Largest Subsets of Triangles in Triangulations

Boris Aronov Marc van Kreveld
Maarten Löffler Rodrigo Silveira

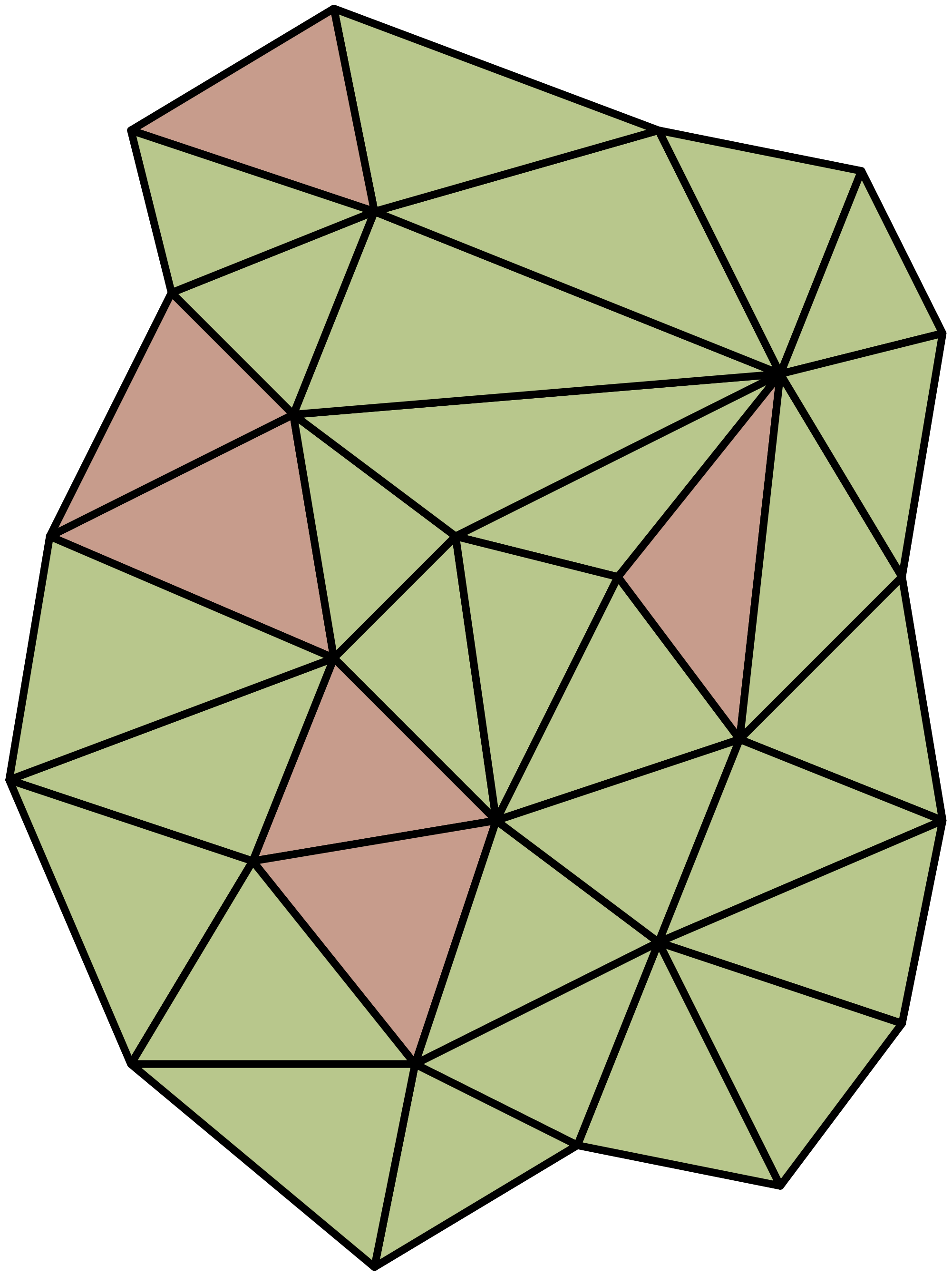
Polytechnic University, New York, USA
Utrecht University, the Netherlands

Overview

- Introduction
  - Triangulations
  - Subregions of *good* triangles
- Overview of our results
- The interesting stuff
  - Convex regions: constructive result
  - Bounded angular change: NP-hardness result
- Concluding remarks

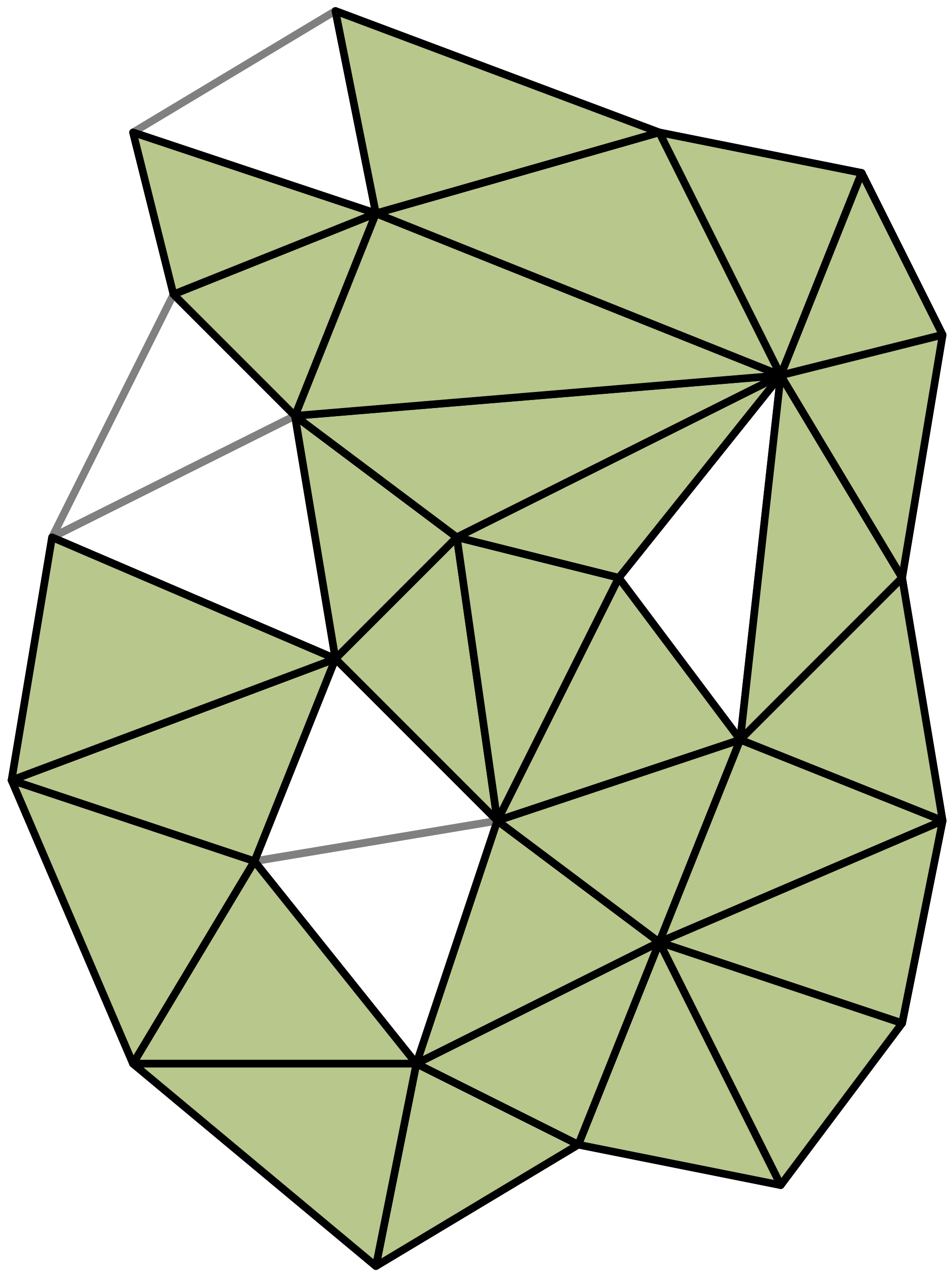
Triangulations

Good Subregions



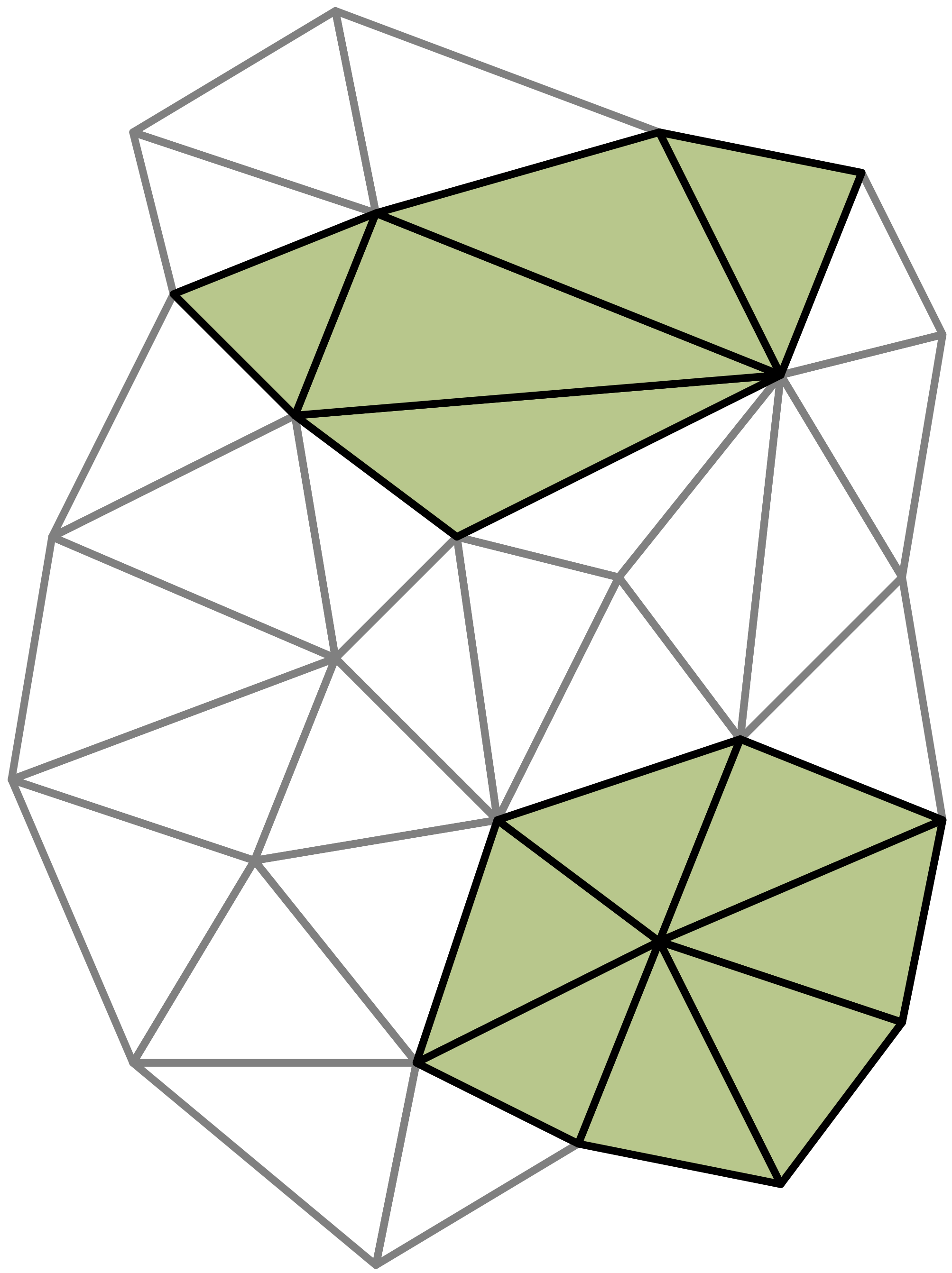
- We want a nicely shaped good region:

Good Subregions



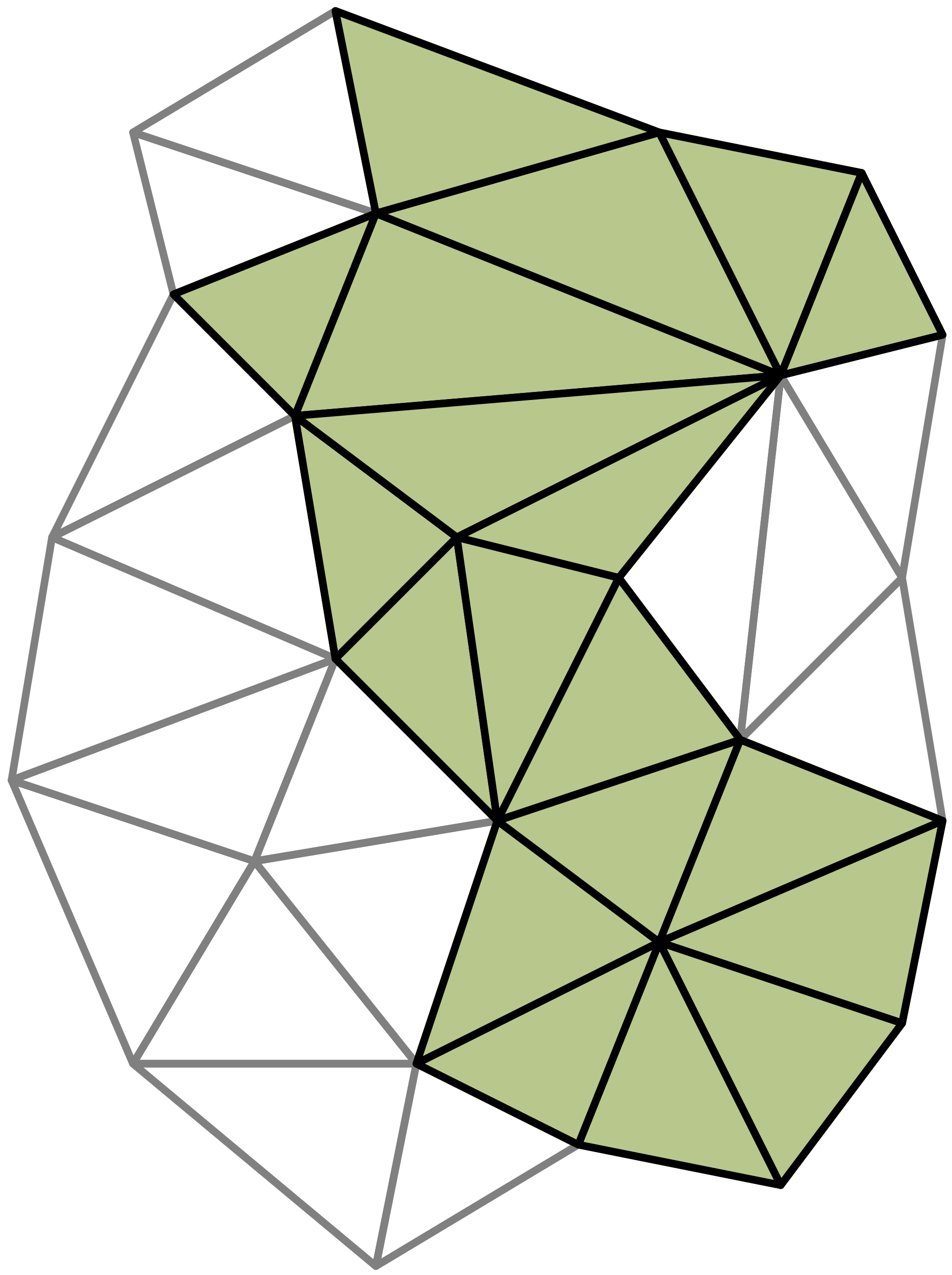
- We want a nicely shaped good region:
  - All good triangles

Good Subregions



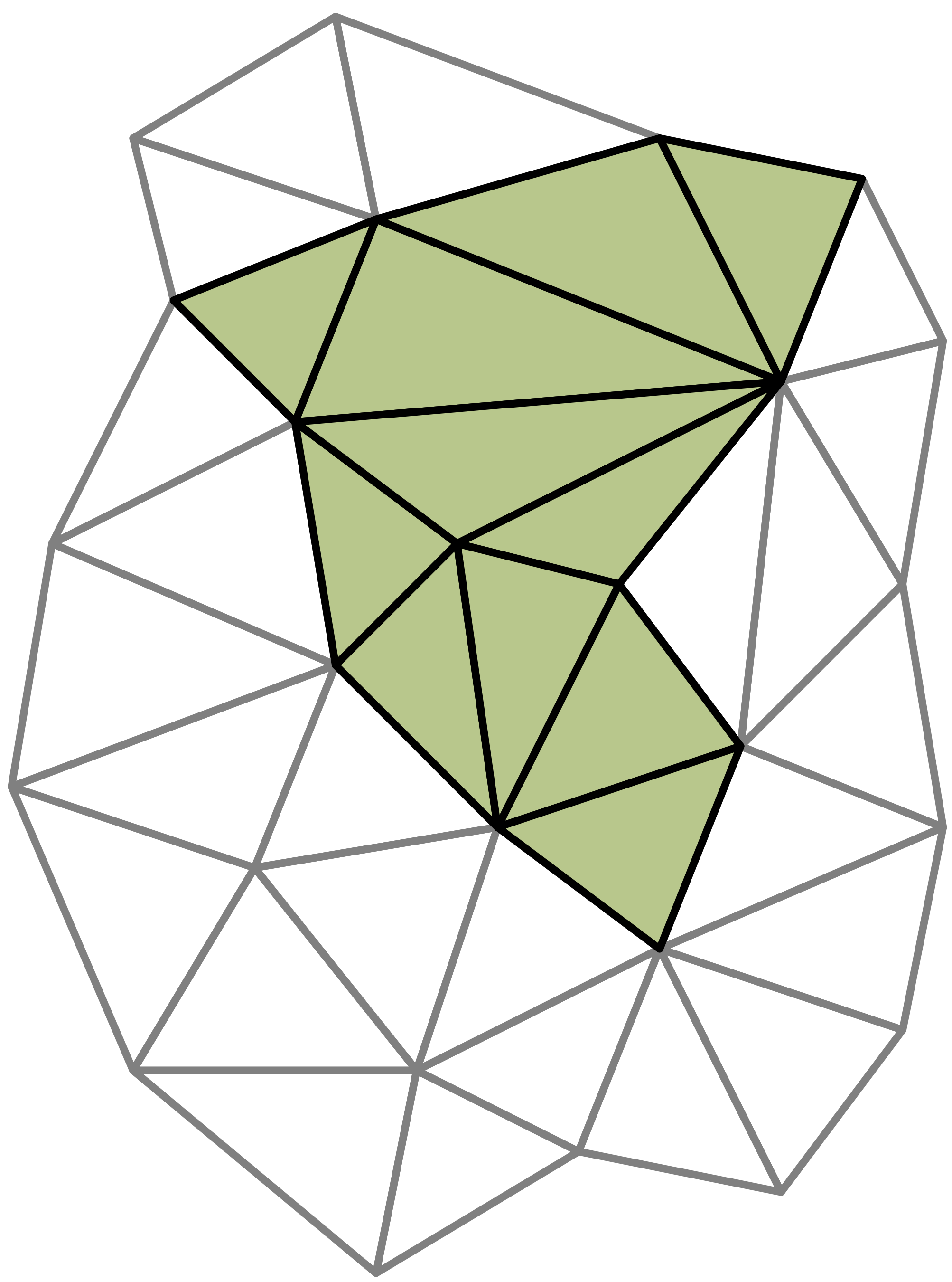
- We want a nicely shaped good region:
  - All good triangles
  - Convex polygon

Good Subregions



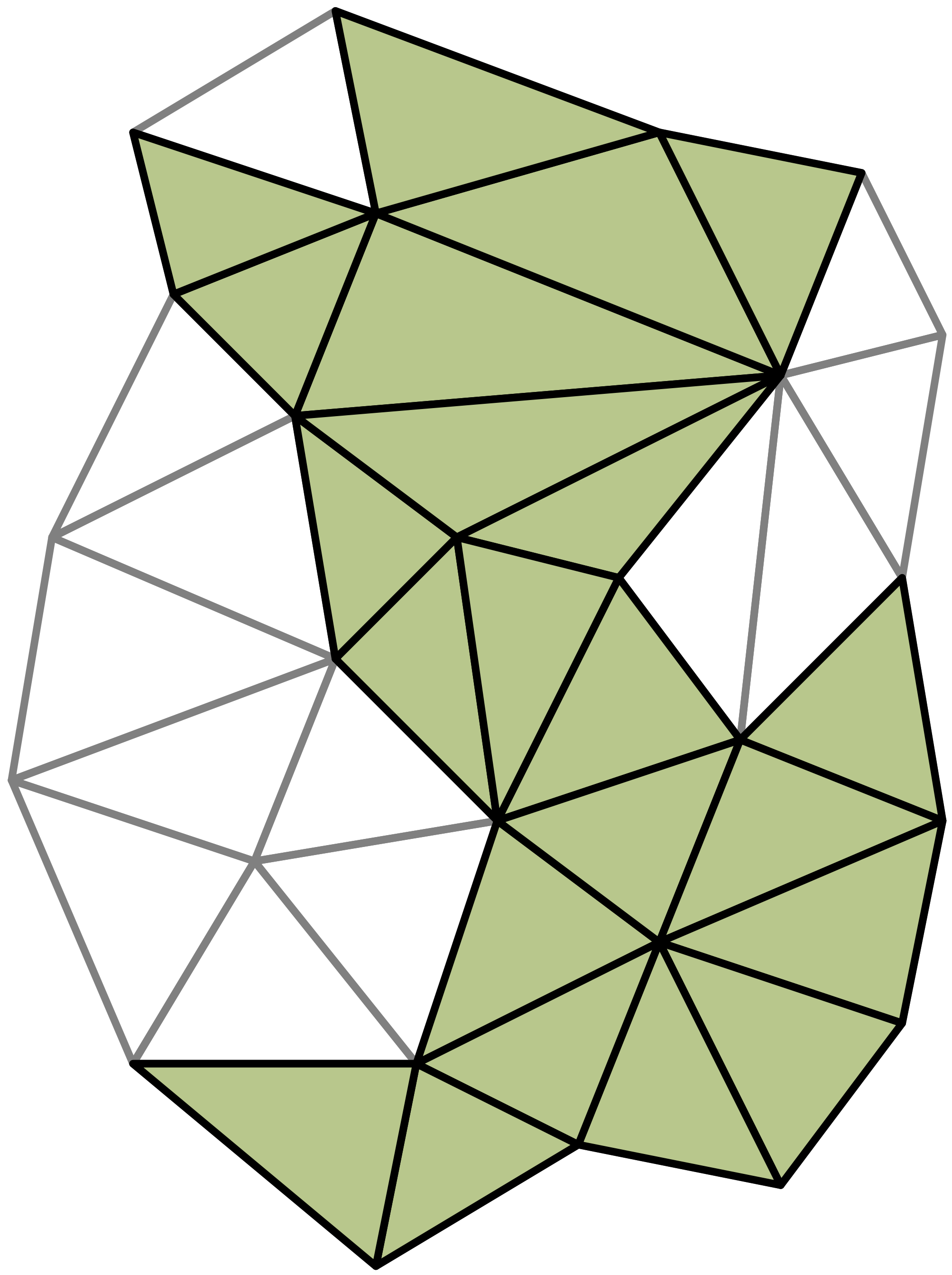
- We want a nicely shaped good region:
  - All good triangles
  - Convex polygon
  - Monotone polygon

