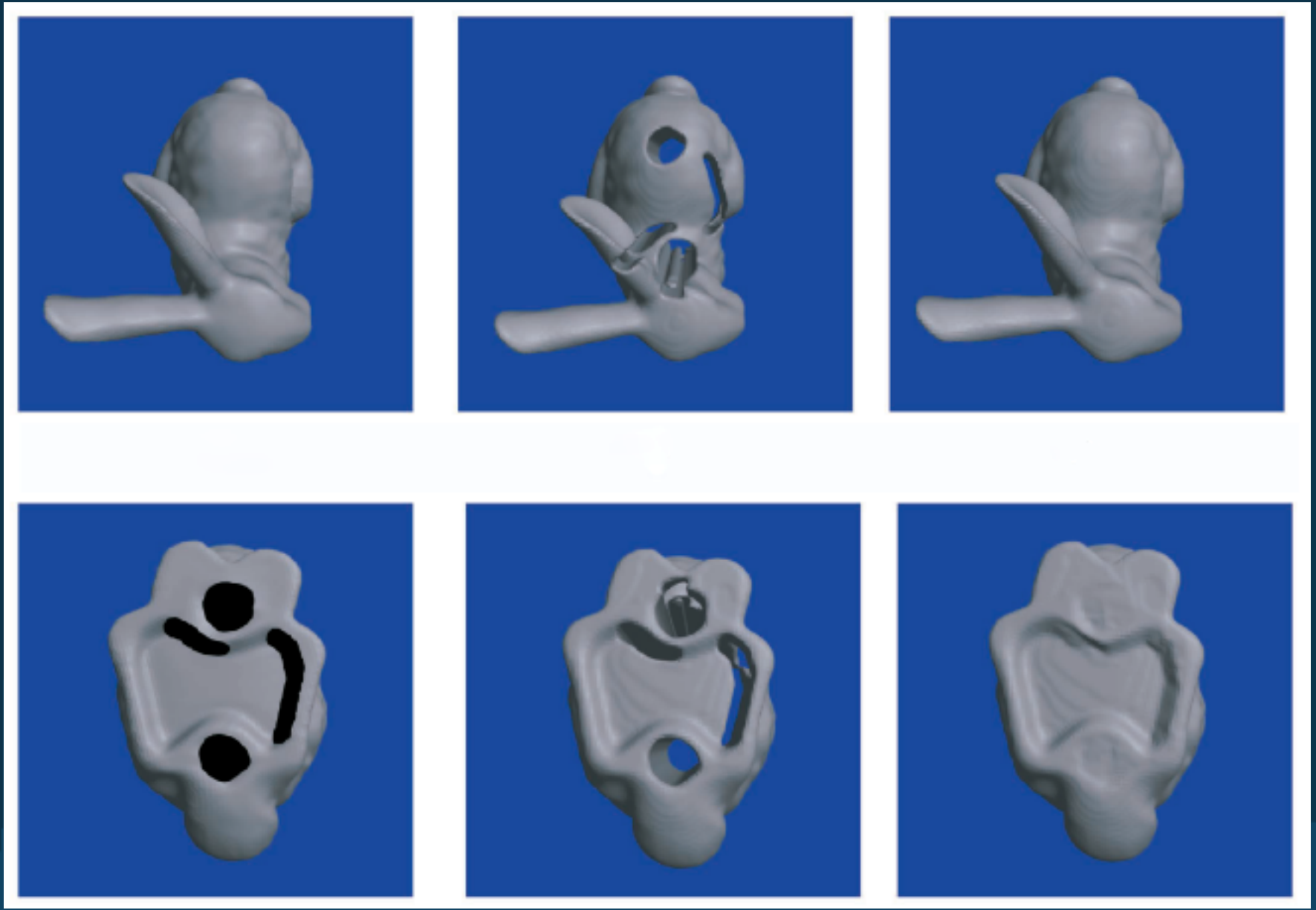


Limitations of Parity Counting

- Unable to handle overlaps and thin walls

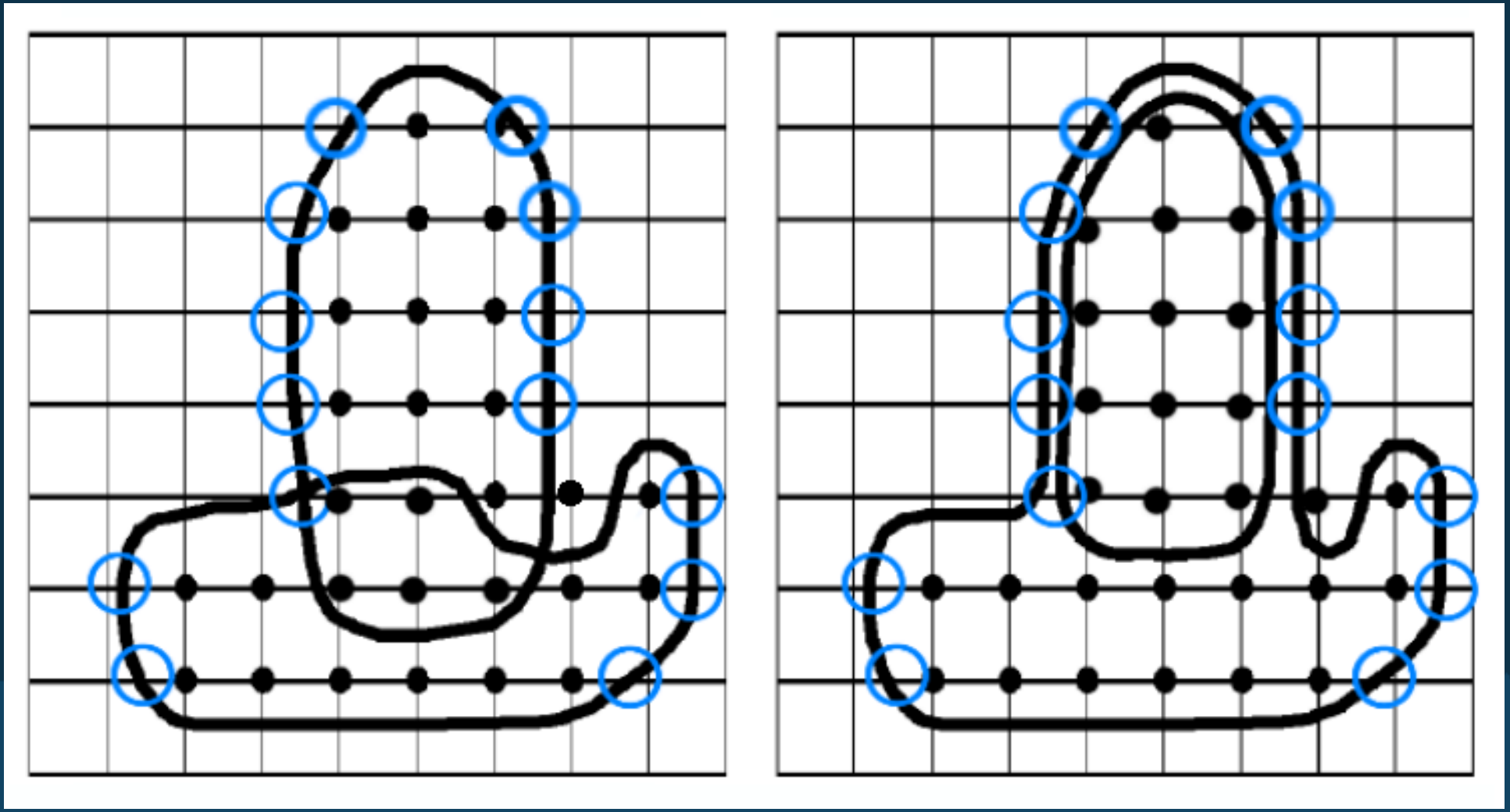


Parity Counting - 3D results



Ray Stabbing

- Look only at the first and last intersection
- Multiple projections, voxel exterior if any projection say so



