Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

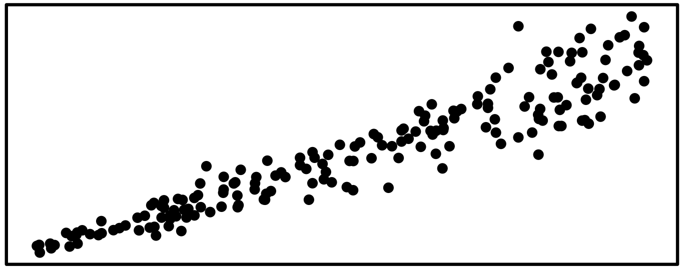
Selecting the Aspect Ratio of a Scatter Plot Based on Its Delaunay Triangulation

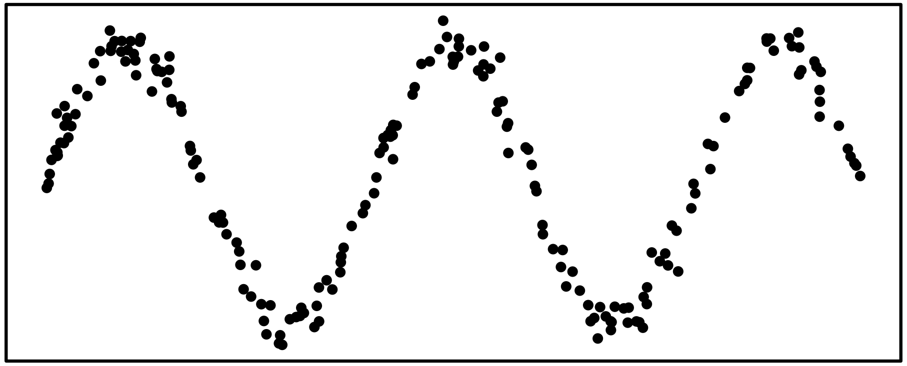
Martin Fink
Lehrstuhl für Informatik I
Universität Würzburg

Joint work with
Jan-Henrik Haunert, Joachim Spoerhase & Alexander Wolff

Scatter Plots . . .

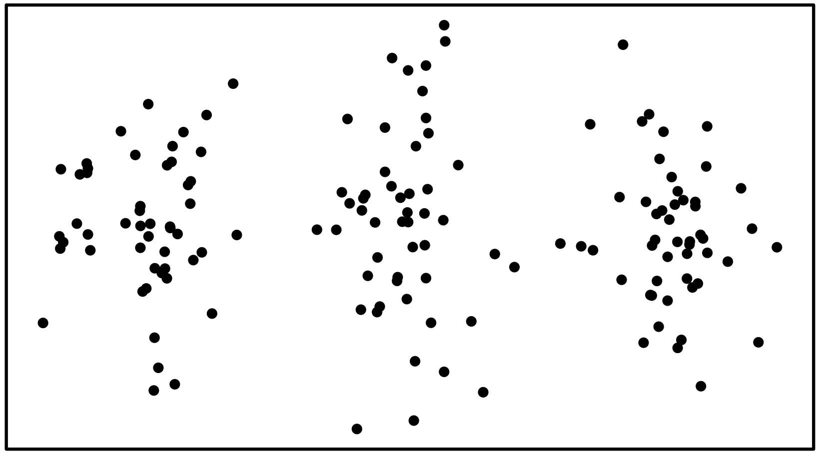
- . . . reveal trends . . .

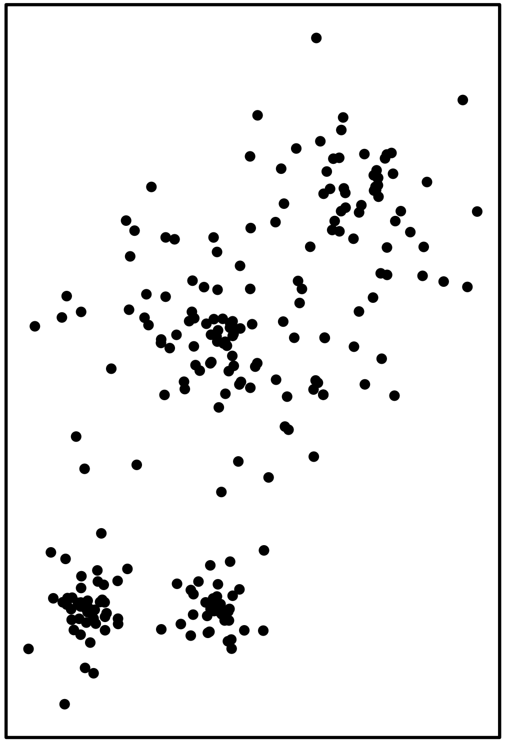




Scatter Plots . . .

- . . . reveal trends . . .
- . . . or clusters.



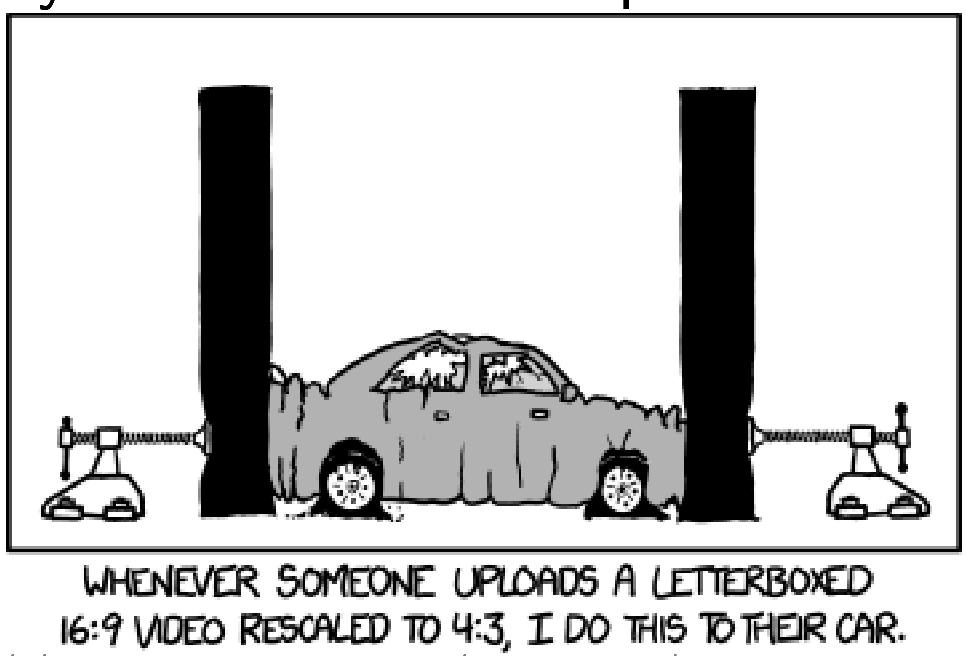


Scatter Plots . . .

- . . . reveal trends . . .
- . . . or clusters.
- . . . are most-frequently used visualizations in scientific publications. [Tufte, 2001]

Scatter Plots . . .

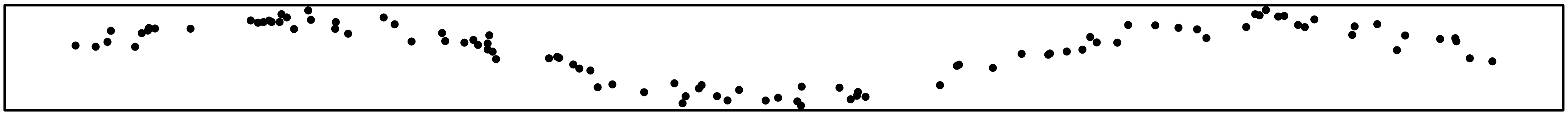
- . . . reveal trends . . .
- . . . or clusters.
- . . . are most-frequently used visualizations in scientific publications. [Tufte, 2001]
- . . . heavily rely on the chosen aspect ratio.



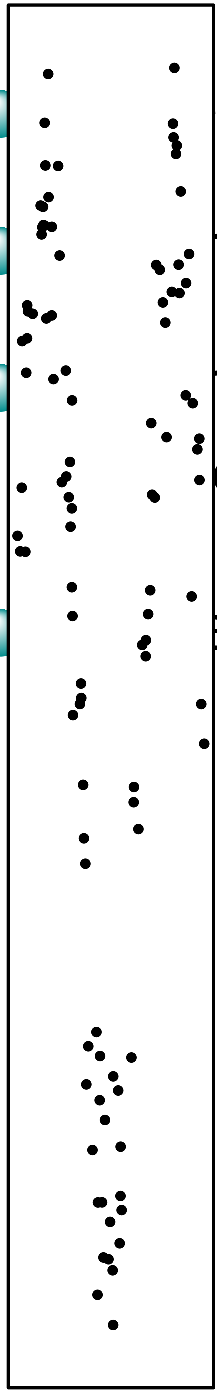
[http://imgs.xkcd.com/comics/aspect_ratio.png]

Scatter Plots . . .

- . . . reveal trends . . .
- . . . or clusters.
- . . . are most-frequently used visualizations in scientific publications. [Tufte, 2001]
- . . . heavily rely on the chosen aspect ratio.



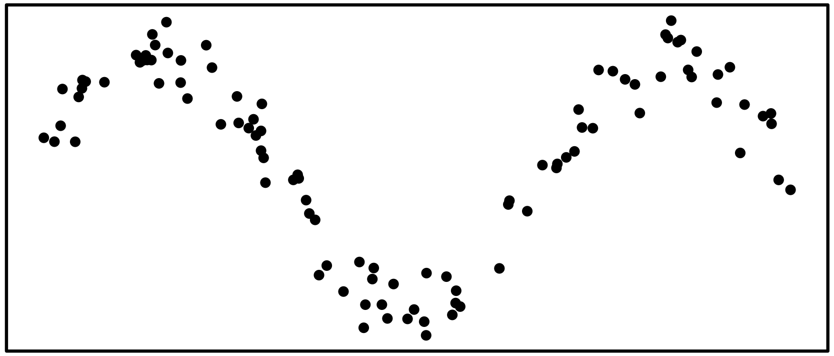
Scatter Plots . . .



- veal trends . . .
- clusters.
- e most-frequently used visualizations in scientific cations. [Tufte, 2001]
- eavily rely on the chosen aspect ratio.

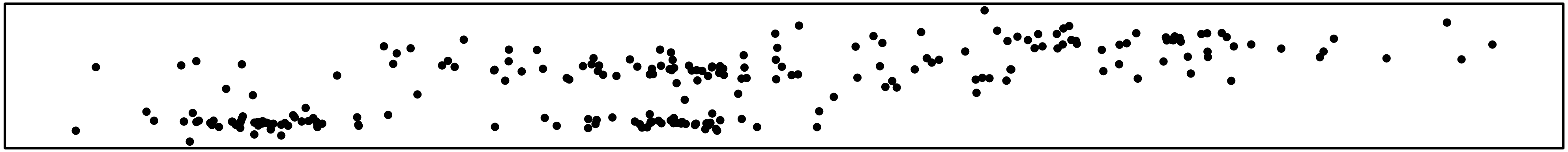
Scatter Plots . . .

- . . . reveal trends . . .
- . . . or clusters.
- . . . are most-frequently used visualizations in scientific publications. [Tufte, 2001]
- . . . heavily rely on the chosen aspect ratio.



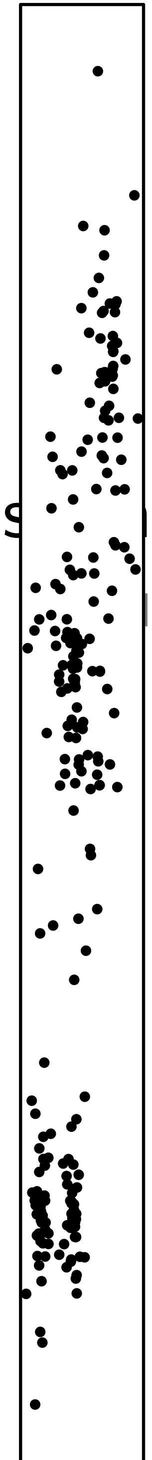
Scatter Plots . . .

- . . . reveal trends . . .
- . . . or clusters.
- . . . are most-frequently used visualizations in scientific publications. [Tufte, 2001]
- . . . heavily rely on the chosen aspect ratio.



