Jack Spalding-Jamieson (Jack S-J)
jacksj@uwaterloo.ca

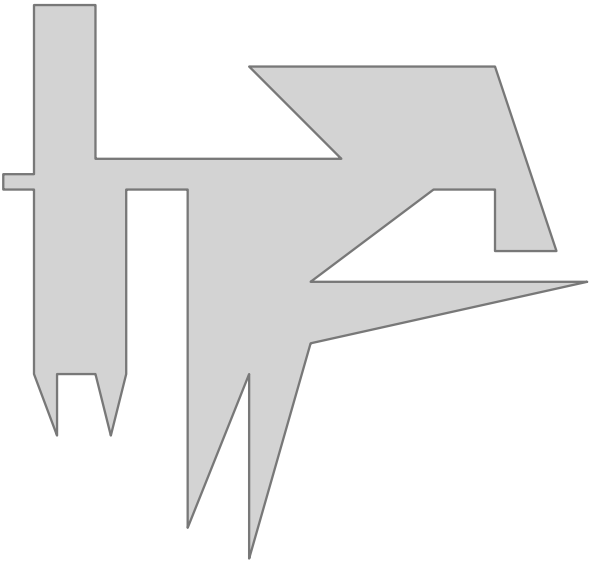
Independent

The Analytic Arc Cover Problem and its Applications to Contiguous Art Gallery, Polygon Separation, and Shape Carving

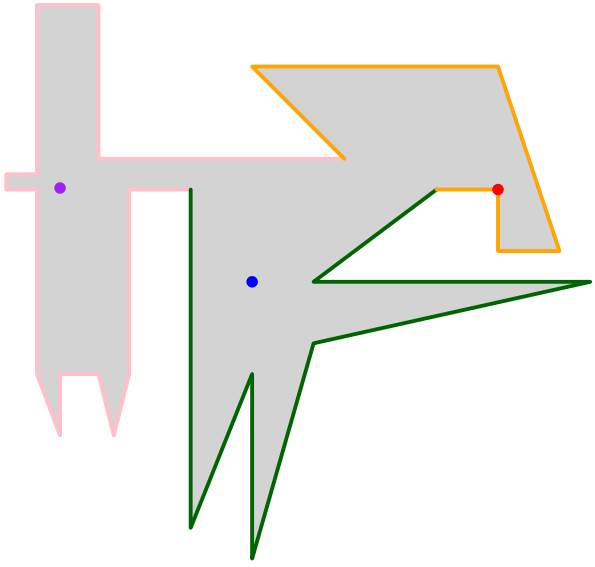
Joint work with Eliot Robson and Da Wei Zheng

