Building simplification using offset curves obtained from the straight skeleton

Martijn Meijers

November 23, 2016

STW UC, TU Delft, BK–00.WEST.670

Building simplification using offset curves obtained from the straight skeleton

- Straight skeleton
- Offset curves
- Building simplification
- Results
- Discussion and Future work

Straight skeleton

- Polygon as input
- Shrink polygon inwards, with edges moving at constant speed
- Vertices move as well (some faster than others, depend on angle of the vertex)
- If moving vertex collides with non-adjacent edge: polygon is split and shrinking continues for each piece
- Moving vertices trace out set of curves: this is the straight skeleton

Straight skeleton

- Polygon as input
- Shrink polygon inwards, with edges moving at constant speed
- Vertices move as well (some faster than others, depend on angle of the vertex)
- If moving vertex collides with non-adjacent edge: polygon is split and shrinking continues for each piece
- Moving vertices trace out set of curves: this is the straight skeleton

Straight skeleton

- Polygon as input
- Shrink polygon inwards, with edges moving at constant speed
- Vertices move as well (some faster than others, depend on angle of the vertex)
- If moving vertex collides with non-adjacent edge: polygon is split and shrinking continues for each piece
- Moving vertices trace out set of curves: this is the straight skeleton

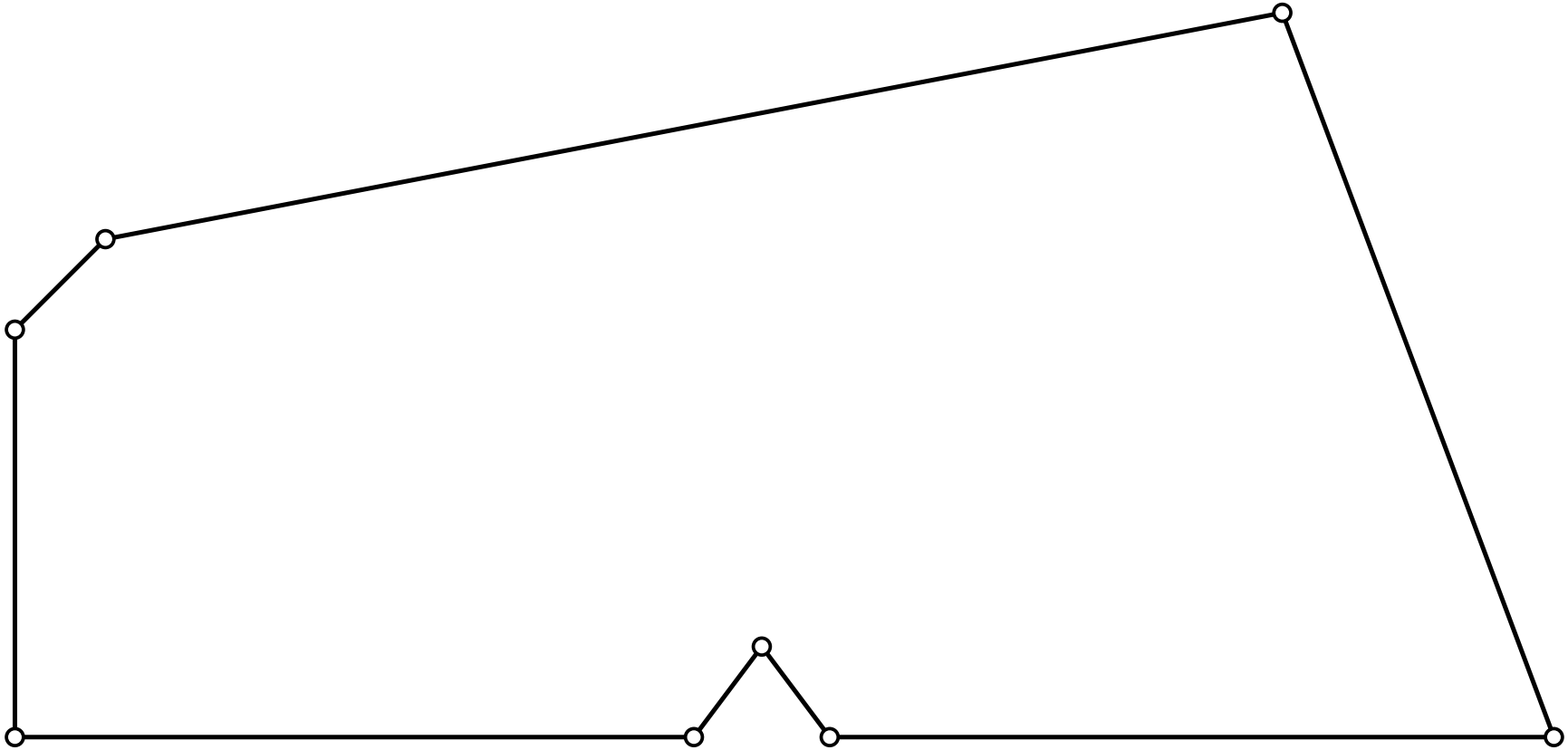
Straight skeleton

- Polygon as input
- Shrink polygon inwards, with edges moving at constant speed
- Vertices move as well (some faster than others, depend on angle of the vertex)
- If moving vertex collides with non-adjacent edge: polygon is split and shrinking continues for each piece
- Moving vertices trace out set of curves: this is the straight skeleton

Straight skeleton

- Polygon as input
- Shrink polygon inwards, with edges moving at constant speed
- Vertices move as well (some faster than others, depend on angle of the vertex)
- If moving vertex collides with non-adjacent edge: polygon is split and shrinking continues for each piece
- Moving vertices trace out set of curves: this is the straight skeleton

