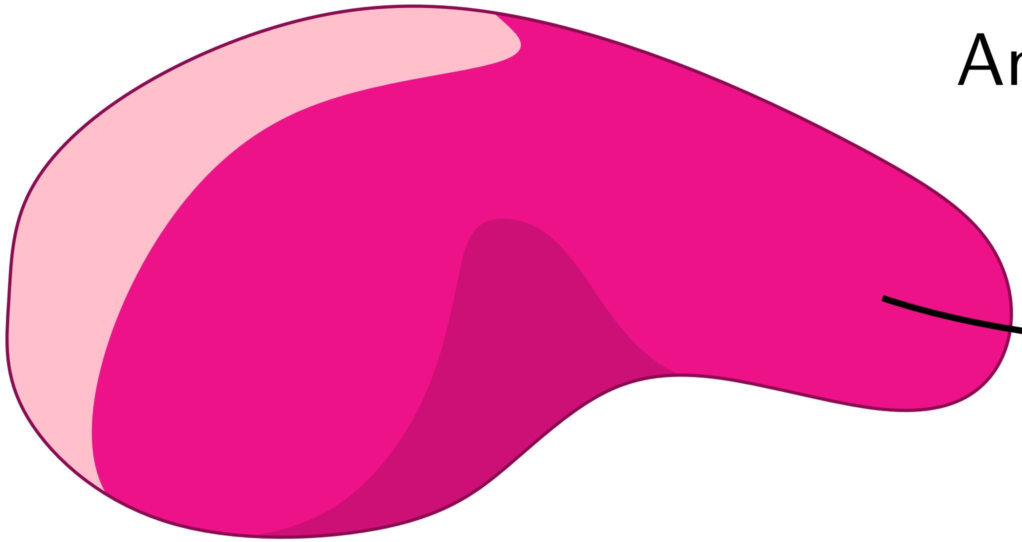
Variable-width contouring for additive manufacturing

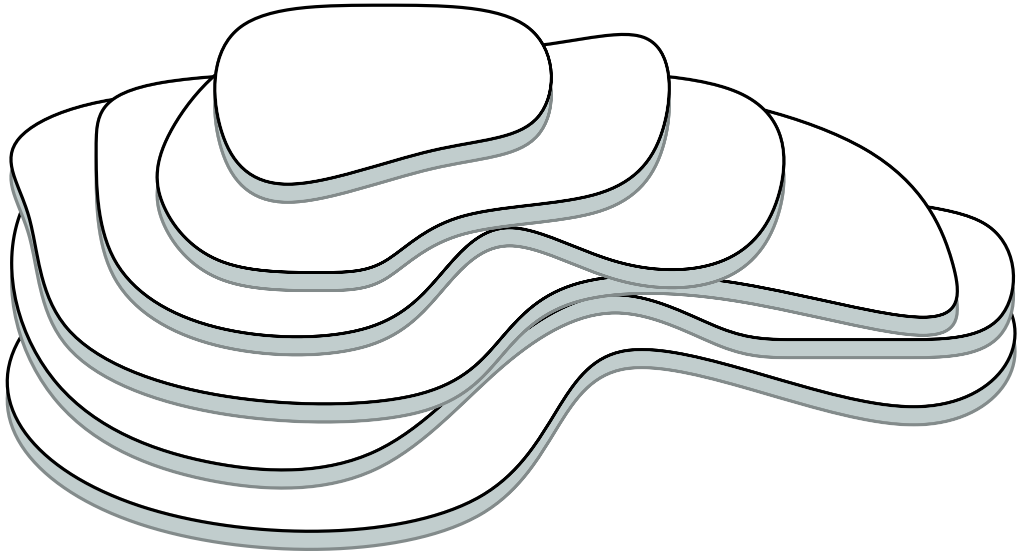
Samuel **Hornus**, Tim **Kuipers**, Olivier **Devillers**, Monique **Teillaud**,
Jonàs **Martínez**, Marc **Glisse**, Sylvain **Lazard** and Sylvain **Lefebvre**

SIGGRAPH 2020

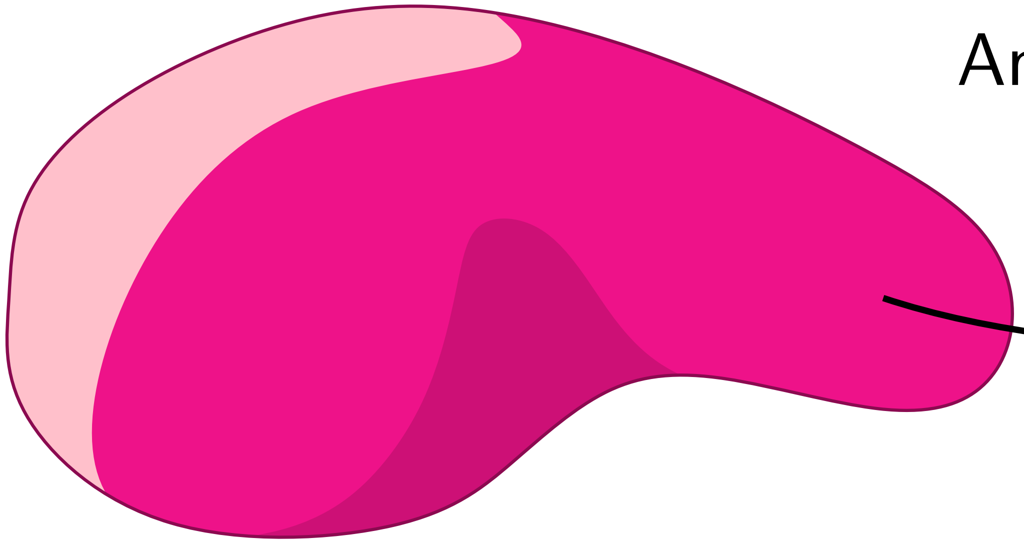
Context: 3D printing



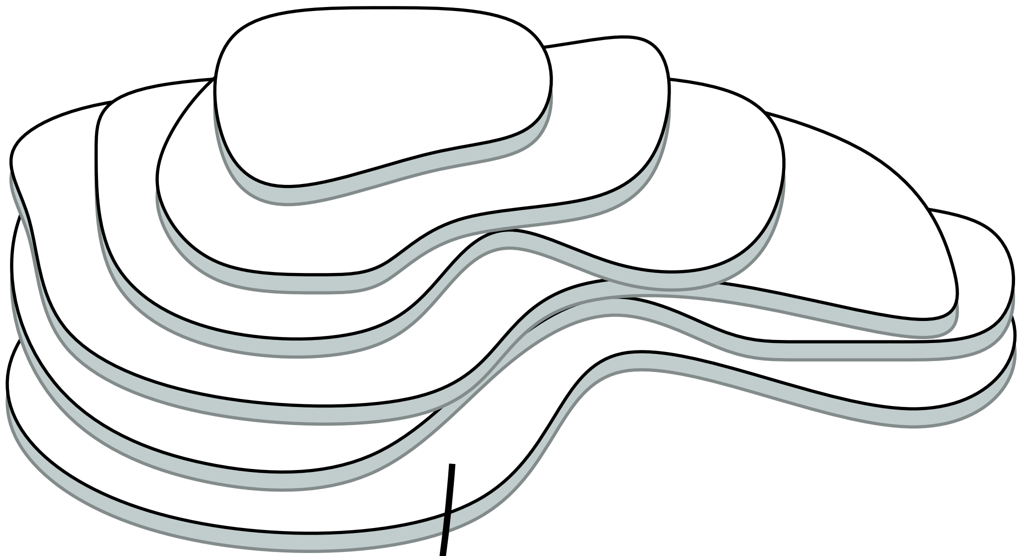
An object is fabricated as a stack of horizontal **layers**.

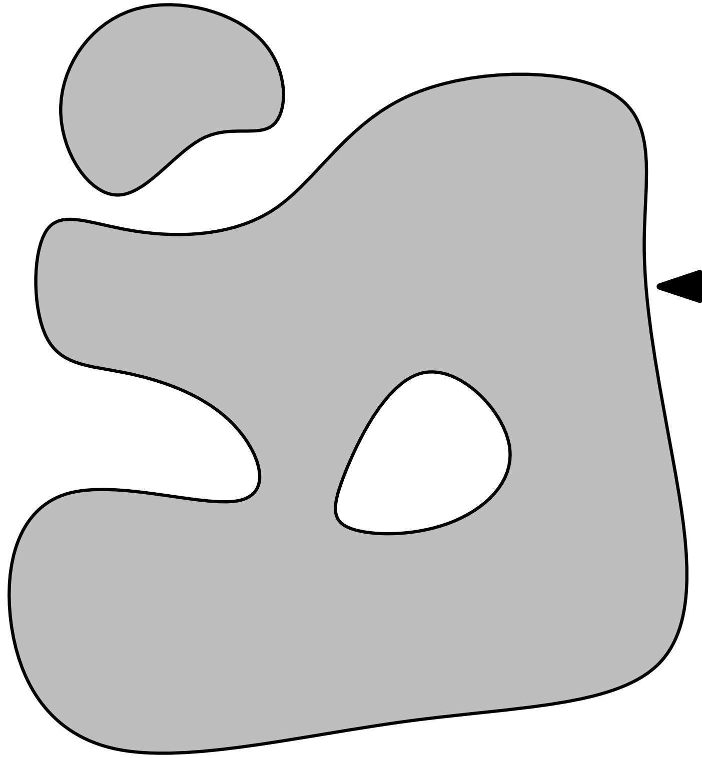


Context: 3D printing



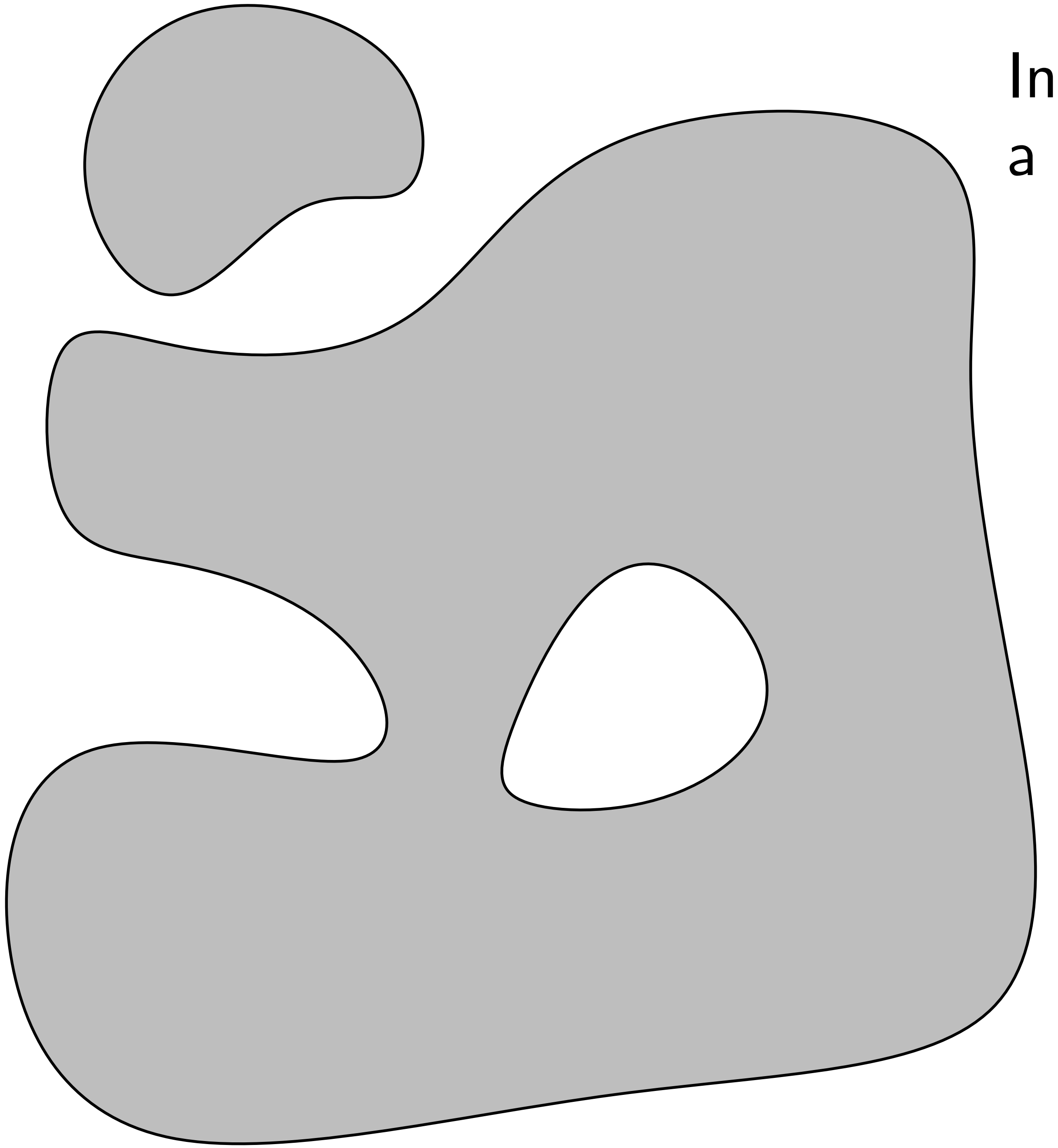
An object is fabricated
as a stack of
horizontal **layers**.





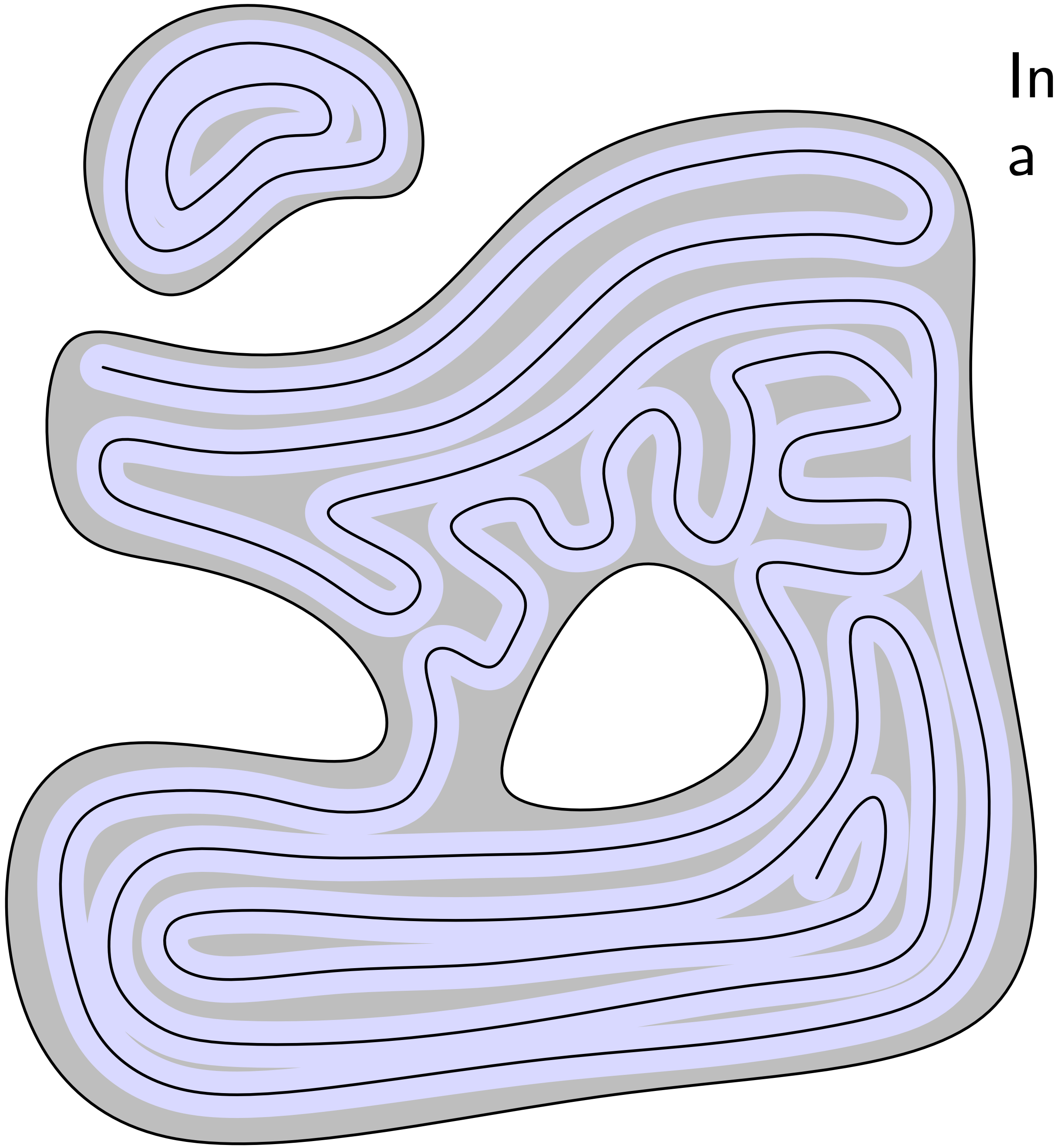
Each solid layer is modeled by
a **slice**: a planar shape.