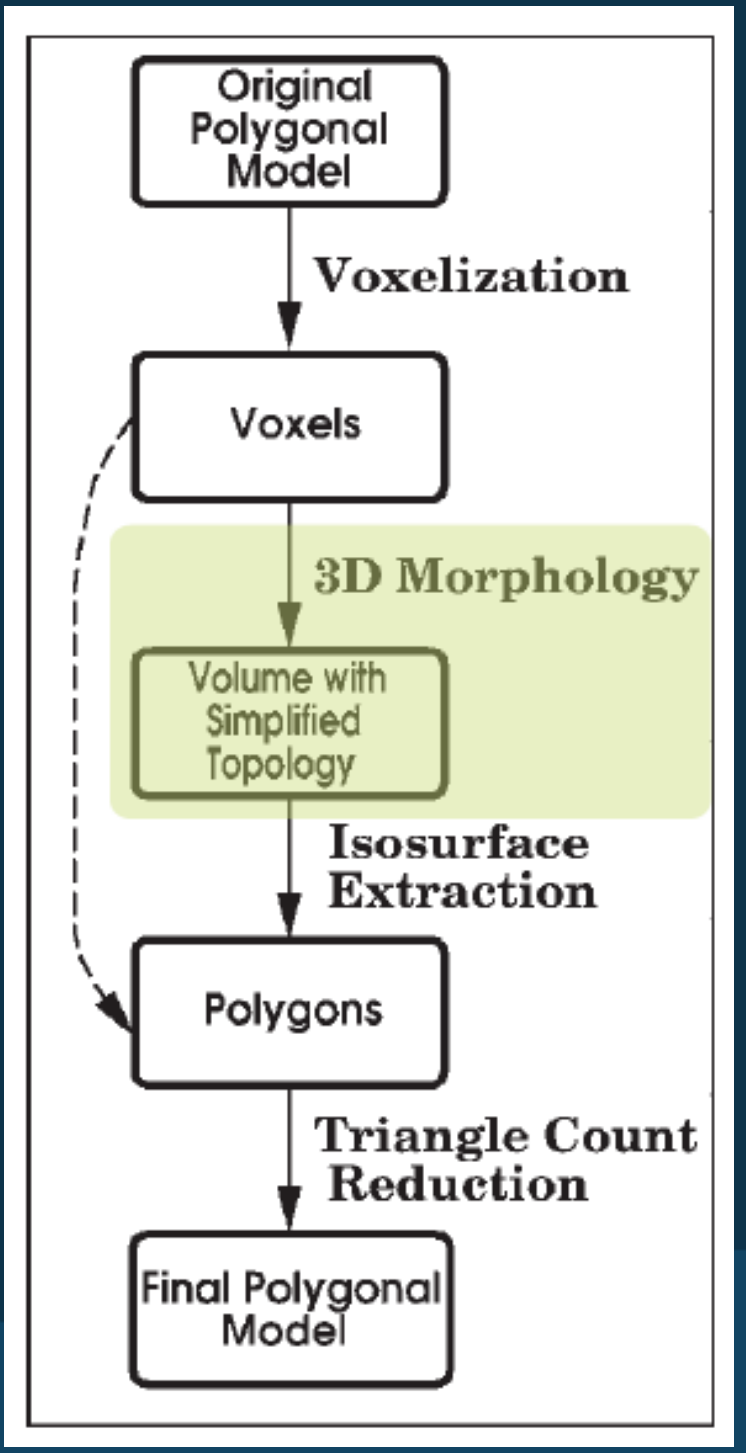
Parity Count vs Ray Stabbing

- Voxelization method based on types of degeneracy in mesh

Types of Degeneracies		
	Parity Count	Ray Stabbing
Fixes T-Joints	Y	Y
Fixes Cracks / Holes	Y	N
Retains Interior Detail	Y	N
Merges Interpenetrating Surfaces	N	Y
Fixes Non-manifold edges and vertices	Y	Y

