Approximating Largest Convex Hulls for Imprecise Points

Maarten Löffler Marc van Kreveld

Center for Geometry, Imaging and Virtual Environments

Utrecht University

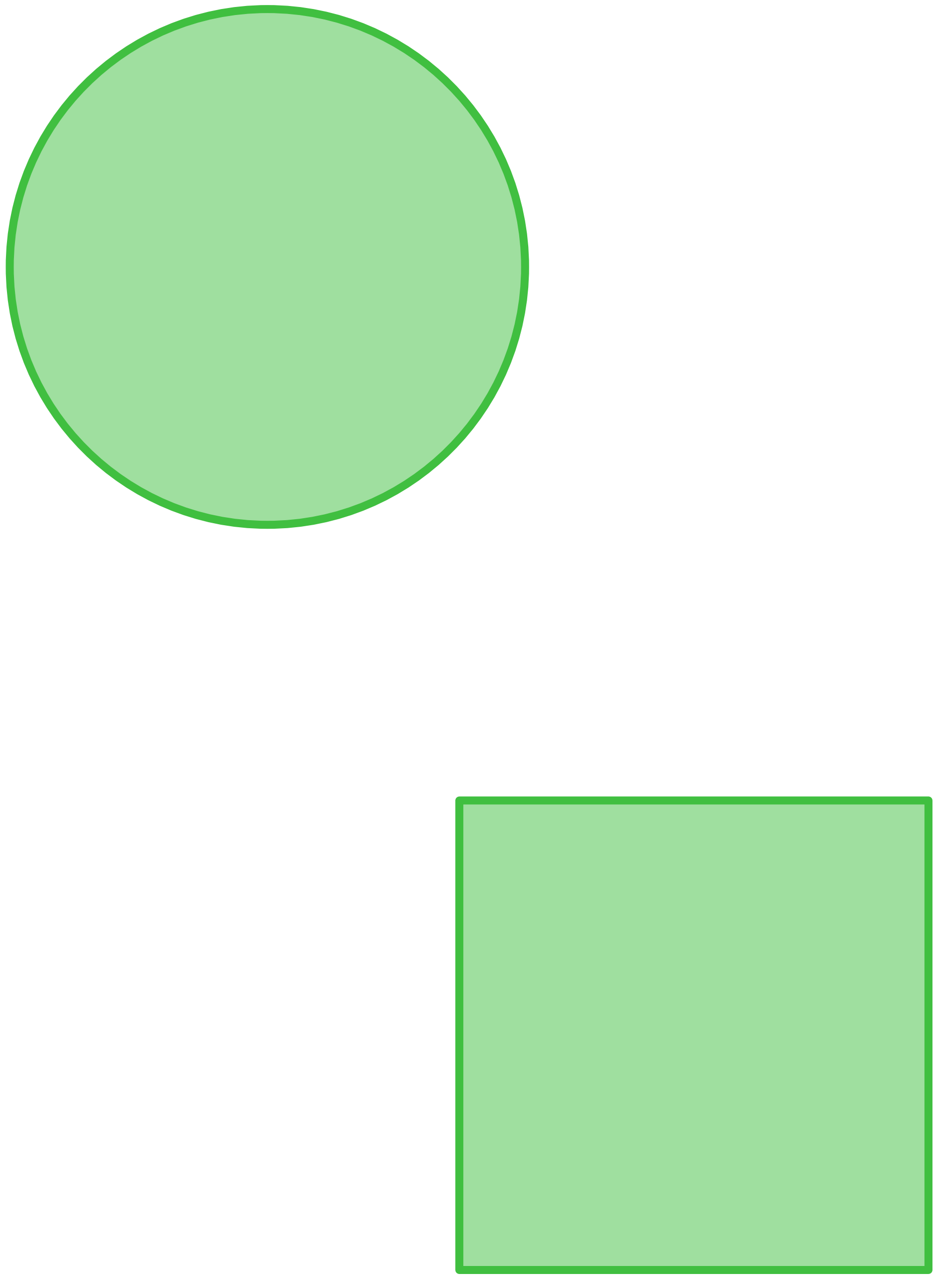
Overview

- Introduction
  - Data Imprecision
  - Convex Hulls
  - Problem Statement
  - Exact Time Bounds
- Approximation
  - Core-Sets
  - Approximation Time Bounds
  - Algorithms
- Concluding remarks

Imprecision

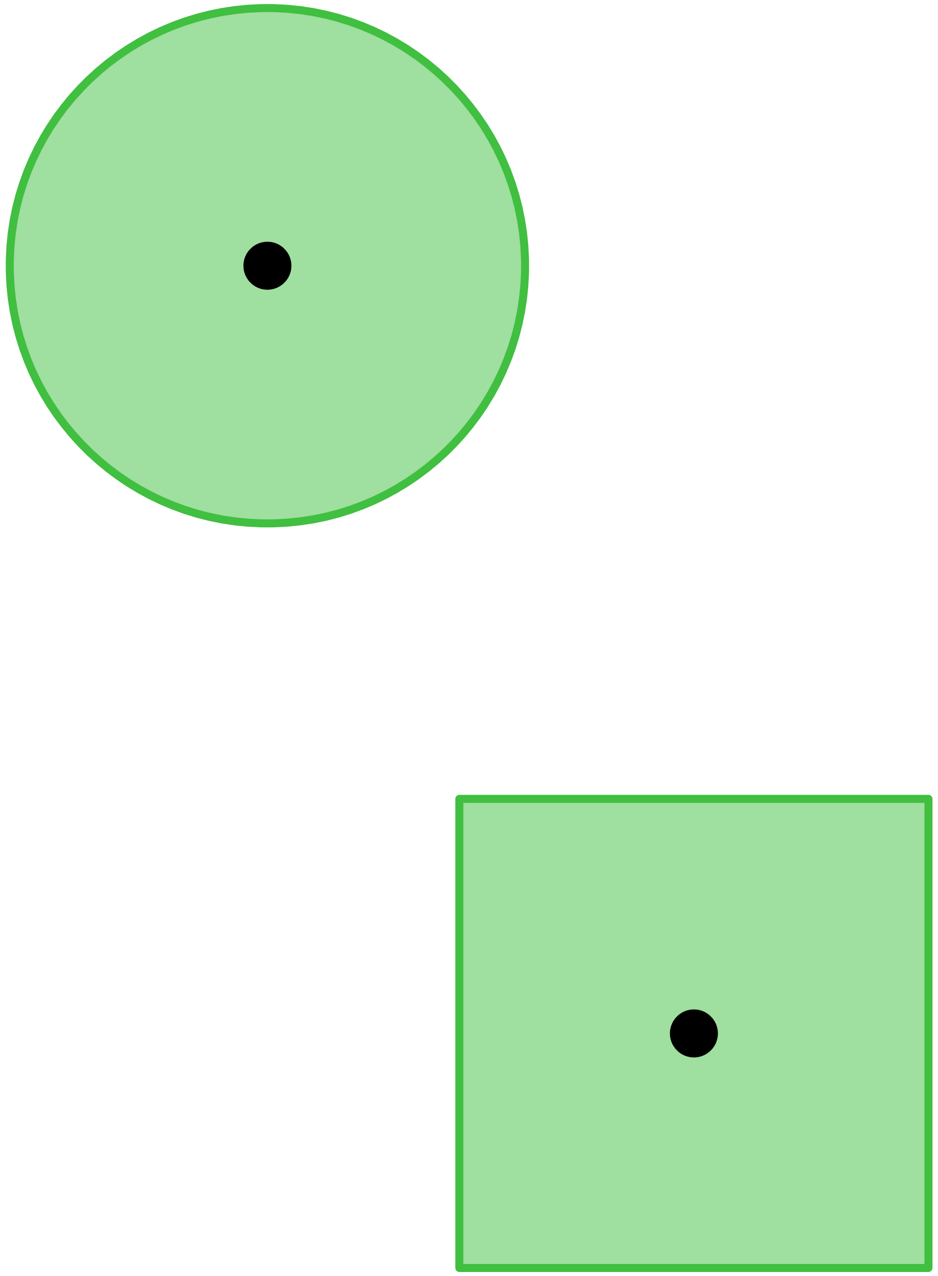
- Traditional algorithms assume exact input
- In practice, input data is often not exact
  - Measured from the real world
  - Stored with limited precision
  - Computed by inexact algorithms
- Output of traditional algorithms is unreliable
- Given exact description of input imprecision, we can exactly predict output imprecision

Imprecise Points



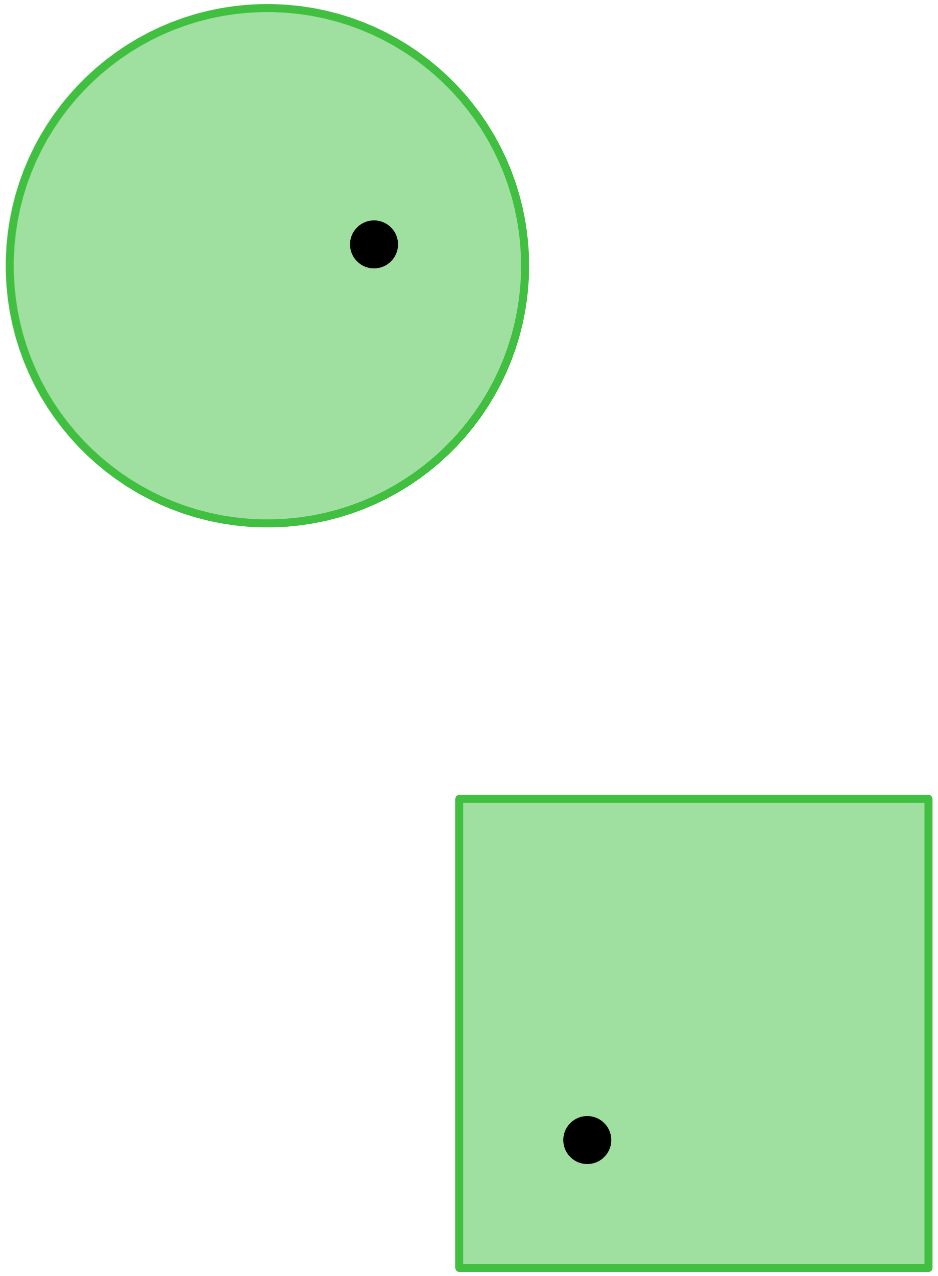
- Unknown location
- Known region of possible locations
- Regions are simple geometric objects
  - Disc
  - Square
  - Rectangle
  - Convex polygon

Imprecise Points



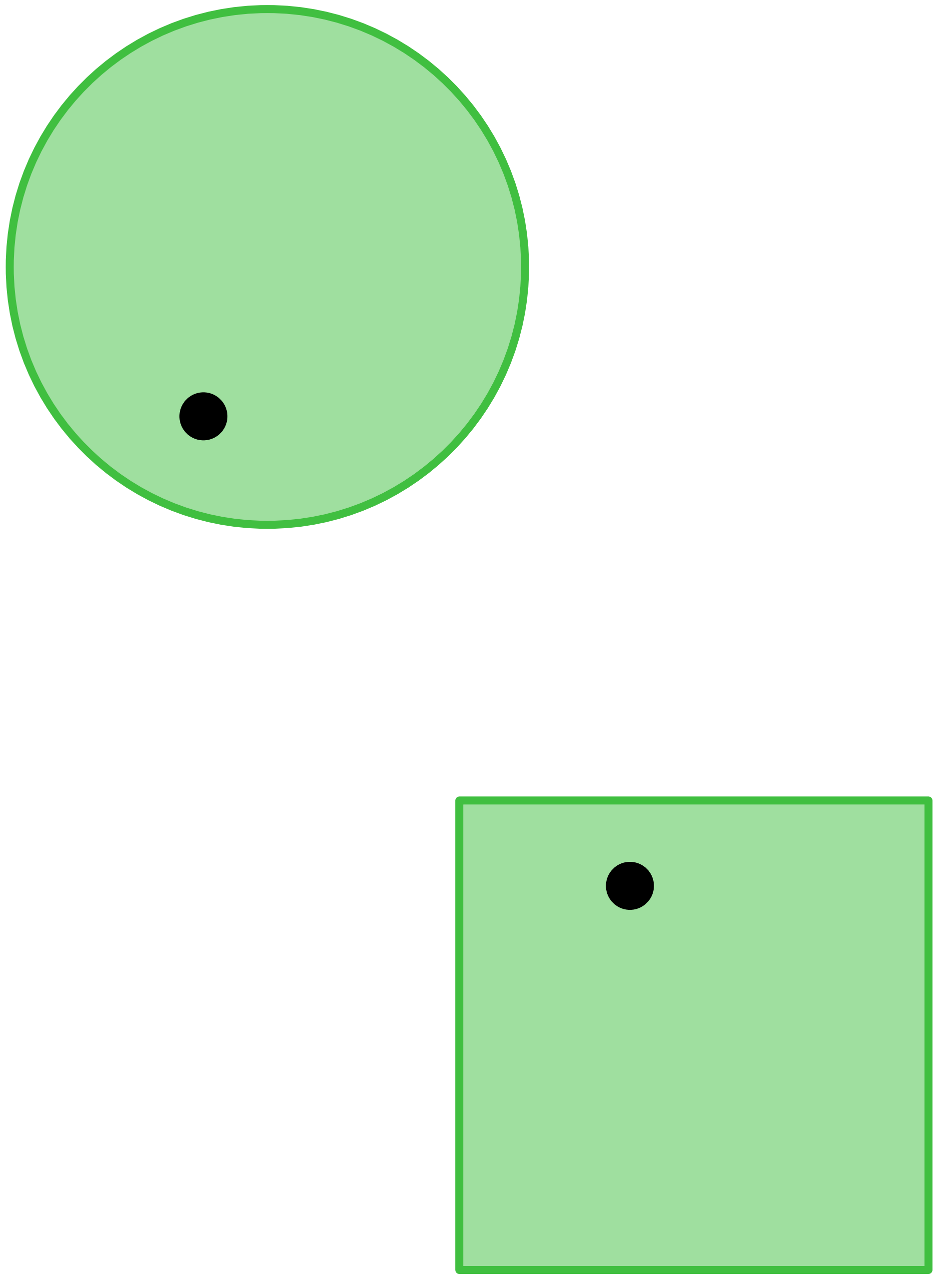
- Unknown location
- Known region of possible locations
- Regions are simple geometric objects
  - Disc
  - Square
  - Rectangle
  - Convex polygon

