

1. Record Nr.	UNINA9910538499703321
Autore	Meyer Fernand <1952->
Titolo	Topographical tools for filtering and segmentation . 2 Flooding and marker-based segmentation on node- or edge-weighted graphs // Fernand Meyer
Pubbl/distr/stampa	London : , : ISTE, Ltd. Hoboken, NJ : , : Wiley, , 2019 [Piscataway, New Jersey] : , : IEEE Xplore, , [2019]
ISBN	1-119-57515-X 1-119-57513-3 1-119-57512-5
Descrizione fisica	1 online resource (289 pages)
Collana	Digital signal and image processing series THEi Wiley ebooks.
Disciplina	551.4
Soggetti	Relief models Topographical drawing
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Notations xi -- Introduction xxv -- Part 1. Flooding 1 -- Chapter 1. Modelling Flooding in Edgeor Node-weighted Graphs 3 -- 1.1. Summary of the chapter 3 -- 1.2. The importance of flooding 4 -- 1.2.1. Flooding creates lakes 4 -- 1.2.2. Flooding for controlling watershed segmentation 4 -- 1.2.3. Flooding, razing, leveling and flattening 5 -- 1.3. Description of the flood covering a topographic surface 6 -- 1.3.1. Observing the same flooding on two levels of abstraction 6 -- 1.3.2. Modeling the two scales of flooding: at the pixel level or at the region level 7 -- 1.3.3. Modeling a flooded topographic surface as a node-weighted graph 8 -- 1.3.4. Modeling an edge-weighted graph as a tank network 15 -- 1.4. The relations between n-floodings and e-floodings 19 -- 1.4.1. Modeling flooding on two scales: the equivalence of both models 19 -- 1.5. Flooding a flowing graph 21 -- 1.5.1. Flowing graphs: reminder 21 -- 1.5.2. Starting from an edge-weighted graph $G[nil, \η]$ 22 -- 1.5.3. Starting from a node-weighted graph $G[\ν, nil]$ 24 -- 1.5.4. Summarizing 24 --