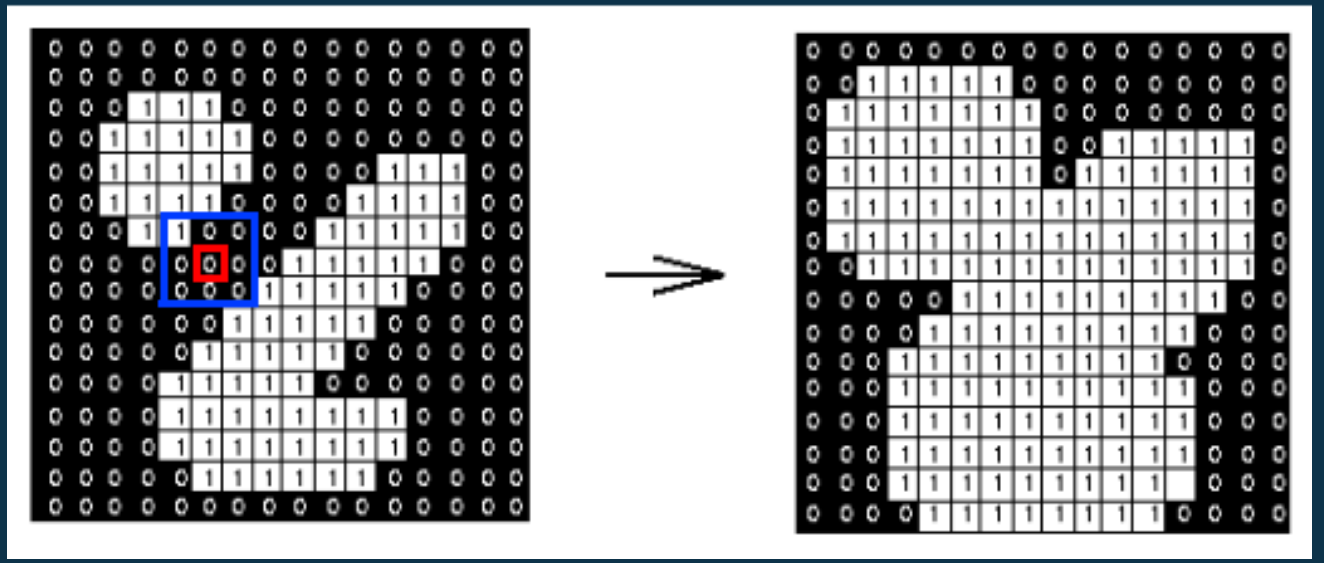
Volume Manipulation

- perform topological changes in volume representation
- manifold output
- only necessary for simplification