Imprecise Points



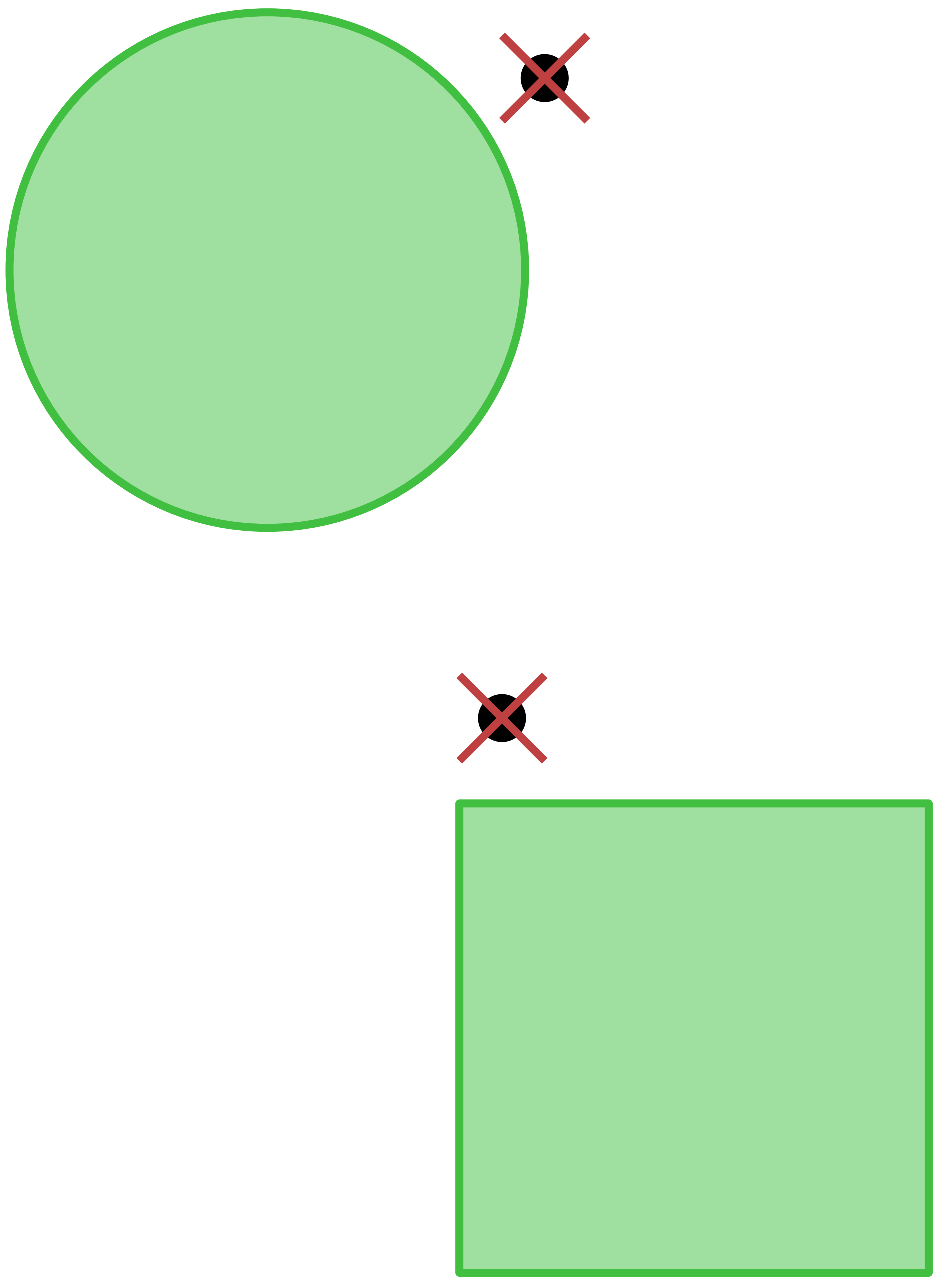
- Unknown location
- Known region of possible locations
- Regions are simple geometric objects
  - Disc
  - Square
  - Rectangle
  - Convex polygon

Imprecise Points



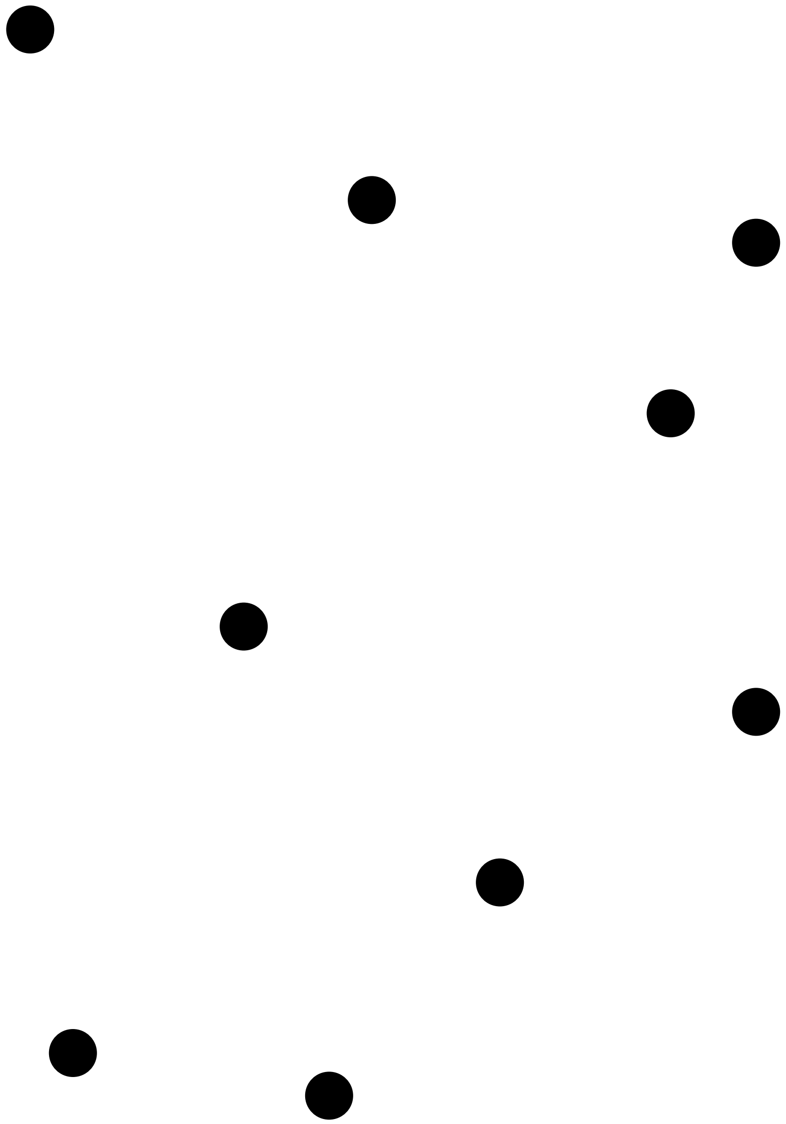
- Unknown location
- Known region of possible locations
- Regions are simple geometric objects
  - Disc
  - Square
  - Rectangle
  - Convex polygon

Imprecise Points



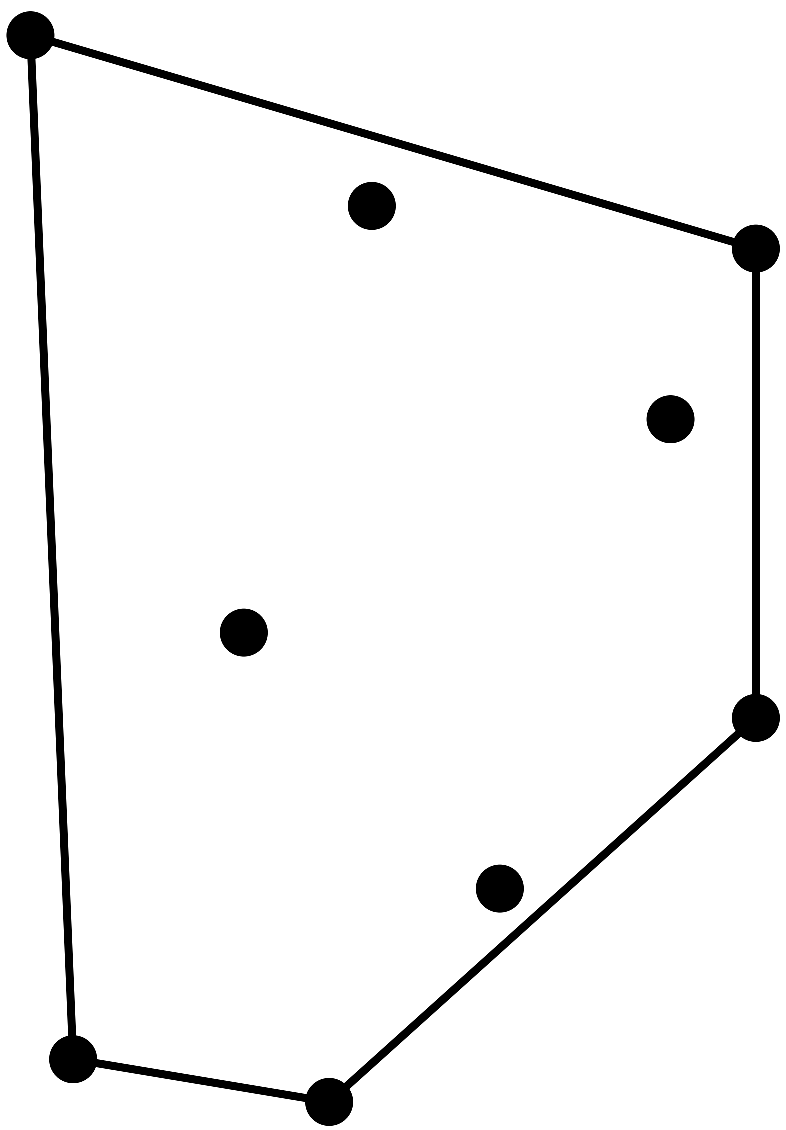
- Unknown location
- Known region of possible locations
- Regions are simple geometric objects
  - Disc
  - Square
  - Rectangle
  - Convex polygon

Convex Hull



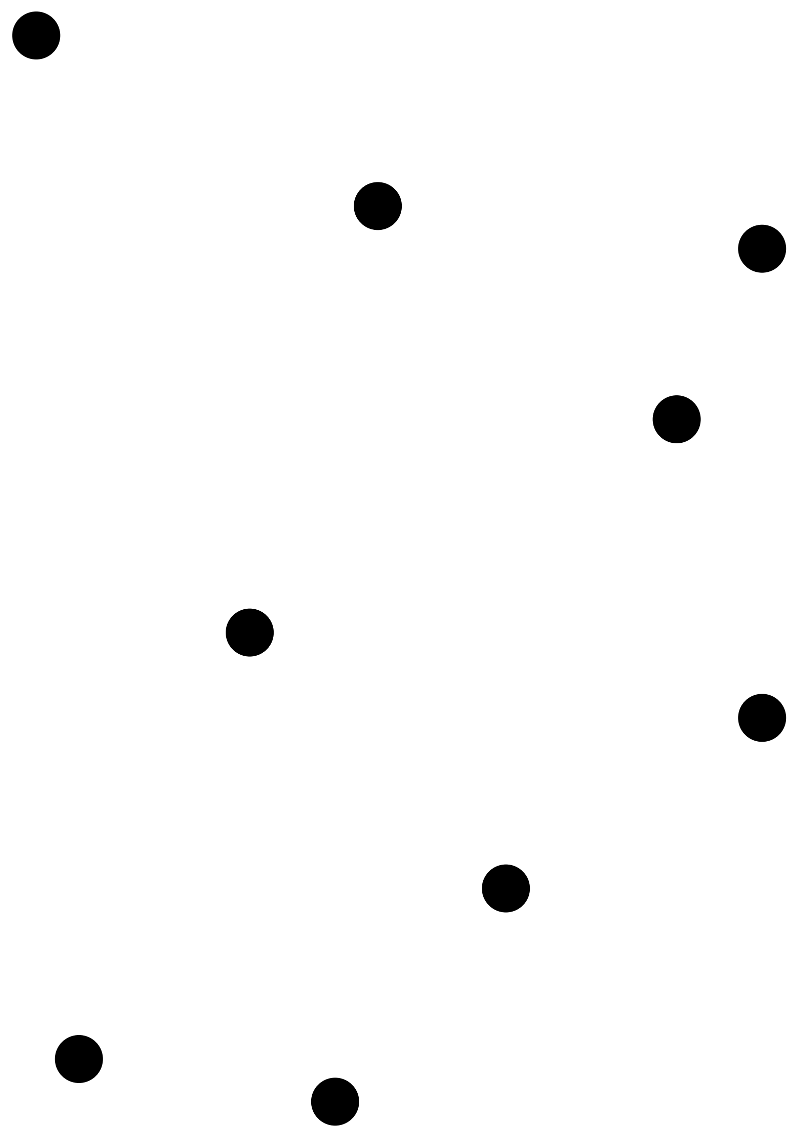
- Given a set P of n points
- Smallest convex set containing P
- Can be computed in $O(n \log n)$ time
- Solved long ago

Convex Hull



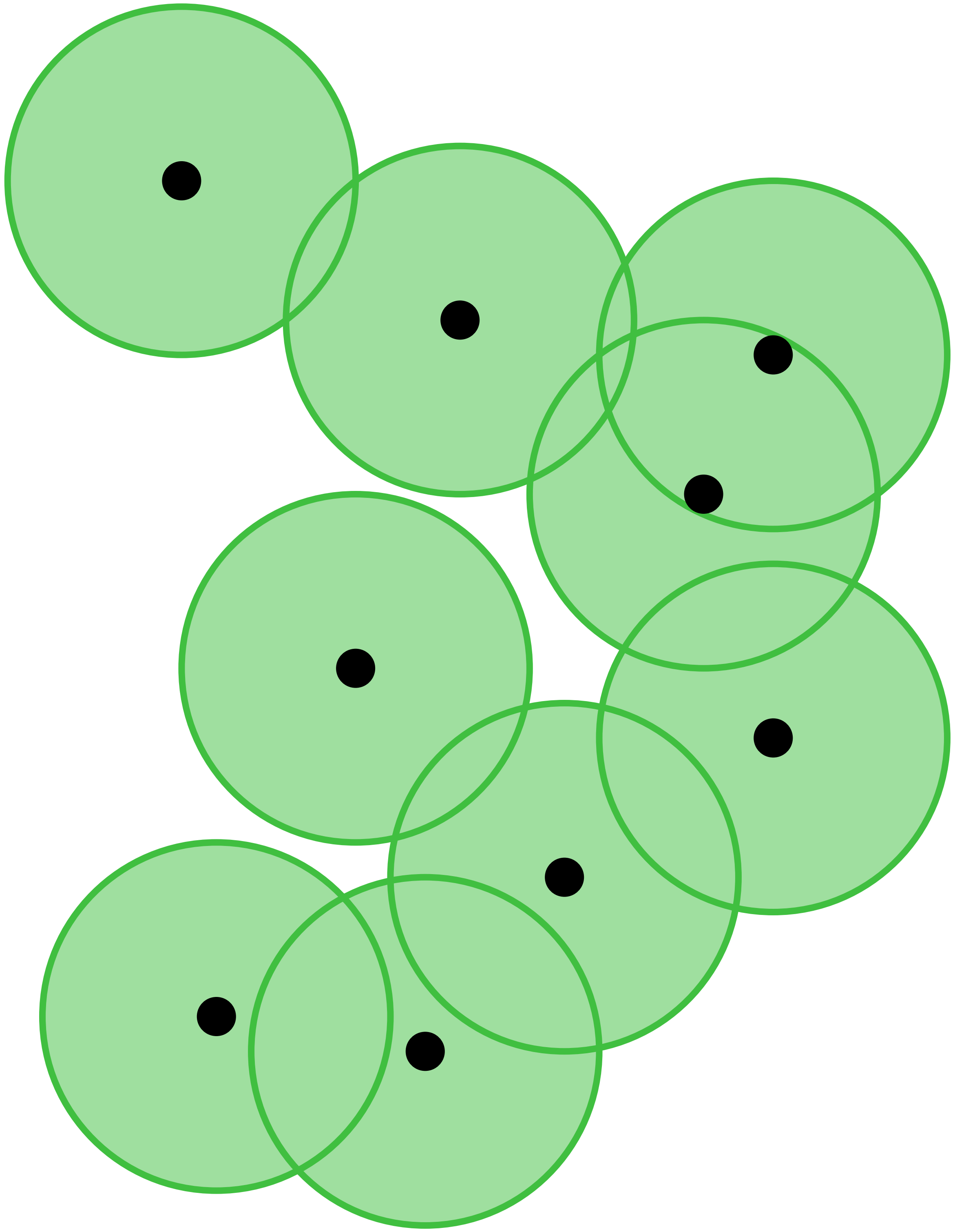
- Given a set P of n points
- Smallest convex set containing P
- Can be computed in $O(n \log n)$ time
- Solved long ago

Imprecise Convex Hull



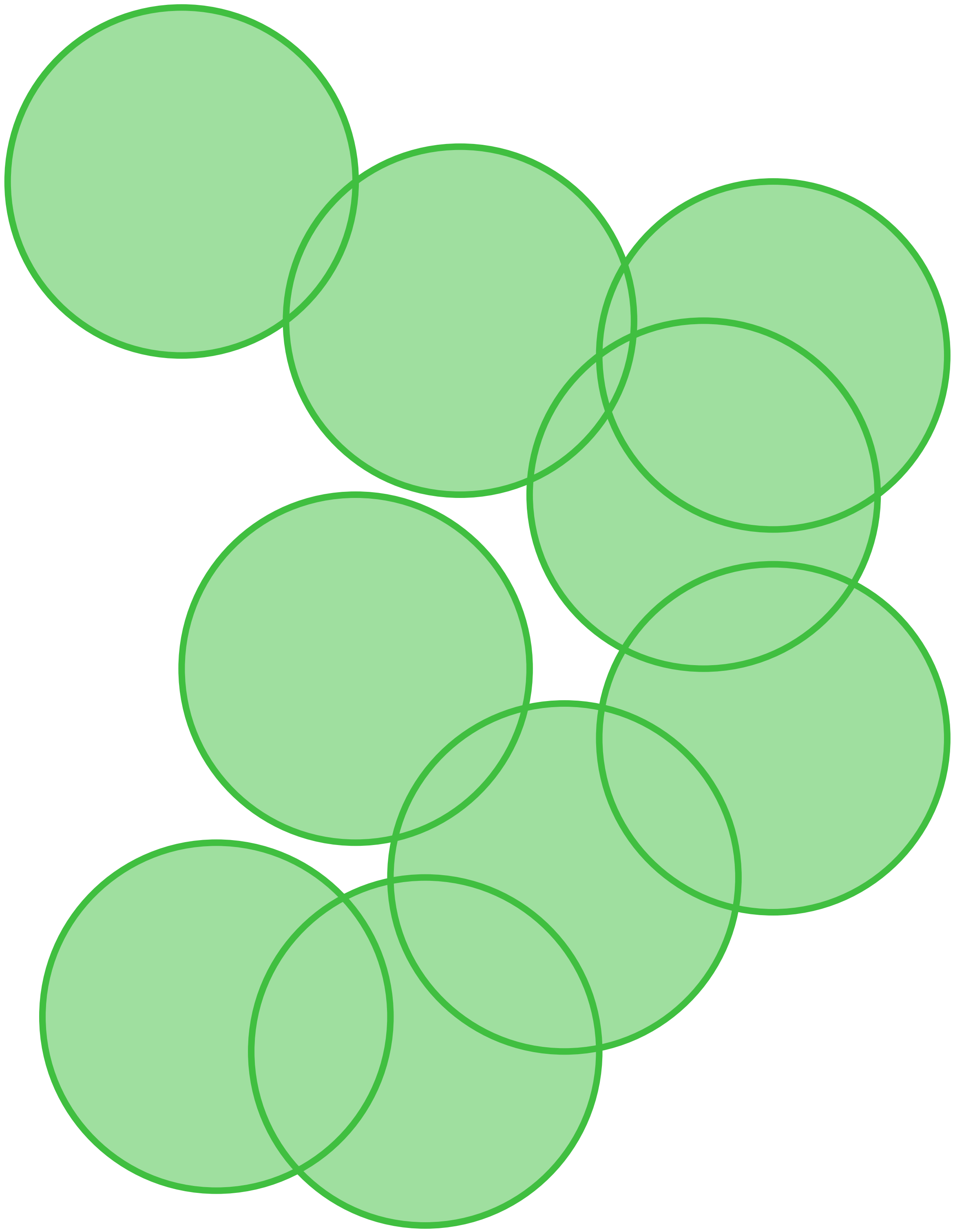
- Given a set $\mathcal{L}$ of n imprecise points
- What is the convex hull?
- Multiple possibilities
- True structure is unknown
- We want to capture the imprecision in the output

