RIJKSUNIVERSITEIT GRONINGEN

Topologic, Geometric, or Graph-Theoretic Properties of Skeletal Curves

Proefschrift

ter verkrijging van het doctoraat in de
Wiskunde en Natuurwetenschappen
aan de Rijksuniversiteit Groningen
op gezag van de
Rector Magnificus, dr. F. Zwarts,
in het openbaar te verdedigen op
vrijdag 26 januari 2007
om 14.45 uur

door

Gisela Klette
geboren op 22 oktober 1952
te Halle, Duitsland

Promotor:
Prof. dr. cs. techn. N. Petkov

Beoordelingscommissie:
Prof. dr. W. G. Kropatsch
Prof. dr. J. B. T. M. Roerdink
Prof. dr. M. Vento

ISBN: 90-367-2844-4
ISBN: 90-367-2845-2 (electronic version)

For my family in New Zealand and my parents in Germany.

Abstract

Keywords: analysis of digital curves, simple points, thinning, topology, branch nodes, medical image processing.

Medical imaging is still mainly focusing on visualization and data communication. However, applications of digital image analysis methods in biomedical research are also increasing, due to demands and developments in hardware, image analysis software, and methodology. Shape simplification procedures are already important modules of software packages. The literature offers a large number of papers about algorithms, which aim at deforming images into topologically equivalent images; the latter ones should represent the shape of complex objects in a simplified form. This thesis reviews and extends the diversity of approaches published in this area with respect to properties of algorithms and characterizations of simple points. It contributes in particular with new theoretical results. Topologic thinning methods deliver digital curves (skeletons), which are used to describe objects in digital images. The thesis shows that different applications require different constraints and adjustments of general thinning or curve analysis procedures. It also studies the effect of increased grid resolution (e.g., trying to utilize progress in hardware) for the potential accuracy of measurements based on skeletons. The thesis illustrates results by contributing to one particular application (i.e., analysis of astrocytes in human brain tissue), for example by verifying the efficiency of calculated features for classification.

Acknowledgments

I thank Professor Nicolai Petkov for his support to undertake these studies. Repeated stays at Groningen University, also including my seminar presentations, have been of great value for my research. For discussions following those seminars I am also especially grateful to Professor Jos Roerdink and Dr. Michael Wilkinson. Coming to Groningen has always been a very valuable inspiration.

I thank Professor Colin Green at the Medical School of The University of Auckland for the opportunity to participate in exciting biomedical projects. I supervised students in this context, and their work also contributed to this thesis. My special thank goes to Shuping Cao, Kennant Kin Lun, Ana Stilinovich, and Mian Pan.

I thank my husband for his encouragement to start with this work, and that he gave me the opportunity to take part in discussions on discrete mathematics during many years.

Gisela Klette
Groningen
November 23, 2006

Contents

Chapter 1

Skeletal Curves and Scope of Thesis

This chapter briefly introduces (with references to historic origins) skeletal curves, which are either medial axes or linear skeletons. (Detailed definitions follow in the next chapter.) The chapter informs then about motivation, intentions, and structure of this thesis.

1.1 Skeletal Curves

The *linear skeleton* has been introduced into topology by (Listing 1861), defined by continuous contractions of a set into a one-dimensional (1D) subset, but not by intentions of having "centered lines". On the opposite, the *medial axis* is defined by centers of maximal disks contained in a set; it was informally specified by (Blum 1962).

1. DEFINITION. *A skeletal curve in the continuous space is either a (not uniquely defined) linear skeleton or a (uniquely defined) medial axis, aiming at topologic or geometric studies.*

Curves in general have been defined by P. Urysohn and K. Menger in the 1920*s* and 1930*s* [see (Urysohn 1923) and (Menger 1932)], also introducing *branch* or *end points* of curves. A curve finally defines, at a more abstract level, an undirected graph, whose nodes are identified with branch or end points, and edges symbolize existing arcs. A curve is a 1D (or *linear*) set, and a skeletal curve is an abstract disambiguation of a (biomedical) skeleton (see Figure 1.1).

Both concepts of skeletal curves have been "digitized". Digital medial axes go back on (Rosenfeld and Pfaltz 1966), who defined them based on *distance transforms*. For a given connected set, they are not necessarily connected. Digital linear skeletons are defined by topology-preserving *thinning* [see (Rosenfeld 1970)], which is a repeated removal of "deletable elements"; today those are called *simple pixels* for two-dimensional (2D) pictures, or *simple voxels* for three-dimensional (3D) pictures.

The calculation of skeletal curves proved to be a very useful preprocessing step in image analysis, and it is still an active field of research – now already for nearly

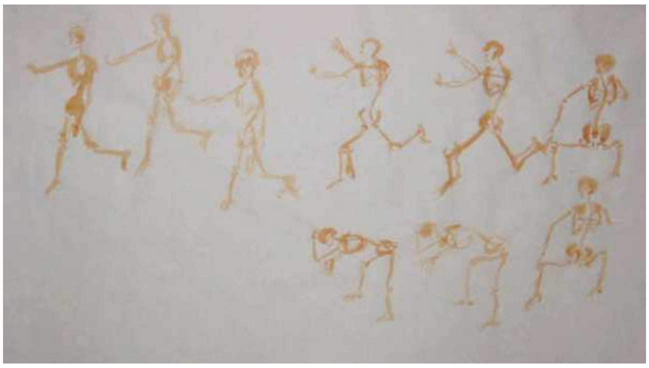


Figure 1.1: *A skeleton in its biomedical meaning is not 1D or linear; it is composed of 3D volume elements (copy of a painting by Kurt Grosjean from www.deviantart.com).*

40 years! A large number of papers has been published on digital medial axes (distance transforms) or digital linear skeletons (thinning, or simple pixels or voxels), see (Klette and Rosenfeld 2004).

Algorithms for calculating skeletal curves are typically also based on heuristic arguments. For example, when calculating a linear skeleton, a "modified thinning" should stop such that a resulting linear "stick figure represents all body parts" (see Figure 1.2), and will not continue with removing "arms or legs". Thinning is the operation which implements topologic contraction.

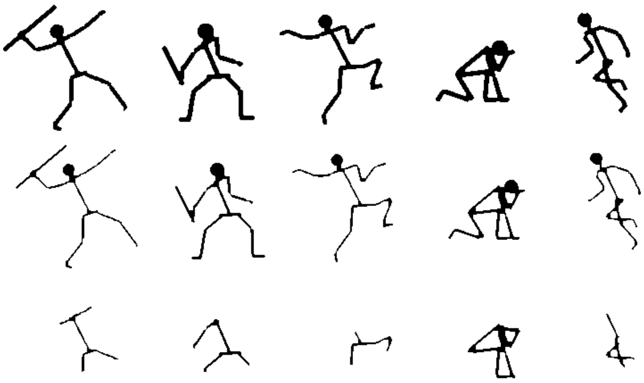


Figure 1.2: *The upper row is contracted into the middle row, and then further into the bottom row. All three "stick figures" on the left will finally each contract into a single pixel, defining a linear skeleton. Both stick figures on the right contract into more complex linear skeletons.*

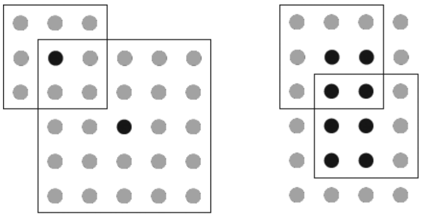


Figure 1.3: *Left: two digital squares representing maximal disks in the Minkowski metric L_1 define a disconnected medial axis (both black nodes). Right: all black nodes are centers of maximal disks (two disks are shown), defined by the same metric.*

Applied heuristics for the calculation of medial axes aim on preserving connectedness (on the expense of allowing redundancy), and on ensuring results which are 1D curves (i.e., which do not contain parallel digital line segments; see Figure 1.3).

Algorithms for calculating skeletal curves can be characterized as being *ill-posed*: Minor differences between algorithms or input data can deliver totally different results, also with respect to run-time complexity, or properties derived from skeletal curves.

Skeletal curves are very useful tools for understanding the shape or structure of 2D or 3D objects. The thesis discusses briefly the generation of such skeletal curves, and focuses then on their analysis by "digitizing" concepts of the theory of curves in Euclidean spaces (as established by Menger and Urysohn).

1.2 Motivation and Scope of Thesis

Modern imaging techniques produce digital images of high resolution, and support for a (partially) automated exploitation of those huge amounts of data requires ongoing research for improvements of applied algorithms. Shape analysis is one of the important issues. In 2D image analysis, the extraction of features from skeletons is used in the context of pattern recognition, for example for character recognition, finger print recognition, prenatal diagnosis, or biological cell studies. Skeletonization methods in two dimensions and their applications have been extensively studied, and ongoing work still appears frequently in image processing literature. Approaches in three or higher dimensions are an order of magnitude more difficult to formalize or to describe, and implementations are becoming very complex. Skeletonization results can be curves (i.e., linear skeletons), but also (say, as an intermediate result, or for a "better match" with given shapes) "thin" or "elongated" volume parts.