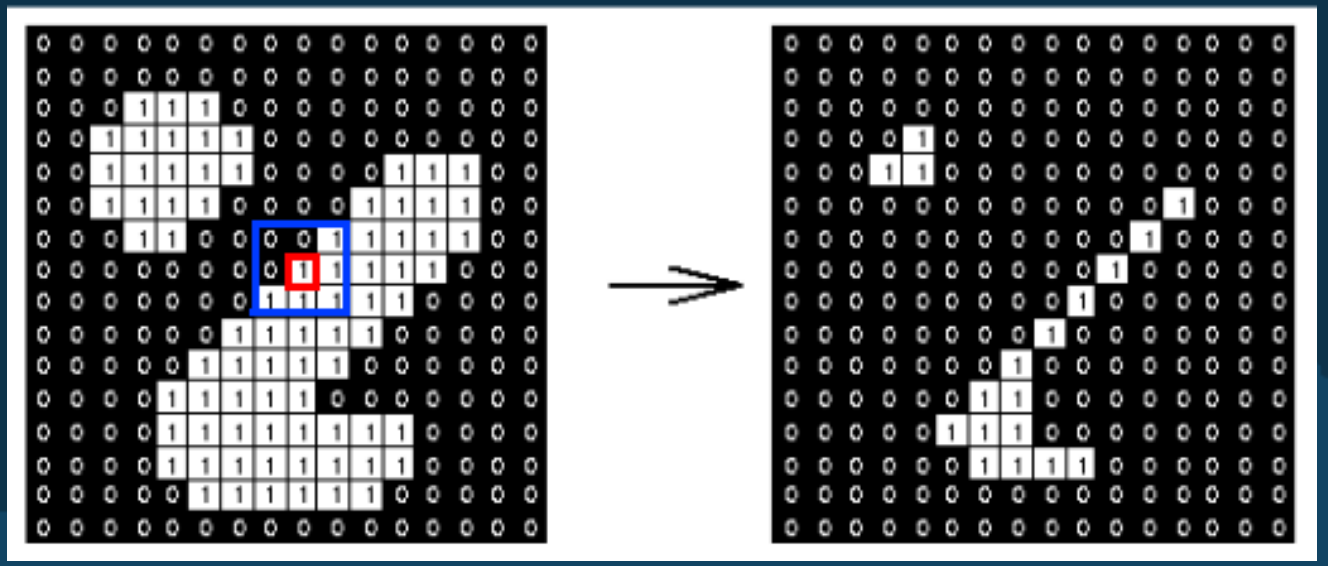


2D Morphology

Dilation

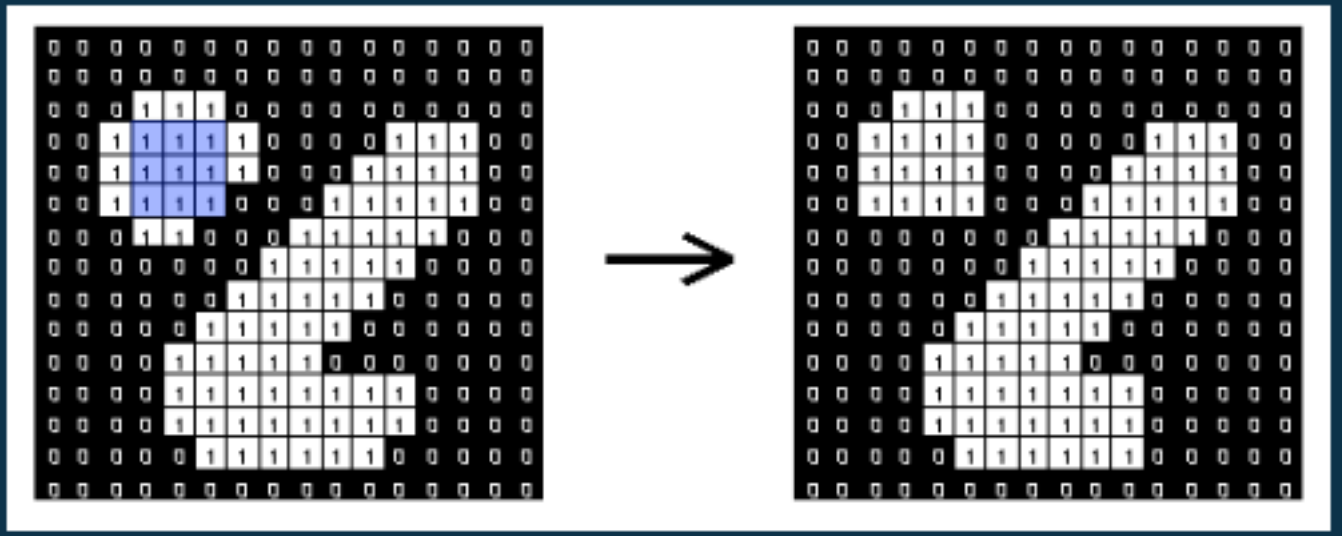


Erosion

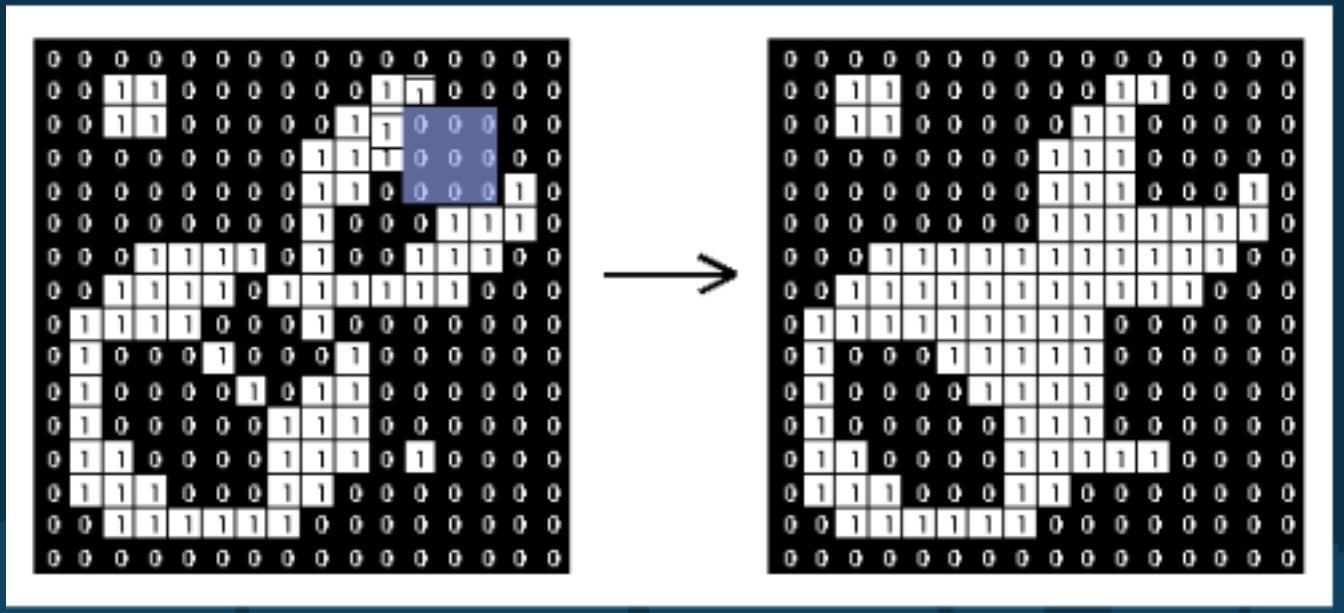


Openings/Closings

Opening:
Erosion + Dilation

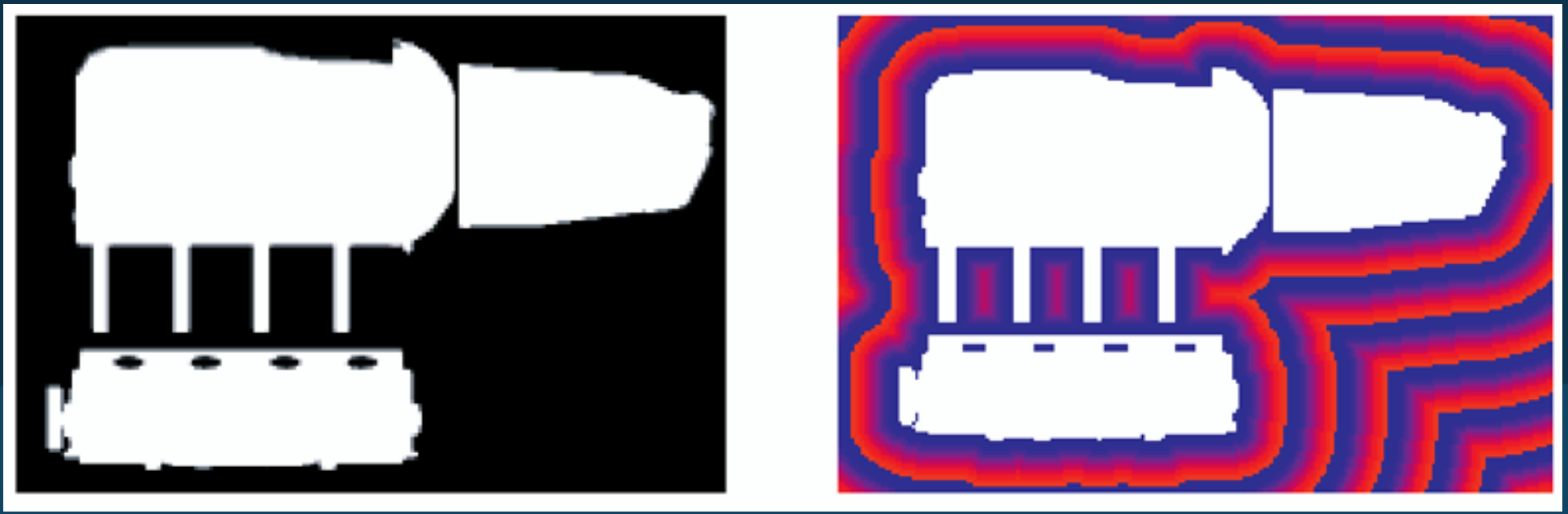


Closing:
Dilation + Erosion

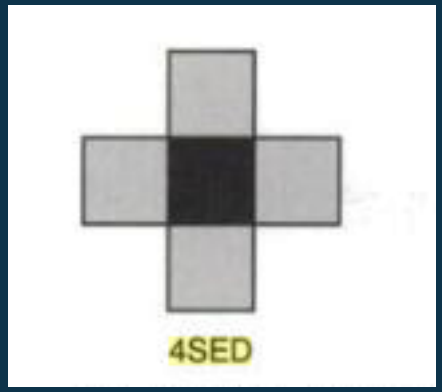


4SED (Four-point Sequential Euclidean distance mapping)

- Pixels store a 2d int vector
- Initialize feature pixels to 0,0
- Otherwise initialized to MAXVAL
- magnitude of vectors