Good Subregions



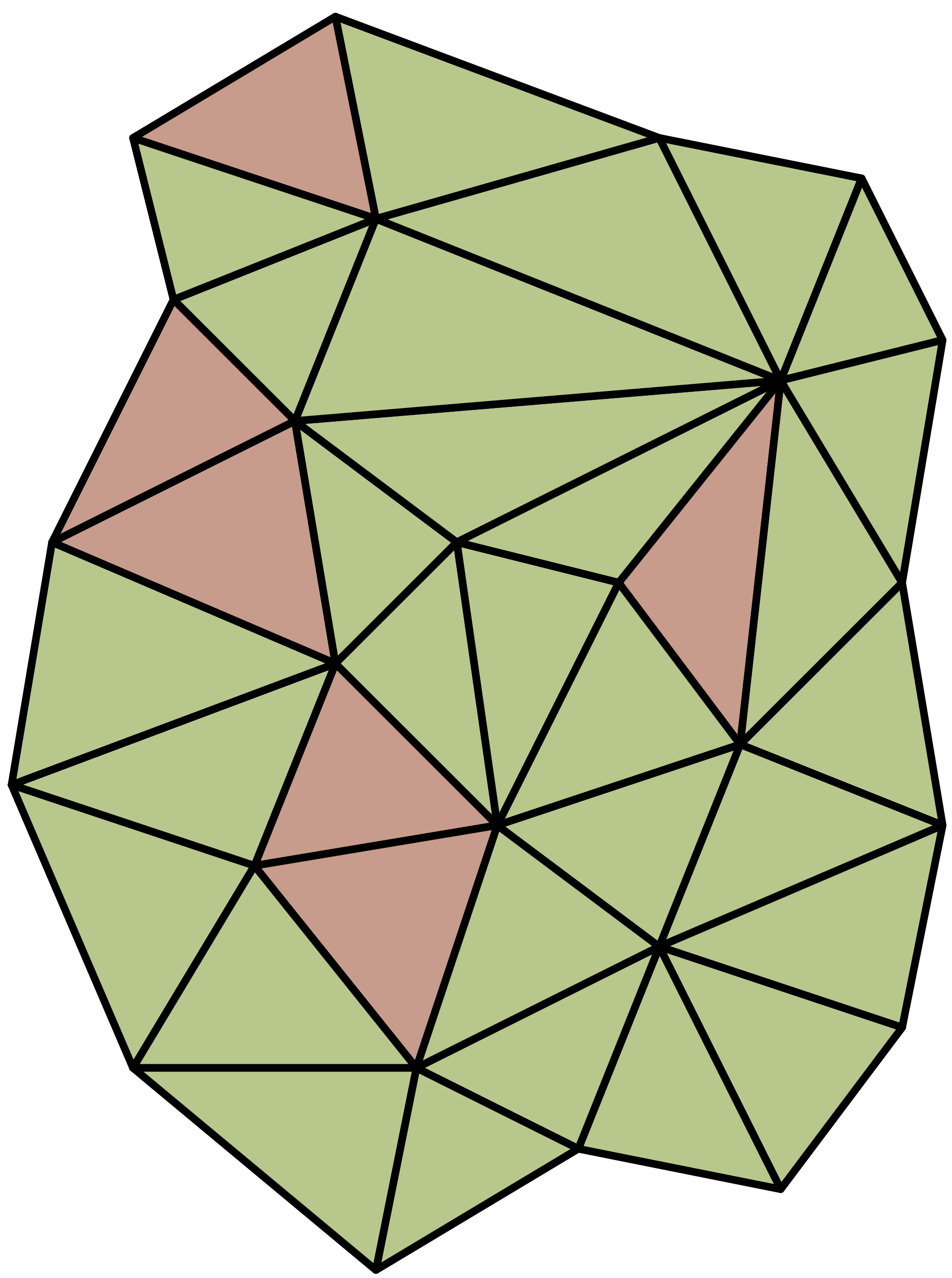
- We want a nicely shaped good region:
  - All good triangles
  - Convex polygon
  - Monotone polygon
  - Bounded angular change (e.g. $< 3\pi$)

Good Subregions



- We want a nicely shaped good region:
  - All good triangles
  - Convex polygon
  - Monotone polygon
  - Bounded angular change (e.g. $< 3\pi$)
  - Bounded negative angle (e.g. $> -\pi$)

Good Subregions



- We want a nicely shaped good region:
  - All good triangles
  - Convex polygon
  - Monotone polygon
  - Bounded angular change (e.g. $< 3\pi$)
  - Bounded negative angle (e.g. $> -\pi$)
- Best measure depends on application

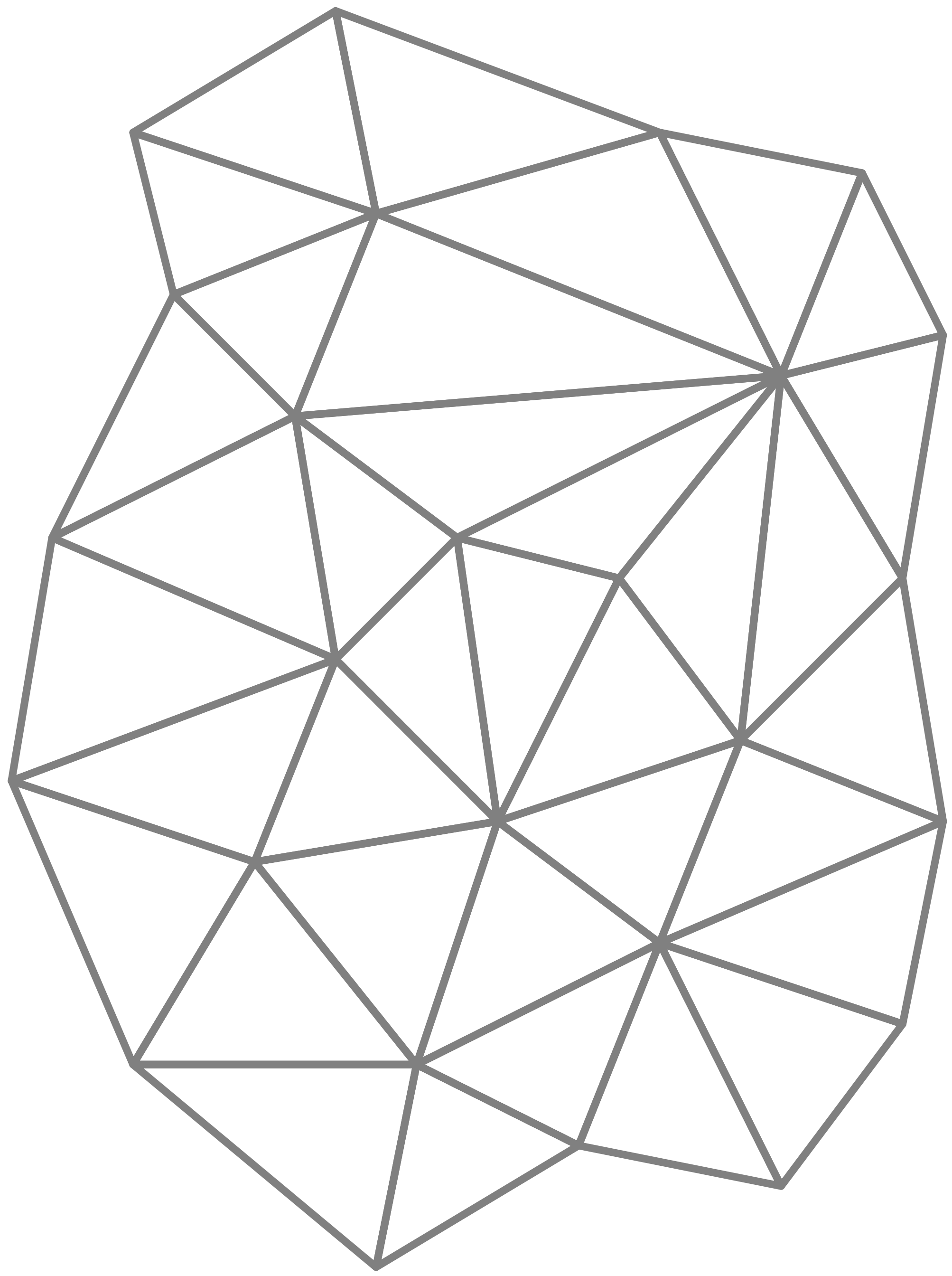
Results

- Largest convex subregion
  - Plane sweep and dynamic programming $O(n^2)$
- Largest monotone subregion
  - Direction given: similar to convex $O(n^2)$
  - Direction not given: try all directions $O(n^3)$
- Bounded angular change
  - Reduction from KNAPSACK NP-hard
- Bounded negative angle
  - Dynamic programming $O(n^6)$

Largest Convex Good Region

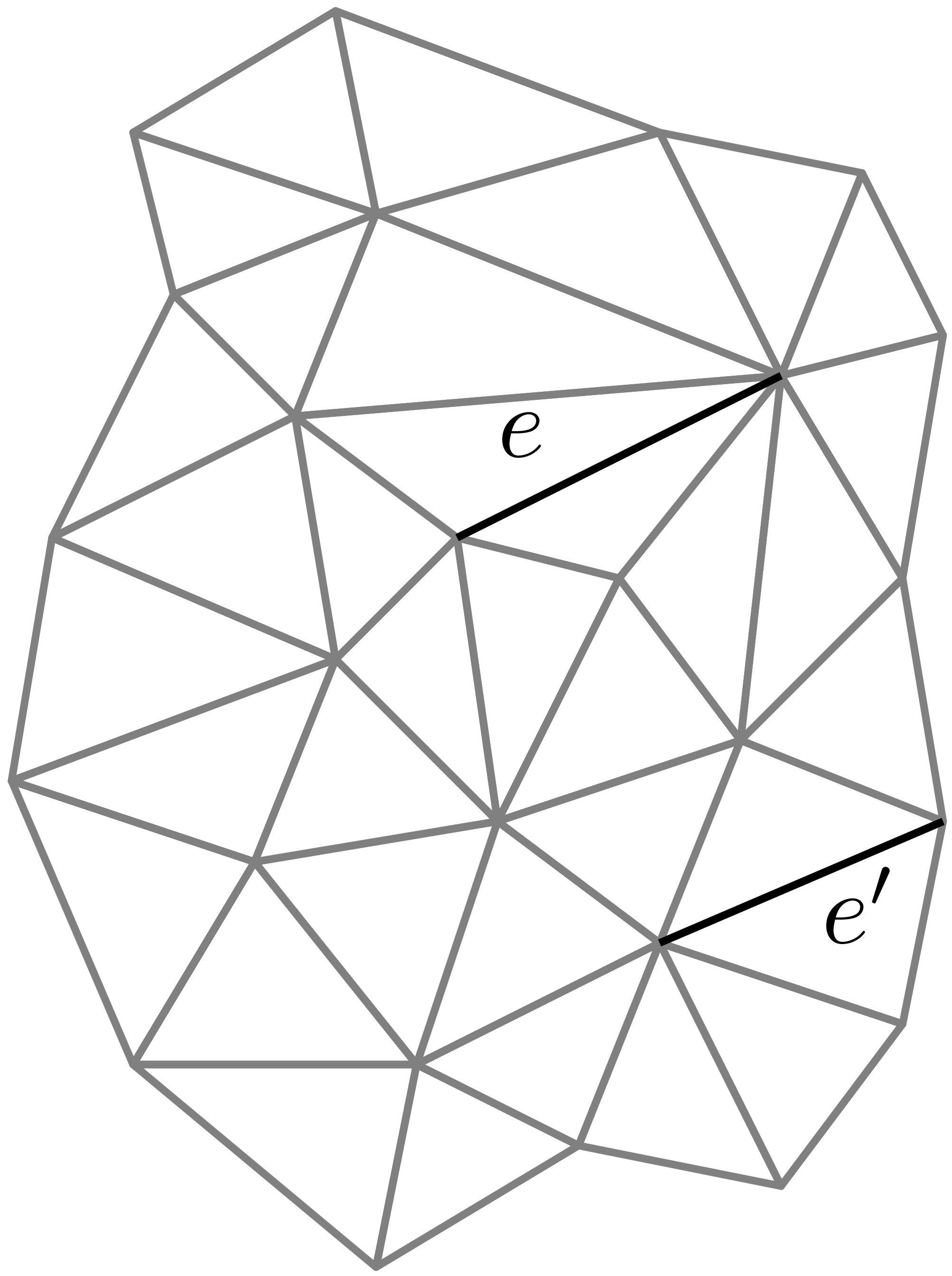
- Peeling potatoes
  - Given a simple polygon
  - What is the largest convex subpolygon?
  - Can be solved in $O(n^7)$ time [Chang & Yap, 1986]
- Peeling *meshed* potatoes
  - Given a *triangulated* polygon (with Steiner points)
  - What is the largest *triangle-respecting* convex subpolygon?
  - Can be solved in $O(n^2)$ time

