Branch Voxels and Junctions in 3D Skeletons

Gisela Klette

CITR, University of Auckland, Tamaki Campus, Building 731
Auckland, New Zealand

Abstract. Branch indices of points on curves (introduced by Urysohn and Menger) are of basic importance in the mathematical theory of curves, defined in Euclidean space. This paper applies the concept of branch points in the 3D orthogonal grid, motivated by the need to analyze curve-like structures in digital images. These curve-like structures have been derived as 3D skeletons (by means of thinning). This paper discusses approaches of defining branch indices for voxels on 3D skeletons, where the notion of a junction will play a crucial role. We illustrate the potentials of using junctions in 3D image analysis based on a recent project of analyzing the distribution of astrocytes in human brain tissue.

Keywords: 3D skeletons, 3D curve analysis, branch nodes, branch index, thinning, medical image analysis, astrocytes

1 Introduction and Basic Notions

Our theoretical studies on discrete versions of 3D branch indices have been initiated within a recent project about the analysis of confocal microscope images of human brain tissue. Those images are taken layer by layer and constitute a volume, which we assume to be defined within a regular orthogonal grid in 3D space. Figure 1 shows such a volume where 3D rendering has been used.

This 3D view clearly shows some type of "curve-like structures", which can be analyzed after segmentation, skeletonization, and property calculations for voxels on skeletons. Similar curve-like structures appear in other biomedical images such as, for example, in 3D scans of blood vessels, or in 3D ultrasound images.

Long term observations in the School of Medicine at The University of Auckland produced the hypothesis that the number, distribution and "complexity" of astrocytes (i.e., brain cells whose shape resembles that of a star) are related to brain normality or defined types of abnormality (e.g., epilepsy). However, the intuitive concept of "complexity" requires a definition and quantitative studies,

Following discussion with colleagues from the School of Medicine we decided to focus on the number, distribution and complexity of "junctions" of curve-like structures. This paper will provide a definition of such junctions, and discusses consequences of the chosen definition. The definition is an adaptation of basic concepts in 3D curve theory of Euclidean space.

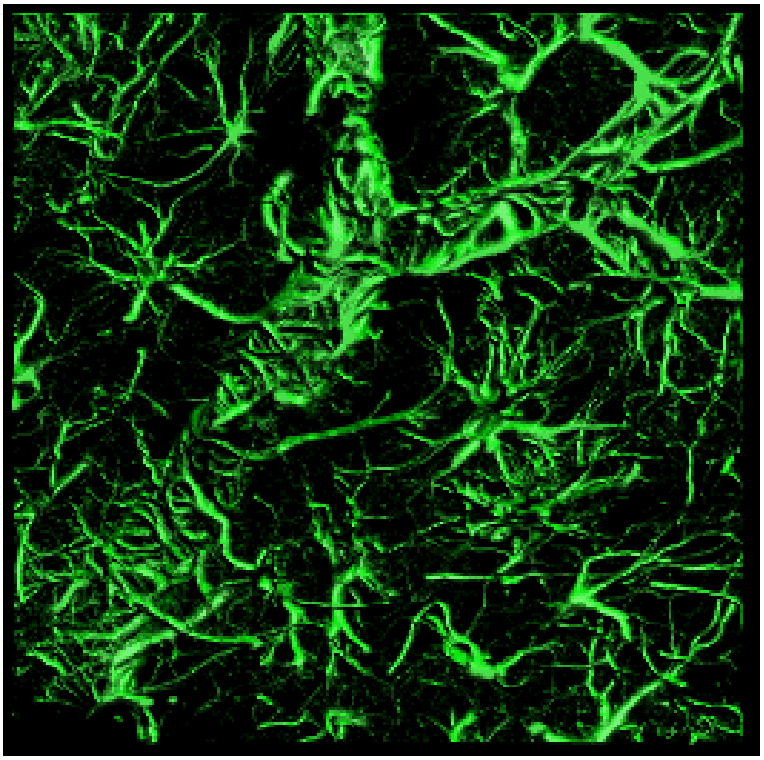

Fig. 1. Example of an input data set of 42 slices of 256×256 density images, all generated by confocal microscopy from a sample of human brain tissue: The astrocytes are partially located around a blood vessel having approximately "Y-shape" (from lower left to upper right) in this sample.