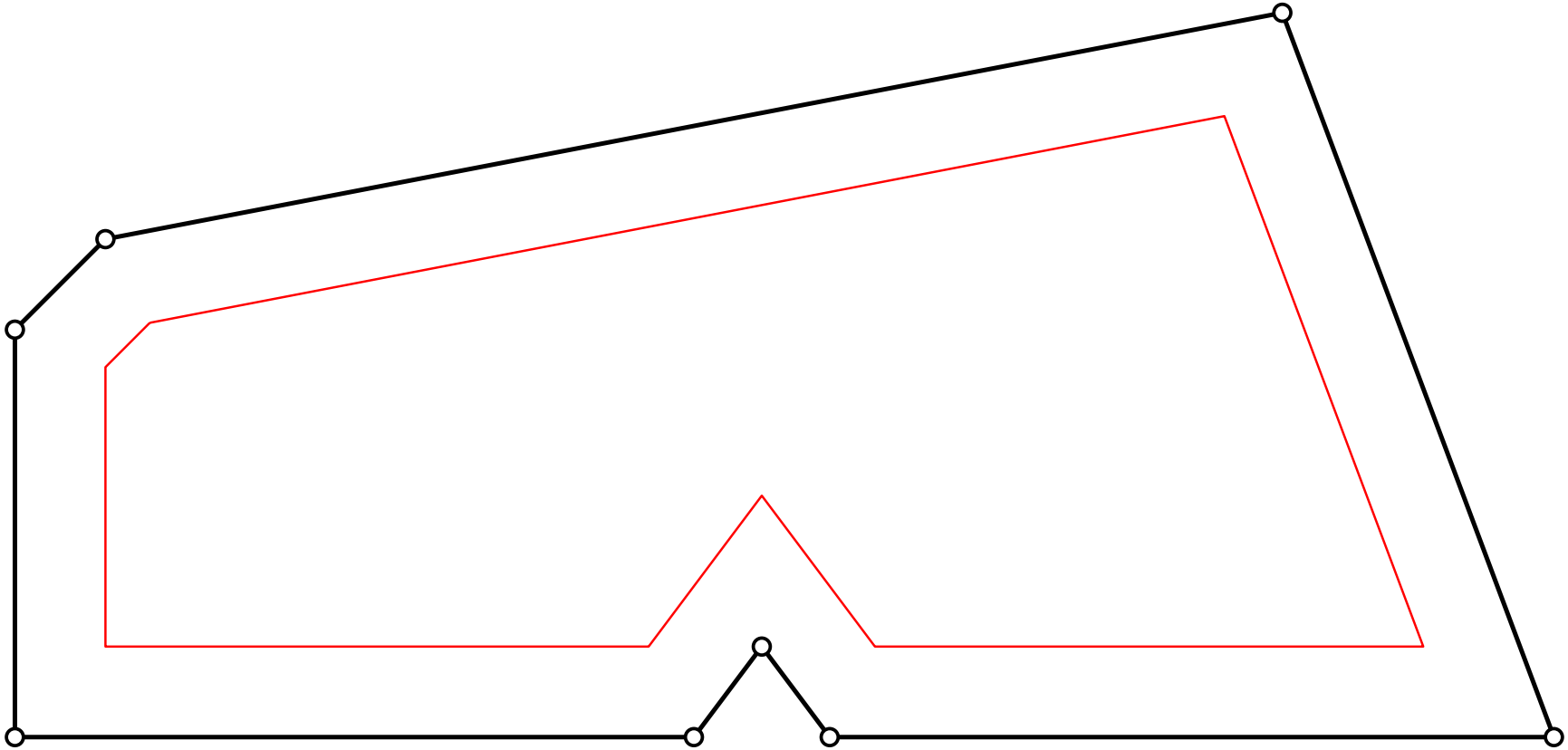
Straight skeleton



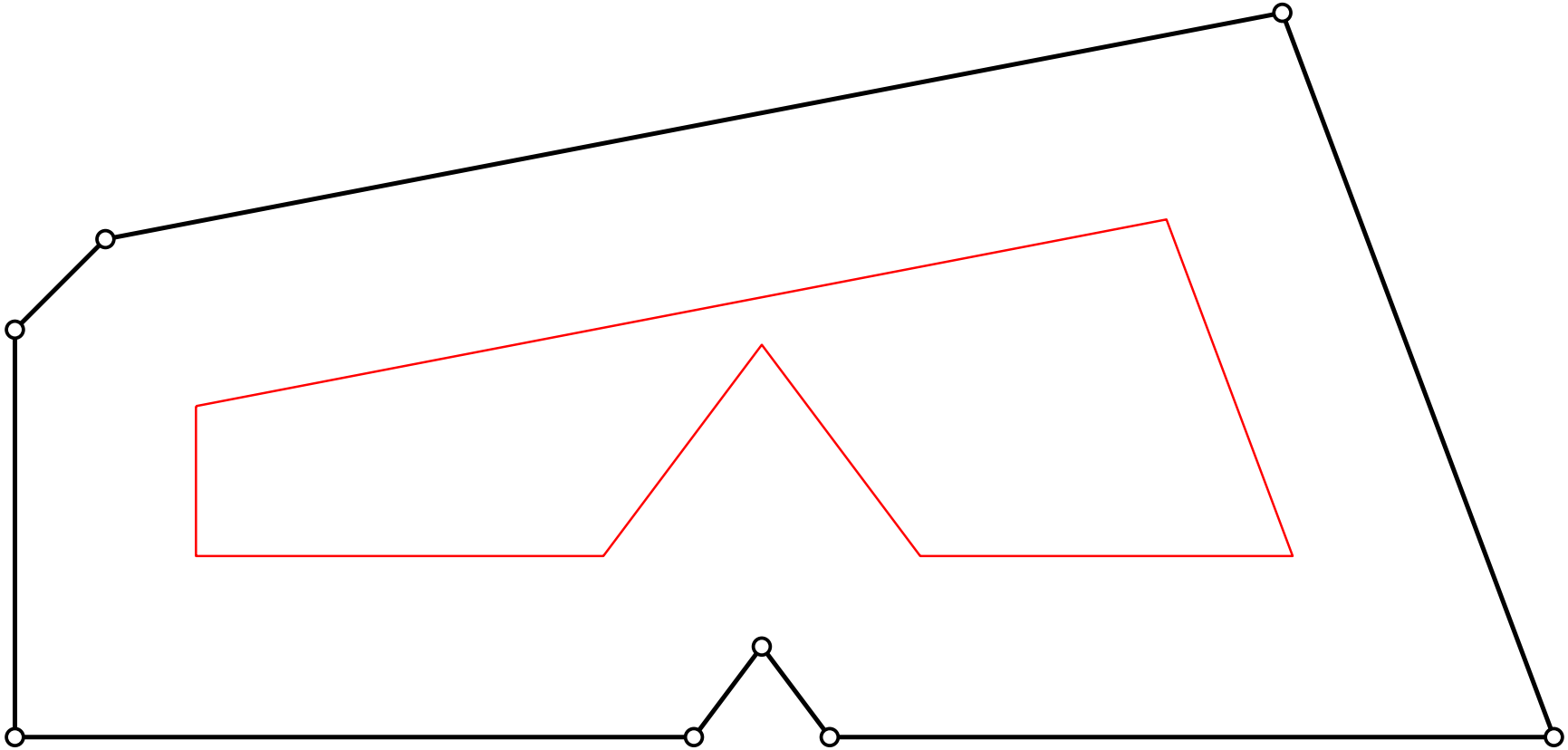
Edges move inward

Image after: Peter Palfrader, `http://www.palfrader.org/research/`

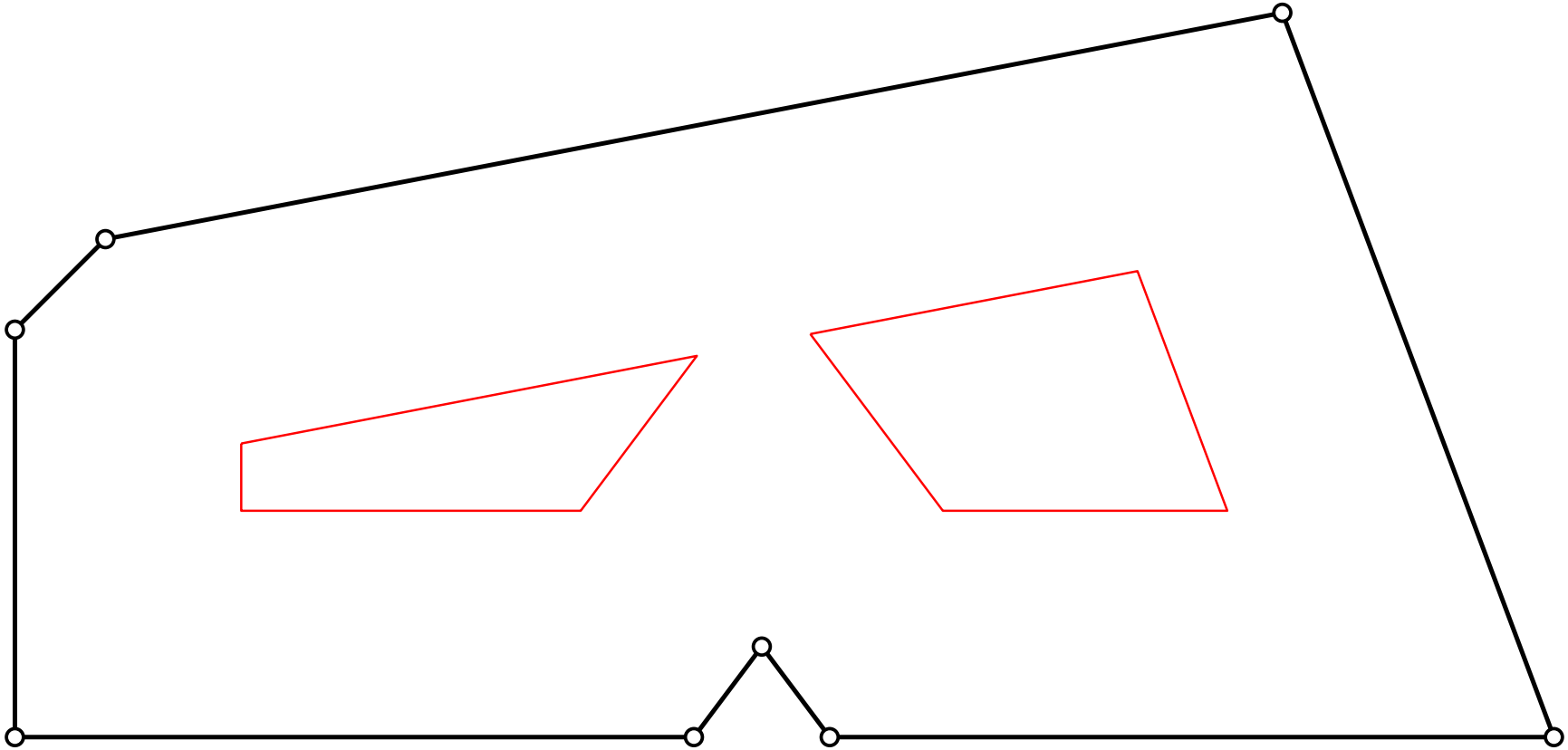
Straight skeleton



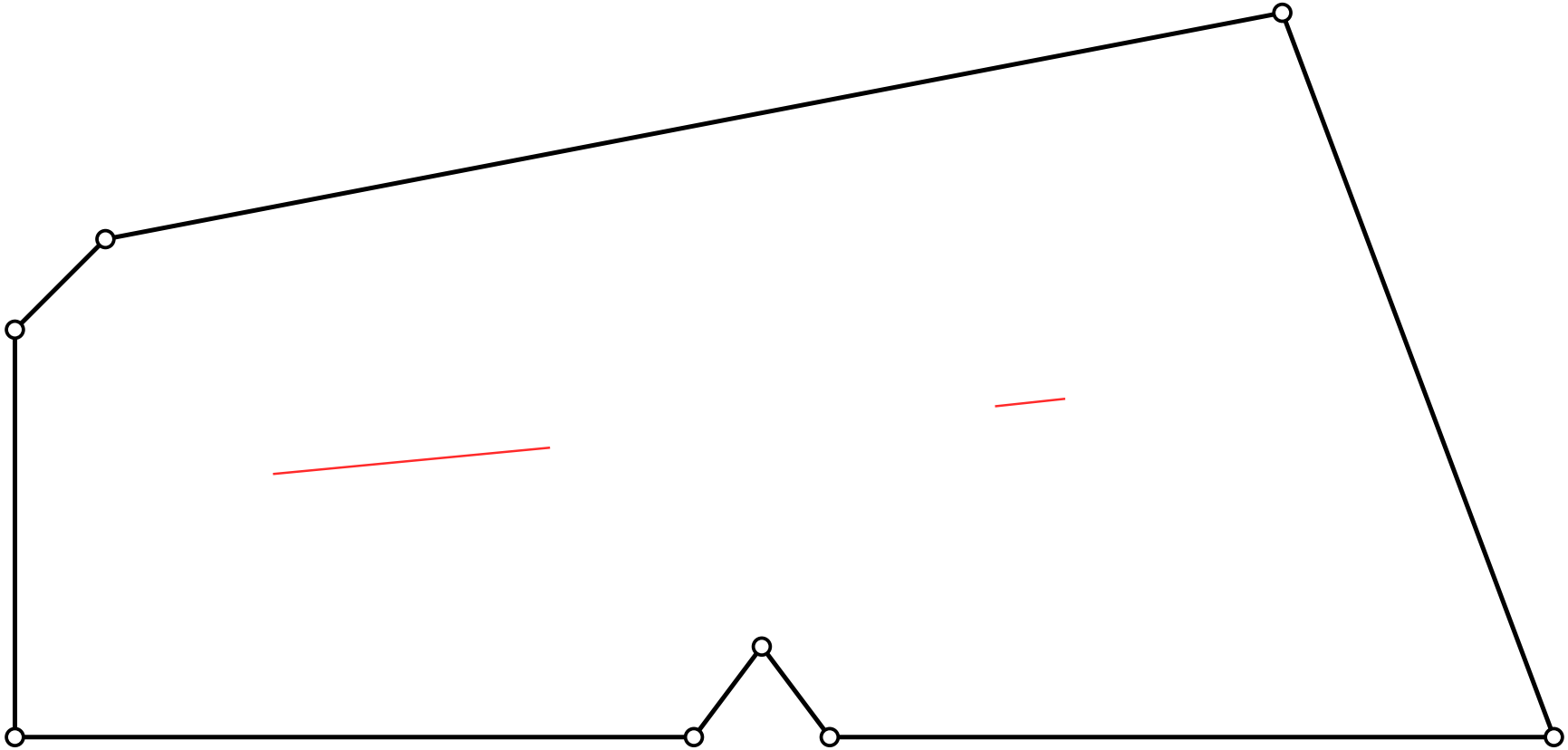
Straight skeleton



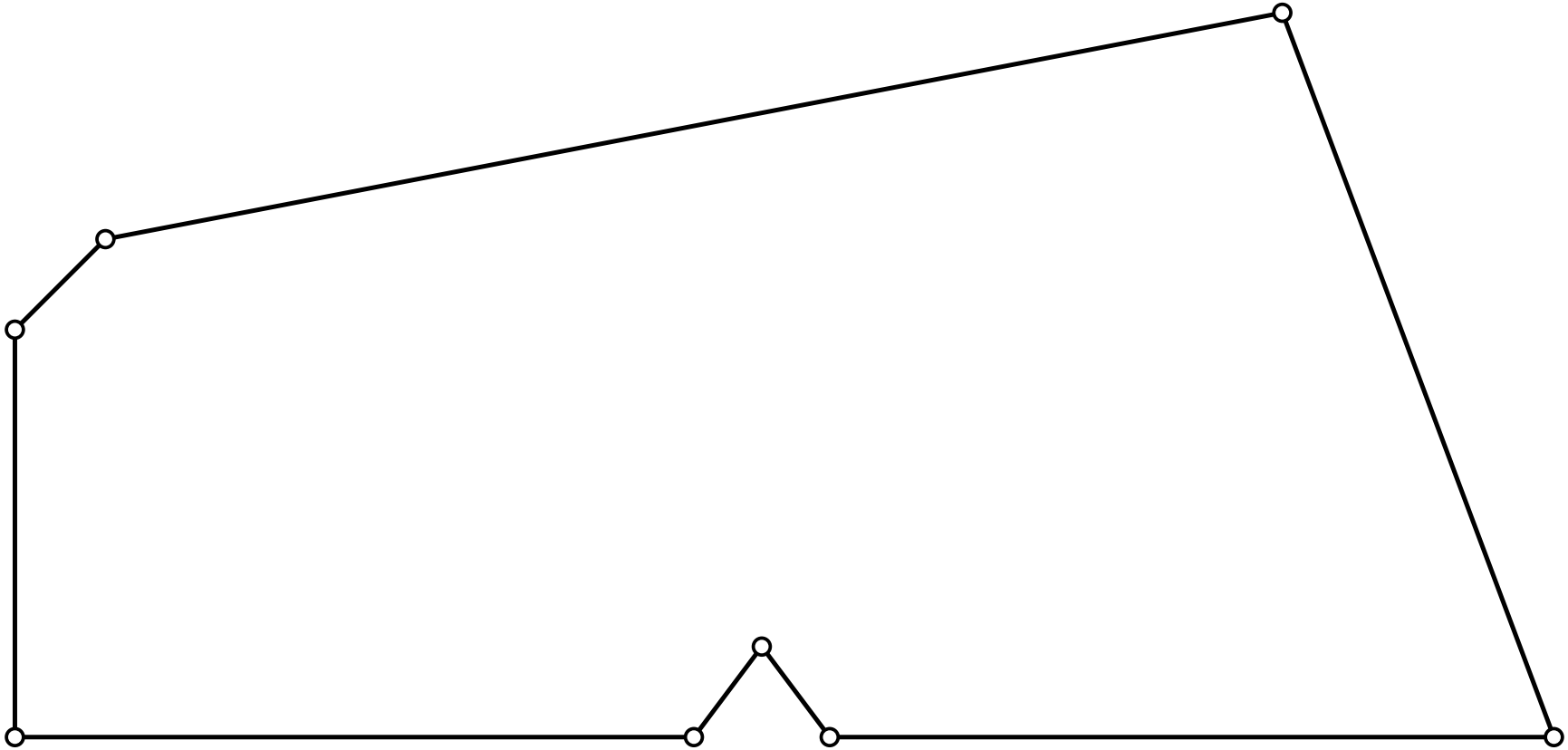
Straight skeleton



Straight skeleton

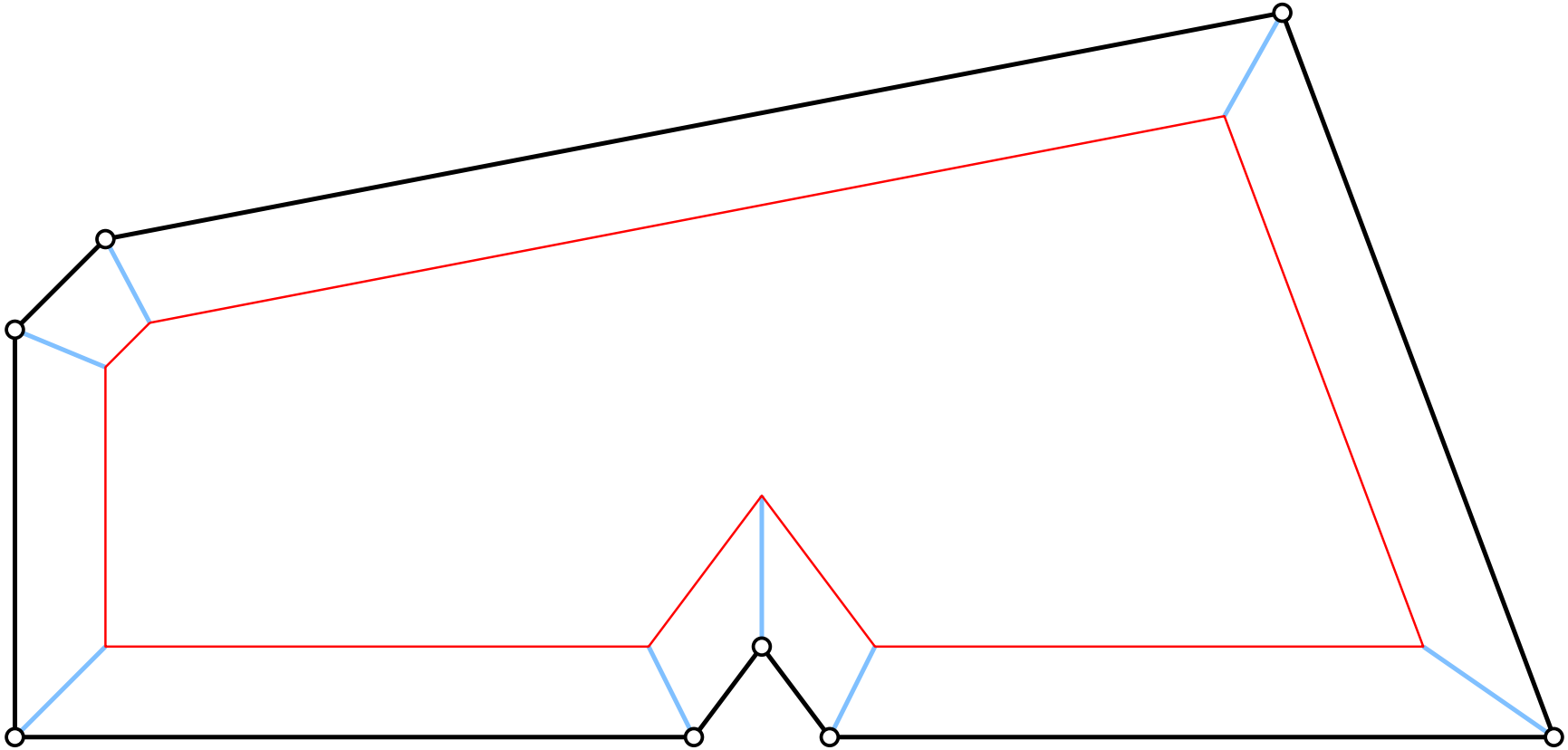


Straight skeleton

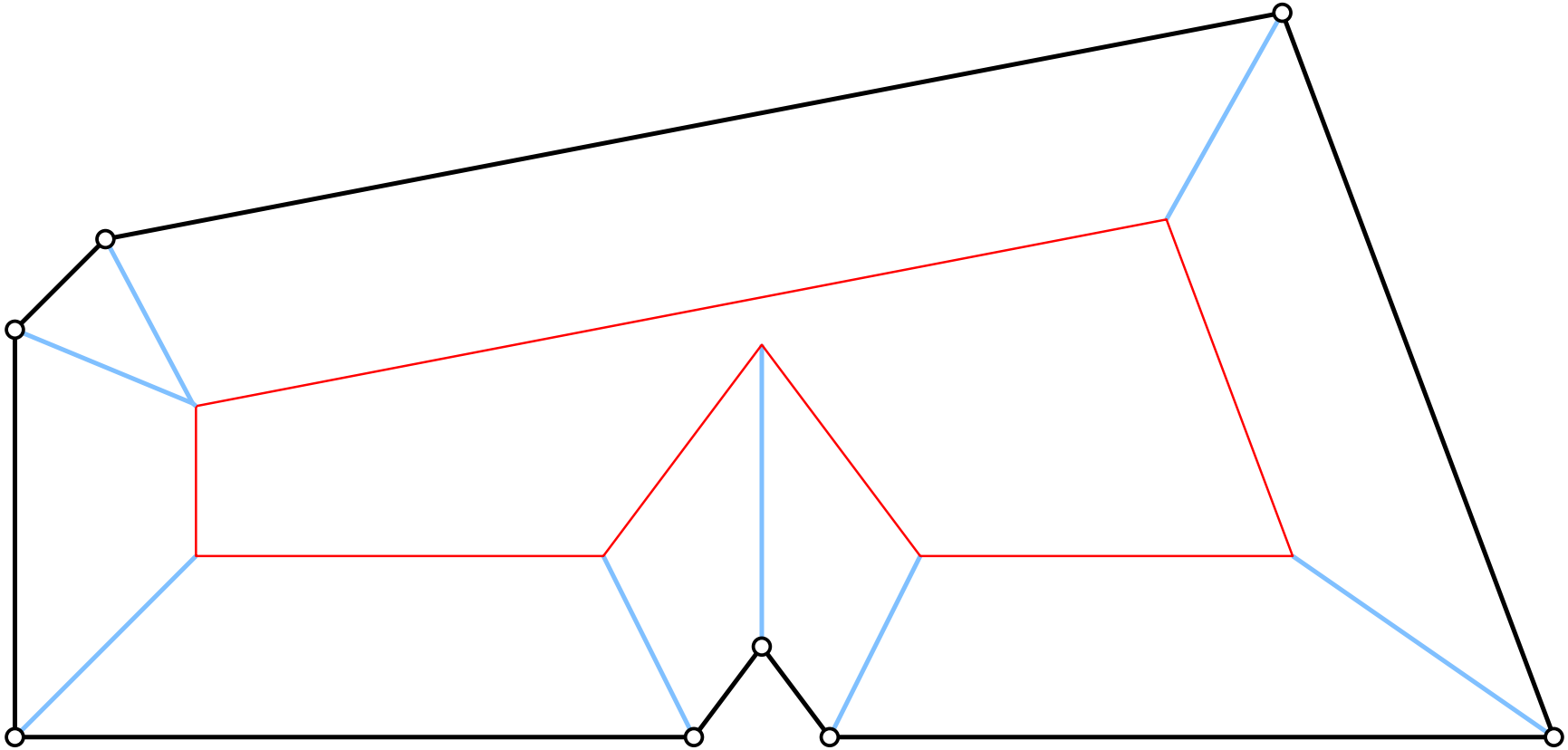


Vertices also move (some faster than others)

Straight skeleton

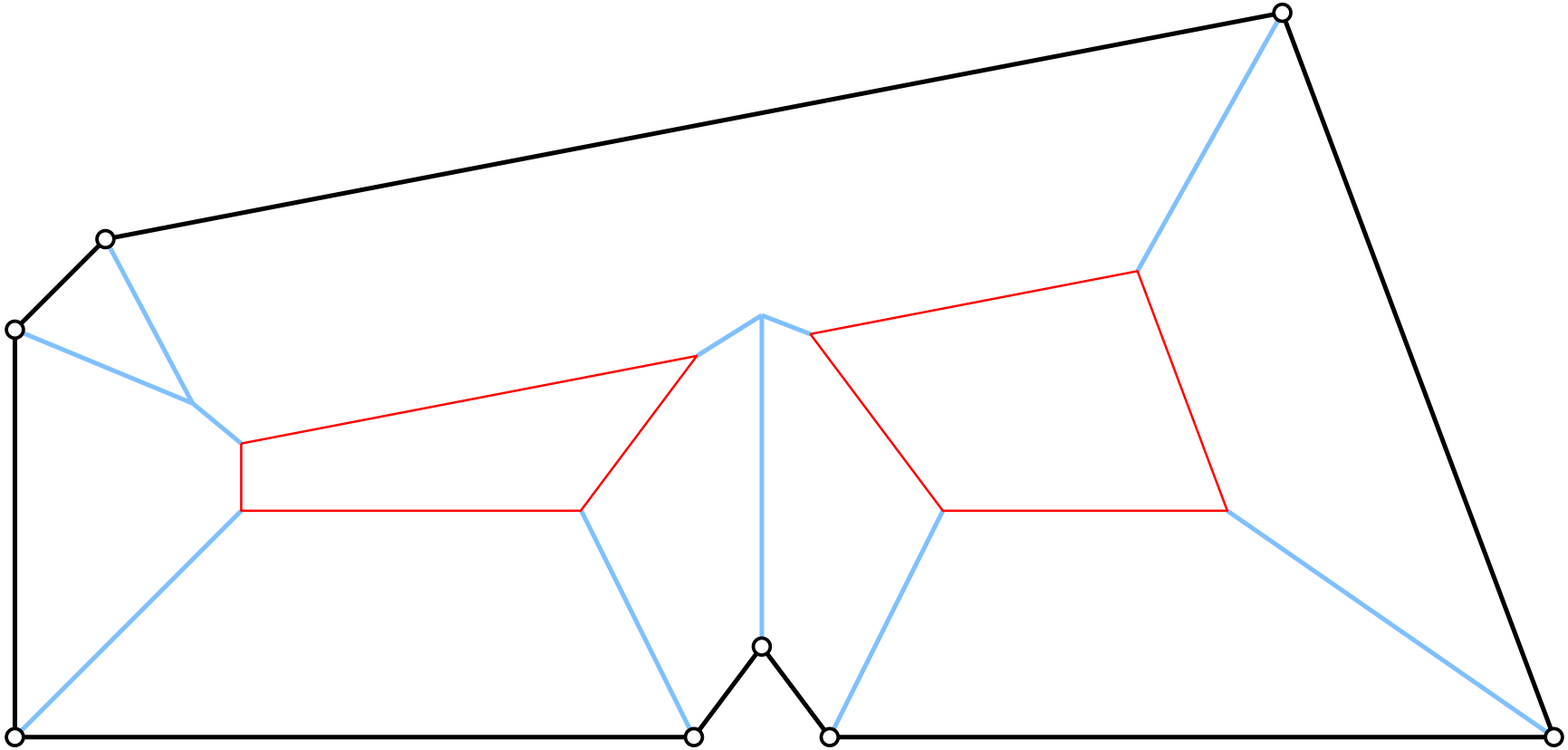


Straight skeleton

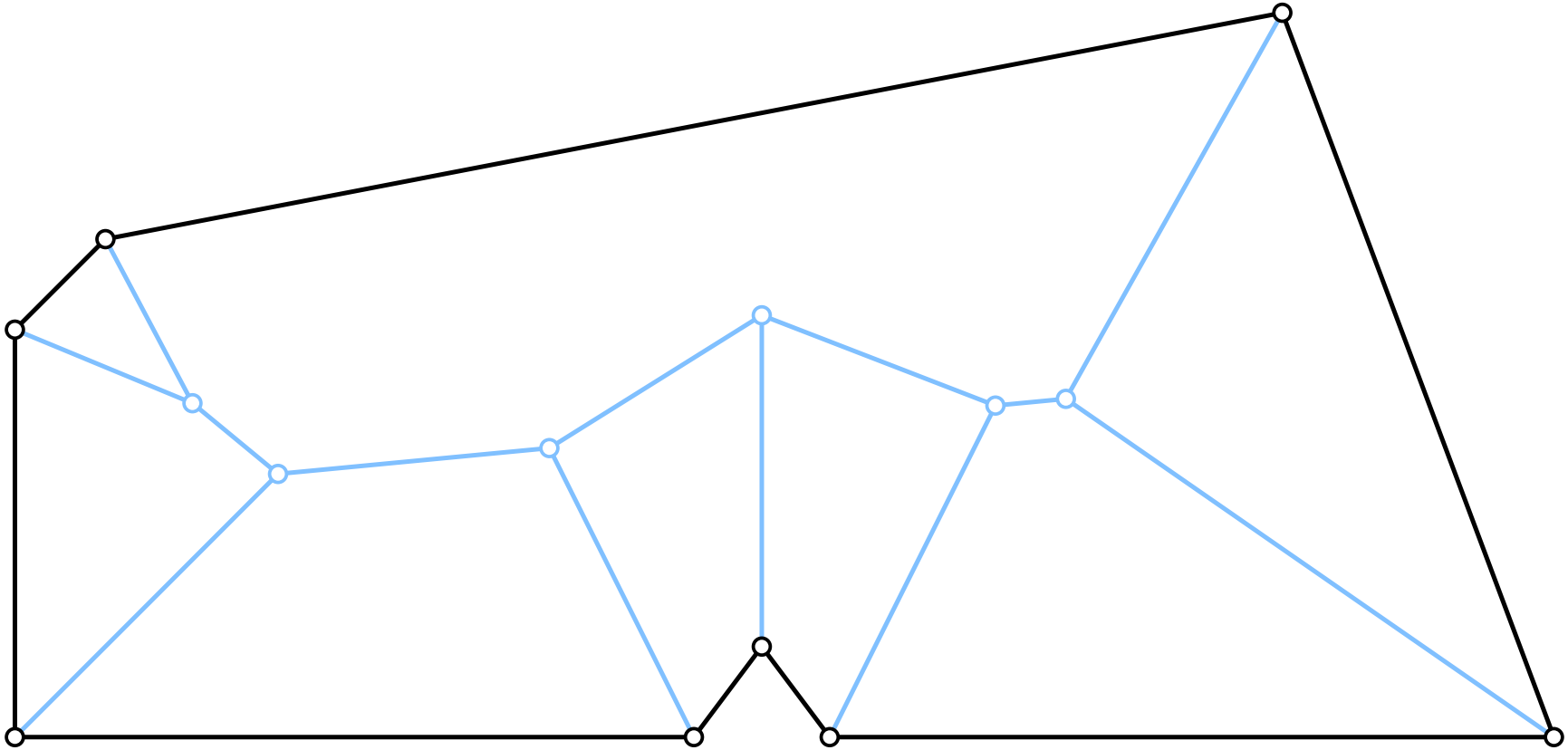


Polygon can be split, if vertex collides with non-adjacent edge

Straight skeleton



Straight skeleton



Straight skeleton

- Definition can be generalized for Planar Straight Line Graph (PSLG)
- Elegant algorithm for construction (Aichholzer and Aurenhammer, 1996; Palfrader, 2013):
  1. Triangulate regions between input segments
  2. Moving vertices does change size of triangles
  3. Triangle collapse (zero size) indicates change in structure of Straight Skeleton

Straight skeleton

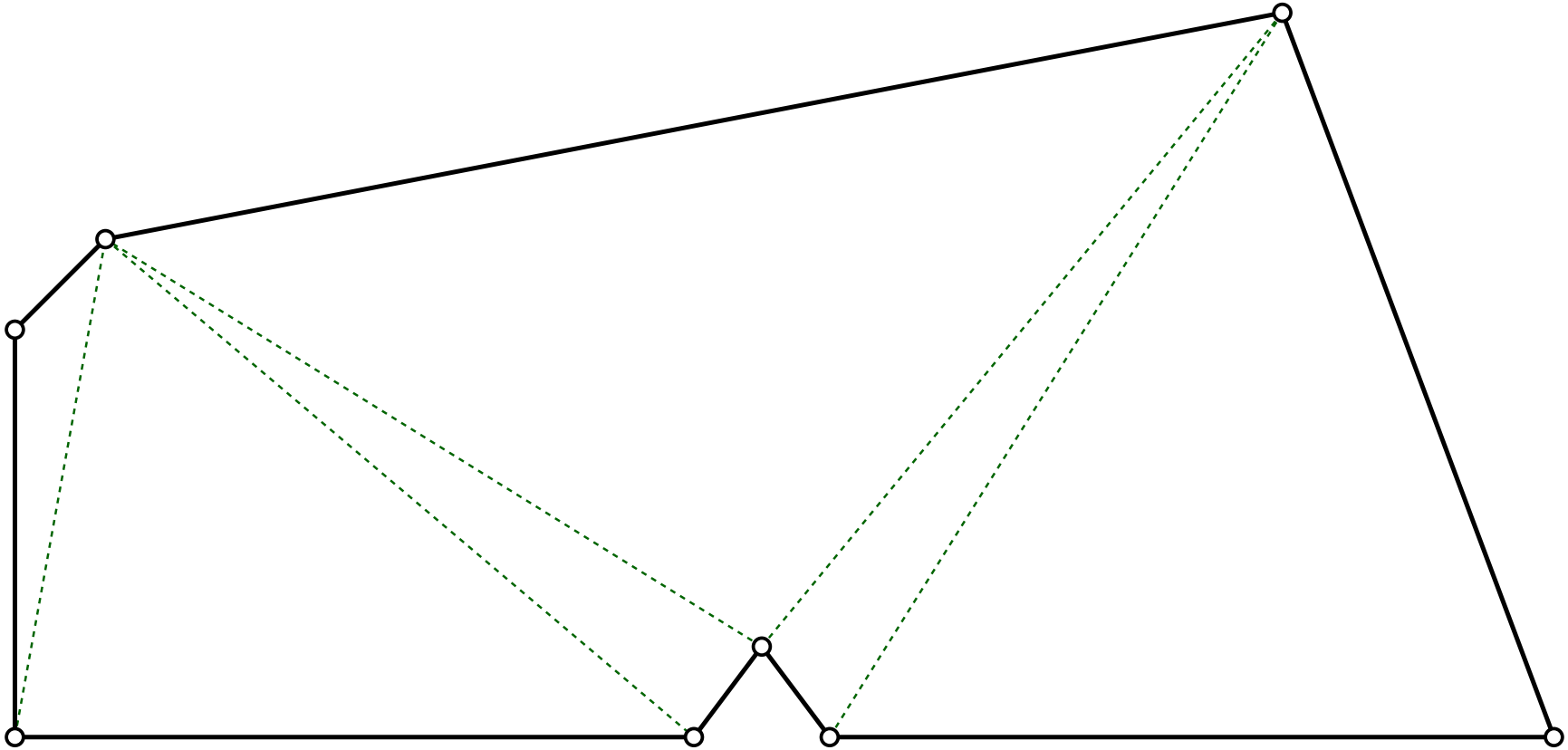
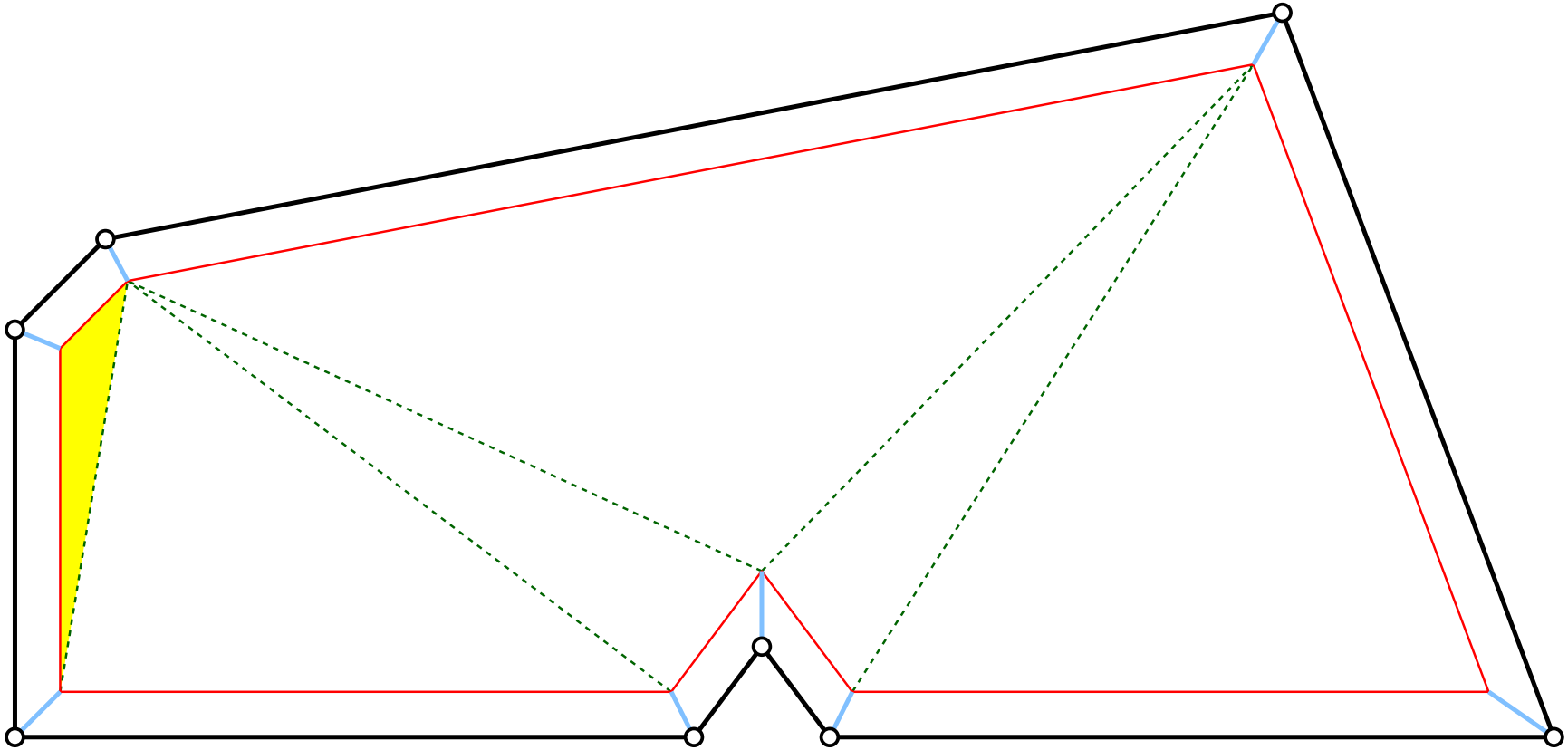
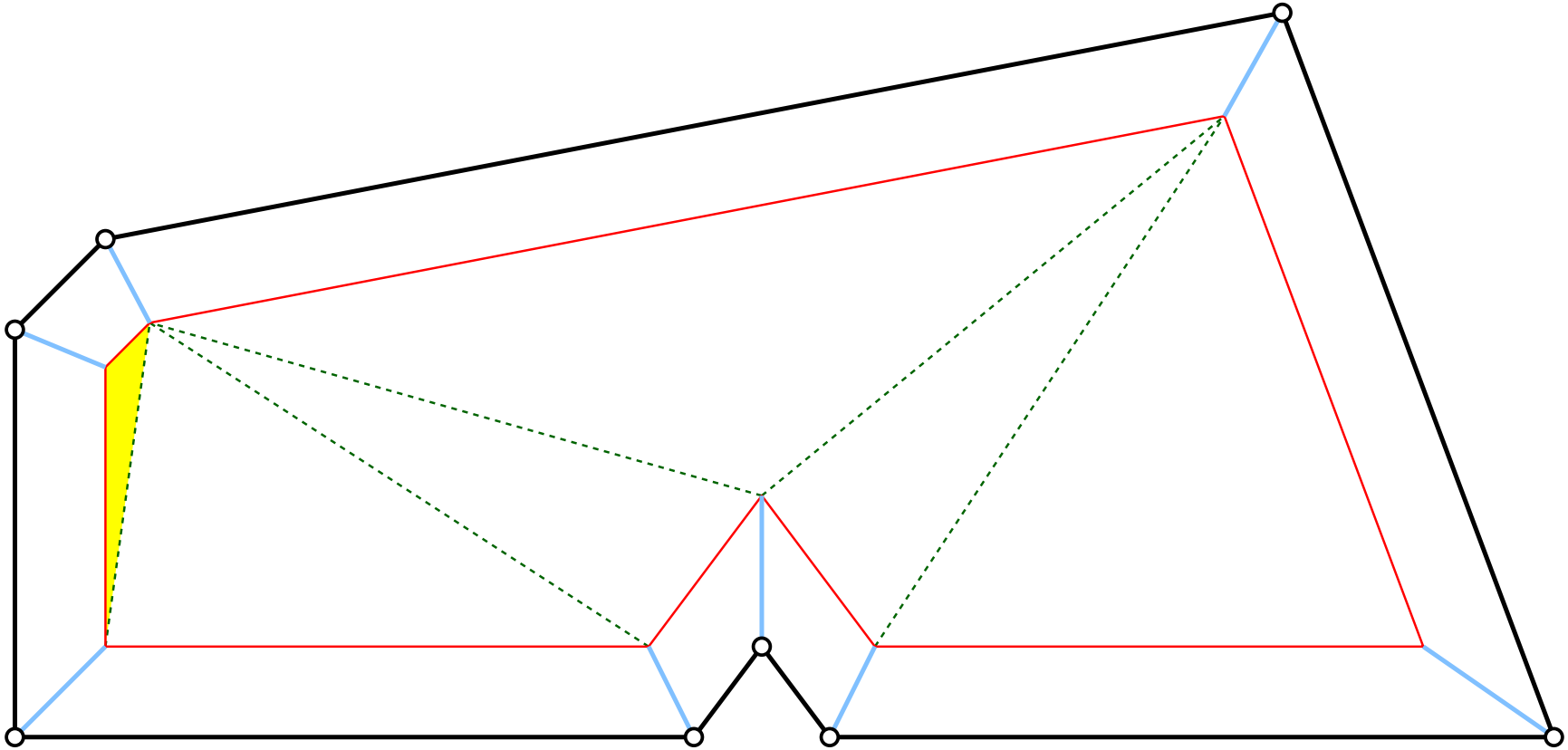


