Image Structure from Monotonic Dual Graph Contraction*

Roman Englert and Walter Kropatsch

Vienna University of Technology
Institute for Computer Aided Automation
Pattern Recognition and Image Processing Group
Treitlstraße 3, 183/2, A-1040 Vienna, Austria
krw@prip.tuwien.ac.at
http://www.prip.tuwien.ac.at

Abstract. The qualitative structure of images is much like the qualitative structure of landscapes. 'Critical points' of a landscape are the summits, the immits, and the saddle points. These points are connected through special curves on the surface of the landscape. The new approach computes this basic qualitative structure of an image or a landscape from the neighborhood structure of a sampled grid by a process called monotonic dual graph contraction (MDGC). The vertices of the graphs store information about gray level or height as attributes. Edges represent surface curves connecting the vertices. MDGC successively removes non-extrema from the original graphs while it preserves the connectivity between extrema and the connectivity level, a new property expressing the least height difference when moving from one extremum to another extremum. Since the graph represents a surface it is planar and the dual graph is well defined. MDGC performs simplifications such that in one graph all local maxima survive and in the dual all local minima survive. Hence we call them *'maximum graph'* and *'minimum graph'* respectively. The focus in this paper is on the description of the neighborhood and the hierarchy of the local extrema of height. Monotonic properties of the gray level image are preserved during the contraction process. The implementation of the approach is described and experimental results are discussed.

1 Introduction

In this paper the structure of images from the monotonic contraction of a pair of dual graphs is described. This method provides an interpretation of properties like neighborhoods and hierarchies of features. As application the sampling grid of the pixels in a two-dimensional digital gray level image is replaced by a pair of dual graphs adapted to the image's critical points. If the gray levels are interpreted as heights, the image can be regarded as a digital terrain model (DTM).

* The authors gratefully acknowledge the assistence of Roland Glantz in the preparation of this paper. This work is supported by the Austrian Science Foundation (FWF) under grant S7002-MAT.

Koenderink[Koe84,Kv97] defined the qualitative structure of a digital terrain in terms of: *summits* as the local maxima of height, *immits* as the local minima of height, *topological curves* as lines which connect summits or immits with each other, and *saddle points* on topological curves.

In our approach the structure is computed in two steps: First, the image is transformed into an attributed graph, where the vertices represent pixels, the edges represent neighborhoods of pixels, and the vertex and edge attributes are gray levels. In the main step, this graph is contracted until it consists of (a) vertices which represent summits and faces which represent immits; (b) these extrema are connected by curves on the surface passing through a saddle.

The proposed approach has several merits: for reasons of speed the contraction is performed in parallel in both the graph and in the corresponding dual graph. Furthermore, this *dual graph contraction* is based on a theory with well-known properties [Kro95]. As novelty, the dual graph contraction performed in this paper preserves monotonic properties like height differences of critical points and, additionally, it results in a compact representation of the structure.

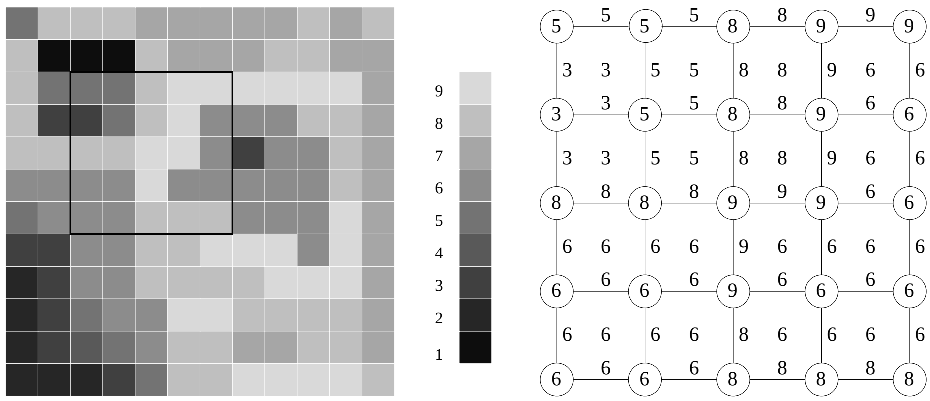
The structuring of gray level images can also be achieved by other approaches. Hereby, watershed transformations are in the center of efficient approaches [MR98]. The monotonic dual graph contraction (MDGC) within this paper differ from those in several points:

1. MDGC computes a dual pair of contracted graphs which describe the neighborhood and the hierarchy of the summits and immits.
2. Watersheds are represented by a set of pixels. MDGC computes a compact representation of a DTM, where the summits and immits are represented by vertices and the topological curves are represented by paths.
3. Watershed transformations have to take into account the plateaus (more precisely the behavior of water flow in the interior of a plateau) and this requirement must be fulfilled by additional effort. This need not to be done in MDGC.
4. Due to the attributes an explicit height information of the summits and immits is provided within MDGC.

The remainder of the paper is organized as follows: In Section 2 the basic concepts of MDGC are defined in detail. The algorithm MDGC and the properties are discussed, too. The implementation of MDGC is described in Section 3. Afterwards, experimental results are presented in Section 4. We conclude in Section 5 with an outlook for future work.

2 Monotonic Dual Graph Contraction

Our application, the computation of the image structure, relies on the contraction of a pair of dual graphs $(G, \overline{G})$. In the following the pair of dual graphs and the dual graph contraction are defined. Afterwards, the monotony preserving property is provided and the correctness is proved. The basic concepts from the field of graph theory are adopted from [TS92].



(a) (b)

Fig. 1. (a) The gray levels of the pixels. (b) The maximum graph of the marked sub-image (inside the black border), where the vertices are represented as circles. The numbers in the vertices and at the edges indicate the vertex values and the edge values of the maximum graph. The numbers in the middle of the square regions are the attributes of the vertices of the minimum graph.

The array of image pixels (Fig. 1(a)) is represented by an attributed graph, where the vertices and the edges contain additional information [ECS98]. Each pixel of the image is represented by a vertex of the graph. A vertex is adorned with an attribute $\zeta(\cdot)$, which is in our application the gray level of the corresponding pixel (short: *vertex value*). Vertices are connected by an edge if their corresponding pixels are neighbors[1]. Analogously, the edges have attributes $\xi(\cdot)$ (short: *edge values*). Their definition is motivated by the problem of how to get, e.g., from one summit to another summit on a topological curve, without descending into deep valleys. We are looking for a path, the smallest edge value of which is maximal with respect to the smallest edge values along the alternative paths [GEK99b]. This edge value is called *max-connectivity level*. We first introduce formally the dual maximum and minimum graphs, and then define for each the corresponding connectivity level.

Definition 1 (Maximum Graph, Minimum Graph). *A* MAXIMUM GRAPH $G = (V, \zeta, E, \xi)$ *consists of a vertex set* V*, a mapping* ζ *for the vertex values, an edge set* E*, and a mapping* ξ *for the edge values if the following constraint on the attributes are satified:*

$$\forall e = (x, y) \in E \quad \min\{\zeta(x), \zeta(y)\} \geq \xi(e)$$

[1] 4-neighborhood is used to make the graph planar.

The MINIMUM GRAPH $\overline{G} = (\overline{V}, \overline{\zeta}, \overline{E}, \overline{\xi})$ *is the dual graph of* G *which consists of a vertex set* $\overline{V}$*, a mapping* $\overline{\zeta}$ *for the vertex values, an edge set* $\overline{E}$*, and a mapping* $\overline{\xi}$ *for the edge values if the following constraint on the attributes are satified:*

$$\forall \overline{e} = (\overline{x}, \overline{y}) \in \overline{E} \quad \max\{\overline{\zeta}(\overline{x}), \overline{\zeta}(\overline{y})\} \leq \overline{\xi}(\overline{e})$$

The background vertex of $\overline{V}$ *is denoted as* $\overline{v}_\infty$.