As a first step we point out that 3D topological thinning algorithms (see [10, 4]) deliver skeletons which consist of different "types" of voxels, characterized by branching index. Branching indices will then allow to cluster specific voxels into "junctions". We then map the 3D skeleton into an undirected graph where nodes are defined by junctions and endvoxels. We also propose ways of labeling this graph for supporting the quantitative analysis of curve-like structures. This will be illustrated for the shown sample (in Figure 1) of brain tissue.

We use common adjacency concepts: 4-, 8- (in 2D), 6-, 18-, 26- (in 3D) for the grid point model, and 0-, 1- (in 2D), 0-, 1-, 2- (in 3D) for the grid cell model (see Figure 2 for an illustration in the grid cell model), with notations as in [6]. Any of these adjacency relations A_α, $\alpha \in \{0, 1, 2, 4, 6, 8, 18, 26\}$, are irreflexive and symmetric. The *α-neighborhood* $N_\alpha(p)$ of a pixel or voxel location p includes p and its α-adjacent pixel or voxel locations.

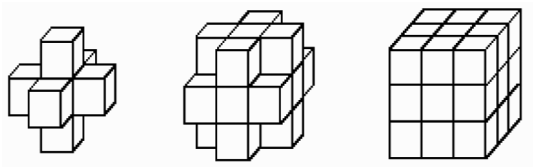

Fig. 2. Neighborhoods (left) $N_2(p)$, (middle) $N_1(p)$ (right), and $N_0(p)$.

Concepts for describing curve points in a continuous space are known for more than 80 years (see [6] for a review). P. Urysohn in 1923 and K. Menger in 1932 proposed (independently) equivalent definitions for simple curves (arcs) based on the notion of the branching index of a point on a curve (arc). The *branching index* of a point on a curve was defined as follows:

Definition 1. *Let p be a point, ε be a positive real, and $U_\varepsilon(p)$ be the ε-neighborhood of p. A curve γ has branching index $m \geq 0$ at $p \in \gamma$ iff (read: if and only if) for any $r > 0$, there is an $\varepsilon < r$ such that the cardinality of the γ-frontier of $U_\varepsilon(p) \cap \gamma$ is at most m.*

Figure 3 shows two examples where q has branching index 4 and p has branching index 2. It is obvious that the branching index of a curve point $p \in \gamma$ in the Euclidean space is the number of crossings of a circle (with radius $\varepsilon < r$ and middle point p) and curve γ. For all circles close enough to p this number has to be constant for defining a branching index.

A *simple curve* in the Euclidean space is a curve γ in which every point $p \in \gamma$ has branching index 2. A *simple arc* is either a curve in which every point p has branching index 2 except for two endpoints, which have branching index 1, or a simple curve with one of its points labeled as an endpoint.

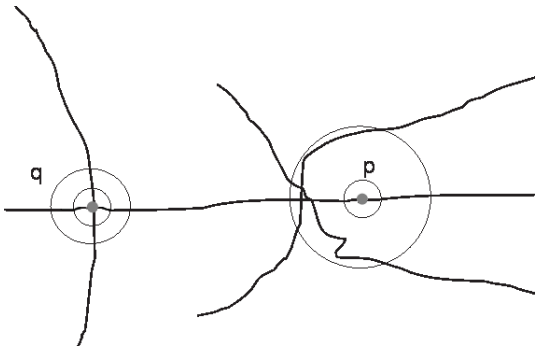

Fig. 3. $U_\varepsilon(q) \cap \gamma = 4$ and $U_\varepsilon(p) \cap \gamma = 2$, assuming that ε is sufficiently small.