EuroCG 2025

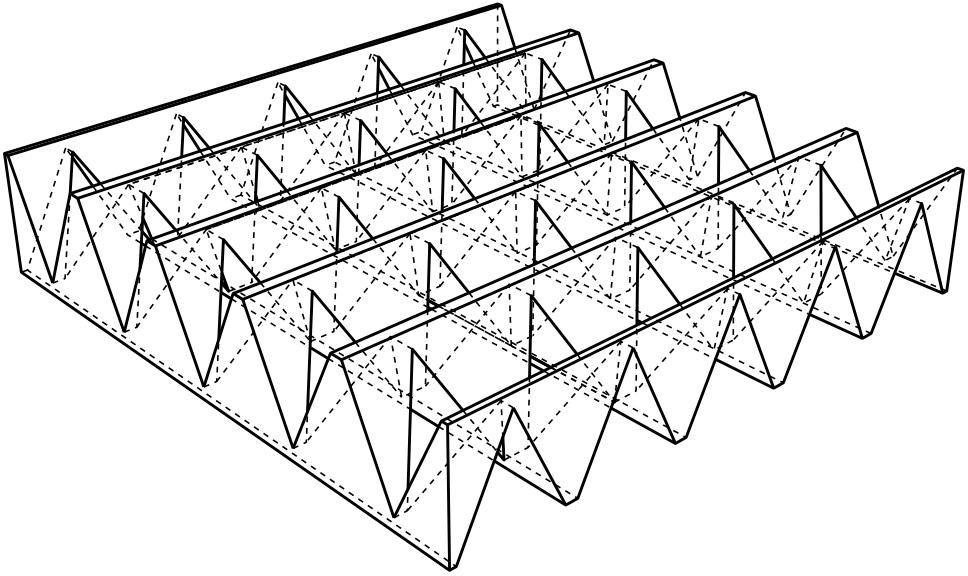
Contiguous Art Gallery



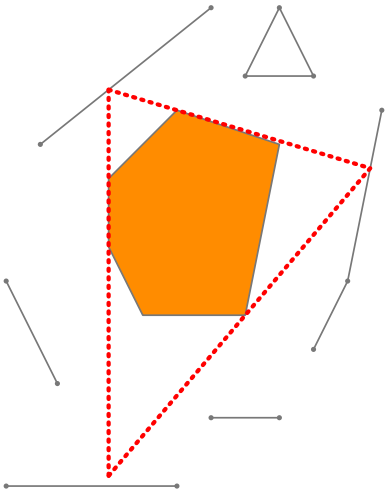
Contiguous Art Gallery



Other Problems: Half-Plane Carving

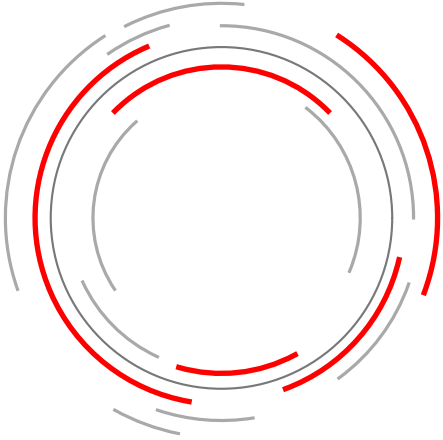


Other Problems: Line Segment/Convex Polygon Separation by a Polygon

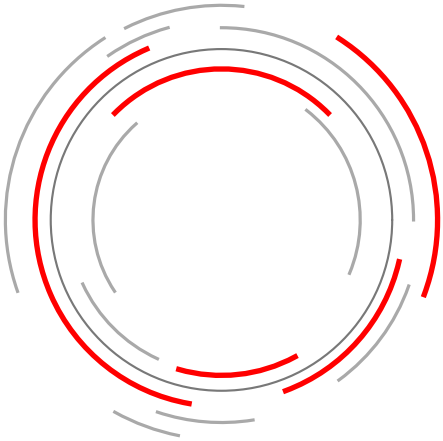


Arc Cover

Arc Cover

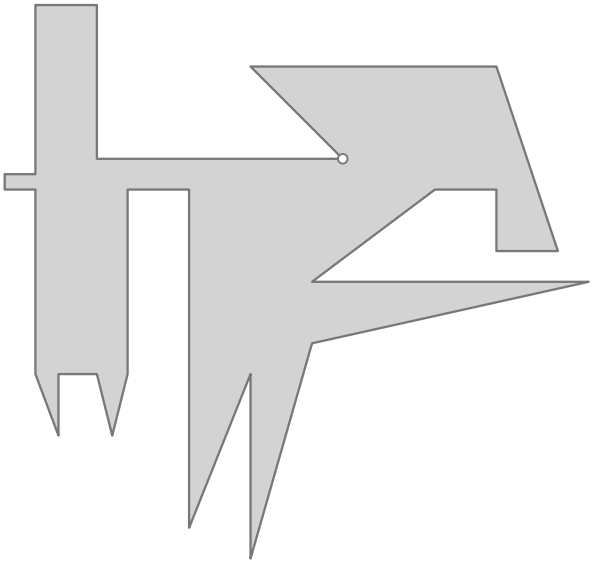


Arc Cover

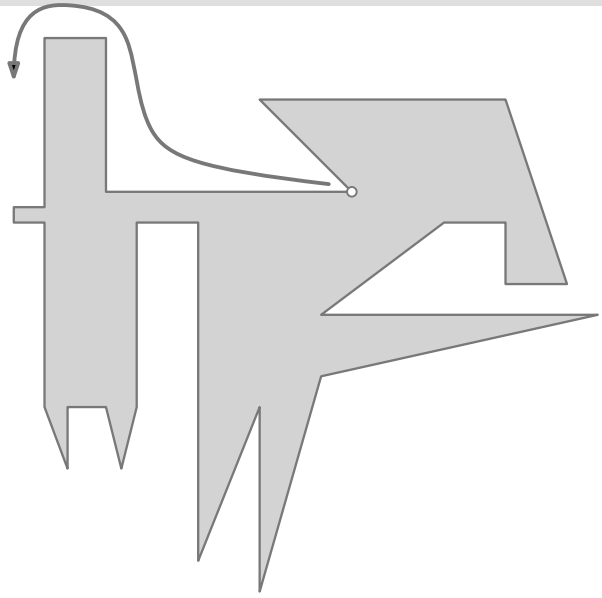


∞ arcs?