The duality in the above definition holds for the structure of the graphs but not for the attribute values. In our application an image with pixels P and integer gray levels L is given (Fig. 1): The maximum graph $G = (V, \zeta, E, \xi)$ consists of a vertex set V (bijectively mapped to P), a mapping $\zeta : V \to L$, an edge set E (vertices are connected if their corresponding pixels are 4-neighbors), and a mapping $\xi : E \to L$. Initially $\xi(e) = \xi(v, w) = \min\{\zeta(v), \zeta(w)\}$ is chosen. The minimum graph $\overline{G} = (\overline{V}, \overline{\zeta}, \overline{E}, \overline{\xi})$ is the dual graph of G which consists of

- a vertex set $\overline{V}$ (bijectively mapped to the faces of G),
- a mapping $\overline{\zeta}$ of the vertices to the smallest edge value of the edges surrounding the face,
- an edge set $\overline{E}$ (dual vertices are connected if their faces share a common boundary segment),
- and a mapping $\overline{\xi} : \overline{E} \to L$ with $\overline{\xi}(\overline{e}) = \xi(e)$ for all dual edges $\overline{e} \in \overline{E}$.

For an illustration of dual edges see Fig. 3. Summarizing, a vertex value of the maximum graph stores the maximum value of its receptive field, and a vertex value of the minimum graph stores the minimum value of its receptive field. Notice, the concepts of maximum and minimum graphs enable one to encode any features and not only gray levels.

Definition 2 (Max-Connectivity Level). *Given a maximum graph* $G = (V, \zeta, E, \xi)$*. Let* $C(v, w)$ *be the set of all paths between a pair of distinct vertices* (v, w)*,* $v \in V$ *and* $w \in V$*. The* MAX-CONNECTIVITY LEVEL $maxCL(v, w)$ *is the heighest point one has to descend when moving from* v *to* w*:*

$$maxCL(v, w) = \max\{\min\{\xi(e) \in C(v, w)\} | C(v, w)\}.$$

Analogously, a min-connectivity level is defined for minimum graphs.

Definition 3 (Min-Connectivity Level). *Given a minimum graph* $\overline{G} = (\overline{V}, \overline{\zeta}, \overline{E}, \overline{\xi})$*. Let* $\overline{C}(\overline{v}, \overline{w})$ *be the set of all paths between a pair of distinct vertices* $(\overline{v}, \overline{w})$*,* $\overline{v} \in \overline{V}$ *and* $\overline{w} \in \overline{V}$*. The* MIN-CONNECTIVITY LEVEL $minCL(\overline{v}, \overline{w})$ *of the pair of distinct vertices* $(\overline{v}, \overline{w})$ *is the lowest height which has to be climbed when moving from* $\overline{v}$ *to* $\overline{w}$*:*

$$minCL(\overline{v}, \overline{w}) = \min\{\max\{\overline{\xi}(\overline{e} \in \overline{C}(\overline{v}, \overline{w})\} | C(v, w)\}.$$

In the following the basic operations for the contraction of graphs are defined in three steps: First, the dual graph contraction for planar graphs (Section 2.1). Then, the monotonic contraction operations for edges and vertices of maximum and minimum graphs (Section 2.2). Third, the approach MDGC for the monotonic dual graph contraction (Section 2.3).

2.1 Dual Graph Contraction

The operation of *dual graph contraction* is defined for an embedded, planar graph G and the dual graph $\overline{G}$ of G. It is controlled by the following decimation parameters [Kro95]:

Definition 4 (Decimation Parameter, Contraction Kernel). *Given a graph G. A subgraph D of G is a* DECIMATION PARAMETER *of G, if and only if D is a spanning forest of G. The connected components (trees) of D are called* CONTRACTION KERNELS. *The roots of the trees are called* SURVIVING VERTICES. *All other nodes of the trees are called* NON-SURVIVING VERTICES. *If G has a background vertex v_∞, then v_∞ must survive.*

In the subsequent operation every contraction kernel of D shrinks to a single vertex, the root, within G (Figures 2(a) and 2(b)), while all other connections are preserved [Kro95]. Notice that the contraction of an edge requires the deletion of its dual edge.