An application of those concepts in the 3D digital space (based on 0-adjacency in the cell model) may lead to the following definitions:

- digital curve ρ has *branching index* $m > 0$ at voxel $p \in \rho$ iff exactly m 0-adjacent voxels are elements of ρ;
- voxel $p \in \rho$ is *regular* iff p has branching index 2;
- $p \in \rho$ is a *branch voxel* iff p has a branching index of at least 3;
- $p \in \rho$ is an *endvoxel* iff p has branching index 1;
- $p \in \rho$ is a *singular voxel* iff p is either a branch voxel or an endvoxel;
- the digital curve ρ in 3D space is *simple* iff every voxel in γ is regular; and
- ρ is a *simple arc* iff it is either a curve in which every voxel p is regular except for two endvoxels, or a simple curve where one of its voxels is labeled to be a (double) endvoxel.

This results into limitations of branching indices at voxels which restricts the generality of the concept.

2 Junctions and Abstract Curve Graphs

For ensuring unlimited branching indices we introduce specific clusters of branch voxels. The need for unlimited branching indices occurred when studying 3D skeletons of curve-like structures, produced by a 3D topological thinning algorithm. See, for example, [10, 4] for a discussion of 3D skeletonization. (Those algorithms iteratively delete simple voxels until only non-simple voxels or endvoxels remain.) A 3D curve skeleton ρ is a digital curve, which we consider with respect to 0-adjacency.

Definition 2. *A* 0*-region of branch voxels of a digital curve* ρ *is called a* junction. *The branching index of a junction* J *in* ρ *is the number of regular voxels or end voxels in* ρ *being* 0*-adjacent to any one of the branch voxels in* J.

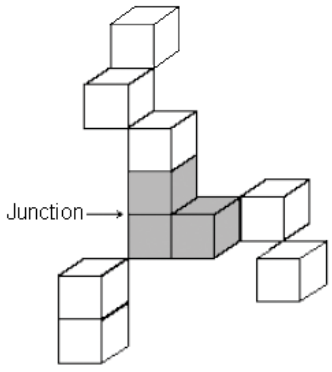

Fig. 4. Example of a junction containing three voxels.

Figure 4 illustrates a junction which consists of three branch voxels. Note that a junction is a non empty 0-connected set of branch voxels. A single branch voxel also represents a junction (with cardinality one).

It follows that a junction has a branching index greater than 2. For example, the branching index of the junction shown in Figure 4 is 3. The complexity of a junction is measured by its branching index.

The following definition is useful for determining the geometric location of a junction. Let J be a junction, n be the number of branch voxels p_i constituting J, with $p_i = (x_i, y_i)$, $1 \leq i \leq n$. The *centroid* $c(J)$ of J is a 3D point with coordinates:

$$x = \frac{\sum_{i=1}^{n} x_{p_i}}{n}, \quad y = \frac{\sum_{i=1}^{n} y_{p_i}}{n}, \quad z = \frac{\sum_{i=1}^{n} z_{p_i}}{n} \tag{1}$$

We identify the geometric position of a junction with that of its centroid.

Definition 3. *A digital curve ρ is mapped into an undirected graph G, where a node of G is either a junction or an endvoxel of ρ. Two nodes in G are connected by an edge iff the corresponding junctions or endvoxels are 0-connected in ρ. G is the abstract curve graph of ρ.*

G is uniquely defined by the chosen adjacency (0-adjacency in our case). The geometric positions of a junction or of an endvoxel define the geometric positions of the nodes of G.

