Randomized Incremental Construction of a 3D-Convex Hull

Wolfgang Mulzer

1 The Algorithm

Let $P = \{p_1, p_2, \ldots, p_n\}$ a set of n point in $\mathbb{R}^3$ in general position (i.e., no four points from P lie on a common plane). We would like to compute $\mathrm{CH}(P)$, the convex hull of P. For $i = 4, \ldots, n$, set $P_i = \{p_1, p_2, \ldots, p_i\}$. The incremental algorithm successively computes $\mathrm{CH}(P_4), \mathrm{CH}(P_5), \ldots, \mathrm{CH}(P_n)$. In each step, we find $\mathrm{CH}(P_i)$ by inserting the point p_i into $\mathrm{CH}(P_{i-1})$. Each convex hull is stored as a DCEL, representing the planar graph $G(\mathrm{CH}(P_i))$.

To facilitate the insertion of the next point, the algorithm stores *conflict information*: for each point $p \in P \setminus P_i$, there is a pointer C_p to a facet f of $\mathrm{CH}(P_i)$, so that p lies to the left of the oriented plane through f. The facet f muss be deleted when p is inserted. Thus, it is called a *conflict facet* of p. For each facet f of $\mathrm{CH}(P_i)$, we store a *conflict list* C_f that contains all points $p \in P \setminus P_i$ with $C_p = f$. Initially, $\mathrm{CH}(P_4)$ and the conflict information can be computed easily.

Now we describe how to use the conflict information in order to insert p_i into $\mathrm{CH}(P_{i-1})$. If $C_{p_i} = \perp$, then p_i lies inside of $\mathrm{CH}(P_{i-1})$, and nothing has to be done. Otherwise, we know that the facet $f = C_{p_i}$ must be deleted. We perform a breadth-first search from f in the dual graph of $G(\mathrm{CH}(P_{i-1}))$, visiting only facets of $G(\mathrm{CH}(P_{i-1})$ that are in conflict with p_i. That is, we only insert an unvisited facet f' into the BFS-queue, if p_i lies to the left of the oriented plane spanned by f'. Let D_i be the set of all facets that are in conflict with p_i, the *conflict region of* p_i. The conflict region is bounded by a simple cycle in $G(\mathrm{CH}(Pi-1))$. We call this cycle Z_i. In $\mathrm{CH}(P_i)$, we create new facets between p_i and each edge of Z_i, and we delete all facets in D_i.

It remains to update the conflict information. For this, we consider for each conflict facet $f' \in D_i$ the corresponding conflict list $C_{f'}$. For each point $q \in C_{f'}$ we do the following: similarly to p_i, we perform a BFS from f' to determine the conflict region of q. However, we restrict the BFS to the facets in D_i. In this way, we find all edges of Z_i that are incident to a facet that is in conflict with both q and p_i. For each such edge e, we test whether q is in conflict with the newly created facet of $\mathrm{CH}(P_i)$ incident to e. If so, we update the conflict pointer C_q and the conflict list for the corresponding facet. If no such facet is found, we set $C_q = \perp$. This concludes the description of one insertion.

We repeat this process until we have found $\mathrm{CH}(P_n) = \mathrm{CH}(P)$.

2 Running Time Analysis

2.1 Structural Change

Each facet is deleted at most once, so it suffices to bound the number of facets that are created. In the worst case, this can be quadratic. Using backwards analysis, however, we can show that the expected structural change is $O(n)$ (see class).

2.2 Conflict Change

While updating the conflict information, each facet f is traversed at most once for each point that is in conflict with f. Thus, the total time for the conflict change is proportional to the total number of conflicts that exist between the created facets and the points in P. We can show that this number is $O(n \log n)$ in expectation. For this, we need a three-dimensional version of Clarkson's theorem, which states that the number of oriented planes that are spanned by three points of P and that have at most k points from P to the left is at most $O(nk^2)$.