  = distance to nearest feature pixel



4SED



- look at neighbor pixel
- if distance of neighbor < current
  current pixel = distance of neighbor

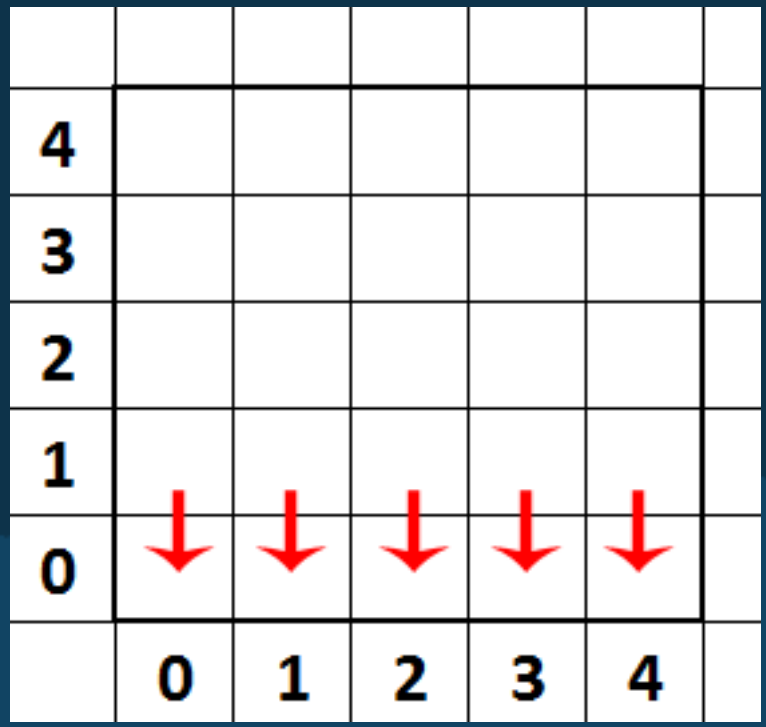
First loop of Danielsson's algorithm (sweeping from bottom-to-top)

for j = 1 to dy -1

Examine pixels below the current row
for i = 0 to dx - 1
if $\overrightarrow{mag(D(i,j))} \leq \overrightarrow{mag(D(i,j-1)} + \overrightarrow{<0,1>})$
$\overrightarrow{D(i,j)} = \overrightarrow{D(i,j-1)} + \overrightarrow{<0,1>}$