Context: Fabricating one layer

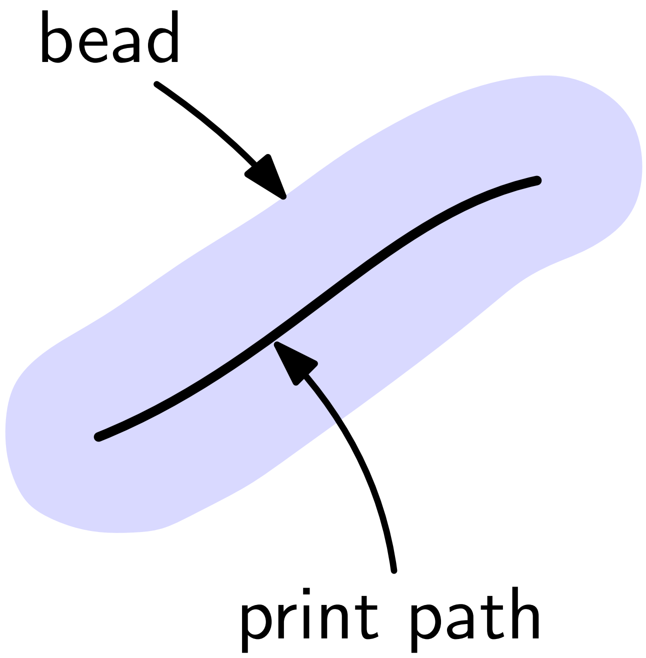


In turn, each layer is fabricated by solidifying a **bead** of some material, along a **print path**.

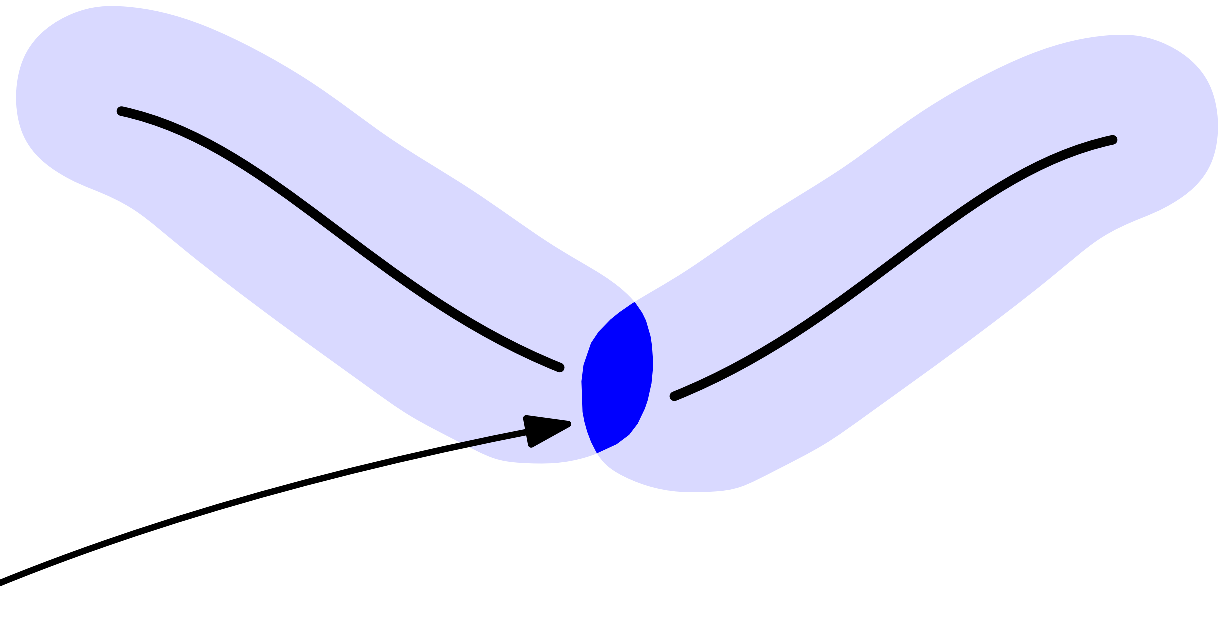
Context: Fabricating one layer



In turn, each layer is fabricated by solidifying a **bead** of some material, along a **print path**.

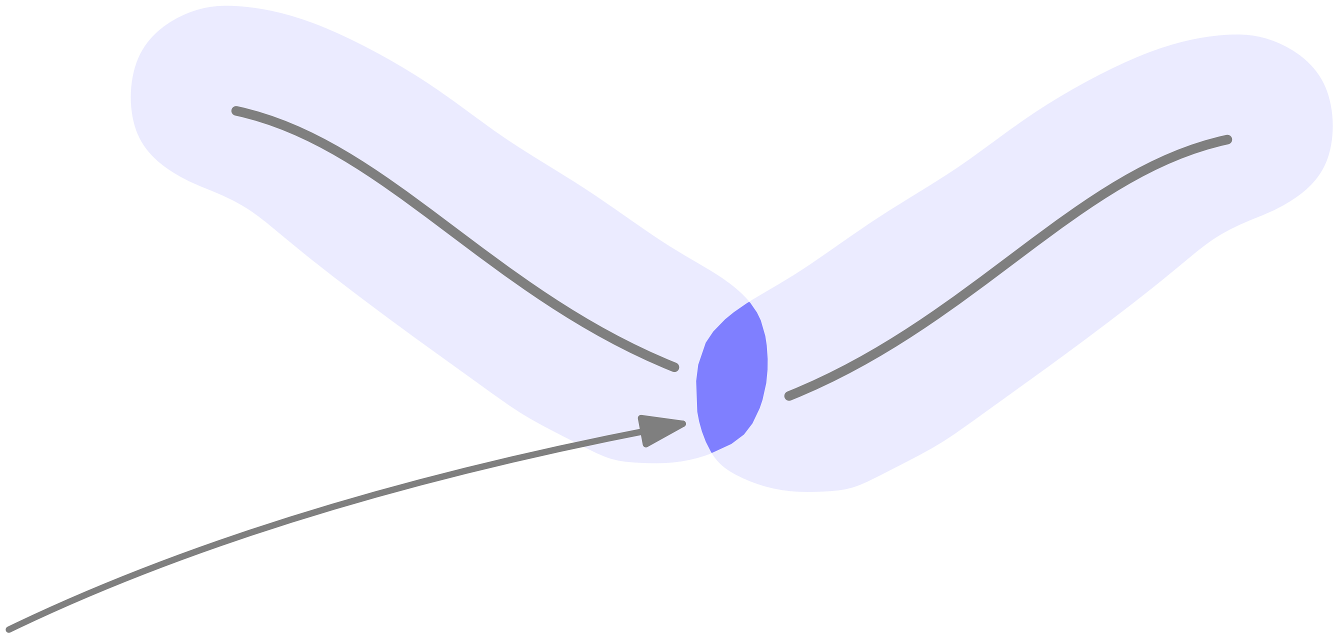


Overfill



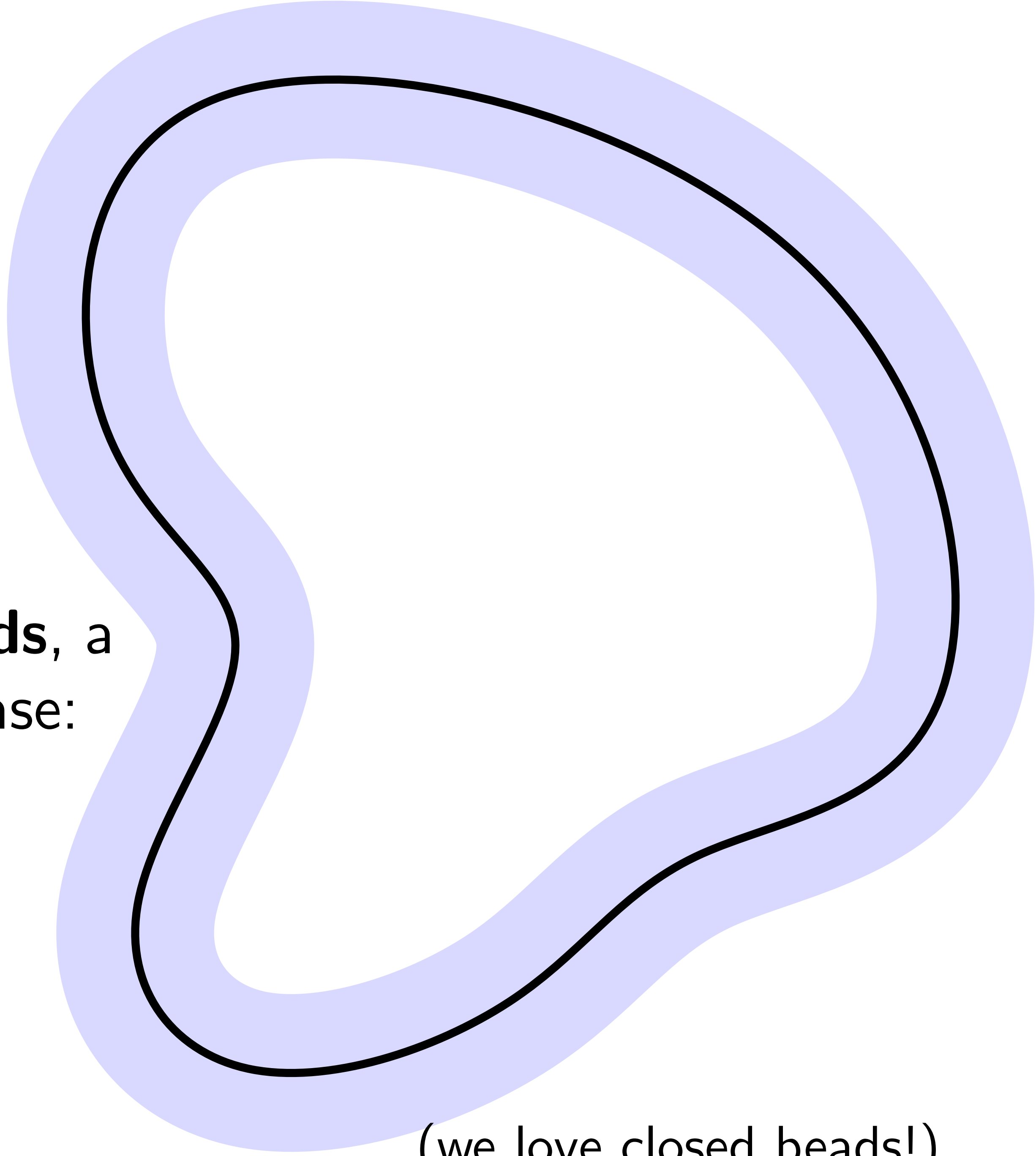
Overfill = forbidden...

Overfill



Overfill = forbidden...

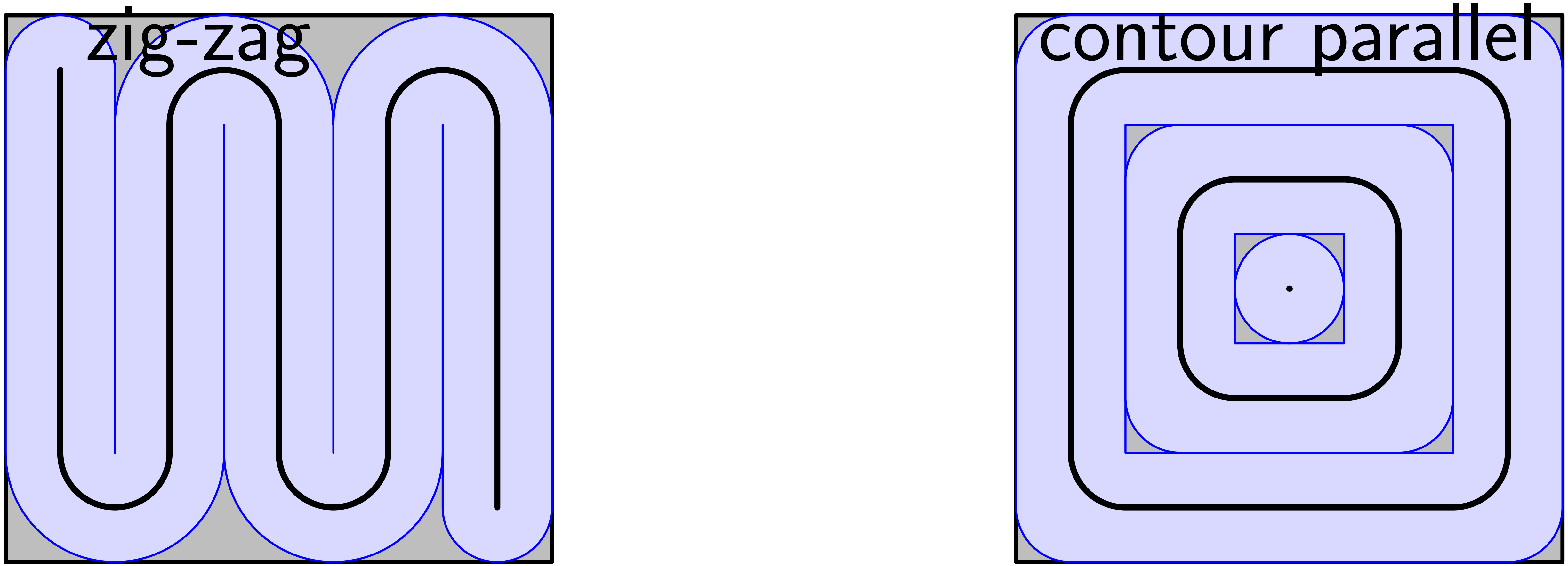
...except for **closed beads**, a well controlled special case:



(we love closed beads!)

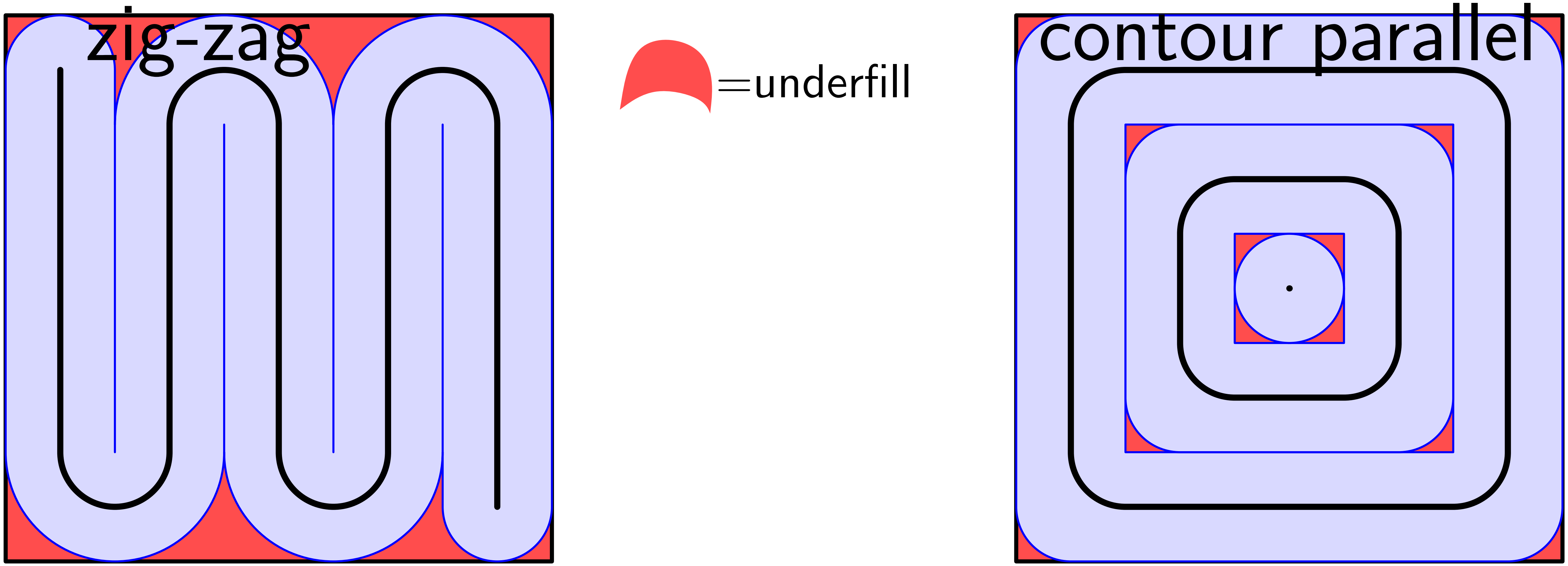
Underfill

Example: two classic ways to fill a square with a **constant-width** bead.



Underfill

Underfill is the existence of areas of the slice **not** covered by a solid bead.

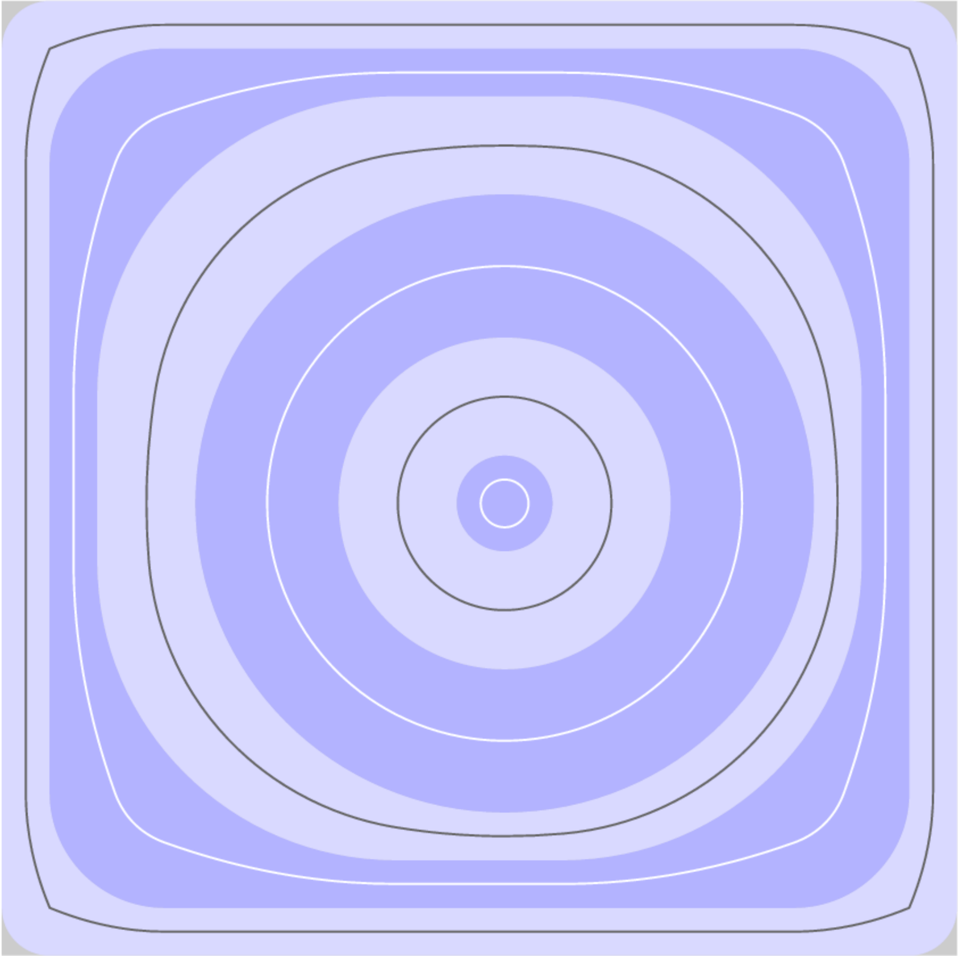


Underfill

Underfill is bad. We want to minimize it.