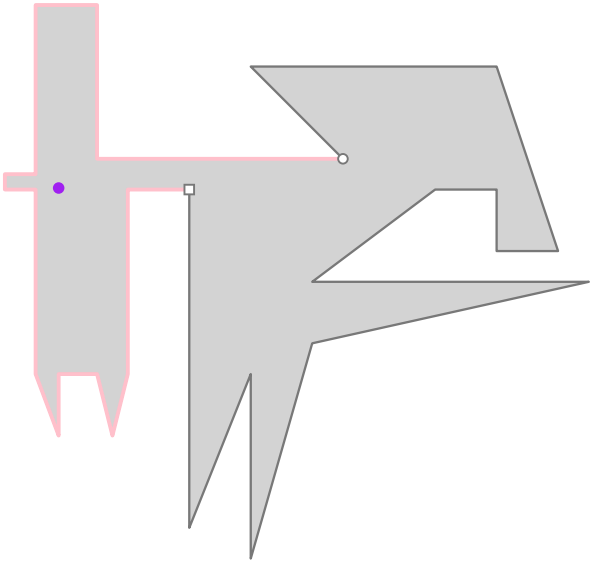
Greedy Next-Generator



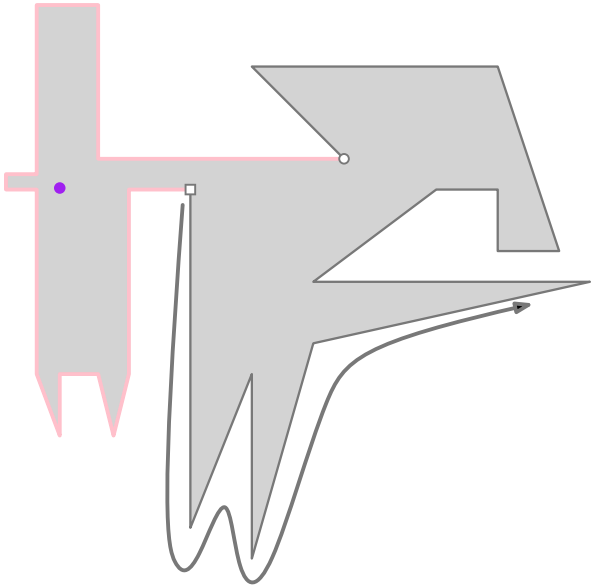
Greedy Next-Generator



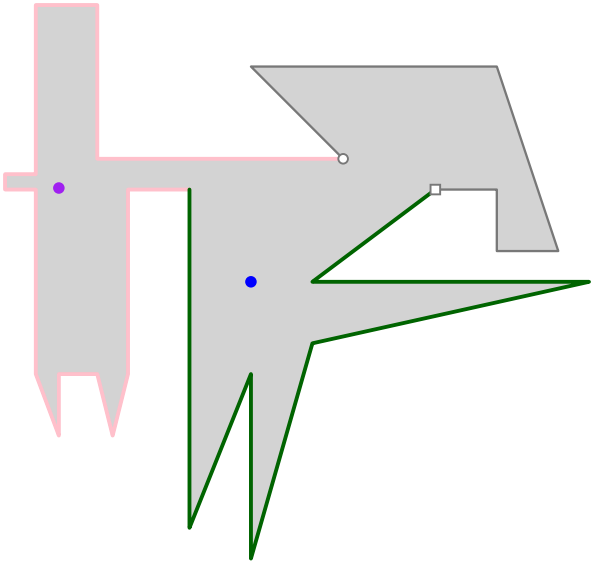
Greedy Next-Generator



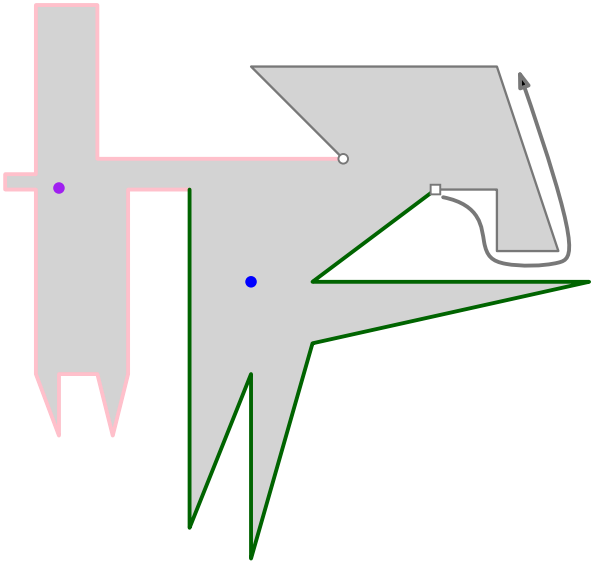
Greedy Next-Generator



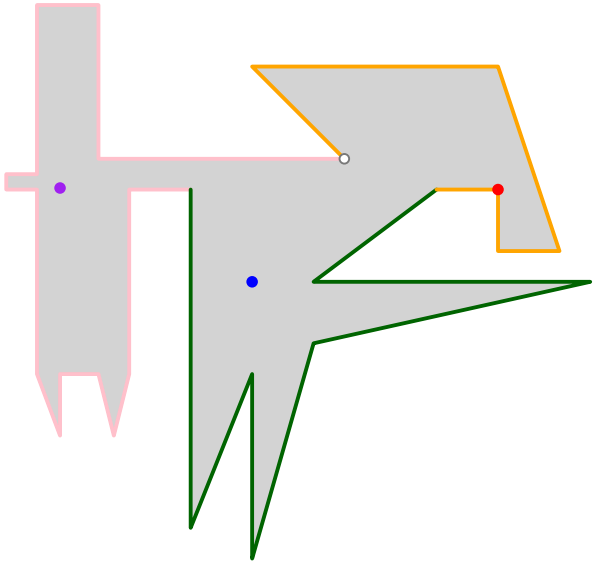
Greedy Next-Generator



Greedy Next-Generator



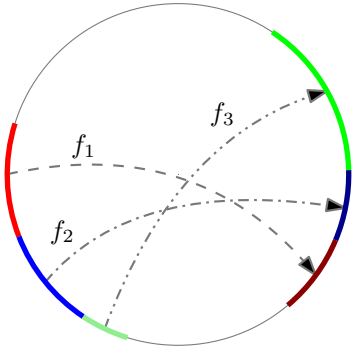
Greedy Next-Generator



Greedy Next-Generator — Function

Greedy next-generator is a function. It can even be computed in polynomial time!
Need something stronger: Piecewise linear rational representation

Next-Generator with Piecewise Linear Rational Representation



$$f_1 = \frac{ax+b}{cx+d},\ f_2 = \frac{a'x+b'}{c'x+d'},\ f_3 = \frac{a''x+b''}{c''x+d''}$$

Bounds: $[a_0 + b_0\sqrt{c_0},\ a_1 + b_1\sqrt{c_1}),\ [a_1 + b_1\sqrt{c_1}, a_2 + b_2\sqrt{c_2}),\ [a_2 + b_2\sqrt{c_2}, a_3 + b_3\sqrt{c_3}), \ldots$

Using a piecewise linear rational representation

Input: f (as a piecewise linear rational function)