Imprecise Convex Hull



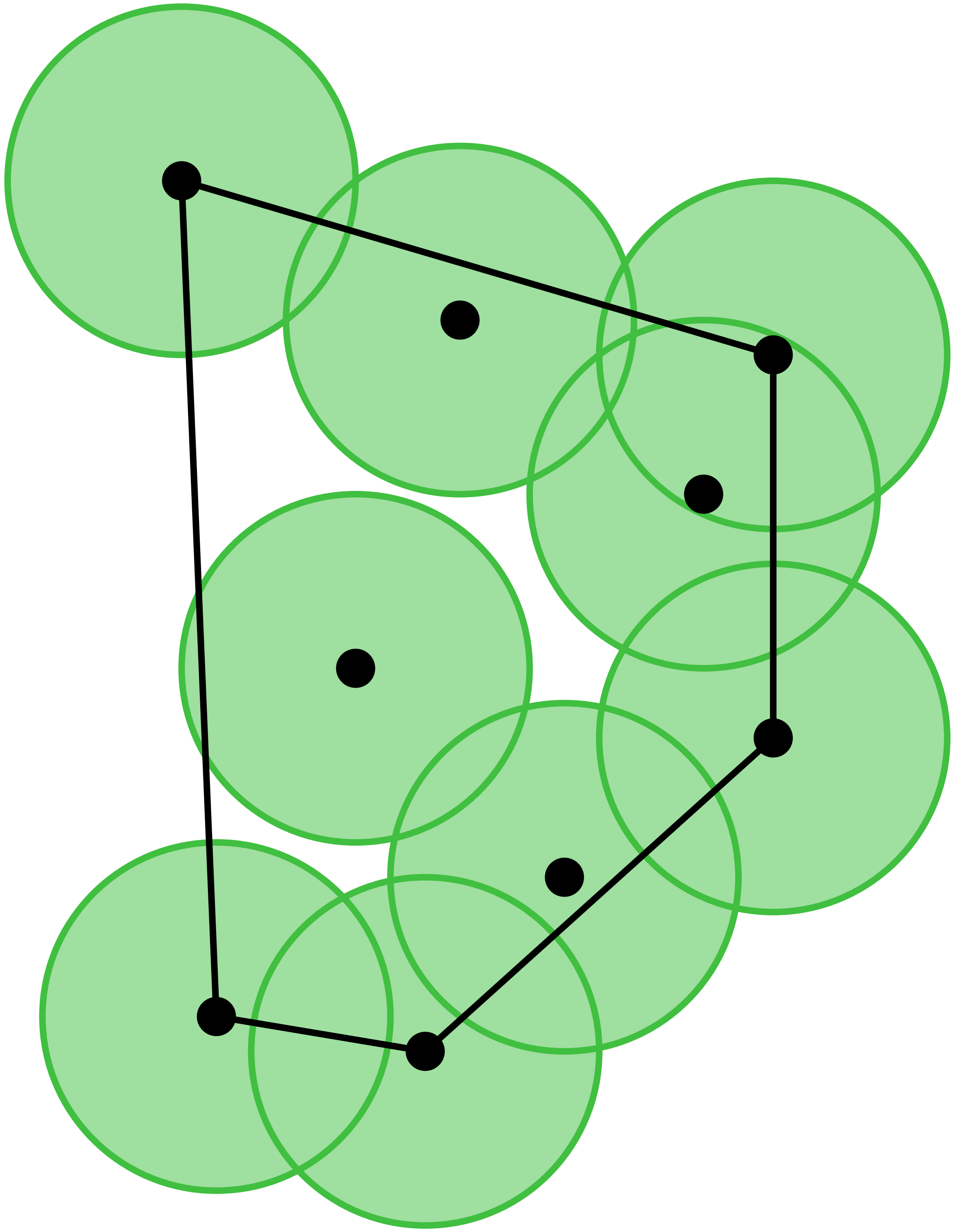
- Given a set $\mathcal{L}$ of n imprecise points
- What is the convex hull?
- Multiple possibilities
- True structure is unknown
- We want to capture the imprecision in the output

Imprecise Convex Hull



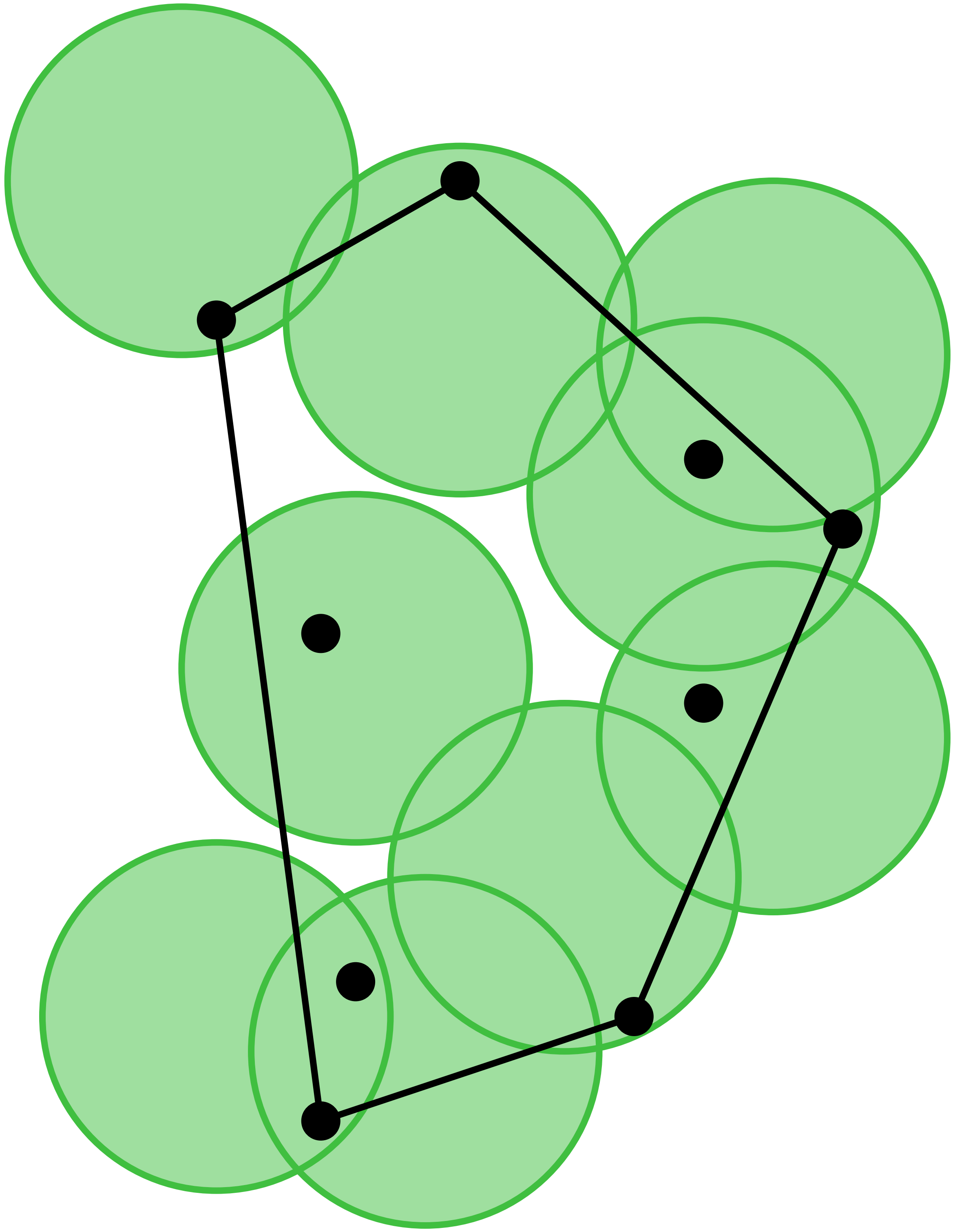
- Given a set $\mathcal{L}$ of n imprecise points
- What is the convex hull?
- Multiple possibilities
- True structure is unknown
- We want to capture the imprecision in the output

Imprecise Convex Hull



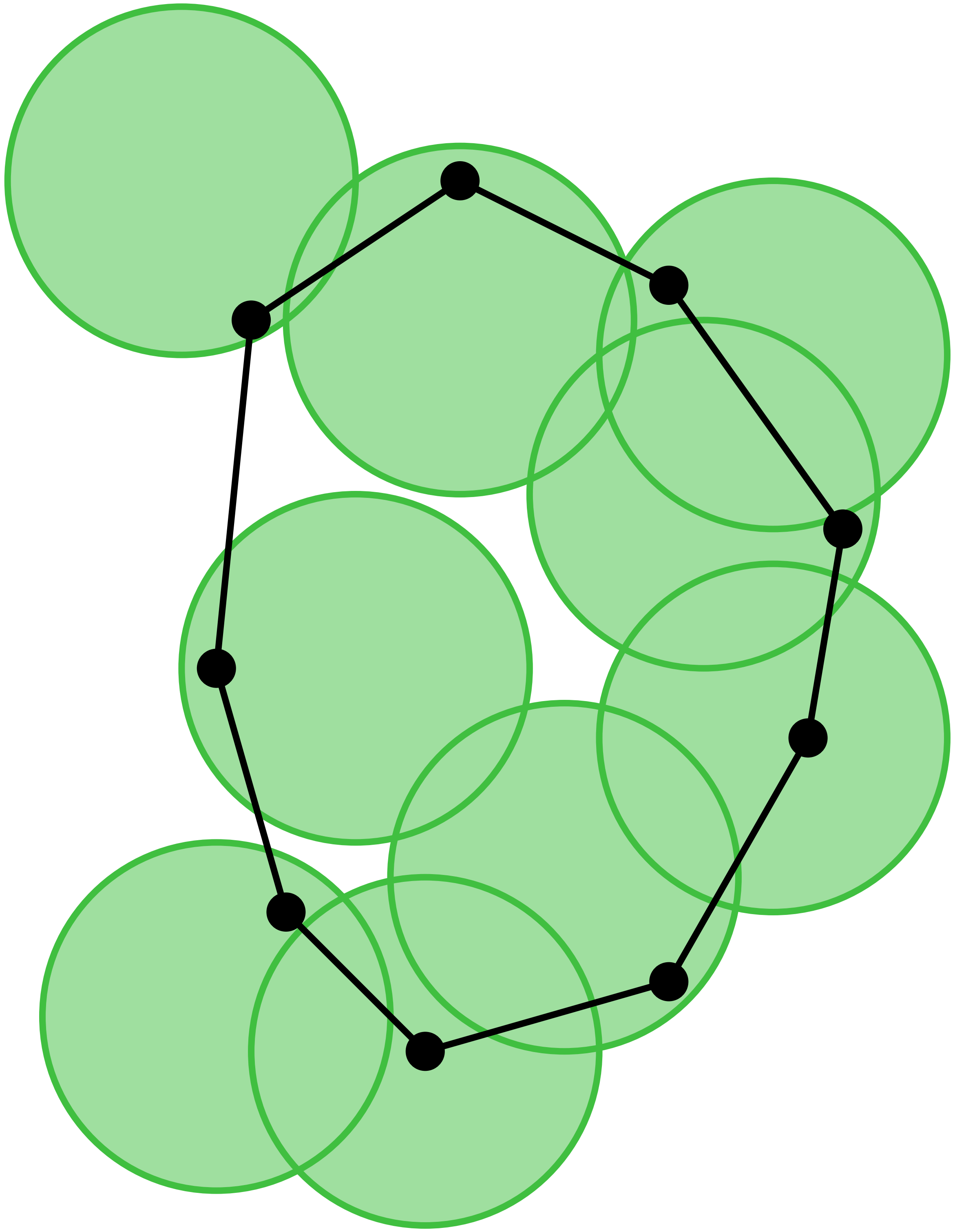
- Given a set $\mathcal{L}$ of n imprecise points
- What is the convex hull?
- Multiple possibilities
- True structure is unknown
- We want to capture the imprecision in the output

Imprecise Convex Hull



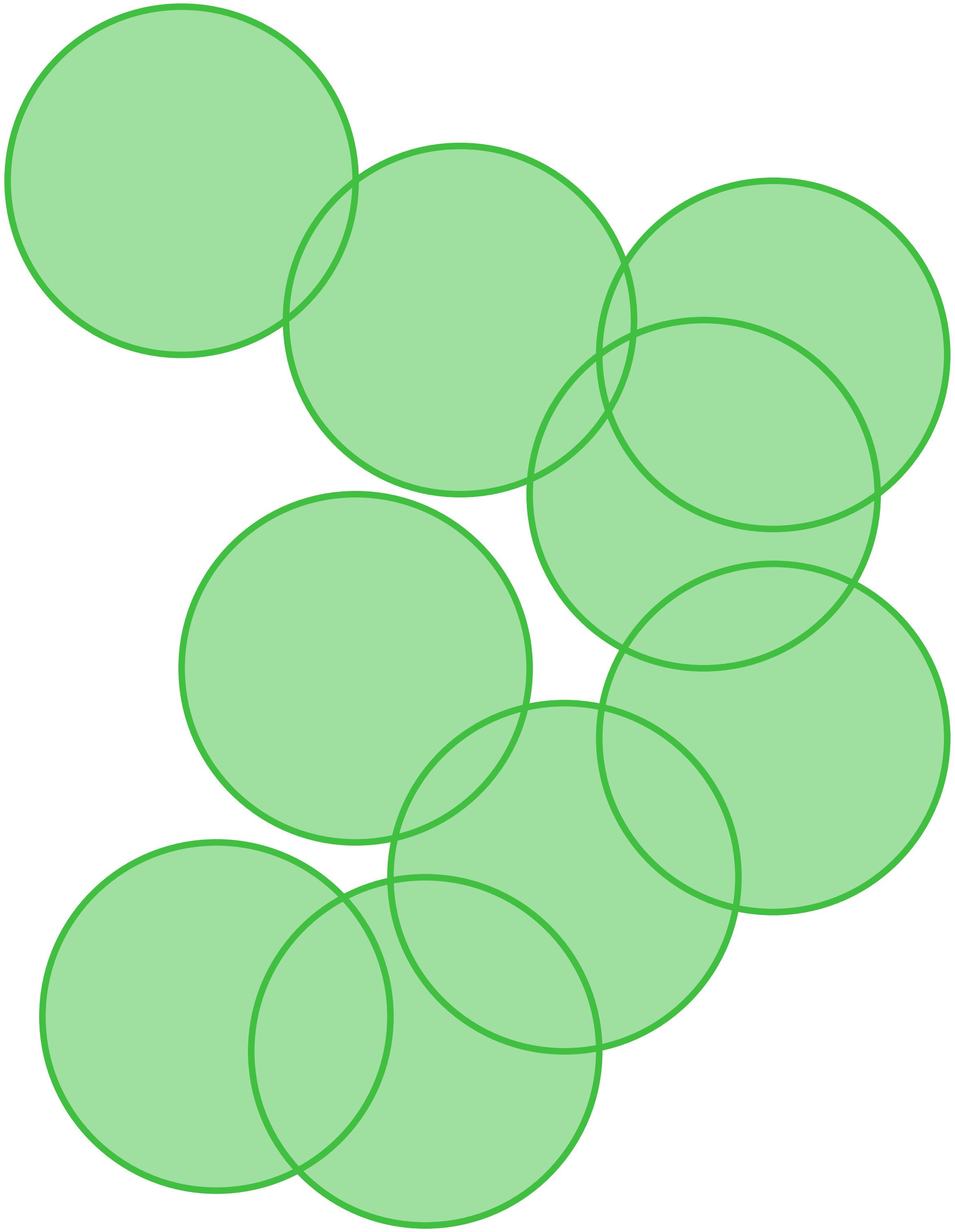
- Given a set $\mathcal{L}$ of n imprecise points
- What is the convex hull?
- Multiple possibilities
- True structure is unknown
- We want to capture the imprecision in the output