Examine pixels to the left of each pixel in a row
for i = 0 to dx - 1
if $\overrightarrow{mag(D(i,j))} \leq \overrightarrow{mag(D(i-1,j)} + \overrightarrow{<1,0>})$
$\overrightarrow{D(i,j)} = \overrightarrow{D(i-1,j)} + \overrightarrow{<1,0>}$

Examine pixels to the right of each pixel in a row
for i = dx - 2 downto 0
if $\overrightarrow{mag(D(i,j))} \leq \overrightarrow{mag(D(i+1,j)} + \overrightarrow{<1,0>})$
$\overrightarrow{D(i,j)} = \overrightarrow{D(i+1,j)} + \overrightarrow{<1,0>}$



4SED

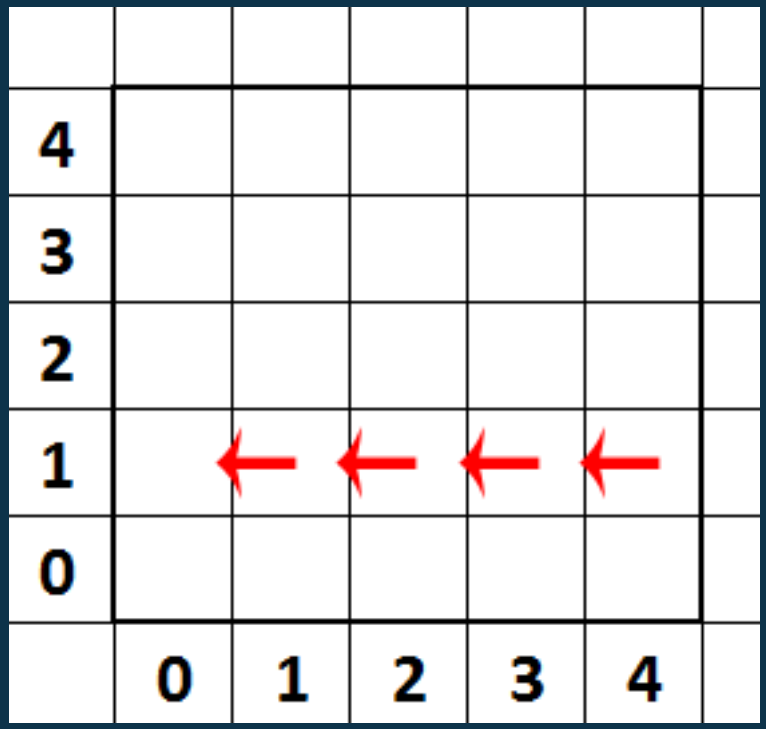
First loop of Danielsson's algorithm (sweeping from bottom-to-top)

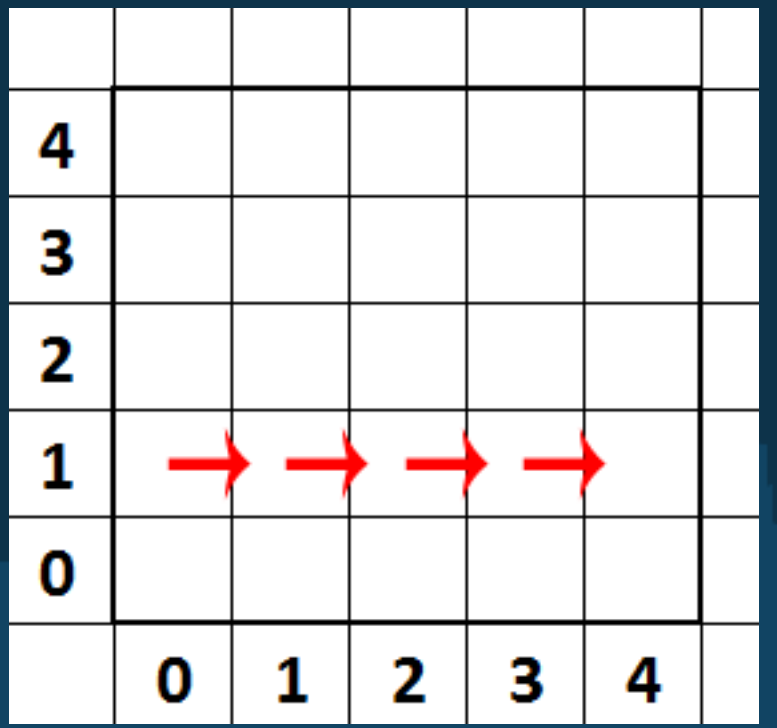
for j = 1 to dy -1

Examine pixels below the current row
for i = 0 to dx - 1
if $\text{mag}(\overrightarrow{D(i,j)}) \leq mag(\overrightarrow{D(i,j-1)} + \overrightarrow{<0,1>})$
$\overrightarrow{D(i,j)} = \overrightarrow{D(i,j-1)} + \overrightarrow{<0,1>}$