Definition 5 (Dual Graph Contraction). *Given an embedded pair of dual graphs $(G, \overline{G})$ and decimation parameters D_e for the contraction of edges in G and D_f for the contraction of faces in $\overline{G}$.* DUAL GRAPH CONTRACTION *(DGC) consists of two phases:*

1. DUAL EDGE CONTRACTION *described by a function*

$$C_e : (G, \overline{G}) \to C_e[(G, \overline{G}), D_e] = (G', \overline{G'}) \qquad , and$$

2. DUAL FACE CONTRACTION $C_f : (\overline{G'}, G') \to C_f[(\overline{G'}, G'), D_f] = (\overline{G''}, G'')$.

Figures 2(c) and 2(d) demonstrate an example of the DGC: The contraction kernels for the first phase and the results are shown. During the contraction a non-surviving vertex is *identified* with a surviving vertex which is the root of the contraction kernel. Note, the dual edge contraction deletes dual edges in G and $\overline{G}$ (Fig. 2(b) and 3). Afterwards the second phase can be executed in order to remove degenerated faces. Degenerated faces are, e.g., cycles of length less than three. DGC has been shown to preserve the connectivity, the structure and the planarity of the graphs [Kro95]. Applications for the DGC are described in [GEK99a].

In the following we define the generation of decimation parameters for a maximum graph and for the corresponding minimum graph. Decimation parameters are based on the decision whether an edge of a maximum graph is max-contractible or not. An edge of a minimum graph must be min-contractible for the monotonic dual graph contraction.

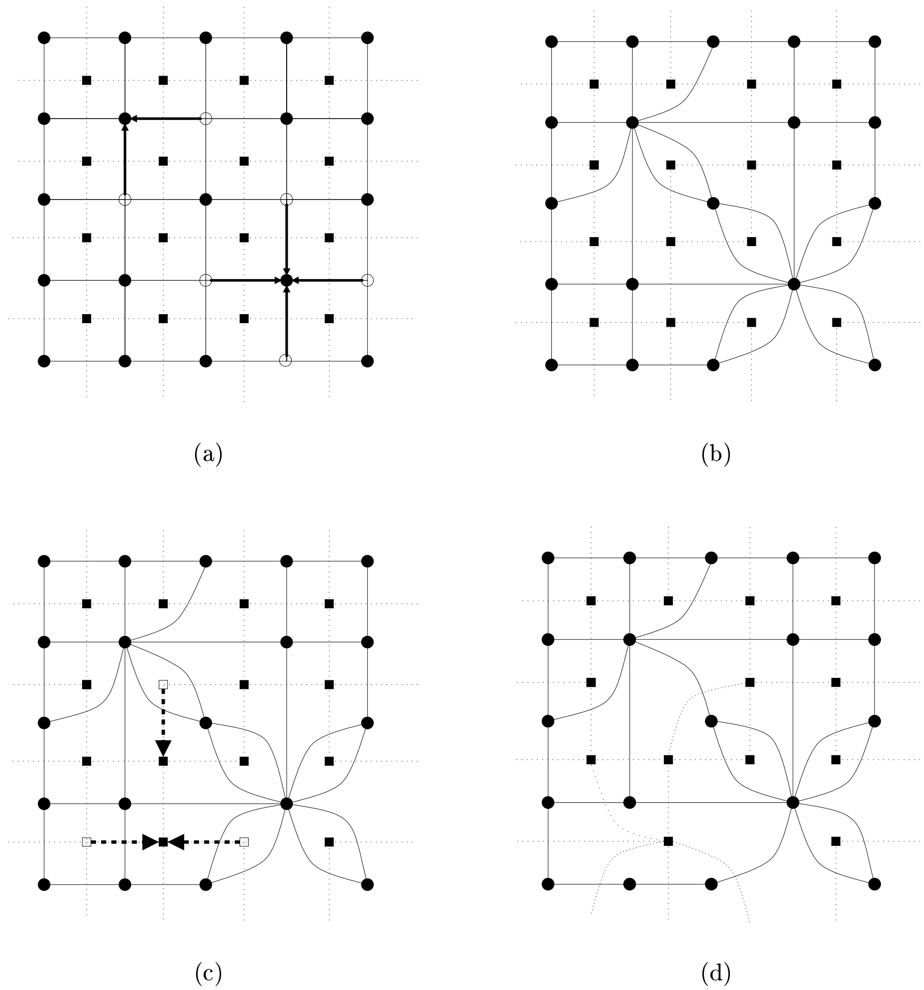


Fig. 2. Part (a) shows an embedded graph (vertices = '•', edges = '–') and its dual graph (vertices = '▪', edges = '-'), where the vertex representing the background region and all its incident edges are omitted for sake of simplicity. The contraction kernels are marked ('∘' are non-surviving vertices and '→' point at the surviving vertices). Part (b) shows the result of the contraction. The parts (c) and (d) depict a contraction of the dual graph ('□' are non-surviving vertices and '->' point at the surviving vertices).