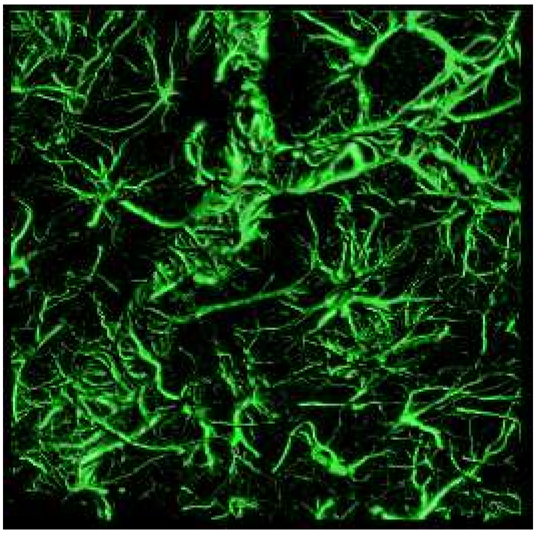


Figure 1.4: *A volume scan composed of 42 slices of* 256×256 *density images. This 3D view was produced using commercial software.*

This thesis was initiated in 2002 by one particular research project at the Medical School of The University of Auckland (but developed then on its own). Figure 1.4

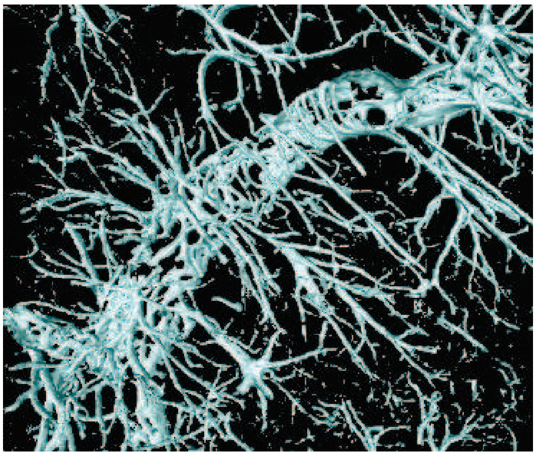


Figure 1.5: *Isosurface generated by a marching cubes algorithm.*

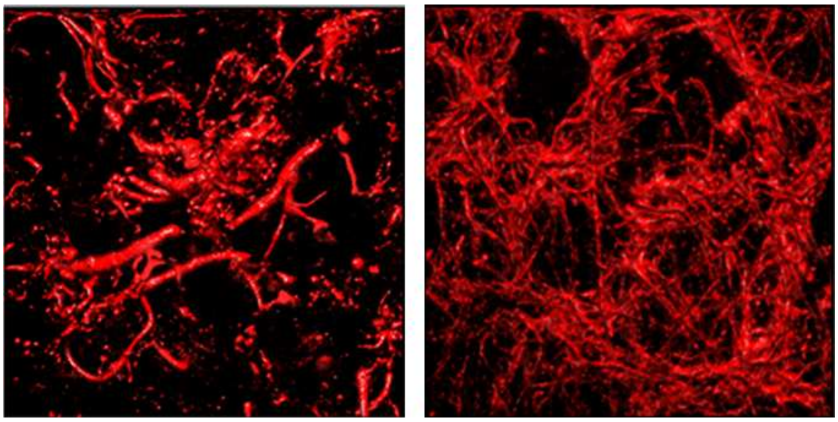


Figure 1.6: *Left: Normal control human hippocampus CA1 area, right: MTLE-damaged human hippocampus CA1 area*

shows a 3D (volume) sample of human brain tissue. The volume is composed of slices, being confocal microscope images, and it shows astrocytes (star-shaped cells of the nervous system which provide nutrients, support and insulation for neurons). Medical experts developed the hypothesis that features (such as the number, "distribution" or "structural complexity") of astrocytes in such a volume scan are sufficient (or, at least essential) for defining states between normal or abnormal tissue.

Figure 1.5 shows the isosurface of segmented astrocytes as calculated (by one co-supervised student) at an early stage of this project. Figures 1.4 and 1.5 provide a good insight into the topologic and geometric complexity of these volume scans. Astrocytes enclose blood vessels; Figure 1.4 contains a Y-shaped blood vessel (from lower left corner to upper right), and Figure 1.5 shows a diagonally intersecting, slightly curved blood vessel.

Medical experts are able to evaluate the state of neurological diseases of patients just by looking at such images. However, these evaluations are based on experience, and they are influenced by subjective judgments. It proved to be difficult to describe explicitly those observed shapes or structures that differentiate normal tissue from tissue of people with neurological diseases. Figure 1.6 shows normal control tissue on the left and (mesial temporal lobe epilepsy) MTLE-damaged tissue on the right. The structure in the picture on the right can be described as "more dense" or "more complex" compared to the picture on the left. We need to define those descriptions. Image analysis appears as a suitable way to deliver objective features, which might be useful for those evaluations.

This PhD project was initiated by the understanding that topologic, geometric, or graph-theoretic properties for such volume scans are needed for further progress in those studies. The medical project itself still requires further preparation (e.g., each volume scan as illustrated by Figures 1.4 and 1.5 requires weeks of preparation and expensive scanning, and obtained volume scans are often still unsatisfactory for subsequent automated analysis).

This thesis is about the mathematics and algorithmic challenges related to the calculation of properties of skeletal curves. It discusses problems related to skeletal curves on a much more general level than just in the context of one particular application.

Curve-like structures appear frequently in 3D biomedical image analysis (e.g., analysis of blood vessels, of neurons, or in ultrasound medical imaging), material sciences (e.g., analysis of porous media where cavities define curve-like structures to be studied), or different disciplines in physics (e.g., bubble chamber data). Studies of moving 3D objects (e.g., of beating hearts) also lead to models defined in four-dimensional (4D) spaces. Besides our main interest in 2D or 3D applications, the thesis often formulates in a general n-dimensional (nD) way, with $n \geq 2$. At least, this unifies 2D and 3D.

The thesis is restricted to studies of linear skeletons, and we do not consider skeletons whose components are also surfaces or volumes.

Furthermore, the thesis prefers the term *picture* rather than *image*, because considered data in multidimensional grids may have resulted from other processes than just imaging (e.g., measurements or calculations). The common representation of considered pictures P are nD arrays, whose elements $p, q, r, \ldots$ are called *pixels* for $n = 2$, *voxels* for $n = 3$, and (in general) *picture elements* for $n \geq 2$.

We only consider pictures defined on a regular orthogonal grid (and not, for example, a triangular or hexagonal grid for $n = 2$). Picture elements are labeled, and labels are denoted by $P(p), P(q), P(r), \ldots$. We assume only scalar discrete labels in the sense of measured density. In particular, we use (after picture segmentation) *binary pictures* with $P(p) \in \{0, 1\}$. $P(p) = 1$ indicates an *object element* p, and $P(p) = 0$ a *non-object element* p. (More basic definitions follow in Chapter 2.)

In this thesis we assume that binary pictures define the input. We do not discuss picture segmentation algorithms for the creation of binary pictures, and refer to related text books such as (Haralick and Shapiro 1992, Sonka et al. 1999).

1.3 Outline of this Thesis

Chapter 2 reviews basic concepts and definitions which are then used throughout the thesis. Chapter 3 reports about mathematical models for linear skeletons which have been proposed for continuous spaces. This concludes Part 1 of the thesis, which is on fundamentals.

Two classes of algorithms are considered for the generation of curve-like structures. *Distance transforms* are fundamental tools for generating *distance maps*. Those allow to calculate *digital medial axes* or *distance skeletons*, and are suitable for elongated objects in input pictures P. The choice of the metric influences the result and the computational complexity of the algorithms. The review of existing algorithms for distance transforms in Chapter 4 creates a grouping. Each group of algorithms represents a different strategy for the computation of distance maps. Algorithms of the same group differ by applied metrics or by applied picture models[1]. The Chapter provides a short but concise description of a linear-time Euclidean distance transform which calculates exact Euclidean distance maps. A new proposition provides a justification for the selection of elements of the medial axis.

The second class of algorithms for the generation of skeletal curves are iterative thinning processes, which are characterized by removals of *simple elements*. The removal of a simple element does not change the topology of a given picture. Iterative removals continue until the result is 1D (i.e., a set of digital arcs or curves). The application of different forms of characterizing simple elements can change the time complexity of a thinning algorithm. We prove several new theorems which emphasize that a number of existing characterizations of simple elements are actually equivalent. We show that the new characterization of non-simple elements is of benefit. The literature offers a large number of thinning algorithms. Chapter 5 explains theoretical fundamentals of such approaches. It reviews four algorithms which have been originally developed for 2D pictures and it describes the resulting curves. Each of those algorithms represents a prototype which has been further developed for 2D pictures and for higher dimensional pictures in many publications. We combine extended versions of two prototypes with the new characterization of non-simple elements. This concludes Part 2 of the thesis, which is on both basic strategies (i.e., either geometric distance transforms or topologic thinning).

Chapter 6 classifies all elements (i.e., pixels or voxels in 2D or 3D) of skeletal curves based on *branching indices*. It reviews the definitions of a branching index and a *branching point* for the continuous space, and it introduces analogous concepts in digital space. One option is based on the use of the adjacency set (or of the smallest non-trivial neighborhood) of a curve element. A second option uses topologic properties of components which are generated in a larger (global) neighborhood of a curve element. This chapter provides a classification of elements in skeletal curves which is used for a new method to generate abstract curve graphs for subsequent

---

[1] Some authors use non-square or non-cubic grid models such as for example hexagonal grids or elongated voxel grids (Sintorn and Borgefors 2004).

property calculation.

The analysis of digital curves or arcs is a subject of research since (Freeman 1961). Chapter 7 proposes and discusses a diversity of properties for describing skeletal curves. It also contains examples illustrating the use of the proposed properties for volume scans of astrocytes. It proposes features of connected components by computing properties of the approximated skeleton adapted from graph theory. This concludes Part 3 of the thesis, which is on curve segmentation at branch elements and subsequent property measurements.

Chapter 8 provides our conclusions and offers ideas for further developments.

1.4 Summary

The chapter gave a brief introduction into three alternative points of view (i.e., topologic, geometric, or graph-theoretic) when defining or studying digital approximations of skeletal curves. Major statements are as follows: the thesis

- is about methodological aspects rather than on an application;
- it recalls and groups existing algorithms for the approximative and exact computation of the Euclidean distance transform;
- it provides a new algorithm for thinning which is based on characterizing non-simple elements;
- the objects of interest are digital approximations of skeletal curves in (digital) nD pictures ($n \geq 2$), with a focus on $n = 2$ and $n = 3$;
- a digital concept of branch nodes is introduced for resulting digital curves;
- the generation of abstract curve graphs is explained to adapt properties known from graph theory;
- the thesis discusses concepts to analyze digital approximations of skeletal curves with the special aim to support various ways of quantitative evaluations;
- it informs about conclusions derived from theoretical studies and experiments.

Chapter 2

Basic Notions and Definitions

This chapter recalls briefly that part of digital geometry which is relevant for this thesis. We review notations, concepts and definitions which are required for the following chapters. Some of them we discuss more detailed.