Image credit: Peter Palfrader

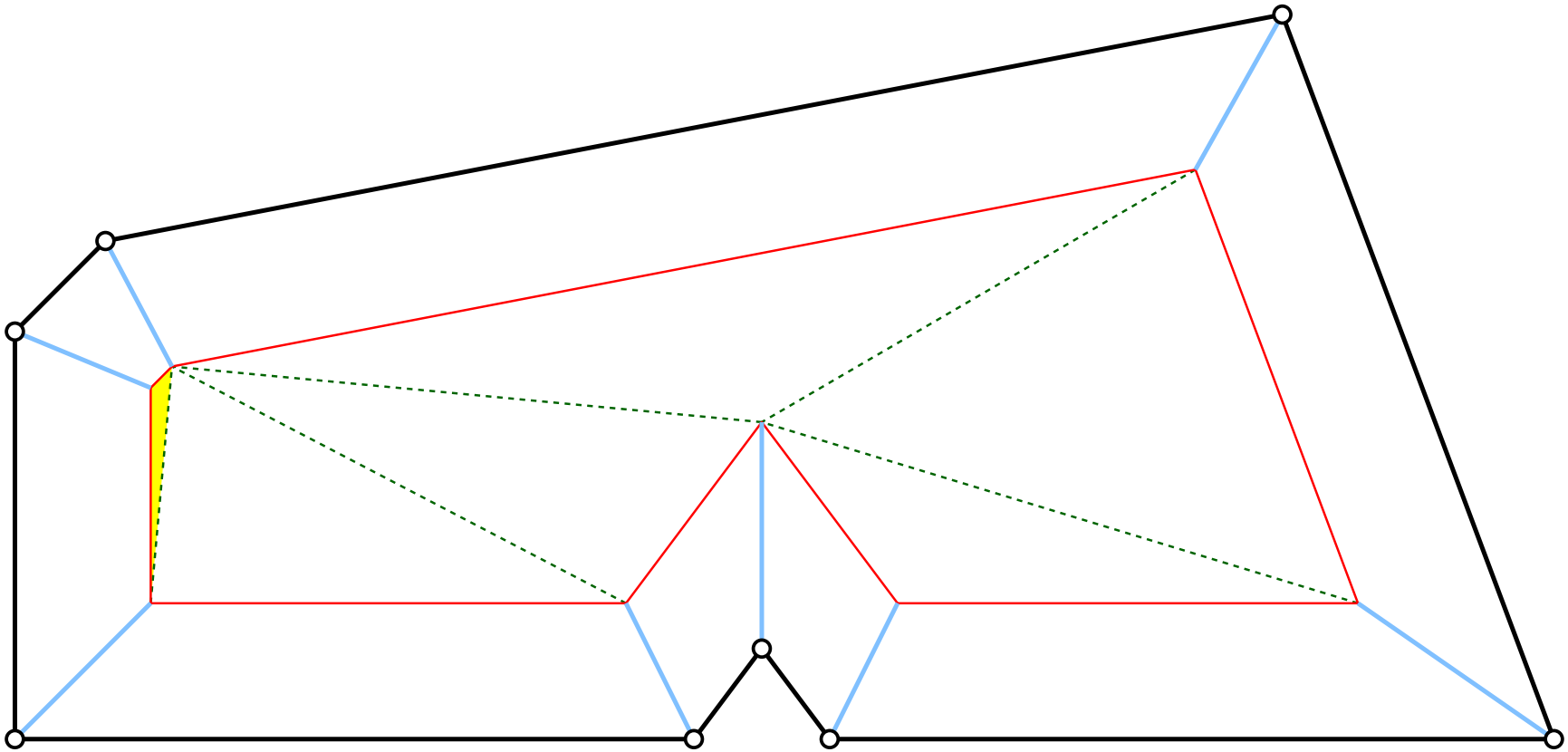
Straight skeleton



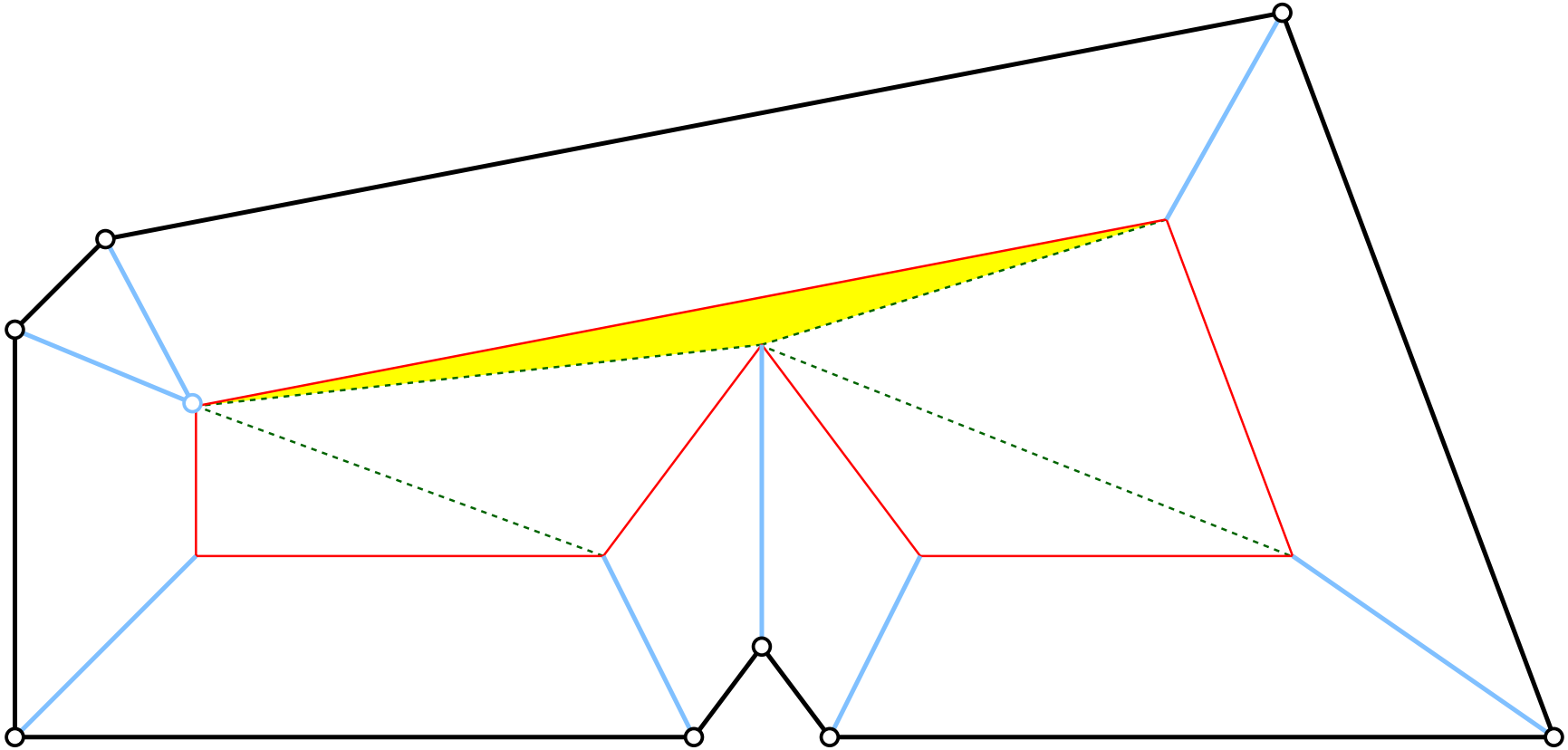
Straight skeleton



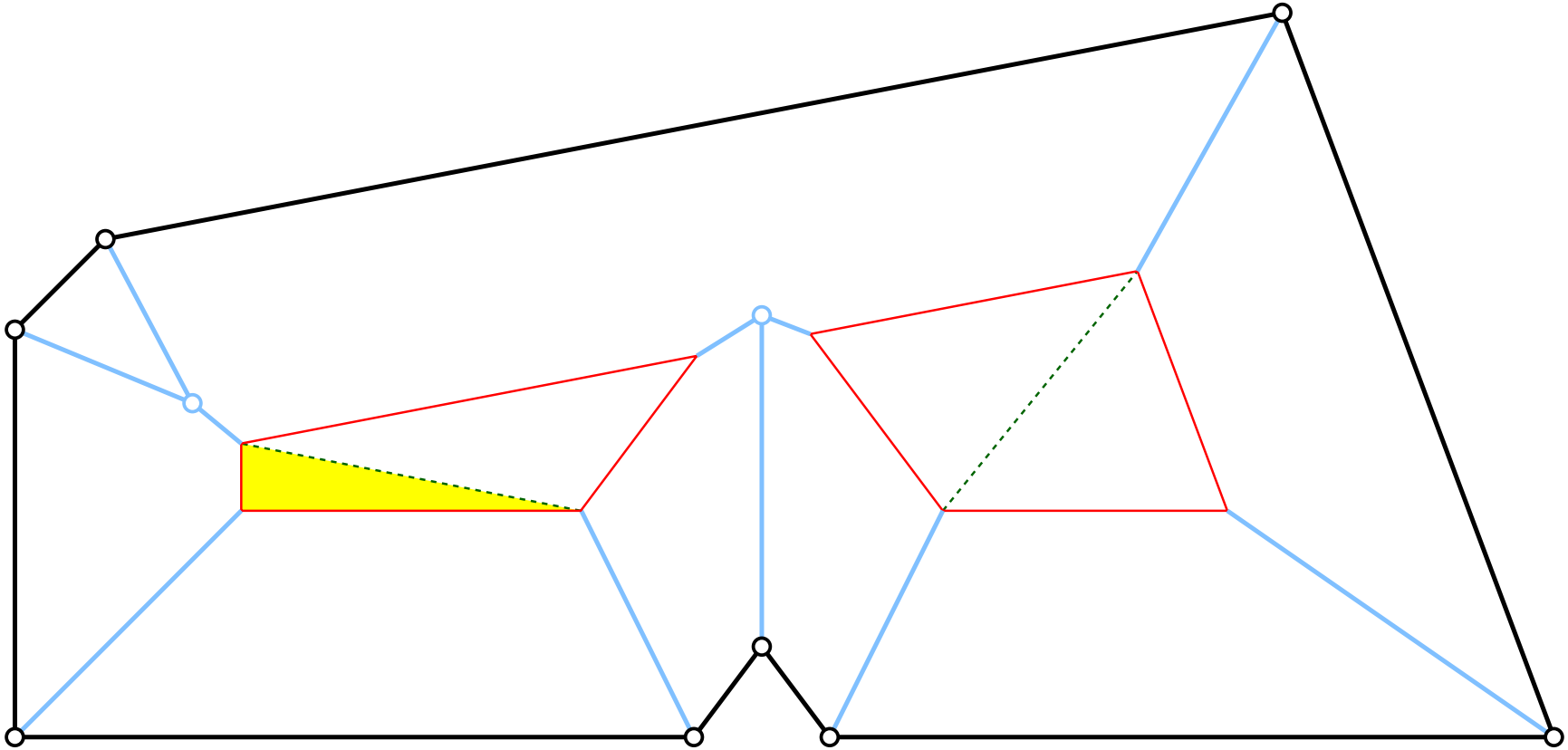
Straight skeleton



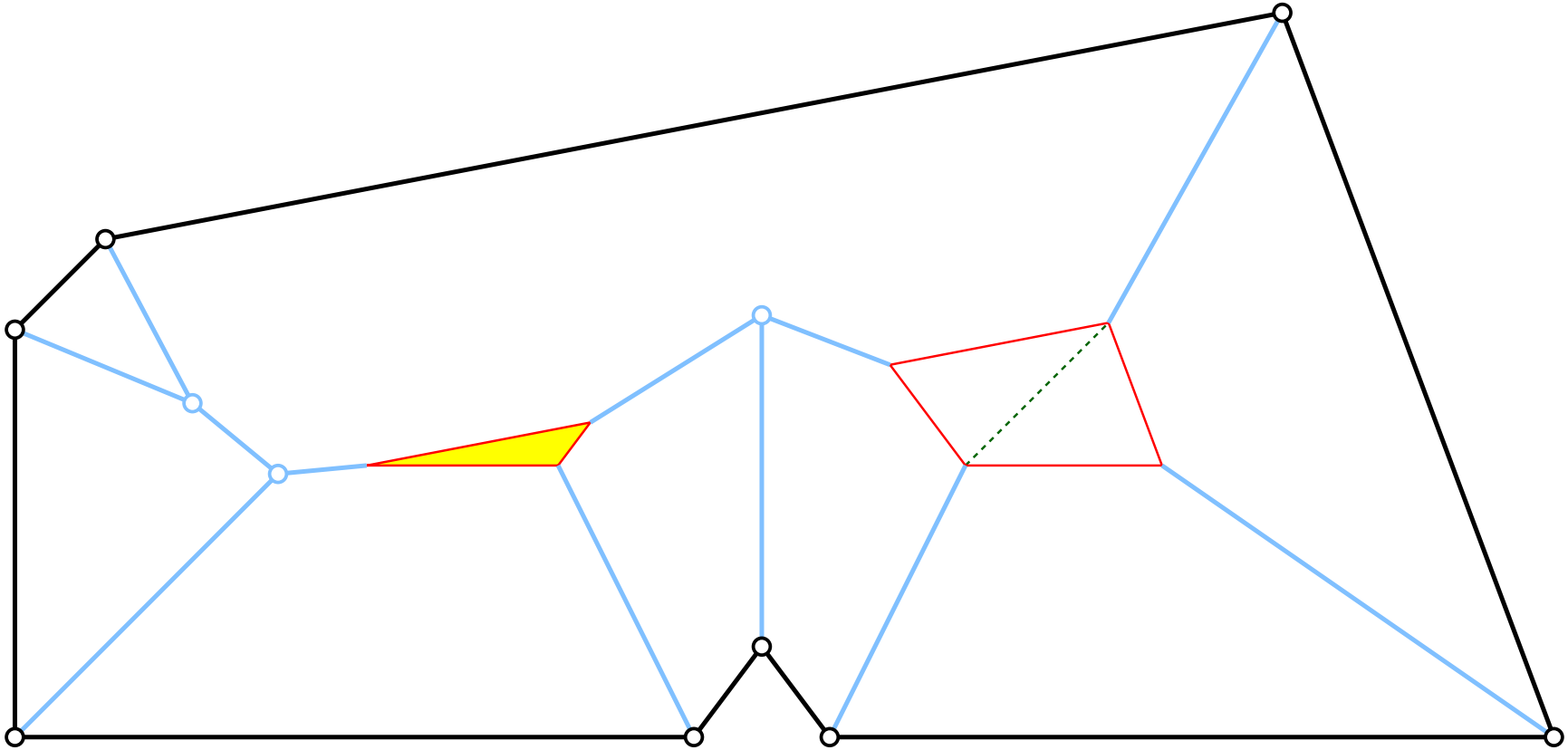
Straight skeleton



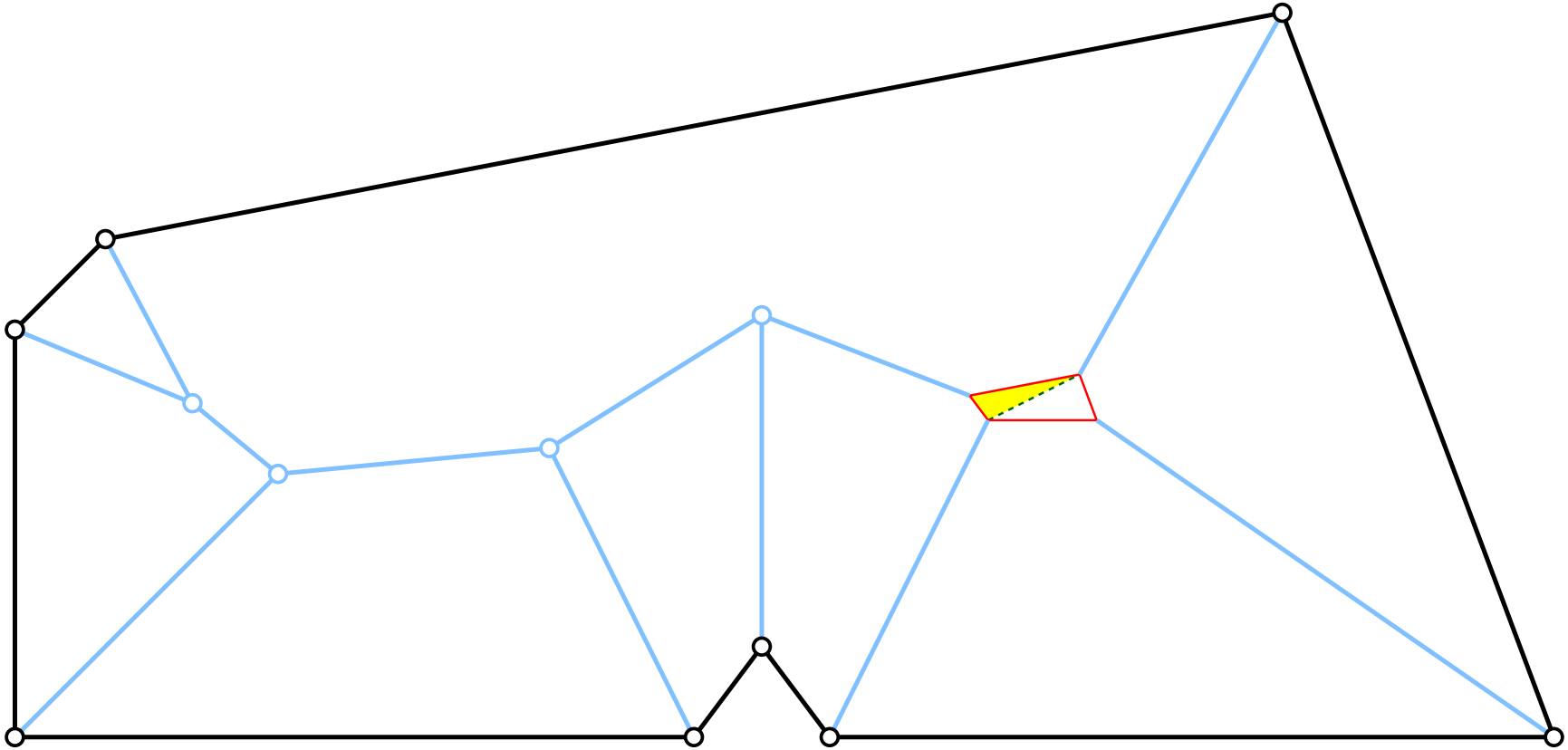
Straight skeleton



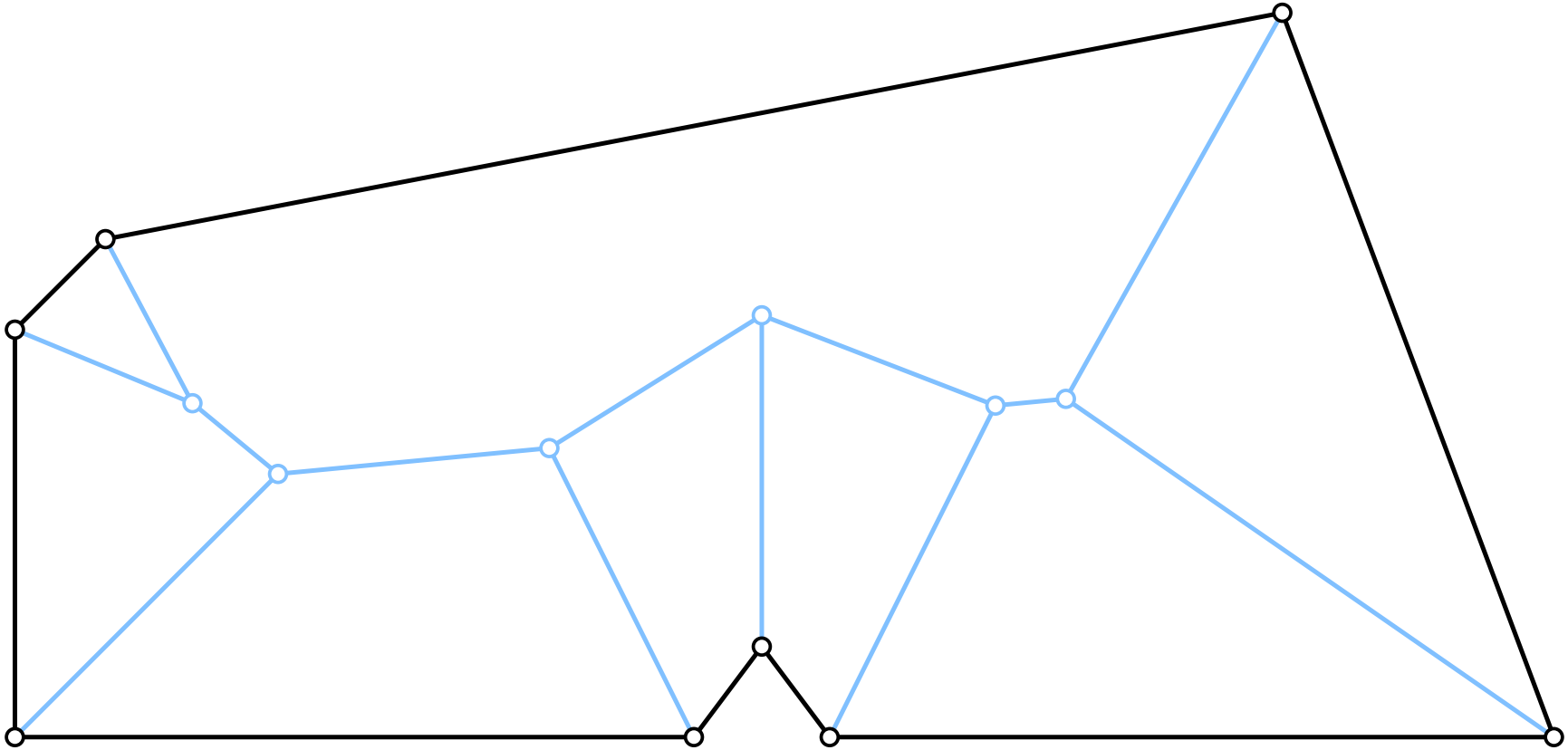
Straight skeleton



Straight skeleton



Straight skeleton



Offset curves

- Once traces of vertices are known, easy/cheap to generate offset curves (Palfrader and Held, 2015)
- For each vertex, keep track of neighbouring vertices at any time t

Offset curves

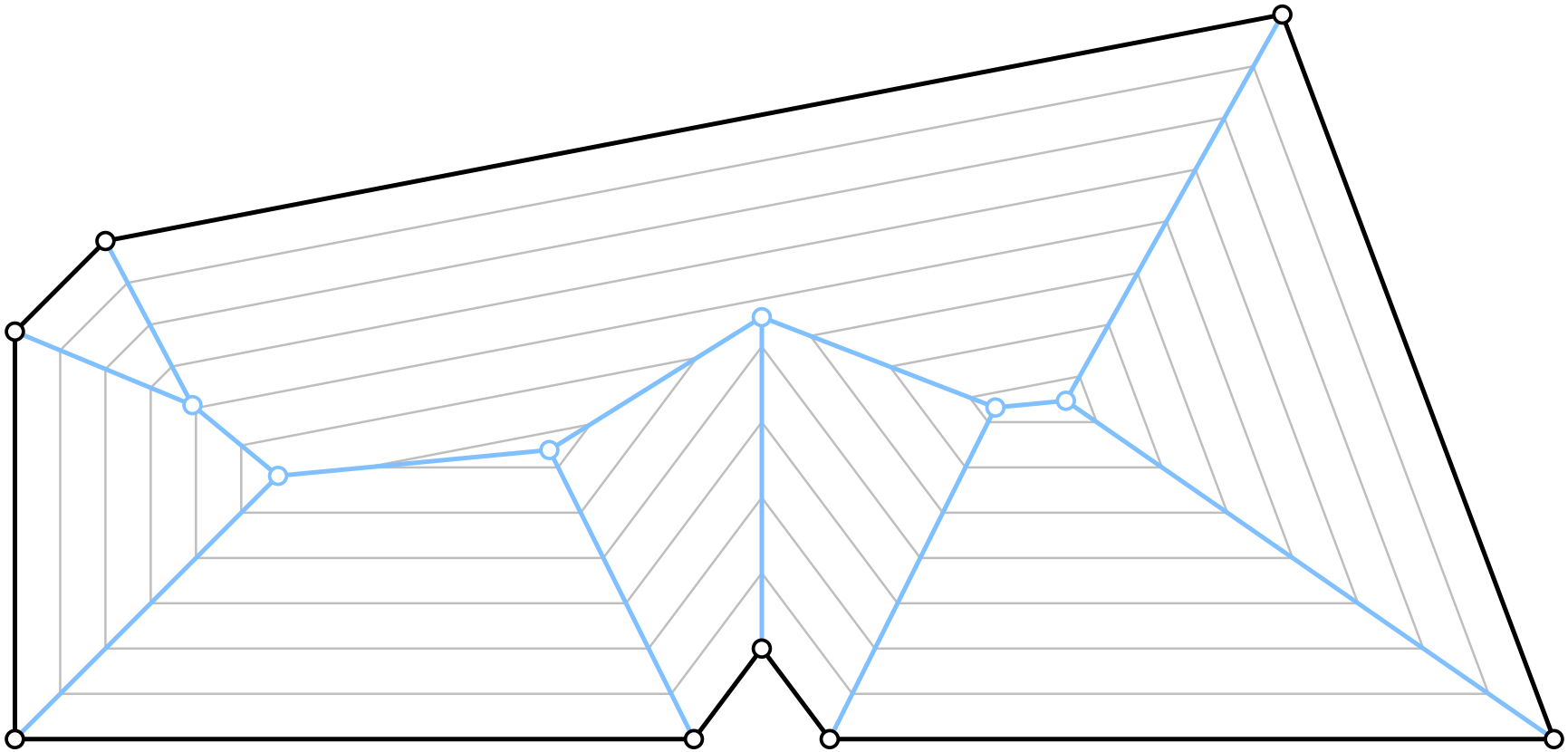


Image credit: Peter Palfrader

Area simplification

Algorithm for area simplification (Haunert and Sester, 2007):

1. Compute straight skeleton
2. Generate offset curves — ϵ inwards
3. Compute straight skeleton, on resulting shape
4. Generate offset curves — ϵ outwards

Area simplification

Algorithm for area simplification (Haunert and Sester, 2007):

1. Compute straight skeleton
2. Generate offset curves — ϵ inwards
3. Compute straight skeleton, on resulting shape
4. Generate offset curves — ϵ outwards

Note: Order of inwards — outwards can be reversed (giving different results)