2.2 Monotonic Contraction Operations for Maximum and Minimum Graphs

Definition 6 (Max-contractible, Min-contractible). *Given a maximum graph* $G = (V, \zeta, E, \xi)$ *and a minimum graph* $\overline{G} = (\overline{V}, \overline{\zeta}, \overline{E}, \overline{\xi})$. *Let* $e = (v, w) \in E$ *and* $\overline{e} = (\overline{v}, \overline{w}) \in \overline{E}$ *be edges,* v *and* $\overline{v}$ *being non-surviving vertices,* $\overline{v}$ *not being the background face, and* w *and* $\overline{w}$ *being surviving vertices.*
The edge e *is* MAX-CONTRACTIBLE, *if and only if*

$$\zeta(v) \leq \xi(e) \leq \zeta(w).$$

The edge $\overline{e}$ *is* MIN-CONTRACTIBLE, *if and only if*

$$\overline{\zeta}(\overline{v}) \geq \overline{\xi}(\overline{e}) \geq \overline{\zeta}(\overline{w}).$$

As final step of the contraction of an edge within a maximum graph or a minimum graph the edge values of the edges incident to the surviving vertex are updated as follows:

Definition 7 (Max-dual Contraction, Min-dual Contraction). *Given a maximum graph* $G = (V, \zeta, E, \xi)$. *A* MAX-DUAL CONTRACTION *of a max-contractible edge* $e = (v, w) \in E$ *with surviving vertex* w *and non-surviving vertex* v *is a contraction of* e, *e.g. any edge* $e' = (x, v), e' \neq e$, *becomes a new edge* (x, w), *and the attributes of the surviving elements remain unchanged. Analogously is defined: Given a minimum graph* $\overline{G} = (\overline{V}, \overline{\zeta}, \overline{E}, \overline{\xi})$. *A* MIN-DUAL CONTRACTION *of a min-contractible edge* $\overline{e} = (\overline{v}, \overline{w}) \in \overline{E}$ *with surviving vertex* $\overline{w}$ *and non-surviving vertex* $\overline{v}$ *not being the background face is a contraction of* $\overline{e}$, *e.g. any edge* $\overline{e'} = (\overline{x}, \overline{v}), \overline{e'} \neq \overline{e}$ *becomes a new edge* $(\overline{x}, \overline{w})$, *and the attributes of the surviving elements remain unchanged.*

2.3 Monotonic Dual Graph Contraction

All the above defined local operations consider edges and vertices of maximum and minimum graphs. Finally, the approach MDGC can be formalized as follows:

Definition 8 (Monotonic Dual Graph Contraction). *Given a maximum graph* $G = (V, \zeta, E, \xi)$ *and the corresponding minimum graph* $\overline{G} = (\overline{V}, \overline{\zeta}, \overline{E}, \overline{\xi})$. *The* MONOTONIC DUAL GRAPH CONTRACTION *consists of two phases:*

1. MAX-DUAL CONTRACTION *of* $(G, \overline{G})$ *with max-contractible edges of the maximum graph* G *as selected decimation parameters, and*
2. MIN-DUAL CONTRACTION *of* $(\overline{G}, G)$ *with min-contractible edges of the minimum graph* $\overline{G}$ *as selected decimation parameters.*

The following property ensures the correctness of MDGC:

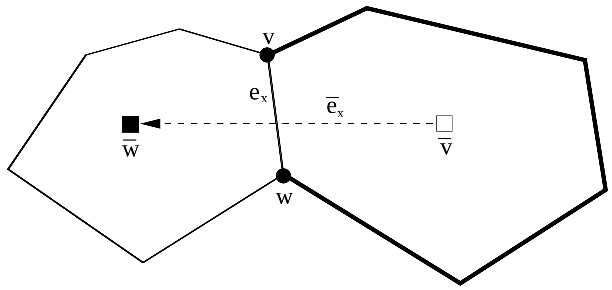


Fig. 3. Min-dual contraction of $\overline{e_x} = (\overline{v}, \overline{w})$: the bold path $B(\overline{v}) \setminus \{e_x\} \subset C(v, w)$ preserves the max-connectivity level $maxCL(v, w)$.