Our contribution is a new technique for designing print paths that produces

- no overfill (this is somewhat easy)
- a small amount of underfill (almost 10x less than the state of the art)

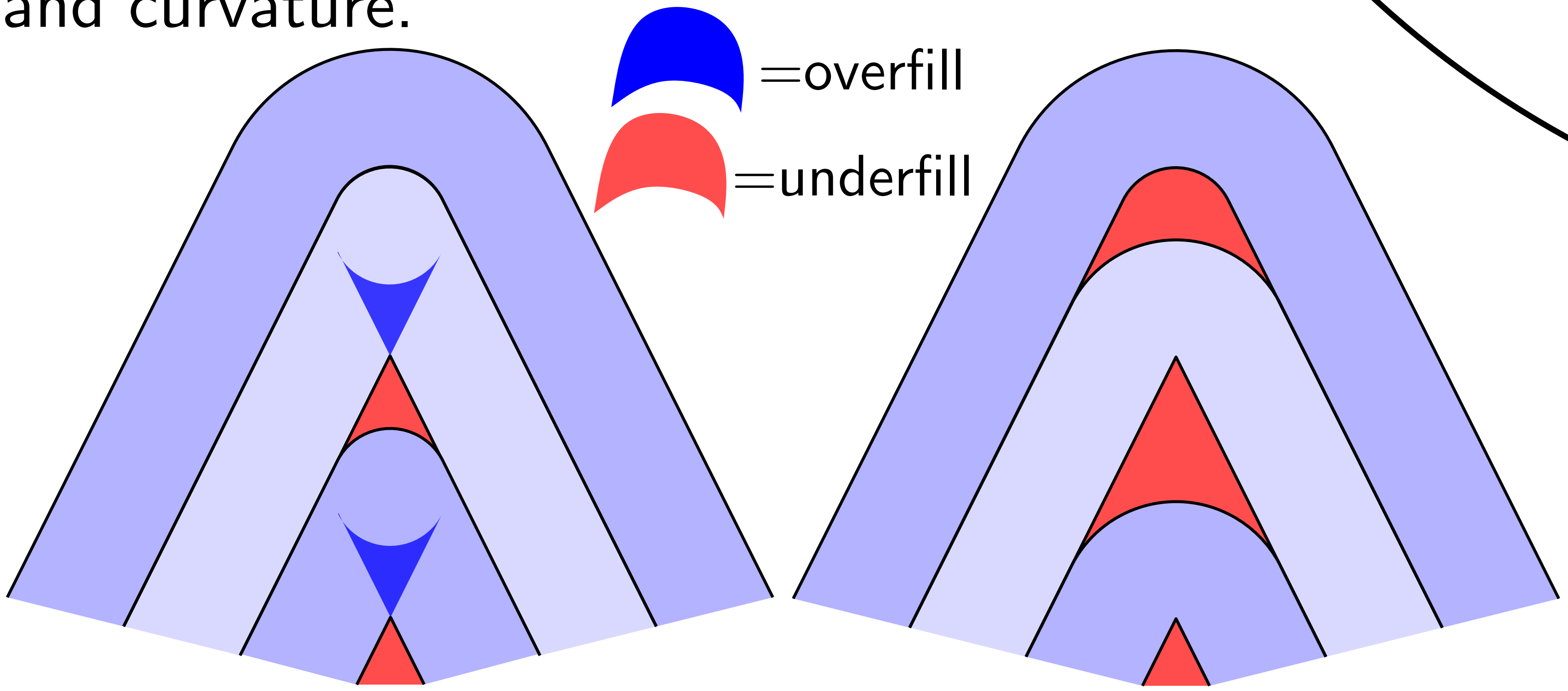


What to do?

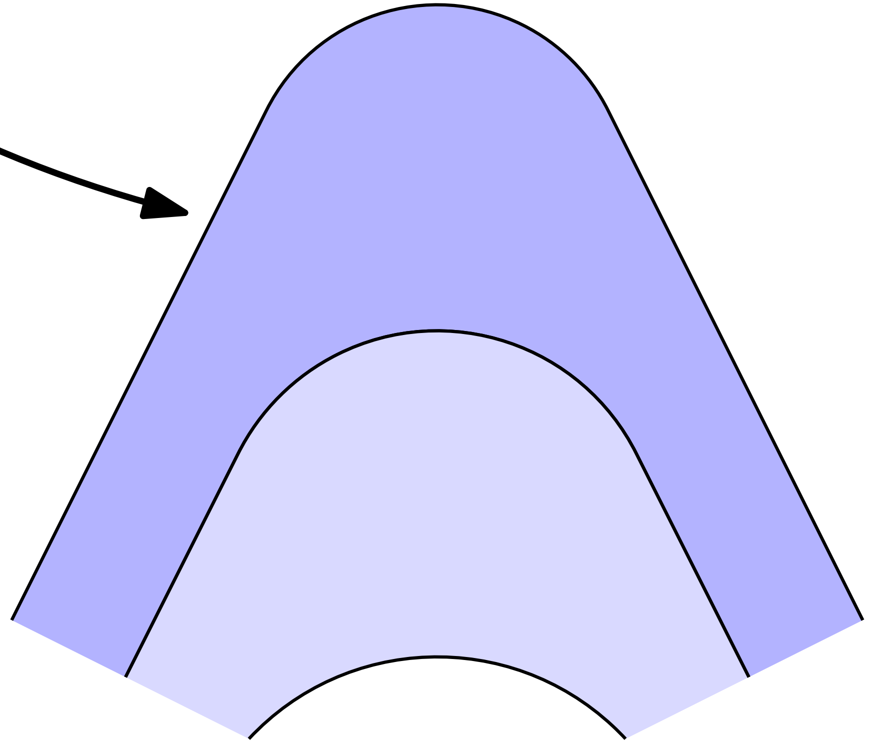
Earlier works suggesting to use variable-width beads:

- Jin, Du, and He. Journal of Manufacturing Systems 44 (2017).
- Kuipers, Doubrovski, Wu, and Wang. Computer-Aided Design 128 (2020).

We follow suit, use **closed, variable-width beads** and try to minimize their number and curvature.



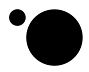
Uniform-width parallel contouring wo. (left) and w. (right) regularization.



Our technique.

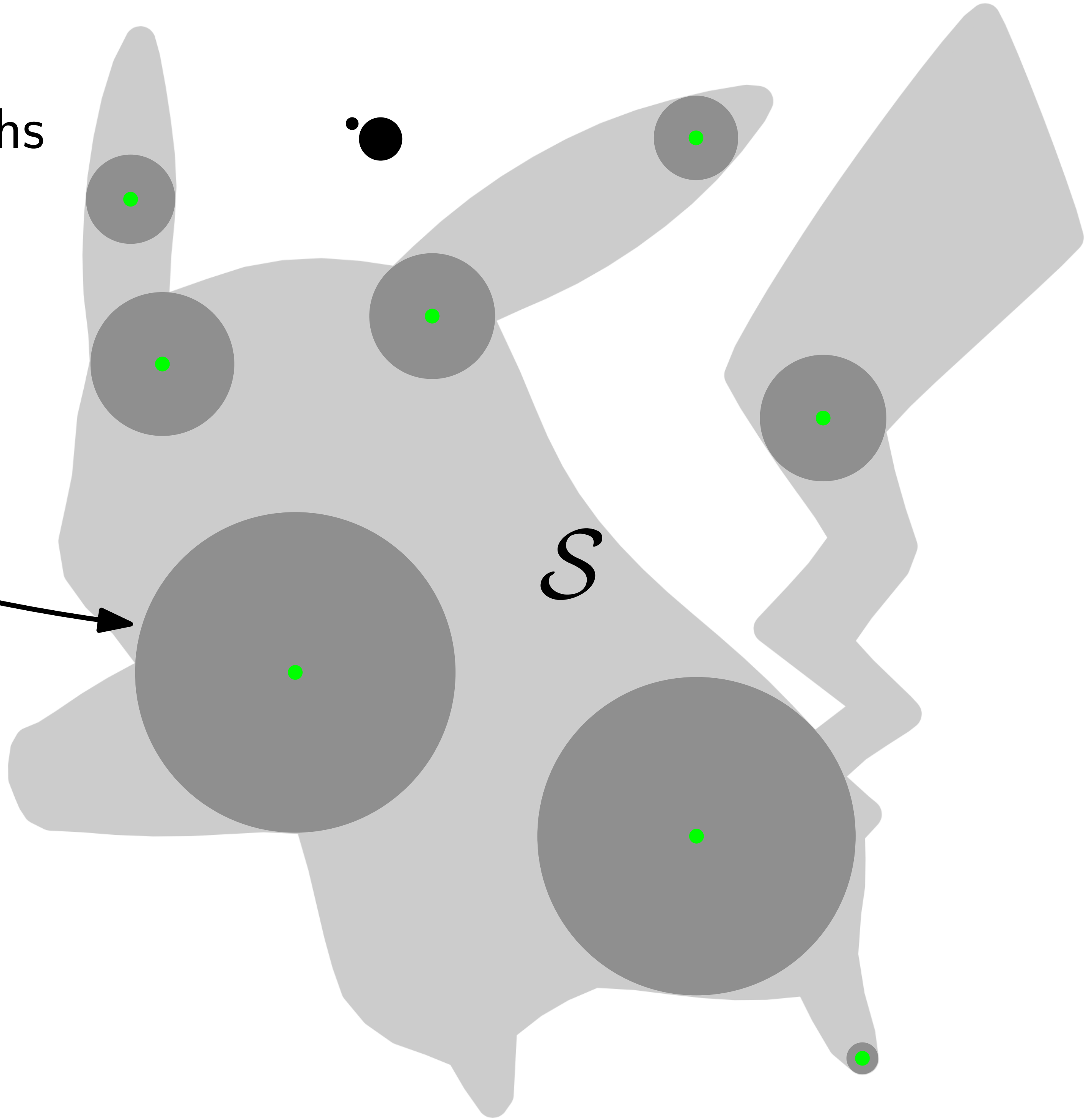
Inputs & data structure

- A range $[2\gamma, 2\Gamma]$ of feasible bead widths (specific to target 3D printer).



Inputs & data structure

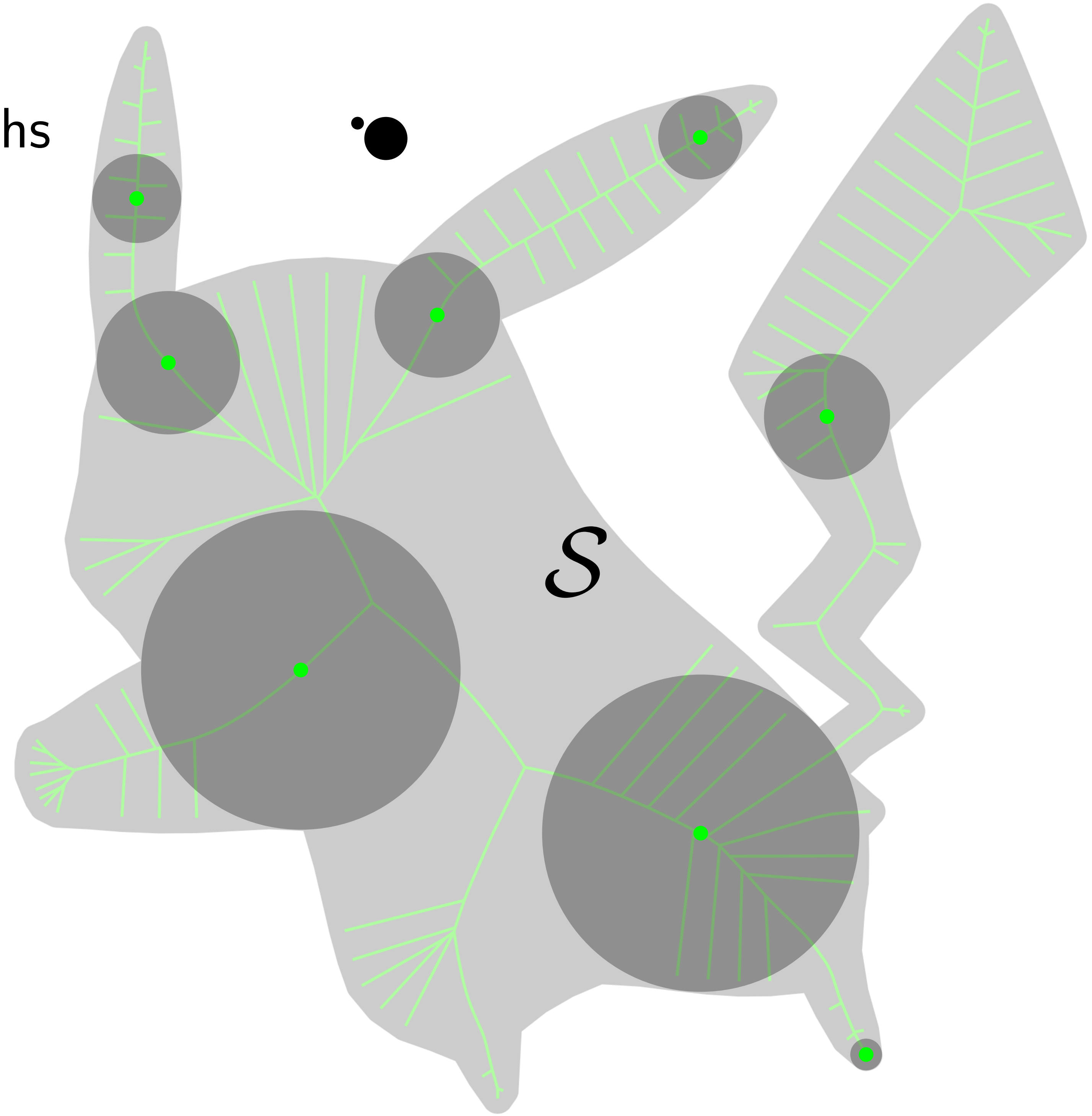
- A range $[2\gamma, 2\Gamma]$ of feasible bead widths (specific to target 3D printer).
- A $\mathbf{2\gamma}$**-fat** planar shape $\mathcal{S}$: all the **maximal disks** inside $\mathcal{S}$ have radius $\geq 2\gamma$. [1]



[1] In practice, slices are polygons. We process them into 2γ-fat shapes.