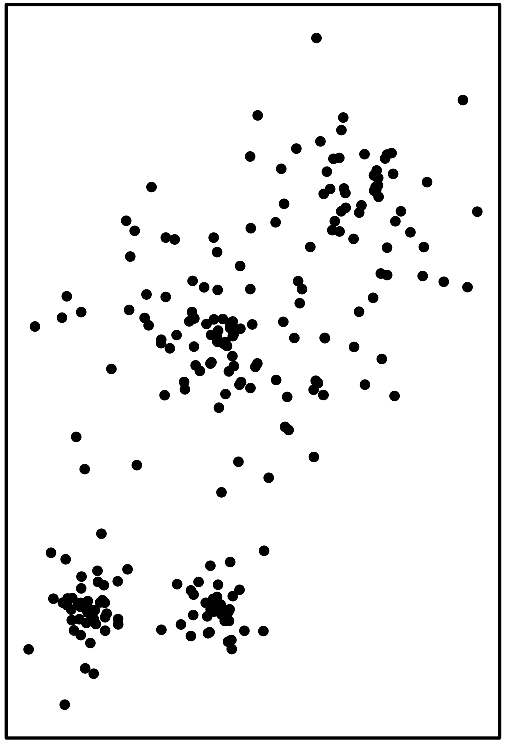
Scatter Plots . . .

- . . . reveal trends . . .
- . . . or clusters.
- . . . are most-frequently used visualizations in scientific publications. [Tufte, 2001]
- . . . heavily rely on the chosen aspect ratio.



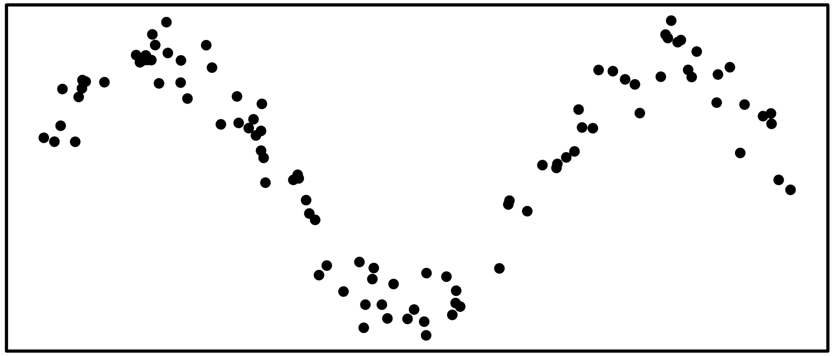
Scatter Plots . . .

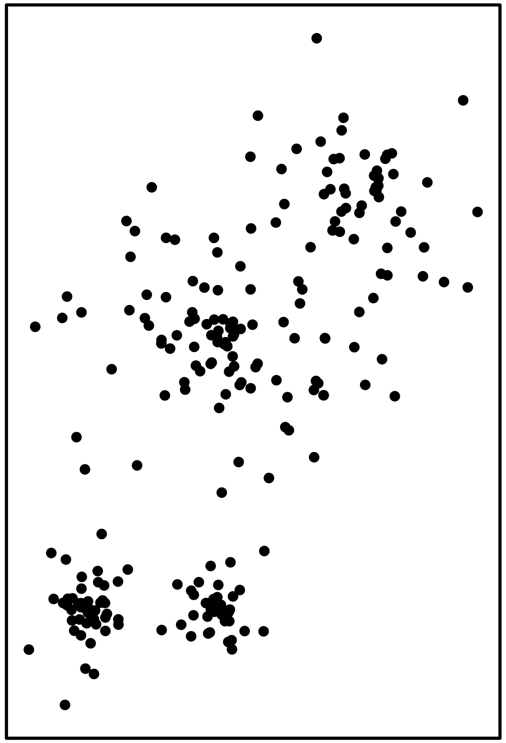
- . . . reveal trends . . .
- . . . or clusters.
- . . . are most-frequently used visualizations in scientific publications. [Tufte, 2001]
- . . . heavily rely on the chosen aspect ratio.



Scatter Plots . . .

- . . . reveal trends . . .
- . . . or clusters.
- . . . are most-frequently used visualizations in scientific publications. [Tufte, 2001]
- . . . heavily rely on the chosen aspect ratio.

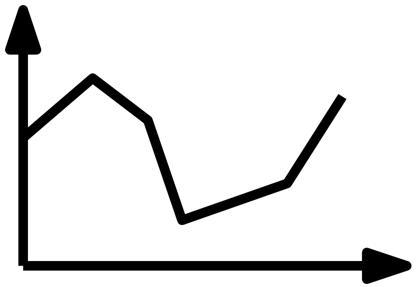




task:
automatically select a good aspect ratio

Previous Work

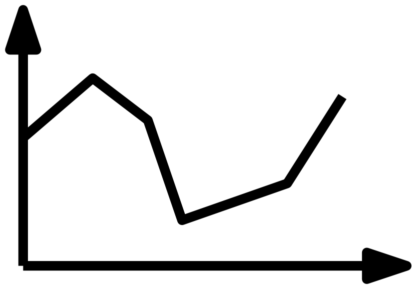
- aspect-ratio selection for line charts



e.g. *banking to* 45° [Heer + Agrawala, 2006]

Previous Work

- aspect-ratio selection for line charts

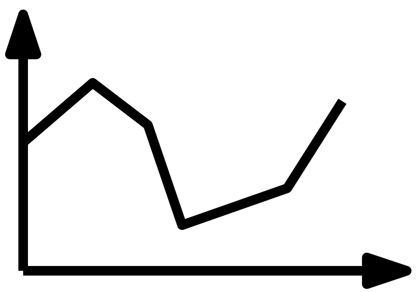


e.g. *banking to* 45° [Heer + Agrawala, 2006]

- [Cleveland et al., 1988] suggest to use *virtual line segments*

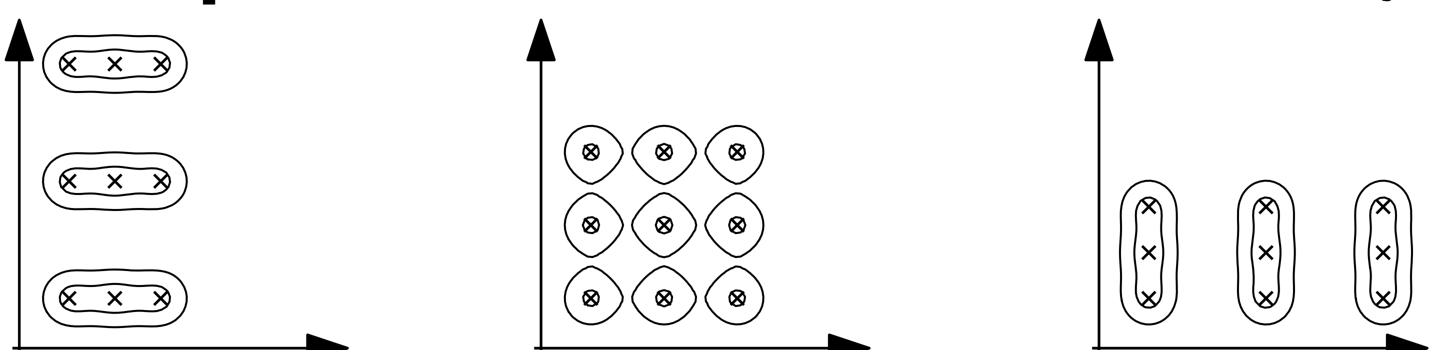
Previous Work

- aspect-ratio selection for line charts



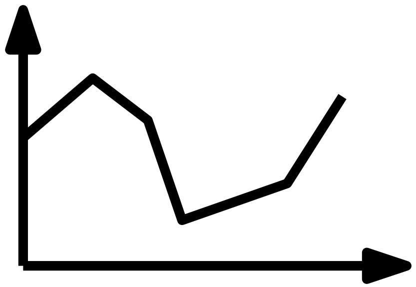
e.g. *banking to* 45° [Heer + Agrawala, 2006]

- [Cleveland et al., 1988] suggest to use *virtual line segments*
- [Talbot et al., 2011]: use contour lines from kernel density estimator



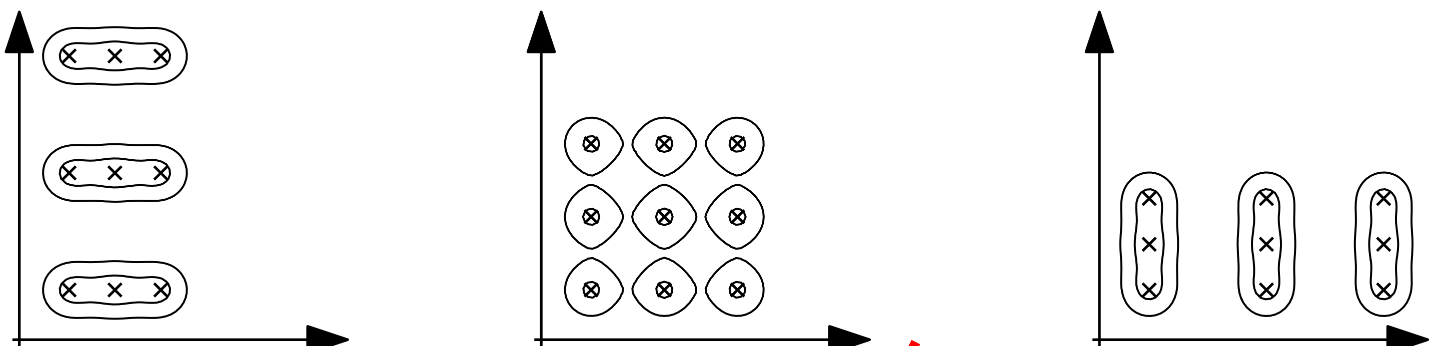
Previous Work

- aspect-ratio selection for line charts



e.g. *banking to* 45° [Heer + Agrawala, 2006]

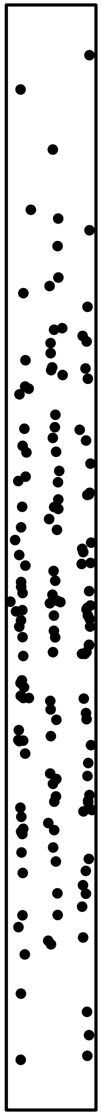
- [Cleveland et al., 1988] suggest to use *virtual line segments*
- [Talbot et al., 2011]: use contour lines from kernel density estimator



results depend on initial aspect ratio

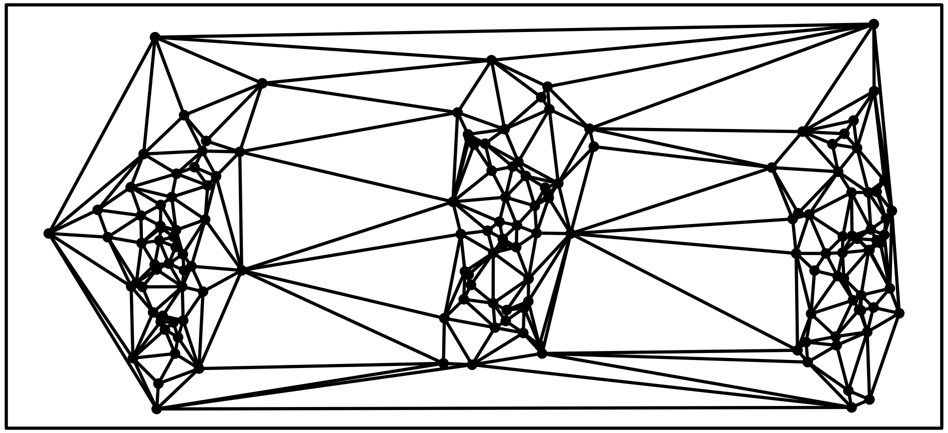
Our Approach

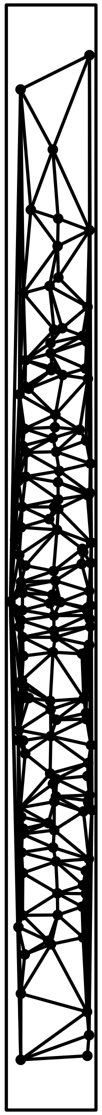
- measure quality of different aspect ratios independently