2.1 Carrier of Pictures

We follow the introduction of basic definitions in (Rosenfeld and Klette 2002). We assume an nD picture P which is composed of equally sized nD cubes, where edges have length 1 and centers have integer coordinates.

2. DEFINITION. *A* digital picture *P is a function defined on a discrete set $\mathbb{C}$, which is a subset of the nD regular orthogonal grid, with $n \geq 2$. $\mathbb{C}$ is called the* carrier *of picture P.*

Note that the carrier is specifying the general term of the domain of a function for discrete pictures. The elements of $\mathbb{C}$ are either nD grid points in $\mathbb{Z}^n$, or mD grid cells (called *m-cells*), for $m \leq n$, whose vertices are half-integers. The first option defines the *grid point model*, and the second the *grid cell model*. Note that we have 0-, 1-, ..., and n-cells in the nD case, defining an underlying digital topology [equivalently modeled by either the digital topology in (Khalimsky 1986), abstract cell complexes in (Kovalevsky 1989), or incident pseudographs in (Klette and Rosenfeld 2004)]. Each of those two models will offer specific ways for discussing skeletal curves, and we will use both in this thesis.

The range of a (scalar) picture is $\{0, ...G_{max}\}$, with $G_{max} \geq 1$. The range of a binary picture is $\{0, 1\}$, where 0 is identified with "white" and 1 with "black". We call the black elements p of a binary picture *object elements*, and the white elements p *non-object elements*. For brevity, we call them sometimes also just 1's and 0's.

We are interested in shape simplification of (black) objects in binary pictures. $\langle P \rangle$ is the set of all object elements $p \in \mathbb{C}$ (i.e., with $P(p) = 1$), and $\langle \overline{P} \rangle$ is the set of all (white) non-object elements (i.e., $P(p) = 0$). Note that the complement of $\langle P \rangle$ is taken with respect to the infinite digital space $\mathbb{Z}^n$, that means, all elements outside of the carrier $\mathbb{C}$ are assumed to be non-object.

We use the term *frontier* for identifying the difference set between the topologic closure of an open set and the set itself. For example, the frontier of a 2-cell is the union of its four edges, also containing its four vertices, and the frontier of a 3-cell is the union of its six faces, also containing its twelve edges and eight vertices.

Elements of a 2D (3D) picture are pixels (voxels), and, accordingly, we have *object pixels*, *object voxels*, and so forth. Using the grid cell model, a pixel p is a square (2-cell) in the Euclidean plane, possibly[1] also containing some of the 0- or 1- cells of its frontier; a voxel is a cube (3-cell) in the Euclidean space, possibly also containing some of the 0-, 1-, or 2- cells of its frontier, where edges are of length 1 and parallel to the coordinate axes, and centers have integer coordinates. Using the grid point model, a 2D or 3D pixel or voxel location is a point in $\mathbb{Z}^2$ or $\mathbb{Z}^3$, respectively.

Let $m \leq n$. Two picture elements (n-cells) p and q in the grid cell model are called *m-adjacent* iff (read: if and only if) $p \neq q$ and their frontiers (closures) share at least one m-cell. Figure 2.1 shows three types of neighborhoods in 3D.

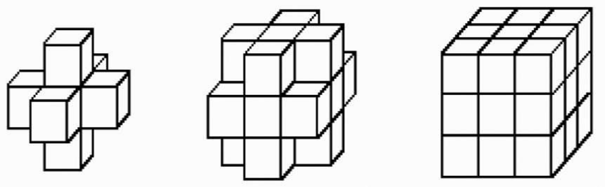


Figure 2.1: *Neighborhoods (left)* $N_2(\mathrm{p})$, *(middle)* $N_1(\mathrm{p})$ *(right), and* $N_0(\mathrm{p})$.

Adjacencies in the grid point model are named by cardinalities. 0-adjacency in the grid cell model is dual to 8-adjacency in 2D, or 26-adjacency in 3D, if the grid point model is used. 1-adjacency in the grid cell model is dual to 4-adjacency in 2D, or 18-adjacency in 3D, if the grid point model is used. Finally, two voxel p and q in the grid cell model are 2-adjacent iff both are 6-adjacent in the grid point model. Altogether, we have *adjacency relations* A_α, $\alpha \in \{0, 1, 2, 4, 6, 8, 18, 26\}$, for $n = 2$ or $n = 3$, which are irreflexive and symmetric on a picture carrier $\mathbb{C}$.

The *α-neighborhood* $N_\alpha(p)$ of a picture element p includes p and its α-adjacent picture elements. The *adjacency set* of p is the set of all α-adjacent picture elements. Let p_i be the i-th coordinate of p. Assume an nD picture P and the grid point model. We have the following

---

[1] Exactly defined by the underlying digital topology; see the assumed good-pair adjacency further below.

1. COROLLARY. *The* $(3^n - 1)$*-neighborhood of a picture element* p *equals*

$$N_{3^n-1}(p) = \{q \in \mathbb{Z}^n : \max_{1 \le i \le n} |p_i - q_i| \le 1\} \tag{2.1}$$

and its $2n$*-neighborhood equals*

$$N_{2n}(p) = \{q \in \mathbb{Z}^n : \sum_{i=1}^{n} |p_i - q_i| \le 1\} \tag{2.2}$$

Cardinalities of all other neighborhoods are between 3^n and $2n + 1$.

The transitive and reflexive closure of an adjacency relation defines connectedness. Two elements $p, q \in \mathbb{C}$ are α*-connected* with respect to $M \subseteq \mathbb{C}$ iff there is a sequence of elements $p = p_0, p_1, p_2, ..., p_n = q$ such that p_i is α-adjacent to p_{i-1}, for $1 \le i \le n$, and all elements of this sequence are either in M, or all in the complement $\overline{M}$ of M. A subset $M \subseteq \mathbb{C}$ of a picture carrier is called α*-connected* iff M is not empty and all points in M are pairwise α-connected with respect to set M itself. An α*-component* of a subset S of $\mathbb{C}$ is a maximal α-connected subset of S. The study of connectivity in digital pictures has been introduced in (Rosenfeld and Pfaltz 1966).

The infinite component of 0's is called the *background*. A *region* is a finite component. There are object regions of 1's, non-object regions of 0's (*holes* for $n = 2$ and *cavities* for $n = 3$), and the background. This partitions the digital space $\mathbb{Z}^n$.

As common, to ensure a digital topologic space[2] we use good pairs $(3^n - 1, 2n)$, that means, $(3^n - 1)$-adjacency for black elements $p \in \langle P \rangle$, and $2n$-adjacency for white elements $p \in \langle \overline{P} \rangle$. We state an obvious conclusion:

2. COROLLARY. *Object components are closed sets in the underlying digital topology.*

For $n = 2$, the resulting region adjacency graph is a tree (Rosenfeld 1974). The *border* of a set M of 1's is the set of elements of M that are $2n$-adjacent to $\overline{M}$; in case of 0's, the border is defined by $(3^n - 1)$-adjacency. See Figure 2.2 for an illustration for border versus frontier, and also for the following definition.

3. DEFINITION. *The cubic adjacency set* $A_c(p)$ *of an object picture element* p *is the union of all closed object* n*-cells which are* 0*-adjacent to* p*. The cubic adjacency set of a non-object picture element* p *is the union of all closed non-object* n*-cells which are* $(n-1)$*-adjacent to* p*.*

Note that the closure of an n-cell is defined in the underlying digital topology. (A set is closed iff, for every cell p in the set, the set also contains all the lower-dimensional cells incident with p.)

---

[2] $3^n - 1$-adjacency defines closed sets, and $2n$-adjacency defines open sets.

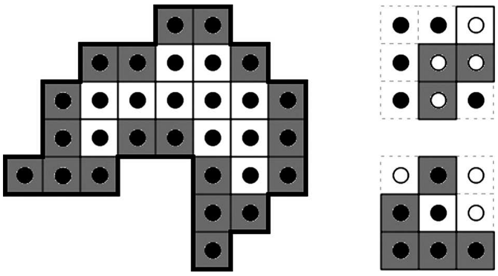


Figure 2.2: *Left: the border of this object region are all shaded pixels; its frontier is shown as a bold line. Right: shaded cubic adjacency sets of a non-object pixel (top) and an object pixel (bottom).*

Let U, V, W be pairwise disjoint sets of picture elements. We say that V α-separates U from W iff any α-path from an element in U to an element in W must intersect V (i.e., must contain at least one element of V).

Consider an operation on a picture which calculates a property for elements p in P. This is a *local operation* iff its results, at any element p, only depend on values in P within a neighborhood of *p of fixed size* (i.e., this size does not depend on the size of the picture, on decisions within the calculation, and so forth).

2.2 Attachment Sets and Euler Characteristic

Attachment sets defined in the cell model are useful for characterizations of simple elements. (Kong 1995) introduced the *P-attachment set* of a cell p for the grid cell model, and we present it slightly modified for our purposes as follows:

4. DEFINITION. *The P-attachment set K_p of an n-cell p in picture P is the union of all m-cells (with $0 \leq m < n$) in the frontier of p that also lie on the frontier of an n-cell $q \neq p$ with $P(p) = P(q)$.*

Note that the cardinality of the P-attachment set equals 1 if it only contains a single 0-cell; otherwise the cardinality of a non-empty P-attachment set is that of the real numbers. Figure 2.3 shows an example for a 3D P-attachment set. To represent a P-attachment set of a voxel we use *Schlegel diagrams* as proposed in (Kong 1995). Such a diagram maps the frontier of a voxel into the plane, where one of the 2-cells becomes the unbounded exterior face of the shown diagram.