Algorithm: Compose f with itself until $f^{(k)}$ exceeds a full loop somewhere.

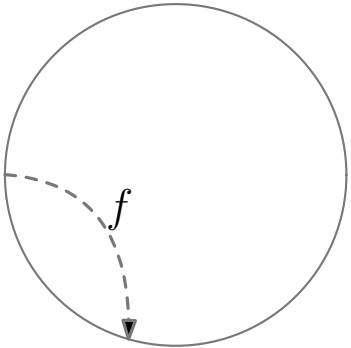
Output: k

Using a piecewise linear rational representation

Input: f (as a piecewise linear rational function)

Algorithm: Compose f with itself until $f^{(k)}$ exceeds a full loop somewhere.

Output: k

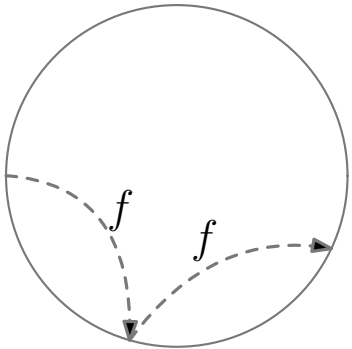


Using a piecewise linear rational representation

Input: f (as a piecewise linear rational function)

Algorithm: Compose f with itself until $f^{(k)}$ exceeds a full loop somewhere.

Output: k

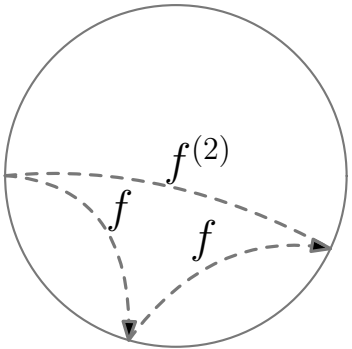


Using a piecewise linear rational representation

Input: f (as a piecewise linear rational function)

Algorithm: Compose f with itself until $f^{(k)}$ exceeds a full loop somewhere.

Output: k

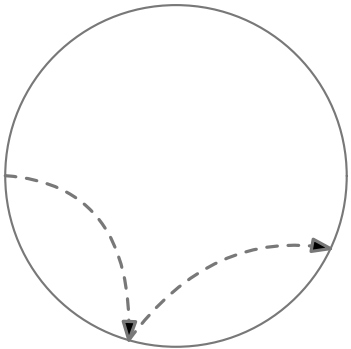


Using a piecewise linear rational representation

Input: f (as a piecewise linear rational function)

Algorithm: Compose f with itself until $f^{(k)}$ exceeds a full loop somewhere.