Our Approach

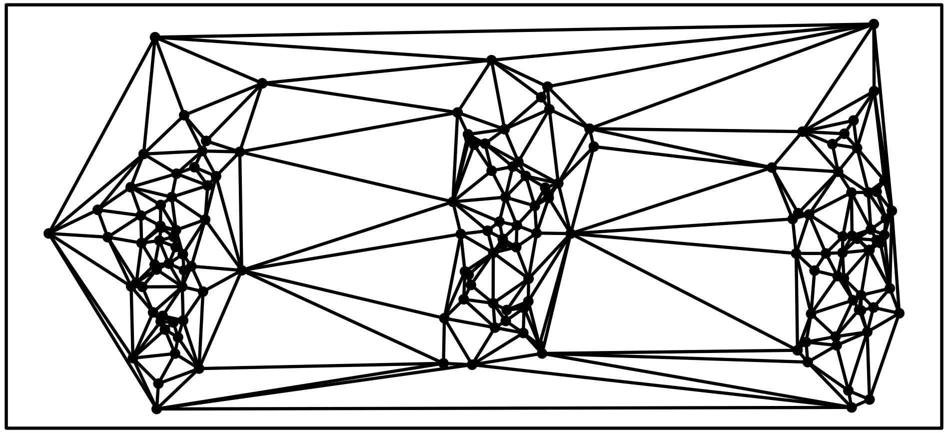
- measure quality of different aspect ratios independently
- use the Delaunay triangulation

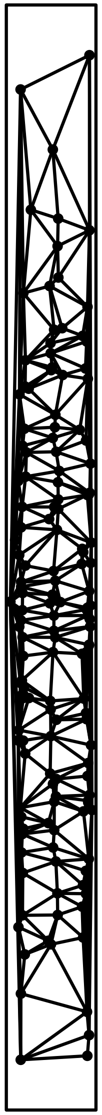




Our Approach

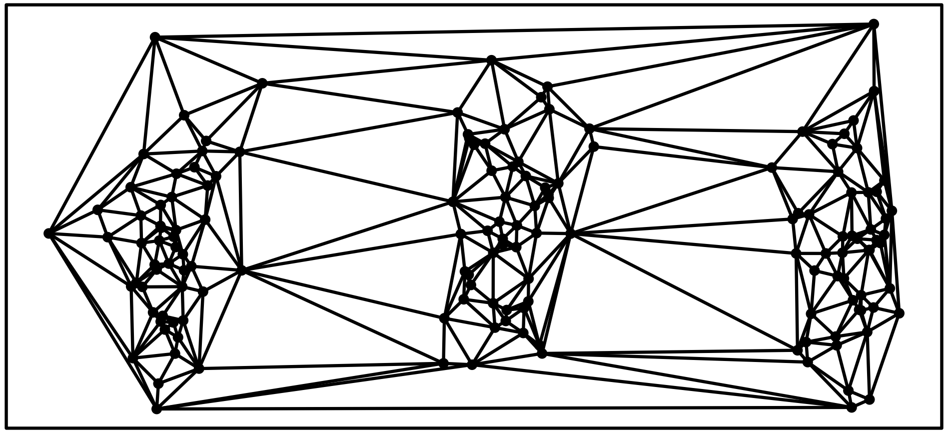
- measure quality of different aspect ratios independently
- use the Delaunay triangulation
- optimization criteria:
  - maximize smallest angle

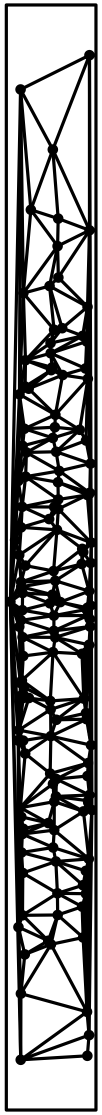




Our Approach

- measure quality of different aspect ratios independently
- use the Delaunay triangulation
- optimization criteria:
  - maximize smallest angle
  - minimize total edge length
  - optimize compactness of triangles
  - etc.





Definitions

- point Set $P = \{p_1, \ldots, p_n\}$
- point $p_i = (x_i, y_i)$

Definitions

- point Set $P = \{p_1, \ldots, p_n\}$
- point $p_i = (x_i, y_i)$
- scale factor s defines aspect-ratio

Definitions

- point Set $P = \{p_1, \ldots, p_n\}$
- point $p_i = (x_i, y_i)$
- scale factor s defines aspect-ratio

$$p_i(s) = (\qquad x_i, \quad s \cdot y_i)$$

Definitions

- point Set $P = \{p_1, \ldots, p_n\}$
- point $p_i = (x_i, y_i)$
- scale factor s defines aspect-ratio

  $$p_i(s) = (1/\sqrt{s} \cdot x_i, \sqrt{s} \cdot y_i)$$

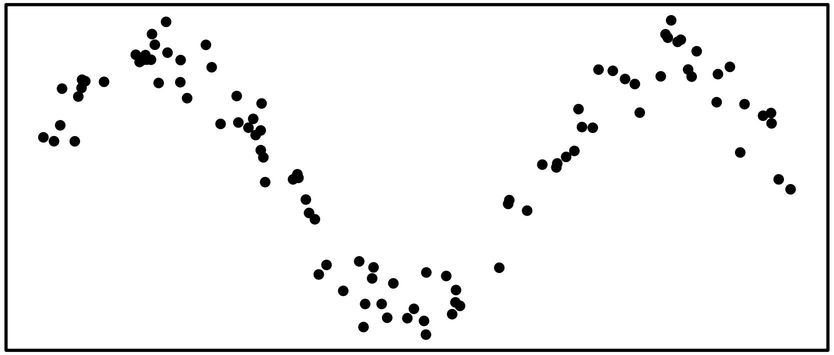
  preserve the area

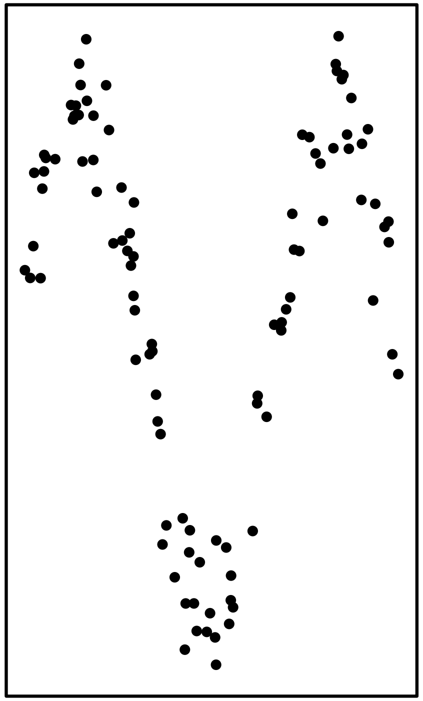
Definitions

- point Set $P = \{p_1, \ldots, p_n\}$
- point $p_i = (x_i, y_i)$
- scale factor s defines aspect-ratio

$$p_i(s) = (1/\sqrt{s} \cdot x_i, \sqrt{s} \cdot y_i)$$

preserve the area



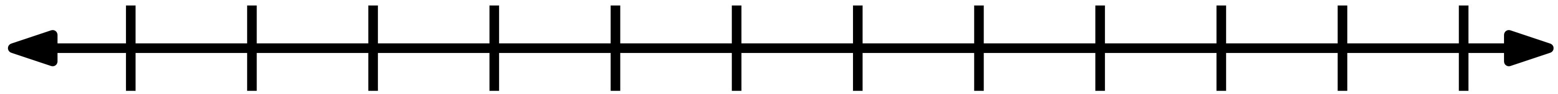


First Idea

aspect ratio s

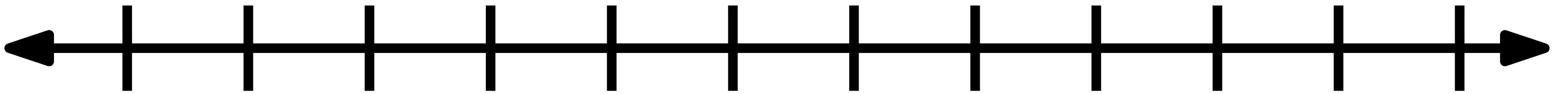
First Idea

aspect ratio s



- discretize into k aspect ratios

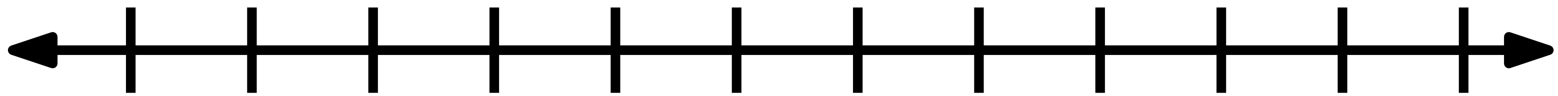
First Idea

aspect ratio s 

- discretize into k aspect ratios
- independently
  - compute Delaunay triangulation
  - measure quality

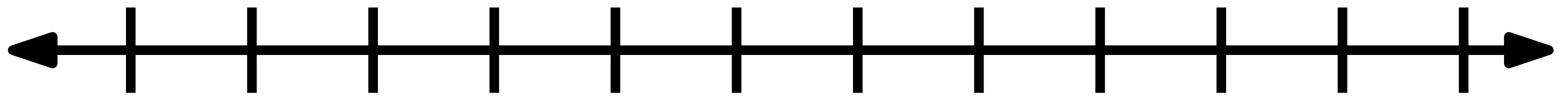
First Idea

aspect ratio s



- discretize into k aspect ratios
- independently
  - compute Delaunay triangulation
  - measure quality
- select best checked aspect ratio

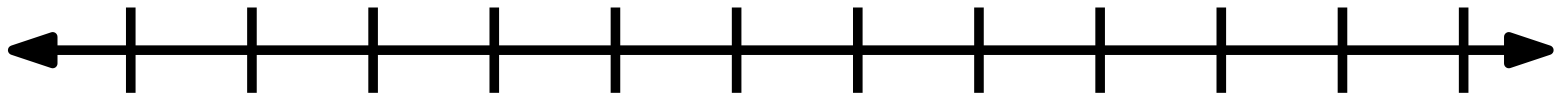
First Idea

aspect ratio s 

- discretize into k aspect ratios
- independently
  - compute Delaunay triangulation $\Theta(n \log n)$
  - measure quality $\Theta(n)$
- select best checked aspect ratio

First Idea

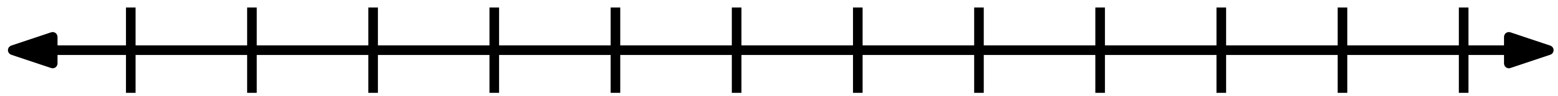
aspect ratio s



- discretize into k aspect ratios
- independently
  - compute Delaunay triangulation $\Theta(n \log n)$
  - measure quality $\Theta(n)$
- select best checked aspect ratio
- runtime: $\Theta(k n \log n)$

First Idea

aspect ratio s



- discretize into k aspect ratios
- independently
  - compute Delaunay triangulation $\Theta(n \log n)$
  - measure quality $\Theta(n)$
- select best checked aspect ratio
- runtime: $\Theta(kn \log n)$
- approximation? which intermediate ratios?