The Euler characteristic $\chi(K)$ is defined in combinatorial topology (Aleksandrov 1956) as the alternating sum of α_ms which are the cardinalities of m-dimensional elements in a given complex K. In case of a P-attachment set we have [see (Gau and Kong 2002) for $n = 3$]:

5. DEFINITION. *The* Euler characteristic $\chi(K_p)$ *of the P-attachment set K_p of a picture element p is equal to the number of 0-cells (in K_p) minus the number of 1-cells (in K_p) plus the number of 2-cells (in K_p) ... (plus or minus) the number of $n - 1$-cells (in K_p).*

The P-attachment set in Figure 2.3 has six 0-cells, five 1-cells and one 2-cell; accordingly its Euler characteristic χ equals $\alpha_0 - \alpha_1 + \alpha_2 = 6 - 5 + 1 = 2$. A basic result in combinatorial topology is that the Euler characteristic equals the alternating sum of the Betti numbers. Here we have $\chi = \beta_0 - \beta_1 = 2 - 0$, because there are two components, but no hole.

The Euler characteristic of a P-attachment set is easy to compute and can simplify the identification of simple elements in thinning algorithms. Figure 2.4 provides another example. Here we have $\chi(K(p)) = \alpha_0 - \alpha_1 + \alpha_2 = 6 - 5 + 0 = \beta_0 - \beta_1 = 2 - 1 = 1$.

2.3 Characteristic Numbers in the Grid Point Model

This section defines characteristic numbers which have been used in thinning algorithms. The main purpose for their introduction was to find criteria for deleting elements based on (local) neighborhood conditions. For 2D pictures the following numbers are easy to compute.

$A_8(p)$ is the 8-adjacency set of p. Elements q_i of this adjacency set are indexed counter-clockwise as follows:

$$\begin{array}{ccc} q_4 & q_3 & q_2 \\ q_5 & p & q_1 \\ q_6 & q_7 & q_8 \end{array}$$

6. DEFINITION. (Rutovitz 1966) *The number of transitions from a 0 to a 1, or vice versa, when the pixels of $A_8(p)$ are traversed in counterclockwise order in a picture P, is called*

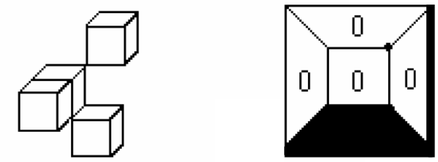


Figure 2.3: *Left: we consider the voxel p located at the middle of this drawing; all the shown voxels have identical P-values. Right: the P-attachment set of p in form of a Schlegel diagram.*

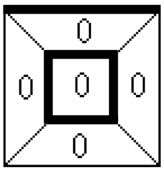


Figure 2.4: *An attachment set which has two components and one hole.*

R-crossing number $X_R(p)$ *and defined as follows:*

$$X_R(p) = \sum_{i=1}^{8} |P(q_{i+1}) - P(q_i)| \quad \textit{where } q_9 = q_1$$

Let $X_A(p)$ be the number of distinct 4-components of 1′s in $A_8(p)$. Then $X_A(p) = X_R(p)/2$ if at least one element in $A_8(p)$ is a 0.

7. DEFINITION. (Hilditch 1969) *The number of transitions from a 0 to a 1 when the pixels in $A_8(p)$ are traversed in order in a picture P, cutting the corner between 8-adjacent 1′s, is called* H-crossing number $X_H(p)$*:*

$$X_H(p) = \sum_{i=1}^{4} b_i$$

where

$$b_i = \begin{cases} 1 & \textit{if } P(q_{2i-1}) = 0 \textit{ and } (P(q_{2i}) = 1 \textit{ or } P(q_{2i+1}) = 1) \\ 0 & \textit{otherwise}\,. \end{cases}$$

The H-crossing number is equivalent to the number of distinct 8-components of 1′s in $A_8(p)$ in case there is at least one 0 in $A_4(p)$, and the H-crossing number is always equal to the number of distinct 4-adjacent 4-components of 0′s in $A_8(p)$.

8. DEFINITION. (Yokoi et al. 1975) *The number of distinct 4-adjacent 4-components of 1′s (0′s) is called* connectivity number $X_Y(p)$ *($\overline{X_Y}(p)$) with:*

$$X_Y(p) = \sum_{i=1}^{4} a_i \quad \textit{and} \quad \overline{X_Y}(p) = \sum_{i=1}^{4} b_i$$

where, in $A_8(p)$ and in picture P,

$$a_i = P(q_{2i-1}) - P(q_{2i-1}) \cdot P(q_{2i}) \cdot P(q_{2i+1}) \;\; \textit{with} \;\; q_9 = q_1$$

and

$$b_i = \overline{P}(q_{2i-1}) - \overline{P}(q_{2i-1}) \cdot \overline{P}(q_{2i}) \cdot \overline{P}(q_{2i+1}) \;\; \textit{with} \;\; \overline{P}(q) = 1 - P(q)$$

Note that the definition of b_i for $X_H(p)$ and for $X_Y(p)$ coincide. For the case of 3D pictures in the grid point model, we will also make use of the following:

9. DEFINITION. (Bertrand and Malandain 1994) *Let* $p \in M \subset \mathbb{Z}^3$. *The* geodesic neighborhood $G_6(p, M)$ *or* $G_{26}(p, M)$ *of voxel p is defined as follows:*

1. $G_6(p, M) = (A_6(p) \cap M) \cup \{q \in (A_{18}(p) \cap M) \mid q$ *is 6-adjacent to a voxel in* $(A_6(p) \cap M)\}$
2. $G_{26}(p, M) = A_{26}(p) \cap M$

The topologic number *associated to p and* M*, denoted by* $T_\alpha(p, M)$ *for* $(\alpha, \alpha') \in \{(6, 26), (26, 6)\}^3$, *is defined as the number of* α*-connected components of* $G_\alpha(p, M)$*.*

The calculation of the topologic number for 3D pictures is not straightforward. However, we use it for the characterization of 3D simple elements, and we will explain a 3D thinning method based on this definition.

2.4 Digital Fundamental Group

Two regions are *topologically equivalent* iff the unions of their cells are homeomorphic in the Euclidean topology. It follows, for example, that an object region can only be topologically equivalent to another object region (because the unions of their cells are closed sets in the Euclidean topology).

Following (Klette and Rosenfeld 2004), two nD pictures are *topologically equivalent* iff they are isotopic in the underlying digital topology. It follows that two pictures are topologically equivalent iff their rooted region adjacency trees are isomorphic, where the background defines the root for both pictures.

The *fundamental group* is defined in algebraic topology. There are different ways to introduce the fundamental group into digital picture processing [see, for example, (Kong 1989)]; in our context it is used as a tool to deal with topologically equivalent pictures, or with simple elements. A 3D thinning algorithm must preserve the digital fundamental groups of a given binary picture, for all components of $\langle P \rangle$ or $\langle \overline{P} \rangle$. This ensures isotopy. We introduce the basic concept of a digital fundamental group for the digital space. For the introduction of the α-homotopy relation we start with a common definition: An *α-path* π of length l, from p to q in $M \subset \mathbb{Z}^n$, is a sequence of elements $(p_i)_{i=0,\ldots,l}$ such that for $0 \leq i < l$ the element p_i is α-adjacent to p_{i+1}, with $p_0 = p$ and $p_l = q$.

---

[3] Note that (α, α') stands for a good pair adjacency: M is α-connected and the complement $\overline{M}$ is α'-connected.

The path π is a *closed path* if $p = q$, and it is called a *simple path* if $p_i \neq p_j$ when $i \neq j$ (except for p_0 and p_l if the path is closed). Elements p_0 and p_l are called *end elements* of π. Given a path $\pi = (p_k)_{k=0,...,l}$, we denote by π^{-1} the sequence $(p'_k)_{k=0,...,l}$ such that $p'_k = p_{l-k}$ for $k \in 0, ..., l$.

Following (Fourey and Malgouyres 2003), let $\pi = (p_i)_{i=0,...,l}$ and $\pi' = (p'_k)_{k=0,...,l'}$ be two α-paths and $p_l = p'_0$. We denote by $\pi \bigoplus \pi'$ the path $p_0, ..., p_{i-1}, p'_0, ..., p'_{l'}$ which is the concatenation of the given two paths.

For $n = 3$, two closed α-paths π and π' in $M \subset \mathbb{Z}^3$ with the same end elements are *"almost identical"* in M if they have identical parts such that $\pi = \pi_1 \bigoplus \gamma \bigoplus \pi_2$ and $\pi' = \pi_1 \bigoplus \gamma' \bigoplus \pi_2$. The α-paths γ and γ' have the same end elements, and they are included in a $2 \times 2 \times 2$ cube (a 2×2 square for $n = 2$).

10. DEFINITION. *(Fourey and Malgouyres 2003) Two α-paths $\pi = (p_i)_{i=0,...,l}$ and $\pi' = (p'_k)_{k=0,...,l'}$ are* α-homotopic *with fixed end elements in $M \subset \mathbb{Z}^3$ if there exists a finite sequence of α-paths $\pi = \pi_0, ..., \pi_n = \pi'$ such that, for $i = 0, ...n-1$, the α-path π_i and π_{i+1} are almost identical with fixed end elements (abbreviated by $\pi \simeq_\alpha \pi'$) in M.*

In Figure 2.5 the two 8-paths π_1 and π_2 are 8-homotopic with fixed end elements p and q in M, the white areas inside the grey squares are non-object regions (holes in M). The path π_3 is not 8-homotopic to π_1 and it is not 8-homotopic to π_2.

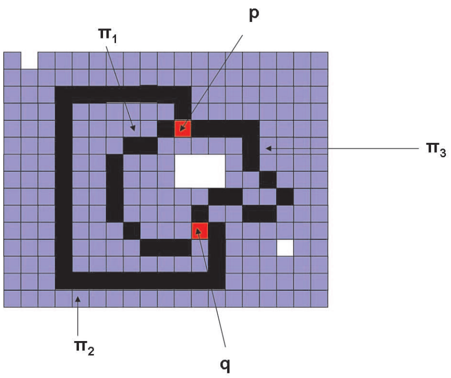


Figure 2.5: π_1 *and* π_2 *are 8-homotopic*

Let b be a fixed element of M that we call the *base element*, and let $A^\alpha_b(M)$ be the set of all closed α-paths $\pi = (p_i)_{i=0,...,l}$ which are included in M and $b = p_0 = p_l$. The α-homotopy relation is an equivalence relation on $A^\alpha_b(M)$. We use the notation $E^\alpha(M, b)$ for the set of equivalence classes of this relation. If $\pi \in A^\alpha_b(M)$ then $[\pi]_{E^\alpha(M,b)}$ is the equivalence class of π.