Building simplification

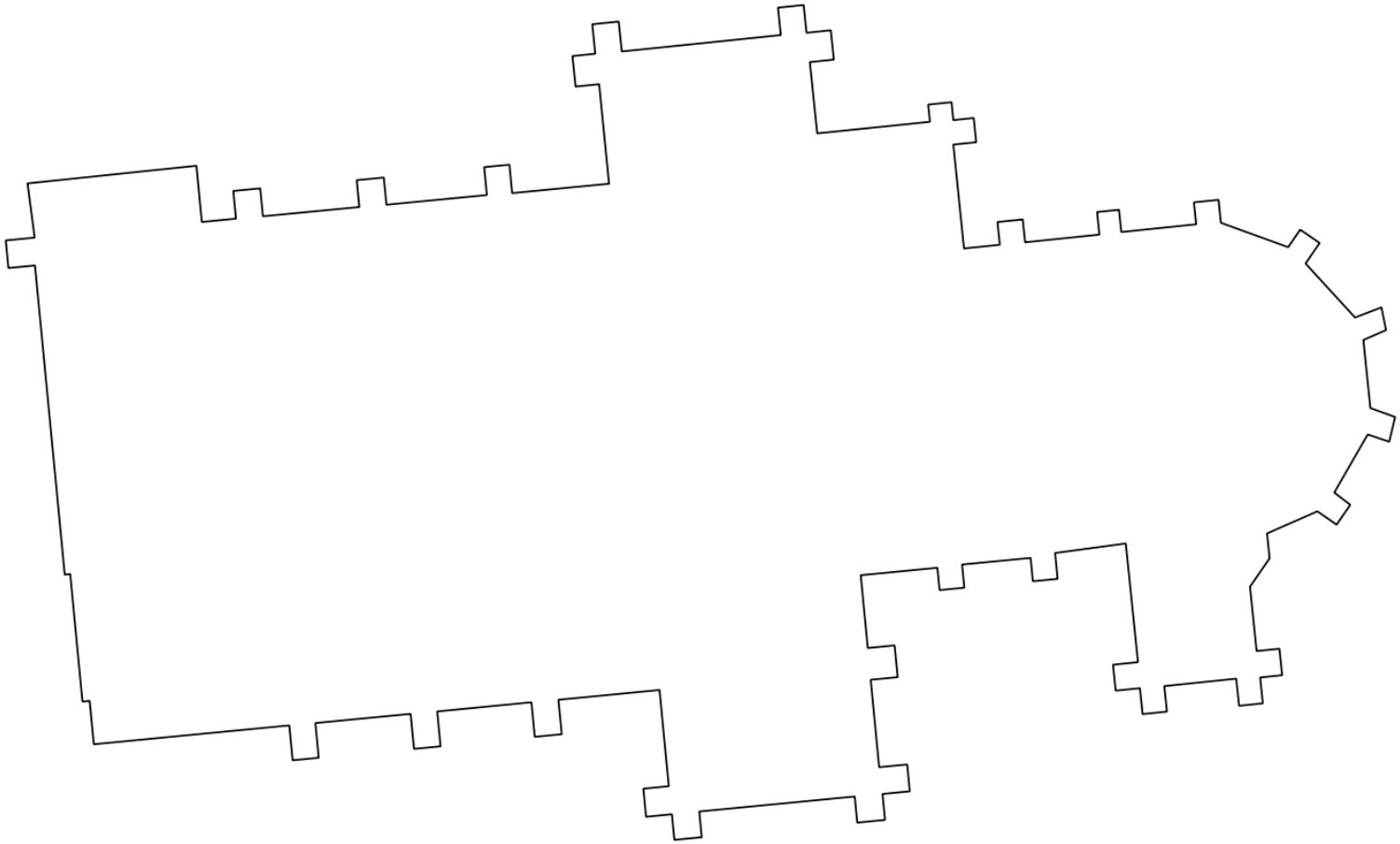


Figure: Input

Building simplification

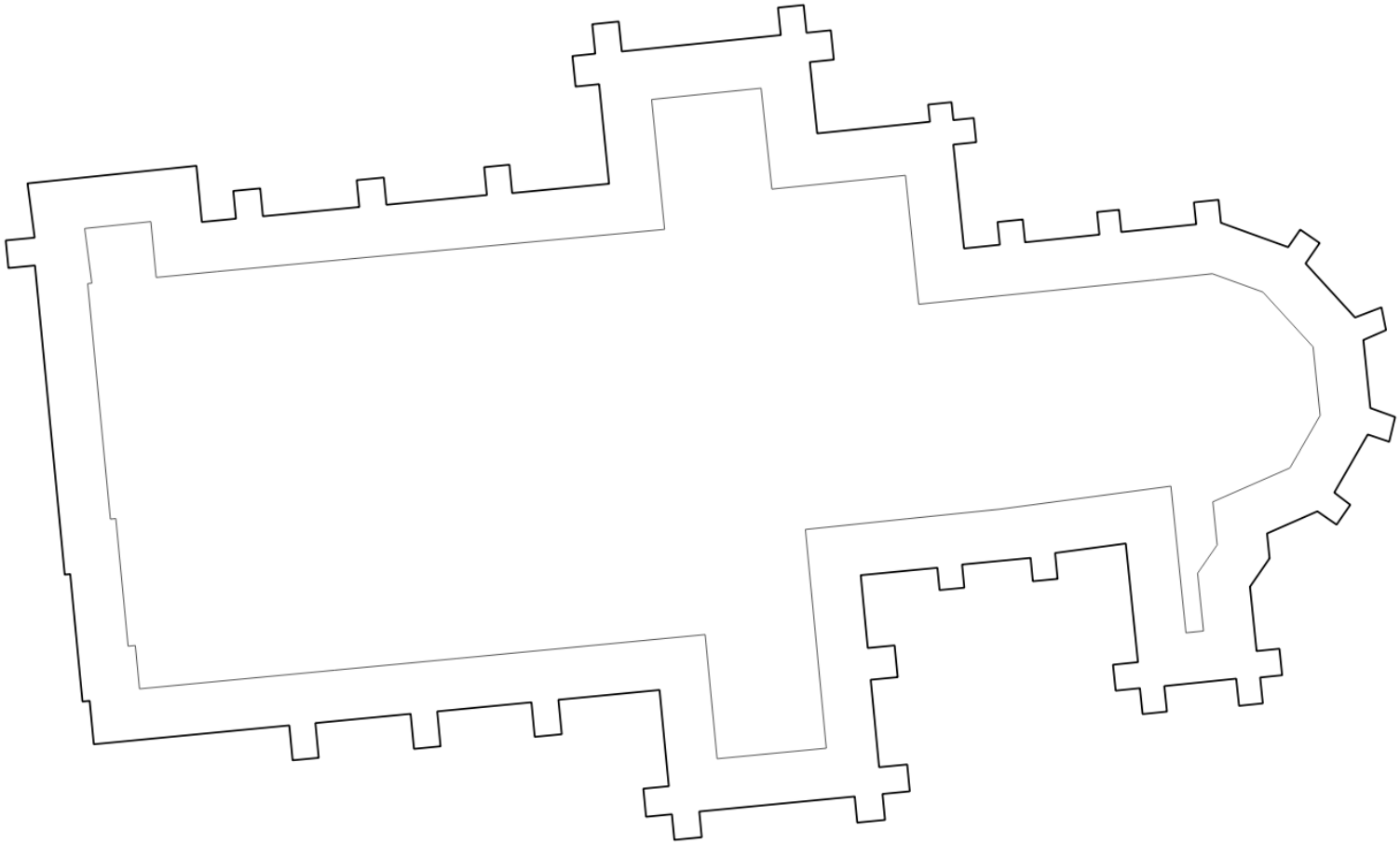


Figure: Offset curve inward

Building simplification

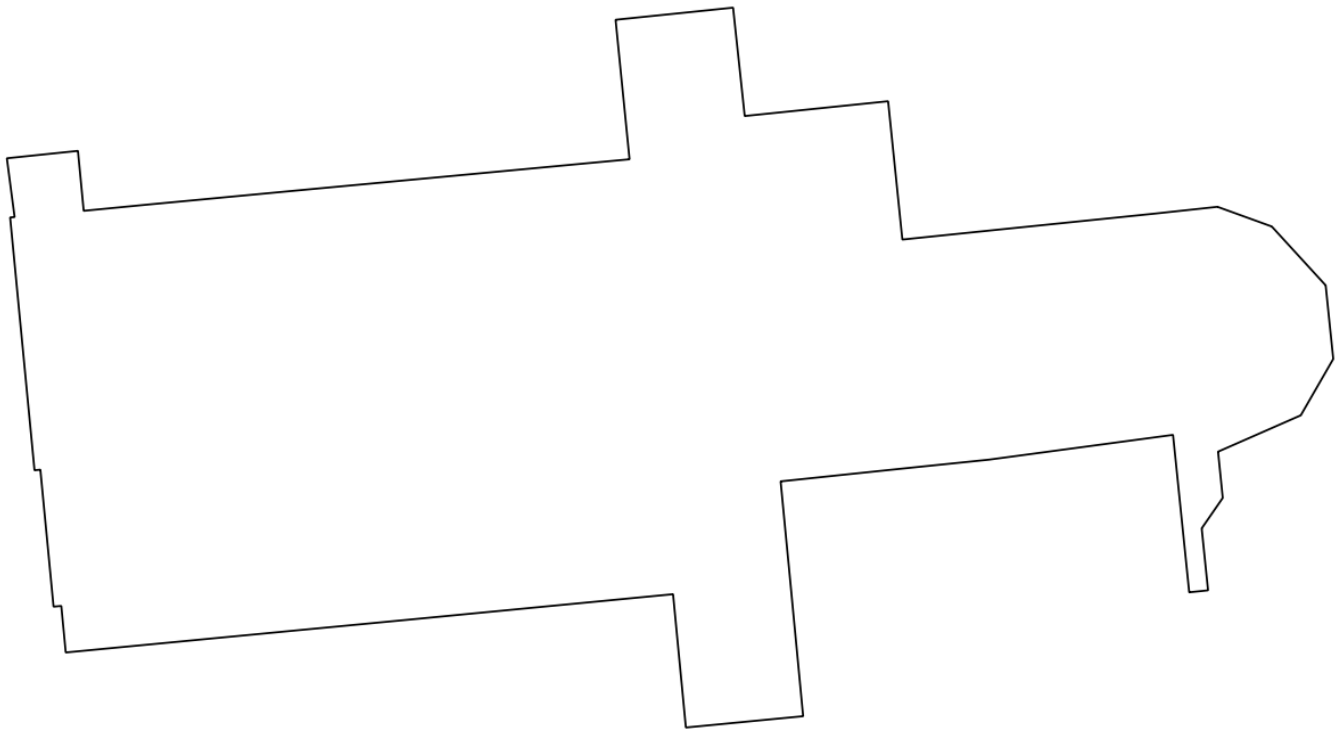


Figure: Input for 2nd step

Building simplification

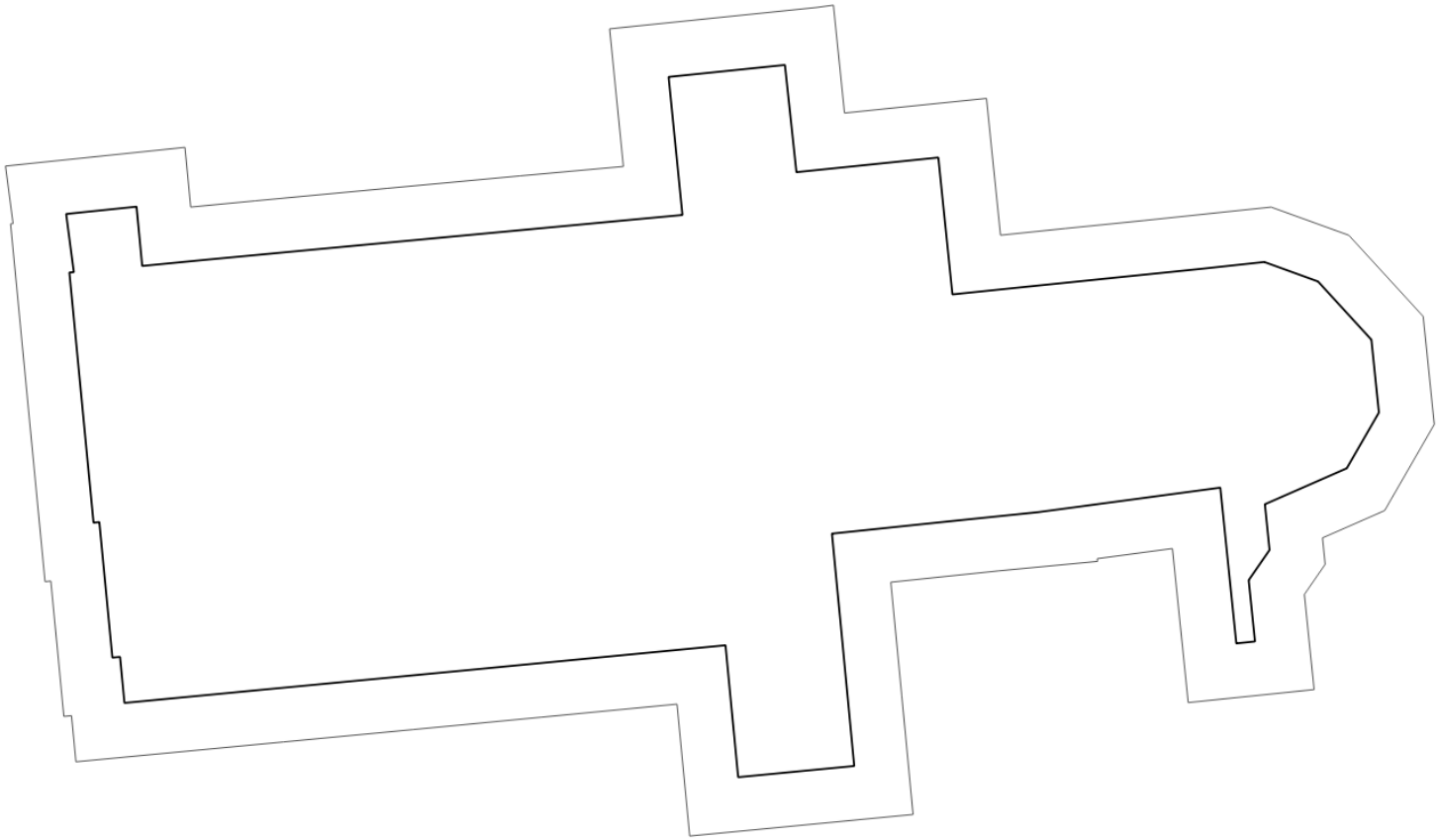


Figure: Offset curve outwards

Building simplification

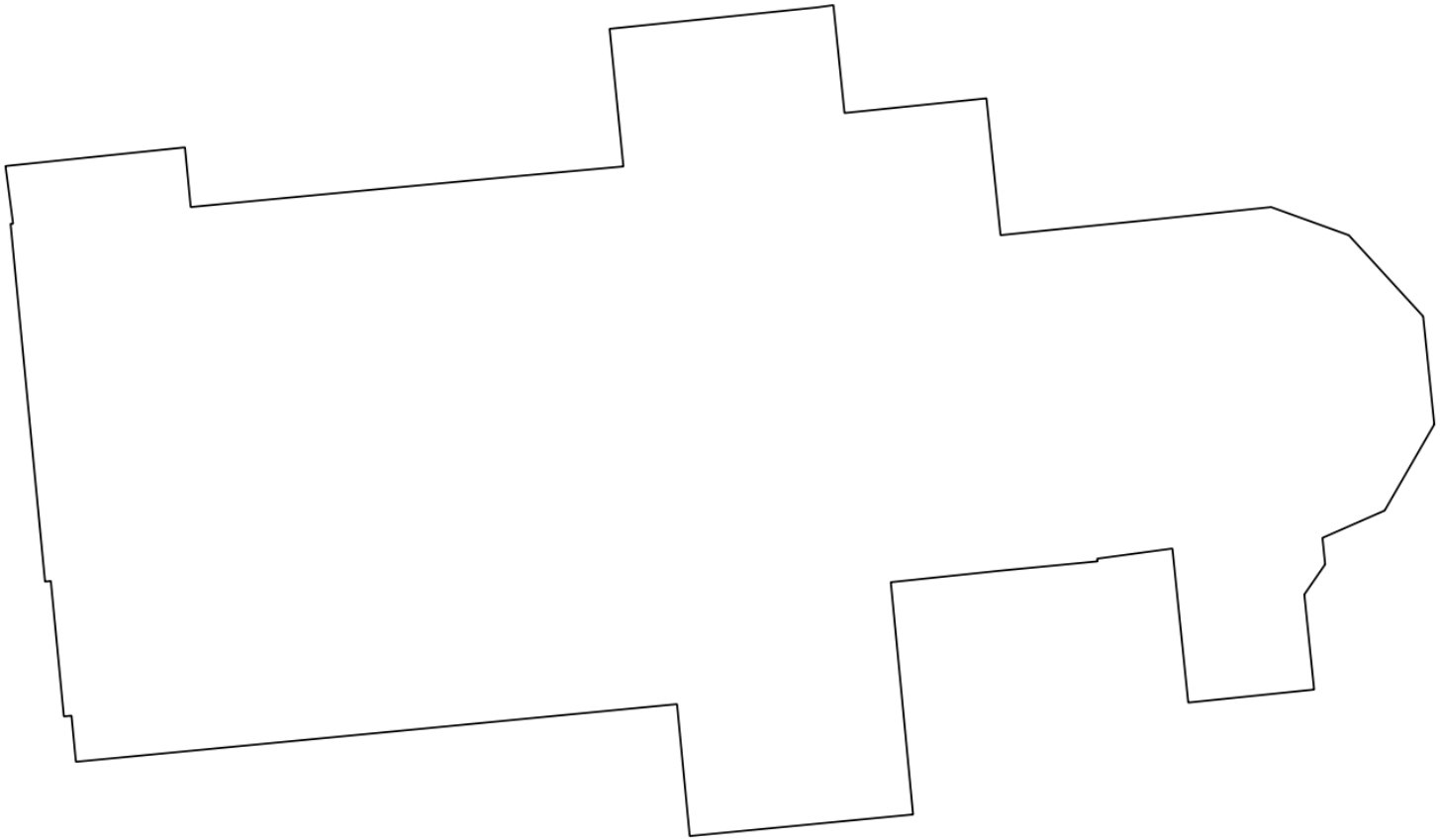


Figure: Result

Building simplification

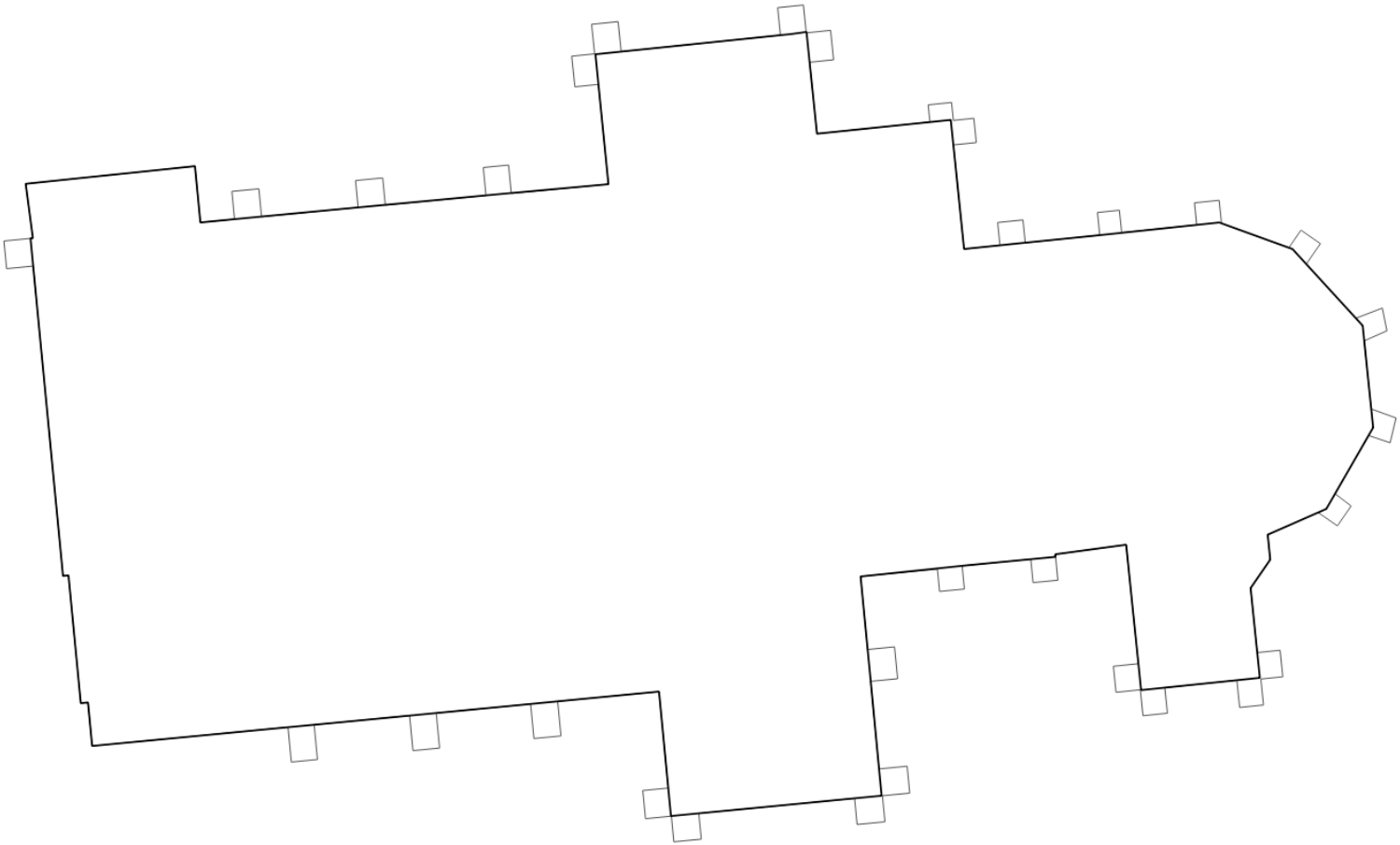


Figure: Result – Comparison with input

Building simplification

A variant of the algorithm:

1. Compute straight skeleton
2. Generate offset curves — $\frac{1}{2}\epsilon$ inwards
3. Compute straight skeleton, on resulting shape
4. Generate offset curves — ϵ outwards
5. Compute straight skeleton, on resulting shape
6. Generate offset curves — $\frac{1}{2}\epsilon$ inwards

Note: Order can be reversed again.