Promising Pairs of Edges



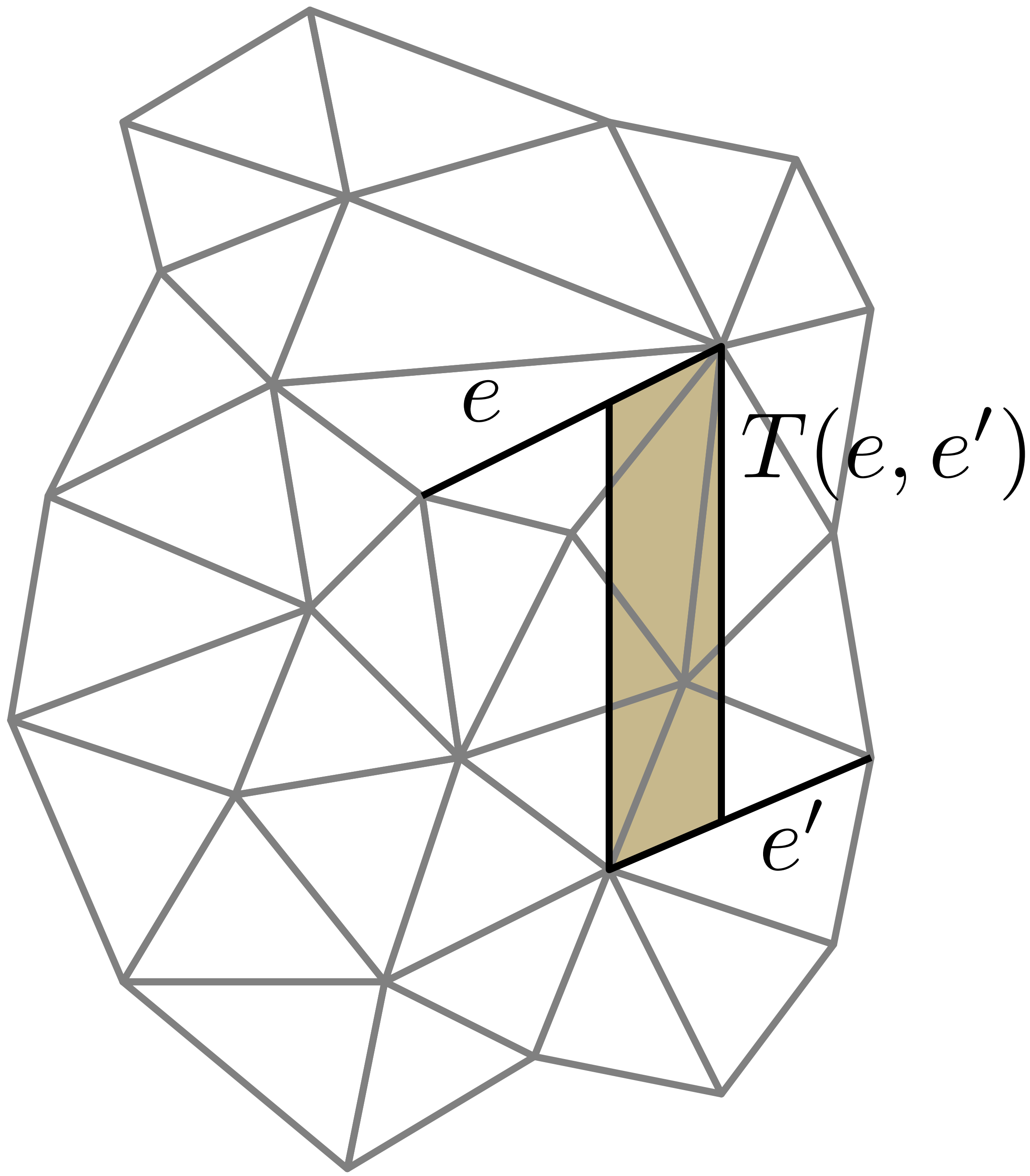
- Given edges e and e'
- Let x_l be rightmost left x-coordinate
- Let x_r be leftmost right x-coordinate
- $T(e, e')$ is bounded by e, e', x_l and x_r
- If $T(e, e')$ intersects no bad triangles, (e, e') is *promising*

Promising Pairs of Edges



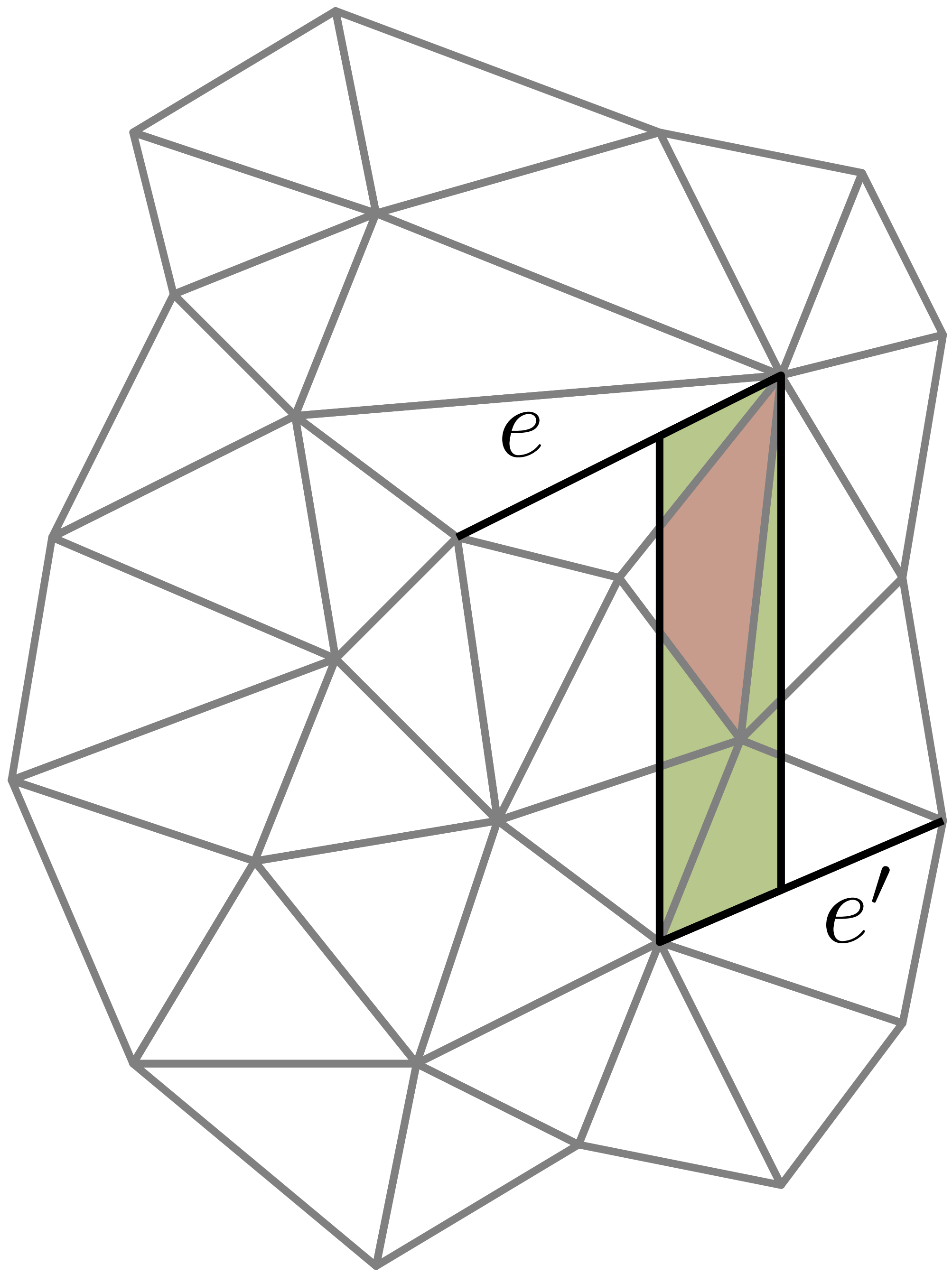
- Given edges e and e'
- Let x_l be rightmost left x-coordinate
- Let x_r be leftmost right x-coordinate
- $T(e, e')$ is bounded by e, e', x_l and x_r
- If $T(e, e')$ intersects no bad triangles, (e, e') is *promising*

Promising Pairs of Edges



- Given edges e and e'
- Let x_l be rightmost left x-coordinate
- Let x_r be leftmost right x-coordinate
- $T(e, e')$ is bounded by e, e', x_l and x_r
- If $T(e, e')$ intersects no bad triangles, (e, e') is *promising*

