Abstract

We describe a new data structure for dynamic nearest neighbor queries in the plane with respect to a general family of distance functions that includes L_p-norms and additively weighted Euclidean distances, and for general (convex, pairwise disjoint) sites that have constant description complexity (line segments, disks, etc.). Our data structure has a polylogarithmic update and query time, improving an earlier data structure of Agarwal, Efrat and Sharir that required $O(n^\varepsilon)$ time for an update and $O(\log n)$ time for a query. Our data structure has numerous applications, and in all of them it gives faster algorithms, typically reducing an $O(n^\varepsilon)$ factor in the bounds to polylogarithmic. To further demonstrate its effectiveness, we give here two new applications: an efficient construction of a spanner in a disk intersection graph, and a data structure for efficient connectivity queries in a dynamic disk graph. To obtain this data structure, we combine and extend various techniques and obtain several side results that are of independent interest. Our data structure depends on the existence and an efficient construction of *"vertical" shallow cuttings* in arrangements of bivariate algebraic functions. We prove that an appropriate level in an arrangement of a random sample of a suitable size provides such a cutting. To compute it efficiently, we develop a randomized incremental construction algorithm for finding the lowest k levels in an arrangement of bivariate algebraic functions (we mostly consider here collections of functions whose lower envelope has linear complexity, as is the case in the dynamic nearest-neighbor context). To analyze this algorithm, we improve a longstanding bound on the combinatorial complexity of the vertical decomposition of these levels. Finally, to obtain our structure, we plug our vertical shallow cutting construction into Chan's algorithm for efficiently maintaining the lower envelope of a dynamic set of planes in $\mathbb{R}^3$. While doing this, we also revisit Chan's technique and present a variant that uses a single binary counter, with a simpler analysis and an improved amortized deletion time.