Results – One building

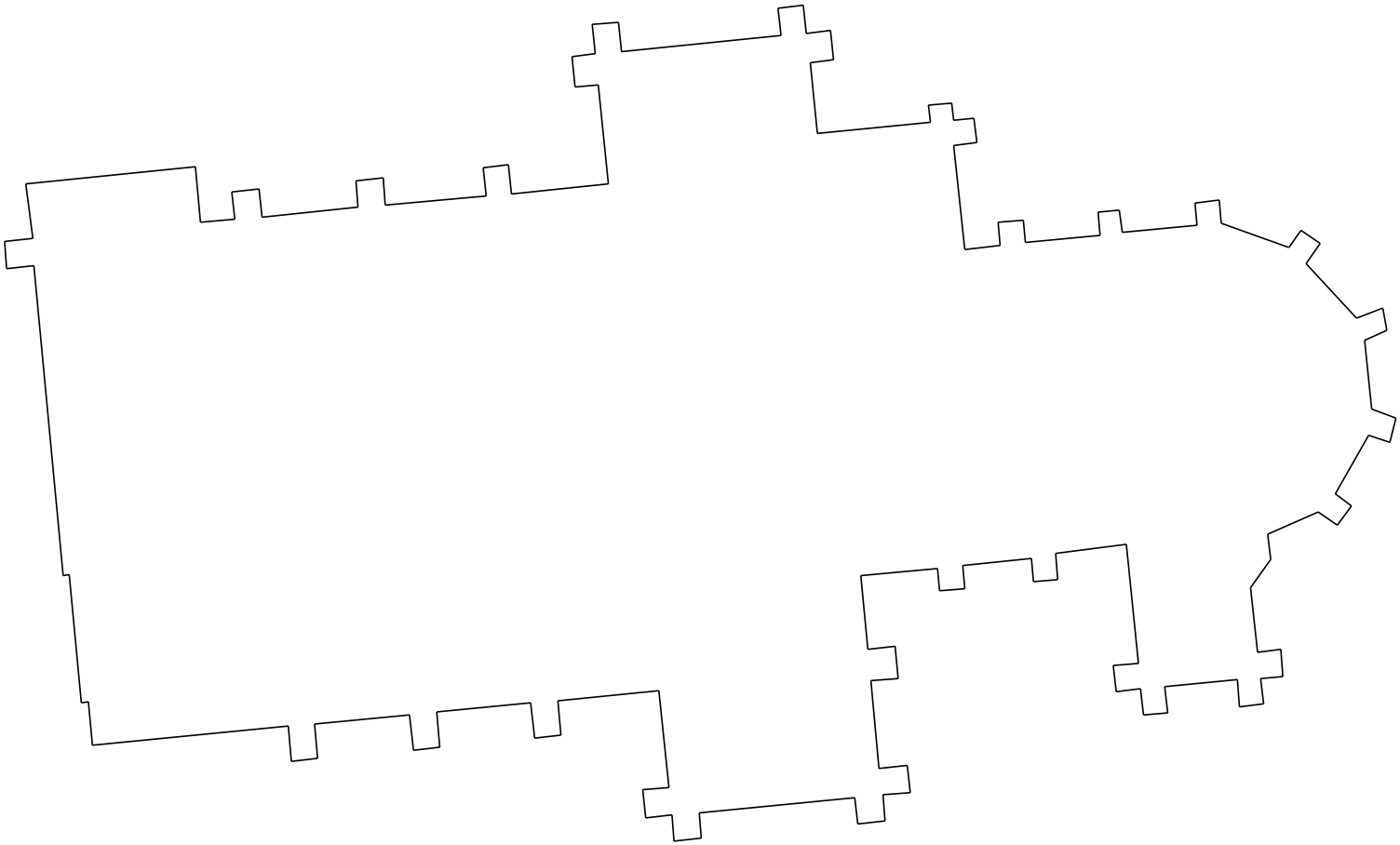


Figure: Input

Results – One building

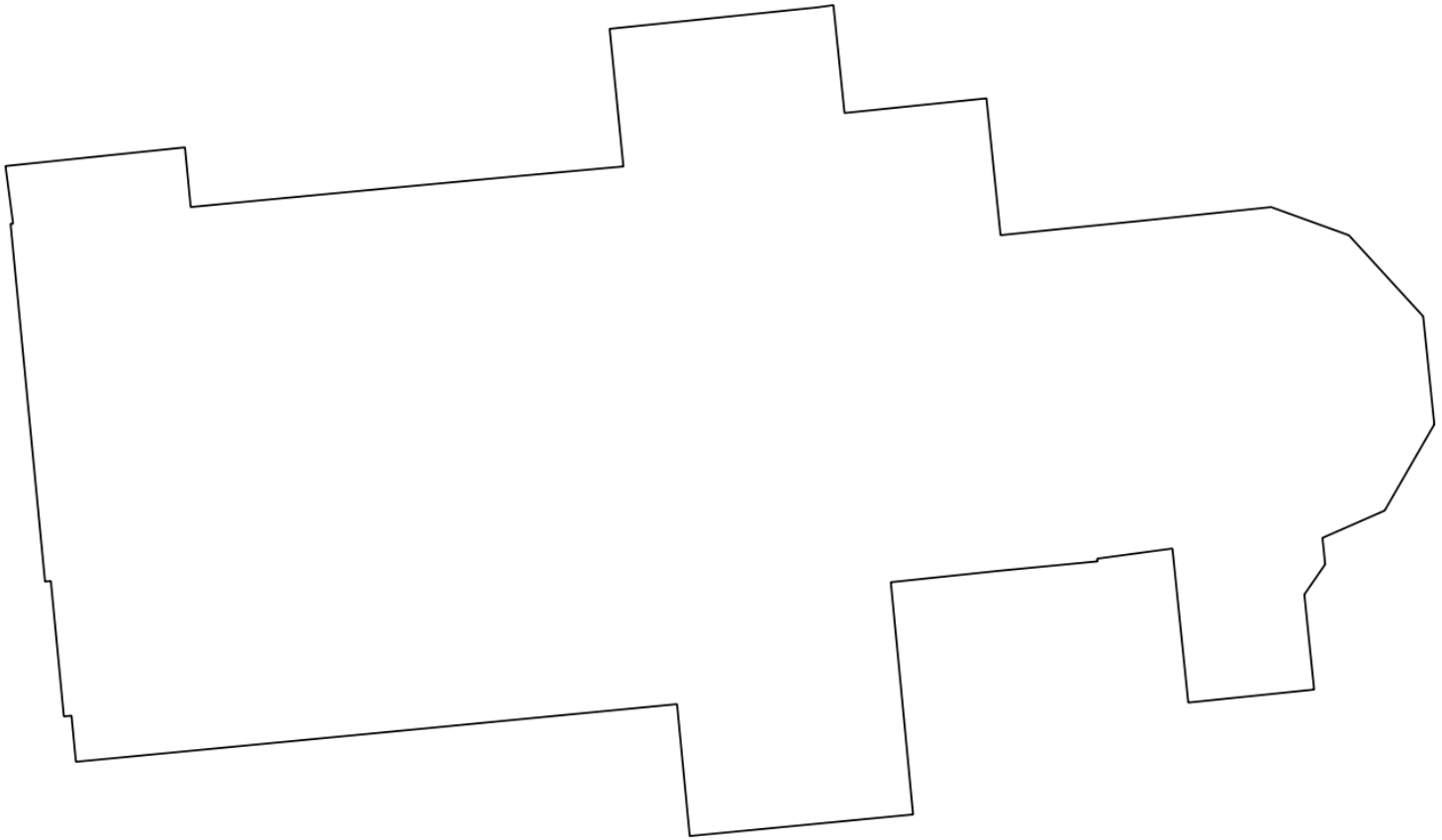


Figure: In–Out Result

Results – One building

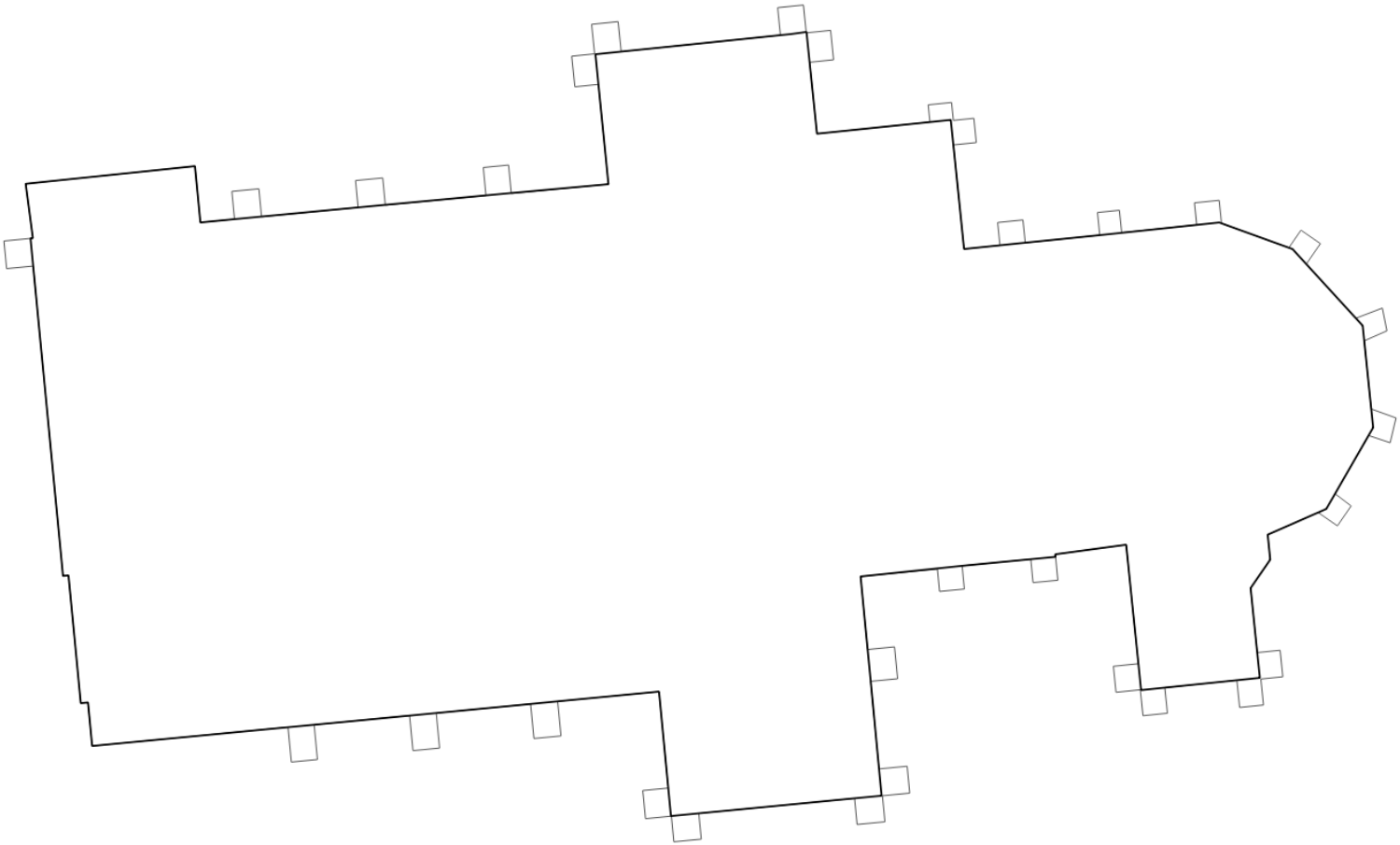


Figure: Result – Comparison with input

Results – One building

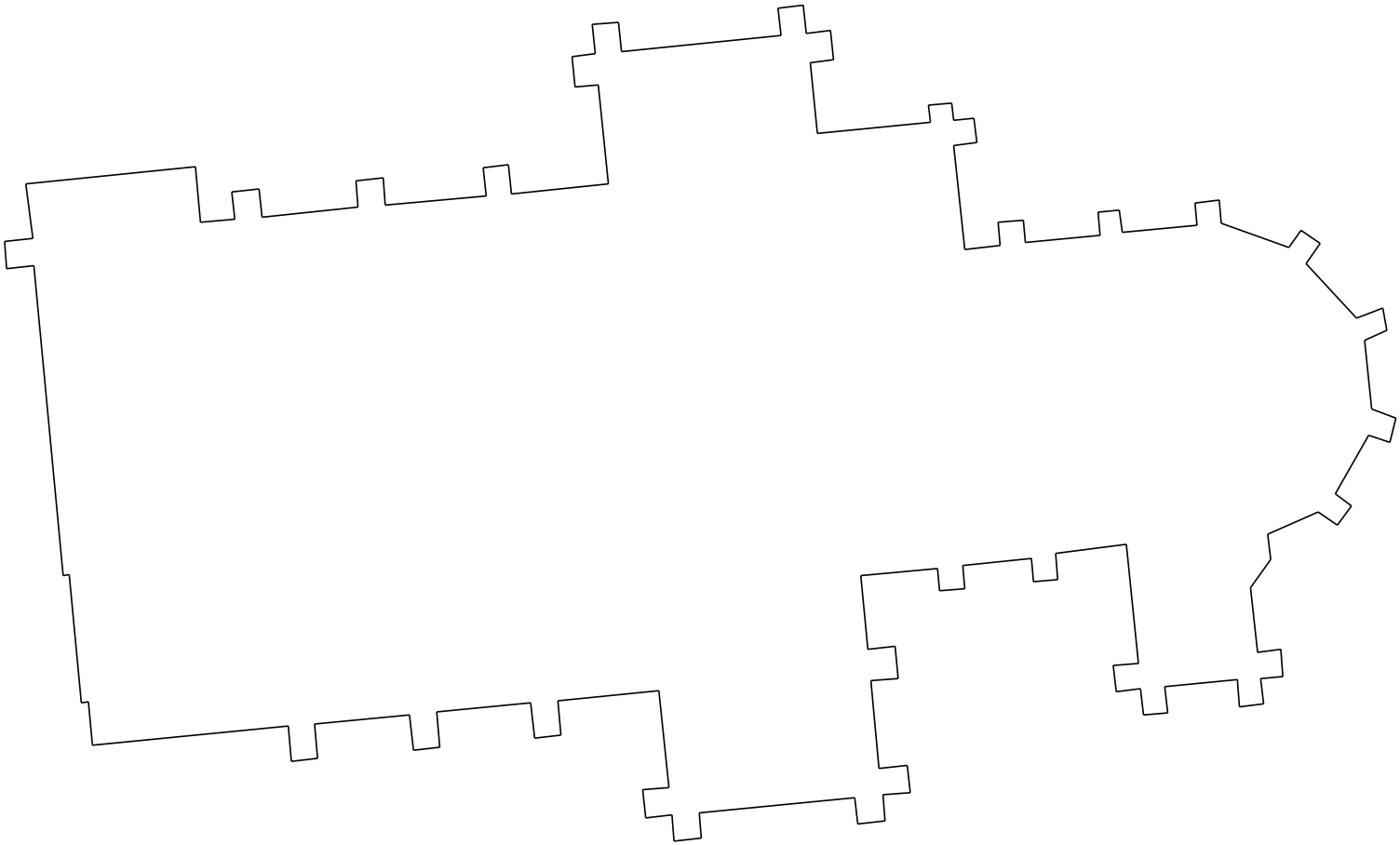


Figure: Input

Results – One building

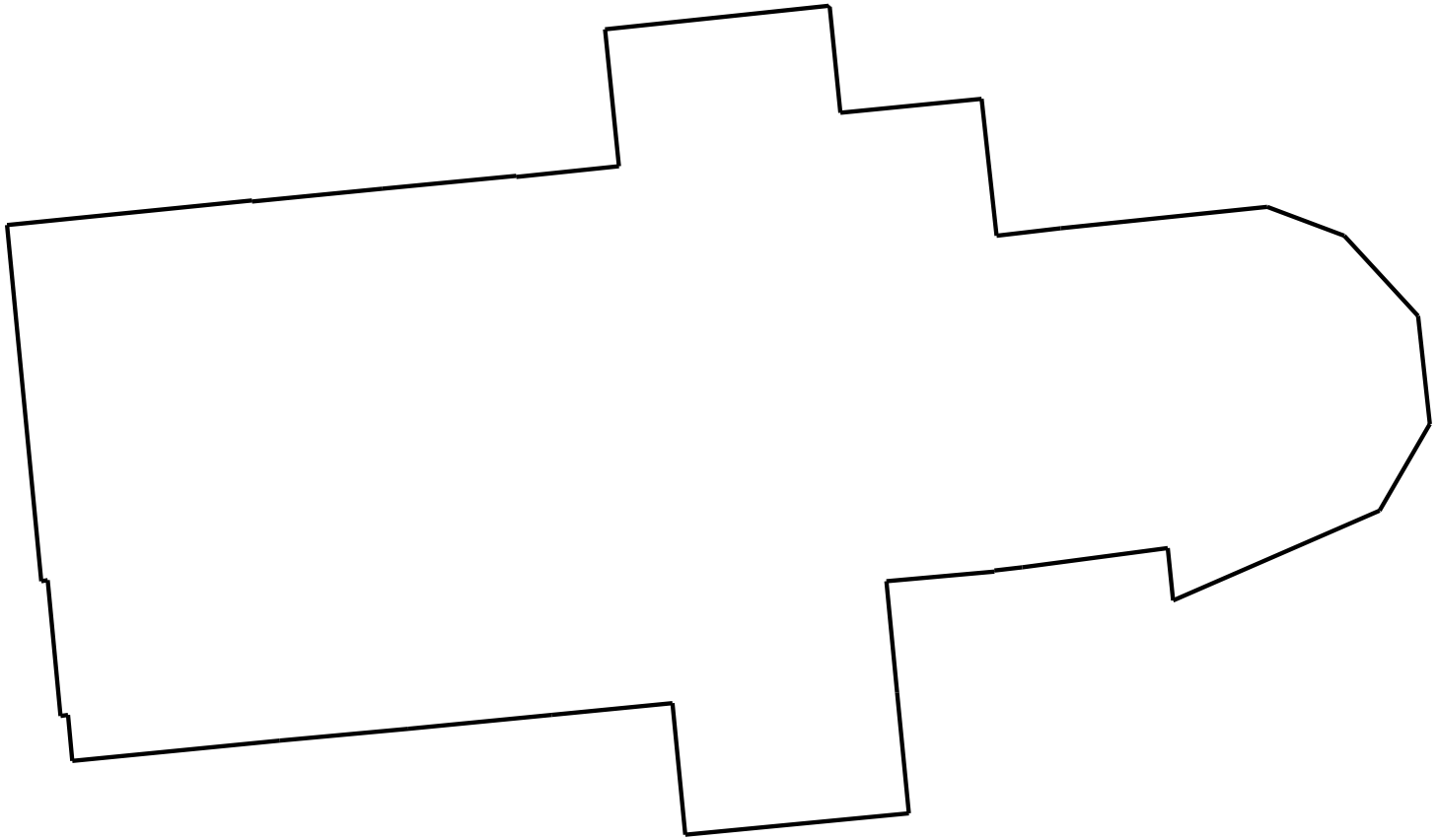


Figure: In–Out result, larger ϵ

Results – One building

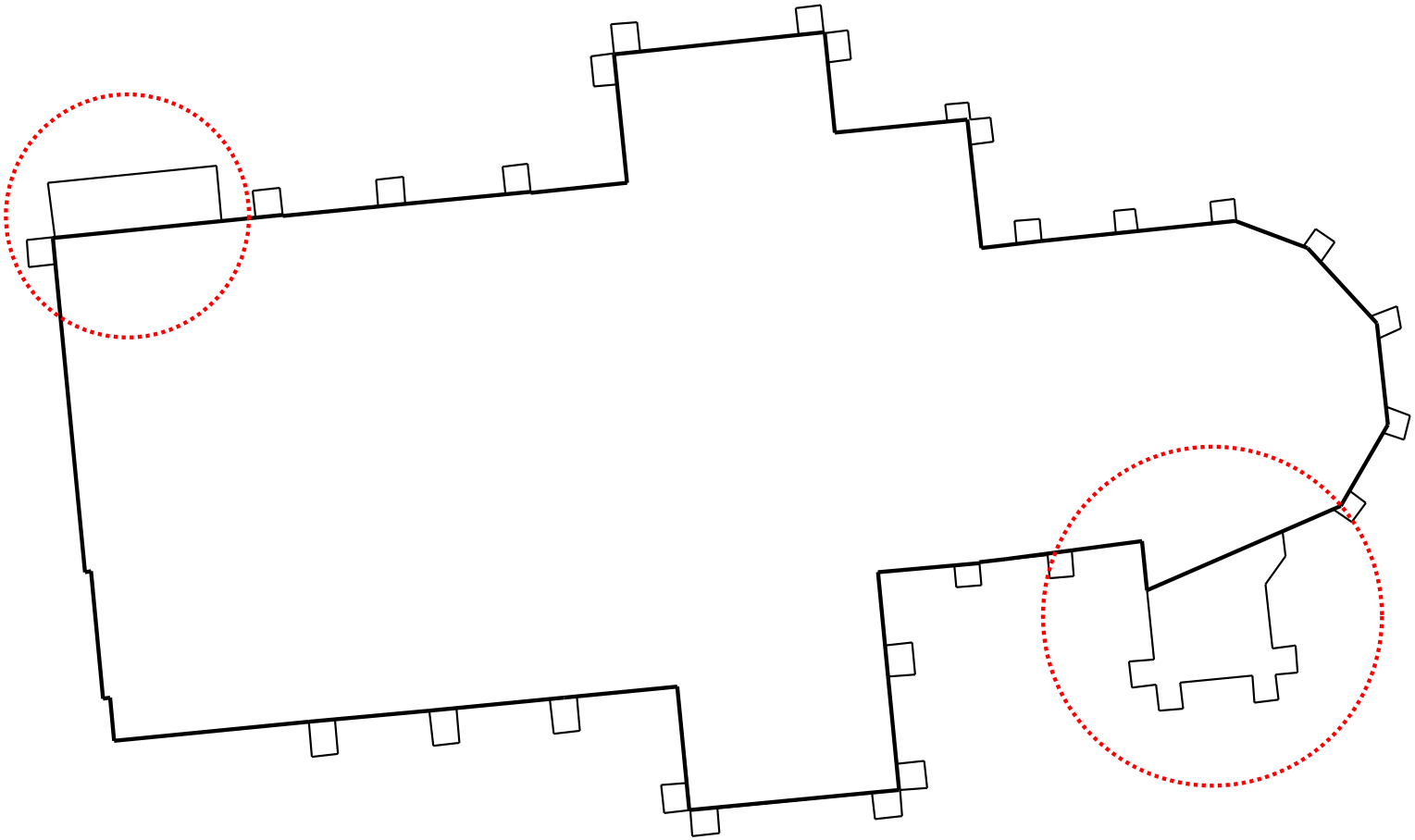


Figure: In–Out result, larger ϵ (compare with input)

Results – One building

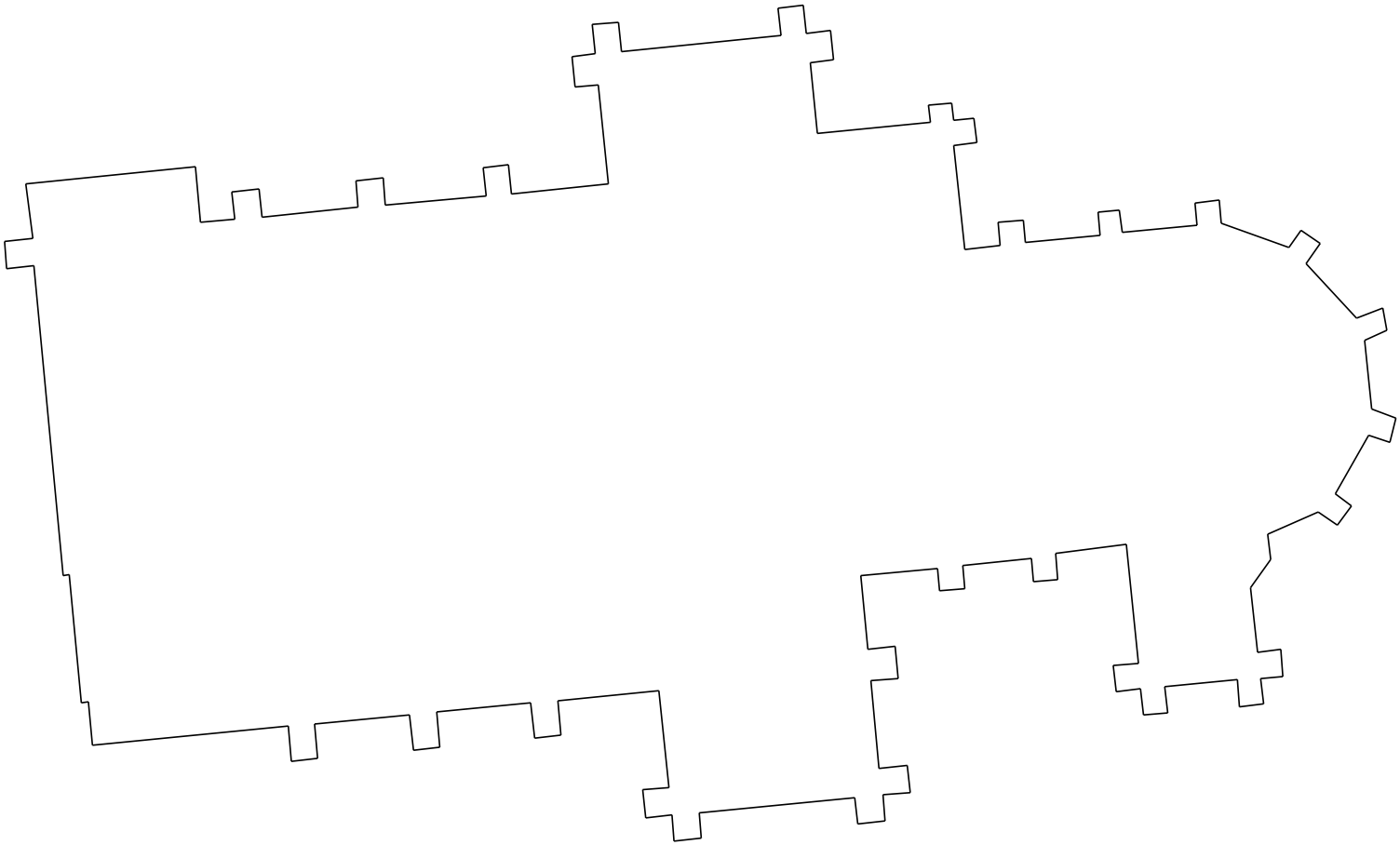


Figure: Input

Results – One building

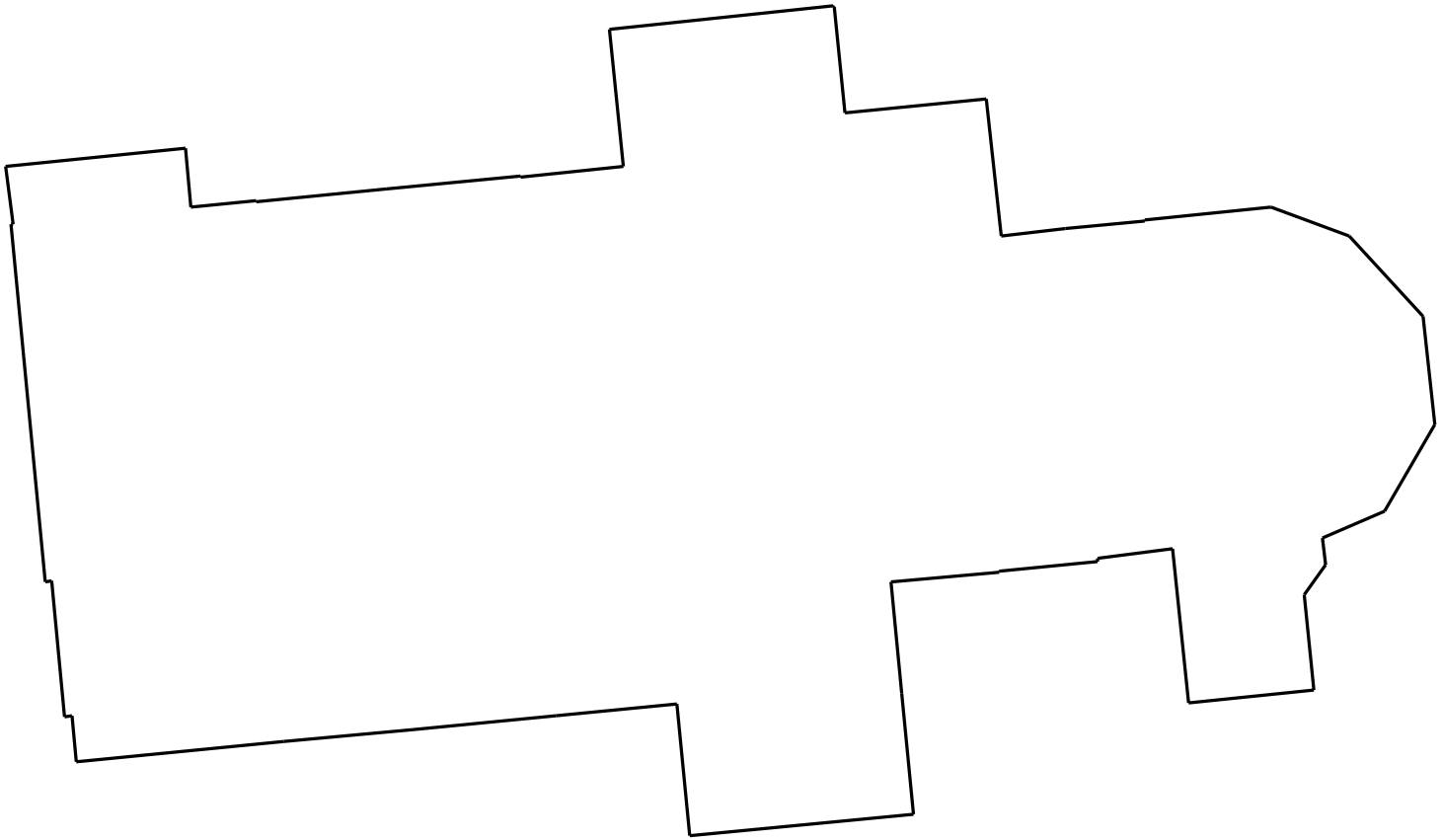


Figure: In–Out–In, result

Results – One building

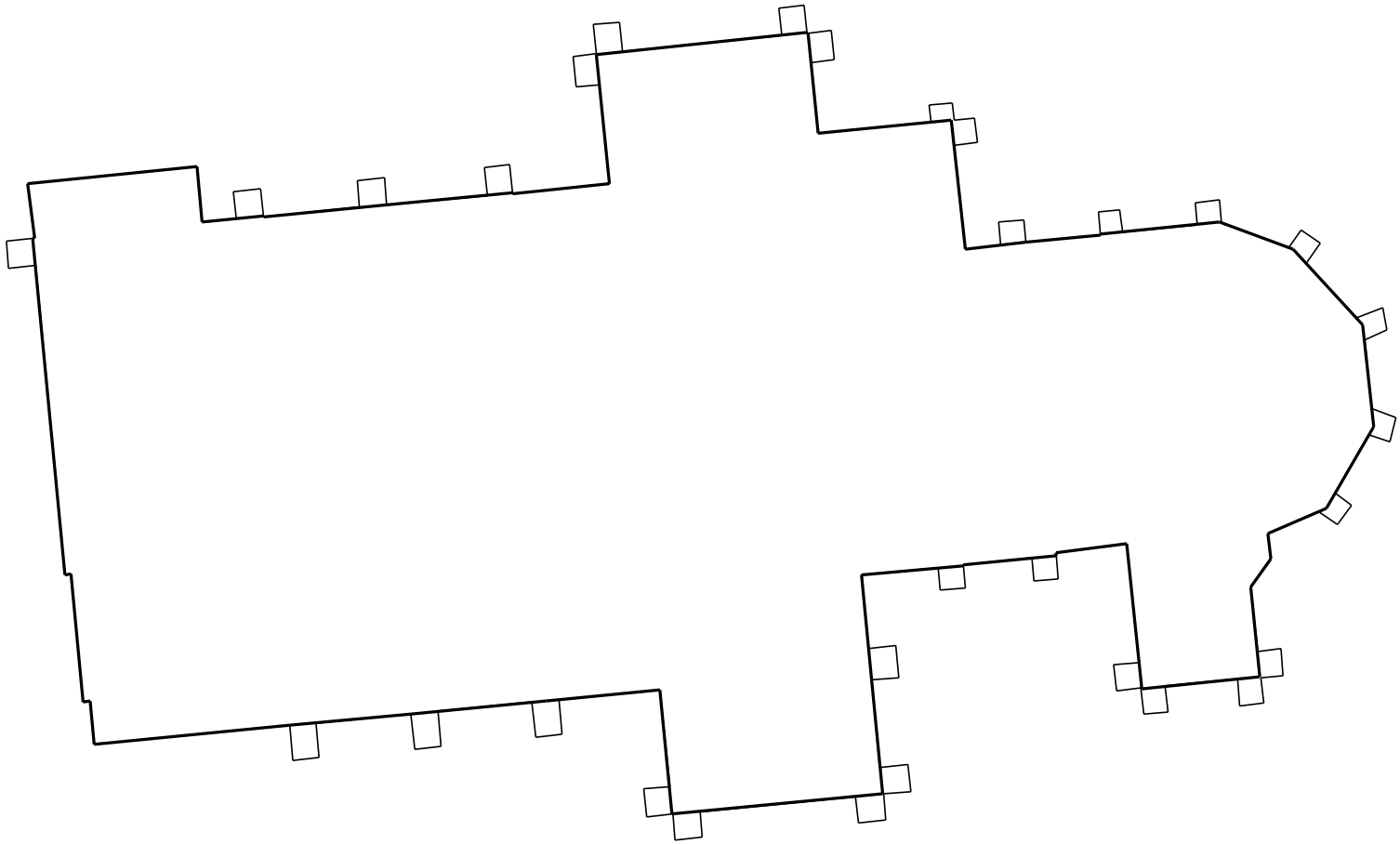


Figure: In–Out–In, result; (compare with input)