Proposition 1. *The min-dual contraction preserves the max-connectivity levels.*

Proof: Given a maximum graph $G = (V, \zeta, E, \xi)$ and a corresponding minimum graph $\overline{G} = (\overline{V}, \overline{\zeta}, \overline{E}, \overline{\xi})$. Note that $\overline{G}$ is the dual graph to G. We show that the min-dual contraction of an edge $\overline{e_x} = (\overline{v}, \overline{w}) \in \overline{E}$ in $\overline{G}$ preserves the max-connectivity levels $maxCL(v, w)$ in G (see Fig. 3): The contraction of $\overline{e_x}$ in $\overline{G}$ goes along with the deletion of its dual edge $e_x = (v, w)$ from G. It suffices to prove, that the max-connectivity level $maxCL(v, w)$ is not decreased if edge e_x is removed. In other words we have to show that $maxCL(v, w) \geq \xi(e_x)$ for the edge e_x which is also a (short) path between v and w. Since the face $\overline{v}$ is not the background face it is surrounded by a closed path $B(\overline{v})$ ('boundary') containing edge e_x. Let us consider the edge values of the alternative path $B(\overline{v}) \setminus \{e_x\}$ which is also a path from v to w. In Fig. 3 this path is depicted bold.

Initially, the edge values $\xi(e_B) = \overline{\xi}(\overline{e_B})$ around a face are never smaller than the face value (cf. the initialization of the maximum and minimum graphs and Fig. 1). This property is not destroyed by min-dual contraction, since a min-contractible edge is always contracted into the face with the smaller value. It is also not destroyed by max-dual contraction, since the edge values may only increase during update.

Since MDGC preserves the property that faces cannot receive attributes higher than their bounding edges we have $\overline{\xi}(\overline{e_B}) \geq \overline{\zeta}(\overline{v})$ for all edges $e_B \in B(\overline{v}) \setminus \{e_x\}$, and also $maxCL(v, w) \geq \overline{\zeta}(\overline{v})$. Furthermore, $\overline{e_x}$ must be min-contractible, and consequently, $\overline{\zeta}(\overline{v})$ is also an upper bound for $\overline{\xi}(\overline{e_x}) = \xi(e_x)$, QED. □

A similar proof yields that the max-dual contraction of an edge in the maximum graph preserves the min-connectivity levels in the corresponding minimum graph.

3 Implementation of MDGC

The algorithm of MDGC has a simple structure: as input a gray level image is taken and as output a structure consisting of summits and immits is computed. Both contractions, the monotonic and the dual monotonic, are applied to the maximum graph and the minimum graph until no further contraction is possible. Note, both the min-dual contractions and the max-dual contractions can be performed in parallel [Kro95]. The method converges in a logarithmic number of steps since the length of the paths between extrema shrinks by a factor of at least two at every (parallel) step.

The implementation of MDGC is based on LEDA [MN99, Library of Efficient Data Structures and Algorithms] and a tool for the dual contraction of graphs [KBBS98]. In contrast to this tool within MDGC the contraction is not executed with the graphs, merely trees which contain the contraction kernels are constructed as follows: The non-surviving edges together with their corresponding edges in the dual are marked. Before contraction, each graph vertex points at a tree consisting of a single tree vertex. The contraction of an edge e is now expressed by the linking of the two trees belonging to the end vertices of edge e. The root of the new tree is set to the tree vertex the surviving graph vertex is pointing at. At each step of the contraction process, the set of surviving graph vertices comprises all graph vertices to point at a tree root. A surviving edge, however, is represented by the corresponding *bridge*, i.e. an edge of the graph, which has not been marked yet. The surviving vertices connected by a surviving edge are identified via the roots of the trees, the end vertices of the corresponding bridge are pointing at. The trees are represented by a collection of trees using the LEDA data structure *dynamic_trees*, where each operation takes $O(\log^2 n)$ amortized expected time, n being the number of vertices. Working with this collection of trees is faster than executing a contraction in a (dual) graph, since edges and vertices need not to be removed in the graph and its dual. Finally as graphic output, the trees are drawn representing the contracted graph and its dual as demonstrated in the following section.