Imprecise Convex Hull



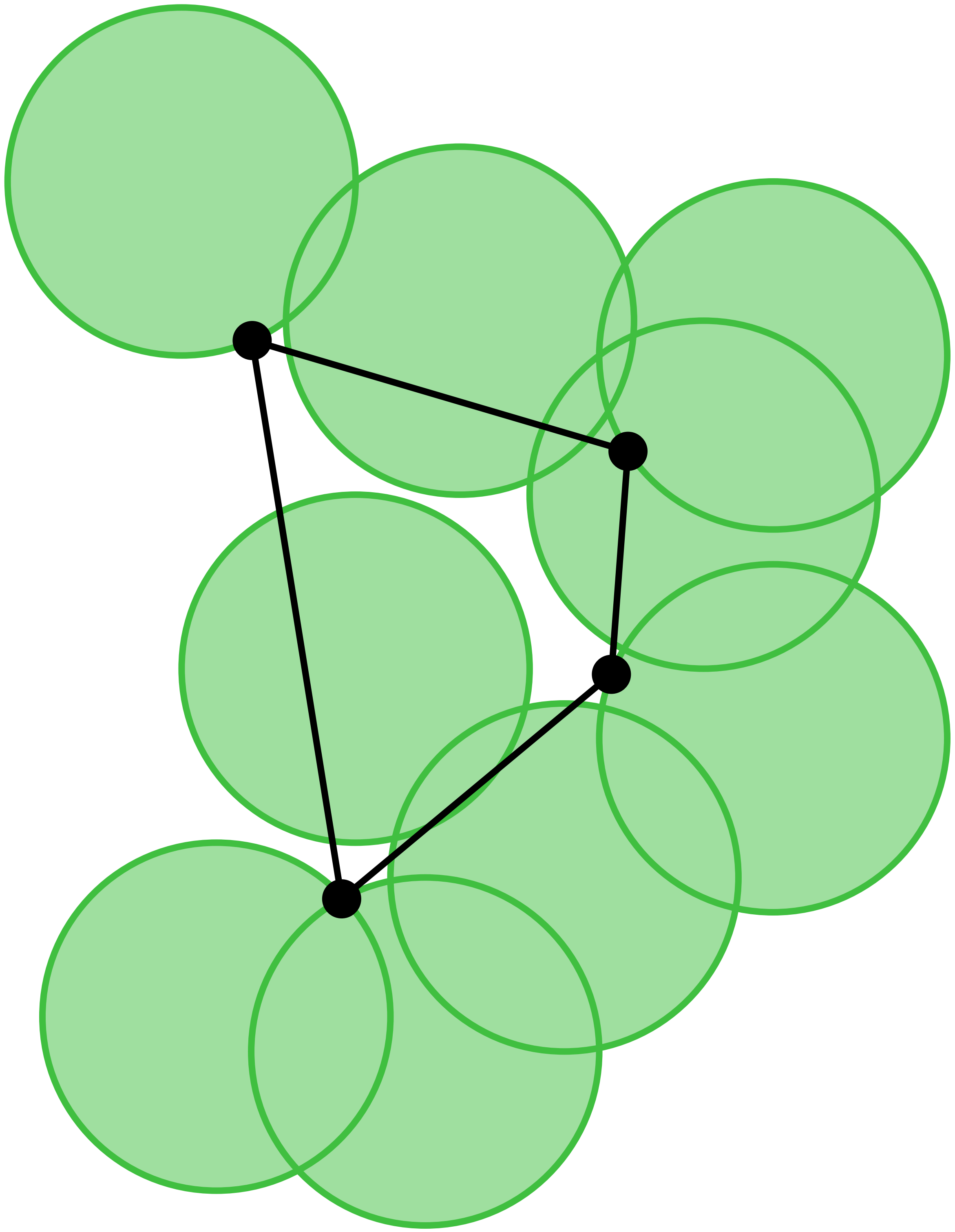
- Given a set $\mathcal{L}$ of n imprecise points
- What is the convex hull?
- Multiple possibilities
- True structure is unknown
- We want to capture the imprecision in the output

Bounds on Measures



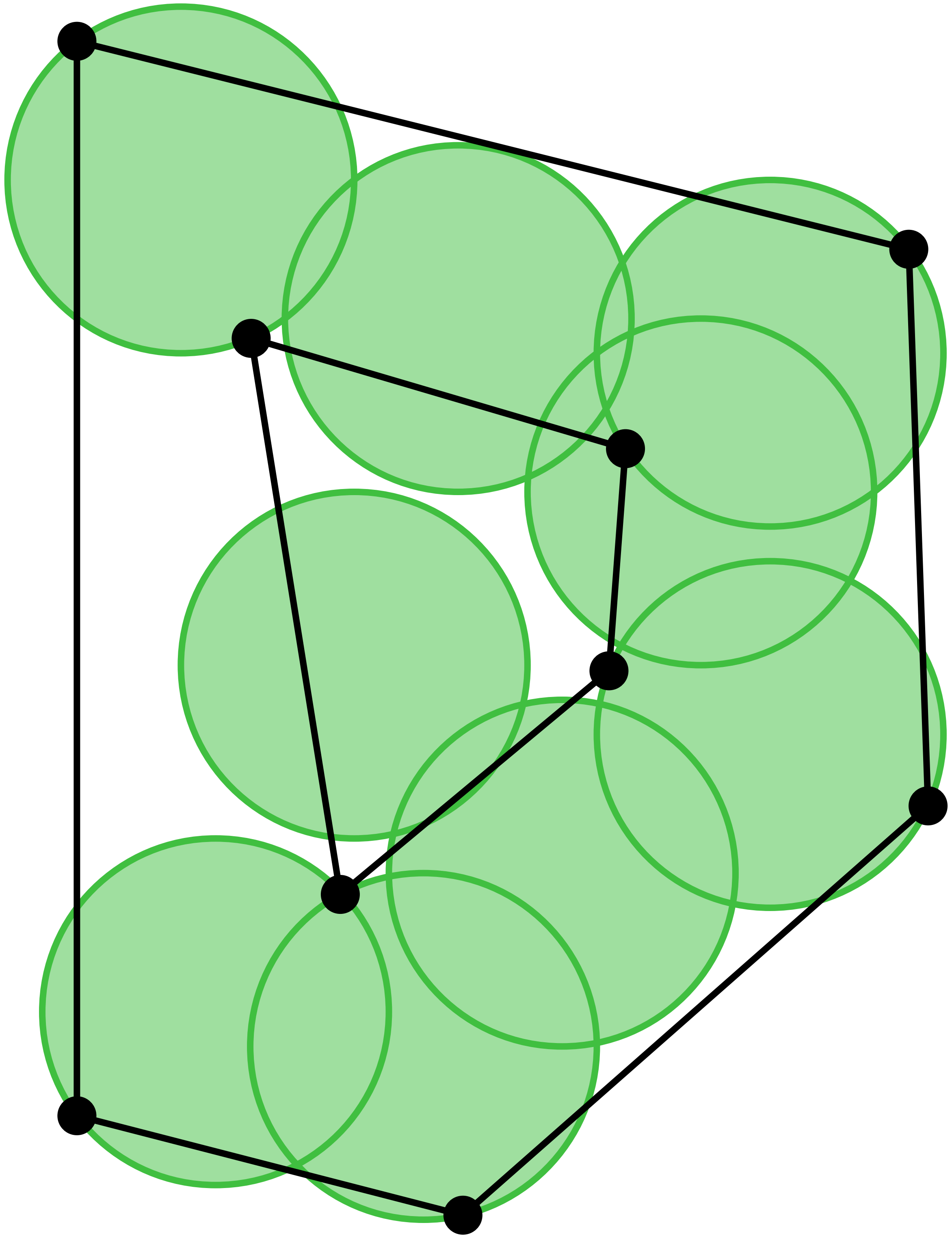
- If we measure the *size* the convex hull . . .
  - Area
  - Perimeter
  - Any other measure
- . . . we can compute *bounds* on the possible values

Bounds on Measures



- If we measure the *size* the convex hull . . .
  - Area
  - Perimeter
  - Any other measure
- . . . we can compute *bounds* on the possible values
  - Smallest convex hull

Bounds on Measures



- If we measure the *size* the convex hull . . .
  - Area
  - Perimeter
  - Any other measure
- . . . we can compute *bounds* on the possible values
  - Smallest convex hull
  - Largest convex hull

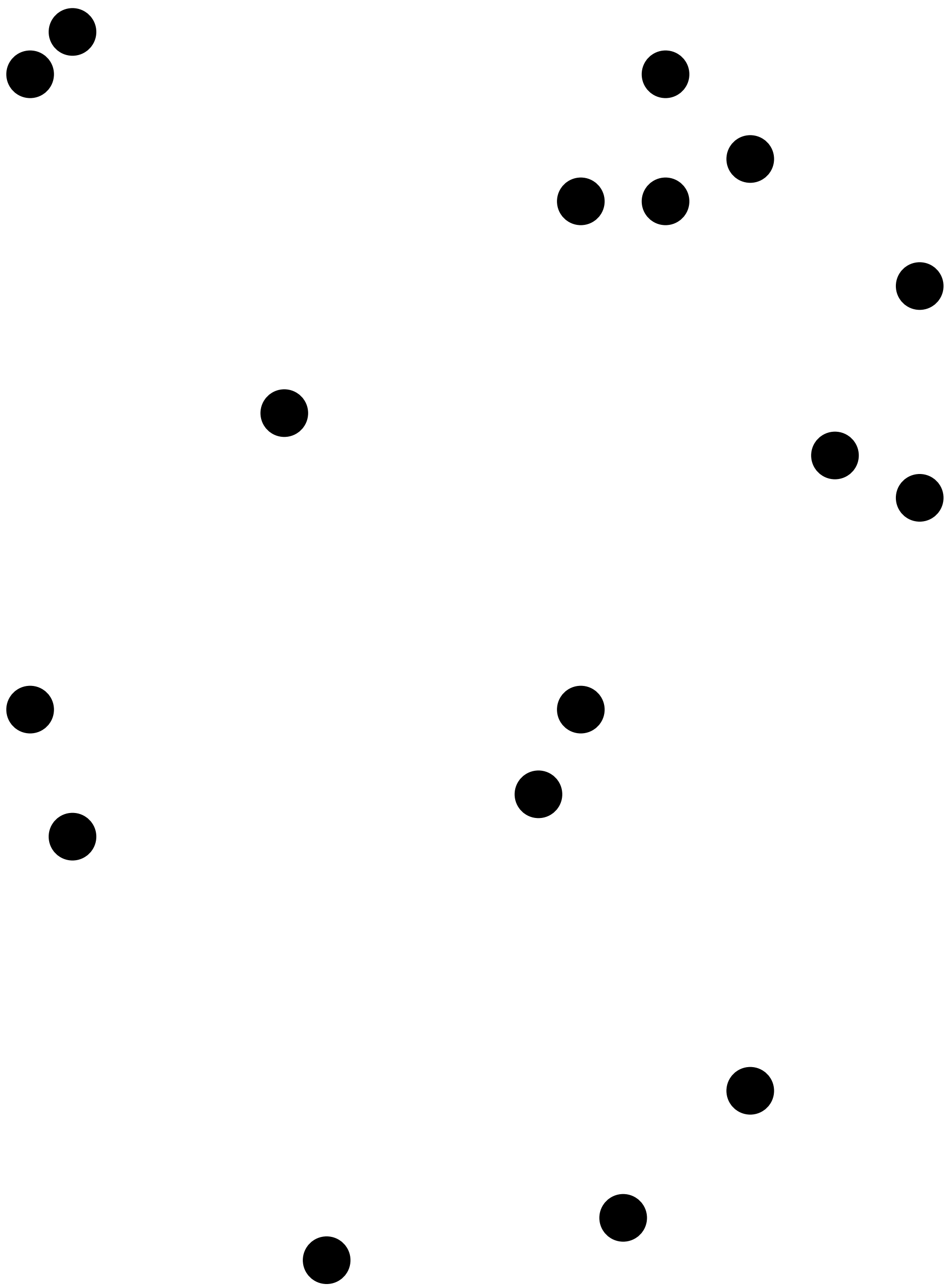
Time Bounds for Computing Them

- Area
  - Largest area, parallel line segments $O(n^3)$
  - Largest area, arbitrary line segments NP-hard
  - Largest area, unit squares $O(n^5)$
  - Largest area, disjoint squares $O(n^7)$
  - Smallest area, parallel line segments $O(n \log n)$
  - Smallest area, squares $O(n^2)$
- Perimeter
  - Largest perimeter, parallel line segments $O(n^5)$
  - Largest perimeter, disjoint squares $O(n^{10})$
  - Smallest perimeter, squares $O(n \log n)$

Approximate Largest Convex Hull

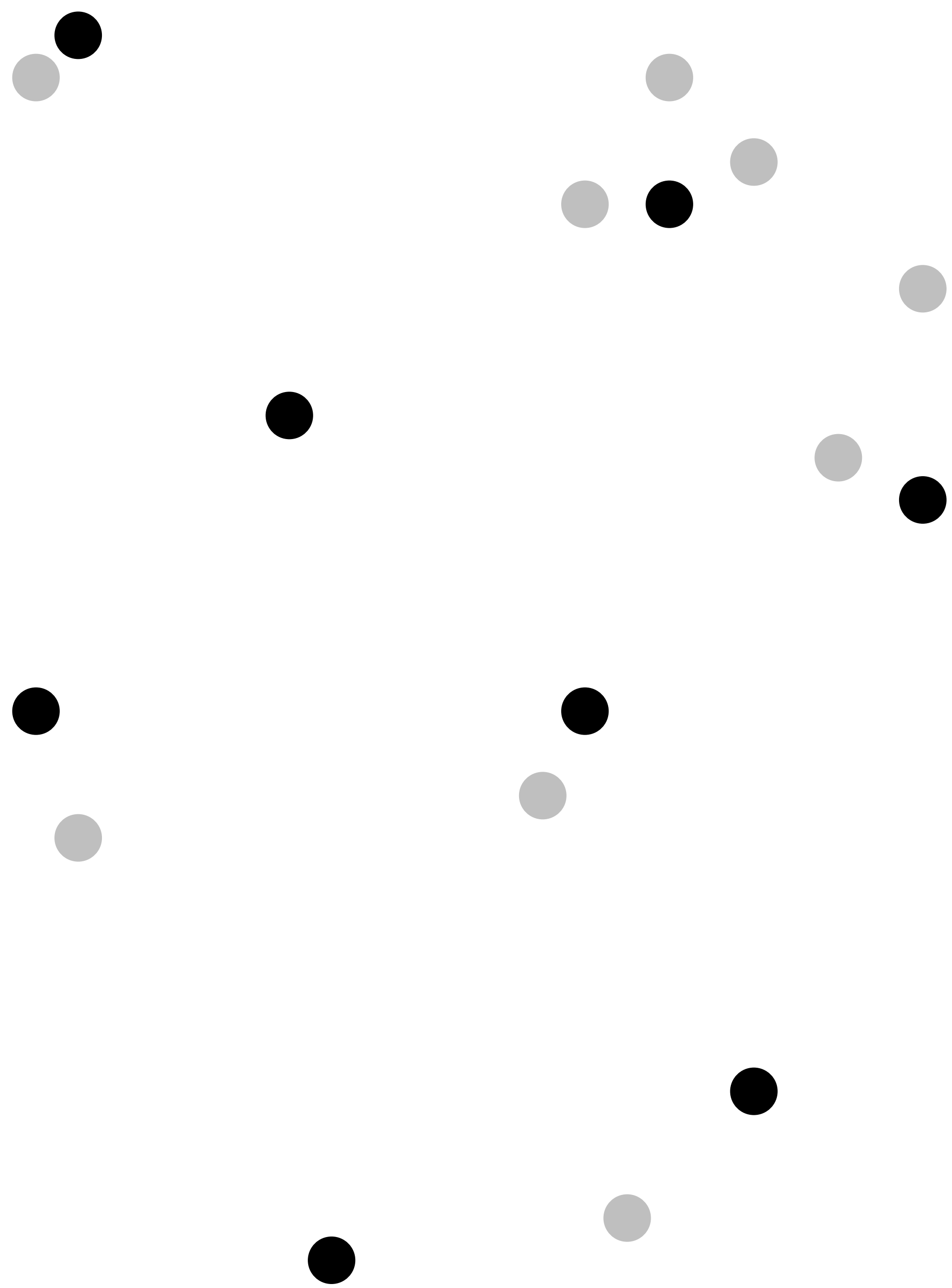
- Computing the largest possible convex hull is hard
- Instead, we will compute the almost-largest convex hull
- A $(1 - \varepsilon)$-approximation of the convex hull gives only a slightly weaker bound on the real size