The concatenation of closed α-paths defines an operation on the set of equivalence classes $E^{\alpha}(M, b)$ which associates to the class of π_1 and the class of π_2 the class of $\pi_1 \bigoplus \pi_2$. With this associative operation we define a group for the set of all equivalence classes.

11. DEFINITION. *The* α-fundamental group *of M with base element b is the group of all equivalence classes of closed α-paths in M with base element b under the concatenation.*

Let us consider p and q as base elements in Figure 2.5. p and q are connected by π_1. It follows that the α-fundamental group of M with base element p and the α-fundamental group of M with base element q are isomorphic.

Let $N \subset M$ and $M \subset \mathbb{Z}^3$ and let $b \in N$ be a base element. A closed α-path in N is also a closed α-path in M. If two closed α-paths in N are homotopic then they are also homotopic in M. The *inclusion map* $i : N \to M$ generates an group isomorphism $i_* : E^{\alpha}(N, b) \to E^{\alpha}(M, b)$. It associates the class of a closed α-path $\pi_1 \in A^{\alpha}_{b}(N)$ in $E^{\alpha}(N, b)$ to the class of the same α-path in $E^{\alpha}(M, b)$. We will use this map in Chapter 5.

2.5 Metrics for Digital Pictures

In general, measurements require a metric space. We recall the well known Minkowski metrics for the nD Euclidean space $\mathbb{R}^n$:

$$L_m(p, q) = \sqrt[m]{\mid x_1 - y_1 \mid^m + ... + \mid x_n - y_n \mid^m} \;\text{ for }\; m \geq 1 \tag{2.3}$$

and

$$L_\infty(p, q) = max\{\mid x_1 - y_1 \mid, ... \mid x_n - y_n \mid\} \tag{2.4}$$

for two points $p = (x_1, x_2, ..., x_n)$ and $q = (y_1, y_2, ...y_n)$.

Picture processing uses a digital subspace of $\mathbb{R}^n$. Based on the introduced adjacency models (grid point model or cell model), algorithms have been developed using discrete Minkowski metrics on those subspaces. Let $p, q \in \mathbb{Z}^2$ be grid points. The *city-block metric* or *Manhattan metric* for the 2D grid is defined as follows:

$$d_4(p, q) = \mid x_1 - y_1 \mid + \mid x_2 - y_2 \mid \tag{2.5}$$

Obviously, the d_4-metric coincides with the 2D Minkowski metric L_1. Analogously, the $d_6(p, q)$ for the 3D space coincides with the 3D Minkowski metric L_1.

The d_8-metric is known as *chessboard metric*. It is defined for the 2D grid as follows:

$$d_8(p, q) = max\{\mid x_1 - y_1 \mid, \mid x_2 - y_2 \mid\} \tag{2.6}$$

The d_8-metric coincides with the 2D Minkowski metric L_∞. Analogously, the $d_{26}(p,q)$ for the 3D space coincides with the 3D Minkowski metric L_∞.

The *Euclidean metric* d_e defines the Euclidean space $\mathbb{E}^n = [\mathbb{R}^n, d_e]$. $d_e(p,q)$ equals the length of a straight line segment between points $p = (x_1, x_2, ..., x_n)$ and $q = (y_1, y_2, ...y_n)$ and is defined as follows:

$$d_e(p,q) = \sqrt{(x_1 - y_1)^2 + ... + (x_n - y_n)^2} \tag{2.7}$$

This metric coincides with the Minkowski metric $L_2(p,q)$, and the result is a real number. The digital geometry literature offers a diversity of metrics, often with the intention to approximate the Euclidean distance while retaining the simplicity of algorithms. The following theorem [see, e.g., (Klette and Rosenfeld 2004)] provides a theoretical background for some of those approximations .

1. THEOREM.

$$d_8(p,q) \leq d_e(p,q) \leq d_4(p,q) \leq 2 \cdot d_8(p,q) \tag{2.8}$$

for all $p, q \in \mathbb{R}^2$*,*

$$d_{26}(p,q) \leq d_e(p,q) \leq d_6(p,q) \leq 3 \cdot d_{26}(p,q) \tag{2.9}$$

for all $p, q \in \mathbb{R}^3$*, and*

$$d_{26}(p,q) \leq d_{18}(p,q) \leq d_e(p,q) \tag{2.10}$$

for all $p, q \in \mathbb{Z}^3$ *such that* $d_e(p,q) \neq \sqrt{3}$*.*

It is common to use different metrics for the calculation of *distance skeletons*, which are realizations of digital medial axes (based, for example, on heuristics to ensure connectedness of the distance skeleton for a given connected set).

Depending on the choice of the metric (and of the applied heuristics), the calculation of distances in digital pictures deliver different distance skeletons.

For example, in Figures 2.6 and 2.7, skeletons are local maxima of distance maps (see Chapter 4), calculated with different metrics.

For approximations of the Euclidean distance, some authors suggested the use of different weights for steps within a grid point neighborhood (Sanniti di Baja 1994). (Montanari 1968) introduced quasi-Euclidean distances, such as $(d_4(p,q) +$

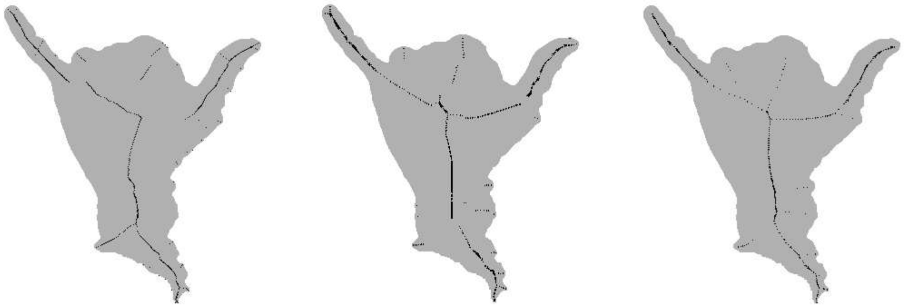


Figure 2.6: *Local maxima of distance maps, calculated with d_4-metric (left), d_8-metric (middle) and d_e-metric (right).*

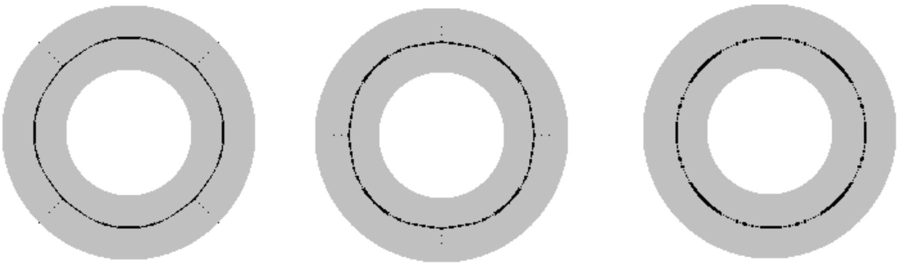


Figure 2.7: *Left: d_4-metric, middle: d_8-metric, right: d_e-metric.*

$d_8(p, q))/2$ or $(d_4(p, q) + 2 \cdot d_8(p, q))/3$, for reasonable approximations of Euclidean distances.

Chamfering[4] is a general method of defining metrics in a grid using weights for individual steps in the grid.

12. DEFINITION. *Given are a neighborhood and weights for steps in this neighborhood. The* chamfer distance *between two elements p and q is the minimum total weight of an arc between p and q.*

The method is closely related to the choice of a particular neighborhood in the nD grid.

For example, consider the possible steps in the 8-neighborhood in the 2D grid. Assume real weights a or b for isothetic or diagonal steps, respectively. Weights have to satisfy criteria such that the resulting distance function is a metric. The chamfer

---

[4] The name originates from the technological processes when chamfering an edge (of metal, timber, and so forth).

distance defined by a and b is a metric iff $0 < a \leq b \leq 2 \cdot a$ (the *Montanari condition*). For example, values $a = 3$ and $b = 4$ ($c = 5$ for 3D) satisfy this condition and are often used in applications. Optimizing those weights and using larger neighborhoods are methods to reduce approximation errors by retaining the simplicity of algorithms (Borgefors 2005) which are based on local operations.

2.6 Summary

This chapter provided an overview of used notions such as adjacency, neighborhoods, connectedness, the grid point model, or the cell model. It also introduced briefly more specific concepts such as attachment sets, the digital fundamental group, or different types of metrics, which are used in this thesis.

Chapter 3

Skeletons

The interest in picture simplification methods started in the early days of digital picture processing. Technical limitations (hardware and software) motivated researchers to find ways for picture compression in order to reduce the amount of data for further processing steps. It is important in this context that connected components (regions) in a binary picture can be recovered from a relatively small subset of the original data. The motivation has slightly changed over the years towards the goal of associating objects in a picture with descriptors (features) for classification or registration. The methods described in this thesis aim to map objects onto their skeletons (short for approximations of skeletal curves), and we are interested in "reliable" invariants of those skeletons to describe pictures. In this chapter we review skeleton models which have been proposed in the literature.

3.1 Skeletons and Skeletonizations

Information in a binary (digital) picture is completely represented by size, shape and location of its object regions. Basic descriptors for connected components such as area, volume, perimeter, centroid, and others are useful for special applications. Derived descriptors such as the *shape factor* (i.e., in 2D the ratio between area and square of diameter) are suitable for the representation of "elongated" versus "round" regions. The choice of descriptors depends not only on the type of the given pictures. It also depends on the final goal of the entire process. *Compression* uses the representation of a picture for storing original data in reduced space. The ideal result of picture compression would be minimum-size data which is sufficient to recover the original picture. *Classification* or *registration* require representations of pictures for identifying significant properties which define classes. Those representations or properties should be sufficient to associate a given picture with a desired class of regions. In biomedical applications, the research is often focused on finding such properties that allow a distinction between biomedical material of a patient versus that of a normal human. At a more advanced stage the interest is moving to a distinction between different stages of diseases, that means that the number of classes increases.

