

Walking algorithms

Roman Soukal

For triangulated data in 2D and 3D including pseudotriangulations in 2D, a location problem usually means how to find a triangle, tetrahedron or pseudotriangle which contains the given point. Many efficient algorithms and data structures exist which are suitable for planar or tetrahedral meshes, such as DAGs (directed acyclic graphs), buckets, quadtrees or octrees, skip lists, data structures based on random sampling, etc. These methods mostly work in $\mathcal{O}(\log n)$ expected or worst-case time per one location query. Unfortunately, memory requirements of location data structures, although usually $\mathcal{O}(n)$, may be for large data prohibitive. Therefore, time-suboptimal approaches utilizing walk algorithms ($\mathcal{O}(n^{\frac{1}{3}})$ up to $\mathcal{O}(n^{\frac{1}{2}})$), have been developed and are used because no extra location data structures are needed and so no extra memory is consumed. This is the main advantage of walking algorithms.