Overview

1. Maintaining the Delaunay Triangulation

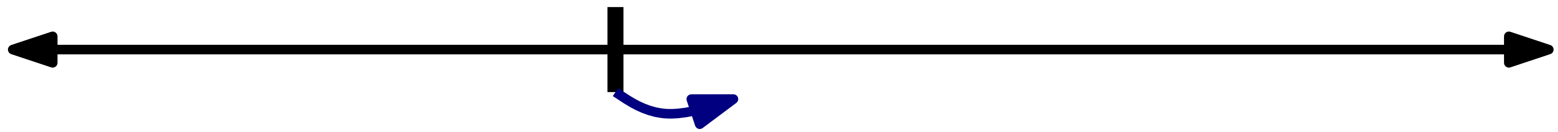
2. Maximizing the Smallest Angle

3. Minimizing the Total Edge Length

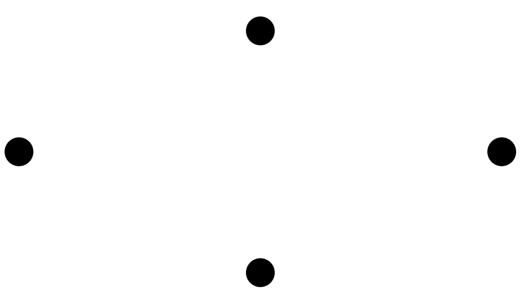
4. Other Optimization Criteria

1. Maintaining the Delaunay Triangulation

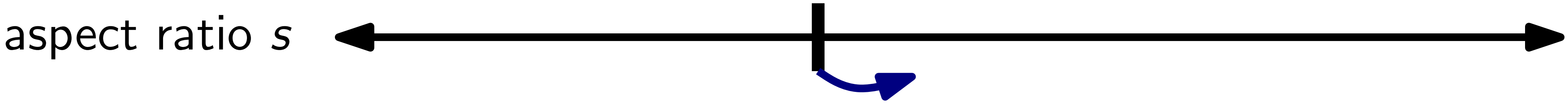
aspect ratio s



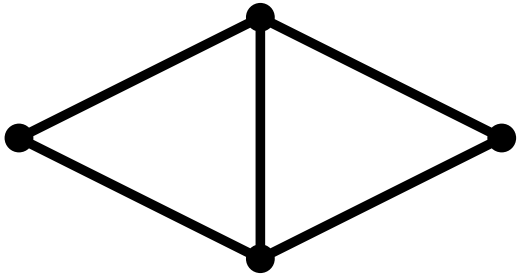
- start at some s



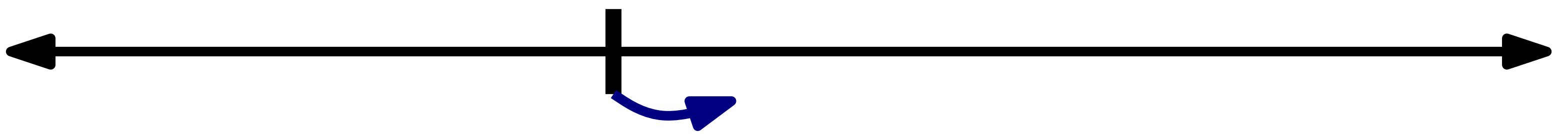
1. Maintaining the Delaunay Triangulation



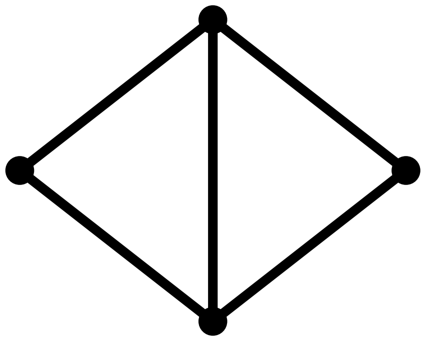
- start at some s
- compute Delaunay triangulation



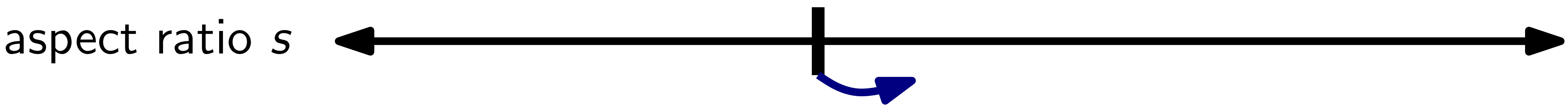
1. Maintaining the Delaunay Triangulation

aspect ratio s 

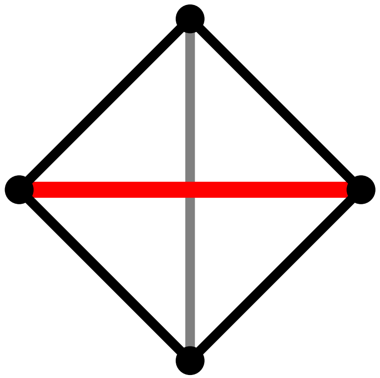
- start at some s
- compute Delaunay triangulation
- continuously change s



1. Maintaining the Delaunay Triangulation

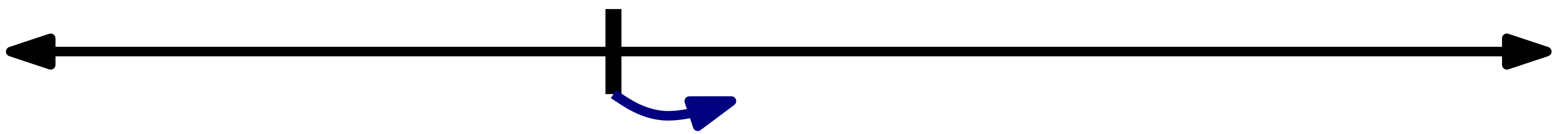


- start at some s
- compute Delaunay triangulation
- continuously change s
- perform *flips* if necessary

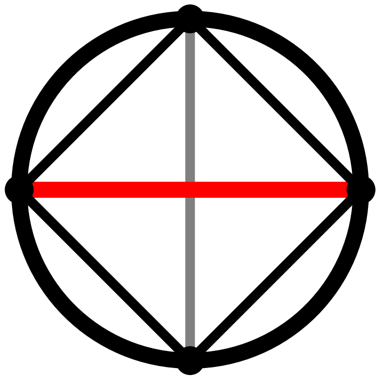


1. Maintaining the Delaunay Triangulation

aspect ratio s



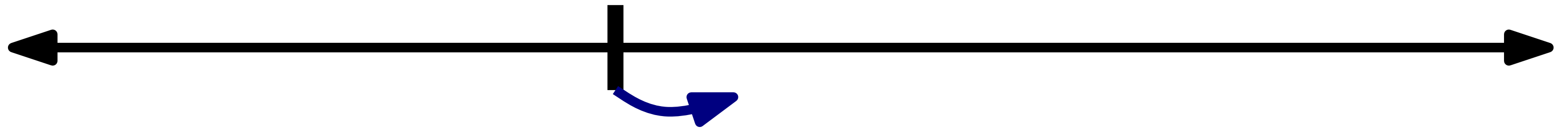
- start at some s
- compute Delaunay triangulation
- continuously change s
- perform *flips* if necessary



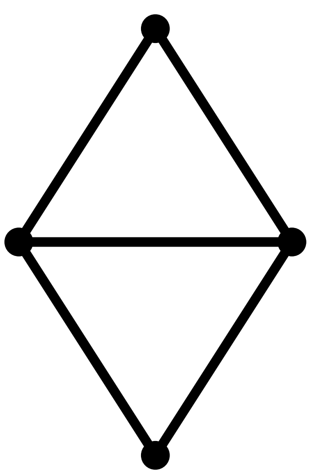
criterion: empty circumcircle of 4 points
easy to check

1. Maintaining the Delaunay Triangulation

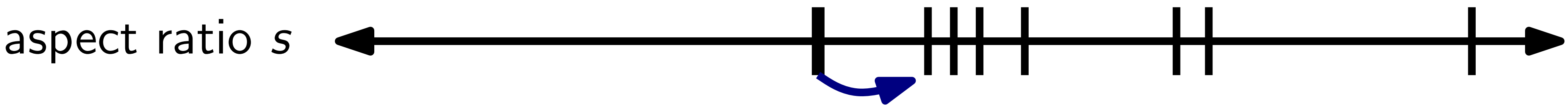
aspect ratio s



- start at some s
- compute Delaunay triangulation
- continuously change s
- perform *flips* if necessary



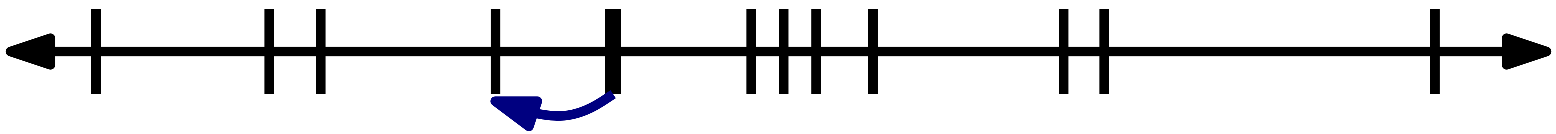
1. Maintaining the Delaunay Triangulation



- start at some s
- compute Delaunay triangulation
- continuously change s
- perform *flips* if necessary
- go through all flips

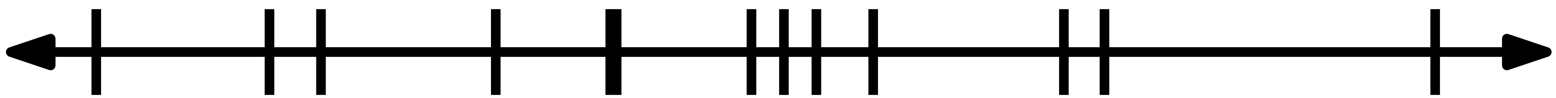
1. Maintaining the Delaunay Triangulation

aspect ratio s



- start at some s
- compute Delaunay triangulation
- continuously change s
- perform *flips* if necessary
- go through all flips

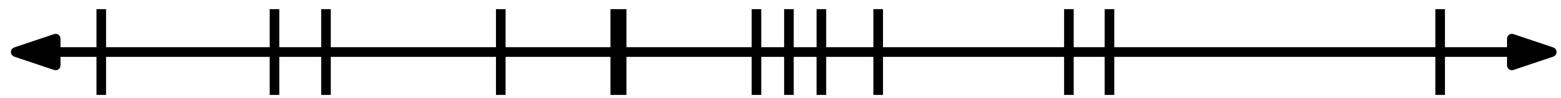
1. Maintaining the Delaunay Triangulation f.

aspect ratio s 

- sweep over possible aspect ratios
- handle event queue of edge flips

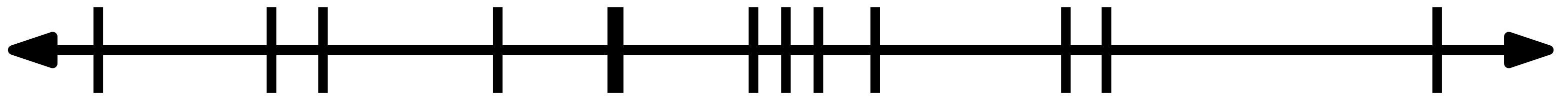
1. Maintaining the Delaunay Triangulation f.

aspect ratio s



- sweep over possible aspect ratios
- handle event queue of edge flips
- update takes $O(\log n)$ time [Roos, 1993]

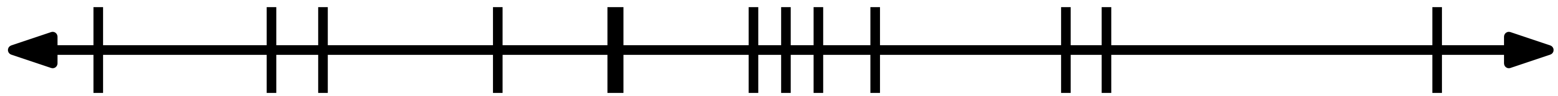
1. Maintaining the Delaunay Triangulation f.

aspect ratio s 

- sweep over possible aspect ratios
- handle event queue of edge flips
- update takes $O(\log n)$ time [Roos, 1993]
- $O(n^{2+\epsilon})$ flips [Rubin, 2012]

1. Maintaining the Delaunay Triangulation f.

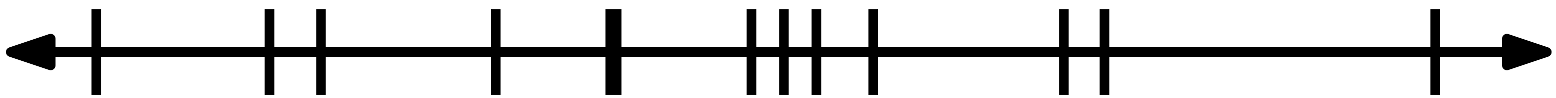
aspect ratio s



- sweep over possible aspect ratios
- handle event queue of edge flips
- update takes $O(\log n)$ time [Roos, 1993]
  here: at most 2 flips per possible edge
- $O(n^{2\cancel{+\varepsilon}})$ flips [Rubin, 2012]

