

**3.1** Sang Won Bae describes an algorithm that computes Voronoi diagrams with line segments (allowing intersections) in  $O(n\alpha(n) \log(n) + k)$  time where  $\alpha(n)$  is the functional inverse of the Ackermann function and  $k$  is the number of non-trivial intersections. The algorithm decomposes the plane  $\mathbb{R}^2$  into faces which is bounded by  $O(n\alpha(n))$ . Then it uses divide and conquer to merge the results of each part which results in total run time of  $O(n\alpha(n) \log(n) + k)$  time.

**3.2** I would implement Menelaos I. Karavelas algorithm for Voronoi diagram with line segments. The complexity is  $O((n + k) \log^2(n))$  where  $k$  is the number of intersections. This algorithm is a randomized algorithm that inserts each point in random order. It uses the same outline as the one we discussed in class but took addition modifications to support intersecting line segments. It also needs to keep track of information of inserted line segments so that if it's been intersected we can compute the intersection.

In Bae's paper, the author mentioned that by decomposing the intersections into weak intersections, any  $O(n \log(n))$  algorithm for Voronoi diagram with weakly intersecting points can achieve run time of  $O(n + k) \log(n)$  time. Where the  $n + k$  comes from decomposing to weak intersections which have  $n + 2k$  segments. I did not find any source on this other than the author's claims. So when it comes to actually implementation I might explore this way first. Also in Bae's paper, he claimed that the theoretical lower bound for algebraic decision tree solutions of this problem is  $\Omega(n \log(n) + k)$ .

## References

S.W. Bae, *An almost optimal algorithm for Voronoi diagrams of non-disjoint line segments*, Computational Geometry, Vol. 52, Feb. 2016, pp. 34-43.

M.I. Karavelas, *A robust and efficient implementation for the segment Voronoi diagram*, Proc. 1st Int. Sympos. Voronoi Diagrams Sci. Engineer., ISVD'04, 2004, pp. 51-62