Core-Sets



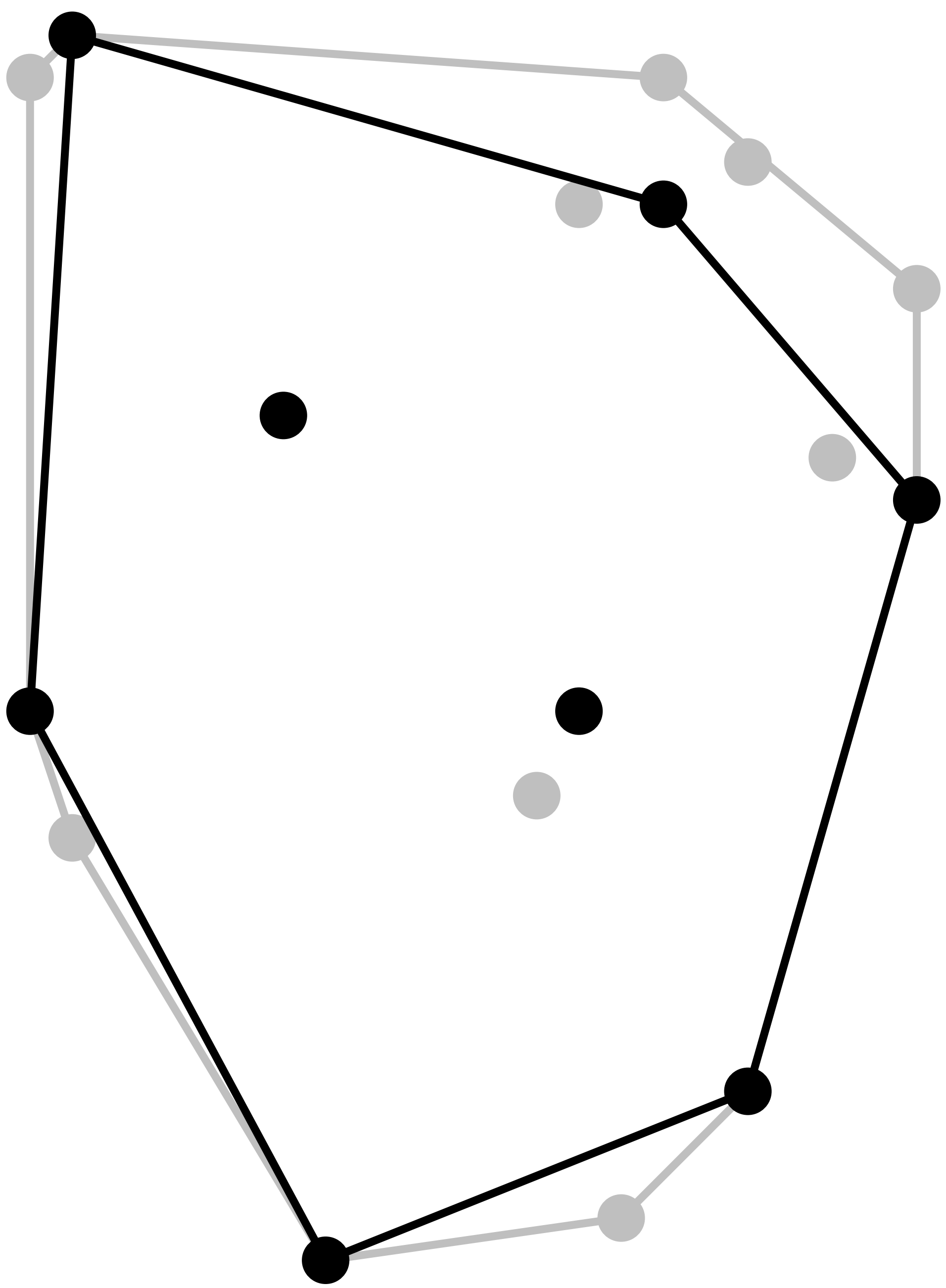
- Let P be a set of n points in $\mathbb{R}^2$
- Suppose we want to compute $\mu(P)$
- $P' \subset P$ is a *core-set* w.r.t. μ if:
  - $\mu(P') \geq (1-\varepsilon)\mu(P)$
  - $|P'|$ is constant
  - $|P'|$ does depend on ε
- Compute first P' and then $\mu(P')$

Core-Sets



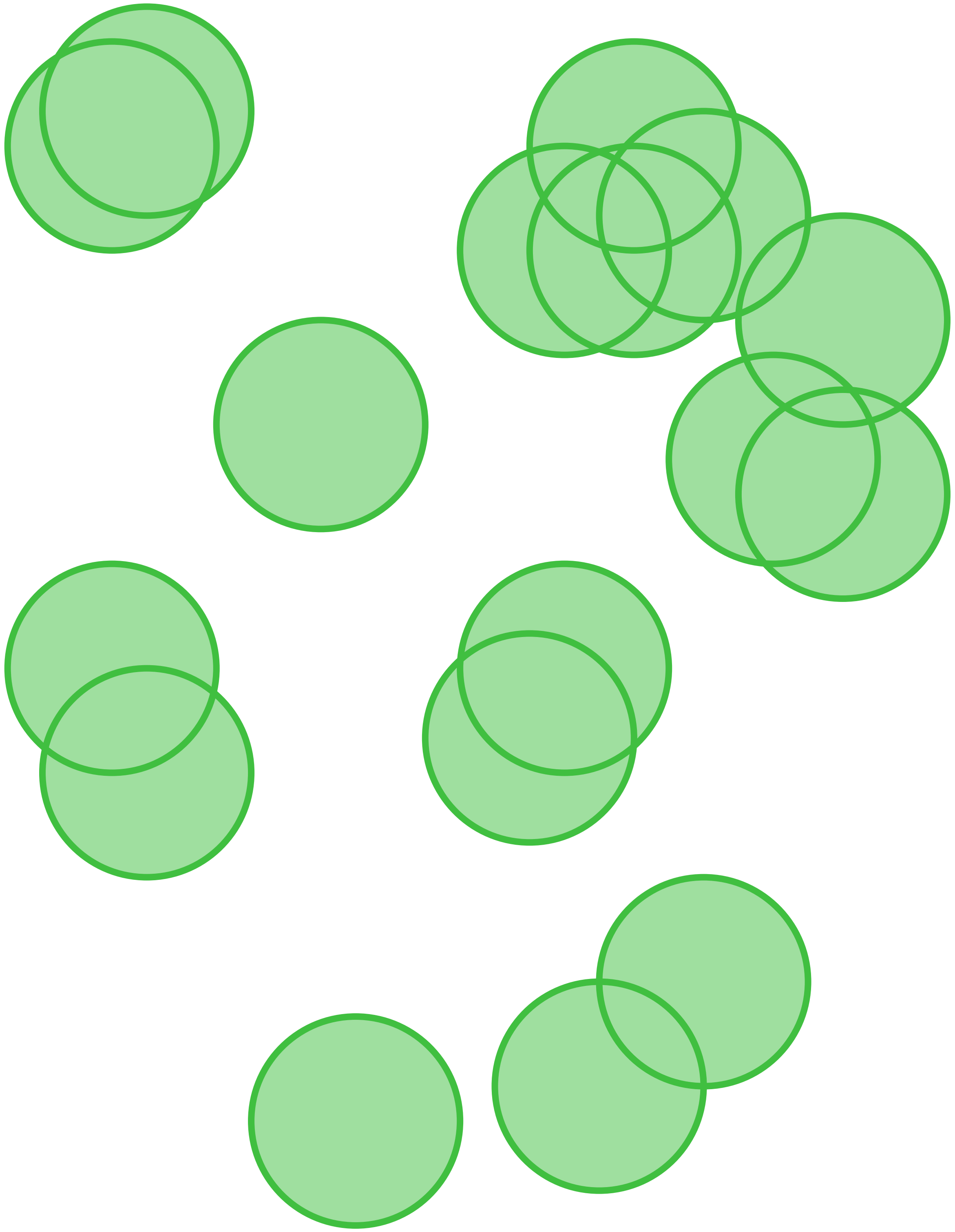
- Let P be a set of n points in $\mathbb{R}^2$
- Suppose we want to compute $\mu(P)$
- $P' \subset P$ is a *core-set* w.r.t. μ if:
  - $\mu(P') \geq (1 - \varepsilon)\mu(P)$
  - $|P'|$ is constant
  - $|P'|$ does depend on ε
- Compute first P' and then $\mu(P')$

Core-Sets



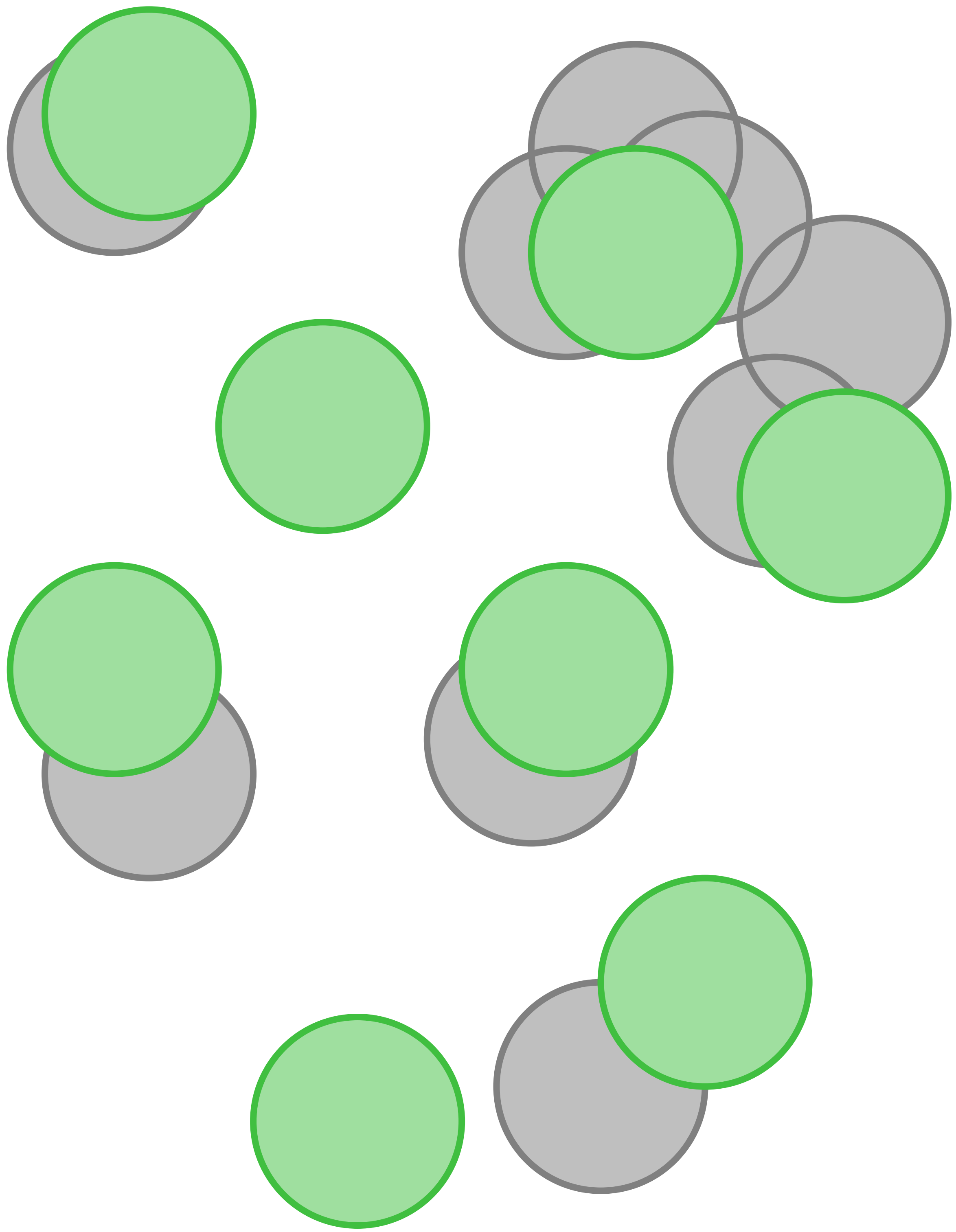
- Let P be a set of n points in $\mathbb{R}^2$
- Suppose we want to compute $\mu(P)$
- $P' \subset P$ is a *core-set* w.r.t. μ if:
  - $\mu(P') \geq (1 - \varepsilon)\mu(P)$
  - $|P'|$ is constant
  - $|P'|$ does depend on ε
- Compute first P' and then $\mu(P')$

Imprecise Core-Sets



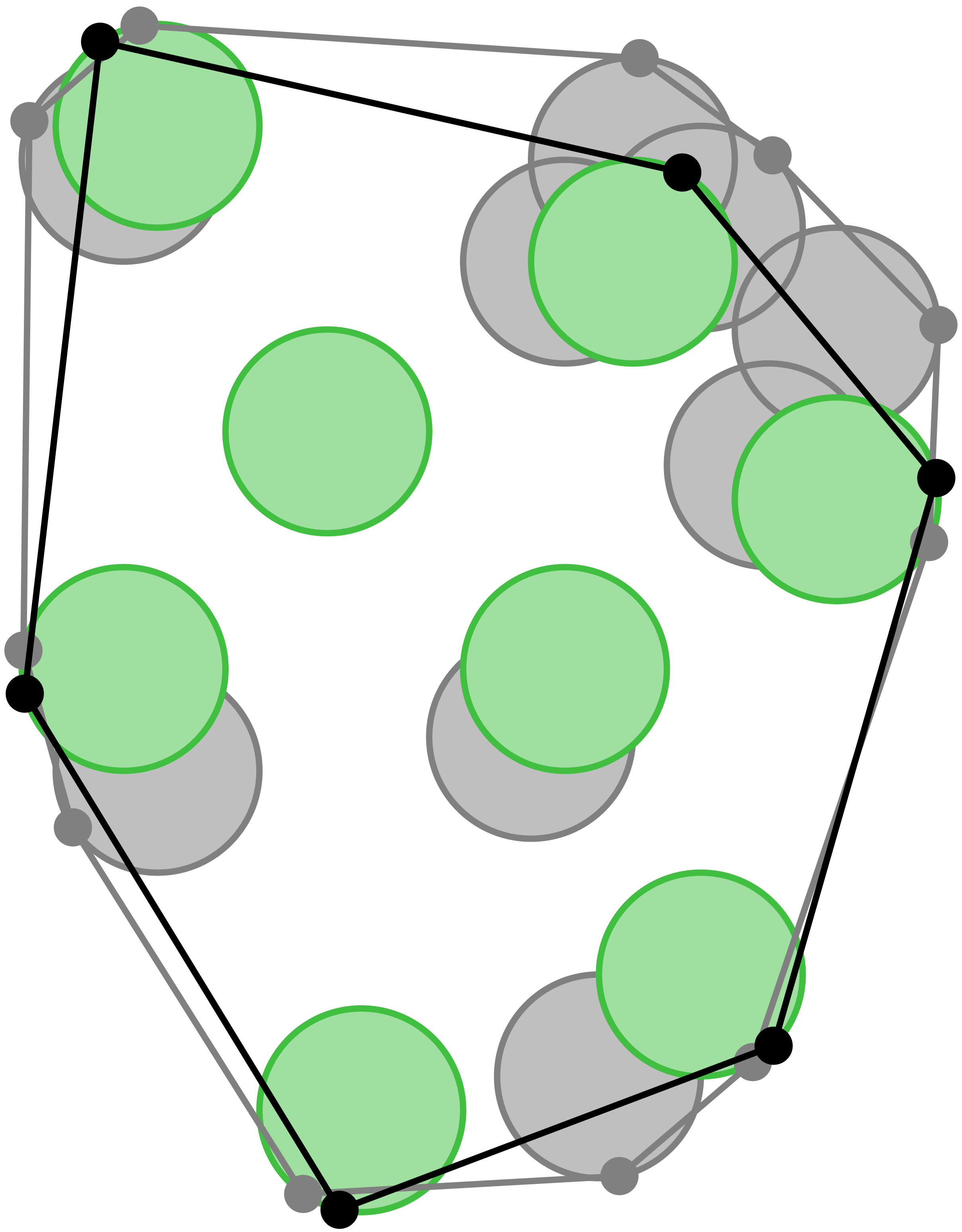
- Let $\mathcal{L}$ be a set of n regions in $\mathbb{R}^2$
- Suppose we want to compute $\mu(\mathcal{L})$
- $\mathcal{L}' \subset \mathcal{L}$ is a *core-set* w.r.t. μ if:
  - $\mu(\mathcal{L}') \geq (1 - \varepsilon)\mu(\mathcal{L})$
  - $|\mathcal{L}'|$ is constant
  - $|\mathcal{L}'|$ does depend on ε
- Compute first $\mathcal{L}'$ and then $\mu(\mathcal{L}')$

Imprecise Core-Sets



- Let $\mathcal{L}$ be a set of n regions in $\mathbb{R}^2$
- Suppose we want to compute $\mu(\mathcal{L})$
- $\mathcal{L}' \subset \mathcal{L}$ is a *core-set* w.r.t. μ if:
  - $\mu(\mathcal{L}') \geq (1-\varepsilon)\mu(\mathcal{L})$
  - $|\mathcal{L}'|$ is constant
  - $|\mathcal{L}'|$ does depend on ε
- Compute first $\mathcal{L}'$ and then $\mu(\mathcal{L}')$