The literature offers a diversity of methods for picture representation (e.g., borders of components, moments). Skeletons have been studied extensively because they are efficient in terms of representing properties of elongated components which occur in many classes of pictures (e.g. road maps, character recognition, finger prints, blood vessels). The components in astrocyte pictures constitute a large number of elongated parts. Thin curve-like structures at central locations are intuitive representations of such pictorial structures.

The word "skeleton" is not uniquely defined in the picture analysis literature. We use it as general term for approximations of skeletal curves (which are defined by topologic or geometric concepts in the continuous space), and give an informal description (for $n = 2$ and $n = 3$) which highlights the expectations on skeletons with respect of using them for classification, registration or compression. Let M be the set of object pixels or voxels in the original picture, and S be the set of pixels or voxels which belong to the derived skeletons of a picture. (Marchand-Maillet and Sharaiha 2000) postulate for this case the following:

1. S consists of a set of digital arcs or curves, which may have branch nodes.
2. S is at a central position inside the object regions.
3. S is sufficient to reproduce the original object regions.
4. S has the same number of connected components of $1s$ as the original picture.
5. The complement of S, $\overline{S}$, has the same number of connected components of $0s$ as the original picture.

Obviously, postulates 4 and 5 do not guarantee topologic equivalence (which is, in 2D, isomorphism of the rooted region adjacency graphs, where the root is the uniquely defined background component; see Section 2.4). For theoretical reasons, 4 and 5 should be replaced by the request to guarantee topologic equivalence between both pictures (i.e., based on isotopy). However, postulates 1-5 are acceptable as a compromise for algorithmic reasons: so far, no algorithmic study is known about isotopy tests between pictures.

A method for generating skeletons is called a *skeletonization*. We can use above postulates as requirements for "good" algorithms.

There are further formulations of conditions in the literature for skeletonization algorithms:

1. The topology of the picture (i.e., in 2D its region adjacency graph) must be preserved.

2. The resulting subset S (for a given region) must be a connected set of digital curves or arcs.
3. The resulting subset S must approximate the medial axis. There has to be an algorithm allowing to reproduce the original data from this subset.
4. The algorithm should be computationally efficient.
5. The algorithm should be robust against noise.
6. End elements (all elements with exactly one α-adjacent object element) must be preserved.

A precise formulation of Condition 1 could be based on isotopy (as in Section 2.4). We could also ask for identity (up to isomorphisms) of the fundamental groups of the object regions, and of the non-object regions. We can also request that S and M must be homotopic. (In Chapter 5 we discuss the definition of homotopic sets.)

Condition 2 is obviously a consequence of Condition 1. Condition 3 contains the vague term "approximate", but the request for an existence of a reconstruction algorithm is a precise statement. Condition 4 should be precisely formulated in asymptotic complexity terms; in 2D we have $M \times N$ pixels in one picture, and linear run time would be $\mathcal{O}(MN)$. Condition 6 is only interesting for iterative thinning algorithms. The algorithm should stop removing elements if all remaining object elements after a number of iterations are "non-simple" elements or end elements.

More precise formulations of such requirements are desirable, and the more precise, the better they can be used for comparisons or evaluations of shape simplification methods.

The individual conditions are also of varying importance depending on the purpose for computing skeletons. For example, it is essential for picture compression that original regions are reconstructible, and for this application it is not important that the skeleton picture is topologically equivalent to the original picture. In case of registration or classification, it is not necessary to be able to reconstruct the original picture, but representations need to be accurate for the interpretation of the content of pictures.

3.2 Topologic Skeletons

Skeletons have been known in continuous space for about 150 years. Listing (Listing 1861) introduced the *linear skeleton* (under the name *cyclomatic diagram*), which is a result of a continuous contraction of a connected subset of an Euclidean space (i.e., without changing the topology of the original set with respect to homotopy), until only a connected curve remains, being a union of arcs and simple curves, possibly (in case of a *simply-connected* input set) also just a single point.

The linear skeleton is defined in topology, and we call its approximation in picture analysis a *topologic skeleton*; they will be considered in detail in Chapter 5. Calculations of topologic skeletons attempt to digitize the concept of topology-preserving continuous contraction by iterative thinning.

Such algorithms delete a set of border pixels or border voxels with special properties in one iteration (typically those picture elements are called *simple pixel* or *simple voxel*), and the result is the input for the next iteration, until a set of pixels or voxels with special properties (e.g., all are non-simple or end elements) remains. The remaining set of pixels or voxels constitutes the topologic skeleton of the picture. The original picture is not reconstructible (i.e., postulates defined by interests in picture encoding will be violated).

One example as an alternative to iterative thinning as a topology-preserving operation is discussed in (Glantz and Kropatsch 2001, Haxhimusa and Kropatsch 2003). Authors propose a dual graph contraction to compute a ”skeleton graph”. A discussion of this approach is not part of this thesis.

3.3 Geometric Skeletons

About one hundred years after Listing’s definition of topologic skeletal curves, a second mathematical approach (i.e., the medial axis) for describing skeletons has been proposed for the continuous space in (Blum 1967).

The medial axis is an example of a geometrically defined skeletal curve. This section reviews *geometric skeletons* , which are either approximations of skeletal curves geometrically defined in continuous space, or directly defined in digital space by geometric means.

Medial or Symmetric Axis

Let $B_r(p)$ be the *Euclidean ball* with radius $r > 0$ (with respect to the Euclidean metric d_e) centered at point p in $\mathbb{R}^n$. (For $n = 2$, we speak about a *Euclidean disk*.) An Euclidean ball $B_r(p)$ is *maximal* in $M \subseteq \mathbb{R}^n$ iff $B_r(p) \subseteq M$ and there is no Euclidean ball $B_s(q) \subseteq M$ with $B_r(p) \subset B_s(q)$. The radius of a maximal ball in $M \subseteq \mathbb{R}^n$ equals the minimum distance from its center to the frontier of M.

We call a point $p \in M$ *symmetric* iff at least two different points q_1 and q_2 exist on the frontier of M with $d_e(p, q_1) = d_e(p, q_2)$. See Figure 3.1. Note that points on the frontier of M cannot be symmetric.

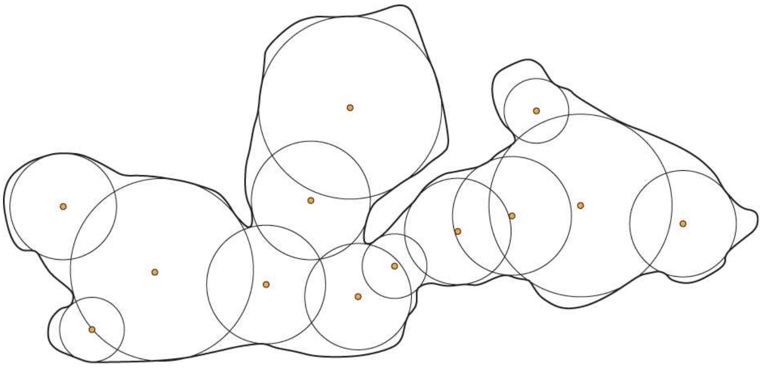


Figure 3.1: *A subset of symmetric points in the Euclidean plane*

13. DEFINITION. *In 2D, the set of symmetric points of a compact set $M \subset \mathbb{R}^2$, each labeled with the radius r of the associated disk, constitute the symmetric axis of M. Analogously, in nD we use the radius of associated balls for defining the symmetric axis. The* medial axis *of M is the set of centers of maximal (disks) balls totally included in M.*

Note that this definition excludes points on the frontier of M from being members of the symmetric axis. If desired, we can also consider the closure of the symmetric axis (i.e., branches then also contain their endpoints), and a point on the frontier is then labeled by $r = 0$ if in the symmetric axis.

The medial axis is a subset of the symmetric axis. If the frontier of a disk (or ball) would cross the frontier of M then points exist which are closer to the center and the disk (or ball) is not maximal. For a convex set M the symmetric axis coincide with the medial axis.

There are many approaches in picture analysis for calculating a medial axis of a region M in a picture, using either Euclidean balls, or balls defined by other metrics (see Section 2.5 for alternative metrics, which define medial axis with respect to the chosen metric). The mapping from a given region M into a set S of centers of maximal discs (or balls), labeled by their corresponding radii r, is called *medial axis transform*. The resulting set S of labeled picture elements is also typically referred to as *medial axis*. See Figure 1.3 for an example of a d_4-medial axis. There are only two pixels in this medial axis. The non-connectedness of medial axes in the discrete space is a drawback for shape representation applications, but not for picture compression or encoding.

Medial axis transforms proceed often in two steps. At first, all the picture elements in a region M are labeled by the (shortest) distance to a picture element in $\overline{M}$.

This step is also called a *distance transform*. Then, the centers of maximal discs (or balls) are identified followed by a postprocessing step which connects the elements of the medial axis.

We just mention at this point that distance transforms for adjacency-related metrics (i.e., not, for example, for the Euclidean metric) can also be obtained by applying morphologic operations (without further use of this approach later on in this thesis). Let M be a subset of a picture, and B a *structuring element* (i.e., a "small" subset of the grid). $(M \ominus kB)$ means $k \geq 0$ successive erosions of M by B, with $0B = \emptyset$. $M \circ B$ means the opening of M by B [(Serra 1982)].

A distance transform (defined by the structuring element) can then be defined in terms of erosions and openings. Following (Serra 1982), the *morphological skeleton* S of a region M can be calculated using the following:

$$S(M) = \bigcup_{k=0}^{K} S_k(M)$$

with

$$S_k(M) = (M \ominus kB) - ((M \ominus kB) \circ B)$$

K is the number of the last iterative step before M erodes to an empty set. During this process we label pixels in $S_k(M)$ with k, which are still in the object at iteration step k.

For example, for determining in 2D the labels of the d_8- or d_4-distance transform, the structural elements are

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \qquad \begin{matrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{matrix}$$

respectively.