Chapter 2. Lakes and Regional Minima 27 -- 2.1. Summary of the chapter 27 -- 2.2. Lakes from e-floodings and n-floodings 27 -- 2.2.1. e-flooding of graphs $G[\text{nil}, \epsilon]$ 27 -- 2.2.2. n-flooding of graphs $G[\nu, \text{nil}]$ 28 -- 2.3. Regional minimum lakes and full lakes 29 -- 2.3.1. e-floodings of graphs $G[\text{nil}, \epsilon]$ 29 -- 2.3.2. n-floodings of graphs $G[\nu, \text{nil}]$ 30 -- 2.4. Coherence between the definitions of lakes in $G[\nu, \text{nil}]$ and in $G[\text{nil}, \delta; \nu]$ 31 -- Chapter 3. Among all Possible Floodings, Choosing One 33 -- 3.1. Summary of the chapter 33 -- 3.2. Various mechanisms for selecting a particular flooding 34 -- 3.2.1. Dominated flooding in node- and edge-weighted graphs 34 -- 3.2.2. Dominated flooding in node- and edge-weighted graphs 36 -- 3.2.3. Dominated flooding as a function of the ceiling function 37 -- 3.3. The topography of dominated flooding 37 -- 3.3.1. The regional minima of dominated flooding in an edge-weighted graph $G[\text{nil}, \epsilon]$ 38. 3.3.2. The regional minima of dominated n-flooding in node-weighted graphs $G[\nu, \text{nil}]$ 39 -- 3.3.3. Algorithmic consequences 41 -- 3.4. Computing dominated flooding by local adjustments 43 -- 3.4.1. The case of edge-weighted graphs $G[\text{nil}, \epsilon]$ 43 -- 3.4.2. The case of node-weighted graphs $G[\nu, \text{nil}]$ 44 -- 3.4.3. Software or hardware implementation of Berge's algorithm 45 -- Chapter 4. Flooding and Flooding Distances 49 -- 4.1. Summary of the chapter 49 -- 4.2. Flooding distances 49 -- 4.2.1. The flooding distance associated with the lakes of node- or edge-weighted graphs 49 -- 4.2.2. Characterization of the flooding distance 50 -- 4.2.3. Flooding distances on a graph or a tree 52 -- 4.2.4. The shortest flooding distances 53 -- 4.2.5. Dominated flooding and flooding distances 56 -- 4.3. The shortest path algorithms for computing dominated flooding 66 -- 4.3.1. Computing the shortest flooding distance with the Moore-Dijkstra algorithm 66 -- 4.4. The flooding core-expanding algorithm 75 -- 4.4.1. The first version of the core-expanding algorithm applied to the augmented graph G° ; 76 -- 4.4.2. The second version of the core-expanding algorithm applied to the initial graph G 78 -- 4.4.3. The third version of the core-expanding algorithm applied to the initial graph G 79 -- 4.5. Marker-based segmentation 81 -- 4.5.1. The case of a node-weighted graph $G(\nu, \text{nil})$ 81 -- Chapter 5. Graph Flooding via Dendrograms 83 -- 5.1. Summary of the chapter 83 -- 5.2. Introduction 84 -- 5.3. Dendrograms: reminder 86 -- 5.3.1. The structure associated with an order relation 86 -- 5.3.2. Dendrograms 87 -- 5.3.3. Stratification index and partial ultrametric distances (PUD) 88 -- 5.4. The hierarchy of lake zones 89 -- 5.4.1. The lake zones of an edge-weighted graph $G(\text{nil}, \epsilon)$ 89 -- 5.4.2. The hierarchy of lake zones, i.e. the closed balls of χ ; 92 -- 5.4.3. Representing of hierarchy of lake zones 94 -- 5.5. The law of communicating vessels 98 -- 5.5.1. The flooding levels in connected subgraphs and closed balls 99. 5.6. Dominated flooding on the dendrogram of lake zones 100 -- 5.6.1. Notations 100 -- 5.6.2. Incidence of the ceiling function on the dendrogram flooding levels 100 -- 5.6.3. Finding the flooding level of a leaf 102 -- 5.6.4. Parallel processing for flooding the dendrogram 105 -- 5.6.5. Strategies for flooding the dendrogram of lake zones 106 -- 5.7. Constructing and flooding a binary dendrogram 111 -- 5.7.1. Two dendrograms representing the same hierarchy 111 -- 5.7.2. Constructing a binary dendrogram representing a hierarchy 112 -- 5.7.3. Flooding a binary dendrogram 113 -- 5.8. A derived algorithm for dominated flooding 113 -- 5.8.1. Algorithm "ancestor-flood without constructing the dendrogram"; 117 -- 5.8.2. Illustration 117 -- Part 2. Modeling a Real Hydrographic Basin 119 -- Chapter 6.