Imprecise Core-Sets

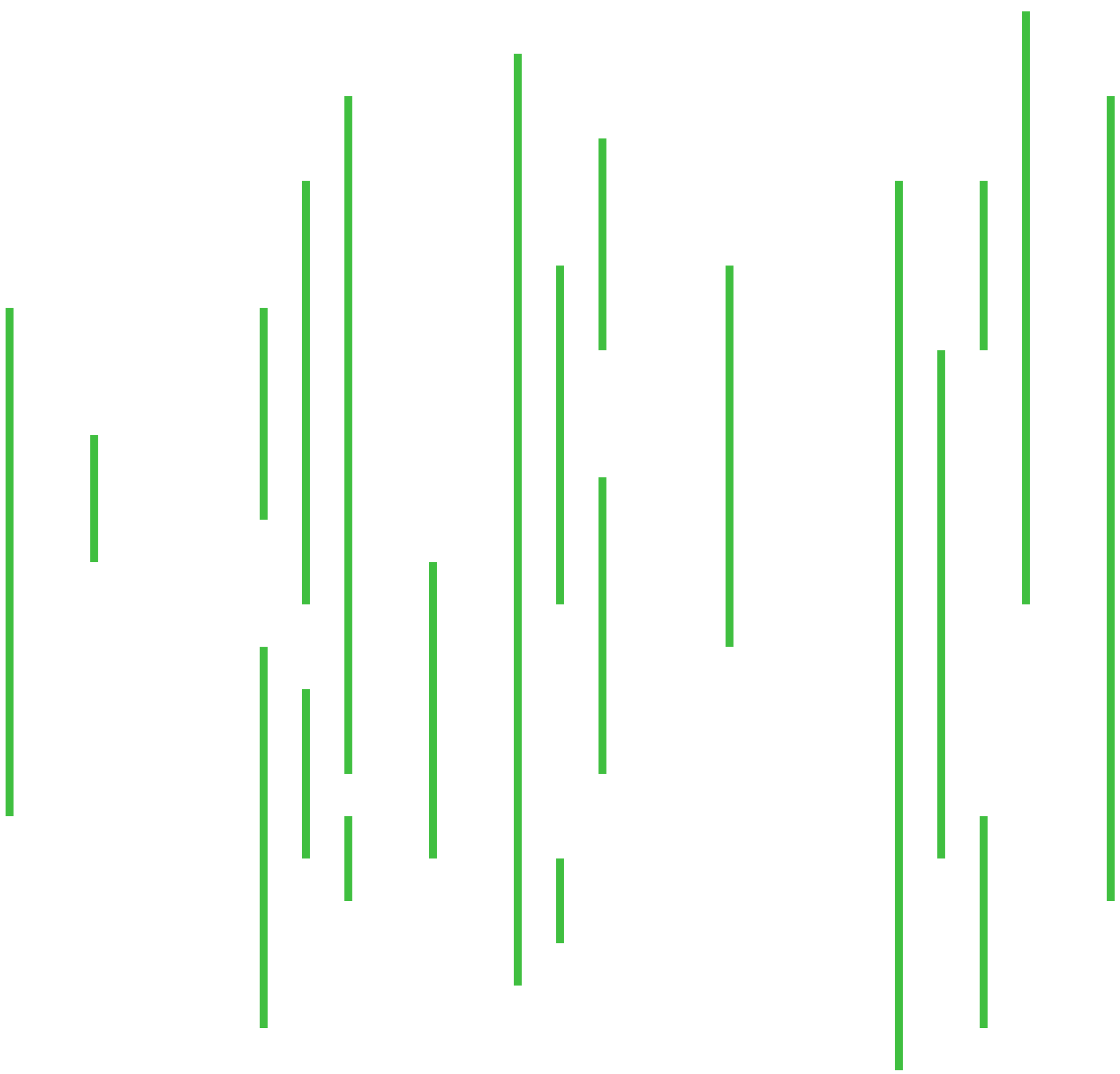


- Let $\mathcal{L}$ be a set of n regions in $\mathbb{R}^2$
- Suppose we want to compute $\mu(\mathcal{L})$
- $\mathcal{L}' \subset \mathcal{L}$ is a *core-set* w.r.t. μ if:
  - $\mu(\mathcal{L}') \geq (1 - \varepsilon)\mu(\mathcal{L})$
  - $|\mathcal{L}'|$ is constant
  - $|\mathcal{L}'|$ does depend on ε
- Compute first $\mathcal{L}'$ and then $\mu(\mathcal{L}')$

Results

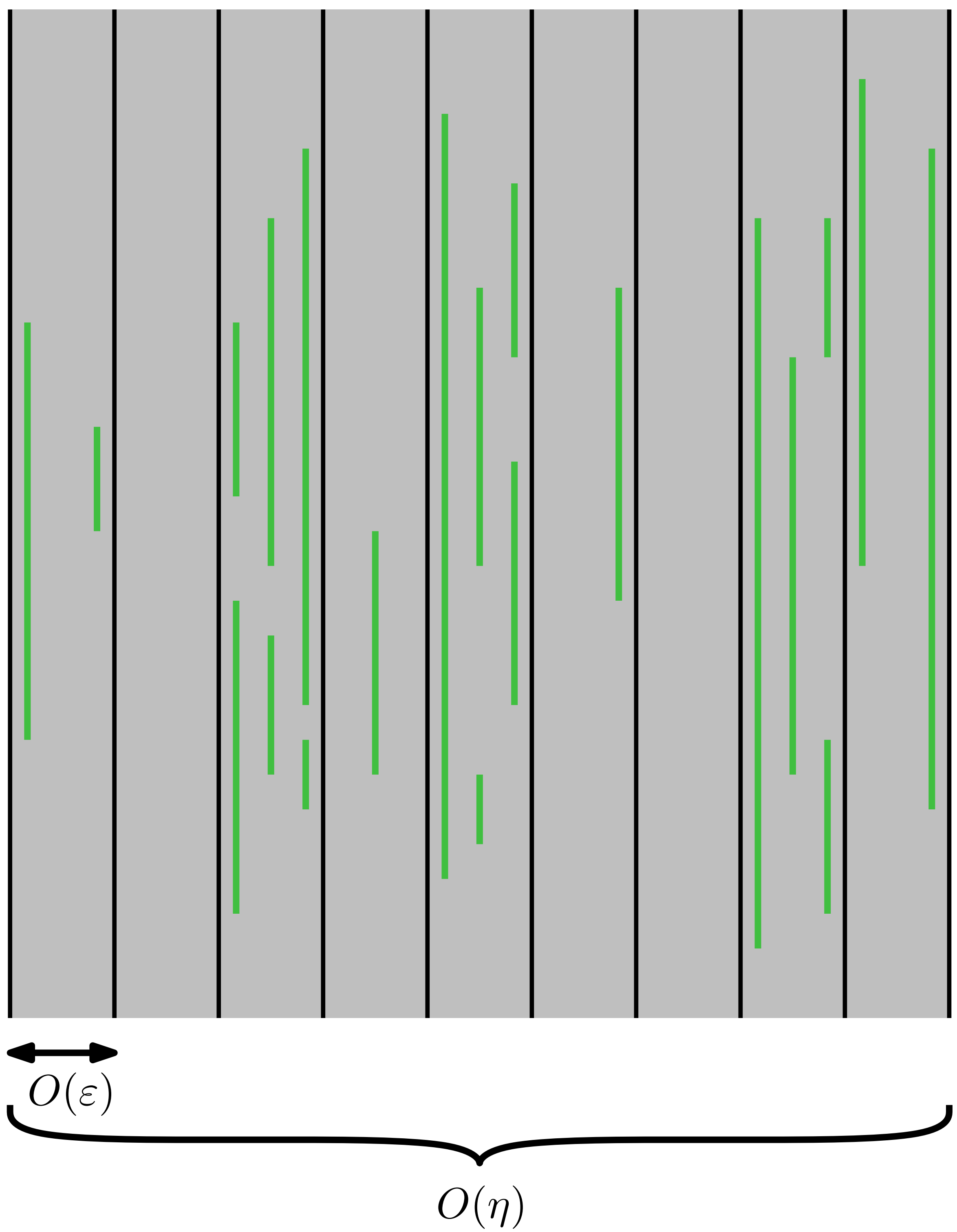
- Line segments, parallel $O(n + \eta^3)$
- Line segments $O(n) + 2^{O(\eta^2)}$
- Squares, disjoint $O(n + \eta^{14})$
- Squares, same size $O(n + \eta^{12})$
- Squares $O(n) + 2^{O(\eta^2)}$
- Regular k-gons, disjoint $O(n) + 2^{O(k \log \eta)}$
- Regular k-gons $O(n) + 2^{O(\eta^2 \log k)}$
- Circles, disjoint $O(n) + 2^{O(\sqrt{\eta} \log \eta)}$
- Circles $O(n) + 2^{O(\eta^2 \log \eta)}$

Vertical Line Segments



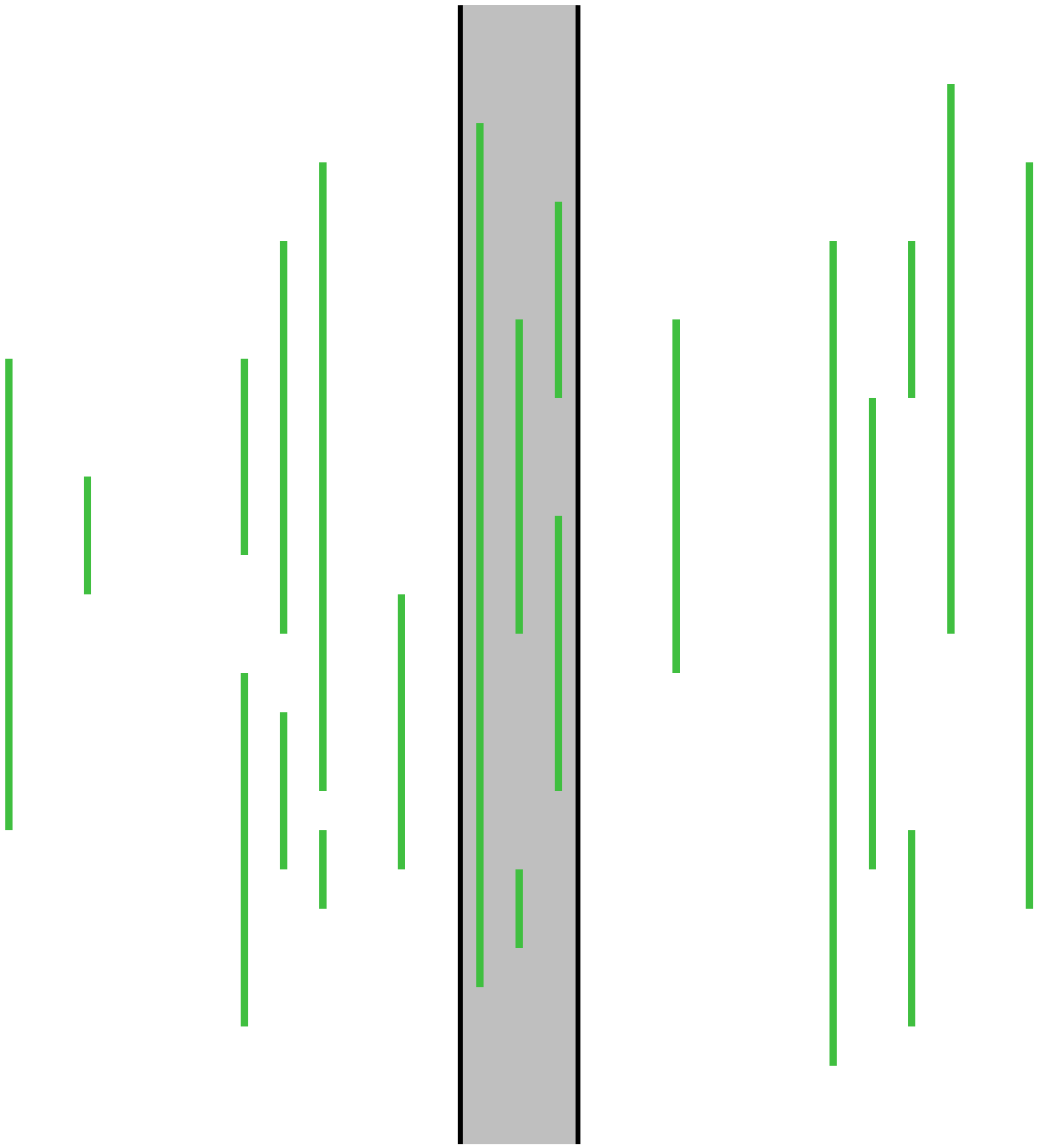
- Given a set $\mathcal{L}$ of line segments
- Divide them into strips of width $O(\varepsilon)$
- Take the two topmost and bottommost segments in each strip
- The result is a core-set $\mathcal{L}'$
- This takes $O(n)$ time

Vertical Line Segments



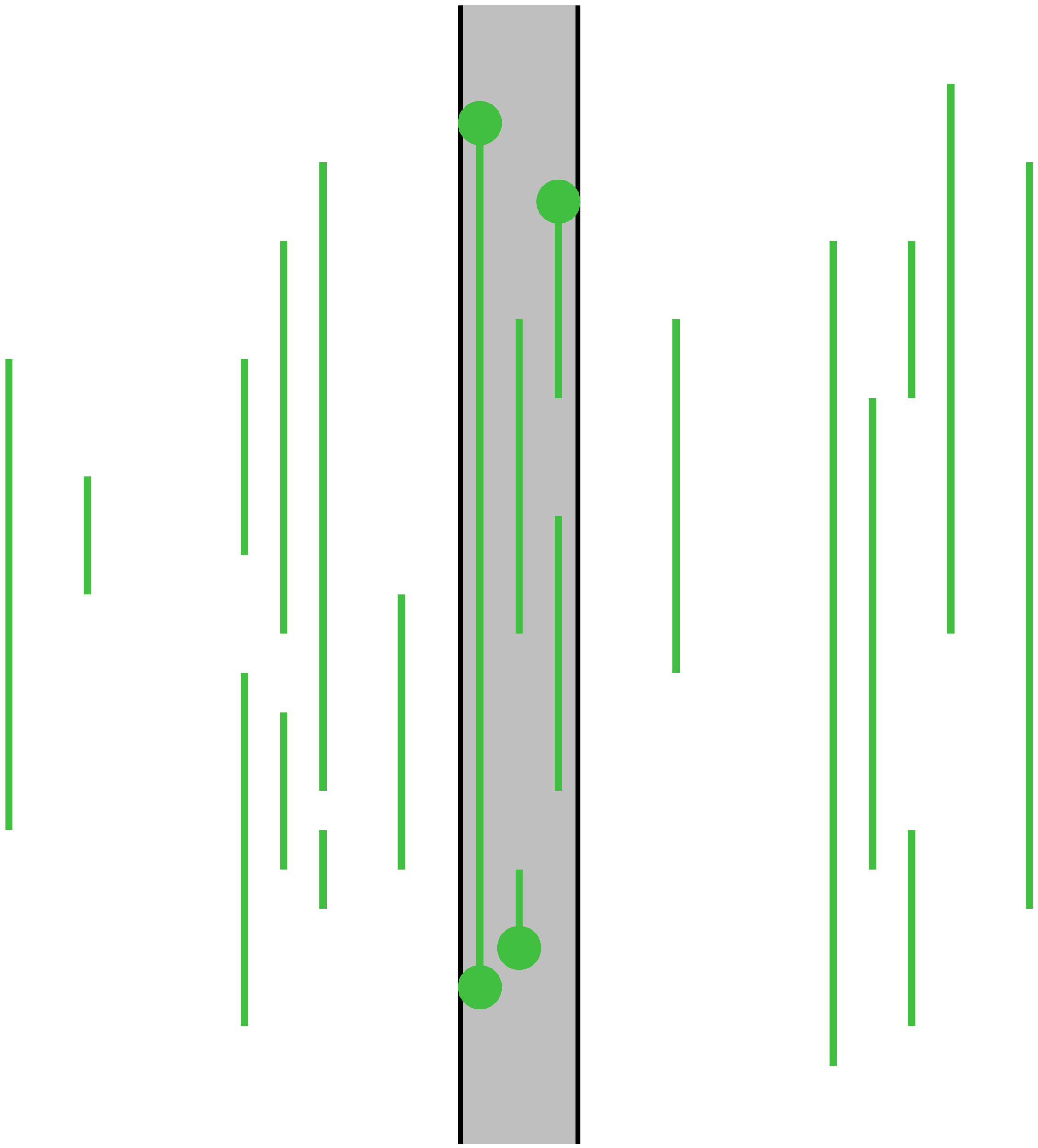
- Given a set $\mathcal{L}$ of line segments
- Divide them into strips of width $O(\varepsilon)$
- Take the two topmost and bottommost segments in each strip
- The result is a core-set $\mathcal{L}'$
- This takes $O(n)$ time

Vertical Line Segments



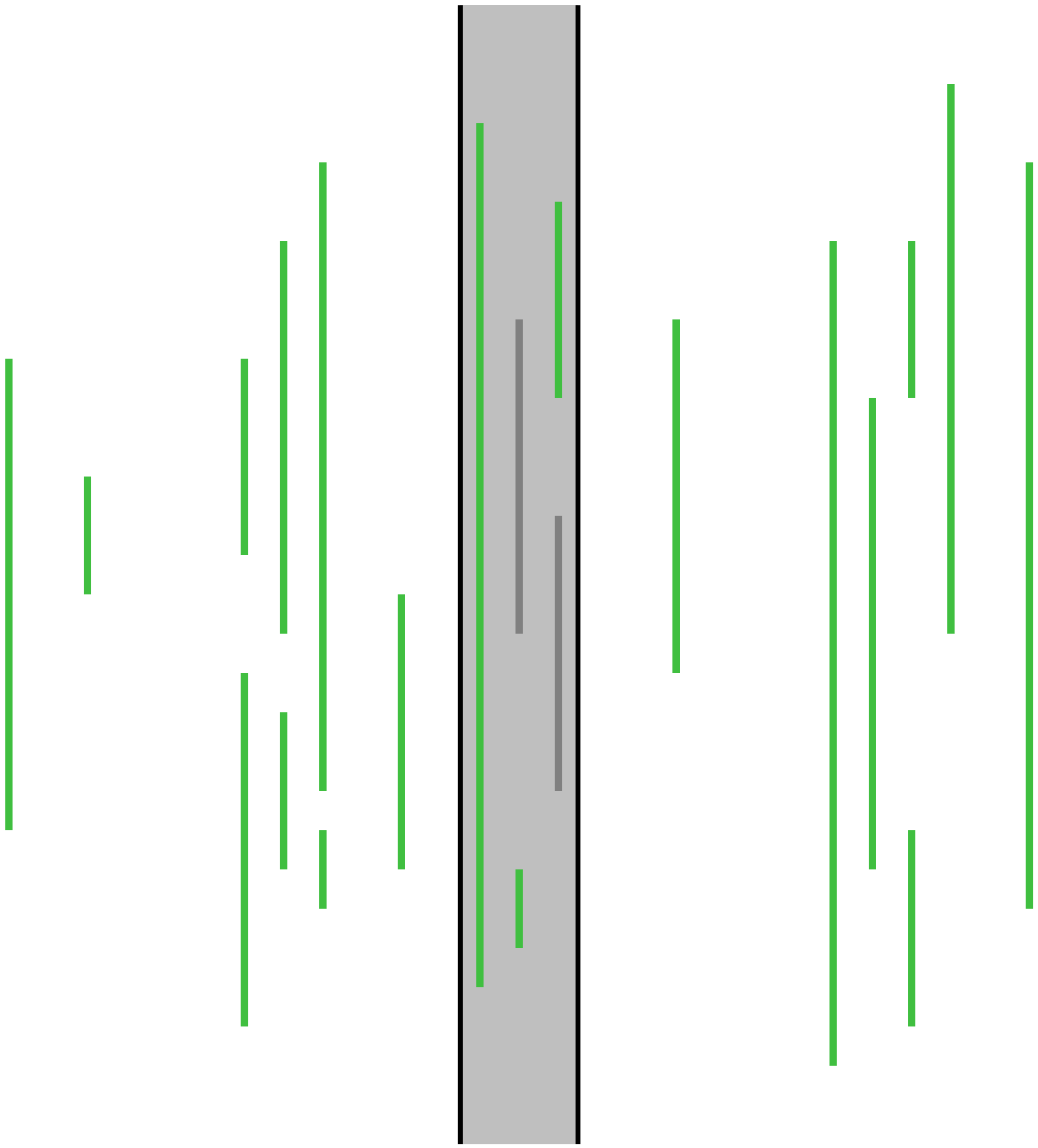
- Given a set $\mathcal{L}$ of line segments
- Divide them into strips of width $O(\varepsilon)$
- Take the two topmost and bottommost segments in each strip
- The result is a core-set $\mathcal{L}'$
- This takes $O(n)$ time