4 Experimental Results

The algorithm MDGC is applied to a test image containing two sole immits (center and bottom) and a pair of nested immits on the upper left (Fig. 1(a)). As a result we expect two sole loops and a pair of nested loops on the upper left in the contracted final graph (Fig. 4). This result will reflect the neighborhood and hierarchy of the local extrema of height. A part of the initial maximum graph is shown in Fig. 1(b).

Fig. 4 shows the computed topological curves, when neither the maximum graph nor the minimum graph is contractible anymore. The line segments represent the edges of the trees. Here the tree roots are identified with the surviving vertices.

The union of all non-surviving edges of the contraction trees and all bridges (Fig. 4) describe the watersheds. Fig. 5 depicts the contracted final graph and

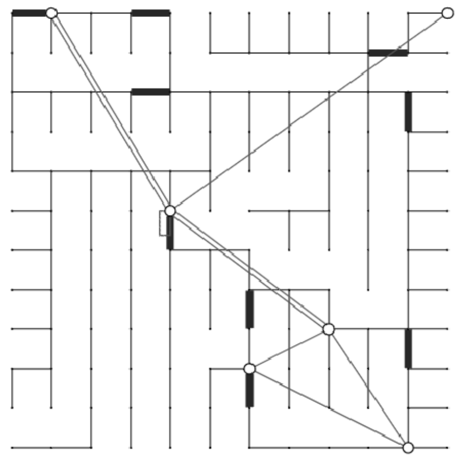


Fig. 4. Topological curves computed by MDGC (surviving vertices = '∘', non-surviving edges pointing at the surviving vertex = '→', bridge = '▬').

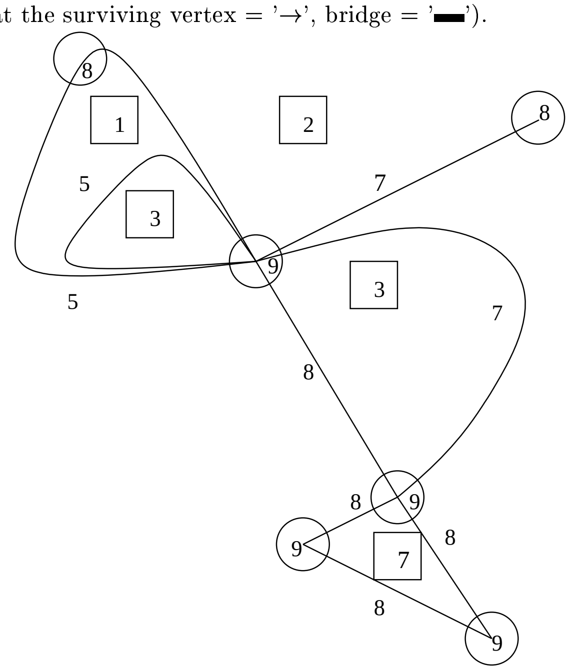


Fig. 5. Contracted maximum graph (vertices = '∘', undirected (curved) edges = '—') and the surviving vertices '□' of the minimum graph (displayed without edges). The vertex labels refer to Fig. 1(b).

the vertices of its dual. Comparing the final maximum graph (Fig. 5) with the gray level image (Fig. 1(a)), we summarize the following results:

1. Each edge of the final maximum graph represents either a topological curve bordering an immit or a topological curve connecting two immits.
2. The two nested immits close to the upper left corner of the image are represented by two nested cycles in the final maximum graph.
3. The cycles from the previous item are loops, because there exists a single saddle point on each of the topological curves bordering the immits.
4. The fact that the immit on the bottom is almost replenished, is reflected by the small differences of the corresponding attributes in the final graph.

5 Conclusions

In this paper we have proposed a new approach to the computation of extrema within attributed graphs. For the representation the class of minimum and maximum graphs has been defined. The approach MDGC has been applied to the structure of gray level images. The structure is represented by a pair of dual graphs. It is compact since each vertex of the contracted graphs represents either a summit or an immit. The edges of the graphs represent contracted topological curves. The graph contains also the information about the local extrema of height on the topological curves.

