A geometric data structure is called *dynamic* if it allows fast insertions and deletions of the geometric objects defining it. Unfortunately, dynamic data structures that perform well in the worst case tend to be complicated to an extend that makes them useless in practice. In the present thesis, the attempt is made to study the design of structures that perform well *on the average*, which is taken over *random update sequences*, modeled after the example of random permutations used as input sequences in randomized incremental algorithms. The key idea is that the only assumption on randomness that is made in the analysis concerns the *order* in which the geometric objects are studied, not their geometric properties with respect to each other.

A model for such random update sequences is defined and analyzed within the abstract setting for geometric algorithms defined by Clarkson and Shor. By specializing to the case where only insertions of objects into the current set are allowed, a new analysis for the complexity of randomized incremental algorithms is obtained that is somewhat simpler than the original analysis by Clarkson and Shor. The analysis is then applied to the problem of dynamically maintaining the convex hull of points in d dimensions.

In three dimensions, two cases are distinguished: If all points are known to be extreme or if only insertions of points need to be handled, points can be inserted into the current hull in time $\mathcal{O}(\log m)$ on the average, where m is the number of points in the current hull, and deletions take average time $\mathcal{O}(1)$. In the general case, both time bounds increase by an $\mathcal{O}(logm)$-factor.

In d dimensions, $d \geq 4$, insertions and deletions of points into the current hull can be done in average time $\mathcal{O}(m^{\lfloor d/2 \rfloor - 1} + m \log m)$ (actually, the dual setting, i.e. the maintenance of the intersection polytope of a set of halfspaces, is considered). The algorithm for deletions is surprisingly simple and basically amounts to repeatedly calling the insertion algorithm in a certain way.

Convex hulls in d dimensions arise as the transformation of *Voronoi diagrams* in $d-1$ dimensions, and the latter are often used for *nearest-neighbor queries*. Thus the question arises whether the techniques presented here can be used to answer nearest-neighbor queries on a dynamically changing set of points. The usual lifting map transforms a nearest-neighbor query into a ray shooting query on the upper envelope of a set of hyperplanes. Therefore, the more general problem of answering arbitrary ray shooting queries in the intersection polytope of a set of halfspaces is studied. A data structure is presented that permits insertions of halfspaces in average time $\mathcal{O}(m^{\lfloor d/2 \rfloor - 1}\text{polylog}m)$, and which permits ray shooting queries with $\mathcal{O}(\log^2 m)$ query time with high probability. The probability is again with respect to the random sequence of insertions. This result makes use of a new, improved solution to the *static* ray shooting problem.