Promising Pairs of Edges

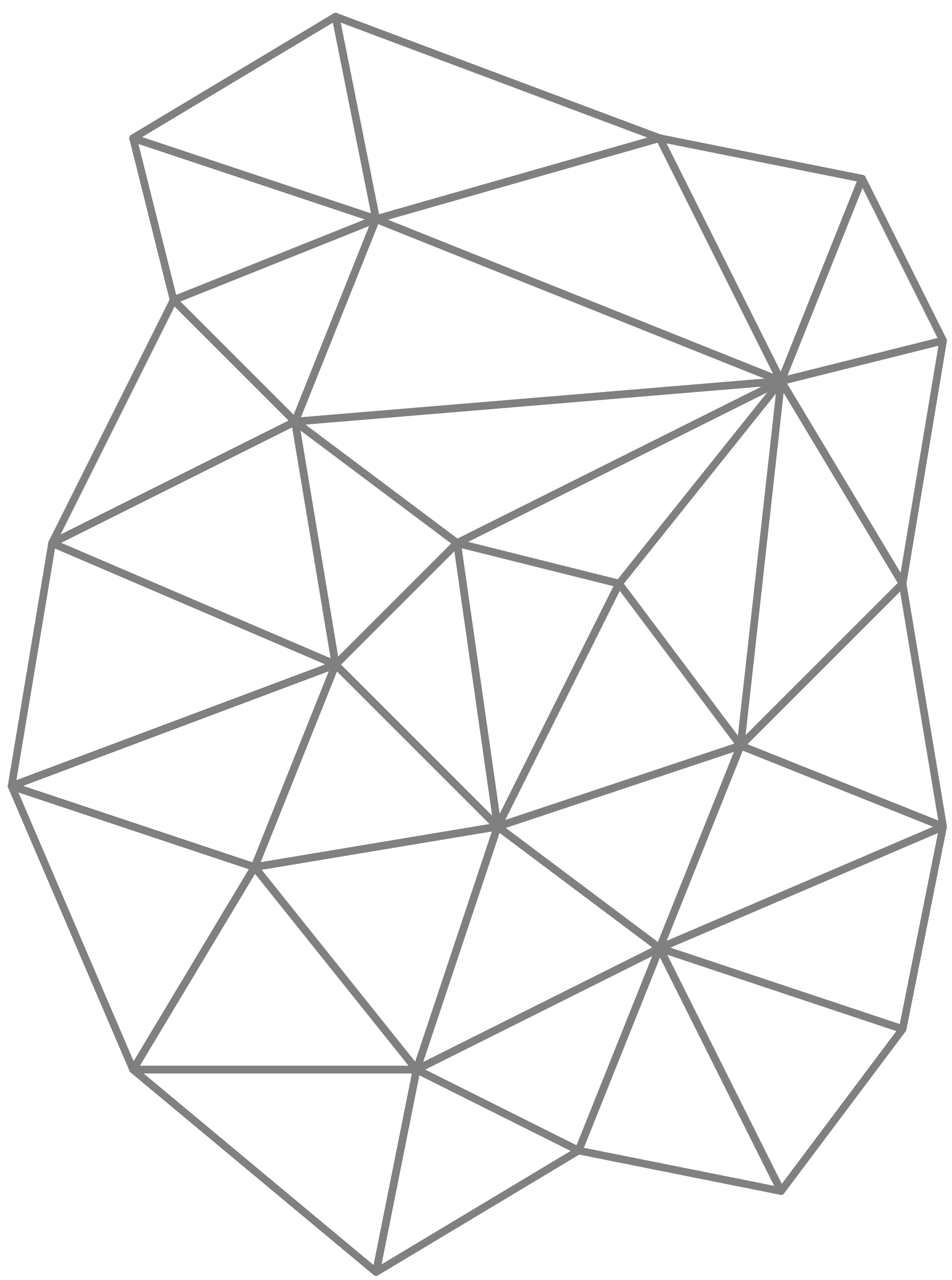


- Given edges e and e'
- Let x_l be rightmost left x-coordinate
- Let x_r be leftmost right x-coordinate
- $T(e, e')$ is bounded by e, e', x_l and x_r
- If $T(e, e')$ intersects no bad triangles, (e, e') is *promising*

Largest Convex Good Region

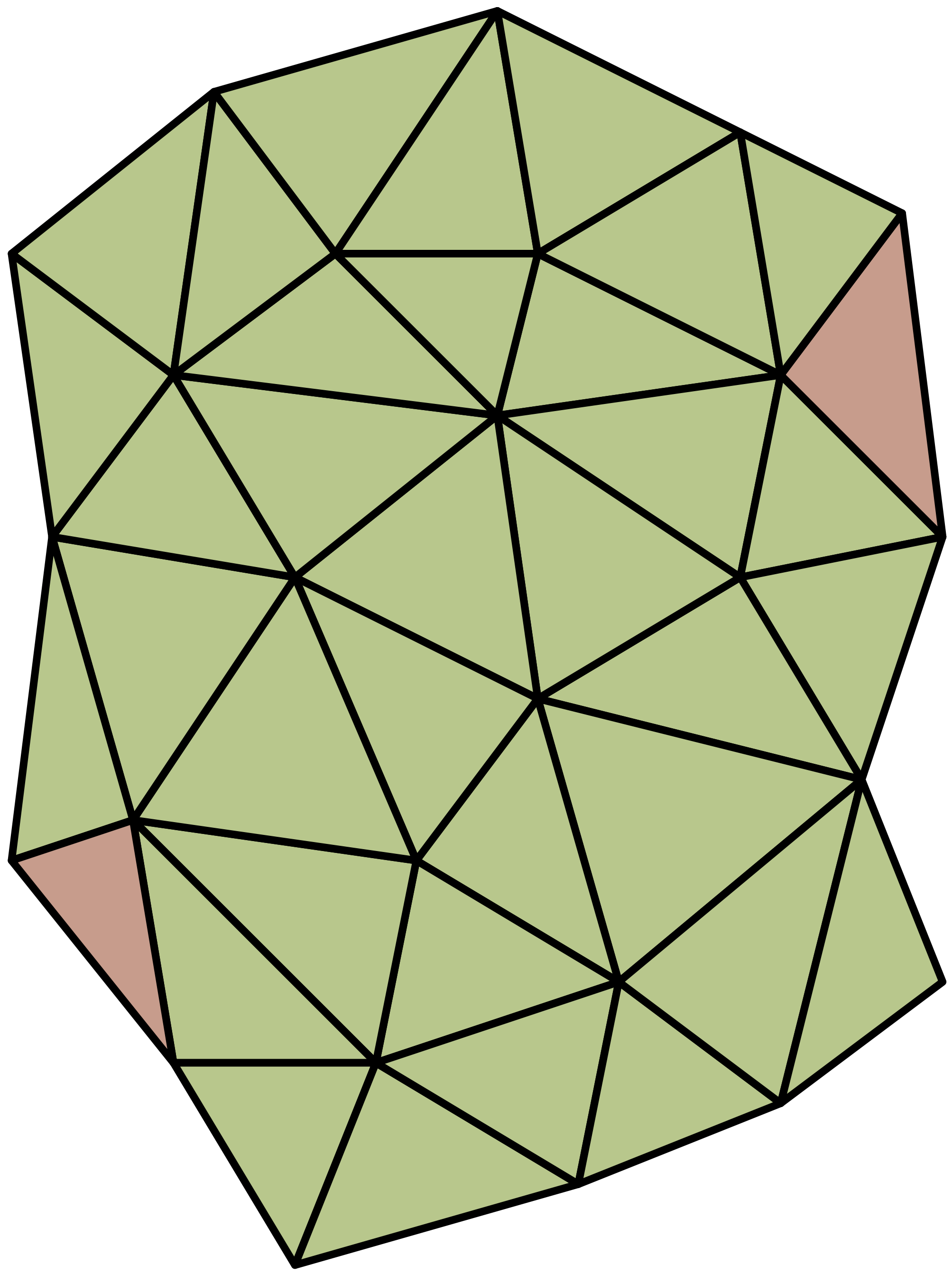
- Two step algorithm
- Step 1: Computing a table of promising pairs
  - Plane sweep
- Step 2: Computing the largest convex region
  - Dynamic programming

Computing All Promising Pairs



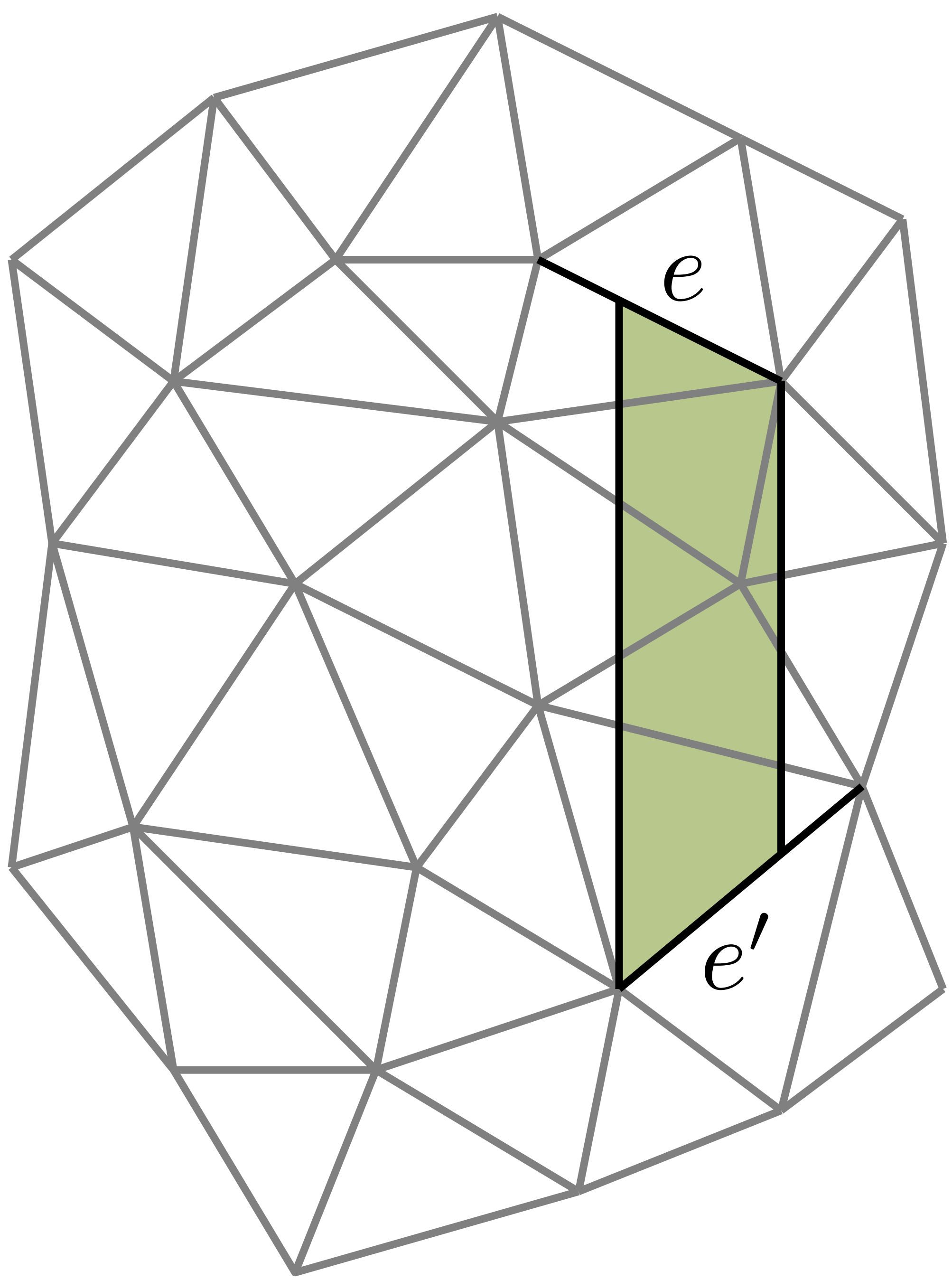
- Sweep vertical line l
- Maintain, for every edge e crossing l:
  - Ehm... does this algorithm actually *work*...?