Grassfire Transform

(Calabi and Hartnett 1968, Montanari 1969) proposed an alternative approach for calculating a digital version of a medial axis. The concept is called *prairie fire* or *grassfire transform*.

Assume a fire front that starts at the same time at every point on the frontier of a compact set $M \subset \mathbb{R}^n$ which moves with constant speed into M, locally perpendicular to the frontier, until a fire front collides (at a point $p \in M$ at time t) with another fire front. These *quench points* p, labeled by t, define then a skeletal curve which coincides with the Euclidean medial axis if the time scale is chosen such that t is identical to the distance r to the frontier.

The benefit of this approach is that it allows to calculate connected skeletal curves when digitizing the concept of medial axes, by removing border points, layer by layer, like peeling an onion, and stopping when two removal processes meet at one pixel.

Voronoi Transform

Some authors [see, for example, (Breu et al. 1995)] describe skeletons based on Voronoi diagrams in the continuous space. By above definition, symmetric axes are sets of symmetric points which are equidistant from at least two points on the frontier. They can be represented as subsets of the frontier of Voronoi cells, defined within a compact set M by points on the frontier of M. (For further details, see the reference.)

Middle-Line Transform

There are also many proposals for geometric skeletons based on heuristics. We only give one example.

(Shapiro et al. 1981) uses frontier segments for the generation of skeletons. The approach assumes that the frontier of an object M is very smooth. Oppositely located segments A and B of the frontier are used to locate midpoints of a *minimum-base segment* which is the shortest straight line between A and B. A simplified version of this model assumes even that the frontier lines have vertical or horizontal directions using the argument that it is possible to calculate the main direction of patterns in some pictures. The approach is defined for the discrete 2D space, and it works only for very special cases. It is also extremely noise- and rotation sensitive. Its generalization to 3D seems to be not of interest for the determination of skeletons in complex 3D structures such as segmented astrocytes.

3.4 Calculation of Geometric Skeletons

There are different ways to apply the mathematic models discussed above in digital picture processing. One category of algorithms identifies sets of skeletal pixels or voxels using distance transforms (see Chapter 4 for details on distance transforms).

For simplicity, the extraction of distance skeletons is often done by propagating distance transform values from already labeled adjacent points. Local maxima (maximum of distance transform values in a given neighborhood) do not always coincide with centers of maximal disks (or balls).

Let $t(p)$ be the result (i.e., distance value to $\overline{M}$) of a distance transform at pixel $p \in M$. Assume that $t(p)$ is a local maximum ["local" means within $N_\alpha(p)$] in a distance transform picture of M, where the distance transform was calculated with respect to a metric d. Then p is the center of a maximal disk or ball if

$$t(q) < t(p) + d(p,q), \quad \forall q \in A_\alpha(p) \tag{3.1}$$

The value $t(p)$ represents [in some way, often in the form $r = t(p) - 1$] the radius r of the maximal disk centered at p. The set of resulting centers of maximal disks or balls, defining the skeleton in this case, depends on the chosen metric d and the type of adjacency. (We discuss distance transform algorithms for different metrics and neighborhoods in the next chapter.)

In discrete spaces, the identified skeleton is in general not connected, which violates common postulates as stated in Section 3.1. A sketch of a procedure to map a disconnected skeleton into a connected one may be as follows:

(i) For initialization, calculated skeletal pixels are candidates for a final set of skeletal picture elements.

(ii) Postprocessing steps are required to ensure topologic equivalence and to produce (connected) skeletons consisting of digital curves or arcs. The next chapter includes a method as a proposal to solve this problem.

The set of skeletal picture elements, each labeled with the radius of the maximal ball, forms the input data for a *reverse distance transform* which allows the reconstruction of the original objects. Algorithms to compute the reverse distance transform are also discussed in Chapter 4.

3.5 Summary

This chapter describes informally expectations on skeletons for different purposes. It informs about mathematical models for topologic or geometric skeletons, typically formulated in the continuous space. It also informs briefly about concepts for calculating topologic or geometric skeletons in the discrete space.

Chapter 4

Distance and medial axis transforms

The distance transform is a "powerful tool" in digital picture processing. The result of this operation is called distance map. The reverse distance transform identifies the original binary picture from a subset of object elements and the associated distance transform values. Distance transforms are important preprocessing steps in complex picture analysis systems. Operations such as skeletonization, merging and smoothing rely on accurate distance maps. Methods need to be time efficient and the results need to be exact for analysis purposes. We review groups of algorithms in the following as they have been developed, and we give precise definitions for different types of distance skeletons.

4.1 Distance transform

The distance transform labels each object element with the distance between this element and the nearest non-object element. For all elements $p \in P$, the algorithm determines

$$t(p) = min_k\{d(p, q_k) : P(q_k) = 0 \ \wedge \ 0 \leq k \leq m\} \tag{4.1}$$

where $d(p, q_k)$ denotes a metric, and m is the total number of elements in the picture. It follows that $t(p) = 0$, for all non-object elements. Obviously, the values for $t(p)$ depend on the chosen metric. Independent of the type of metric, for given sets of object elements M and non-object elements B, the distance transform has the following properties:

1. PROPOSITION. *(i) $t(p)$ represents the radius of the largest disk or ball centered at p and contained in M.*

(ii) If there is only one non-object element $q \in B$ with $t(p) = d(p, q)$ then there are two cases:

(ii.1) an element $p' \in M$ exists such that the largest disk or ball centered at p' contains the disk or ball centered at p;

(ii.2) elements $p' \in M$ and $q' \in B$ exist such that $d(p, q) = d(p', q')$ and p is α-adjacent to p'.

(iii) If there are two (or more) non-object elements $q \in \langle \overline{P} \rangle$ and $q^{'} \in B$ such that $t(p) = d(p, q) = d(p, q^{'})$ then the disk or ball centered at p is a maximal disk in M, and p is symmetric.

In Case (ii.2), elements p and p' are both centers of maximal discs and they are adjacent to each other. Many algorithms extract distance skeletons for those cases which are not 1D curves (see Figure 1.3).

The distance map is a 2D array (3D for volume pictures) of the same size as the original picture which stores the results $t(p)$, for all elements p. The following sections review four groups of algorithms for the calculation of the distance transform in the digital space, based on different metrics.

4.1.1 Two pass algorithms

The first two pass algorithm (TPA) has been published in 1966; see (Rosenfeld and Pfaltz 1966). The authors used the d_4-metric. The basic algorithm works as follows: Let $p = (x, y)$ (x,y are integer coordinates) and $T = t(x, y)$ be the picture which results when f_1 is applied to the picture P in forward raster sequence (from the top left corner to the bottom right corner), followed by f_2 in reverse raster sequence (from the bottom right corner to the top left corner). Then

$$t(x, y) = f_2(f_1(x, y)) \text{ if } P(x, y) = 1 \quad (4.2)$$

$$t(x, y) = 0 \text{ if } P(x, y) = 0 \quad (4.3)$$

is the distance transform $T = t(x, y)$ of P if

$$f_1(x, y) = min\{P(x - 1, y) + 1, P(x, y - 1) + 1\} \quad (4.4)$$

$$f_2(x, y) = min\{f_1(x, y), f_2(x + 1, y) + 1, f_2(x, y + 1) + 1\} \quad (4.5)$$

TPA-algorithms are applicable in connection with different grid metrics, different neighborhoods and higher dimensions. This leads to general definitions for f_1 (result of the first pass) and f_2 (result after second pass) in 2D:

$$f_1(x, y) = min\{S_1\} \quad (4.6)$$

$$f_2(x, y) = min\{S_2\} \quad (4.7)$$

where

$$S_1 = \{P(x - 1, y) + a, P(x, y - 1) + a, P(x - 1, y - 1) + b, P(x + 1, y - 1) + b\} \quad (4.8)$$

and

$$S_2 = \{f_1(x,y), f_2(x+1,y)+a, f_2(x,y+1)+a, f_2(x-1,y-1)+b, f_2(x+1,y+1)+b\} \tag{4.9}$$

The increments a (isothetic step) and b (diagonal step) are different depending on the chosen metric. In TPA algorithms they represent distances between adjacent elements. For the d_4-metric the values are: $a = 1$ and $b = 2$, for the d_8-metric: $a = 1$ and $b = 1$. If we consider 3D pictures then a is defined as weight for steps between 2-adjacent voxels, b is defined as weight for steps between 1-adjacent voxels and an additional weight c is required for 0 adjacent voxels. Chamfer distances have been introduced (see Chapter 2) to approximate the Euclidean distance with the aim to keep the simplicity of TPA-algorithms originally designed for grid metrics d_4 and d_8. Weighted distance transforms for 3D pictures based on this concept have been studied in (Borgefors 1984, Borgefors 1996, Ragnemalm 1993). In general, those algorithms deliver approximations of Euclidean distances. The time complexity of those algorithms is linear $\mathcal{O}(n)$, where n is the total number of pixels in the picture.