In experiments we assign indices to all nodes of the abstract curve graph. All branch voxels of one junction obtain the same label this way. Edges of the abstract curve graph correspond to digital arcs between junctions or endvoxels of the curve. See Figure 5 for an example where the skeleton has been calculated for a 3D brain tissue scan. Due to using only one label for all branch voxels of one junction, different arcs may start from different branch voxels which all have the

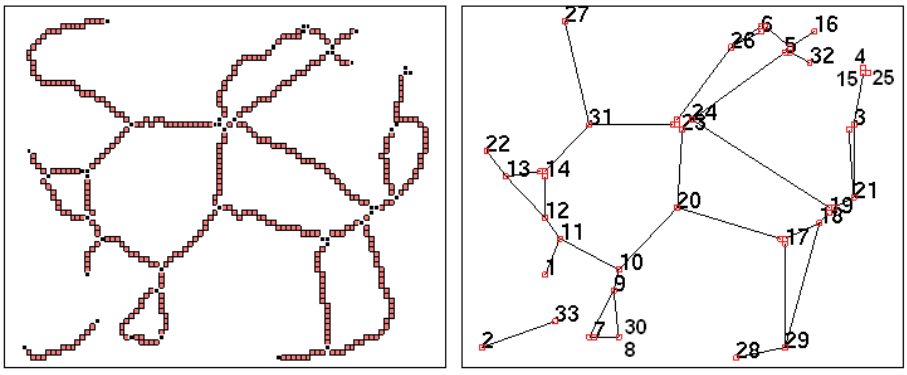

Fig. 5. Left: a skeleton (junctions are shown as black voxels). Right: abstract curve graph for this skeleton (nodes are labeled by indices).

same label. For example, B_{14} in Figure 5 consists of three branch voxels. Each of them is an endvoxel of a digital arc. We use the geometric position (i.e., centroid) of B_{14} as endpoint for all of those three arcs, for example for the calculation of the Euclidean distance to other nodes (i.e., junctions or endvoxels).

A straightforward application of this convention allows the calculation of Euclidean distances between nodes. For a more accurate estimation of distances between nodes we apply length estimation based on connecting digital arcs. First we identify the two endvoxels for each arc. This can be done by a second labeling process where all arcs are uniquely labeled. All voxels in one arc obtain the same label; each branch voxel is mapped (say randomly) to exactly one arc [5]. After these assignments, we apply a (global) DSS-based length measurement (see [6]), where we decided for algorithm **DR1995** as published in [1]. The DSS algorithm cuts an arc into a set of digital straight segments, and the total length is the sum of the lengths of those segments. We have chosenthe way of DSS-based length estimations because (besides a general theoretical benefit of being multigrid convergent to the true length) it also proved to be adequate for characterizing the complexity of distributions of astrocytes.

The distances between pairs of singular voxels can now be used for calculating weights for all edges in the abstract curve graph (using means in case of multiple arcs). Based on the cost matrix for this weighted graph together with the coordinates of all nodes we then applied traditional algorithms from graph theory (such as algorithms for calculating the minimum path between any two nodes, algorithms for determining the total weight of the minimum spanning tree, or algorithms for finding the diameter of the graph and so on) for further analysis of the curve-like structure in the given 3D image.

We determined the *uniformity* of junctions as follows: The volume data are divided into a set of subcubes (small cubes of identical size). For a fixed branching index j, we count the number of junctions in each cube having branching index j. If the number of junctions with branching index j is equal in every subcube then we say that junctions of branching index j are *uniformly distributed* in the whole volume. The deviation from this ideal case characterizes the degree of non-uniformity.

The division into subcubes can be fixed (a segmentation into pairwise disjoint subcubes), or there can be a sliding subcube of varying size. In the examples below we only illustrate the case of a fixed segmentation using pairwise disjoint, uniformly sized cubes of voxels. (Sliding subcubes experiments are not reported in this paper.)

For the description of the *density* of junctions we calculate the shortest path between (unordered) pairs of distinct junctions with the same branching index j. The shorter the path, the more *dense* are the junctions positioned in 3D space. The total number of junctions in a subcube is a (simple) expression of density of junctions in this subcube.