Dynamic Programming



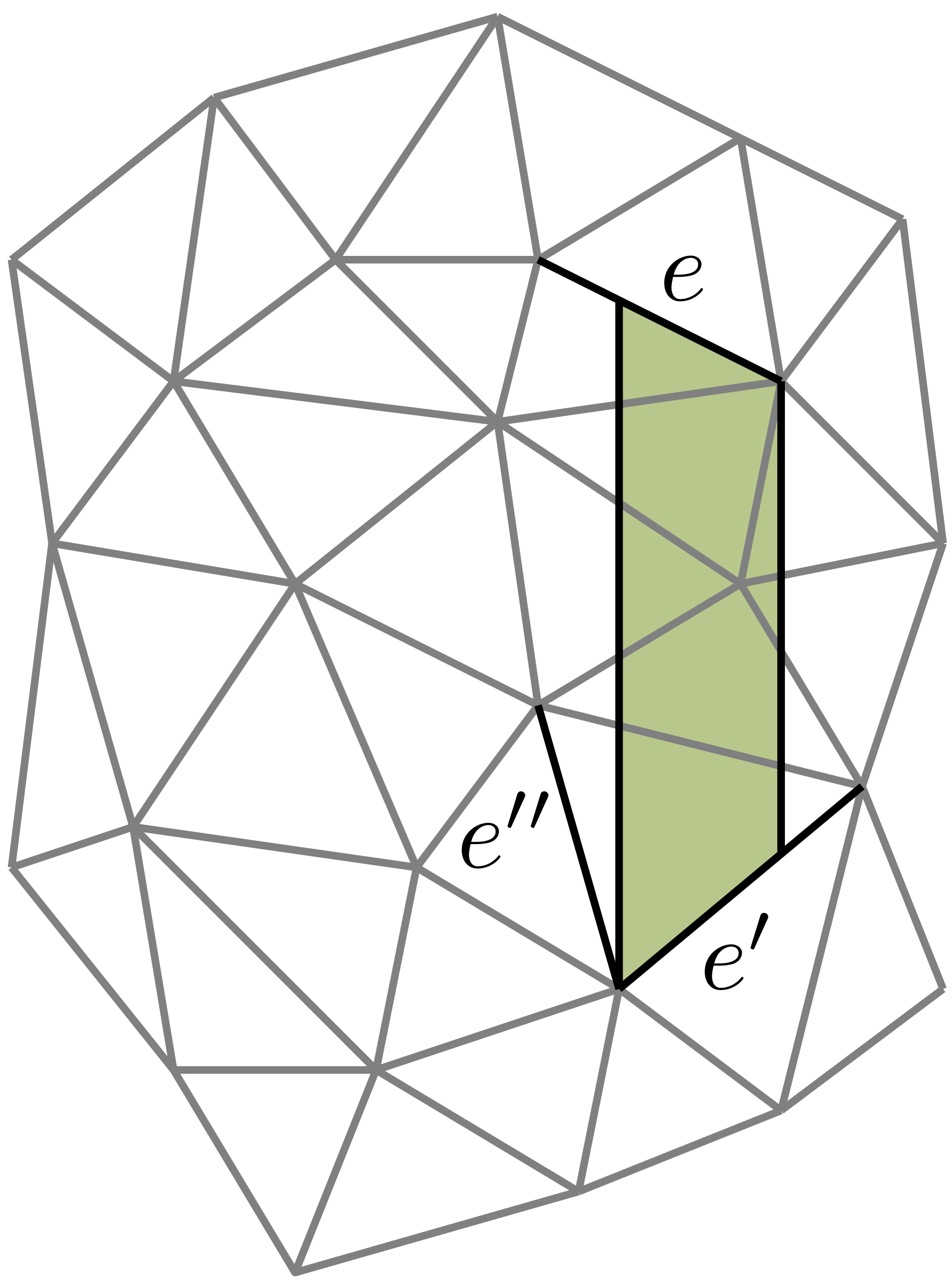
- Given promising pair of edges e and e'
- $P(e, e')$ is largest good convex polygon left and up to $T(e, e')$
- Inspect all edges e'' connected to e or e'
- Take largest P that:
  - Is convex
  - Does not contain bad triangles

Dynamic Programming



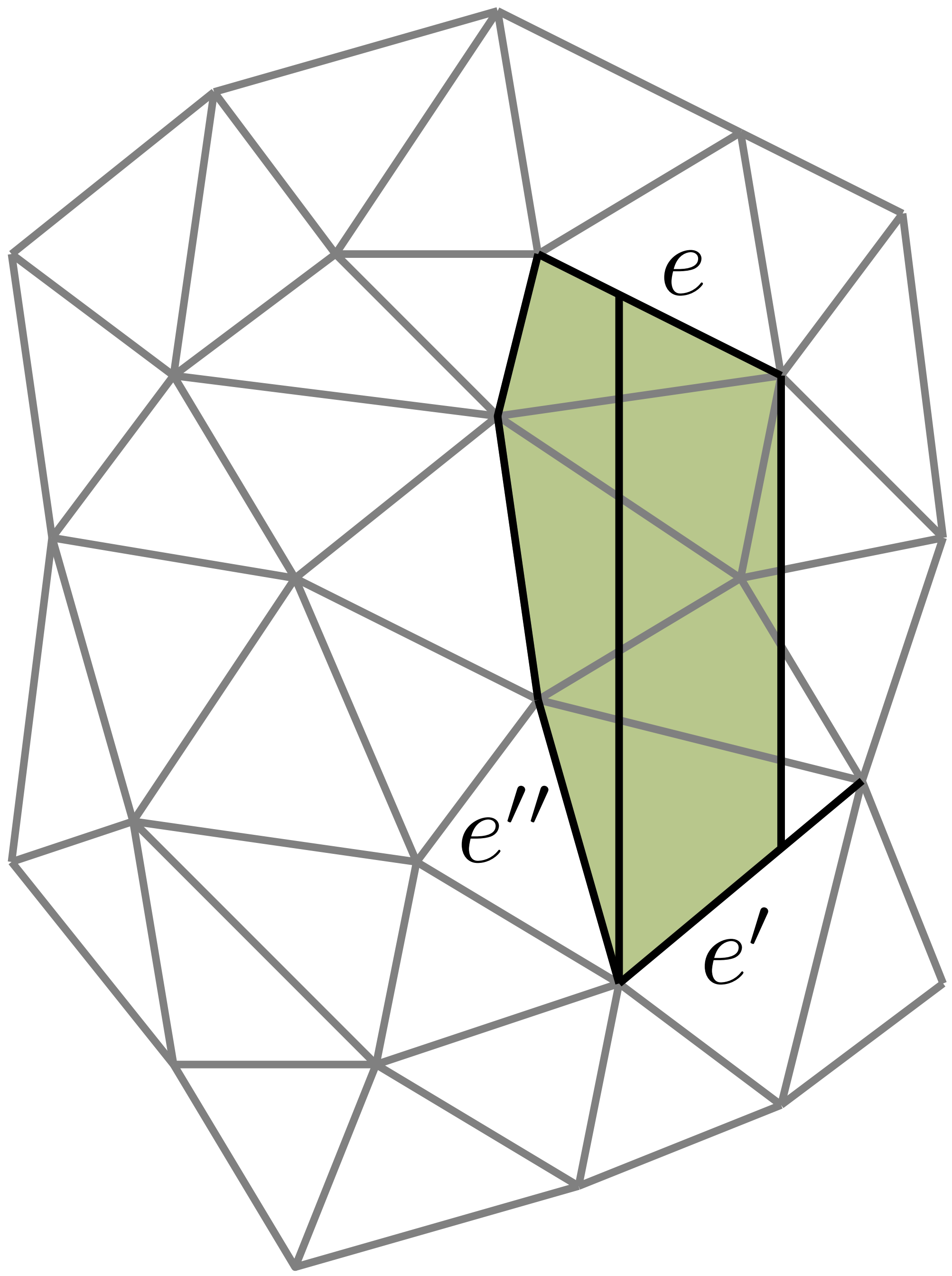
- Given promising pair of edges e and e'
- $P(e, e')$ is largest good convex polygon left and up to $T(e, e')$
- Inspect all edges e'' connected to e or e'
- Take largest P that:
  - Is convex
  - Does not contain bad triangles

Dynamic Programming



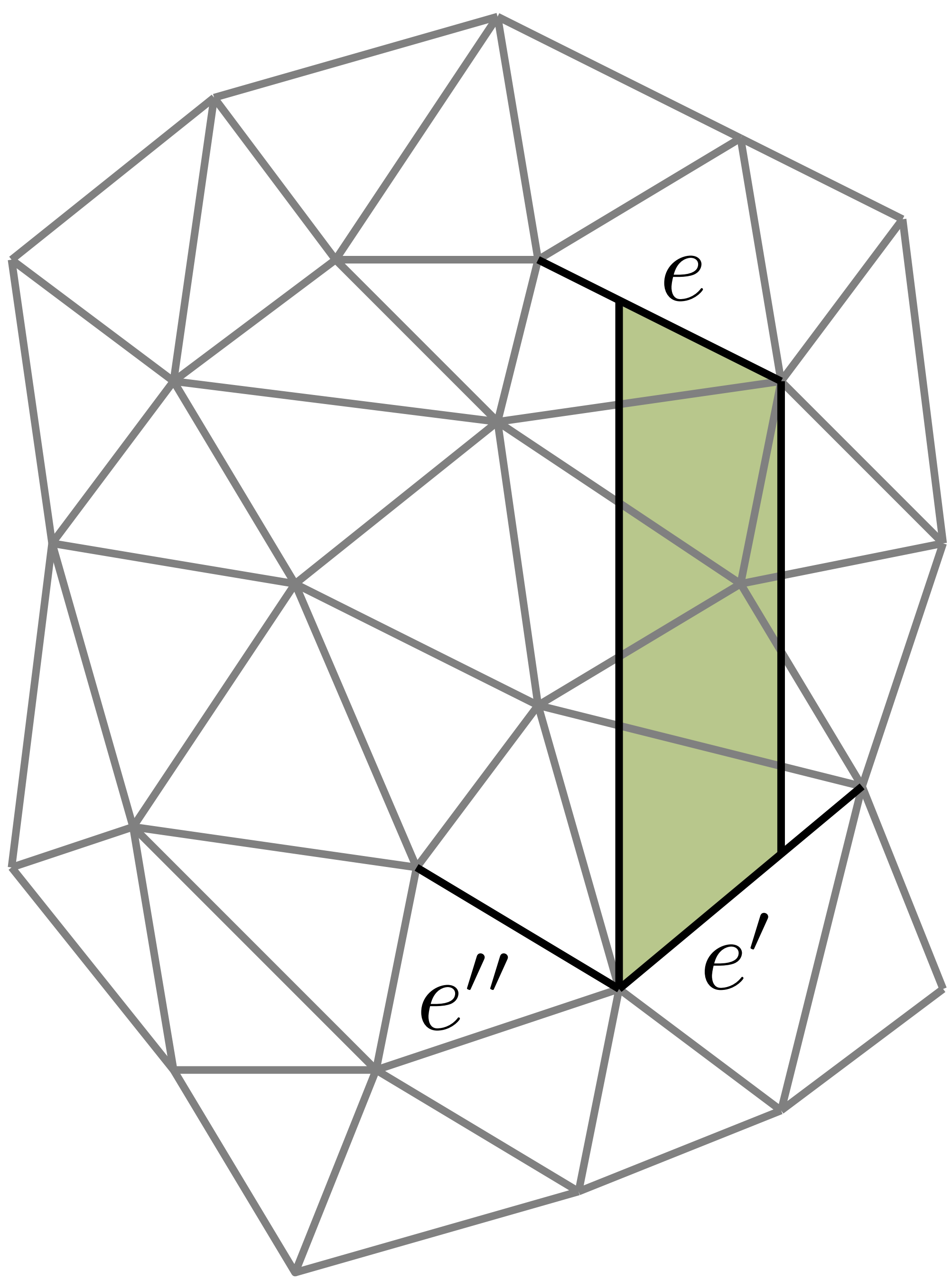
- Given promising pair of edges e and e'
- $P(e, e')$ is largest good convex polygon left and up to $T(e, e')$
- Inspect all edges e'' connected to e or e'
- Take largest P that:
  - Is convex
  - Does not contain bad triangles