The Hydrographic Basin of a Digital Elevation Model 121 -- 6.1. Summary of the chapter 121 -- 6.2. Preprocessing the digital elevation model 121 -- 6.2.1. Suppressing the spurious regional minima 121 -- 6.2.2. Creating an ∞ -steep digraph 123 -- 6.2.3. Local pruning for extracting marked rivers 126 -- 6.2.4. Extracting all rivers 128 -- 6.2.5. Labeling sources and rivers 129 -- 6.2.6. Detection of crest lines 131 -- 6.2.7. Detecting the upstream of sources 132 -- 6.2.8. Analyzing the tree structure of rivers 133 -- 6.2.9. Constructing the catchment zones of riverlets 137 -- Part 3. Watershed Partitions 139 -- Chapter 7. Minimum Spanning Forests and Watershed Partitions 141 -- 7.1. Summary of the chapter 141 -- 7.2. Flooding distance, minimum spanning trees and forests 142 -- 7.2.1. Flooding distances 142 -- 7.2.2. Flooding distance on the minimum spanning tree of the graph $G(\text{nil}, \eta)$ 143 -- 7.2.3. Characterizing the MST 145 -- 7.3. Minimum spanning forests rooted in markers 146 -- 7.3.1. Constructing the minimum spanning forest 147 -- 7.3.2. Converting the minimum spanning forest into a minimum spanning tree 149 -- 7.4. Watershed partitions of weighted graphs 150. 7.4.1. Catchment basins and watershed partitions 150 -- 7.4.2. Flowing paths and catchment basins 151 -- 7.5. Minimum spanning forests rooted in the regional minima 151 -- 7.5.1. A minimum spanning forest corresponds to each watershed partition 151 -- 7.5.2. Inversely, each watershed partition spans a minimum spanning forest 154 -- 7.5.3. A rather unexpected watershed partition 156 -- 7.6. A manifold of different watershed partitions 159 -- 7.6.1. Catchment zones and catchment basins 159 -- 7.7. Reducing the number of watershed partitions 160 -- 7.7.1. Minimum spanning forests of k -steep or ∞ -steep graphs 163 -- 7.7.2. The waterfall hierarchy 168 -- 7.7.3. Usefulness of the waterfall hierarchy 171 -- Chapter 8. Marker-based Segmentation 175 -- 8.1. Dominated flooding and minimum spanning forests 177 -- 8.1.1. Dominated flooding 177 -- 8.1.2. Minimum spanning forests 177 -- 8.1.3. Illustration 178 -- 8.1.4. Minimum spanning forests and dominated flooding 179 -- 8.2. Constructing a minimum spanning forest rooted in the markers 183 -- 8.2.1. Algorithms for constructing a minimum spanning forest 183 -- 8.2.2. Increasing the selectiveness of Prim's algorithm 186 -- 8.2.3. Marker-based segmentation of node-weighted graphs 187 -- 8.2.4. Derived algorithms 190 -- 8.3. Marker-based segmentation after flooding the graph 194 -- 8.3.1. Segmenting the dominated flooding of a graph 194 -- 8.3.2. The case of an edge-weighted graph 194 -- 8.3.3. Constructing a k -steep or ∞ -steep watershed partition for a node-weighted graph $G(\nu, \text{nil})$ 200 -- 8.4. Directly constructing a marker-based ∞ -steep watershed partition with the core expanding algorithm 201 -- 8.5. The early days of marker-based segmentation 202 -- 8.5.1. The level-by-level construction of a watershed 203 -- 8.6. A two scale marker-based segmentation 205 -- 8.7. Instant marker-based segmentation 205 -- 8.7.1. Why and when we need instant marker-based segmentation 205 -- 8.7.2. The reef and cascade distance 206. 8.7.3. Computing the reef and cascade distance for all pairs of nodes in $G(\text{nil}, \eta)$ 209 -- 8.7.4. Computing the smallest reef and cascade distances between all couples of nodes in a graph 212 -- Conclusion 217 -- Appendix 227 -- References 239 -- Index 241.

Sommario/riassunto

Mathematical morphology has developed a powerful methodology for segmenting images, based on connected filters and watersheds. We have chosen the abstract framework of node- or edge-weighted graphs for an extensive mathematical and algorithmic description of these tools. Volume 2 proposes two physical models for describing valid

flooding on a node- or edge-weighted graph, and establishes how to pass from one to another. Many new flooding algorithms are derived, allowing parallel and local flooding of graphs. Watersheds and flooding are then combined for solving real problems. Their ability to model a real hydrographic basin represented by its digital elevation model constitutes a good validity check of the underlying physical models. The last part of Volume 2 explains why so many different watershed partitions exist for the same graph. Marker-based segmentation is the method of choice for curbing this proliferation. This book proposes new algorithms combining the advantages of the previous methods which treated node- and edge-weighted graphs differently. '