The data set shown in Figure 1 is divided into 36 subcubes, all of size 42^3. (This also generates some excessive data.) See Figure 6 for the resulting curve (3D skeleton). We illustrate the approach by results for this example data set.

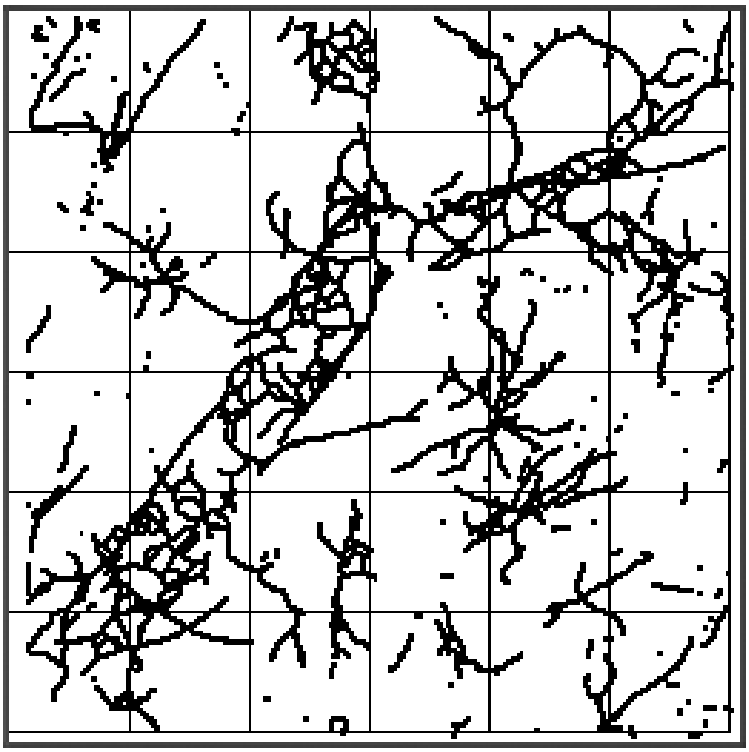

Fig. 6. 3D skeleton of the binarized volume shown in Figure 1.

All identified junctions have branching indeces between 3 and 7. The shaded cubes in Figure 7 correspond to the location of the main blood vessel, and they

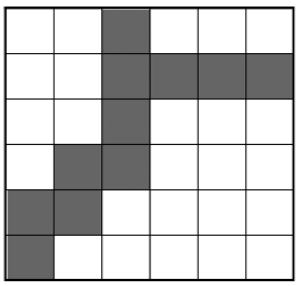

Fig. 7. Location of a main blood vessel (shown as gray cubes) detected by analyzing the 3D skeleton shown in Figure 6.

Branching index j	Junctions in V_1	Junctions in V_2	Ratio between V_1 and V_2
$j = 3$	150	276	54.3%
$j = 4$	53	85	62.4%
$j = 5$	16	21	76.2%
$j = 6$	5	7	71.4%
$j = 7$	2	2	100%
$3 \leq j \leq 7$	226	391	57.8%

Table 1. Number of junctions per branching index in the gray cubes (volume V_1) and in the total volume (volume V_2).

contain in total more than 50% of all junctions, for each branching index between three and seven. Table 1 presents the total number of junctions per branching index for all the gray cubes (volume V_1) and for the whole volume V_2.

We counted the number of junctions of equal types per cube to find out how they are distributed in the volume. Obviously, they are not (ideally) uniformly distributed (see Table 2) in the whole volume. Most of them are located close to the blood vessel.

The cardinality of junctions in this experiment did not exceed four and the maximum branching index did not exceed seven. The original structure of the image (elongated parts) and a range of preprocessing steps (segmentation and noise reduction by a sequence of morphological operations) are reasons for keeping the cardinality and the branching index at low values. Theoretically, this is not always the case.