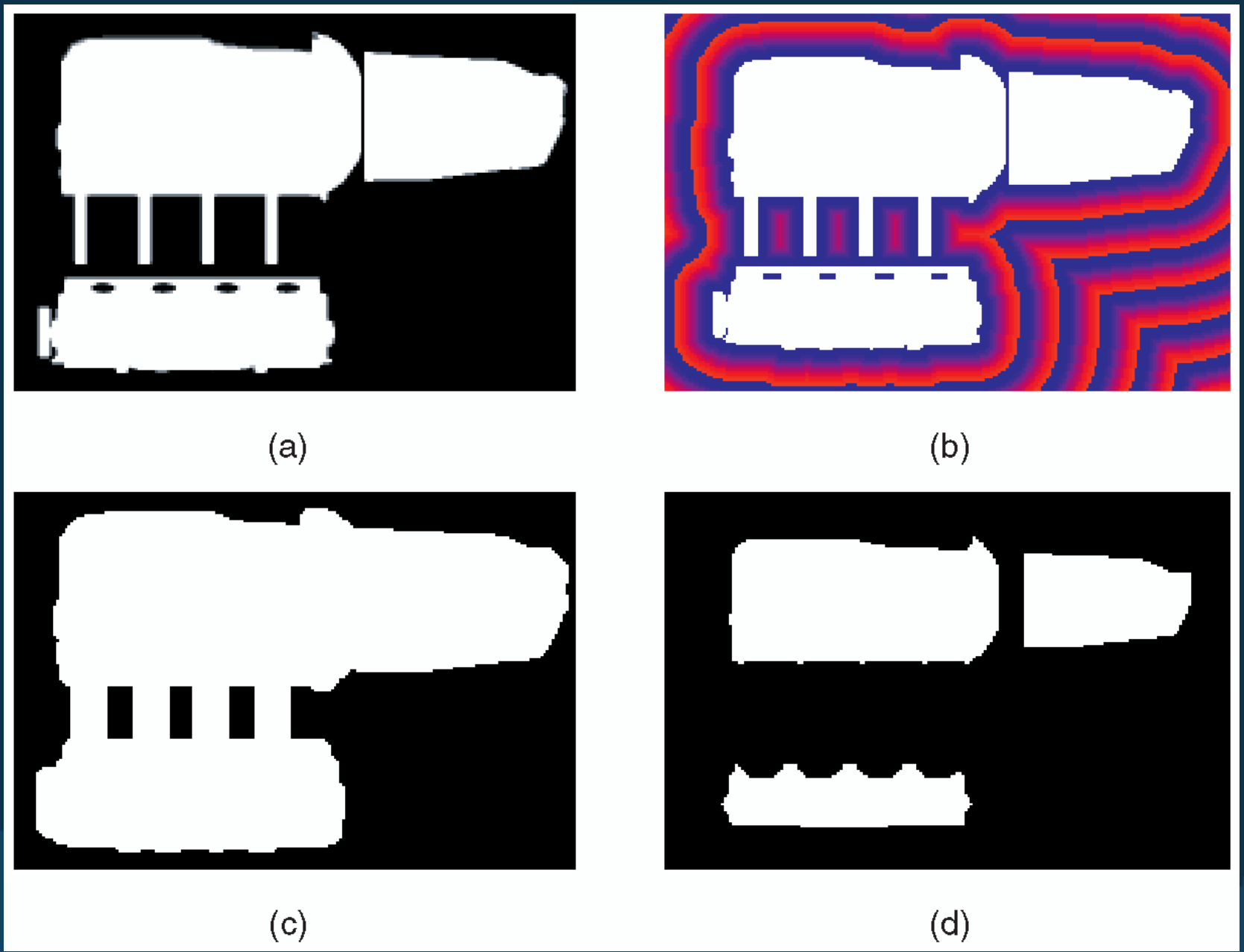
Examine pixels to the left of each pixel in a row
for i = 0 to dx - 1
if $\text{mag}(\overrightarrow{D(i,j)}) \leq mag(\overrightarrow{D(i-1,j)} + \overrightarrow{<1,0>})$
$\overrightarrow{D(i,j)} = \overrightarrow{D(i-1,j)} + \overrightarrow{<1,0>}$

Examine pixels to the right of each pixel in a row
for i = dx - 2 downto 0
if $\text{mag}(\overrightarrow{D(i,j)}) \leq mag(\overrightarrow{D(i+1,j)} + \overrightarrow{<1,0>})$
$\overrightarrow{D(i,j)} = \overrightarrow{D(i+1,j)} + \overrightarrow{<1,0>}$