Results – One building

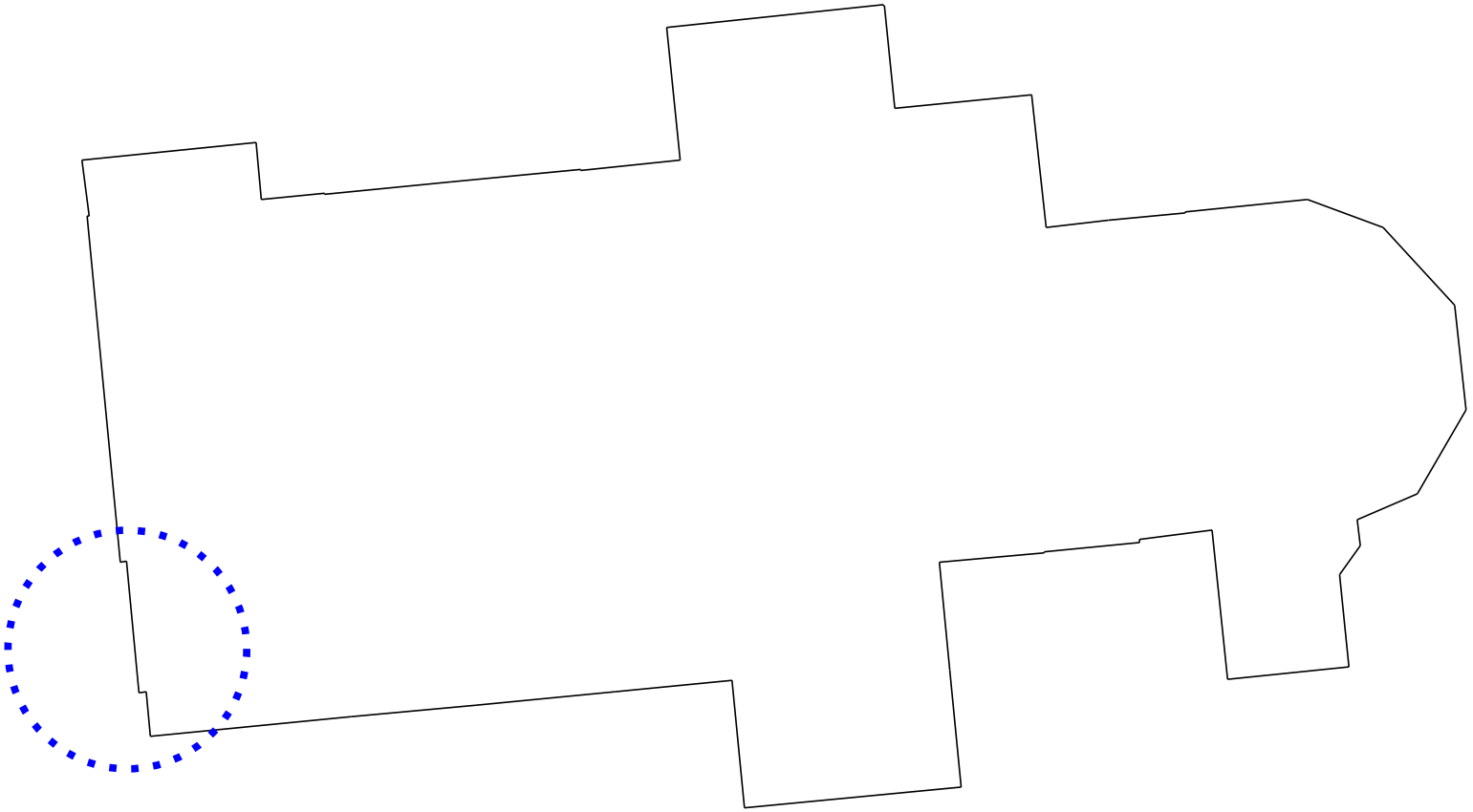


Figure: In–Out–In, result; Small edges still present

Results – One building

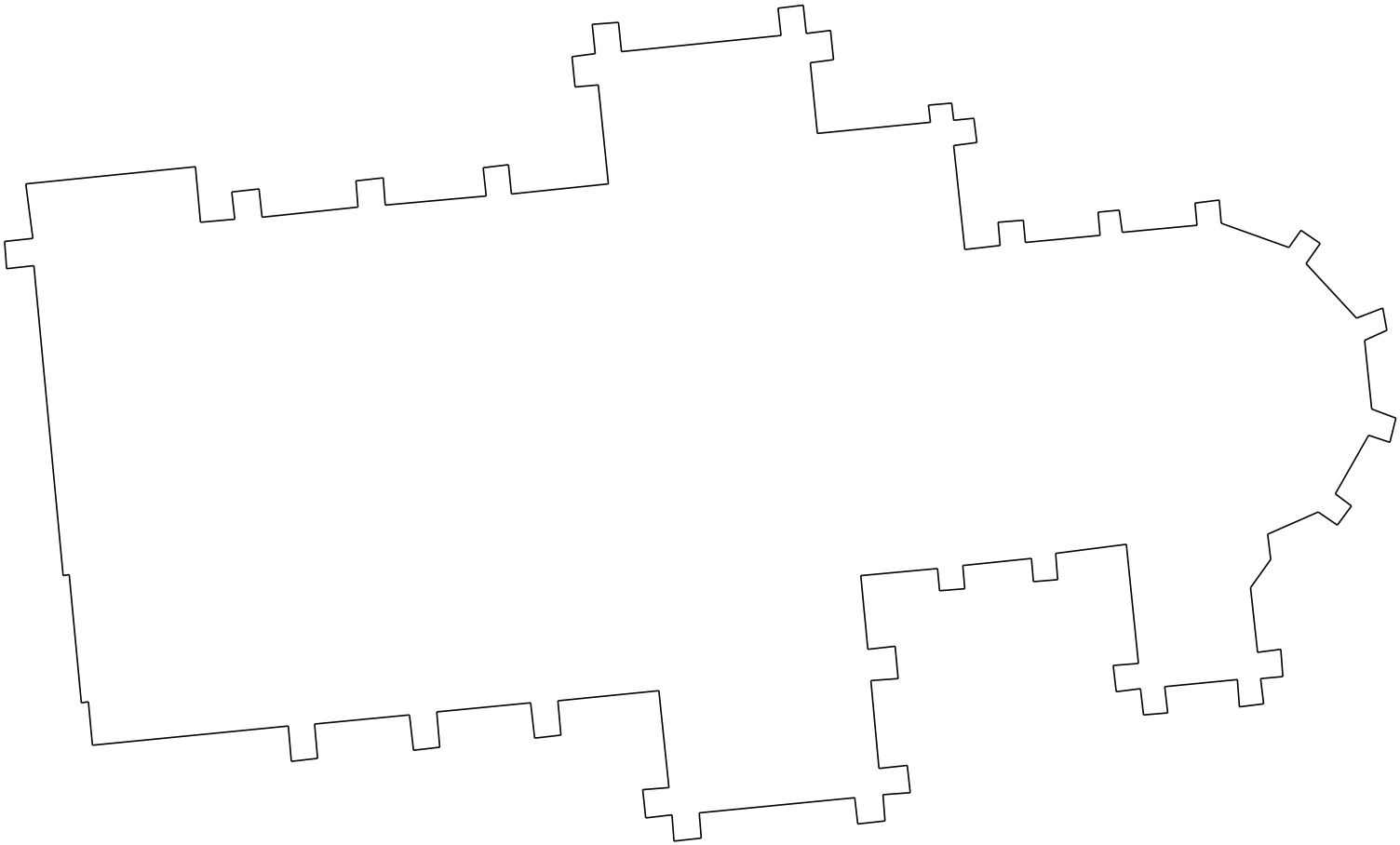


Figure: Input

Results – One building

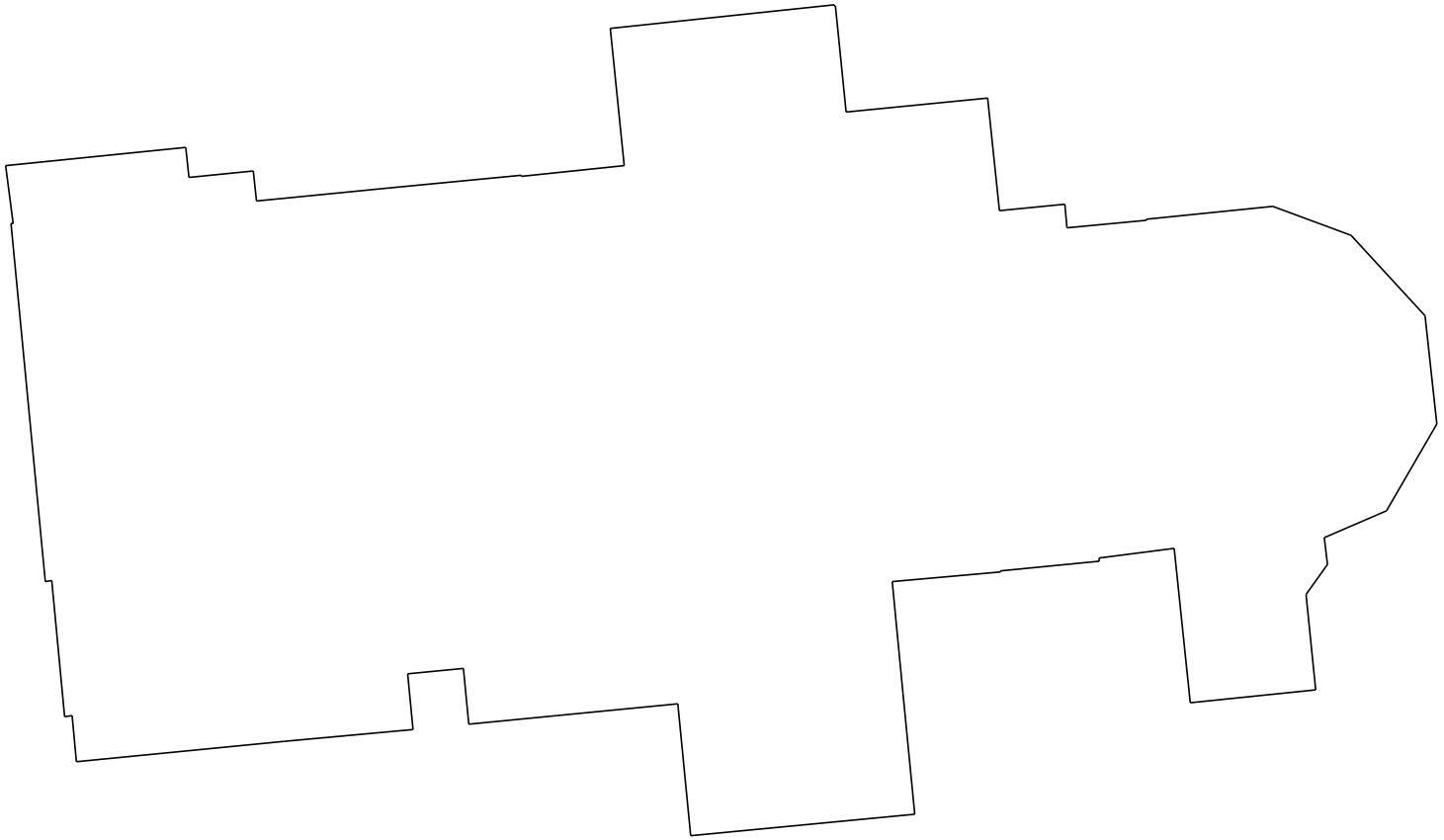


Figure: Out–In–Out

Results – One building

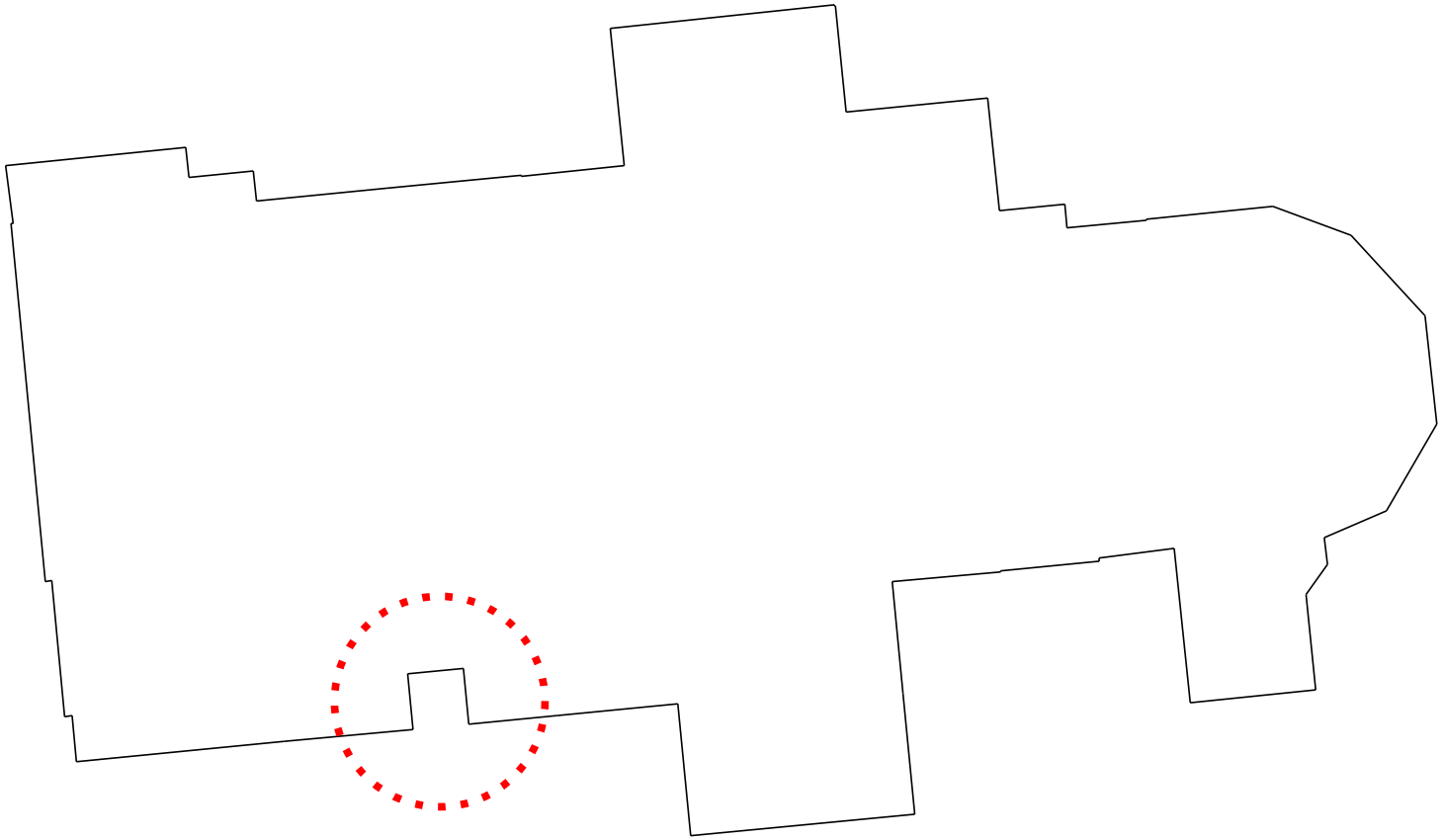


Figure: Out–In–Out; New created dent

Results – Multiple buildings

Figure: Input

Results – Multiple buildings

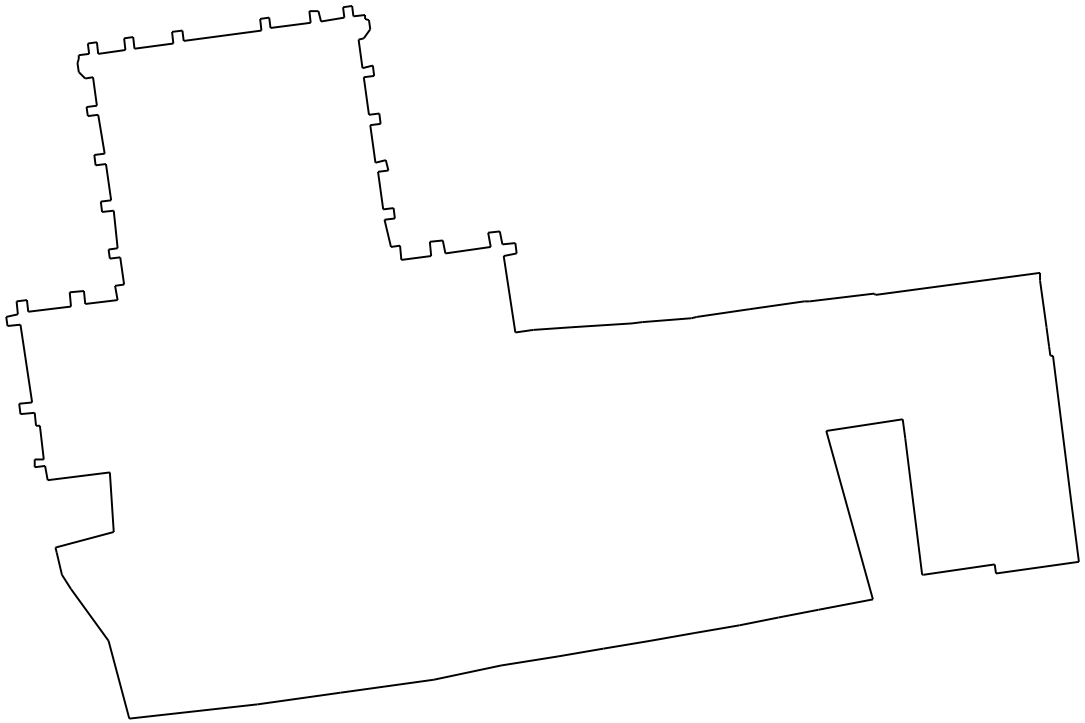


Figure: Out–In

Results – Multiple buildings

Figure: Input

Results – Multiple buildings

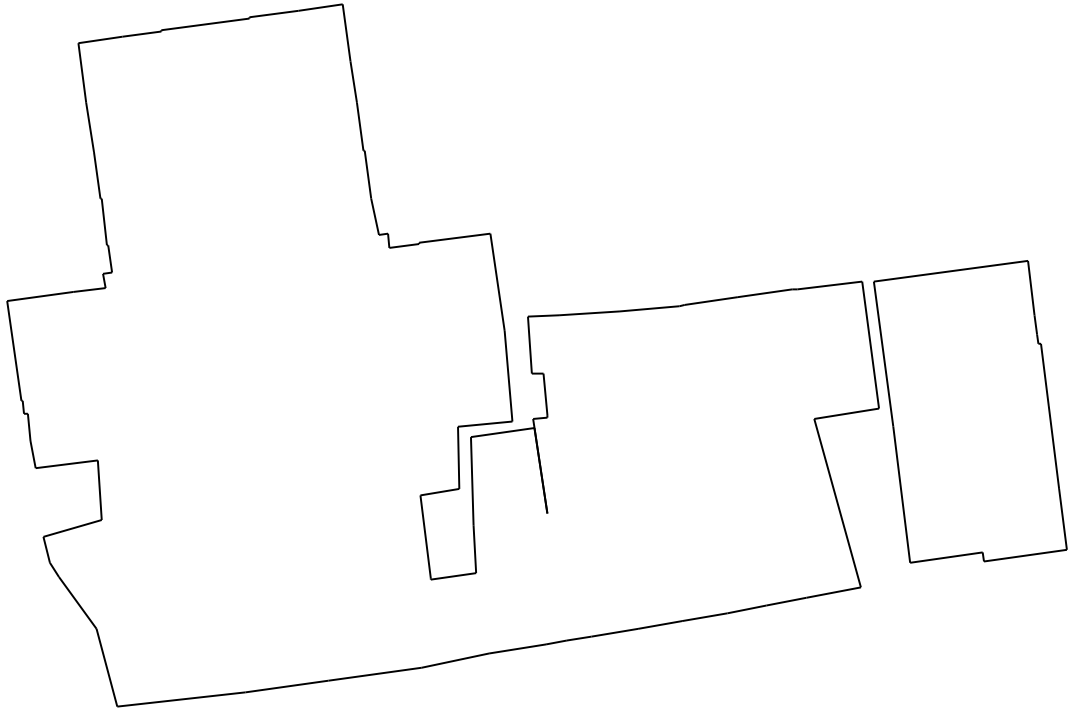


Figure: In–Out

Results – Multiple buildings

Figure: Input

Results – Multiple buildings

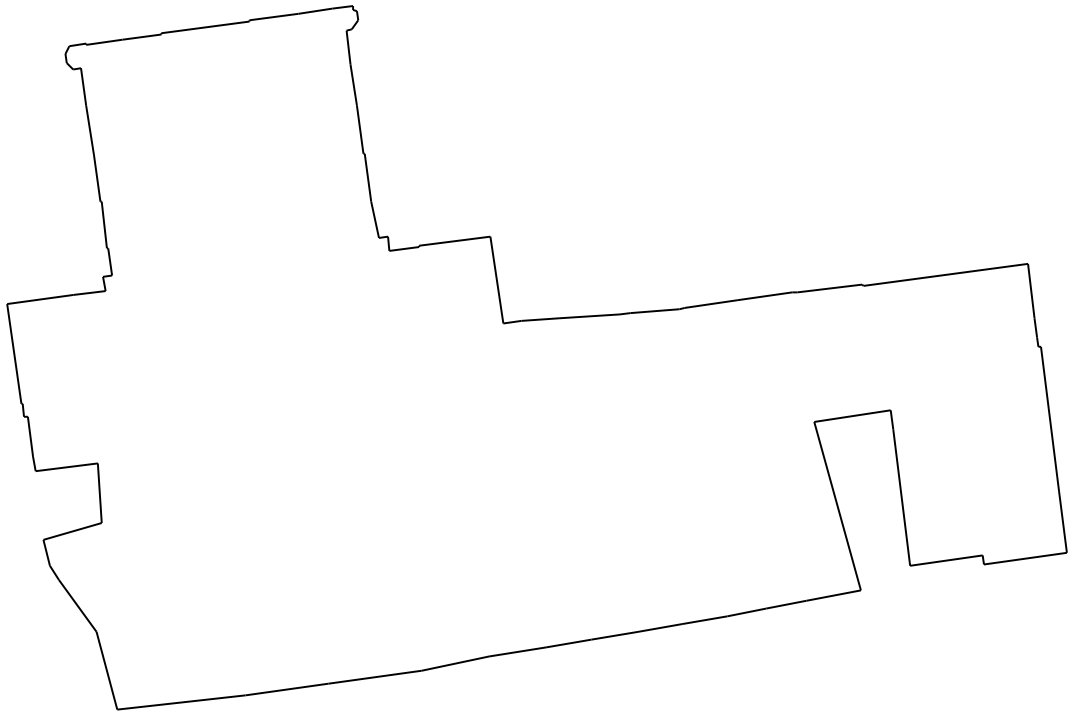


Figure: Out–In–Out

Results – Multiple buildings

Figure: Input

Results – Multiple buildings

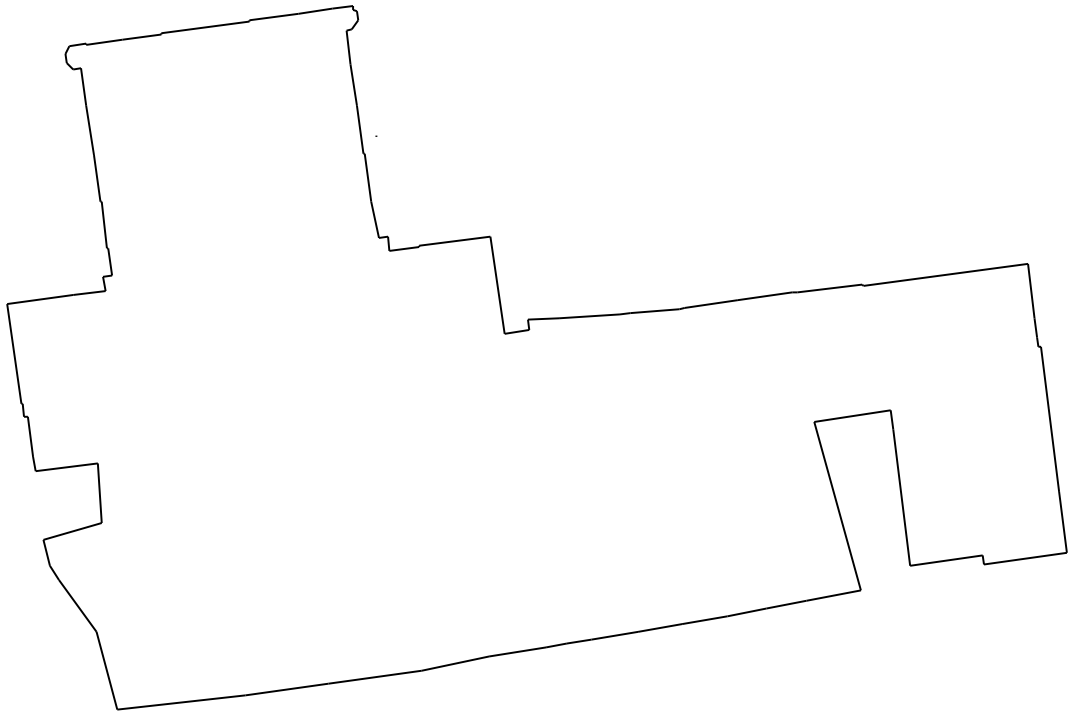


Figure: In–Out–In

Results

- Straightforward algorithm
- Simplifies individual building outlines
- Amalgamates multiple buildings
- Still needs post-processing for small segments