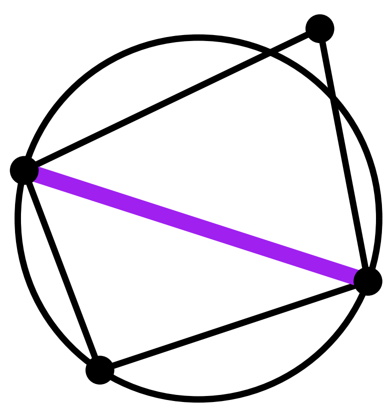
1. Maintaining the Delaunay Triangulation f.

aspect ratio s



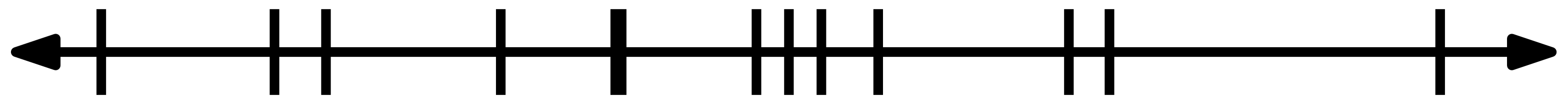
- sweep over possible aspect ratios
- handle event queue of edge flips
- update takes $O(\log n)$ time [Roos, 1993]

  here: at most 2 flips per possible edge
- $O(n^{2\cancel{+\varepsilon}})$ flips [Rubin, 2012]



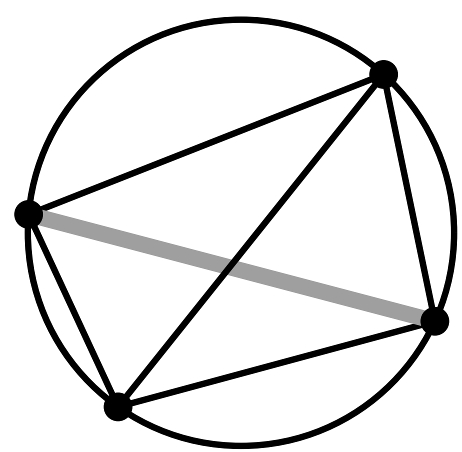
1. Maintaining the Delaunay Triangulation f.

aspect ratio s

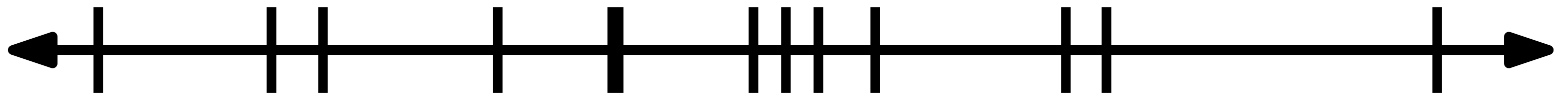


- sweep over possible aspect ratios
- handle event queue of edge flips
- update takes $O(\log n)$ time [Roos, 1993]

  here: at most 2 flips per possible edge
- $O(n^{2\cancel{+\varepsilon}})$ flips [Rubin, 2012]

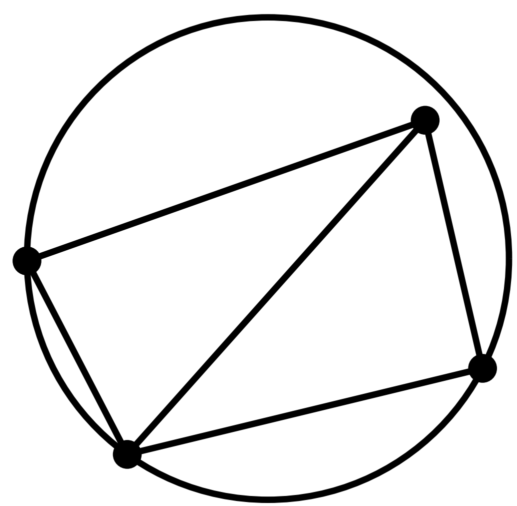


1. Maintaining the Delaunay Triangulation f.

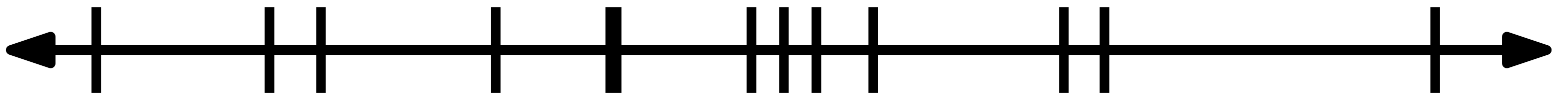
aspect ratio s 

- sweep over possible aspect ratios
- handle event queue of edge flips
- update takes $O(\log n)$ time [Roos, 1993]

  here: at most 2 flips per possible edge
- $O(n^{2\cancel{+\varepsilon}})$ flips [Rubin, 2012]



1. Maintaining the Delaunay Triangulation f.

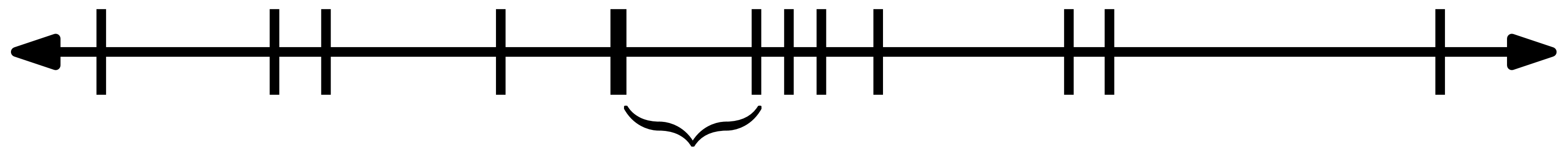
aspect ratio s 

- sweep over possible aspect ratios
- handle event queue of edge flips
- update takes $O(\log n)$ time [Roos, 1993]

  here: at most 2 flips per possible edge
- $O(n^{2\cancel{+\varepsilon}})$ flips [Rubin, 2012]
- total runtime: $O(n^2 \log n)$ for traversing all topologically different Delaunay triangulations

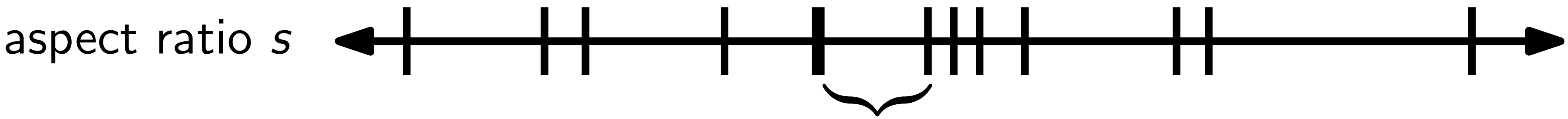
2. Maximizing the Smallest Angle

aspect ratio s

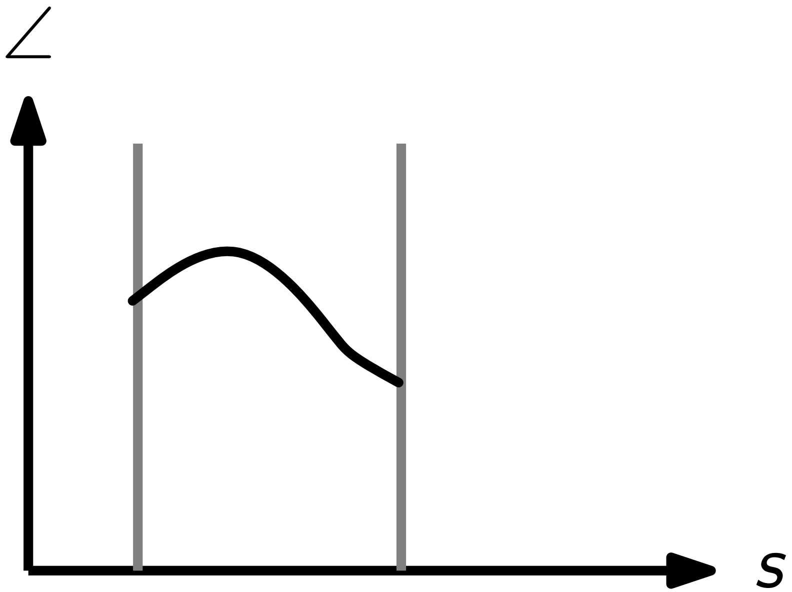


- optimize between event points

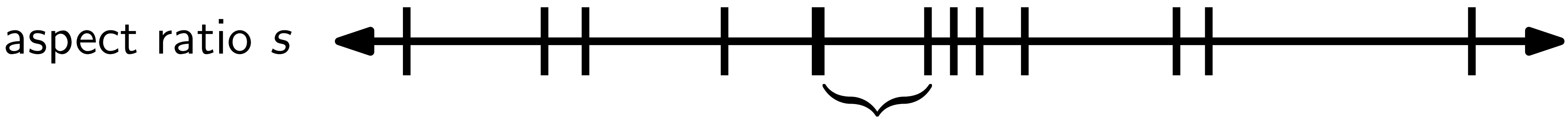
2. Maximizing the Smallest Angle



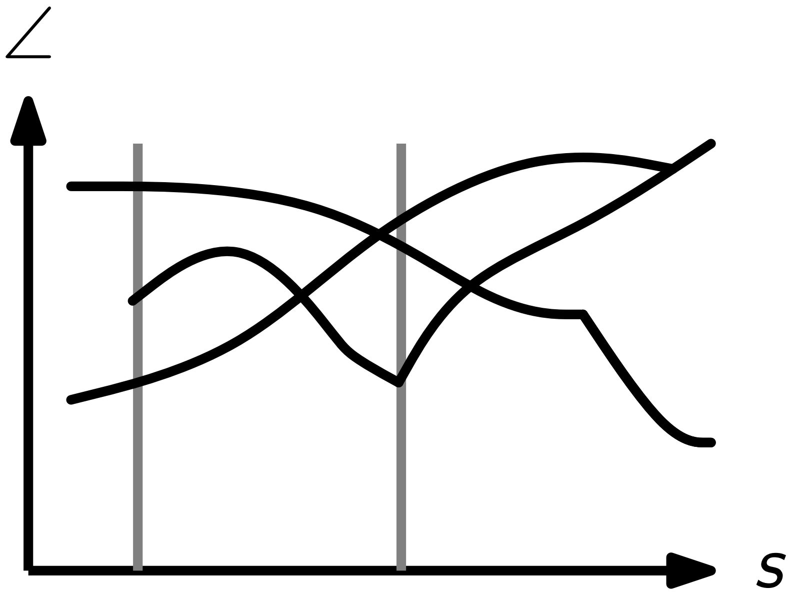
- optimize between event points
- angle α describes function $\alpha(s)$



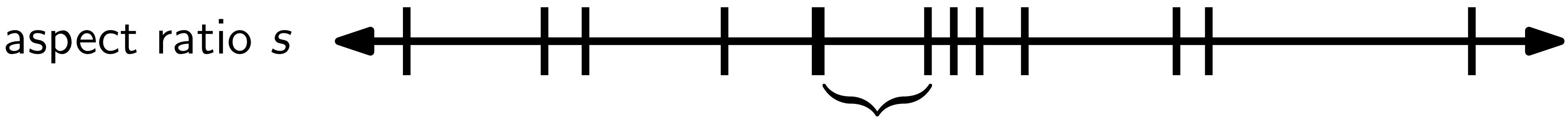
2. Maximizing the Smallest Angle



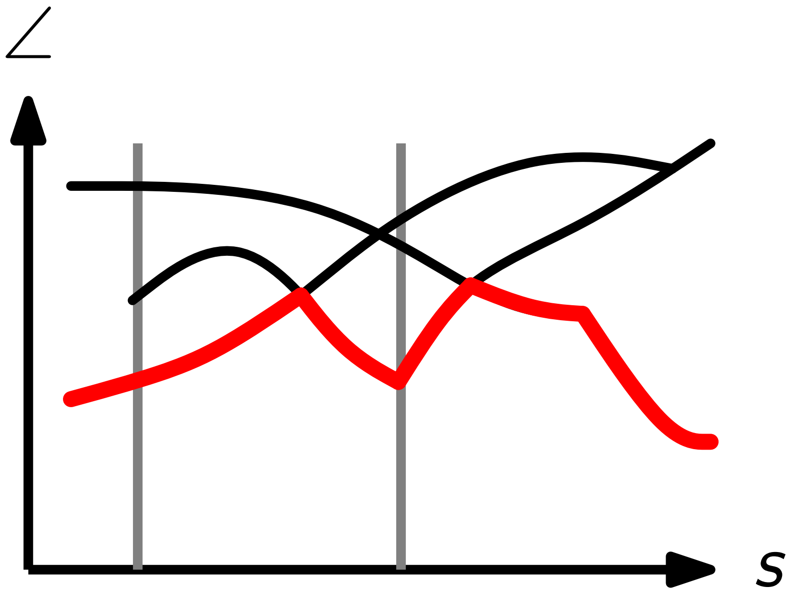
- optimize between event points
- angle α describes function $\alpha(s)$
- put functions together