Output: k

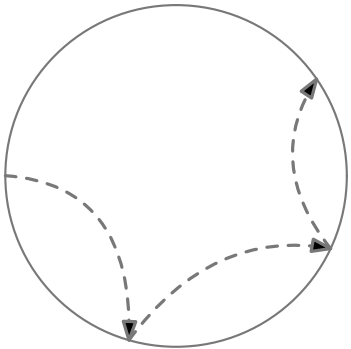


Using a piecewise linear rational representation

Input: f (as a piecewise linear rational function)

Algorithm: Compose f with itself until $f^{(k)}$ exceeds a full loop somewhere.

Output: k

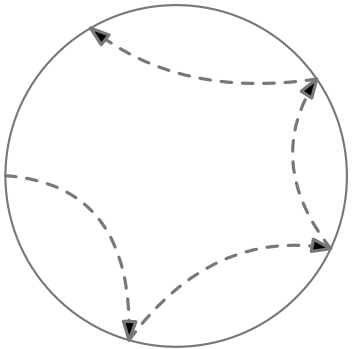


Using a piecewise linear rational representation

Input: f (as a piecewise linear rational function)

Algorithm: Compose f with itself until $f^{(k)}$ exceeds a full loop somewhere.

Output: k

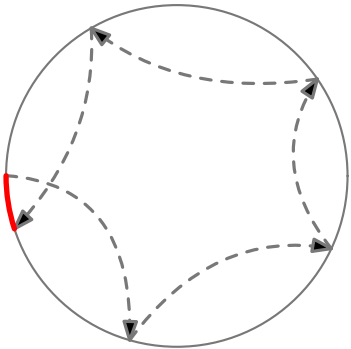


Using a piecewise linear rational representation

Input: f (as a piecewise linear rational function)

Algorithm: Compose f with itself until $f^{(k)}$ exceeds a full loop somewhere.

Output: k

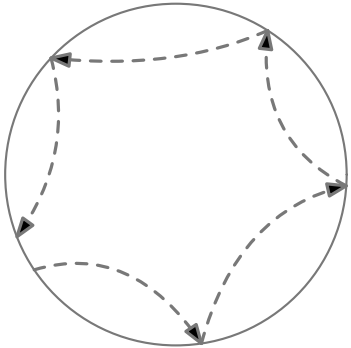


Using a piecewise linear rational representation

Input: f (as a piecewise linear rational function)

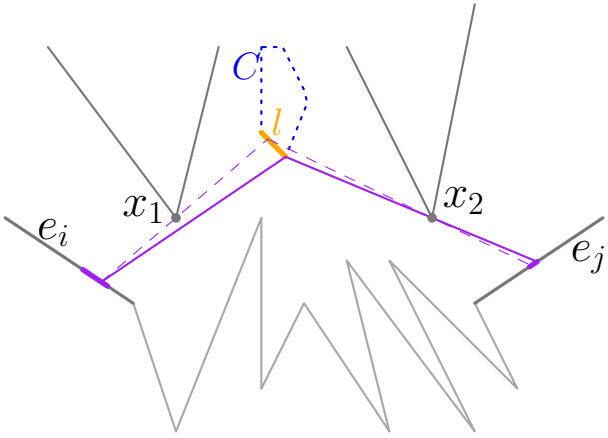
Algorithm: Compose f with itself until $f^{(k)}$ exceeds a full loop somewhere.

Output: k



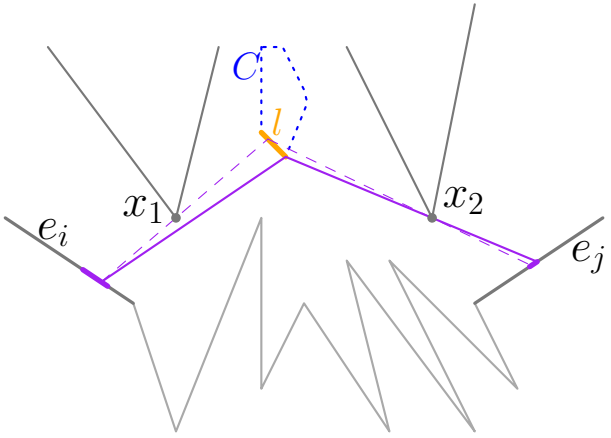
Need to test $f^{(k)}$ everywhere.