Discussion

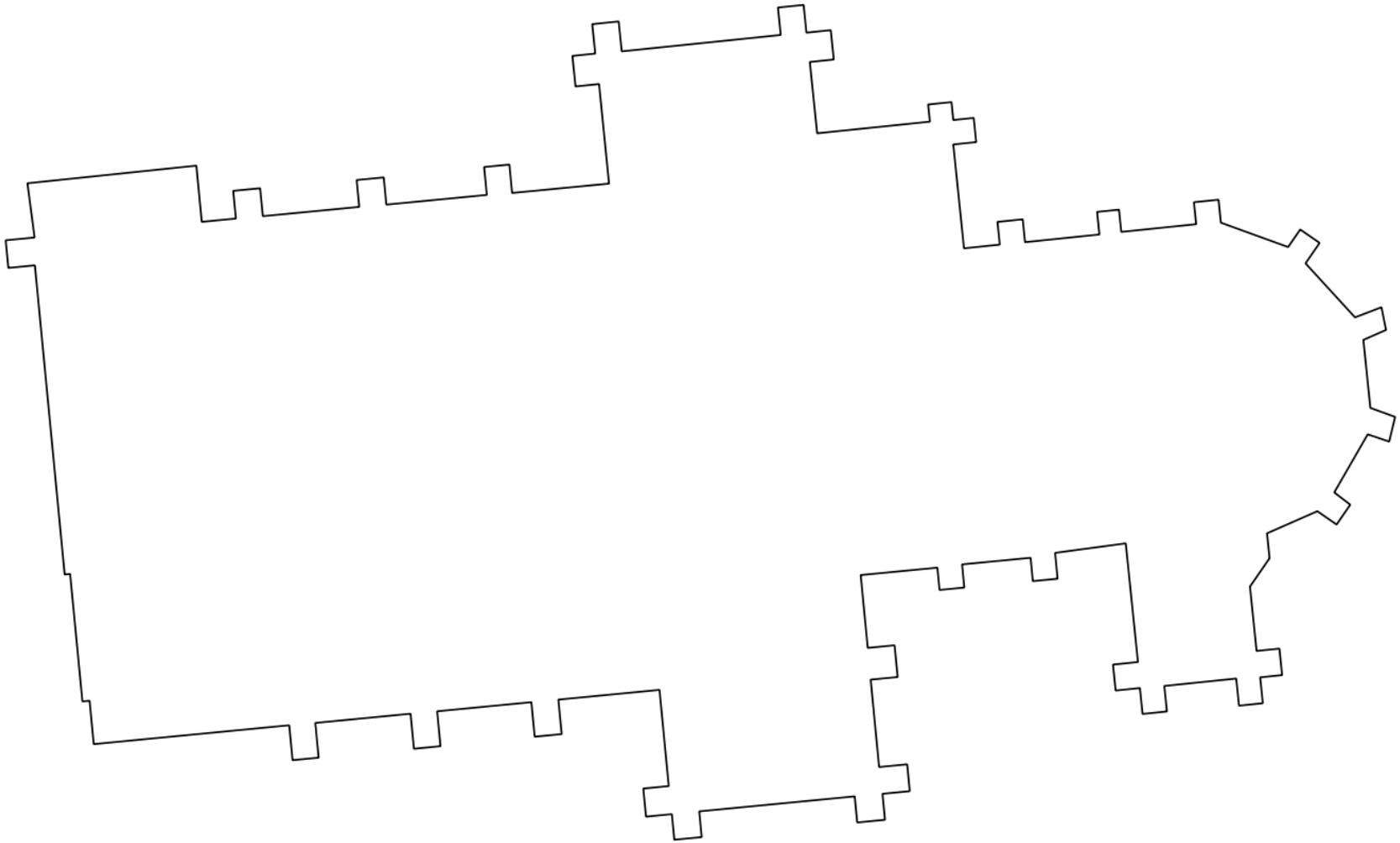
- Improve our implementation of straight skeleton, GrassFire[1]
- Difficult: determine correct ϵ
- No rotation needed for input (compare with Minkowski sum)

[1] https://bitbucket.org/bmmeijers/grassfire

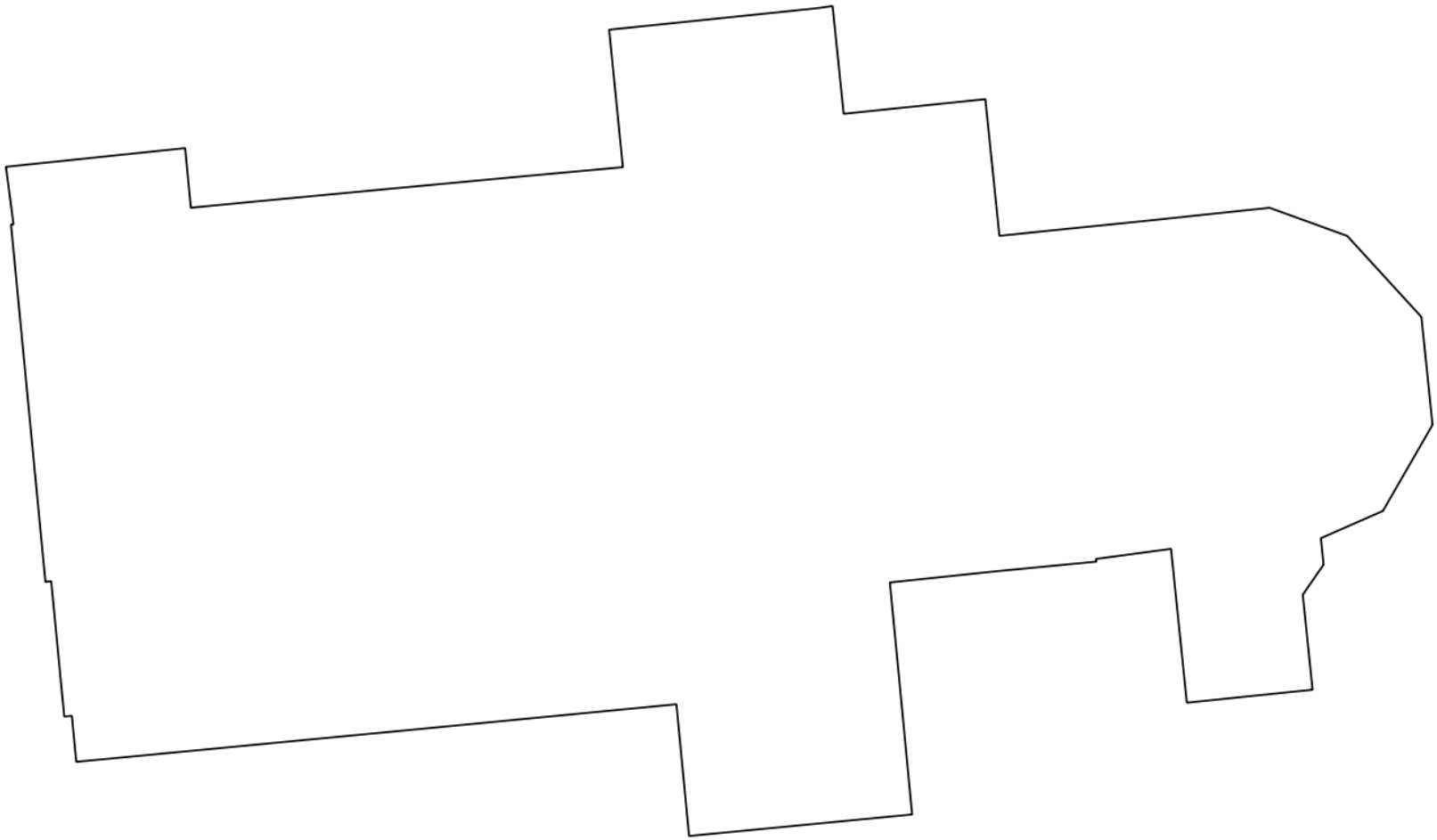
Future work

- Smooth transitions between the input and the output: interpolate what is in between (Barequet et al., 2004; Yakersberg, 2004)