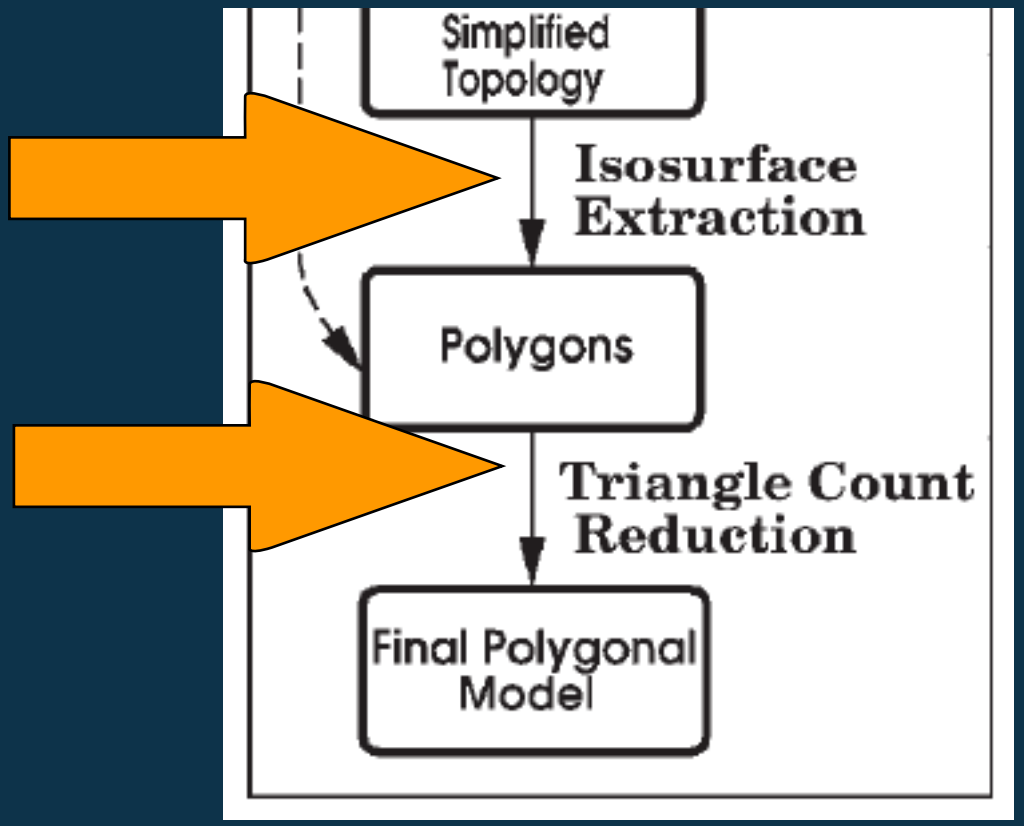
Distance map, Dilation, Erosion



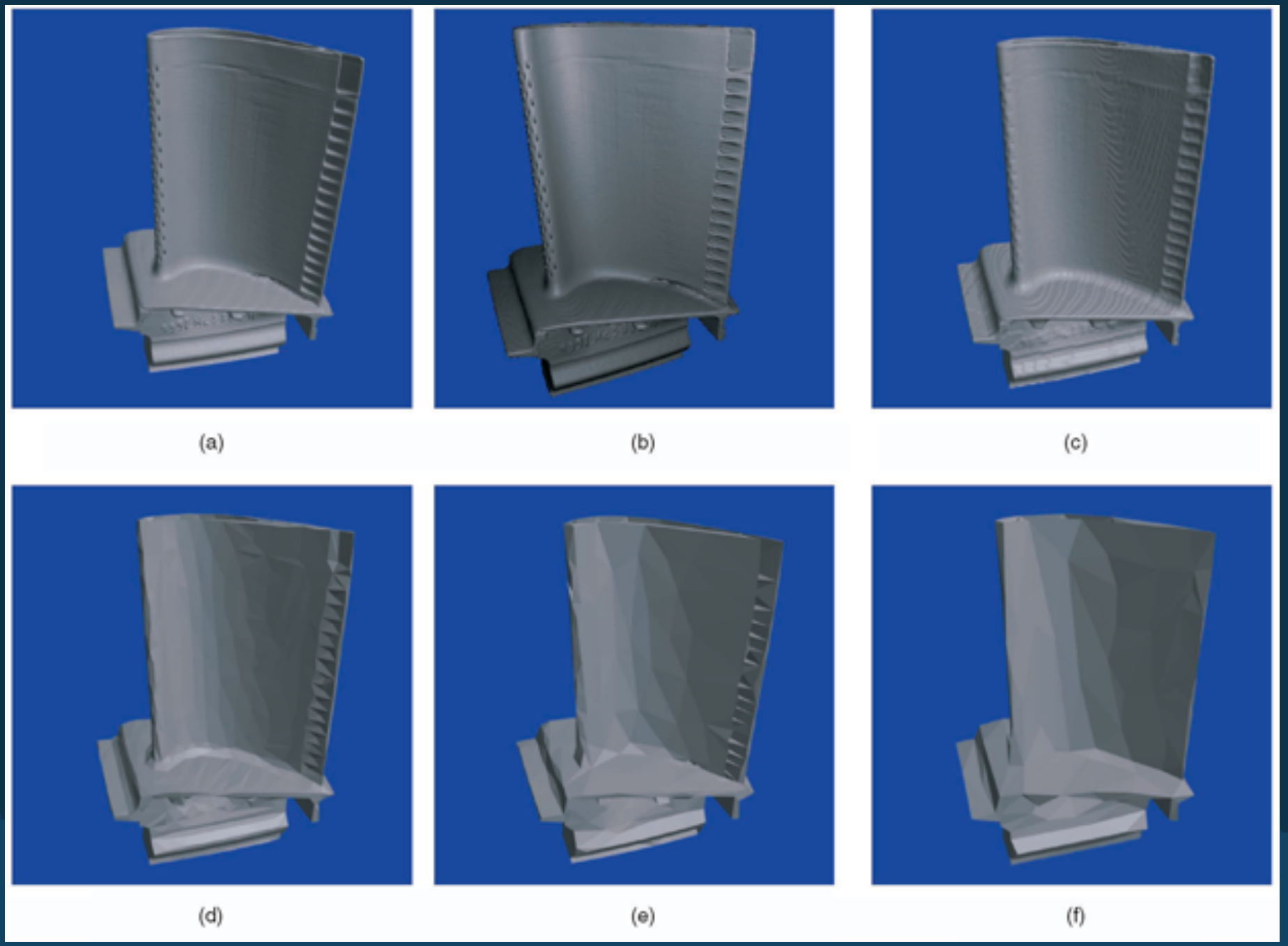
Volume Space to Polygon Model

Marching Cubes

Qslim
(Garland and Heckbert)



Results



Results

Conclusions

Results

- pipeline allows changing of topology
- better visual results than Qslim alone

Future work:

- deal with thin walls
- local modifications
- extend more 2D image processing techniques