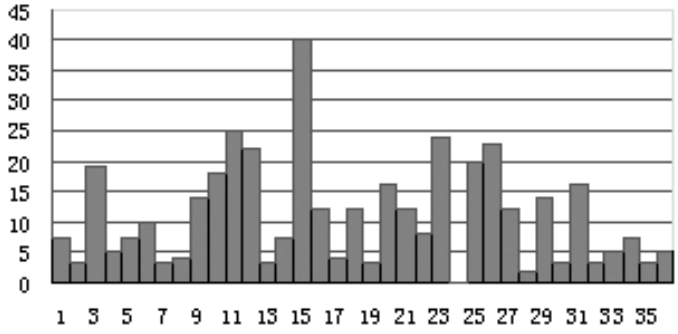

Table 2. Distribution of junctions in subcubes: The horizontal axis represents the numbers of subcubes, and the vertical axis represents the number of all junctions in a subcube.

3 Properties of Junctions in 3D Curves

The branching index in the continuous space is defined for a single point $p \in \gamma$ of a curve γ. With above definitions we merge a set (i.e., a 0-connected region) of branch voxels into a single node in the abstract curve graph G. Interestingly, the size of this region can grow behind any limit.

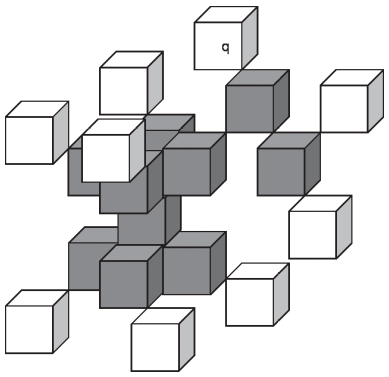

Fig. 8. A junction with cardinality 10 and $m = 9$.

The cardinality of a junction can grow if the image size or the grid resolution grows. All black voxels in Figure 8 are non-regular with a branch index $m \geq 3$, and all white voxels are endvoxels (if the white voxels would be regular then the junction would not change). If endvoxel q (as a grid cube) would share two more vertices with two more voxels, then q would change into a branch voxel; we could continue this process of adding two more voxels to one of the new endvoxels. As a consequence the junction would grow and the branching index could increase behind any limit.

The maximum branching index for a junction with cardinality one is eight; see Figure 9.

We recall the concept of an attachment set to separate branch voxels into two types. The *frontier* of a voxel is the union of its six faces. A face of a voxel includes its 4 edges, and each edge includes its 2 vertices. Let p be an n-cell, $0 \leq n \leq 3$. The frontier of an n-cell p is a union of i-cells with $0 \leq i < n$ (i.e., excluding p itself). For example, if p is a voxel (a 3-cell) then the frontier consists of eight 0-cells, twelve 1-cells and six 2-cells.

Kong [7] defined the I-attachment set of a cell p for the grid cell model as follows, where I is an image:

Definition 4. *Let p and q are grid cells. The I-attachment set of a n-cell p in I is the union of all i-cells, $0 \leq i < n$, on the frontier of p that also lie on frontiers of other grid cells q with $I(p) = I(q), p \neq q$.*

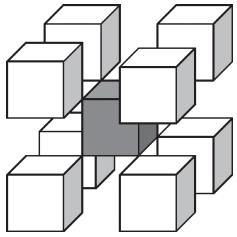

Fig. 9. A junction with cardinality 1 and $m = 8$.

Let m be the number of voxels in $N_0(p) \bigcap S$ and n the number of components in the I-attachment set of p. A branch voxel p is called:

- a *proper branch voxel* if $m = n$,
- a *normal branch voxel* if $m > n$.

A junction is either a 0-region of normal branch voxels, or a proper branch voxel. It follows that a proper branch voxel is a junction of cardinality one. This definition splits the large junction in Figure 8 (for example) into three disjoint junctions. The black voxels (see Figure 10) represent a new junction with cardinality eight and $m = 7$. Voxels p and q are disjoint junctions with $m = 3$ each.