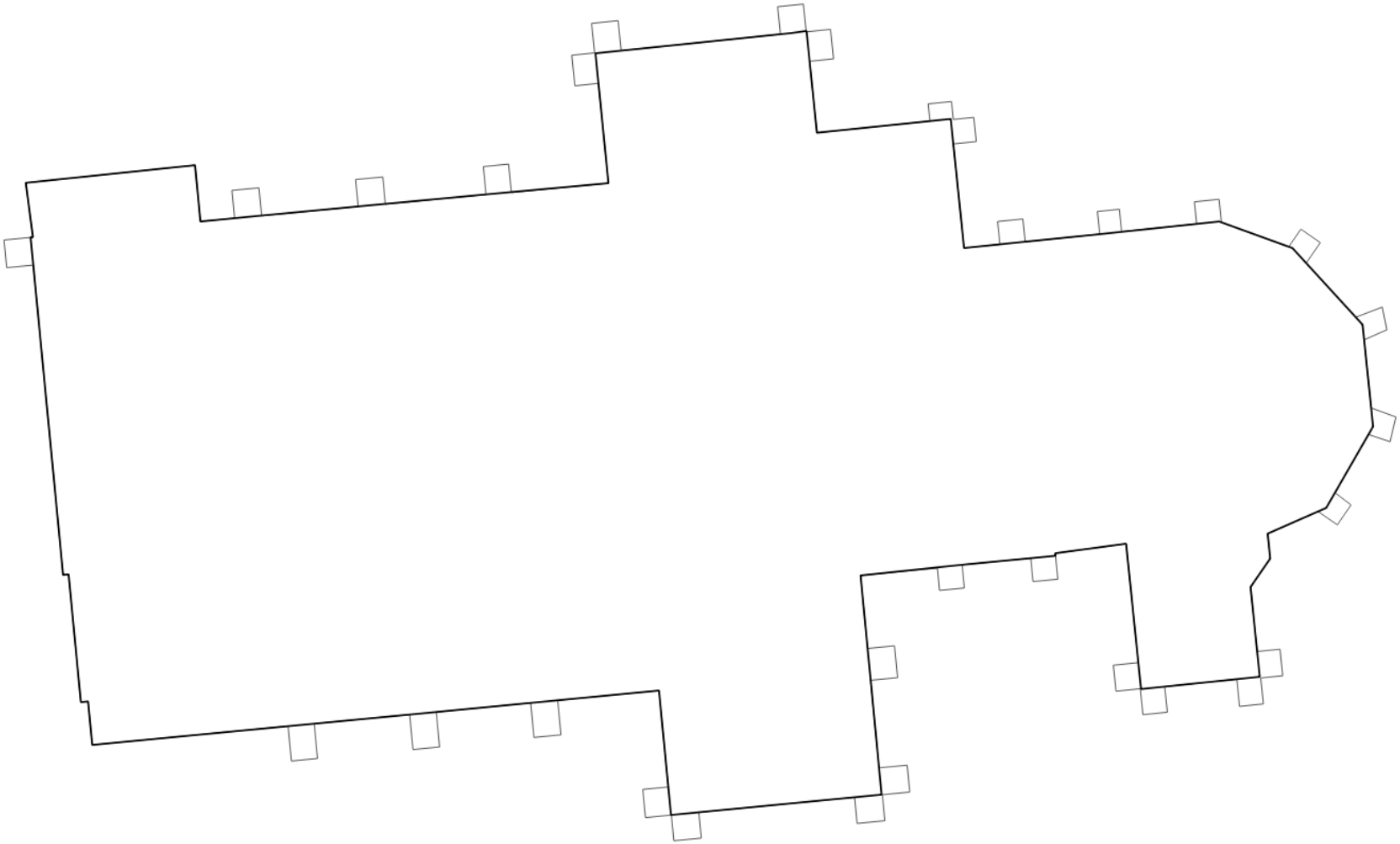
Smooth transitions



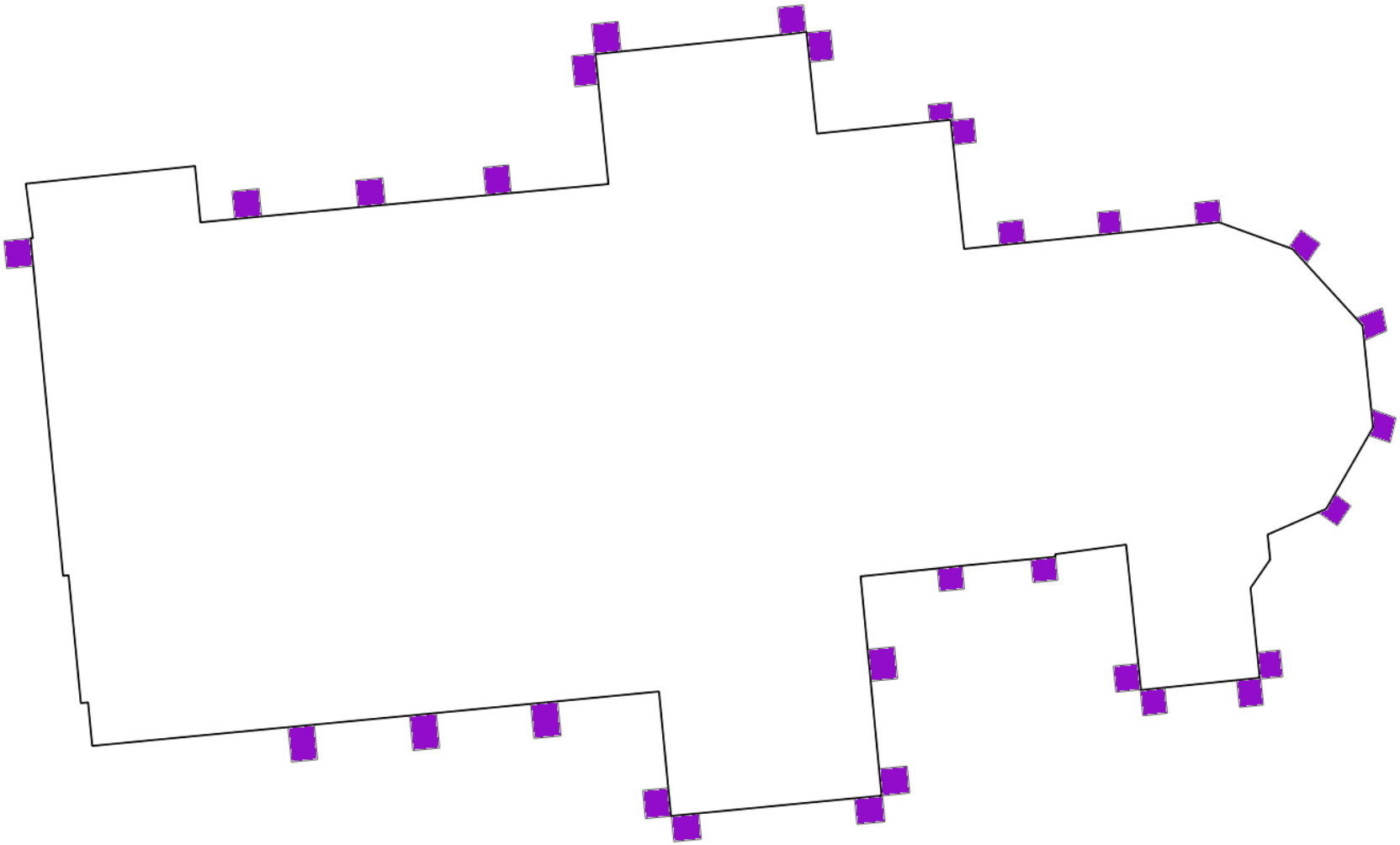
Smooth transitions



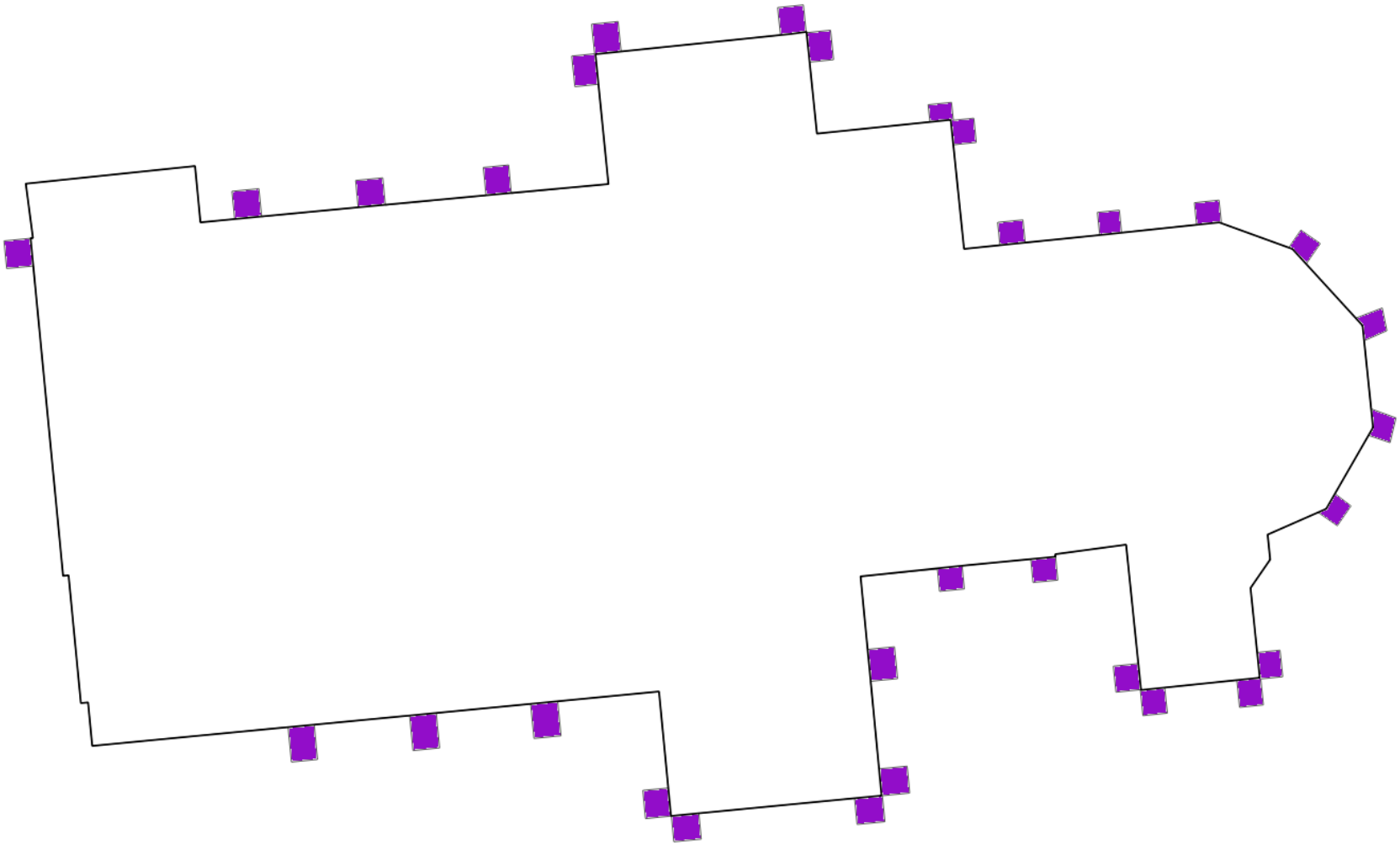
Smooth transitions



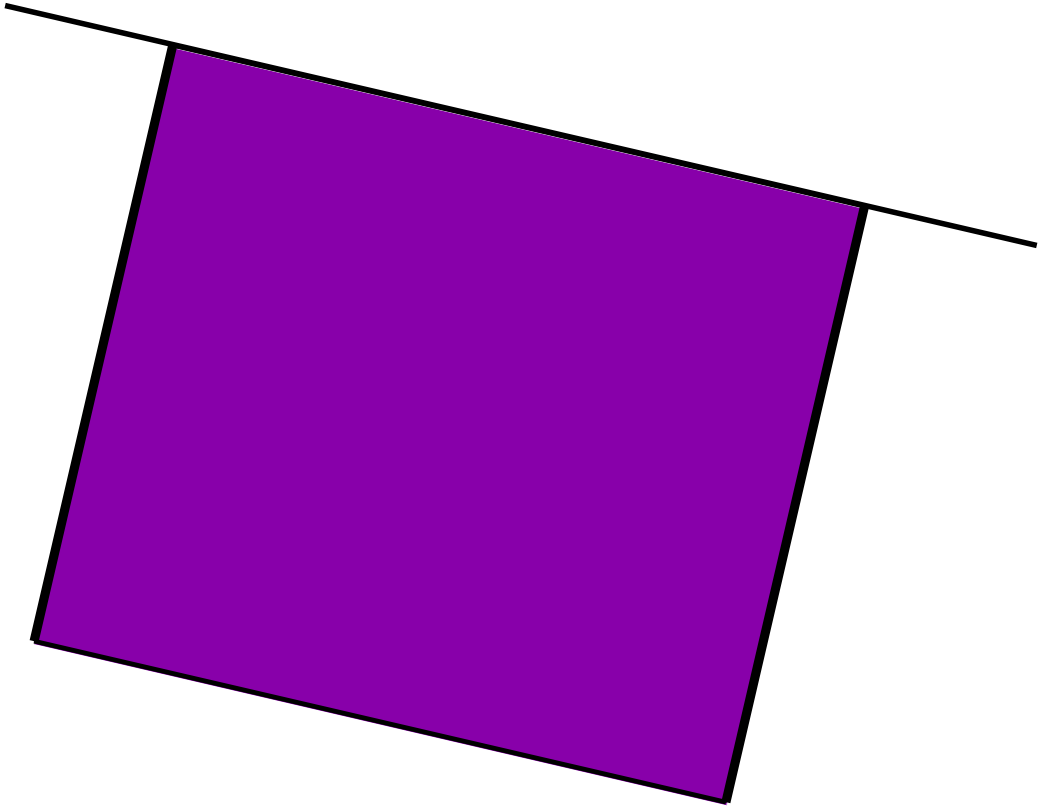
Smooth transitions



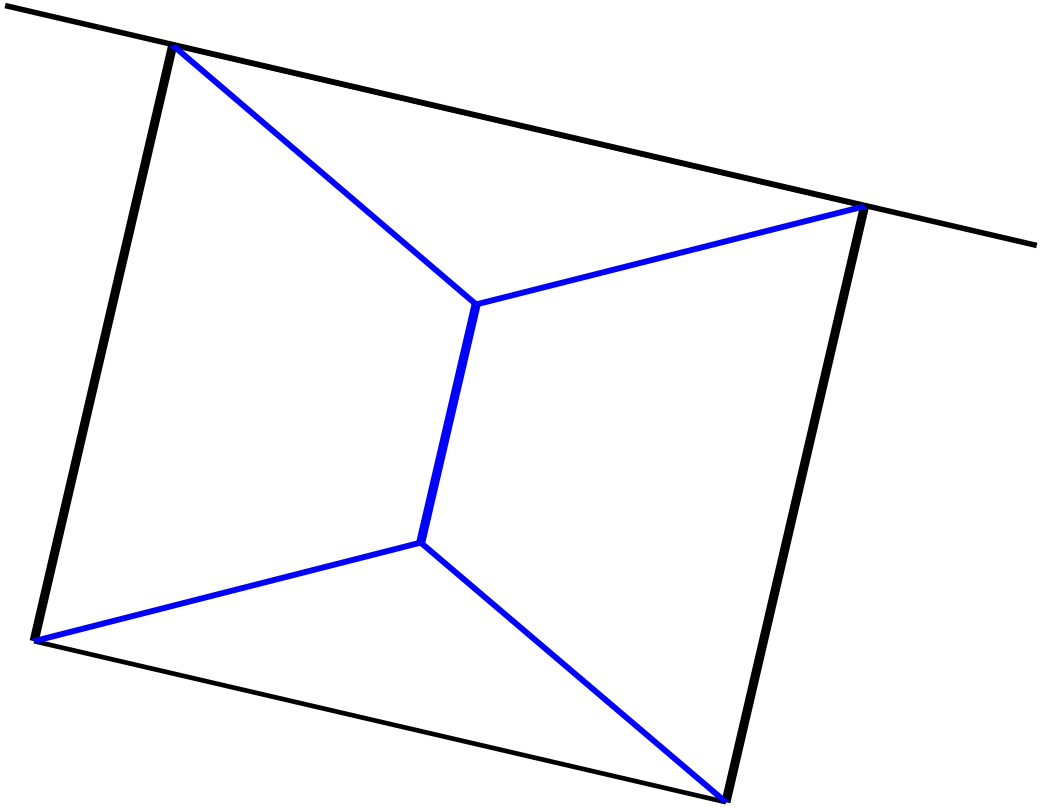
Smooth transitions



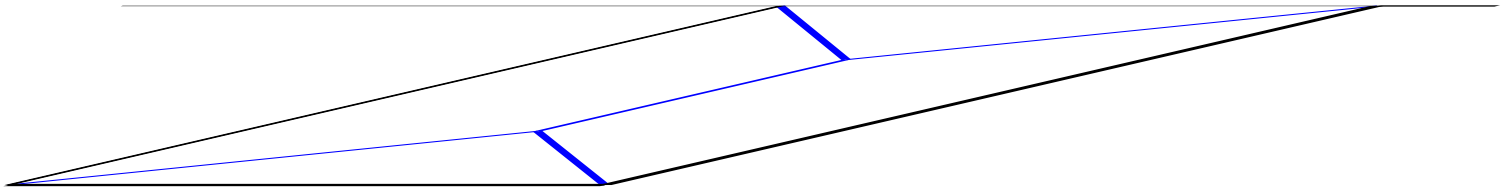
Smooth transitions



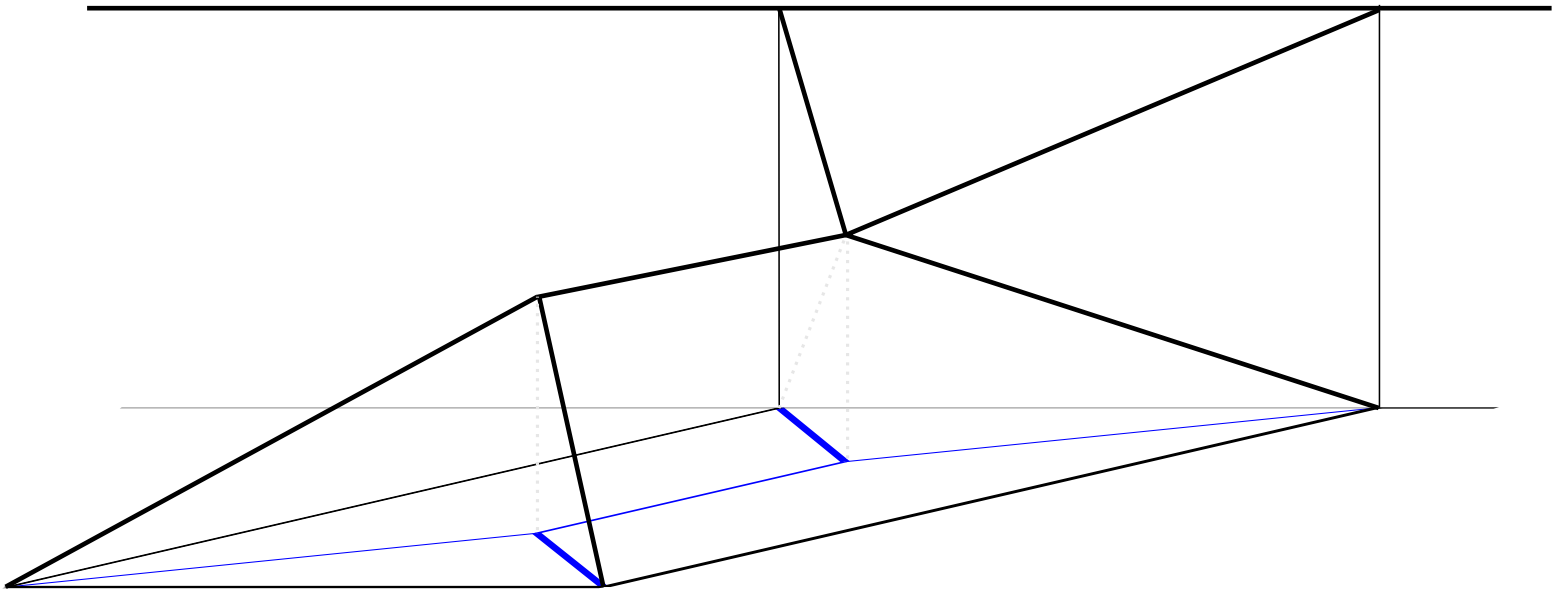
Smooth transitions



Smooth transitions



Smooth transitions



Thank you for your attention

- Questions?
- Delft University of Technology
  Faculty of Architecture and the Built Environment
  OTB Research
  GIS Technology
- dr.ir. Martijn Meijers
  b.m.meijers@tudelft.nl
  http://www.gdmc.nl/martijn/
  tel. (+31) 15 27 856 42

References

Aichholzer, O. and Aurenhammer, F. (1996). Straight skeletons for general polygonal figures in the plane. In Cai, J.-Y. and Wong, C. K., editors, *Computing and Combinatorics*, volume 1090 of *Lecture Notes in Computer Science*, pages 117–126. Springer Science + Business Media.

Barequet, G., Goodrich, M. T., Levi-Steiner, A., and Steiner, D. (2004). Contour interpolation by straight skeletons. *Graphical Models*, 66(4):245–260.

Damen, J., van Kreveld, M., and Spaan, B. (2008). High quality building generalization by extending the morphological operators. In *Proceedings of the ICA Workshop on Generalization*, Montpellier, France.

Haunert, J.-H. and Sester, M. (2007). Area collapse and road centerlines based on straight skeletons. *GeoInformatica*, 12(2):169–191.

Palfrader, P. (2013). Computing Straight Skeletons by Means of Kinetic Triangulations. Master's thesis, University of Salzburg, Austria.

Palfrader, P. and Held, M. (2015). Computing Mitered Offset Curves Based on Straight Skeletons. *Computer-Aided Design and Applications*, 12(4):414–424.

Yakersberg, E. (2004). Morphing between Geometric Shapes using Straight-Skeleton-Based Interpolation. Master's thesis, Technion – Israel Institute Of Technology, Haifa, Israel.

Minkowski sum (dilation–erosion) comparison

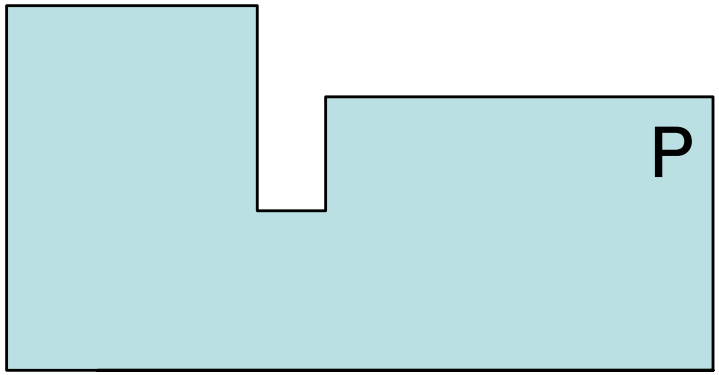


Figure: Input polygon *P*

Image credit: Jonathan Damen (Damen et al., 2008)

Minkowski sum (dilation–erosion) comparison

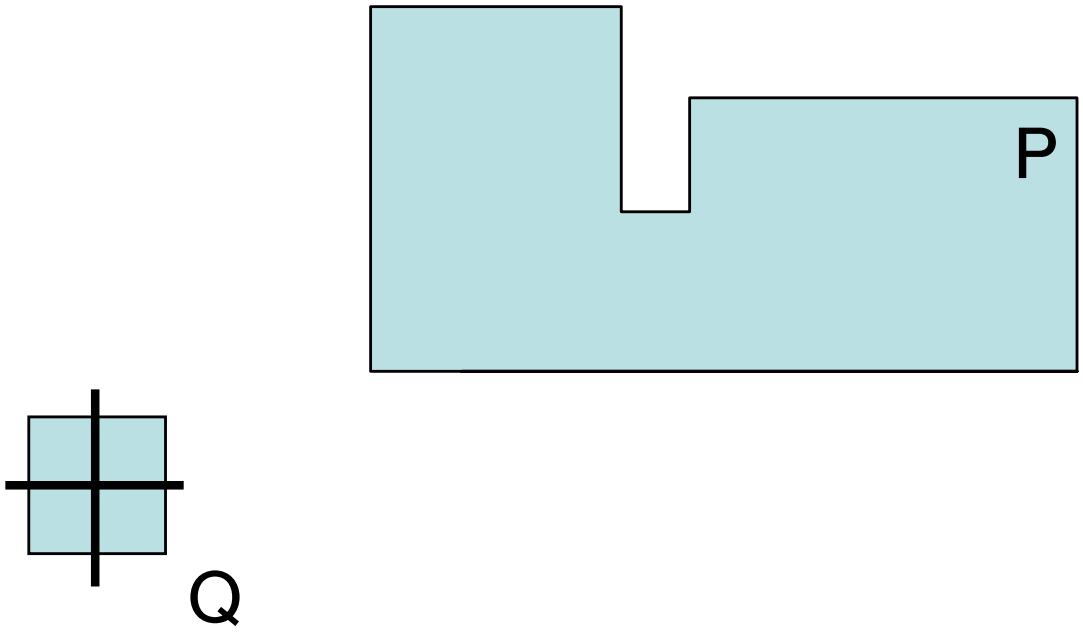


Figure: *Q* ‘traverses’ the boundary of *P*: adding *Q* to *P*

Minkowski sum (dilation–erosion) comparison

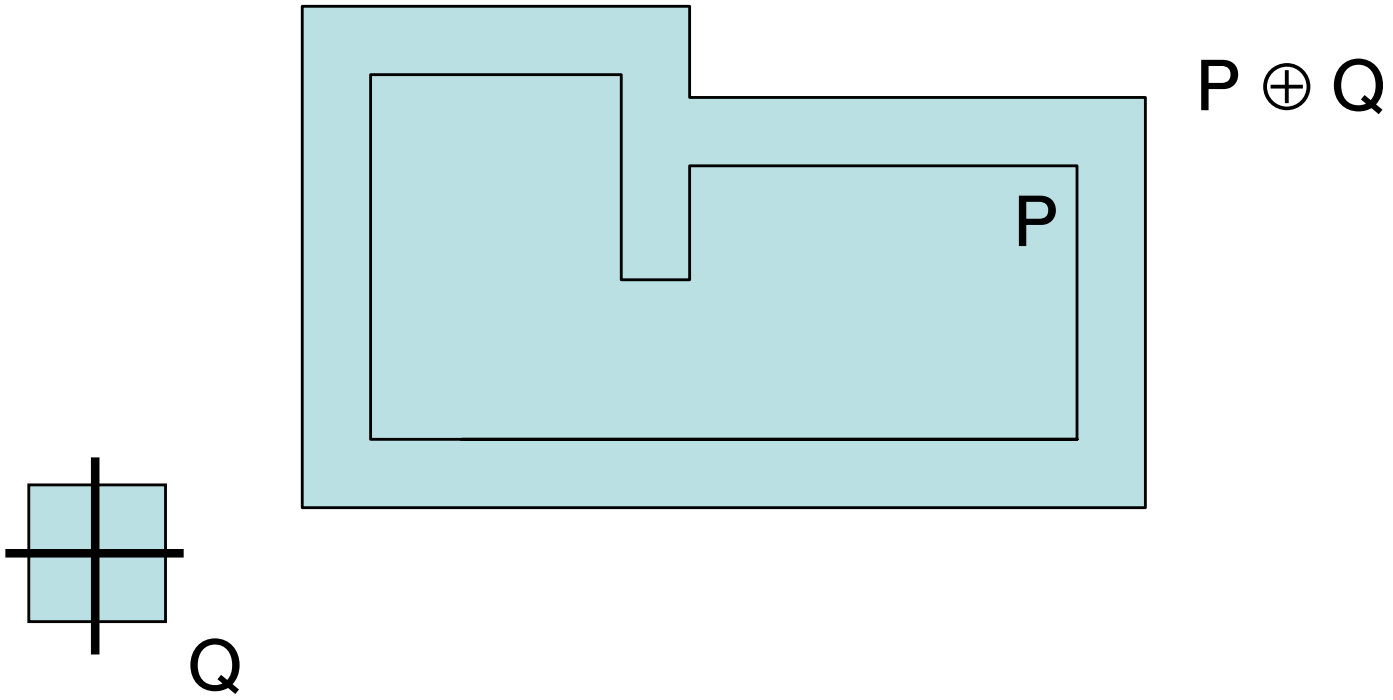


Figure: Dilation

Minkowski sum (dilation–erosion) comparison

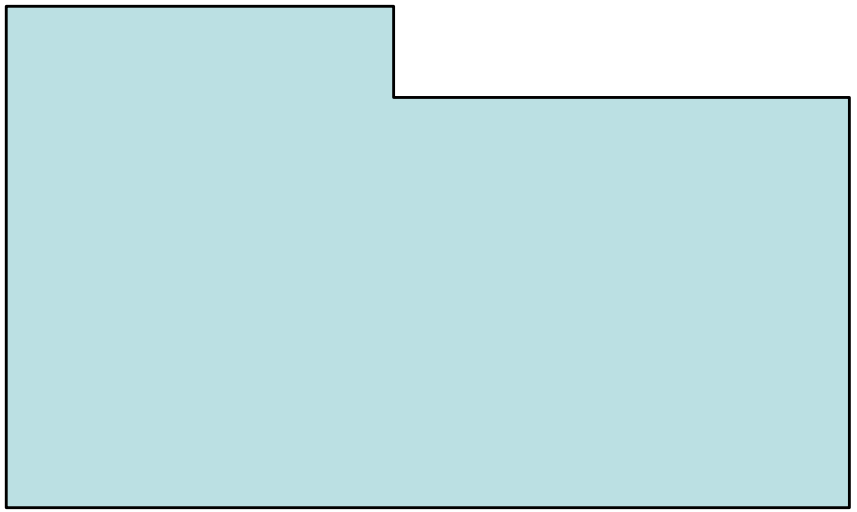


Figure: Dilation

Minkowski sum (dilation–erosion) comparison

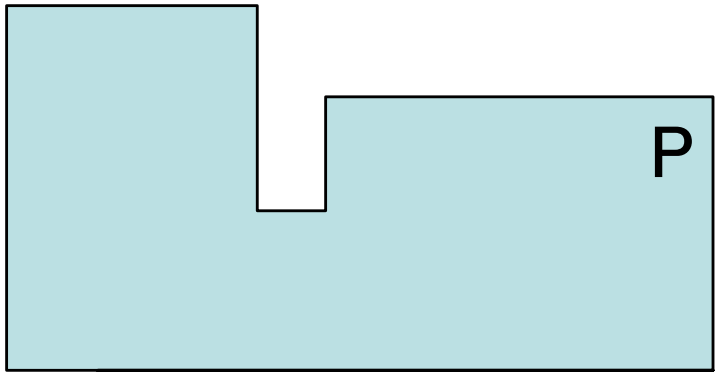


Figure: Input polygon *P*

Minkowski sum (dilation–erosion) comparison

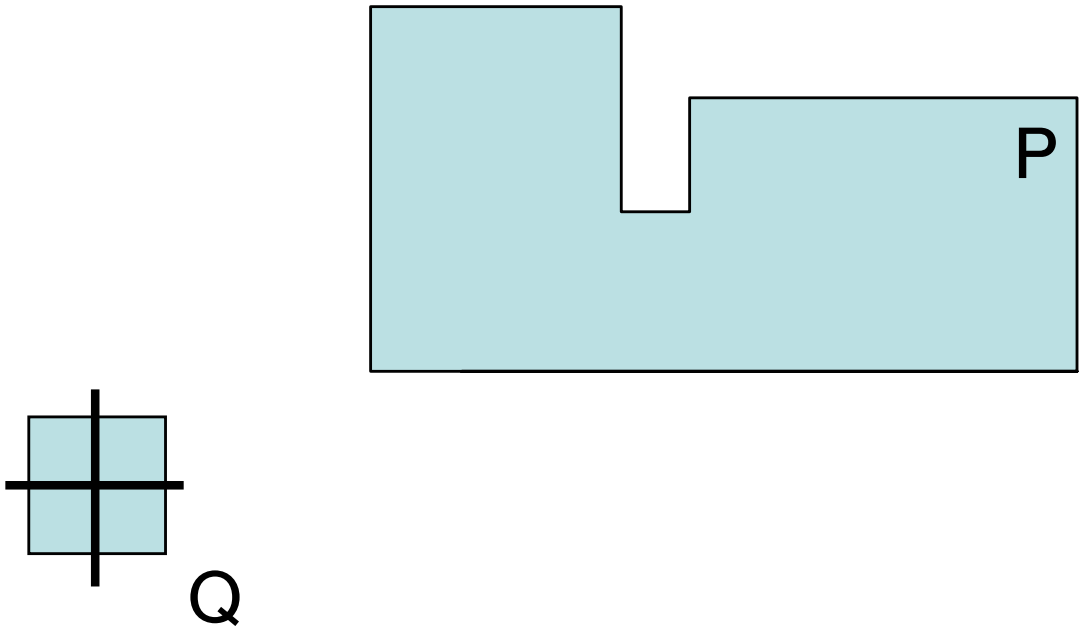


Figure: *Q* ‘traverses’ the boundary of *P*: subtracting *Q* from *P*

Minkowski sum (dilation–erosion) comparison

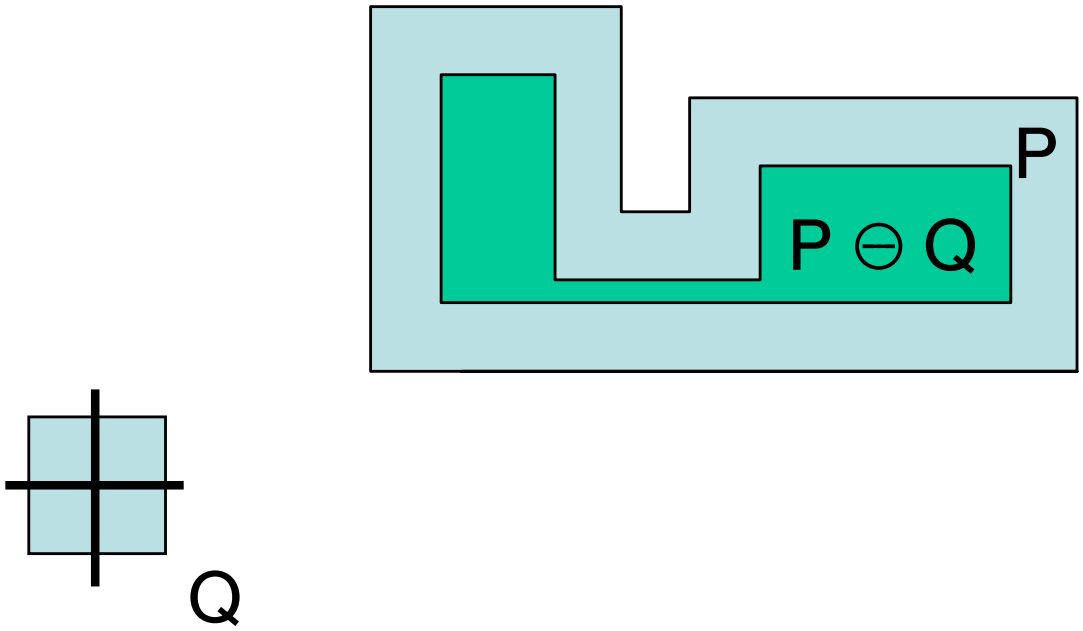


Figure: Dilation

Minkowski sum (dilation–erosion) comparison

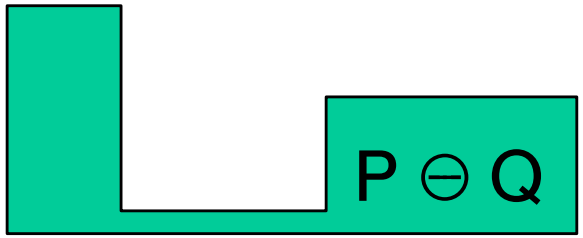


Figure: Dilation

Minkowski sum (dilation–erosion) comparison

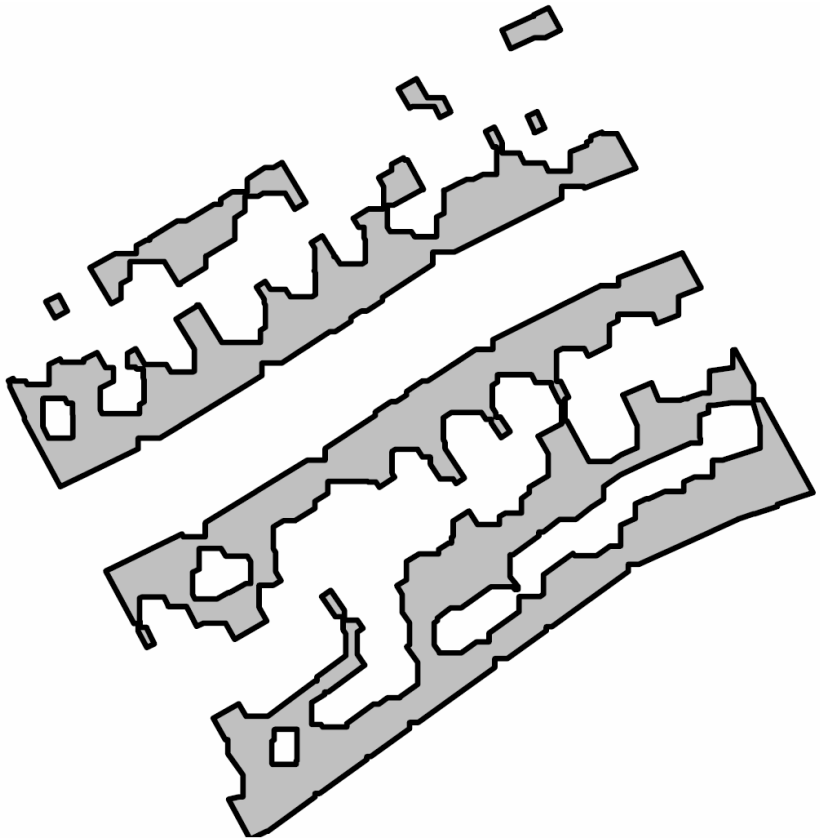


Figure: Dilation-Erosion, axis-aligned buildings

Minkowski sum (dilation–erosion) comparison

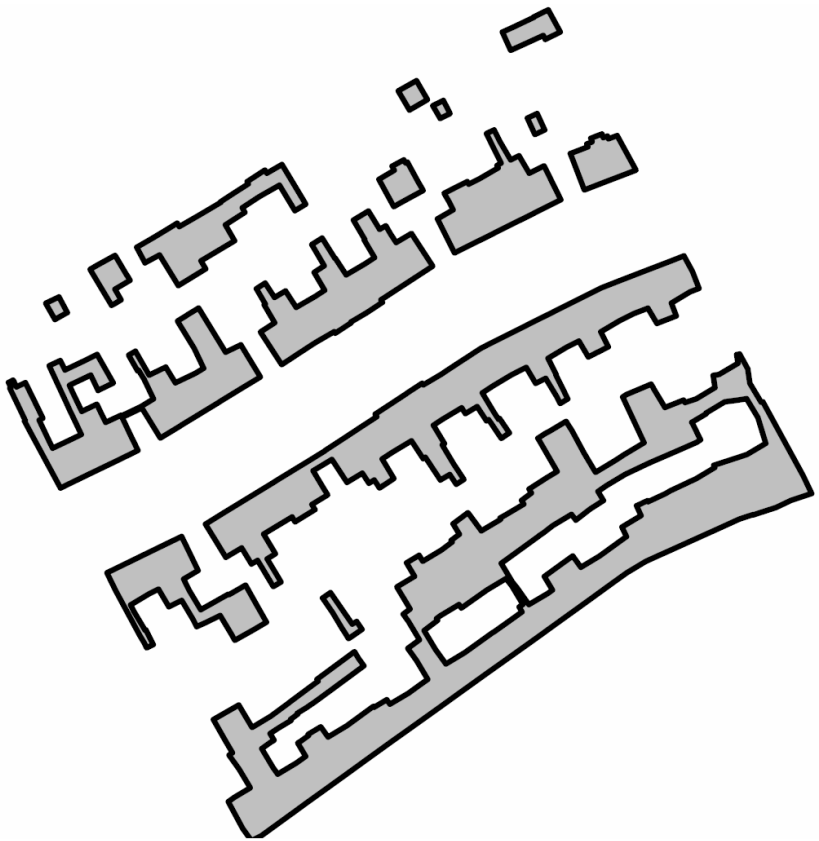


Figure: Dilation-Erosion, rotated buildings (main direction axis of building)

Linear axis around vertices with sharp angles

Image credit: Palfrader and Held (2015)

Linear axis around vertices with sharp angles