Computing a piecewise linear rational function: Contiguous Art Gallery



Input: Left
Output: Right

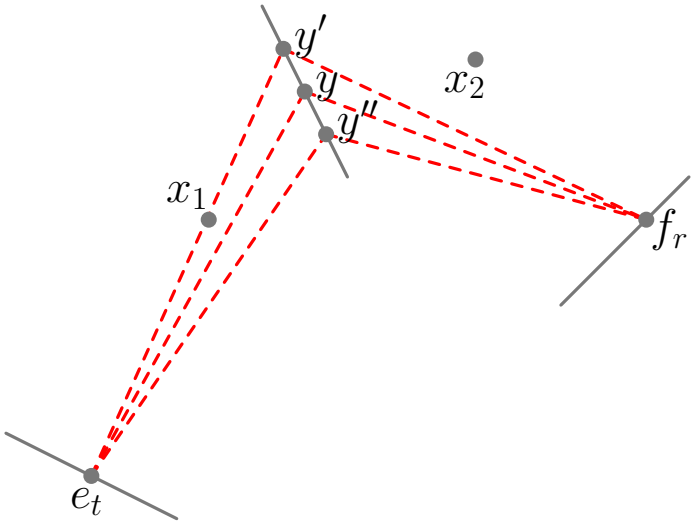
Computing a piecewise linear rational function: Contiguous Art Gallery



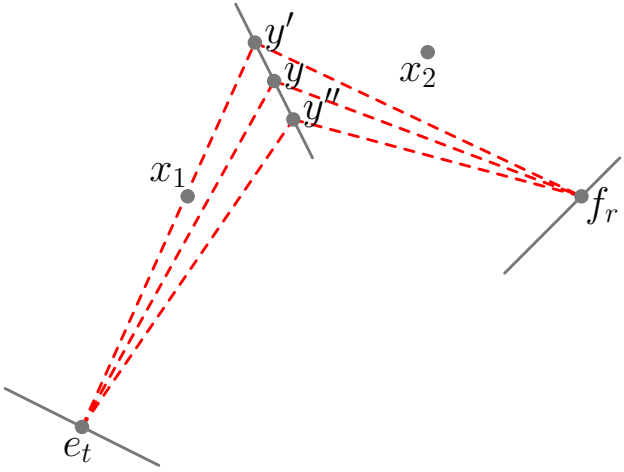
Input: Left
Output: Right

Useful lemma: the min or max of piecewise linear rational functions is piecewise linear rational (other good properties too)

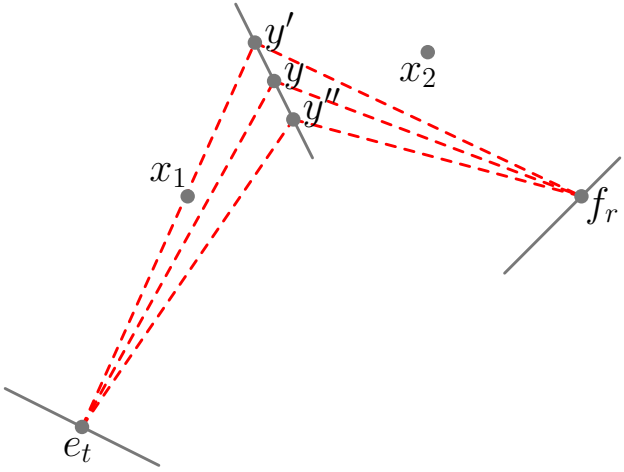
Computing a piecewise linear rational function: Contiguous Art Gallery



Computing a piecewise linear rational function: Contiguous Art Gallery (1a)

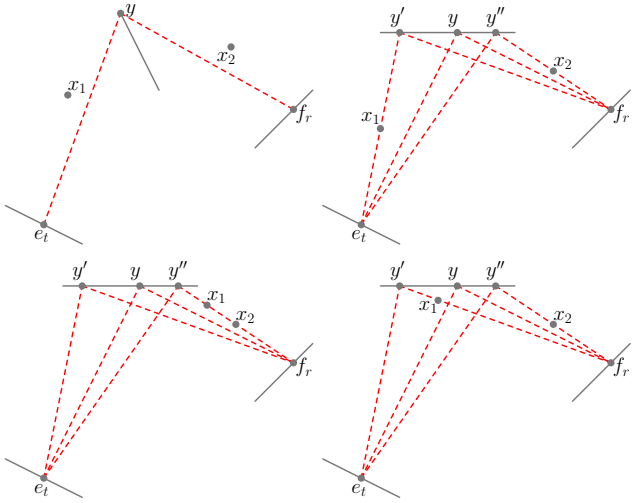


Computing a piecewise linear rational function: Contiguous Art Gallery (1a)



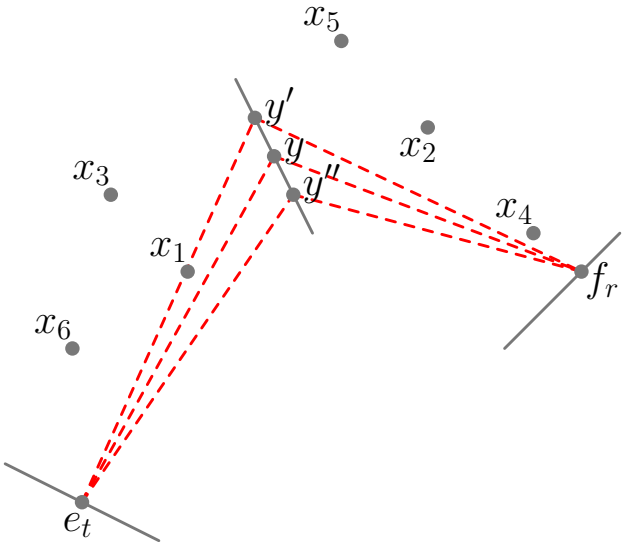
The points x_1 and x_2 restrict.

