Conclusions

In this thesis we developed algorithms for the high-dimensional L_∞-nearest-neighbor problem. We presented their average-case analysis under the assumption that the n data points are chosen independently at random under uniform distribution. We did not assume any distribution of the query point. The algorithms improve considerably the brute-force method, are simple and easy to implement, and have a low storage requirement.

The query algorithms developed in Section 3.1 are based on simple preprocessed data structures. The searching algorithm REJECT_SCAN requires $O(nd)$ storage and $O(nd \ln n)$ preprocessing time, and has an expected running time of $O\left(n \ln\left(\frac{d}{\ln n} + 1\right) + n + d \ln d\right)$. It improves the brute-force method by a factor of $\Omega(\frac{\ln(d/ln\ n+1)}{d})$. The constant in the O-term is close to one, and for many applications the dimension d is in the range of some hundreds, so the speedup factor comes close to 50, which is a considerable improvement in practice. This consideration is confirmed by the experimental comparison with the brute-force method in Section 4.5.

The searching algorithm REJECT uses the same data structure like the algorithm REJECT_SCAN, has an expected running time of $O\left(n \ln d + n + d \ln d\right)$ and can be extended to work efficiently in the external-memory model of computation.

The data structure based on a preprocessed partition of the data set into monotone sequences achieves an expected runtime that is sublinear in n and d. The query algorithm has an expected running time of $O\left(\sqrt{d} \cdot n^{1-\frac{1}{\sqrt{d}}} \cdot \ln n\right)$ for dimensions $d < \left(\frac{\ln n}{\ln \ln n}\right)^2$.

We extended our methods to solve the k-nearest-neighbor problem with an expected asymptotic runtime of $O\left(n \ln\left(\frac{d}{\ln(n/k)} + 1\right) + n + dk + d \log d\right)$. Furthermore, the expected runtime analysis of our algorithms has been generalized to other "well-behaved" probability distributions.

In Chapter 5 we developed a method which provides tradeoffs between the space complexity of the data structure and the time complexity of the query algorithm.