Inputs & data structure

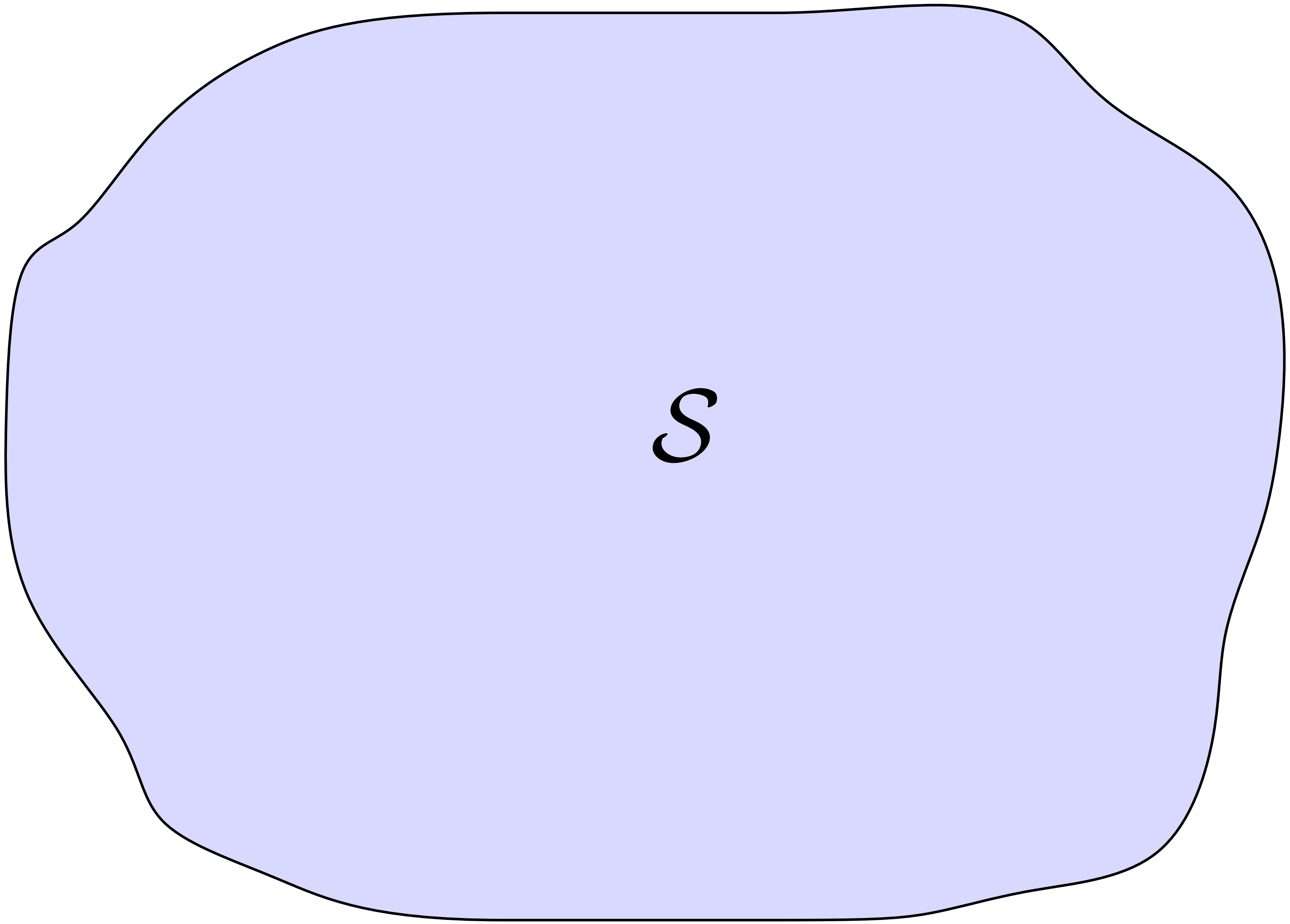
- A range $[2\gamma, 2\Gamma]$ of feasible bead widths (specific to target 3D printer).
- A **2γ-fat** planar shape $\mathcal{S}$: all the **maximal disks** inside $\mathcal{S}$ have radius $\geq 2\gamma$. [1]
- An explicit representation of the medial axis of $\mathcal{S}$: $\mathbf{MA}(\mathcal{S})$ is the closure of the set of centers of maximal disks in $\mathcal{S}$.



[1] In practice, slices are polygons. We process them into 2γ-fat shapes.

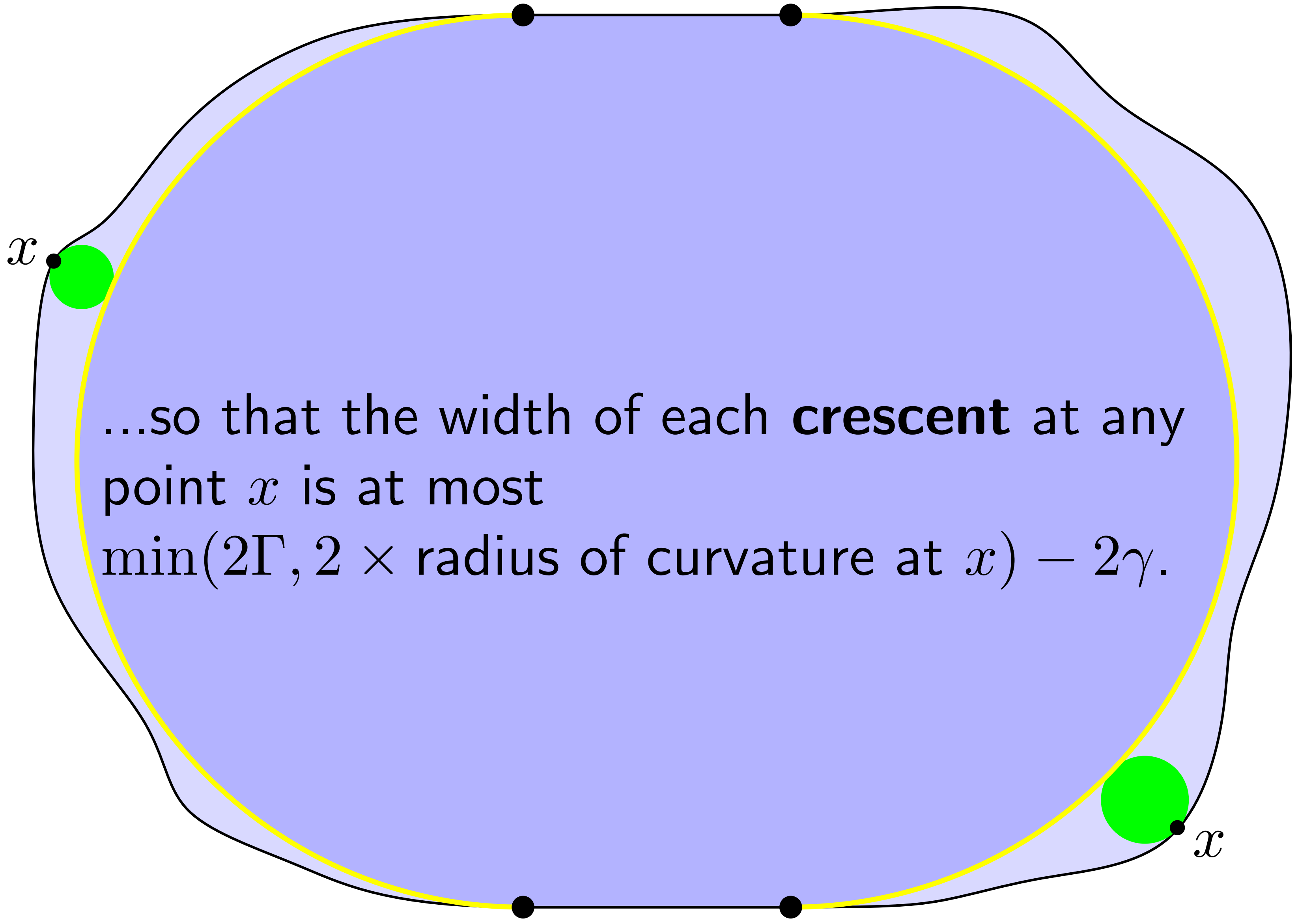
Variable-width contouring

Given a shape $\mathcal{S}$, we model a bead that stays in contact with the boundary of $\mathcal{S}$ and make the remaining inner shape "rounder."



Variable-width contouring

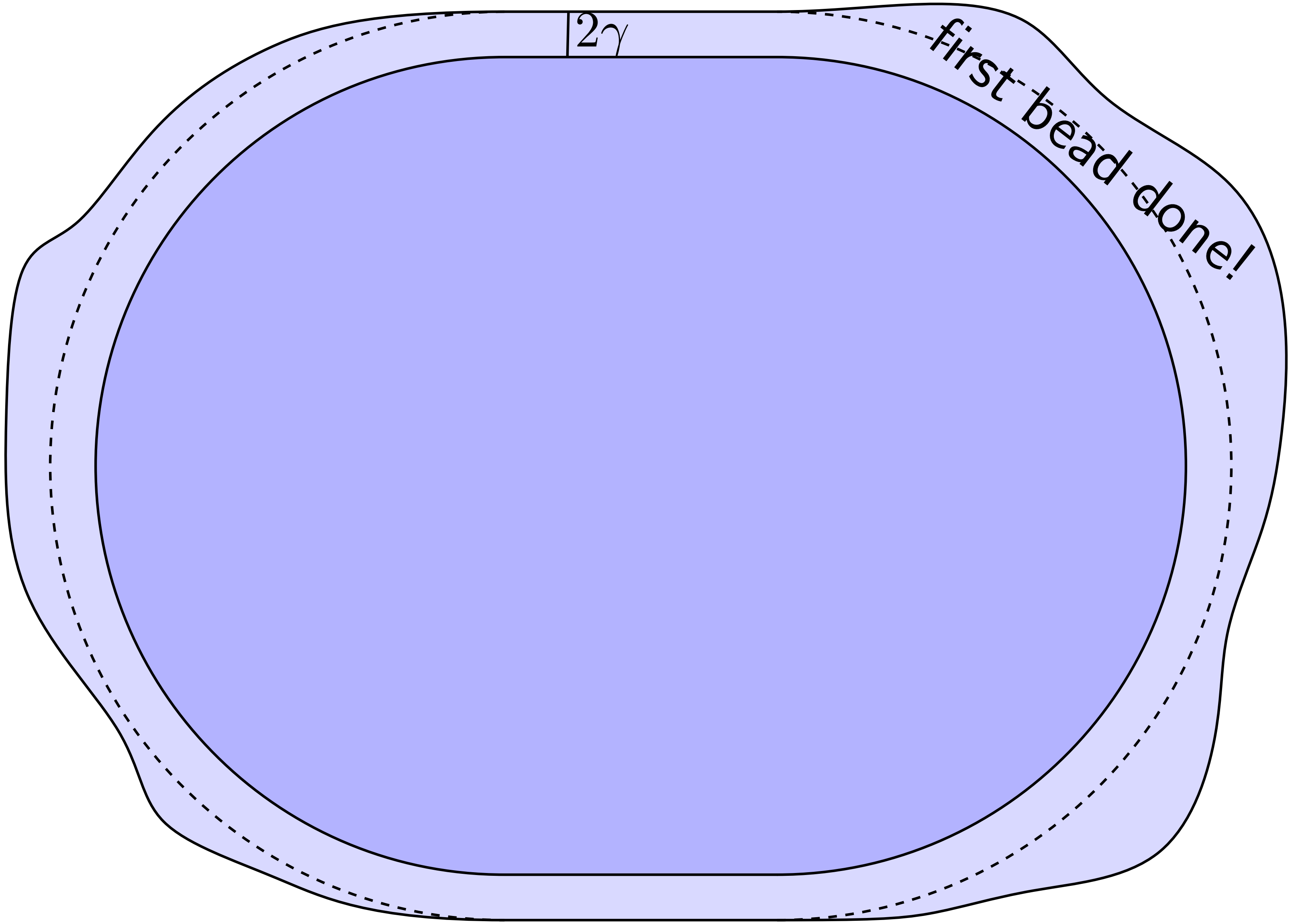
To do so, we replace parts of the boundary $\partial\mathcal{S}$ by inner tangent circular arcs (yellow)...



...so that the width of each **crescent** at any point x is at most $\min(2\Gamma, 2 \times \text{radius of curvature at } x) - 2\gamma$.

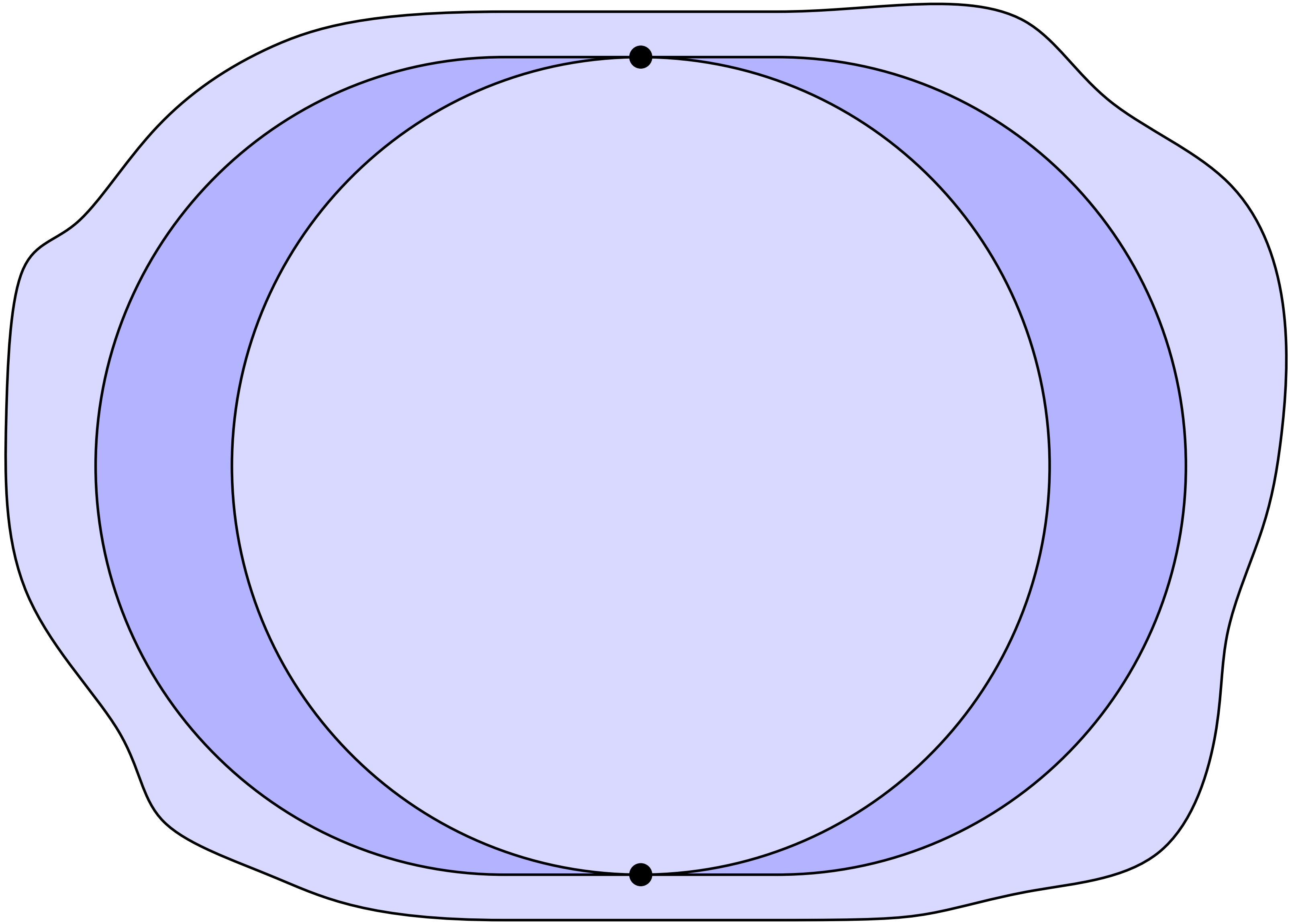
Variable-width contouring

Then we do a parallel offset of 2γ and obtain a bead of width within the allowed range.