2. Maximizing the Smallest Angle

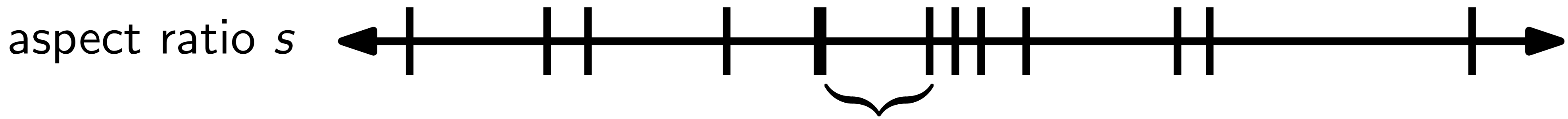


- optimize between event points
- angle α describes function $\alpha(s)$
- put functions together

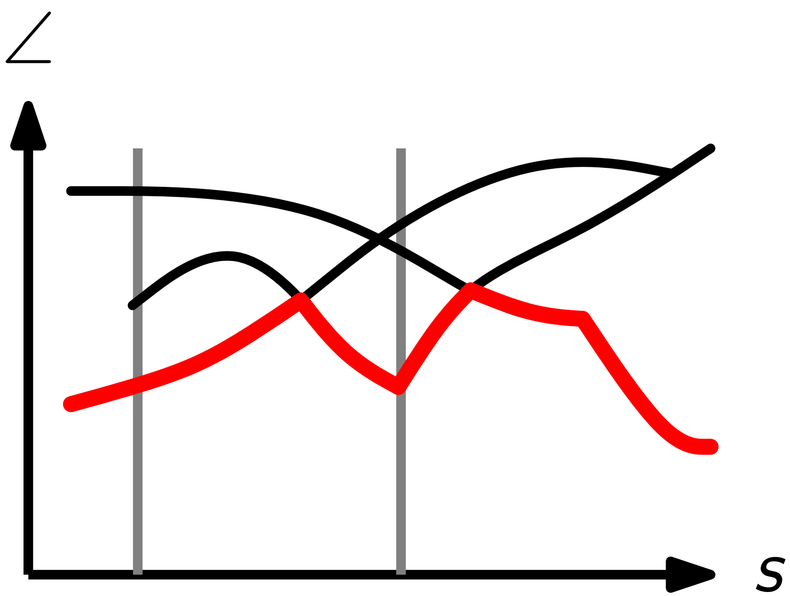


- traverse lower envelope

2. Maximizing the Smallest Angle



- optimize between event points
- angle α describes function $\alpha(s)$
- put functions together



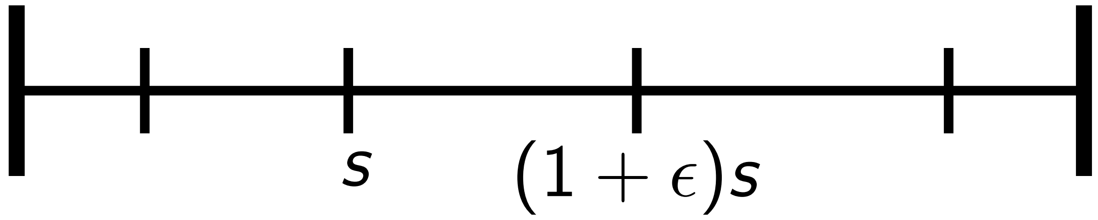
- traverse lower envelope
- Davenport-Schinzel sequences & [Agarwall + Sharir, 1995]: yields globally optimal aspect ratio in $O(n^2 \log n)$ time

3. Minimizing the Total Edge Length

- sum of many functions $\Rightarrow$ previous approach does not work
- find $(1+\epsilon)$-approximation

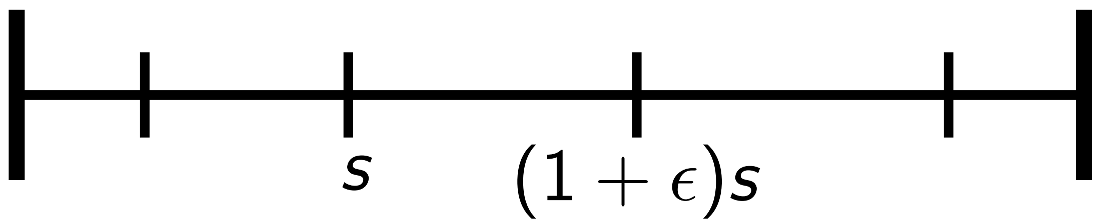
3. Minimizing the Total Edge Length

- sum of many functions $\Rightarrow$ previous approach does not work
- find $(1+\epsilon)$-approximation
- between flips consider $(1+\epsilon)$-intermediate steps



3. Minimizing the Total Edge Length

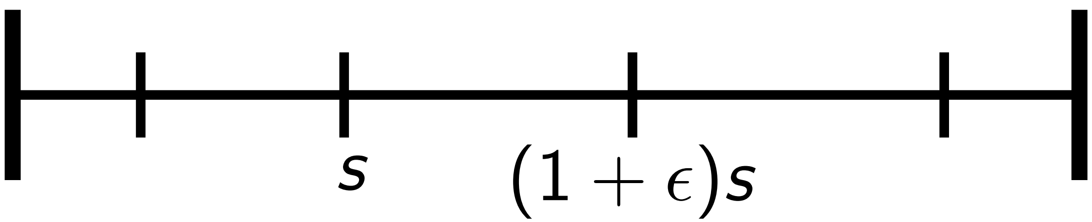
- sum of many functions $\Rightarrow$ previous approach does not work
- find $(1+\epsilon)$-approximation
- between flips consider $(1+\epsilon)$-intermediate steps



- length l_e of edge e within a small intervall:
  $l_e(s(1+\epsilon)) \leq (1+\epsilon)l_e(s)$

3. Minimizing the Total Edge Length

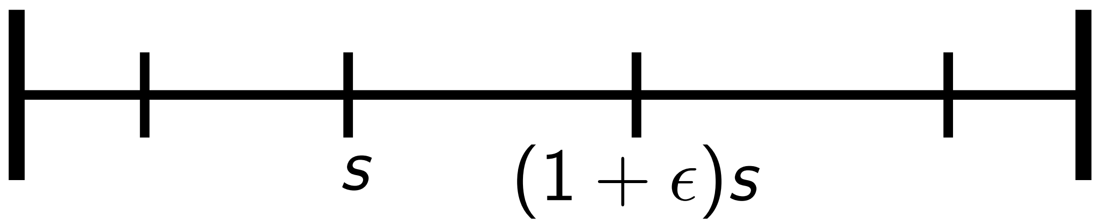
- sum of many functions $\Rightarrow$ previous approach does not work
- find $(1+\epsilon)$-approximation
- between flips consider $(1+\epsilon)$-intermediate steps



- length l_e of edge e within a small intervall:
  $l_e(s(1+\epsilon)) \leq (1+\epsilon)l_e(s)$
- carries over to sum
- find $(1+\epsilon)$-approximation in $O(n^3 + n \cdot \frac{1}{\log(1+\epsilon)})$ time

3. Minimizing the Total Edge Length

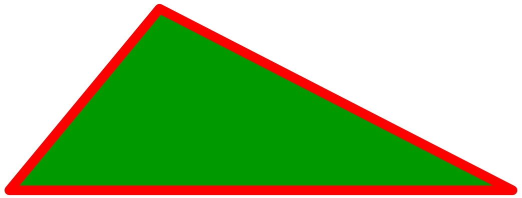
- sum of many functions $\Rightarrow$ previous approach does not work
- find $(1+\epsilon)$-approximation
- between flips consider $(1+\epsilon)$-intermediate steps



- length l_e of edge e within a small intervall:
  $l_e(s(1+\epsilon)) \leq (1+\epsilon) l_e(s)$
- carries over to sum
- find $(1+\epsilon)$-approximation in $O(n^3 + n \cdot \frac{1}{\log(1+\epsilon)})$ time
- also works for other optimization criteria

4. Other Optimization Criteria

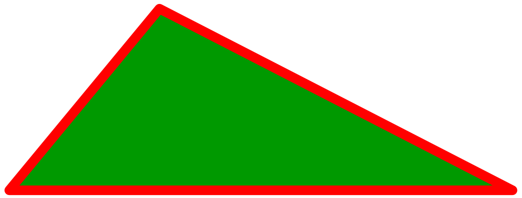
- maximize total compactness of triangles



$$\frac{\sqrt{\text{area}}}{\text{perimeter}}$$

4. Other Optimization Criteria

- maximize total compactness of triangles



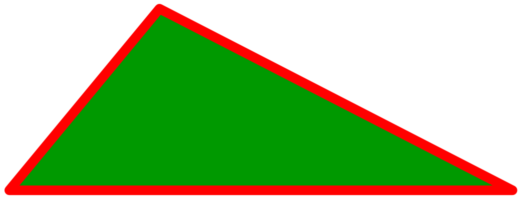
$$\frac{\sqrt{\text{area}}}{\text{perimeter}}$$