Dynamic Programming



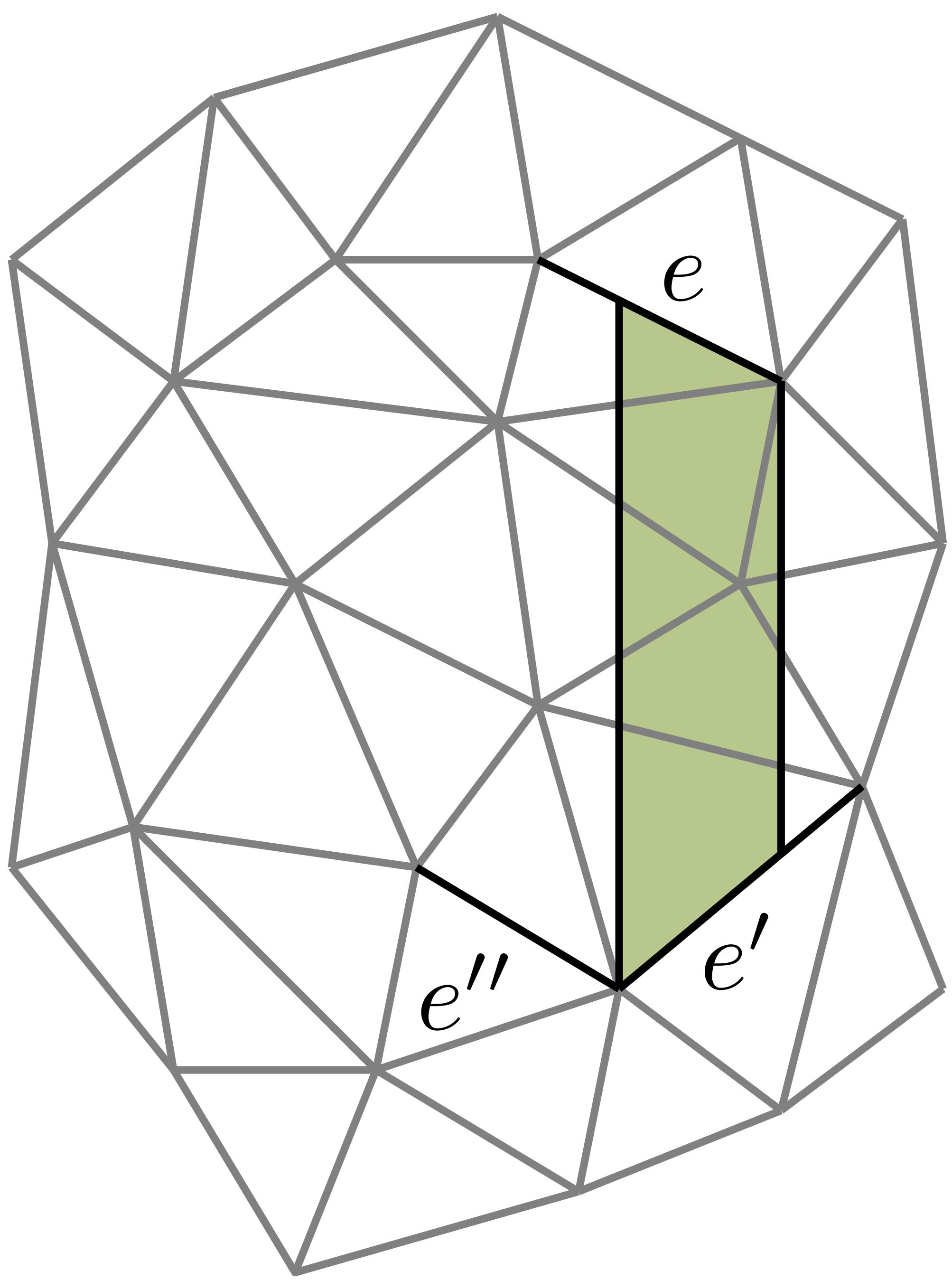
- Given promising pair of edges e and e'
- $P(e, e')$ is largest good convex polygon left and up to $T(e, e')$
- Inspect all edges e'' connected to e or e'
- Take largest P that:
  - Is convex
  - Does not contain bad triangles

Dynamic Programming



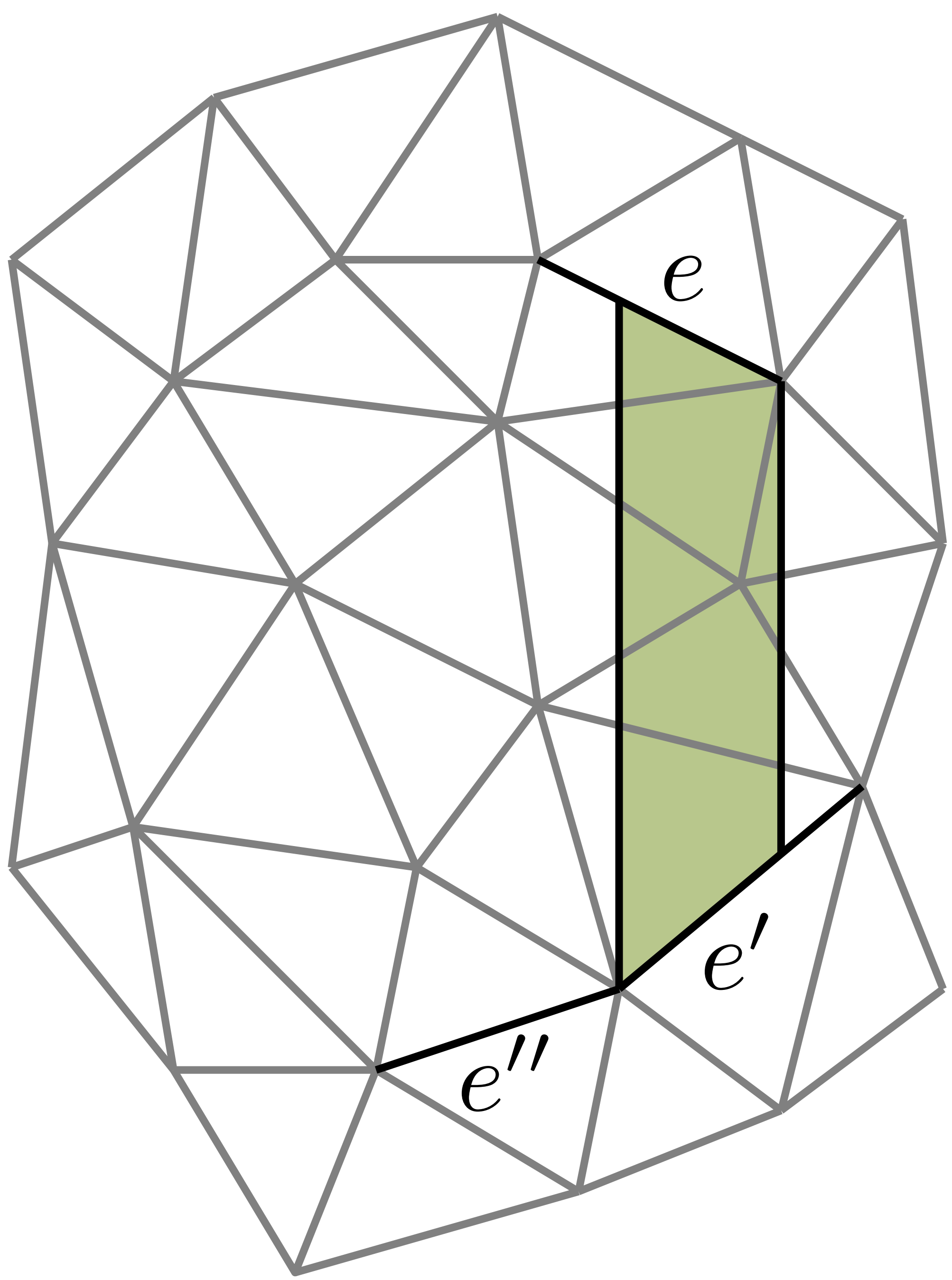
- Given promising pair of edges e and e'
- $P(e, e')$ is largest good convex polygon left and up to $T(e, e')$
- Inspect all edges e'' connected to e or e'
- Take largest P that:
  - Is convex
  - Does not contain bad triangles

Dynamic Programming



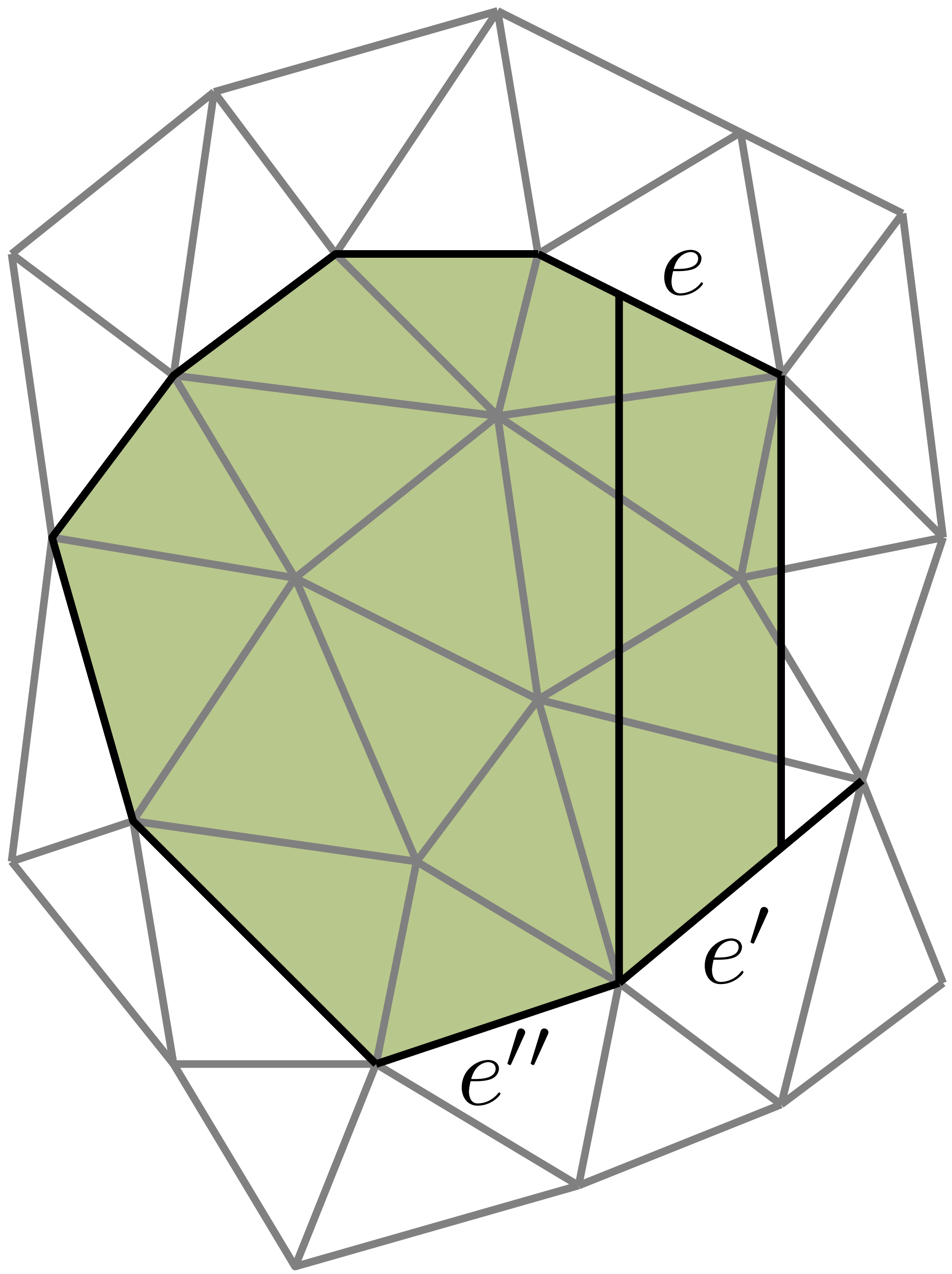
- Given promising pair of edges e and e'
- $P(e, e')$ is largest good convex polygon left and up to $T(e, e')$
- Inspect all edges e'' connected to e or e'
- Take largest P that:
  - Is convex
  - Does not contain bad triangles