This approach increases the number and the density of junctions and it prevents junctions from growing in a certain direction.

For the application of the DSS algorithm we use the centroid of each junction J as a node in the abstract curve graph G, and each 0-connected regular voxel or end voxel to J is a start (or end) voxel for the length measurement of a digital arc between two nodes. We do not use voxels in junctions as start or end voxel

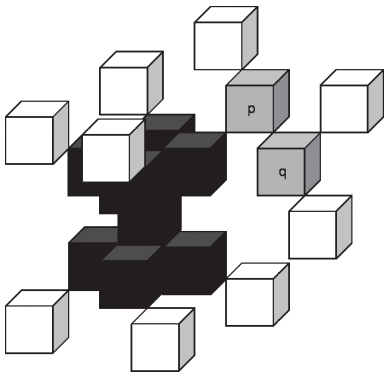

Fig. 10. Three 0-connected junctions, one formed by the dark gray voxels, and two defined by single voxels each (voxels p and q).

for the DSS algorithm. The length between two nodes is the calculated length of the arc between two nodes plus the Euclidean distance from the start voxel of the arc to the centroid of the 0-connected junction (if it is connected to a junction) plus the Euclidean distance from the end voxel of the arc to the centroid of the 0-connected junction (if it is connected to a junction).

4 Conclusions

In this paper we propose a classification of voxels in 3D skeletons for subsequent length measurements of digital arcs. The definition of branch voxels follows curve theory for the Euclidean space. Junctions are defined as 0-connected regions of branch voxels. These junctions and their properties are useful for the analysis of curve-like structures in biomedical images. An adjustment for the definition of junctions is introduced to prevent arbitrary growth and to improve the accuracy of length measurements using the DSS algorithm.

Acknowledgment: The author acknowledges the work of her students, in particular of Mian Pan, who contributed to this work in his Master thesis project.

References

1. I. Debled-Rennesson and J.-P. Reveilles. A linear algorithm for segmentation of digital curves. *Int. J. Pattern Recognition Artificial Intelligence*, **9**:635–662, 1995.
2. G. Klette: A comparative discussion of distance transformations and simple deformations in digital image processing. *Machine Graphics & Vision*, **12**:235–256, 2003.
3. G. Klette: Simple points in 2D and 3D binary images. In Proc. *Computer Analysis Images Patterns*, LNCS 2756, pages 57–64, Springer, Berlin, 2003.
4. G. Klette and M. Pan: 3D topological thinning by identifying non-simple voxels. In Proc. *Int. Workshop Combinatorial Image Analysis*, LNCS 3322, pages 164–175, Springer, Berlin, 2004.
5. G. Klette and M. Pan: Characterization of curve-like structures in 3D medical images. In Proc. *Image Vision Computing New Zealand*, pages 164-175, 2005.
6. R. Klette and A. Rosenfeld. *Digital Geometry – Geometric Methods for Digital Picture Analysis.* Morgan Kaufmann, San Francisco, 2004.
7. T. Y. Kong: On topology preservation in 2-D and 3-D thinning. *Int. J. Pattern Recognition Artificial Intelligence*, **9**:813–844, 1995.
8. K. Palagyi, E. Sorantin, E. Balogh, A. Kuba, C. Halmai, B. Erdohelyi, and K. Hausegger: A sequential 3D thinning algorithm and its medical applications. In Proc. *Information Processing Medical Imaging*, LNCS 2082, pages 409–415, Springer, Berlin, 2001.
9. K. Palagyi and A. Kuba: Directional 3D thinning using 8 subiterations. In Proc. *Discrete Geometry Computational Imaging*, LNCS 1568, pages 325–336, Springer, Berlin, 2003.
10. K. Palagyi and A. Kuba: A 3D 6-subiteration thinning algorithm for extracting medial lines. *Pattern Recognition Letters*, **19**: 613–627, 1998.