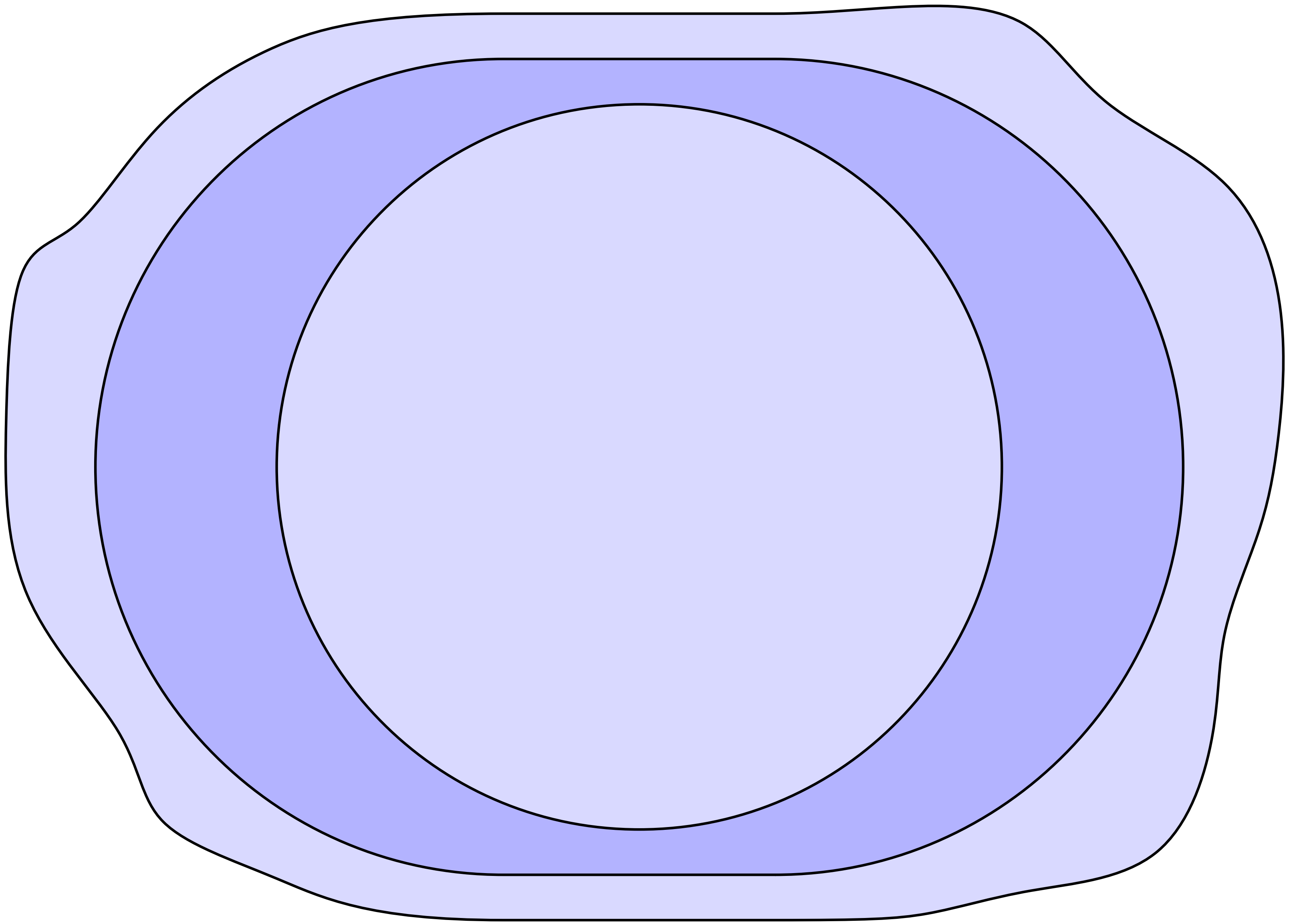
Variable-width contouring

Now we repeat the process



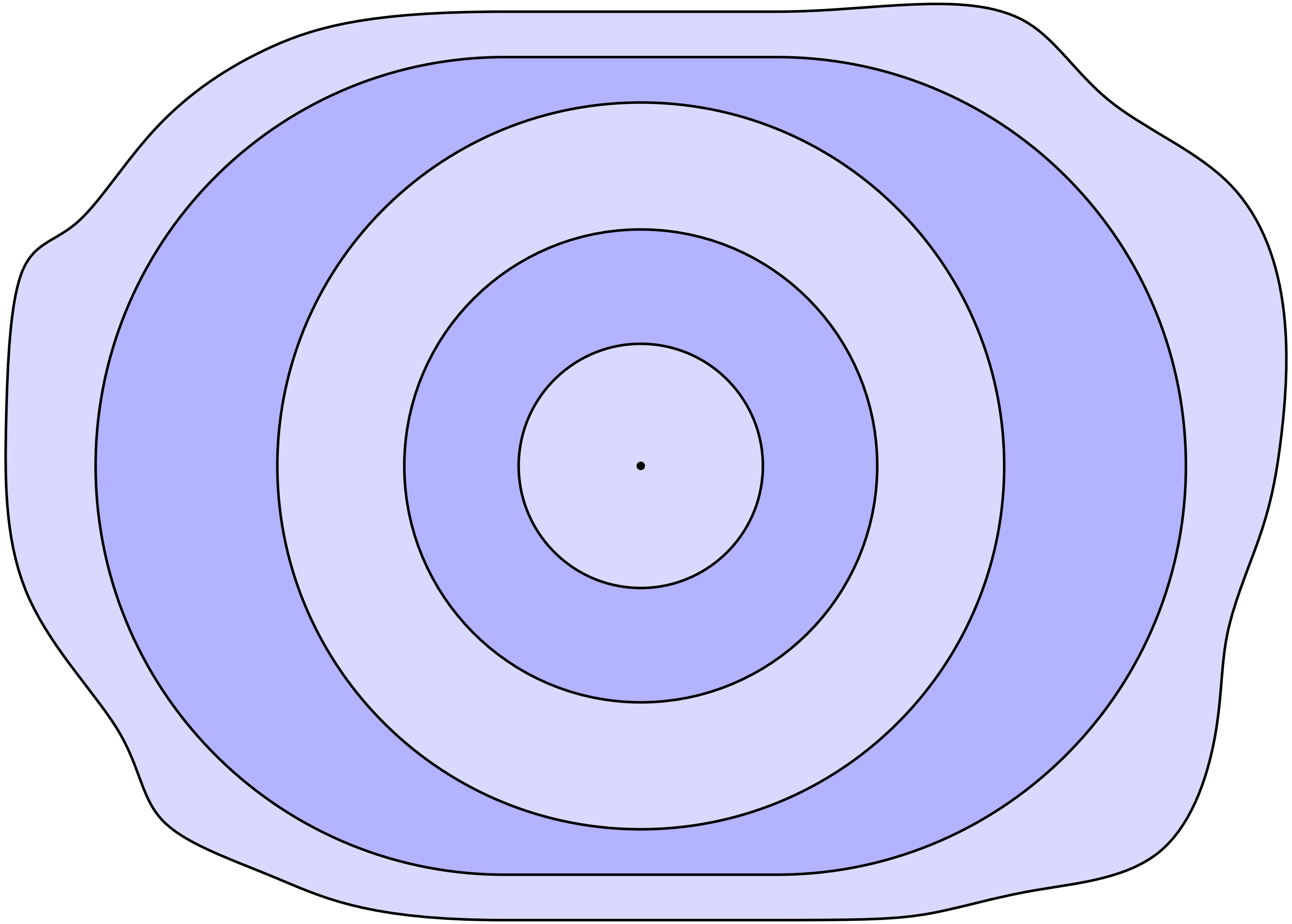
Variable-width contouring

Now we repeat the process



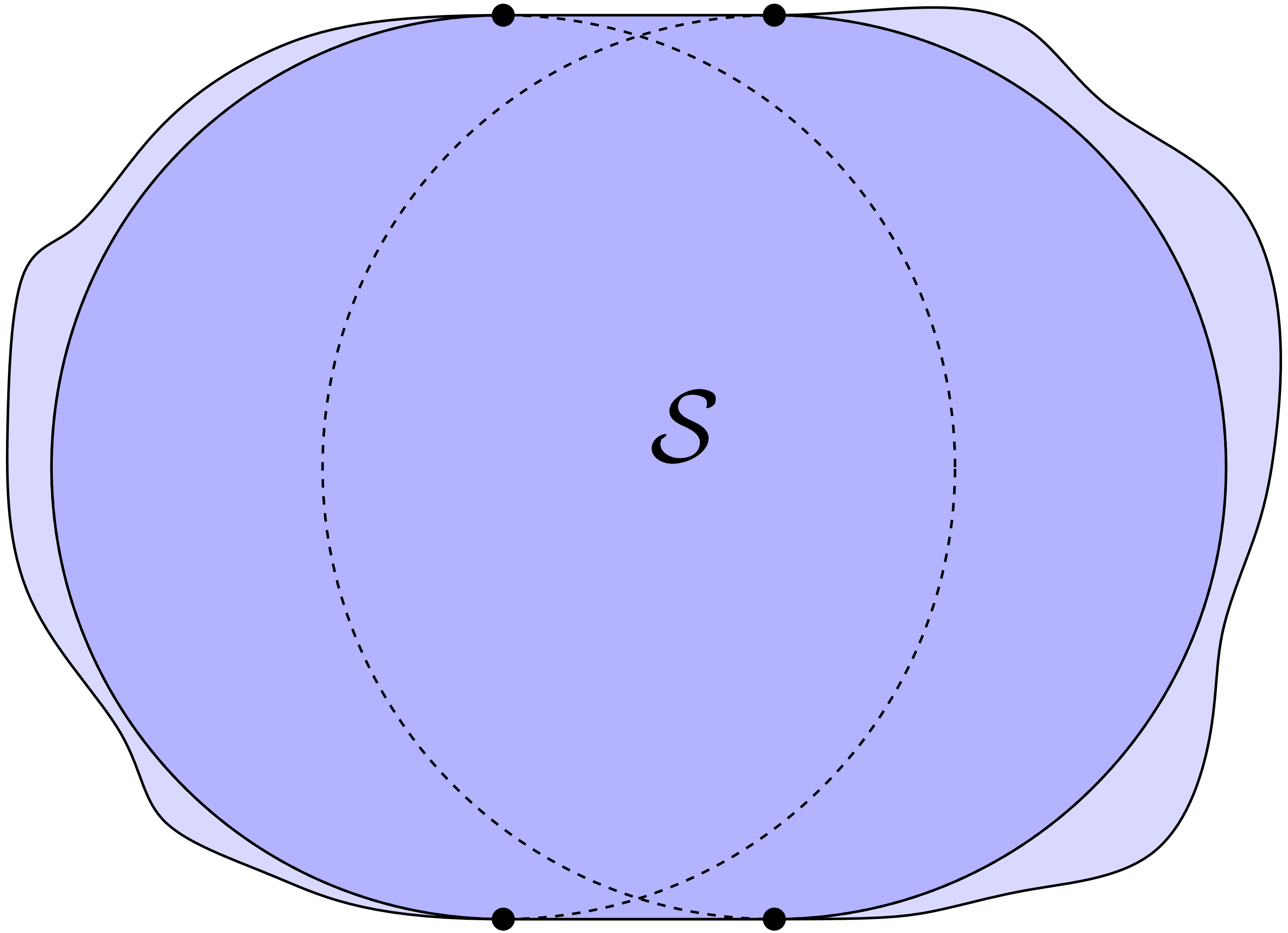
Variable-width contouring

Now we repeat the process



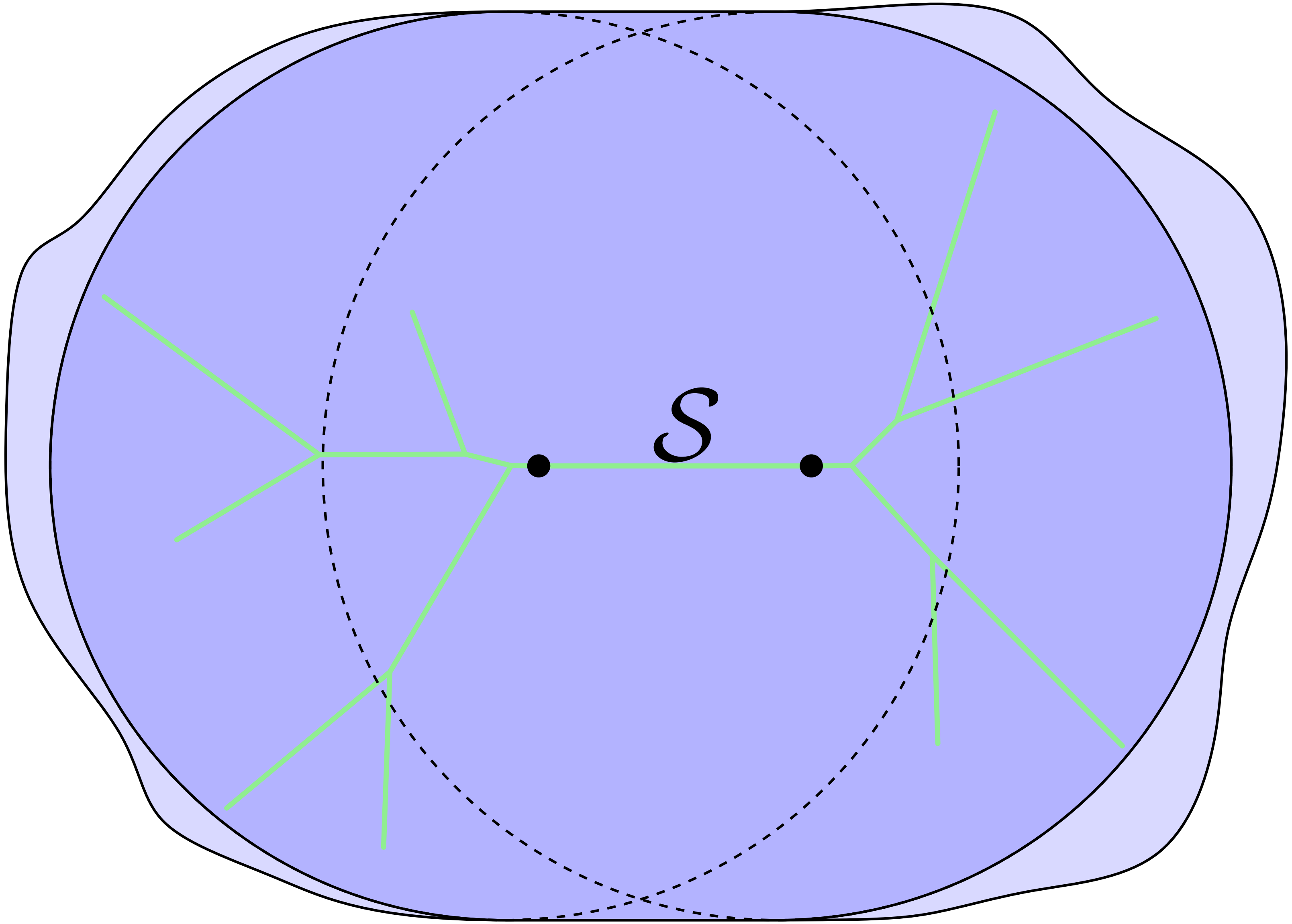
Variable-width contouring

The circles supporting the tangent circular arcs are chosen as the boundary of maximal disks in $\mathcal{S}$. Hence, their center lies on the medial axis $\mathbf{MA}(\mathcal{S})$ of $\mathcal{S}$.



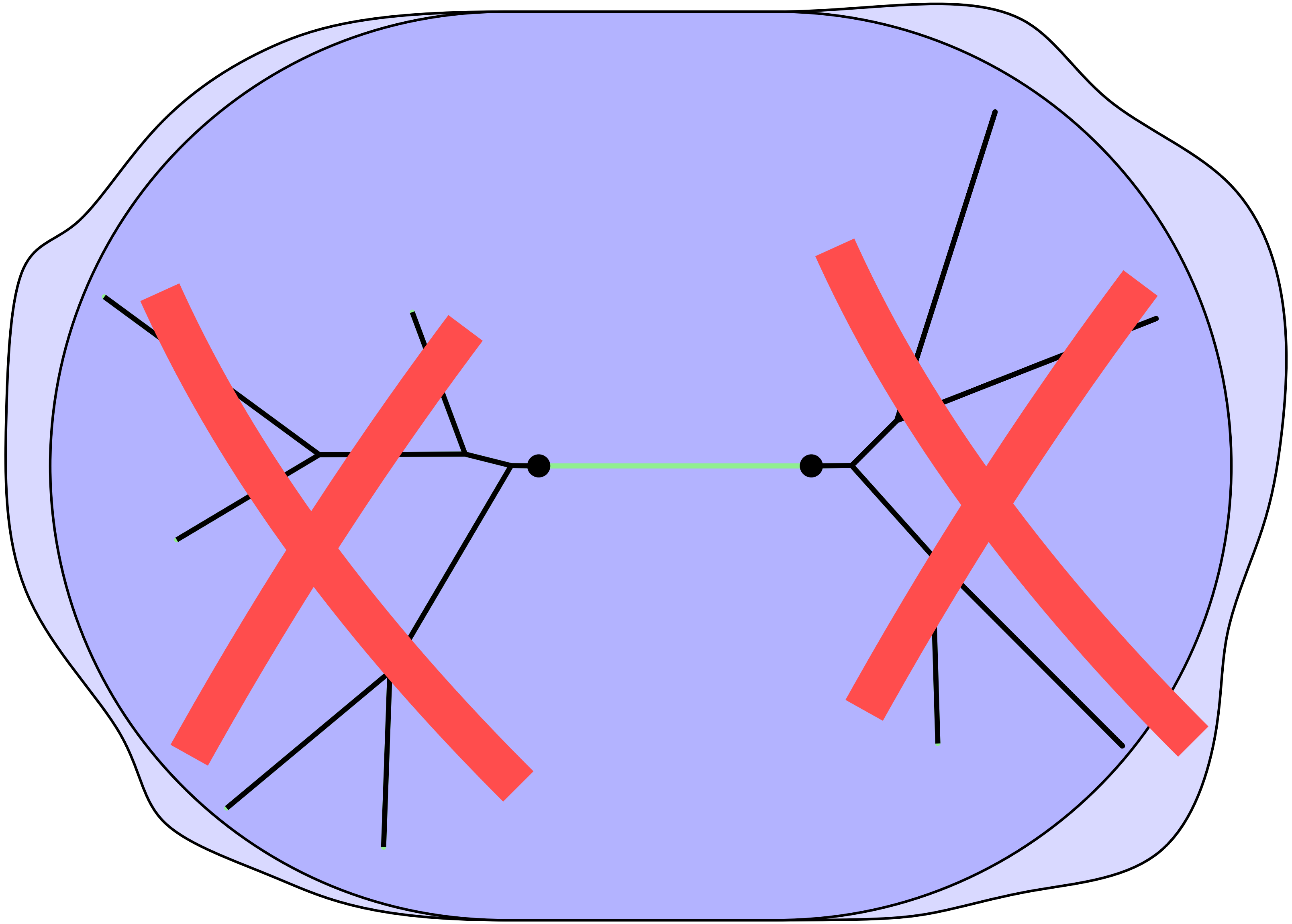
Variable-width contouring

The circles supporting the tangent circular arcs are chosen as the boundary of maximal disks in $\mathcal{S}$. Hence, their center lies on the medial axis $\mathbf{MA}(\mathcal{S})$ of $\mathcal{S}$.



Variable-width contouring

Replacing by circular arc = **trimming** the medial axis!



Variable-width contouring: basics

1. Trimming the medial axis: removes crescents of width $\leq 2\Gamma - 2\gamma$ from the shape.

2. Parallel offset : removes a band of width exactly 2γ, which together with the crescents, form a bead of width varying within $[2\gamma, 2\Gamma]$.

Variable-width contouring: basics

1. Trimming the medial axis: removes crescents of width $\leq 2\Gamma - 2\gamma$ from the shape.

2. Parallel offset : removes a band of width exactly 2γ, which together with the crescents, form a bead of width varying within $[2\gamma, 2\Gamma]$.