Dynamic Programming



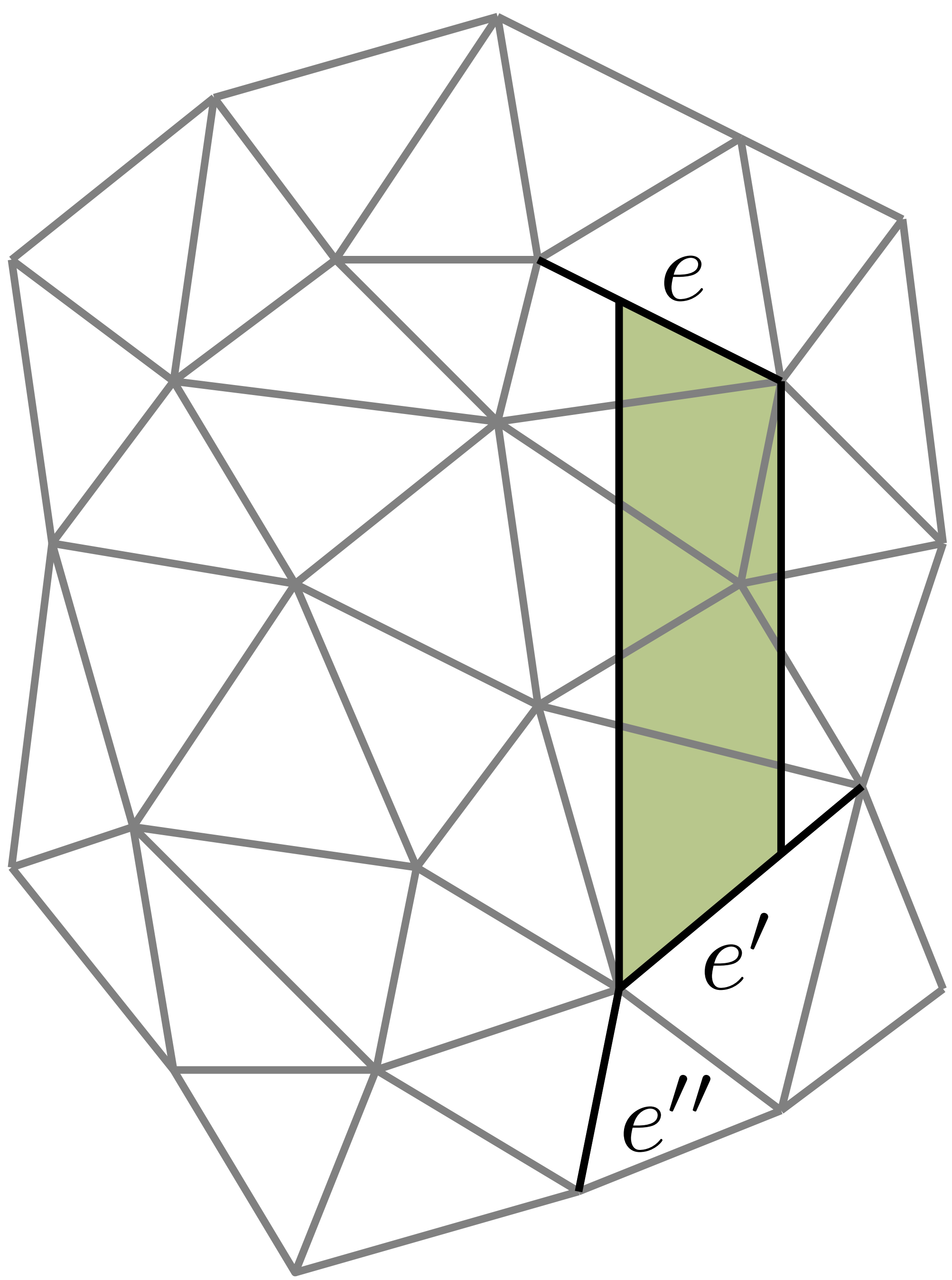
- Given promising pair of edges e and e'
- $P(e, e')$ is largest good convex polygon left and up to $T(e, e')$
- Inspect all edges e'' connected to e or e'
- Take largest P that:
  - Is convex
  - Does not contain bad triangles

Dynamic Programming



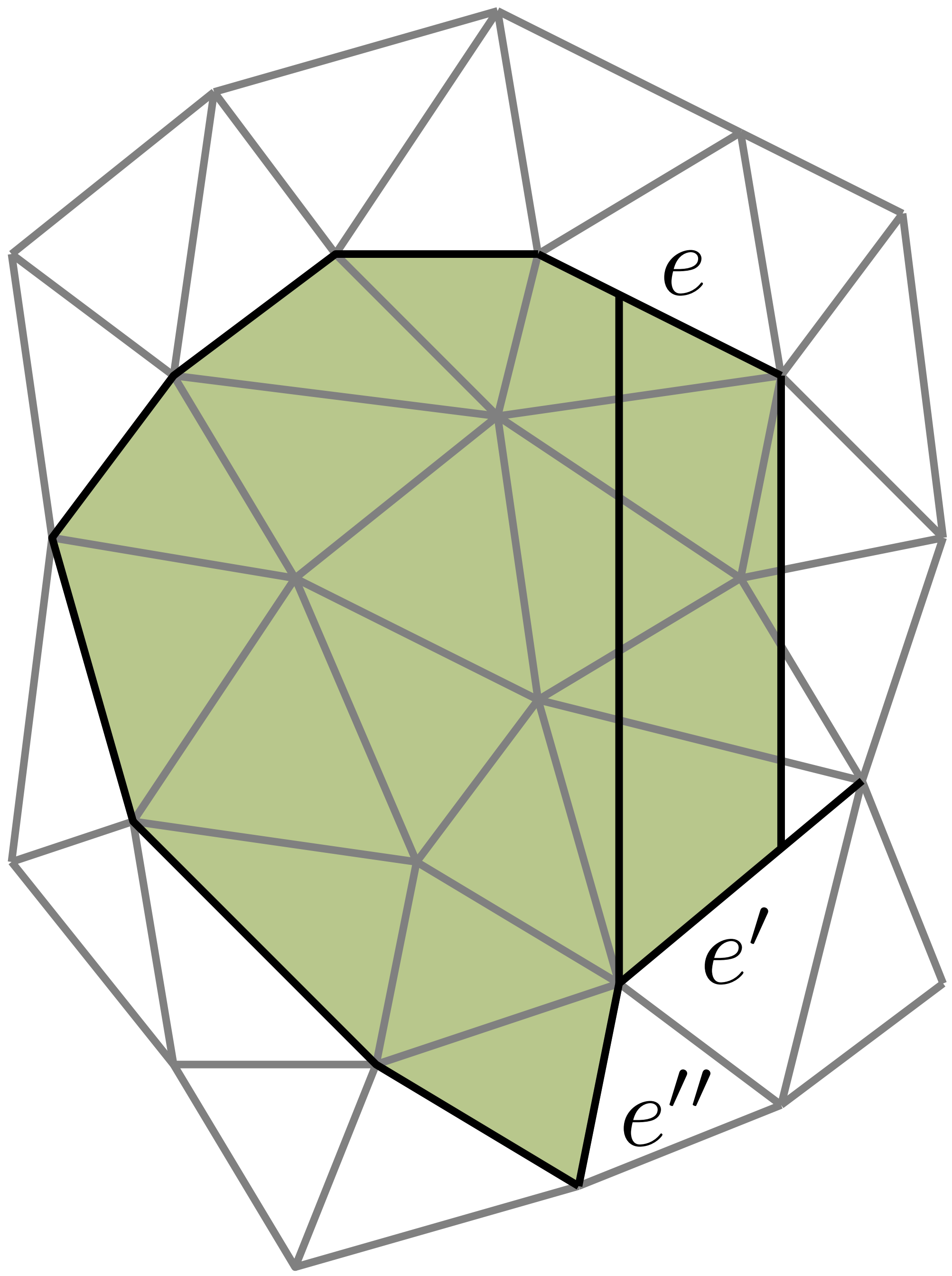
- Given promising pair of edges e and e'
- $P(e, e')$ is largest good convex polygon left and up to $T(e, e')$
- Inspect all edges e'' connected to e or e'
- Take largest P that:
  - Is convex
  - Does not contain bad triangles

Dynamic Programming



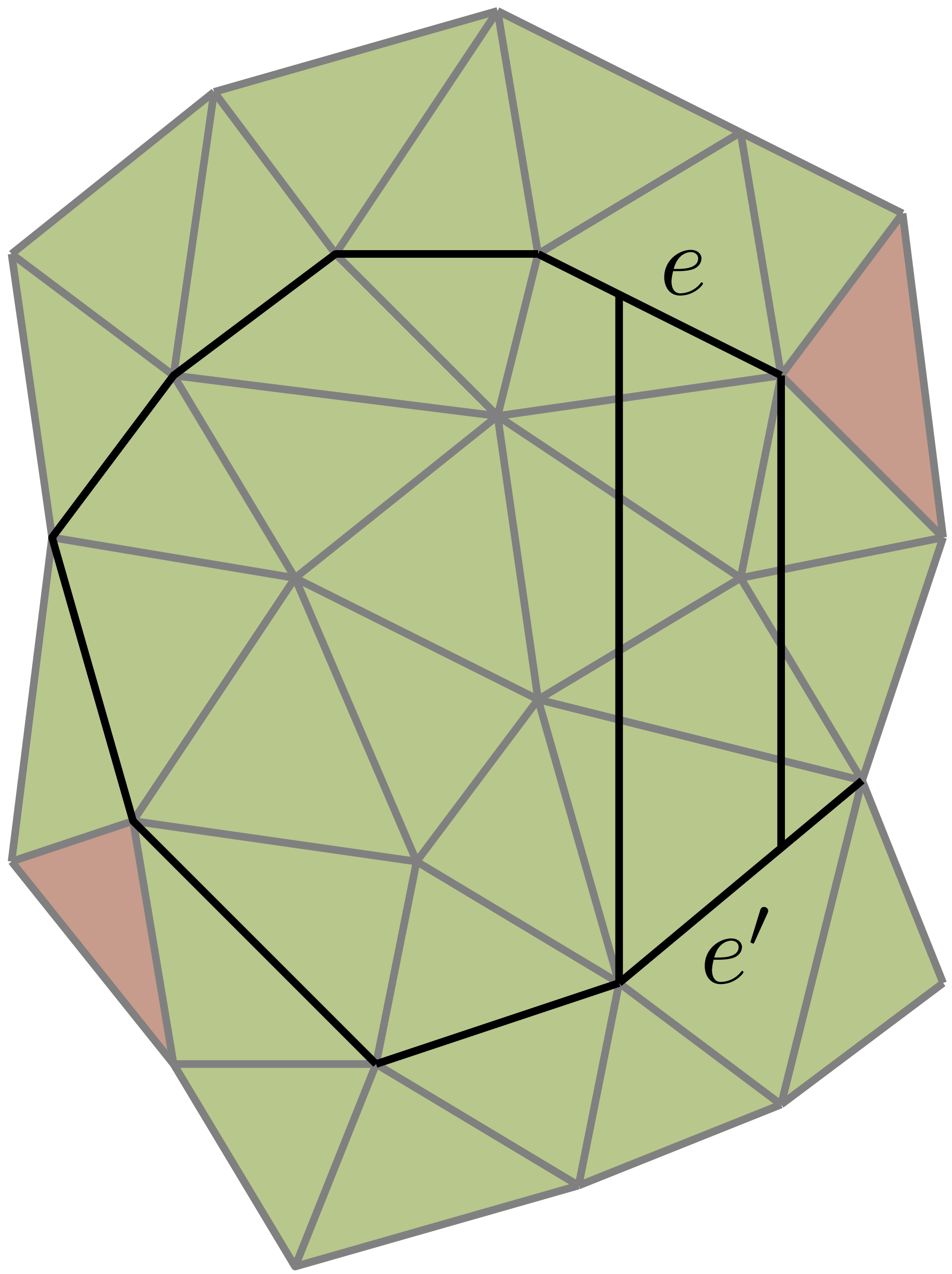
- Given promising pair of edges e and e'
- $P(e, e')$ is largest good convex polygon left and up to $T(e, e')$
- Inspect all edges e'' connected to e or e'
- Take largest P that:
  - Is convex
  - Does not contain bad triangles

Dynamic Programming



- Given promising pair of edges e and e'
- $P(e, e')$ is largest good convex polygon left and up to $T(e, e')$
- Inspect all edges e'' connected to e or e'
- Take largest P that:
  - Is convex
  - Does not contain bad triangles

Dynamic Programming



- Given promising pair of edges e and e'
- $P(e, e')$ is largest good convex polygon left and up to $T(e, e')$
- Inspect all edges e'' connected to e or e'
- Take largest P that:
  - Is convex
  - Does not contain bad triangles

Bounded Angular Change

- NP-hard
  - KNAPSACK

Concluding Remarks

- We have some relitavely efficient results, and an NP-hardness proof
- Open problems
  - Well... maybe the same problem in 3D? ☺

Questions?