Vertical Line Segments



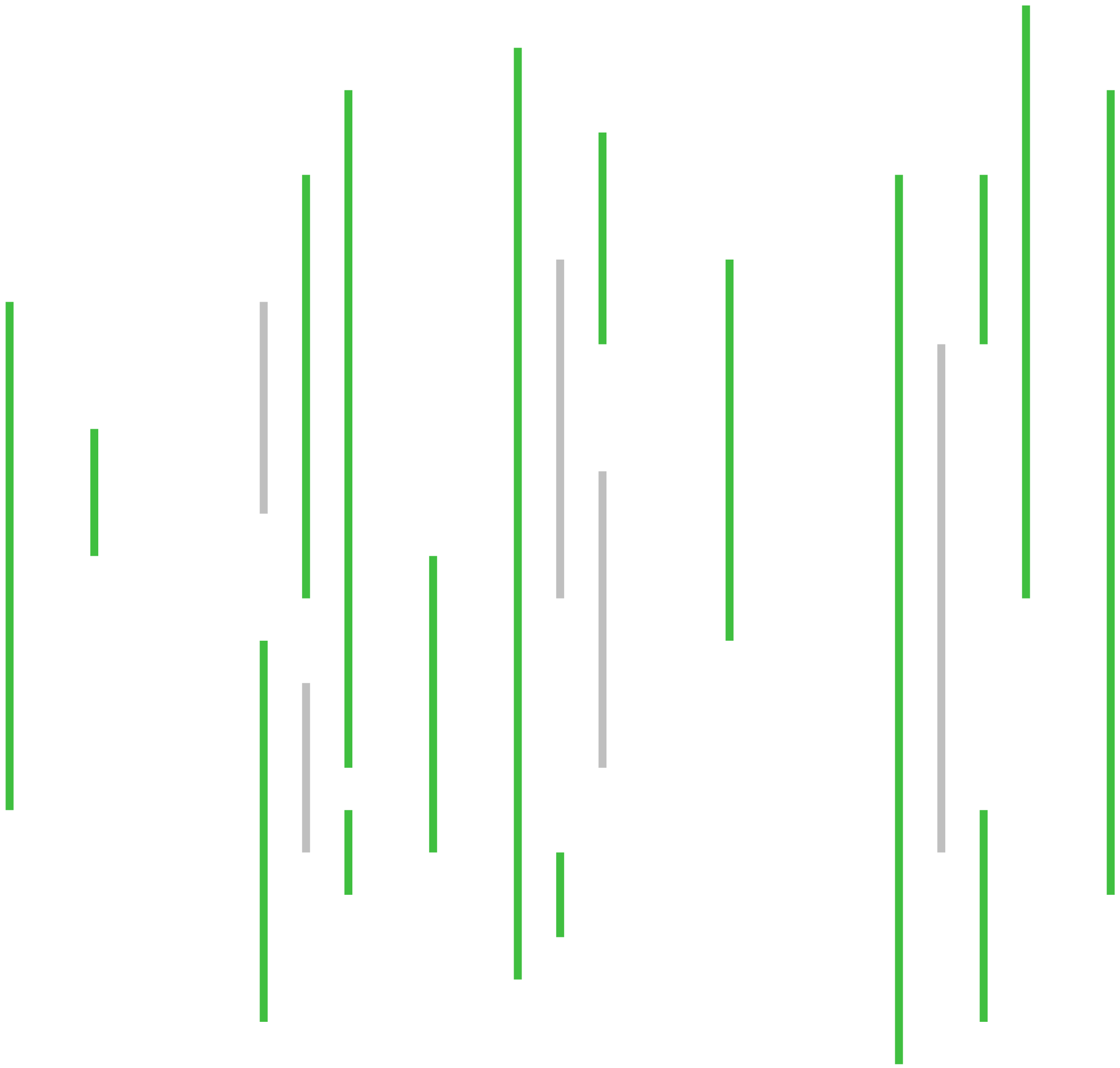
- Given a set $\mathcal{L}$ of line segments
- Divide them into strips of width $O(\varepsilon)$
- Take the two topmost and bottommost segments in each strip
- The result is a core-set $\mathcal{L}'$
- This takes $O(n)$ time

Vertical Line Segments



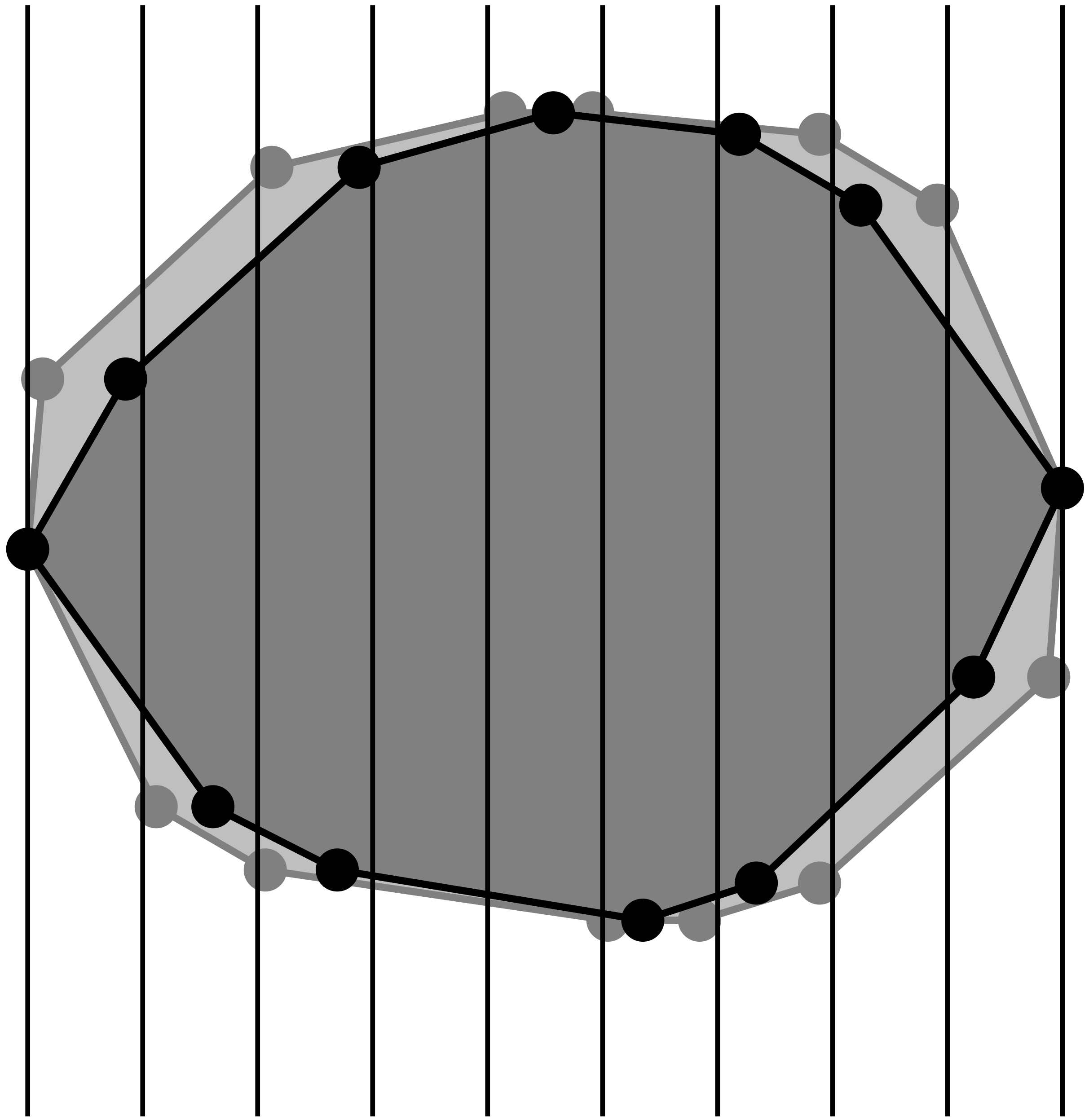
- Given a set $\mathcal{L}$ of line segments
- Divide them into strips of width $O(\varepsilon)$
- Take the two topmost and bottommost segments in each strip
- The result is a core-set $\mathcal{L}'$
- This takes $O(n)$ time

Vertical Line Segments



- Given a set $\mathcal{L}$ of line segments
- Divide them into strips of width $O(\varepsilon)$
- Take the two topmost and bottommost segments in each strip
- The result is a core-set $\mathcal{L}'$
- This takes $O(n)$ time

Why Is This A Core-Set?

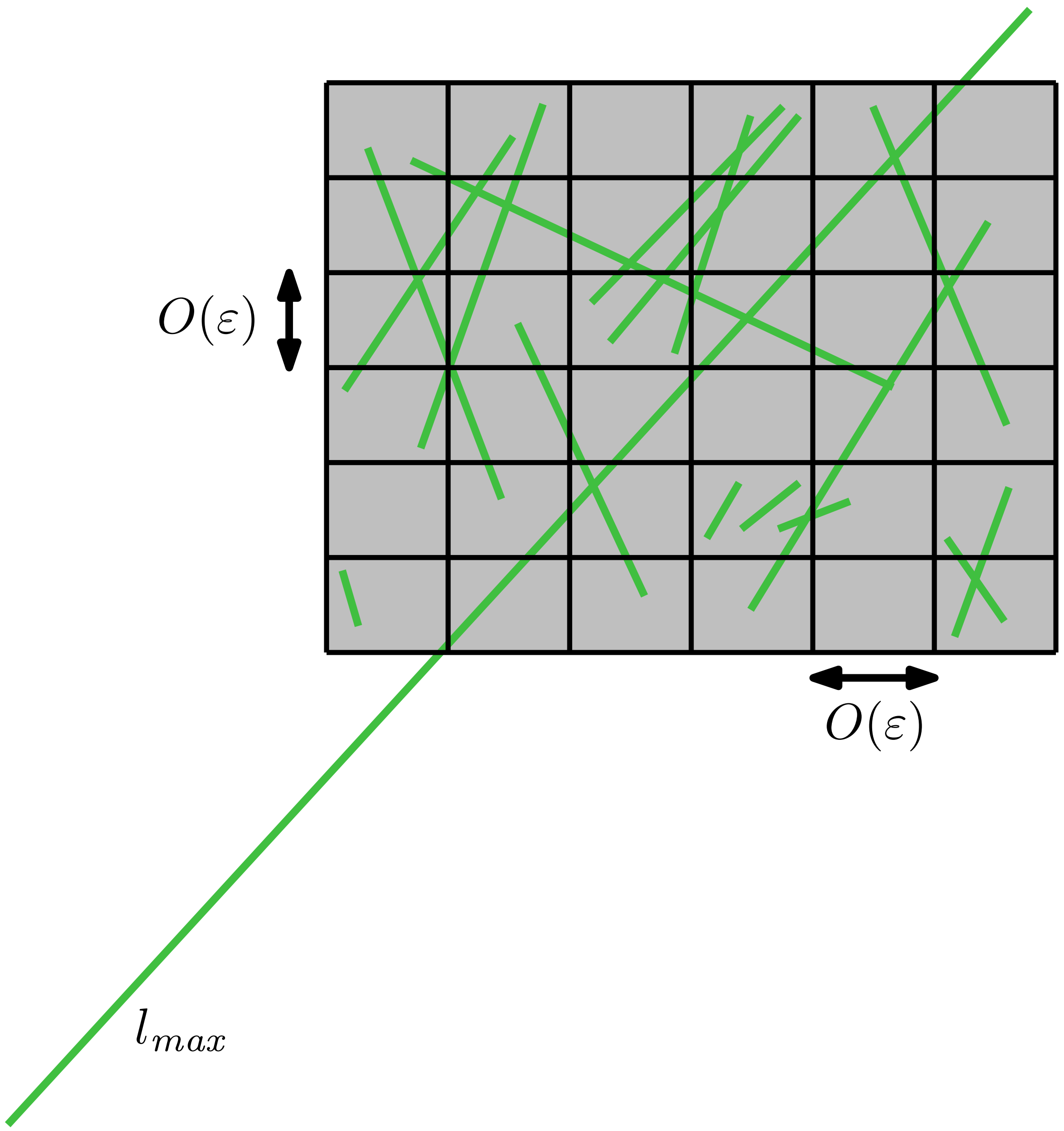


- It has constant size:
  - $O(\eta)$ strips
  - At most four segments per strip
- It approximates the optimal solution:
  - Vertical difference per strip at most 0
  - Horizontal difference per strip at most $O(\varepsilon)$
  - At most $O(\varepsilon h)$

Arbitrary Line Segments

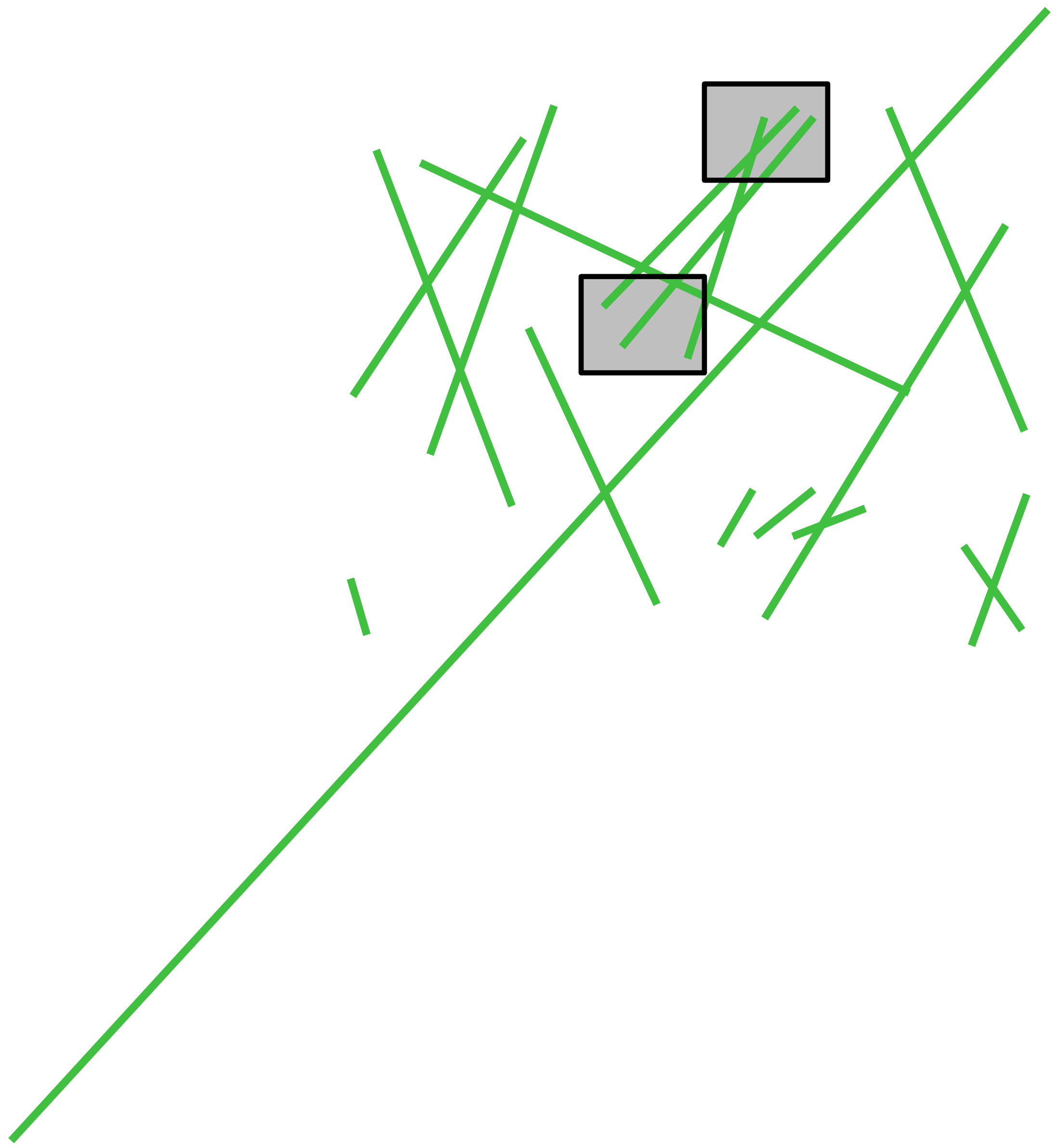
- Given a set $\mathcal{L}$ of line segments
- Put the longest segment l_{max} into $\mathcal{L}'$
- Divide the remaining segments into cells of size $O(\varepsilon)$ by $O(\varepsilon)$
- For any pair of cells, put at most one segment into $\mathcal{L}'$