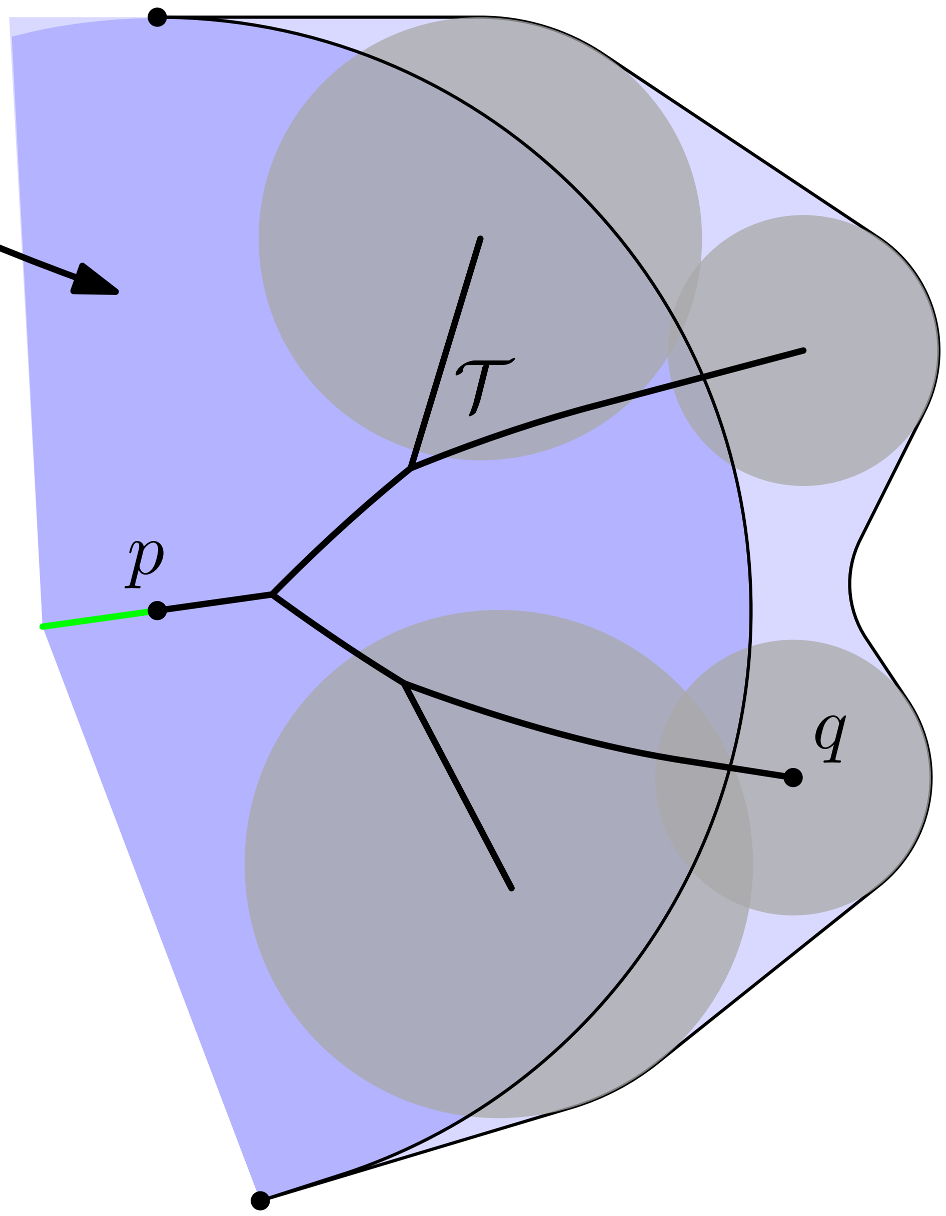
If the input is a polygon, then:

- the medial axis is computable (CGAL, BOOST) and
- the two operations above produce shapes with **linear** or **circular** boundary arcs only.

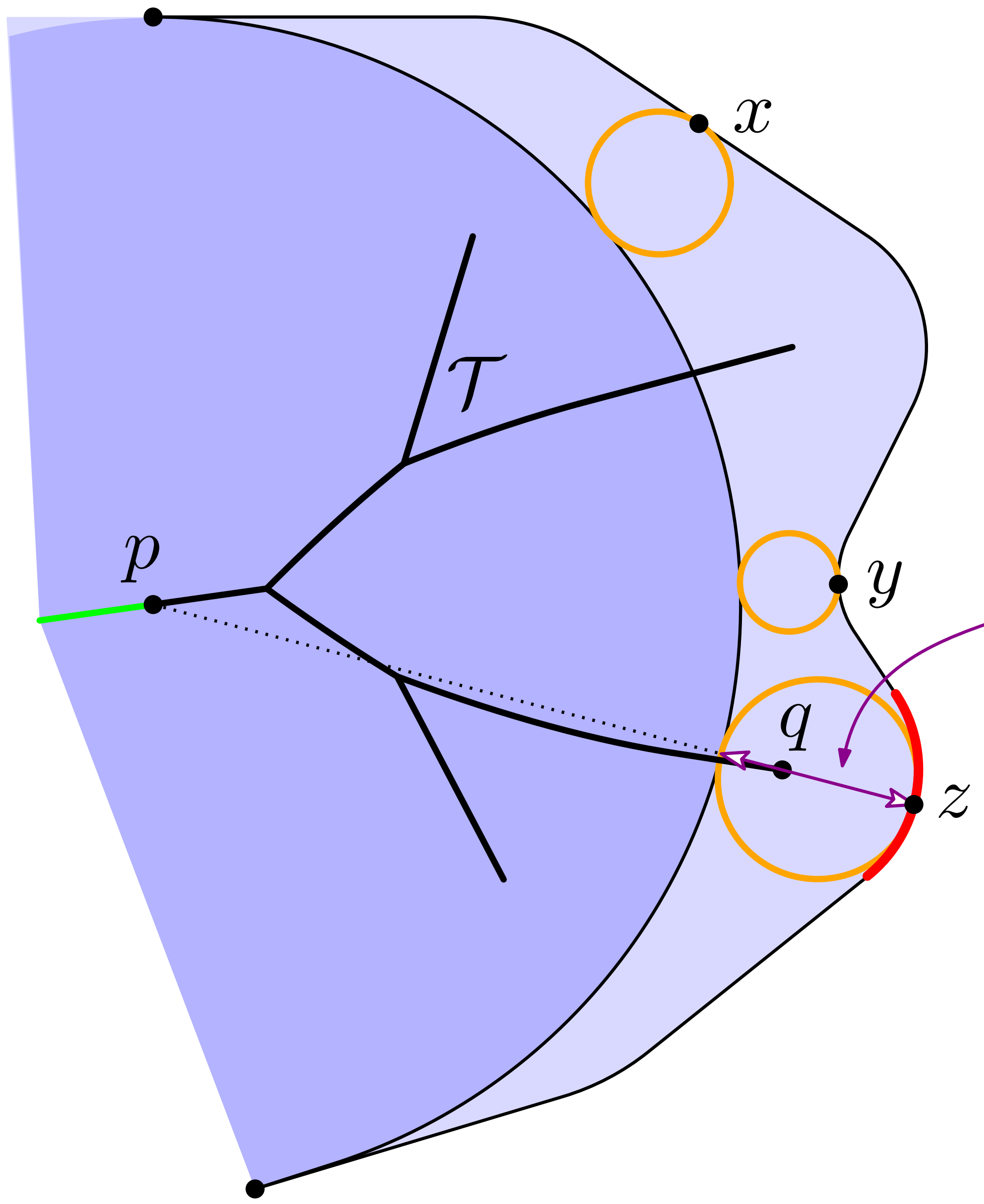
Corollary: in that case, each bead is bounded by linear or circular arcs only.

Trimming

maximal disk
centered on p

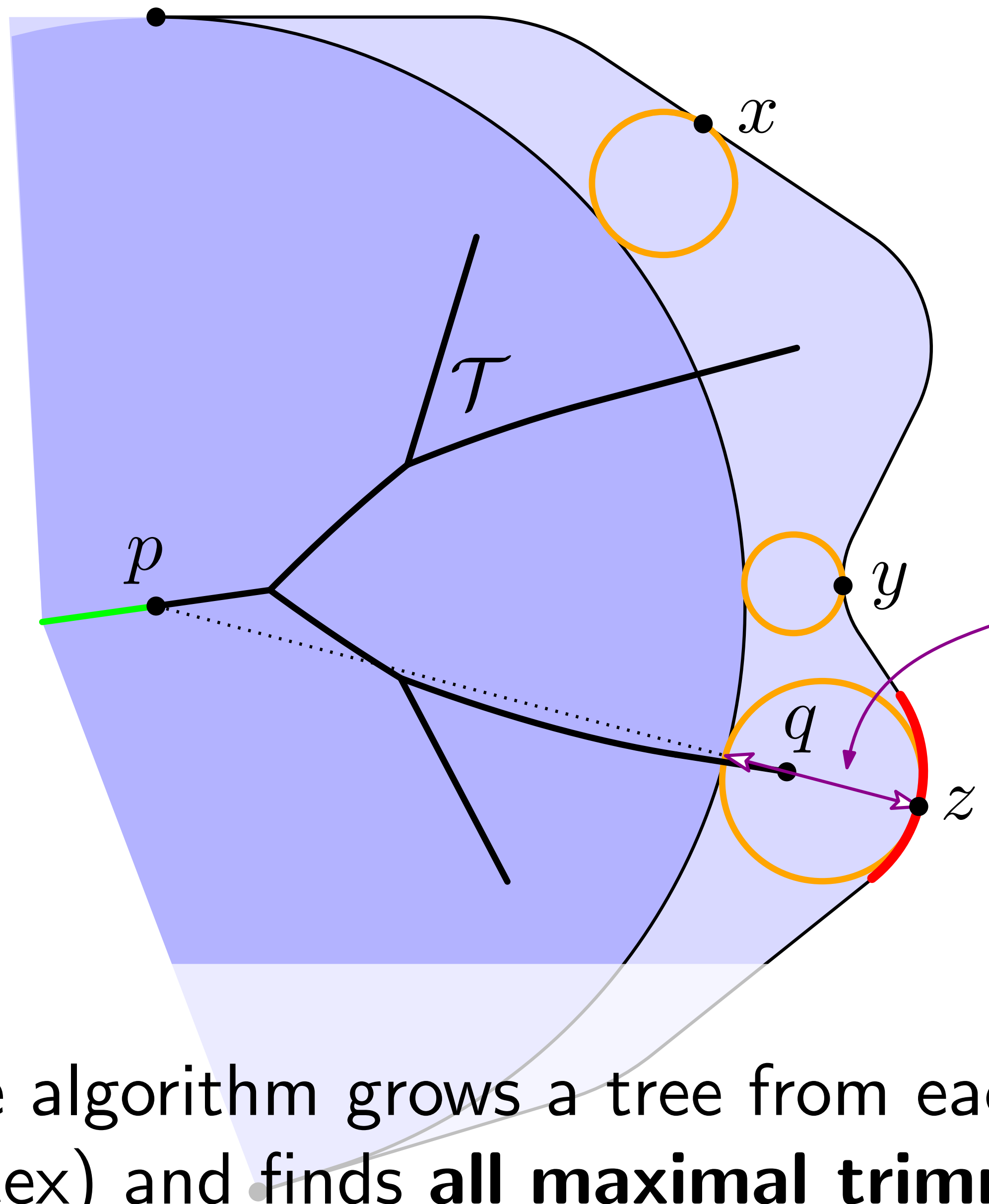


Trimming



$w(q,p)$ is the local maximal crescent width associated with root p and leaf q.

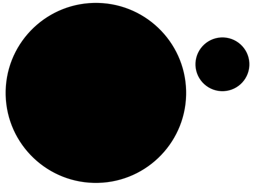
Trimming



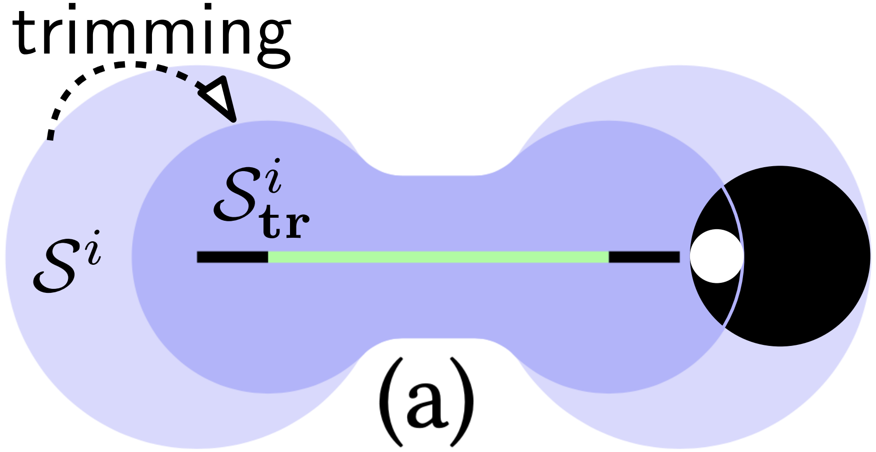
$w(q,p)$ is the local maximal crescent width associated with root p and leaf q.

The algorithm grows a tree from each leaf (degree-1 vertex) and finds **all maximal trimmable trees**.

Complete picture with Collapsing



: max and min bead widths

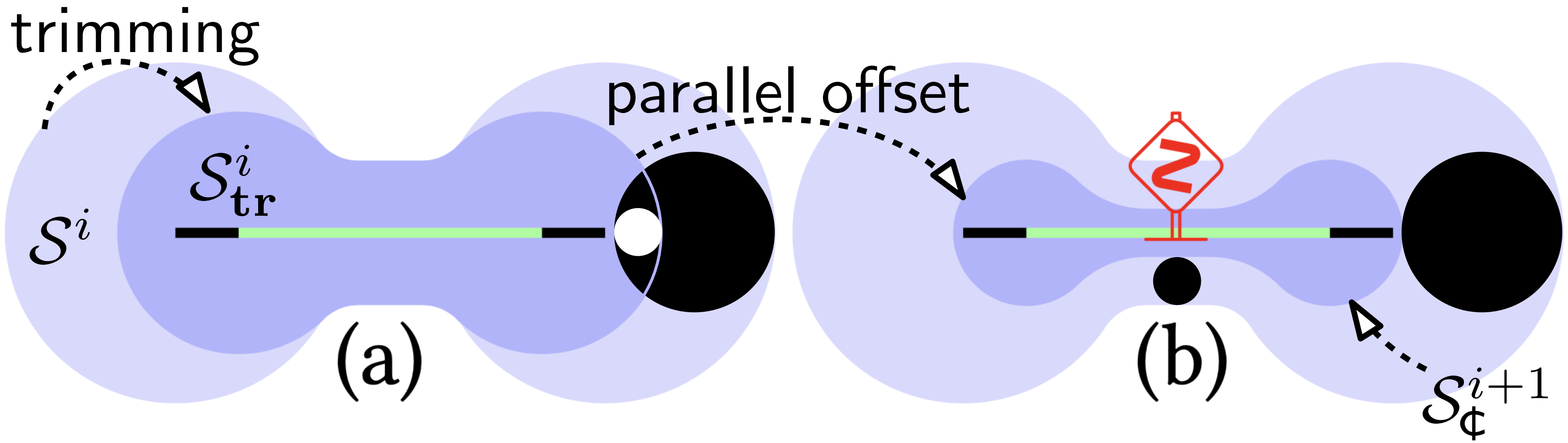


$\mathcal{S}^i$ (a)

$\big\downarrow$ trimming

$\mathcal{S}^i_{\mathbf{tr}}$

Complete picture with Collapsing



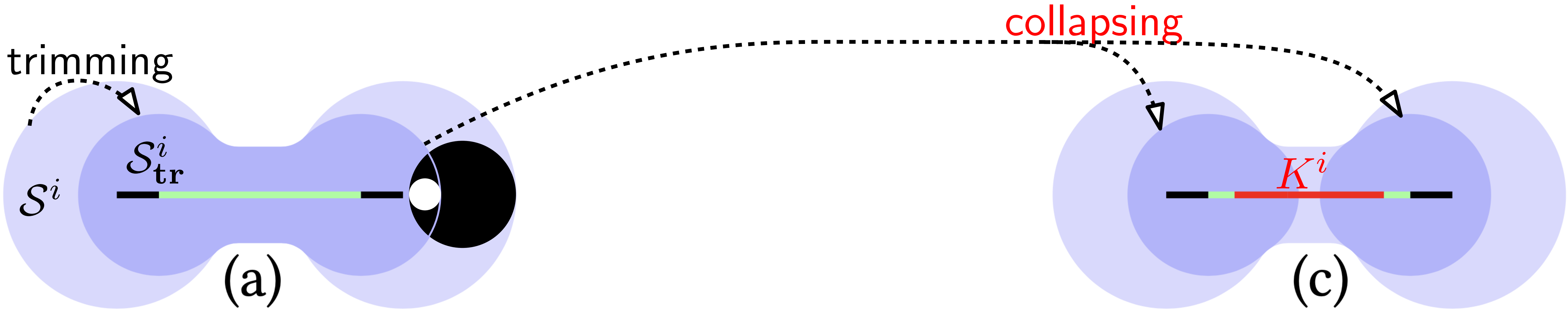
$\mathcal{S}^i$ (a)

$\downarrow$ trimming

$\mathcal{S}^i_{\mathbf{tr}} \xrightarrow{//\text{offset}}$ (b) $\mathcal{S}^{i+1}_{¢}$

Complete picture with Collapsing

: max and min bead widths



$\mathcal{S}^i$ (a)