- minimize total uncompactness of triangles

$$\frac{\text{perimeter}}{\sqrt{\text{area}}}$$

4. Other Optimization Criteria

- maximize total compactness of triangles



$$\frac{\sqrt{\text{area}}}{\text{perimeter}}$$

- minimize total uncompactness of triangles

$$\frac{\text{perimeter}}{\sqrt{\text{area}}}$$

- more:
  - maximize mean inradius
  - minimize sum of squared angles

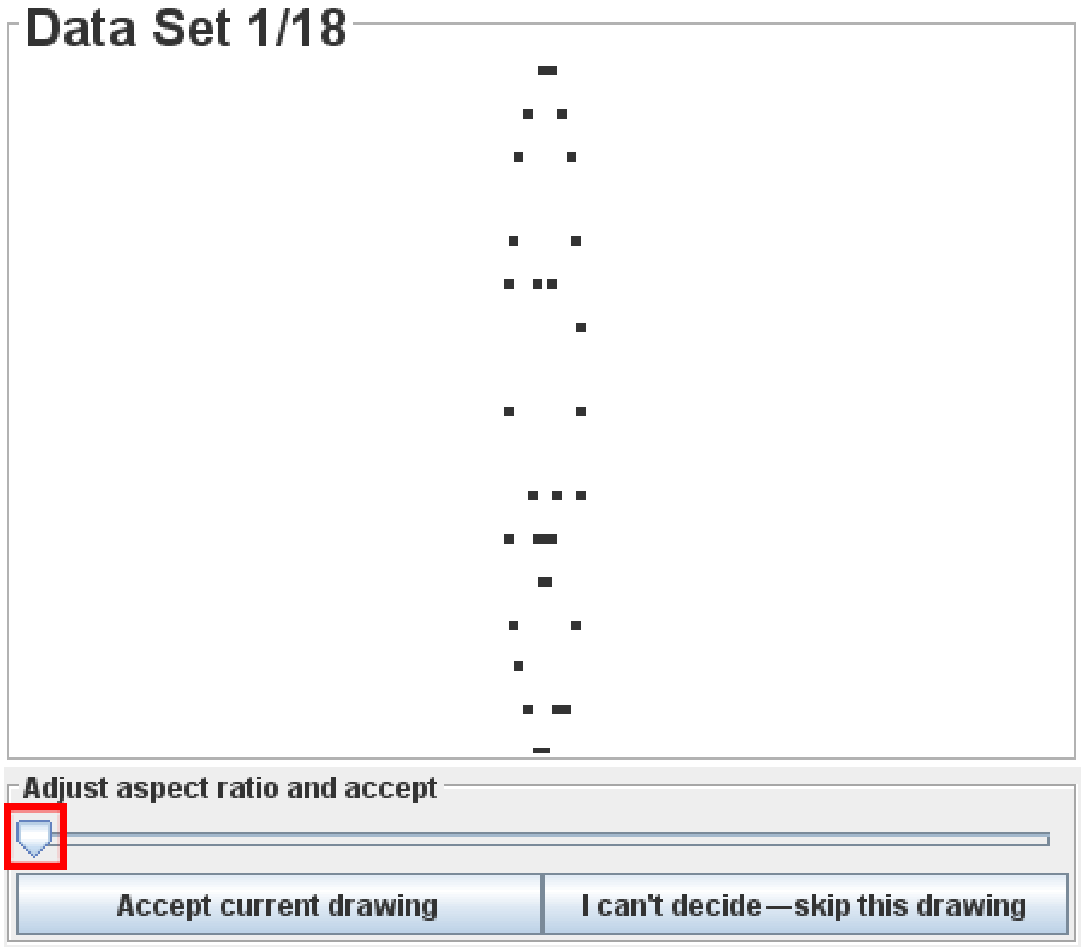
User Study

- What do users want?
- let participants choose

Please participate: **www1.informatik.uni-wuerzburg.de/scatterplots**

User Study

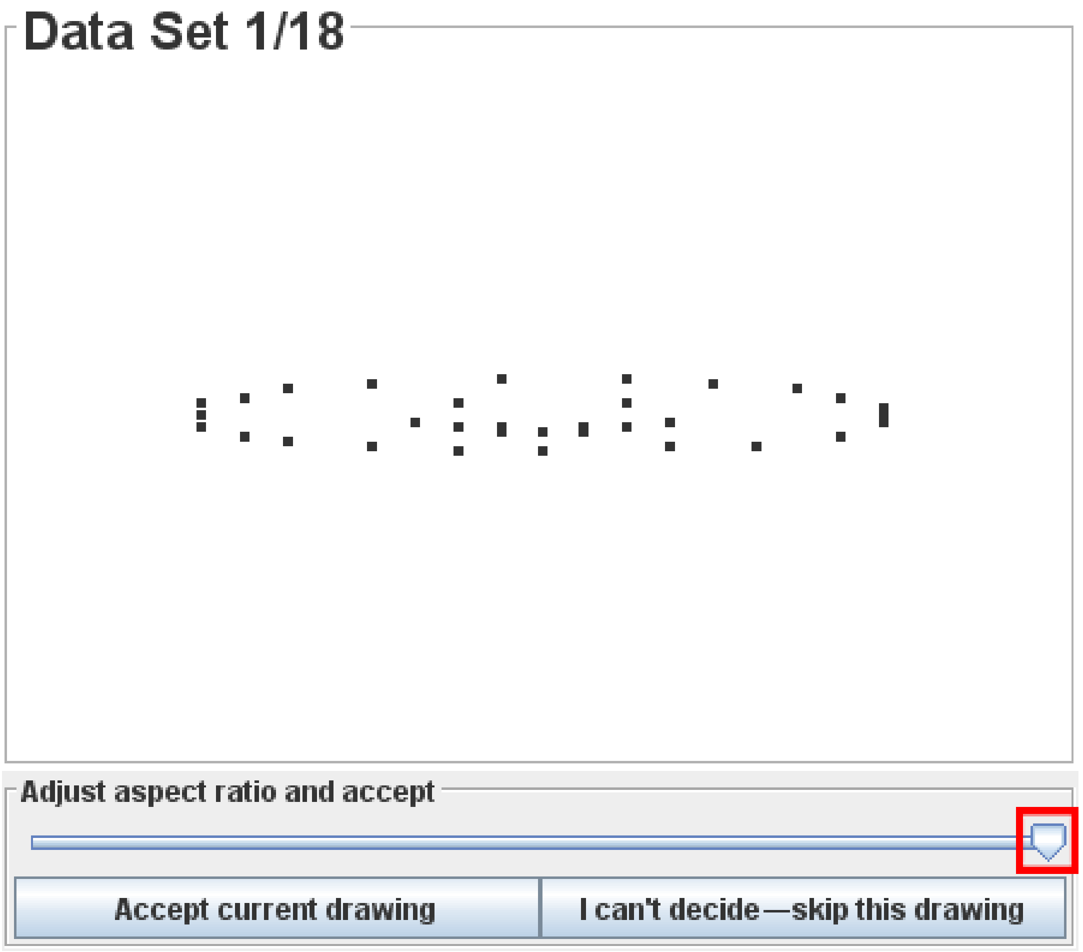
- What do users want?
- let participants choose



Please participate: **www1.informatik.uni-wuerzburg.de/scatterplots**

User Study

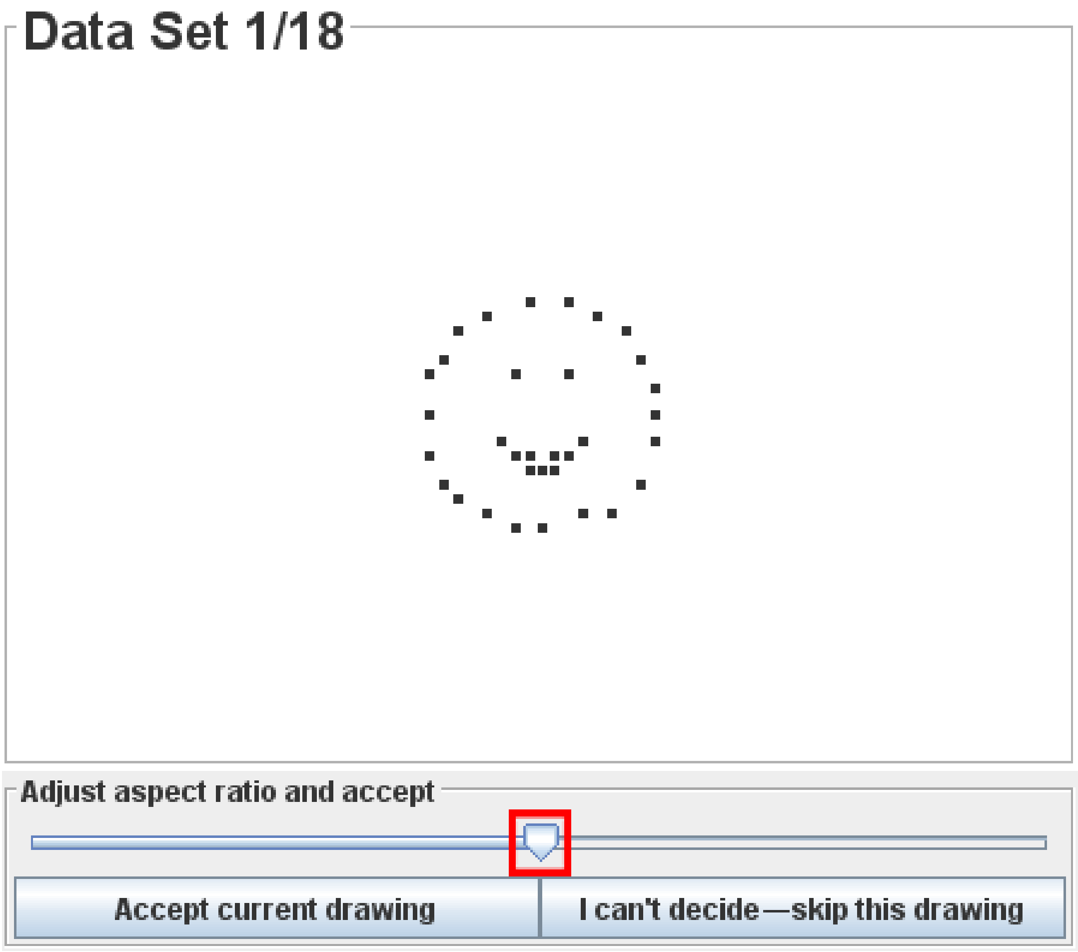
- What do users want?
- let participants choose



Please participate: **www1.informatik.uni-wuerzburg.de/scatterplots**

User Study

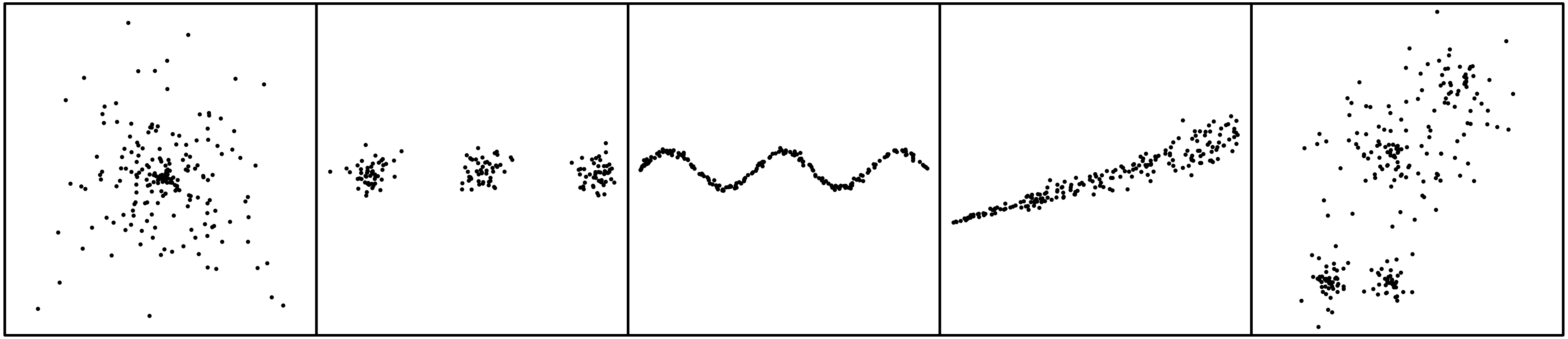
- What do users want?
- let participants choose



Please participate: **www1.informatik.uni-wuerzburg.de/scatterplots**

User Study

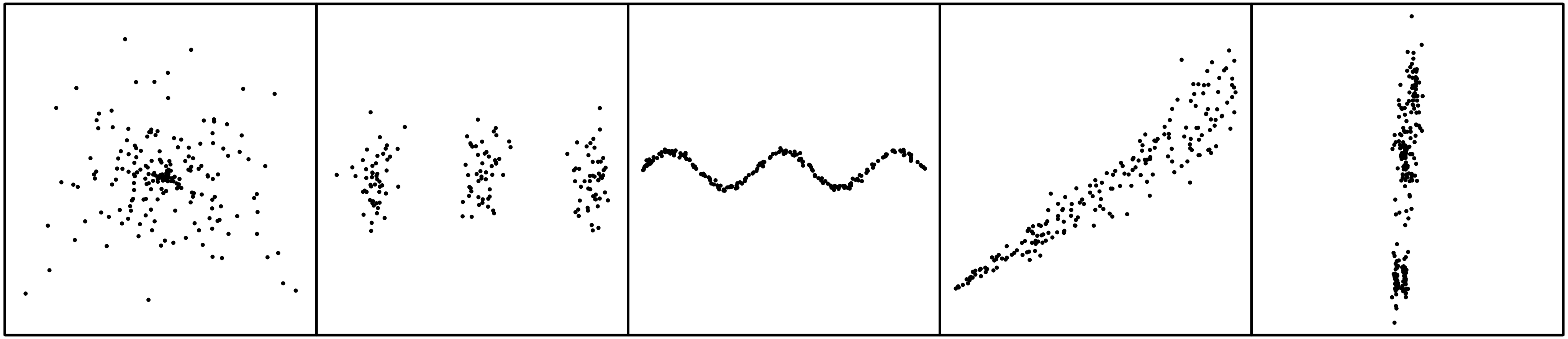
- What do users want?
- let participants choose
- 18 tested instances, e.g. . . .