Computing a piecewise linear rational function: Contiguous Art Gallery (1b)



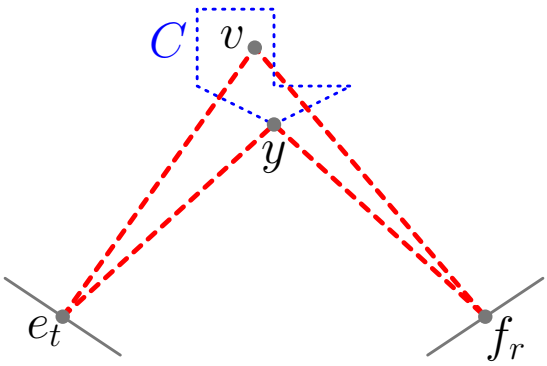
Other cases

Computing a piecewise linear rational function: Contiguous Art Gallery (2)



Take min of all pairs

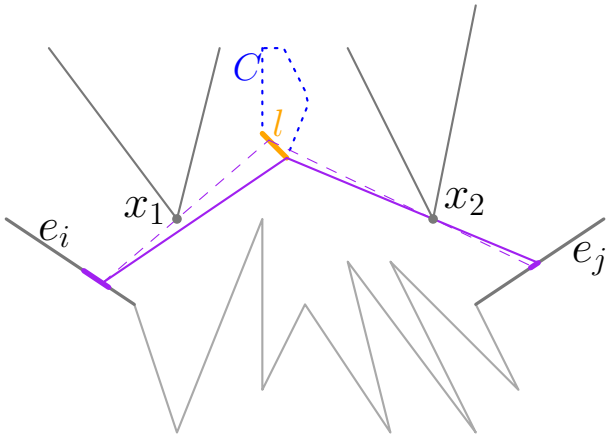
Computing a piecewise linear rational function: Contiguous Art Gallery (3)



Boundary guards suffice! Even with points $\{x\}$

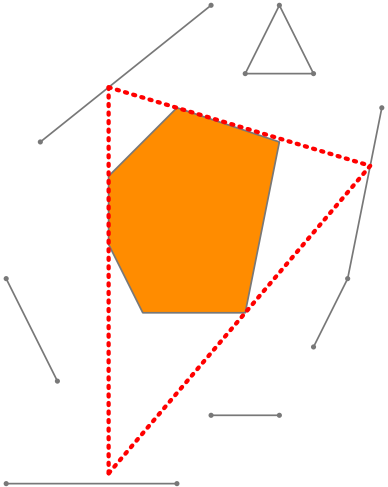
Computing a piecewise linear rational function: Contiguous Art Gallery (4)

Combine across all edge pairs:



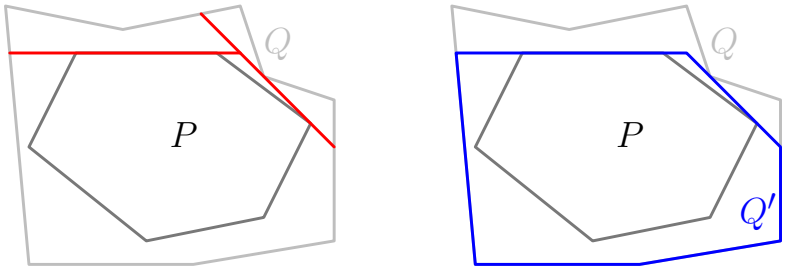
Done all construction steps (not all proof steps)!