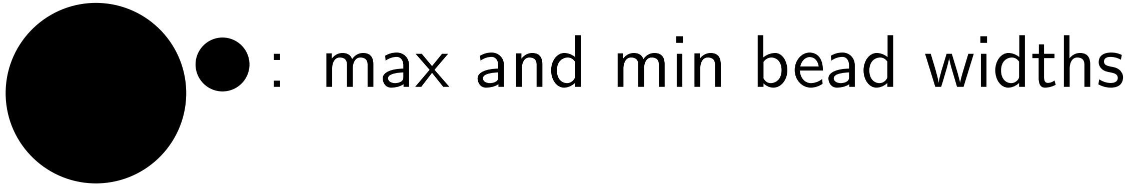
$\downarrow$ trimming

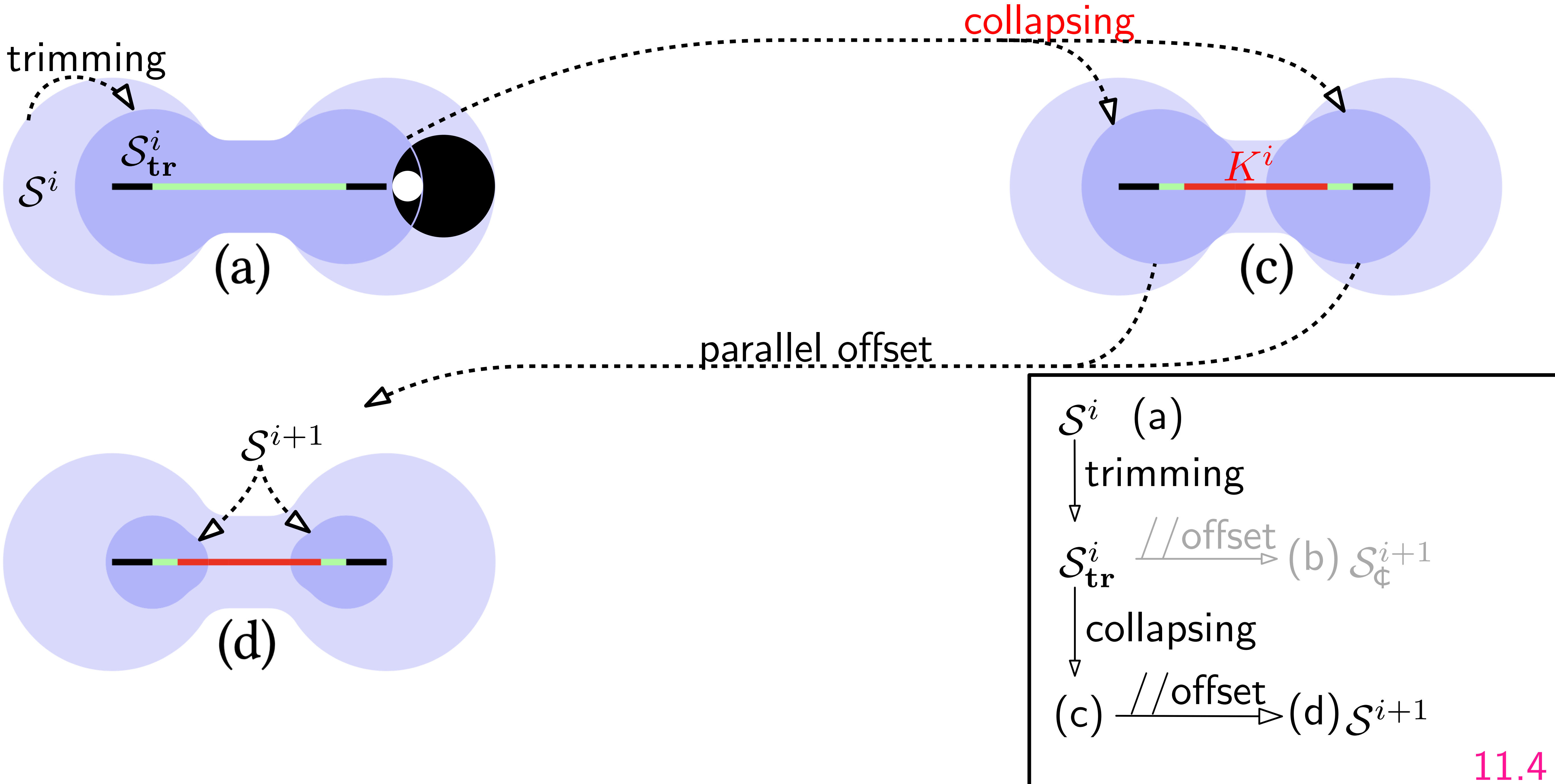
$\mathcal{S}^i_{\mathbf{tr}}$ $\xrightarrow{\text{//offset}}$ (b) $\mathcal{S}^{i+1}_{¢}$

$\downarrow$ collapsing

(c)

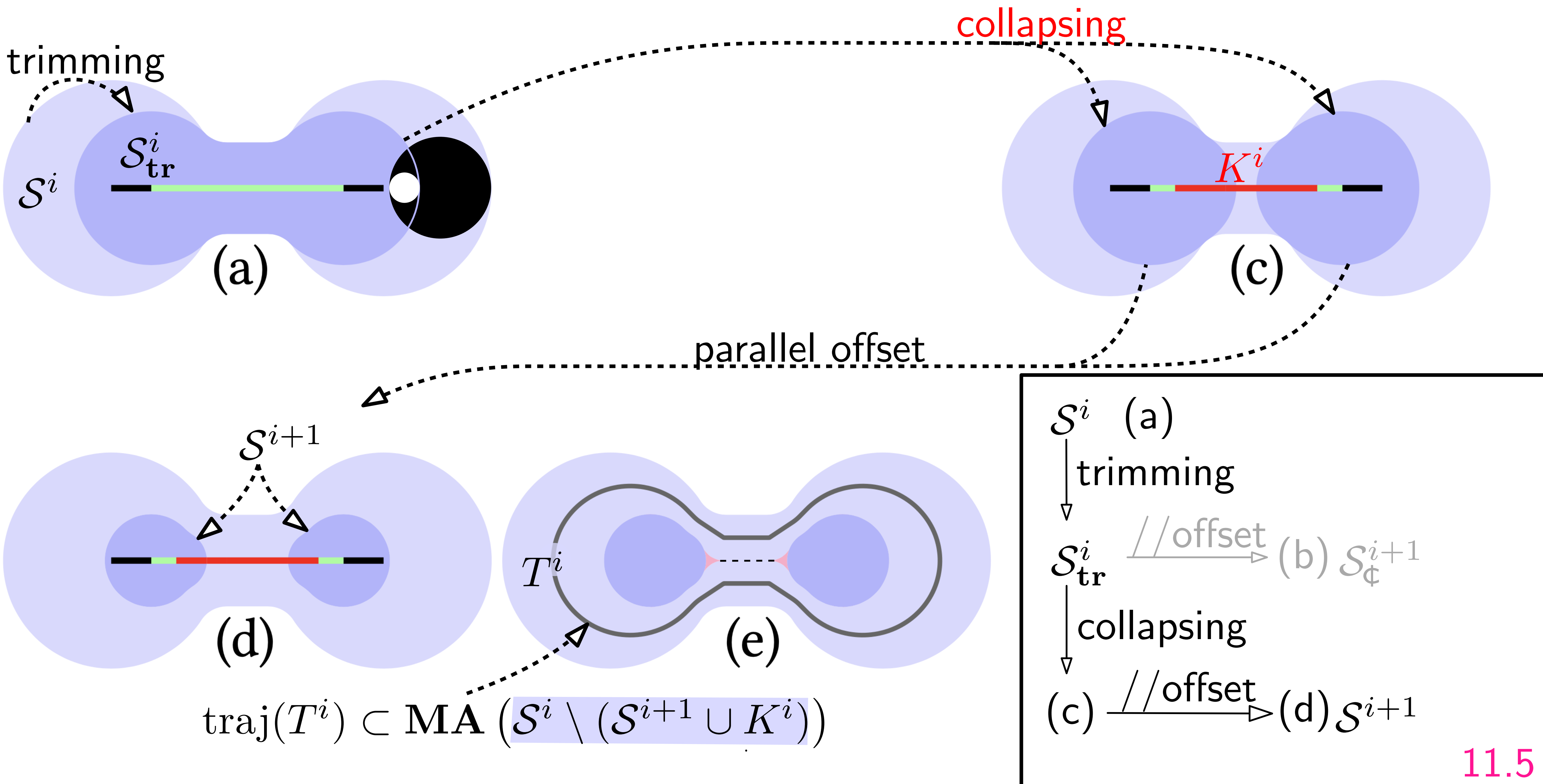
Complete picture with Collapsing



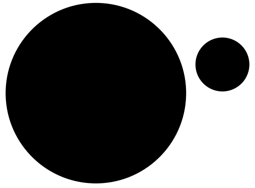


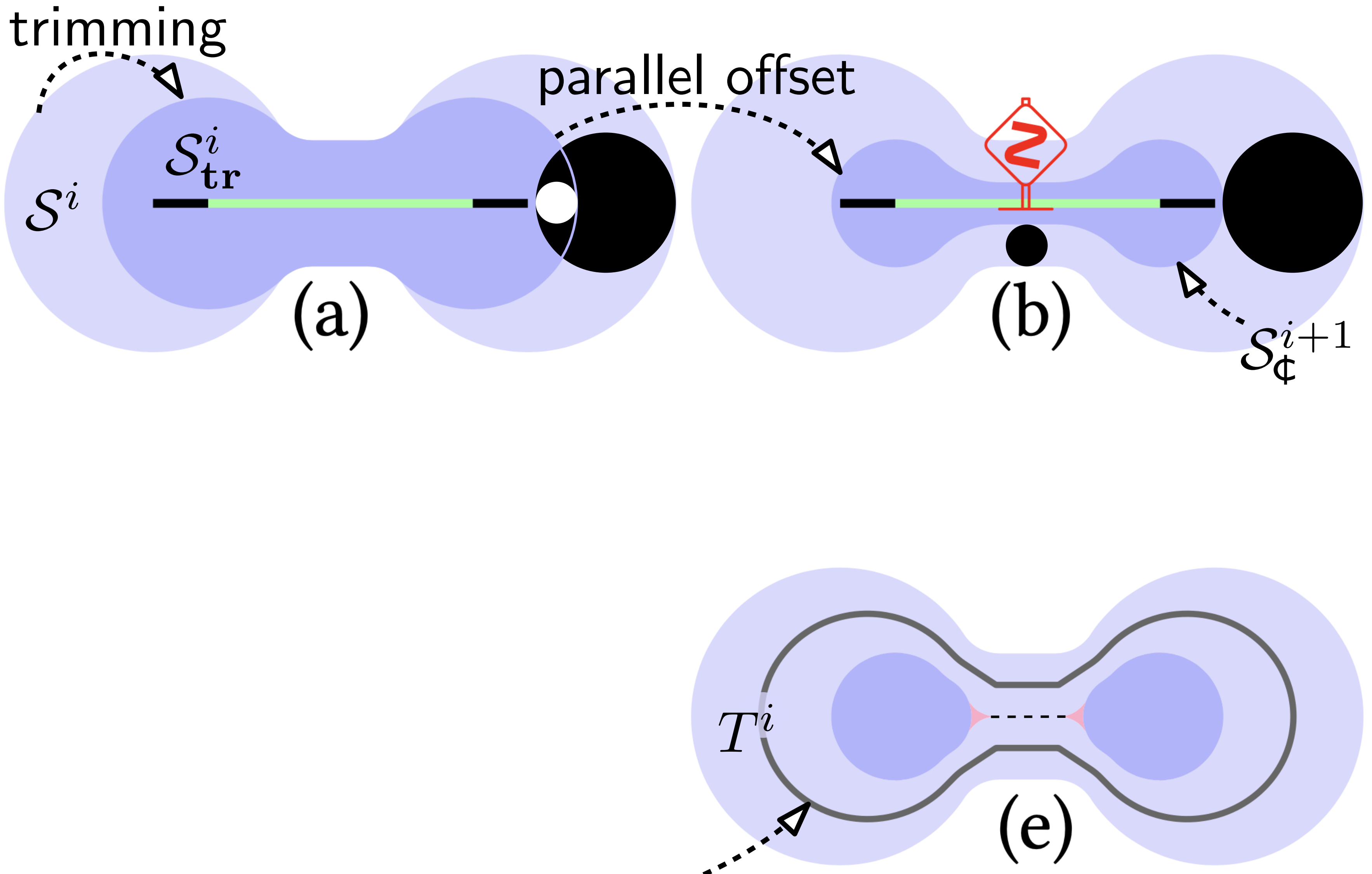
Complete picture with Collapsing

: max and min bead widths



Complete picture with Collapsing


: max and min bead widths



$$\mathrm{traj}(T^i) \subset \mathbf{MA}\left(\mathcal{S}^i \setminus (\mathcal{S}^{i+1} \cup K^i)\right)$$

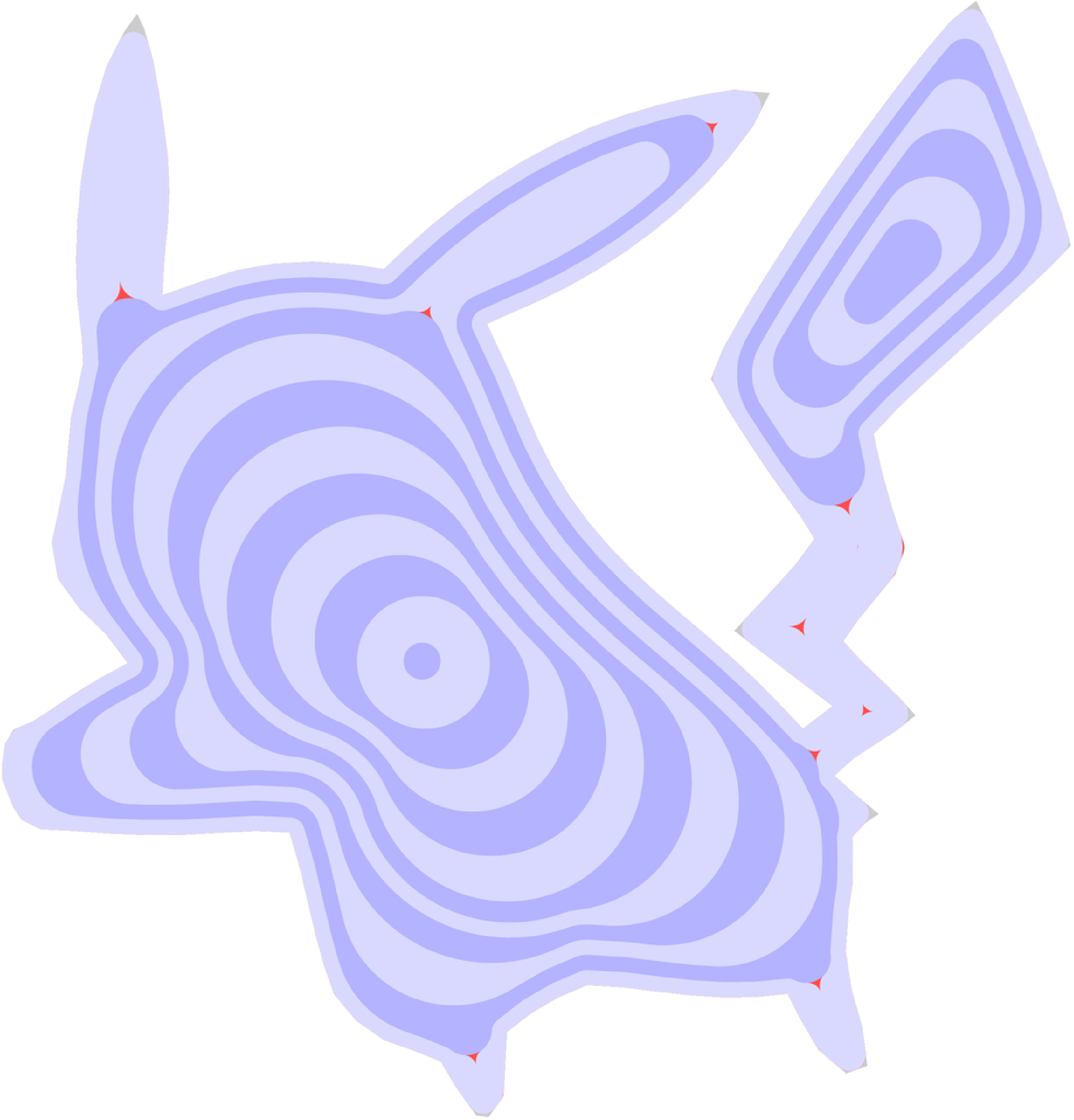
$\mathcal{S}^i$ (a)

