Table of Contents

Introduction 1

1 Fundamental structures 5
- 1.1 Convex hull . . . 7
- 1.2 Voronoi diagram . . . 8
- 1.2.1 Voronoi diagram for points in $\mathbb{E}^d$ . . . 8
- 1.2.2 Higher order Voronoi diagrams . . . 9
- 1.2.3 Voronoi diagram of line segments . . . 13
- 1.3 Arrangements . . . 14
- 1.3.1 Arrangements of hyperplanes . . . 15
- 1.3.2 Trapezoidal map . . . 16
- 1.4 Geometric transformations . . . 17

2 Static randomized incremental algorithms 21
- 2.1 Formalization of the problem . . . 23
- 2.2 A data structure : the conflict graph . . . 25
- 2.3 Techniques of analysis . . . 26
- 2.3.1 Random sampling . . . 26
- 2.3.2 Probabilistic games and Θ series . . . 32
- 2.3.3 Backwards analysis . . . 32

3 The Delaunay Tree 35
- 3.1 Structure . . . 37
- 3.2 Constructing the Delaunay triangulation . . . 39
- 3.2.1 Location . . . 39
- 3.2.2 Creating the new simplices . . . 40
- 3.3 Another structure . . . 42

4 A general structure : the Influence Graph 45
- 4.1 The general framework . . . 47
- 4.1.1 Randomized analysis of the I-DAG . . . 48
- 4.1.2 Influence graph versus Conflict graph . . . 51
- 4.1.3 Another analysis . . . 51
- 4.1.4 Removing the update conditions . . . 54

4.2 Locating with the influence graph . . . 56
4.2.1 Faster object location . . . 56
4.2.2 Queries . . . 56
4.3 Applications . . . 57
4.3.1 Convex hulls . . . 57
4.3.2 Arrangements . . . 61
4.3.3 Voronoi diagrams . . . 66
4.3.3.1 Voronoi diagrams of point sites in $\mathbb{E}^d$ . . . 66
4.3.3.2 Voronoi diagrams of line segments in the plane . . . 69
4.4 About complexity results . . . 72
4.4.1 Randomization . . . 72
4.4.2 Amortization . . . 72
4.4.3 Output sensitivity . . . 74

5 The k-Delaunay Tree **77**
5.1 The k-Delaunay Tree in two dimensions . . . 79
5.1.1 Including and excluding neighbors . . . 80
5.1.2 A semi-dynamic algorithm for constructing the order $\leq k$ Voronoi diagrams . . . 82
5.1.3 Construction of the k-Delaunay Tree . . . 82
5.1.3.1 Initialization . . . 82
5.1.3.2 Inserting a new site . . . 83
5.1.3.3 Structure of the k-Delaunay Tree . . . 85
5.1.3.4 Procedure location . . . 86
5.1.3.5 Procedure creation . . . 86
5.2 Analysis of the randomized construction . . . 88
5.2.1 Results on triangles and bicycles . . . 89
5.2.2 Analysis of the expected space used by the k-Delaunay Tree . . . 92
5.2.3 Analysis of the expected cost of Procedure location . . . 92
5.2.4 Analysis of the expected cost of Procedure creation . . . 93
5.3 l-nearest neighbors . . . 93
5.3.1 Deducing the order l Voronoi diagram from the k-Delaunay Tree ($l \leq k$) . . . 94
5.3.2 Finding the l nearest neighbors . . . 94
5.4 The k-Delaunay Tree in higher dimensions . . . 96
5.4.1 The d dimensional k-Delaunay Tree . . . 97
5.4.2 Analysis of the randomized construction . . . 97
5.5 Experimental results . . . 99
5.5.1 Influence of randomization . . . 99
5.5.2 Influence of k . . . 100
5.5.3 Influence of the point distribution . . . 100

6 Towards a fully dynamic structure **109**
6.1 Removing a site from the Delaunay triangulation 111
6.1.1 Different kinds of modified nodes 112
6.1.2 The Search step 113
6.1.3 The Reinsertion step 115
6.1.4 Analysis 123
6.1.5 d-dimensional case 126
6.1.6 Practical results in the planar case 126
6.2 Removing a segment from an arrangement 126
6.2.1 The Search step 128
6.2.2 Corners and bridges 129
6.2.3 Reinsertion of a triplet $(\mathbf{x}, T, F)$ 129
6.3 Analysis 134

7 Parallel work **137**
7.1 Accelerated static algorithms 139
7.2 Semi-dynamic algorithms 140
7.3 Dynamic algorithms 140
7.3.1 With storage of the history 140
7.3.2 Without storing the history 141
7.4 Strategies 142

Conclusion **145**

Bibliography **151**

Index **157**