Please participate: **www1.informatik.uni-wuerzburg.de/scatterplots**

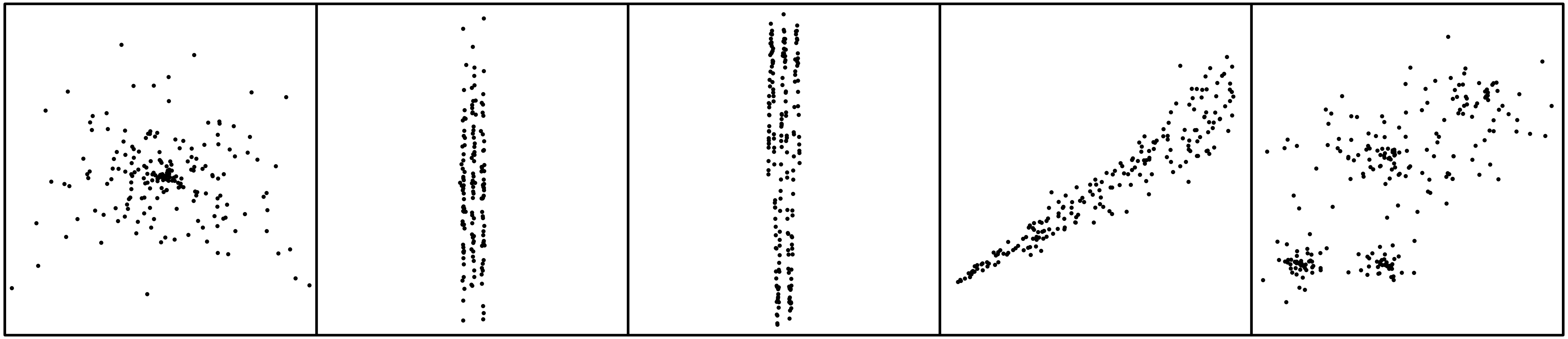
Test Results

- maximize minimum angle



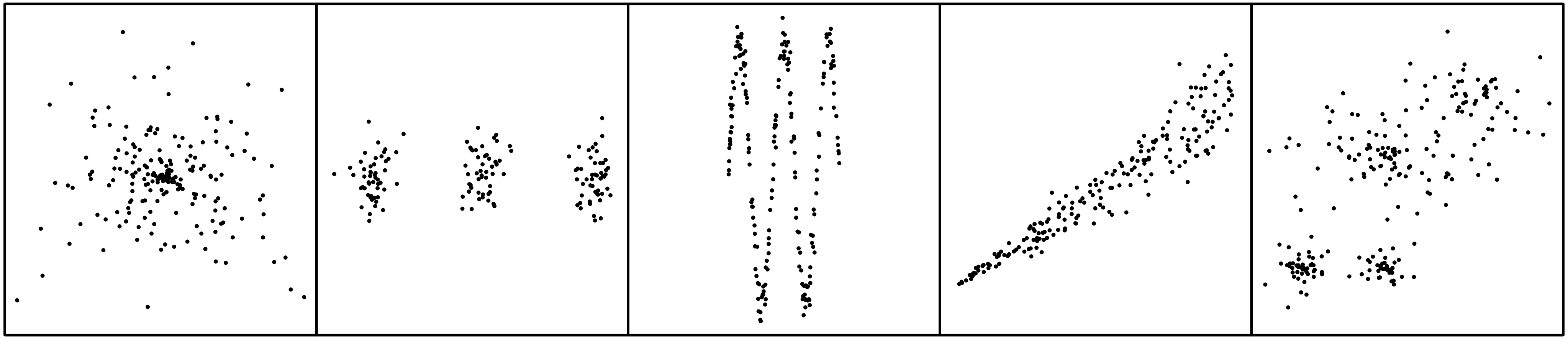
Test Results

- maximize minimum angle
- maximize mean inradius



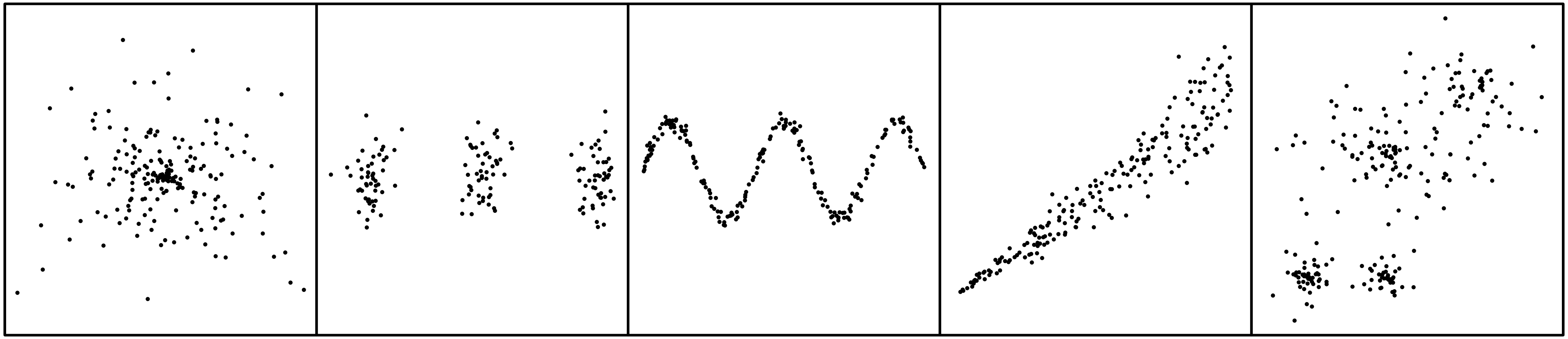
Test Results

- maximize minimum angle
- maximize mean inradius
- maximize total compactness of triangles



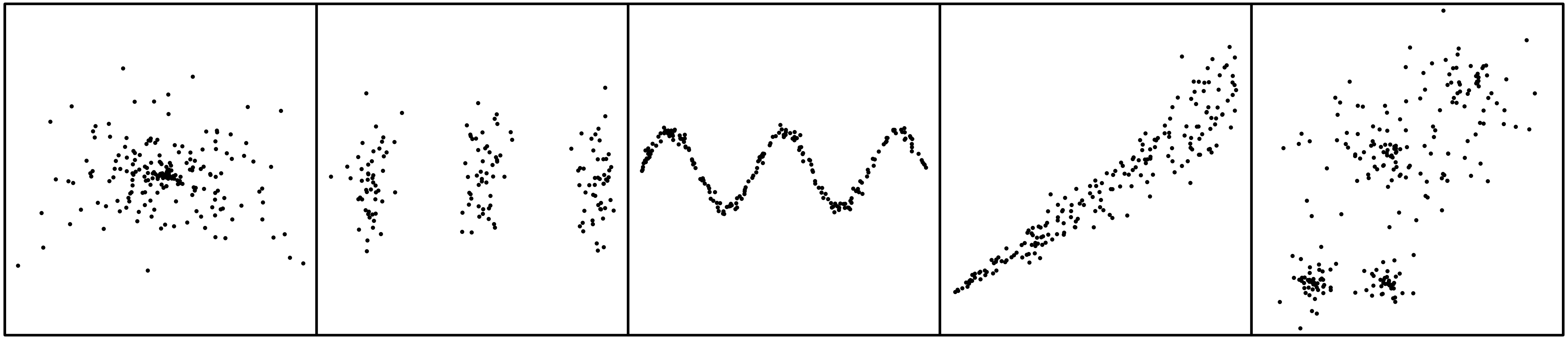
Test Results

- maximize minimum angle
- maximize mean inradius
- maximize total compactness of triangles
- minimize total uncompactness of triangles



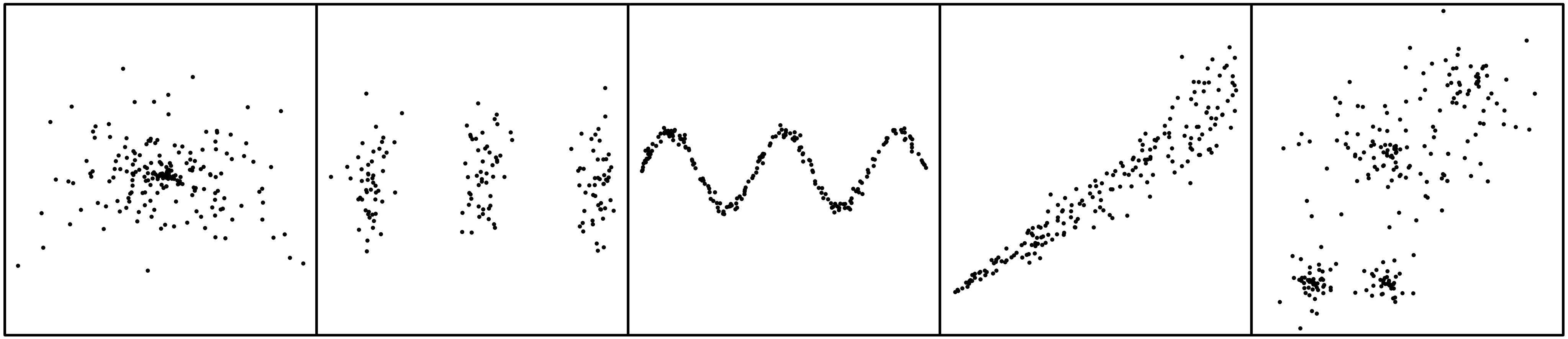
Test Results

- maximize minimum angle
- maximize mean inradius
- maximize total compactness of triangles
- minimize total uncompactness of triangles
- minimize total edge length



Test Results

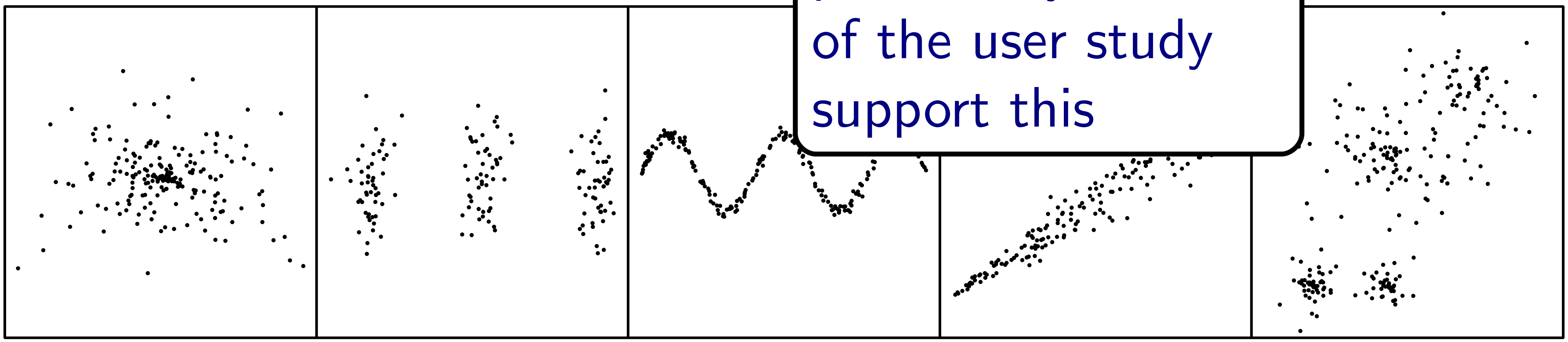
- maximize minimum angle ✗
- maximize mean inradius ✗
- maximize total compactness of triangles ✗
- minimize total uncompactness of triangles ✓
- minimize total edge length ✓



Test Results

- maximize minimum angle ✗
- maximize mean inradius ✗
- maximize total compactness of triangles ✗
- minimize total uncompactness of triangles ✓
- minimize total edge length ✓

preliminary results of the user study support this



Conclusion

- Delaunay triangulation helps to optimize scatter plots

Please participate in our user study!

www1.informatik.uni-wuerzburg.de/scatterplots

Conclusion

- Delaunay triangulation helps to optimize scatter plots
- maintaining the Delaunay triangulation is fast

Please participate in our user study!

www1.informatik.uni-wuerzburg.de/scatterplots

Conclusion

- Delaunay triangulation helps to optimize scatter plots
- maintaining the Delaunay triangulation is fast
- more than one good quality measure

Please participate in our user study!

www1.informatik.uni-wuerzburg.de/scatterplots