↓ trimming

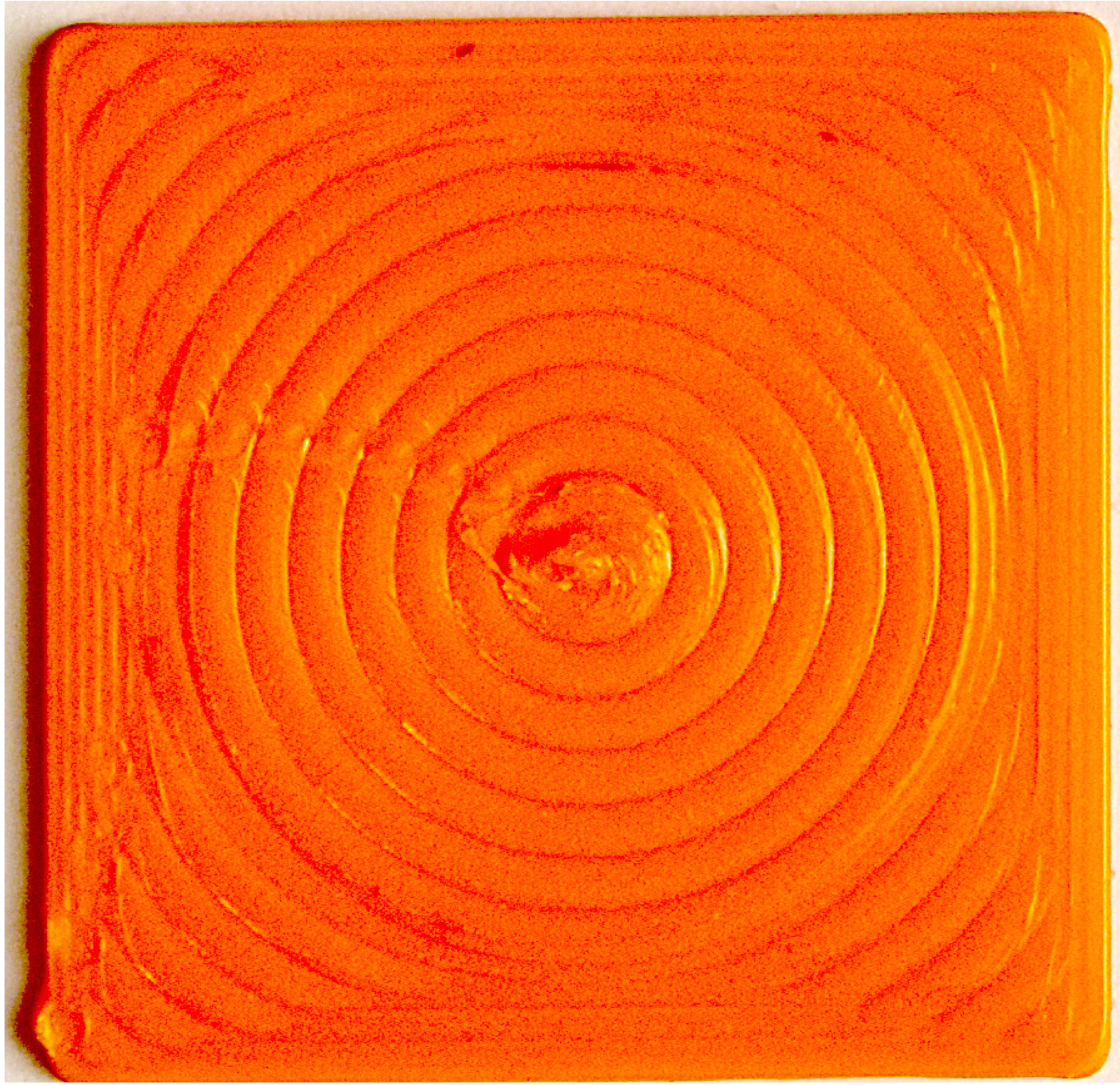
$\mathcal{S}^i_{\mathbf{tr}}$ $\xrightarrow{//\text{offset}}$ (b) $\mathcal{S}^{i+1}_{¢}$

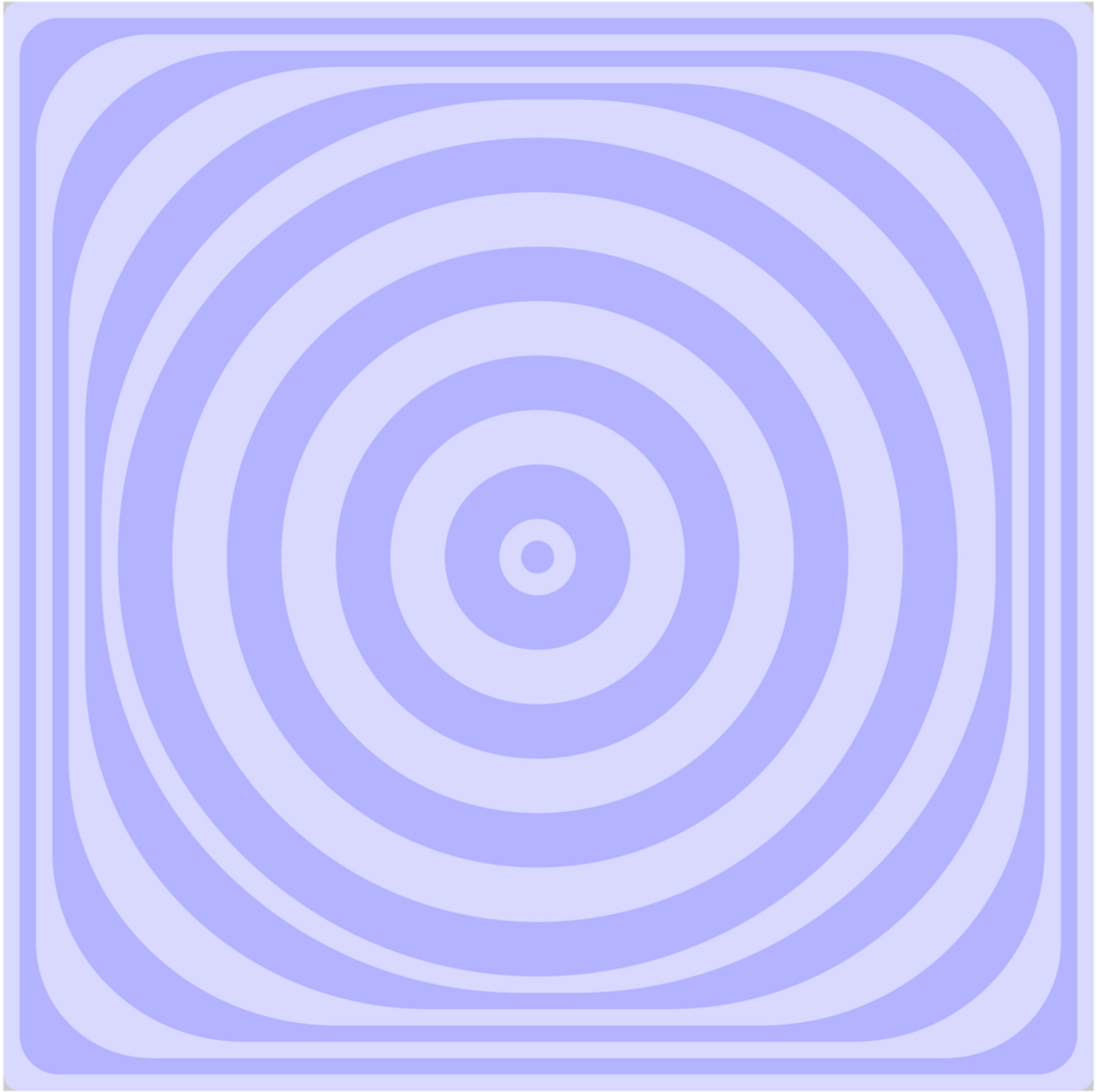
↓ collapsing

(c) $\xrightarrow{//\text{offset}}$ (d) $\mathcal{S}^{i+1}$

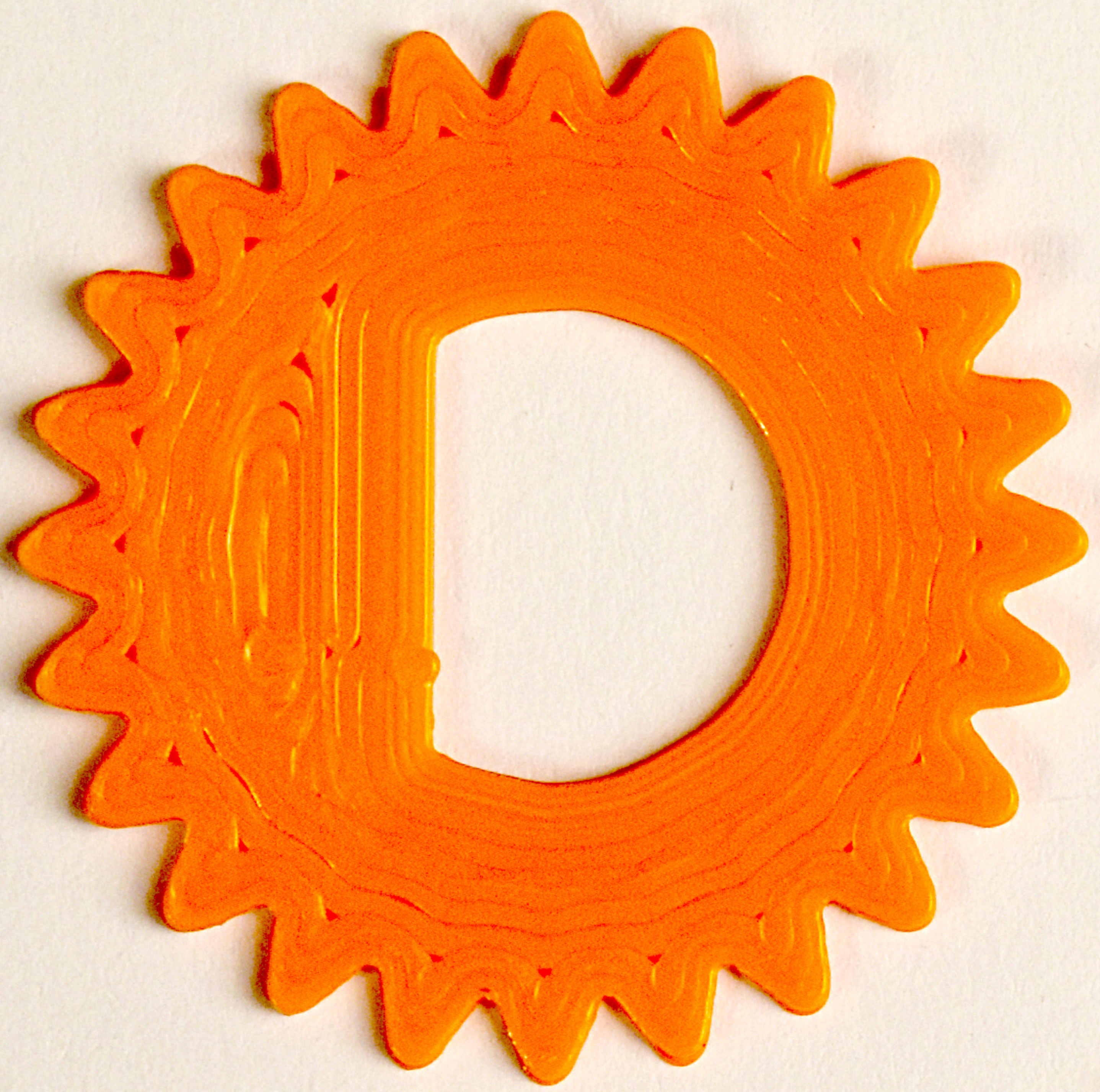


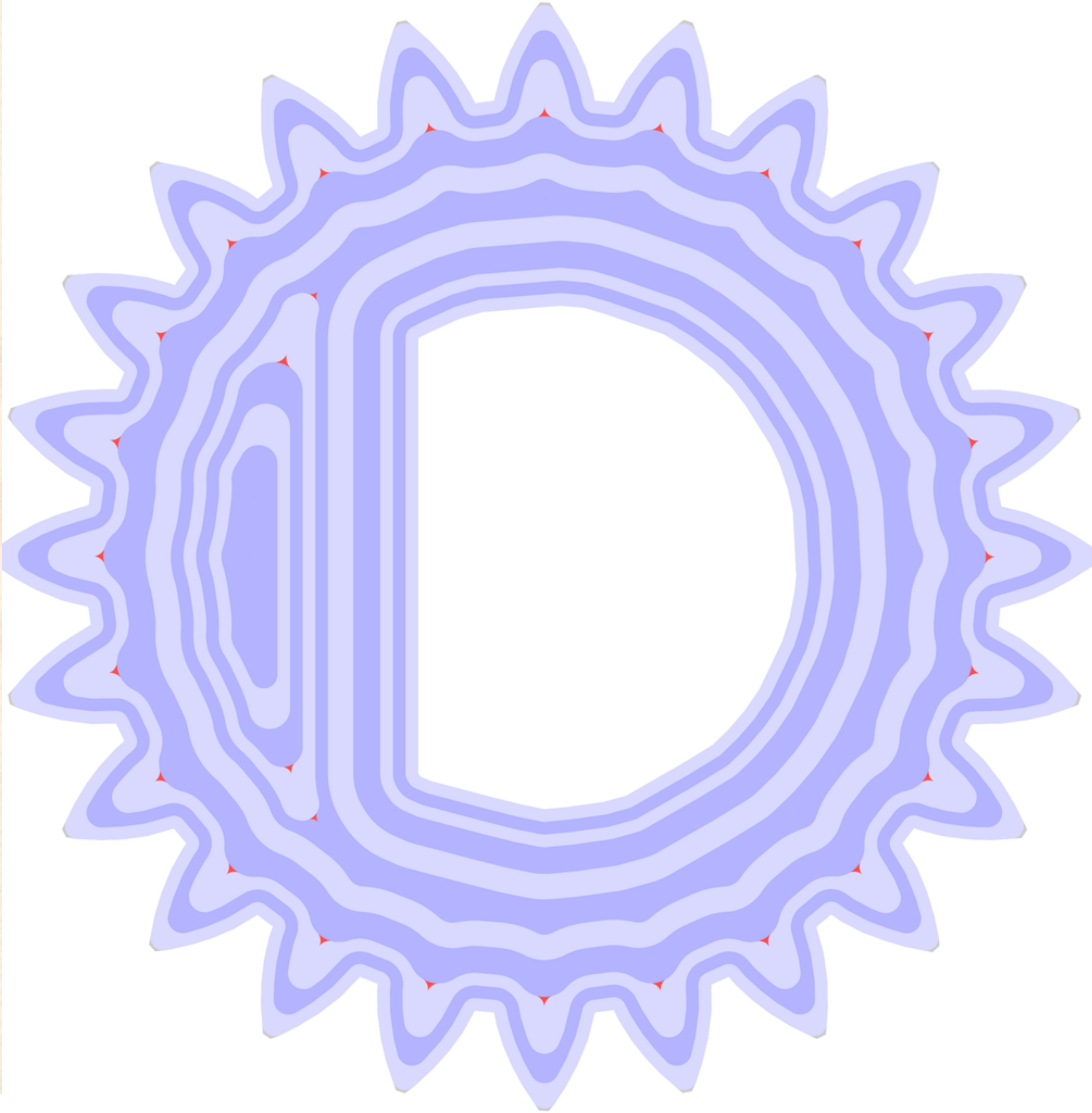
Pictures



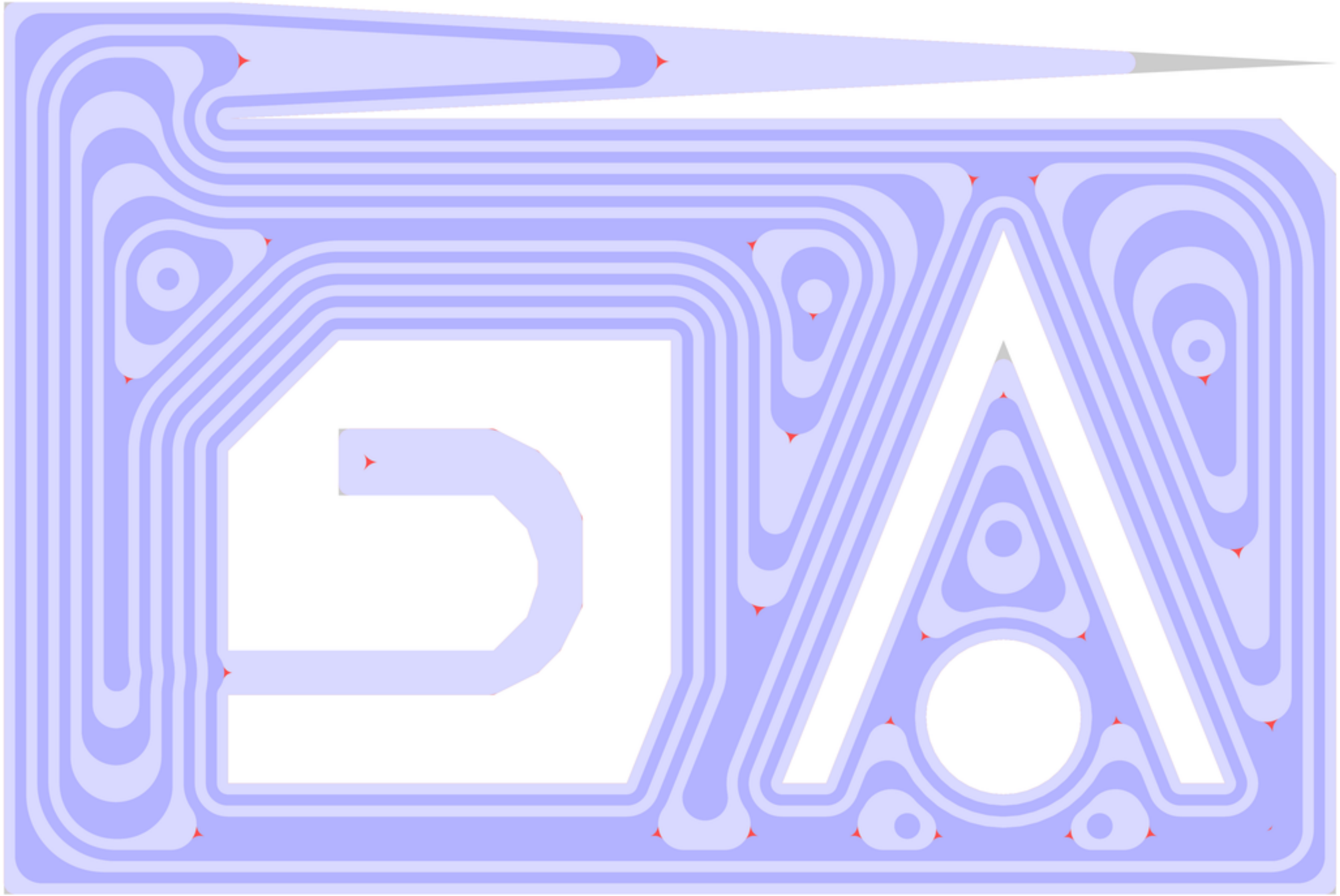


Pictures





Pictures



End of this presentation

See the paper for more, including:

- Less underfill with shaving
- An algorithm for sampling the print path (the center curve of each bead)
- A comparison with the state of the art (almost 10x less underfill)
- A proof of the absence of overfill
- More pictures of fabricated layers

Code: https://github.com/mfx-inria/Variable-width-contouring