Arbitrary Line Segments



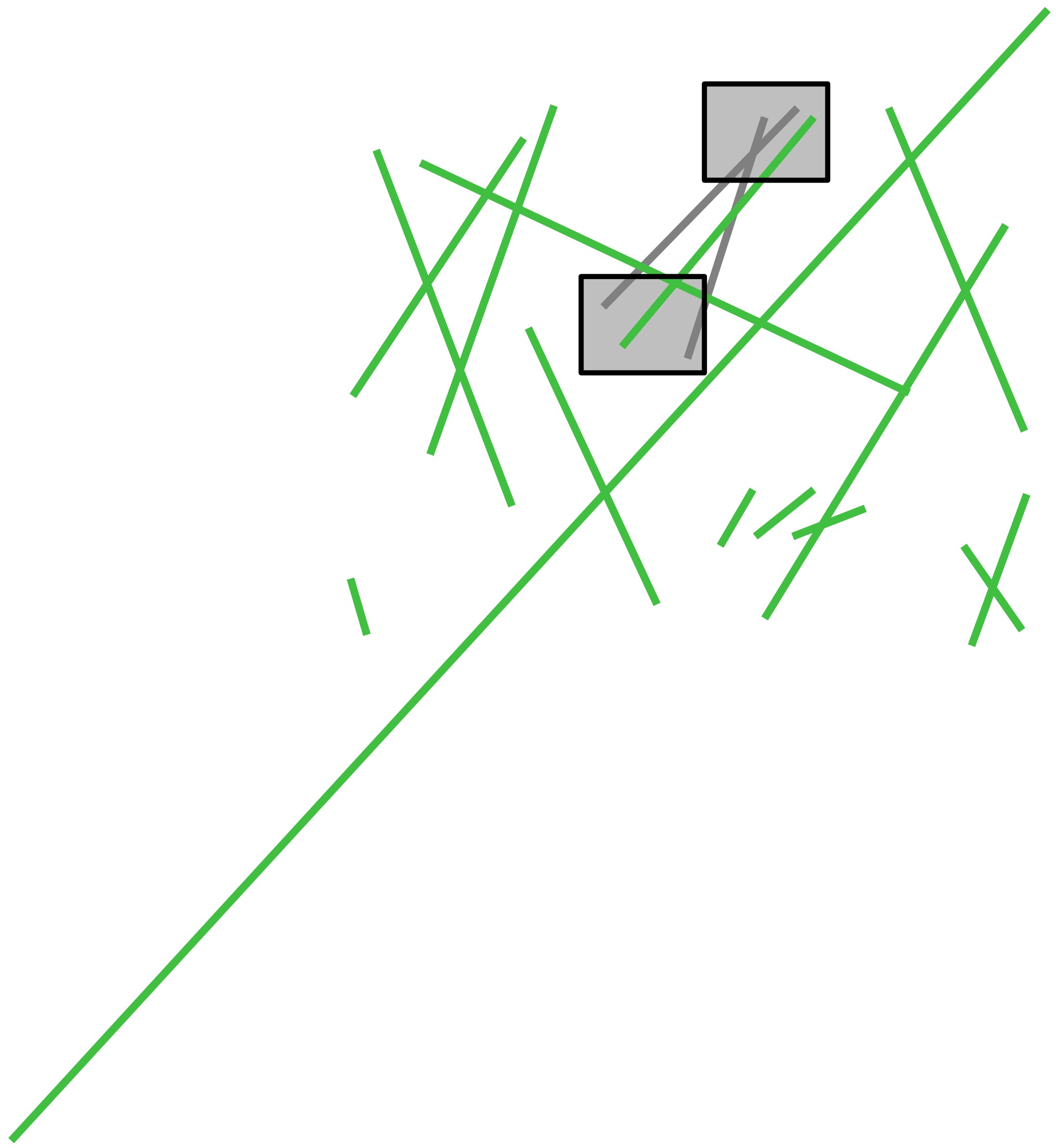
- Given a set $\mathcal{L}$ of line segments
- Put the longest segment l_{max} into $\mathcal{L}'$
- Divide the remaining segments into cells of size $O(\varepsilon)$ by $O(\varepsilon)$
- For any pair of cells, put at most one segment into $\mathcal{L}'$

Arbitrary Line Segments



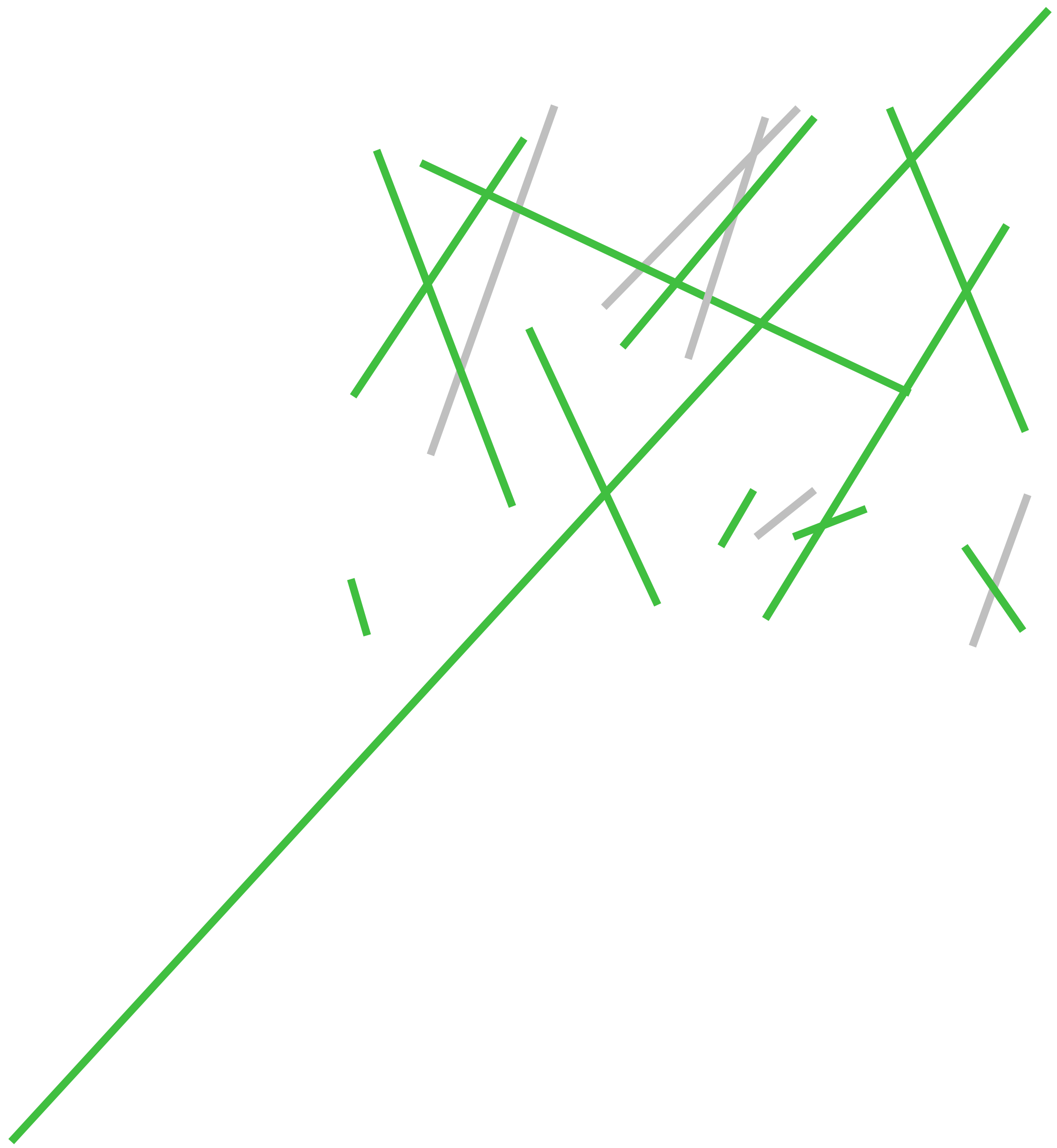
- Given a set $\mathcal{L}$ of line segments
- Put the longest segment l_{max} into $\mathcal{L}'$
- Divide the remaining segments into cells of size $O(\varepsilon)$ by $O(\varepsilon)$
- For any pair of cells, put at most one segment into $\mathcal{L}'$

Arbitrary Line Segments



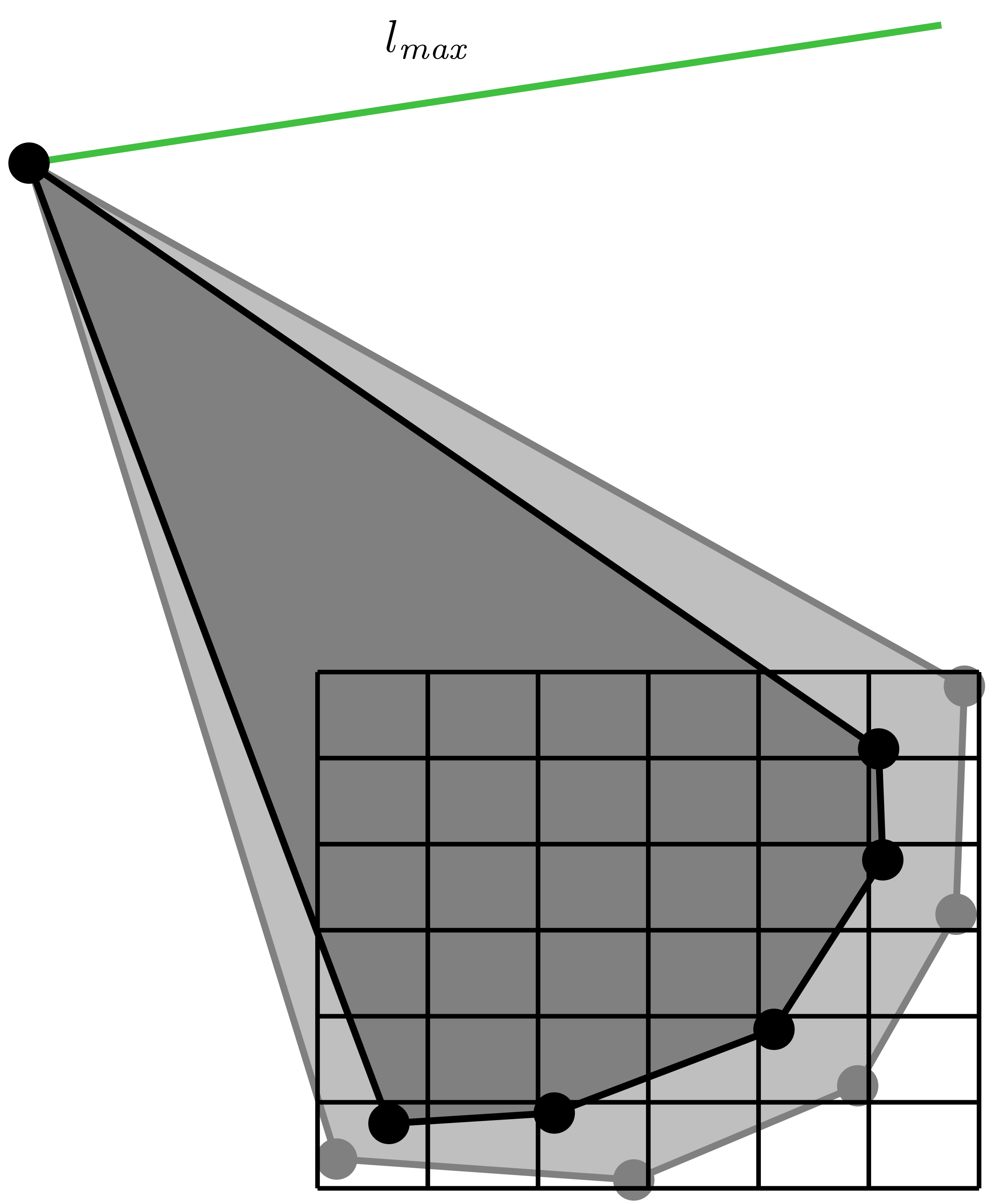
- Given a set $\mathcal{L}$ of line segments
- Put the longest segment l_{max} into $\mathcal{L}'$
- Divide the remaining segments into cells of size $O(\varepsilon)$ by $O(\varepsilon)$
- For any pair of cells, put at most one segment into $\mathcal{L}'$

Arbitrary Line Segments



- Given a set $\mathcal{L}$ of line segments
- Put the longest segment l_{max} into $\mathcal{L}'$
- Divide the remaining segments into cells of size $O(\varepsilon)$ by $O(\varepsilon)$
- For any pair of cells, put at most one segment into $\mathcal{L}'$

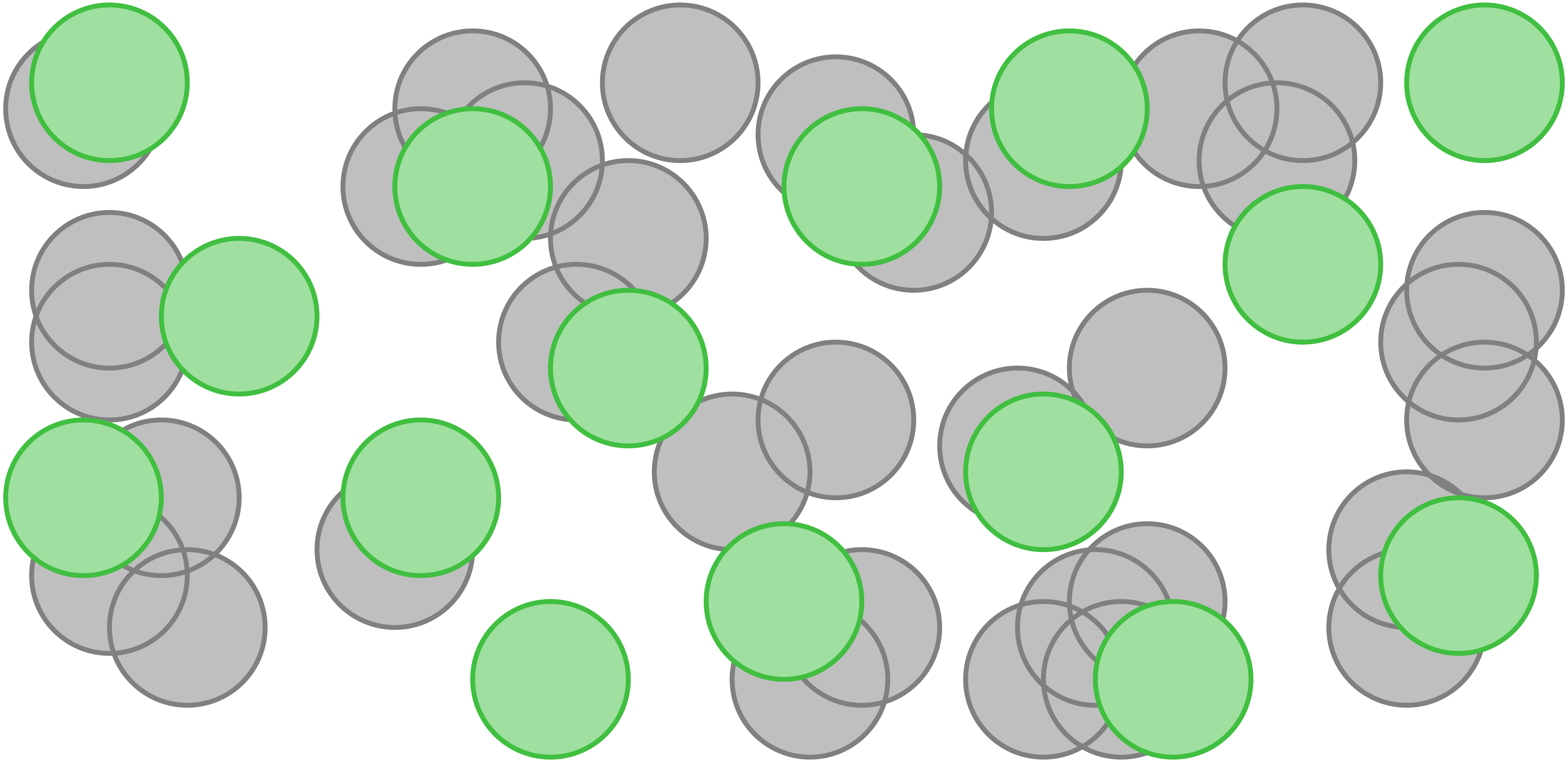
Why Is This A Core-Set?



- It has constant size:
  - $O(\eta^2)$ cells
  - At most one segments for any two cells
- It approximates the optimal solution:
  - Vertical difference per strip at most $O(\varepsilon)$
  - Horizontal difference per strip at most $O(\varepsilon)$

Concluding Remarks

- Data imprecision is a problem in computational geometry
- The largest possible area convex hull cannot be computed exactly efficiently
- Imprecise core-sets yield strongly linear time approximation schemes



Questions?