

## Merge Closest Chains Algorithm

The main idea of this algorithm is that vertices that are far apart from each other should less likely be part of the same edge. During the process, the algorithm builds up long polygonal chains. In each iteration, the polygonal chains that are closest to each other are merged. A single point is also considered a polygonal chain, so in the first iteration the algorithm will merge the two points with the smallest distance.

### Pseudocode

```
function merge_closest_chains(PointSet point_set) -> Polygon {
    polygonal_chains := point_set
    while (polygonal_chains.size > 1) {
        a, b := find_pair_of_nearest_chains(polygonal_chains)
        c := merge_chains(a, b)
        polygonal_chains.remove(a)
        polygonal_chains.remove(b)
        polygonal_chains.add(c)
    }
    return Polygon(polygonal_chains)
}

function find_pair_of_nearest_chains (PolygonalChainSet polygonal_chains) ->
(PolygonalChain, PolygonalChain) {
    // TODO add implementation
}

function merge_chains(PolygonalChain a, PolygonalChain b) -> PolygonalChain {
    // TODO add implementation
}

bla
```