Line Segment Separation via Polygon

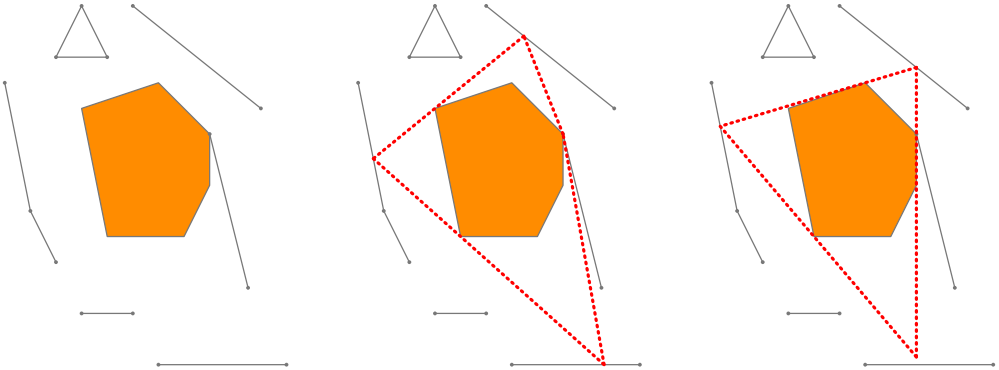


Line Segment Separation via Convex Polygon

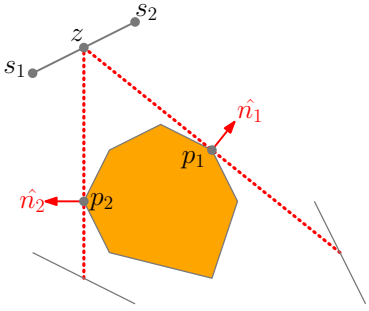
Separating polygon $Q \rightarrow$ convex separating polygon Q'



Line Segment Separation: Next-generator



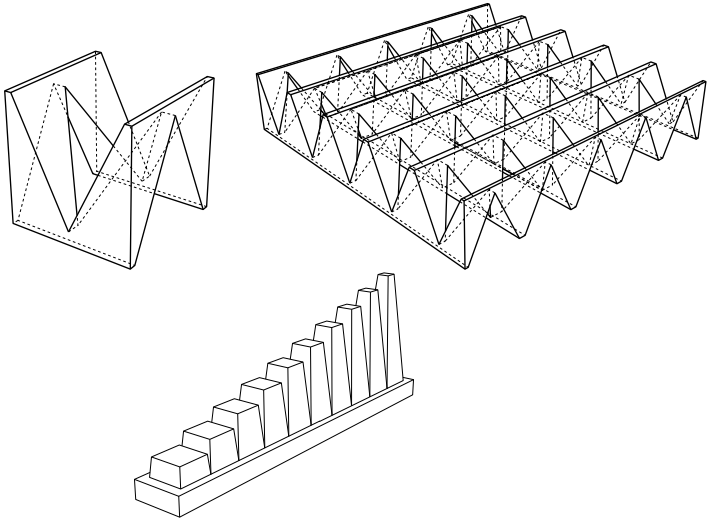
Line Segment Separation: Next-generator Piecewise Linear Rational



$$z = s_1 + \frac{\hat{n_1} \cdot (p_1 - s_1)}{\hat{n_1} \cdot (s_2 - s_1)}(s_2 - s_1)$$

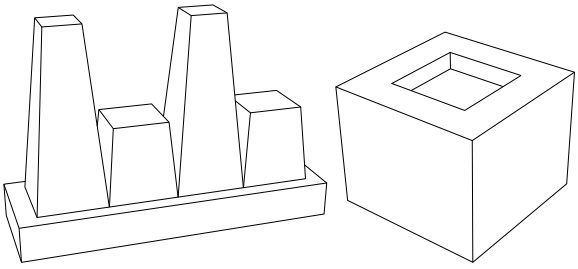
$$\hat{n_2} = ((p_2 - z)_y, -(p_2 - z)_x)$$

YES:

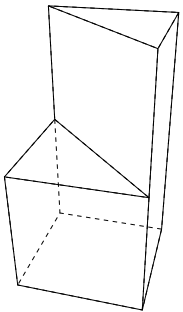


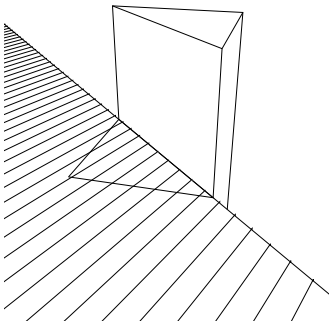
Min Half-Plane Cutting

NO:



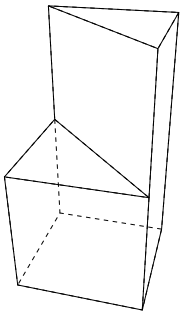
Half-Plane Cutting: Reducing to 2D

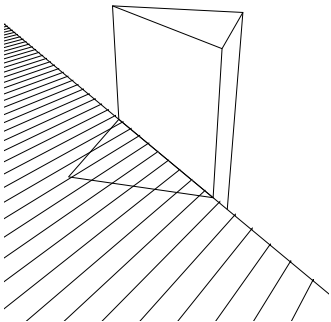




Minimizing in 3D $\rightarrow$ Minimizing (many) in 2D

Half-Plane Cutting: Reducing to 2D





Minimizing in 3D $\rightarrow$ Minimizing (many) in 2D

Becomes line segment separation