Our approach outperforms watershed approaches since the neighborhood and the hierarchy of the summits and immits is computed, and additionally, information about topological curves is provided through the attributes of the graphs. We believe that MDGC is a powerful technique which has been applied to the segmentation of gray level images, and additionally, that image structuring methods based on watersheds can profit from. An important topic of future research is the use of real data (images).

The current approach does not cope with noise. A single outlier, e.g. a wrong local maximum or minimum, may appear in the final representation. Furthermore, even small differences in the attribute values result in many unnecessary and spurious vertices. Hence, a future goal will be to extend the present approach with a concept of "importance" for a given vertex (summit or immit) which can be related to the relative differences within a local neighborhood. A similar criterion has been used in the scale-space approach of Lindeberg [Lin94].

A drawback of our concept goes back to the fact, that the saddles in the digital elevation model are not represented as vertices. We cannot properly describe saddles, which lead to more than two summits, if one follows the ascending crest lines (see Fig. 6). Fig. 6 also makes clear, that the final graph is not unique. The bent edge might as well be situated at the left side. Our future work will aim at the proper representation of the saddles. For this purpose we will have a closer look at the contraction kernels.

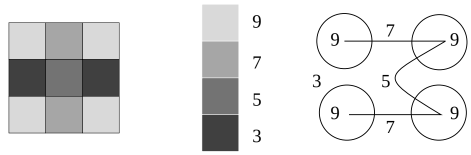


Fig. 6. The gray levels of the pixels (left) and the final graph representing the image (right).

References

[ECS98] Roman Englert, A. Cremers, and J. Seelmann-Eggebert. Recognition of polymorphic patterns in parameterized graphs for 3d building reconstruction. *Computing, Supplementum: Graph Based Representations in Pattern Recognition*, No. 12:pp. 11–20, 1998.

[GEK99a] Roland Glantz, Roman Englert, and Walter G. Kropatsch. Contracting distance maps of pores to pore networks. In *2nd Int. IAPR Workshop on Graph-based Representation*, pages 112–121. Proc. of the 4th Computer Vision Winter Workshop, TU-Wien, Austria, 1999.

[GEK99b] Roland Glantz, Roman Englert, and Walter G. Kropatsch. Dual image graph contractions invariant to monotonic transformations of image intensity. In Walter G. Kropatsch and Jean-Michel Jolion, editors, *2nd Int. IAPR Workshop on Graph-based Representation*, pages –. OCG-Schriftenreihe, Österreichische Computer Gesellschaft, 1999. to appear.

[KBBS98] Walter G. Kropatsch, Mark Burge, Souheil Ben Yacoub, and Nazha Selmaoui. Dual graph contraction with LEDA. *Computing, Supplementum: Graph Based Representations in Pattern Recognition*, No. 12:pp. 101–110, 1998.

[Koe84] Jan J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[Kro95] Walter G. Kropatsch. Building Irregular Pyramids by Dual Graph Contraction. *IEE-Proc. Vision, Image and Signal Processing*, Vol. 142(No. 6):pp. 366–374, December 1995.

[Kv97] Jan J. Koenderink and A. van Doorn. Image structure. In E. Paulus and F. Wahl, editors, *Mustererkennung 1997*, pages 3–35, Braunschweig, Germany, 1997. DAGM - Symposium, Springer Verlag.

[Lin94] Tony P. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, Boston, USA, May 1994.

[MN99] K. Mehlhorn and S. Näher. *The LEDA Platform of Combinatorial and Geometric Computing*. Cambridge University Press, Cambridge, U.K., 1999.

[MR98] A. Meijster and Joos J.J. Roerdink. A disjoint set algorithm for the watershed transform. In S. Theodoridis, I. Pitas, A. Stouraitis, and N. Kalouptsidis, editors, *SIGNAL PROCESSING IX, Theories and Applications*, volume II, pages 1665–1668. EURASIP, 1998.

[TS92] K. Thulasiraman and M.N.S. Swamy. *Graphs: Theory and Algorithms*. Wiley-Interscience, New York, USA, 1992.