4.1.2 Vector propagation algorithms

A first vector propagation algorithm (VPA) has been published in 1980; see (Danielsson 1980). The basic approach remains the same as for TPA-algorithms. The distance is calculated by minimizing the incremental distance from its neighbors. To each pixel $p = (x, y)$ we assign a vector (pair of integers in 2D, triple in 3D). The initial values are:

$$f(x,y) = (0,0) \text{ if } P(x,y) = 0 \tag{4.10}$$

$$f(x,y) = (d,d) \text{ if } P(x,y) = 1 \tag{4.11}$$

The value for d is the length of the diagonal of the picture. We use the following definitions for determining the minimum of those vectors:

$$min\{(u_1,v_1),(u_2,v_2)\} = (u_1,v_1) \text{ if } u_1^2+v_1^2 < u_2^2+v_2^2 \tag{4.12}$$

$$min\{(u_1,v_1),(u_2,v_2)\} = (u_2,v_2) \text{ if } u_1^2+v_1^2 > u_2^2+v_2^2 \tag{4.13}$$

$$min\{(u_1,v_1),(u_2,v_2)\} = (u_1,v_1) \text{ if } u_1^2+v_1^2 = u_2^2+v_2^2 \wedge u_1 < u_2 \tag{4.14}$$

$$min\{(u_1,v_1),(u_2,v_2)\} = (u_2,v_2) \text{ if } u_1^2+v_1^2 = u_2^2+v_2^2 \wedge u_2 < u_1 \tag{4.15}$$

A pair of values (u, v) is calculated in a sequential algorithm, for each pixel. The first set of scans (from the top, left, to the bottom, right) calculates the following:

$$f_1(x,y) = min\{f(x,y), f_1(x,y-1)+(0,1)\} \tag{4.16}$$

$$f_2(x,y) = min\{f_1(x,y), f_2(x-1,y)+(1,0)\} \tag{4.17}$$

$$f_3(x,y) = min\{f_2(x,y), f_3(x+1,y)+(0,1)\} \tag{4.18}$$

The second set of scans (from the bottom, right, to the top, left) calculates the following:

$$f_4(x,y) = min\{f_3(x,y), f_4(x,y+1) + (0,1)\} \tag{4.19}$$

$$f_5(x,y) = min\{f_4(x,y), f_5(x-1,y) + (1,0)\} \tag{4.20}$$

$$f_6(x,y) = min\{f_5(x,y), f_6(x+1,y) + (1,0)\} = (u,v) \tag{4.21}$$

Value u should represent the difference of the x-coordinates between p and the closest pixel q in the background, and v should represent the difference of the y-coordinates between those pixels. However, this is not true in some cases. The Euclidean distance for each pixel to the nearest non-object pixel is easy to compute:

$$d_e(x,y) = \sqrt{u^2+v^2} \tag{4.22}$$

Figure 4.1 is a sketch to show that errors appear in some special cases. Pixel $q = (x-(a+1), y)$, $r = (x-c, y+(d+1))$, and $s = (x, y+(a+1))$ are non-object, and $p = (x,y)$ is an object pixel. The algorithm would identify s as the closest non-object pixel and $d_e(p,s) = a+1$. Obviously, based on the triangle inequality, we have $d_e(p,r) = b < a+1$.

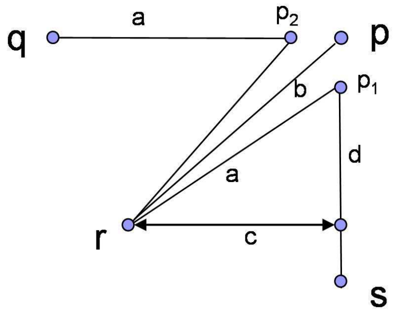


Figure 4.1: *Error case for VPA-algorithm*

Those errors happen if pairs of integers are compared for the minimum calculation which deliver the same values for the sum of squares, for example (3,4) and (0,5). Additional tests for the determination of the minimum can avoid these problems. We can estimate an error in cases as illustrated by Figure 4.1, where we have:

$$
\begin{aligned}
c &= b/\sqrt{2} \\
d &= b/\sqrt{2} - 1 \qquad (4.23) \\
a^2 &= c^2 + d^2 \qquad (4.24)
\end{aligned}
$$

From the above equations it follows that

$$b \simeq a + 1/\sqrt{2} < a + 1 \qquad (4.25)$$

That means, such an error is smaller than the grid constant, and with increasing grid resolution they have almost no impact.

The time complexity of the algorithm is linear $\mathcal{O}(n)$, where n is the total number of pixels in the picture. In (Mullikin 1992), the concept has been adopted for 3D space.

4.1.3 Iterative distance transform algorithms

For completeness reasons, we briefly describe an iterative algorithm which uses the grassfire model and weighted distances between adjacent elements [see for example (Gagvani and Silver 1997)]. We assign ∞ to all object elements, and 0 to all non-object elements. All object elements adjacent to a non-object element are border elements. Let B be the set of those elements.

The initial pass assigns a minimum distance value t_p, to each border voxel $p \in B$, depending on the adjacency type (for example, 3 for 6-adjacency, 4 for 18-adjacency, 5 for 26-adjacency in 3D, see Section 2.5). The following path calculates, for each adjacent element q, the minimum distance $t_q = min\{t_q, t_p + c\}$ where c is the weighted distance for the adjacency type. The algorithm removes all elements $p \in B$ from B and adds all elements q to B after each iteration. The algorithm stops if no changes of distance values occur.

The result is a distance map equivalent to a distance map obtained by two-path algorithms (if the same metric has been used). The time complexity is $\mathcal{O}(n)$ (where n is the number of elements in the picture) but the number of passes is in general larger than 2 depending on the size of objects. Efficient implementations can reduce the number of operations based on the fact that in each iteration only a small subset of elements (only border elements) need to be considered.

4.1.4 Envelope algorithms

Envelope algorithms (EVA-algorithms) calculate Euclidean distance maps without errors, for arbitrary dimensions. The approach starts with integer operations on elements for one dimension. Then it operates in the continues space by computing the lower envelope of a family of parabolas, and it finishes by assigning *squared Euclidean distance* values of the lower envelope to elements. Consider the squared Euclidean distance between two points $p = (x_1, x_2, ..., x_n)$ and $q = (y_1, y_2, ..., y_n)$:

$$d_e^2(p, q) = (x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2 \tag{4.26}$$

We review the algorithm for the 2D case, and we discuss generalizations for other metrics or for arbitrary dimensions. Computations can be done independently for each dimension. The algorithm starts with a 1D transform, and then it adds one step for the next dimension and merges results.

The initial step is the calculation of the distance from each object pixel to the nearest non-object pixel in the same row:

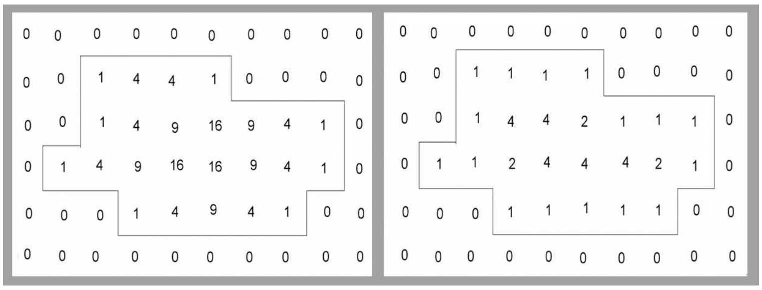
$$f_1(x, y) = f_1(x-1, y) + 1 \text{ if } P(x, y) = 1 \tag{4.27}$$

$$f_1(x, y) = 0 \text{ if } P(x, y) = 0 \tag{4.28}$$

$$f_2(x, y) = min\{f_1(x, y), f_2(x+1, y) + 1\} \text{ if } f_1(x, y) \neq 0 \tag{4.29}$$

$$f_2(x, y) = 0 \text{ if } f_1(x, y) = 0 \tag{4.30}$$

f_1 determines the distance between pixel p and the left nearest non-object pixel q, and f_2 replaces f_1 if the distance to the right border is shorter. The result is a matrix which stores integer values $(f_2(x, y))^2$ in each pixel (see Figure 4.2).



0	0	0	0	0	0	0	0	0	0
0	0	1	4	4	1	0	0	0	0
0	0	1	4	9	16	9	4	1	0
0	1	4	9	16	16	9	4	1	0
0	0	0	1	4	9	4	1	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0	0	0
0	0	1	4	4	2	1	1	1	0
0	1	1	2	4	4	4	2	1	0
0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0

Figure 4.2: *Left: result after row scans. Right: results after column scans*

We can express $f_2(x, y)$ for a fixed y as follows:

$$f_2(x, y) = min\{|x - i| : P(i, y) = 0 \ \wedge \ i = 1, ..., n\} \tag{4.31}$$

The next step determines $f_3(x, y)$, column by column (x fixed, and then for all y). Assuming that we have n columns and n rows, we have:

$$f_3(x,y) = min\{(f_2(x,j))^2 + (y-j)^2 : \ j = 1,...,n\} \tag{4.32}$$

Efficient implementations for solving this minimization problem have been discussed in various papers (Saito and Toriwaki 1994, Hirata 1996, Meijster et al. 2000, Toriwaki and Mori 2001, Bailey 2004). We give a geometric interpretation that illustrates the idea. For a fixed column (i.e., x constant, $f_2(x,j) = g(j)$) and a fixed row (i.e., y constant) the following equation

$$\gamma_y(j) = (g(y))^2 + (y-j)^2 : \ j = 1,...,n \tag{4.33}$$

represents one parabola. For $1 \leq y \leq n$ we consider a family of n parabolas (see Figure 4.3), one parabola for each row. Note that the horizontal axis represents the row number y, and the vertical axis represents $\gamma_y(j)$, with a local minima at $y = j$ and $\gamma_y(j) = (g(j))$.

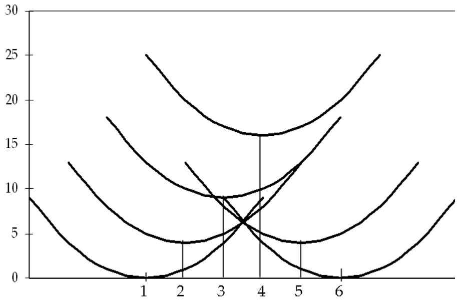


Figure 4.3: *Family of parabolas for column [0,4,9,16,4,0] in Figure 4.2*

The lower envelope of the family of parabolas corresponds to above minimum calculation. Typically, those algorithms calculate the lower envelope of the family of parabolas and then they assign the height of the lower envelope to the point with coordinates (x, y). The computation of the lower envelope of the family of parabolas is the main part of the algorithm.

The example in Figure 4.3 shows a family of six parabolas. The lower envelope consists of two curve segments. The first segment starts at $(1, 0)$ and it ends at the intersection of the first and the last parabola. The second segment begins at the intersection and ends at $(6, 0)$.

The projections of the segments on the horizontal axis are called *sections*. In the given simple example, the interval [1,6] is partitioned into two sections. Only two of six parabolas contribute to the lower envelope of the family. To calculate $f_3(y)$

(x fixed) we need the start and end for each section, and the index of the associated parabola.

This can be done in two more column scans: one scan from top to bottom that identifies the parabola segments of the lower envelope together with its associated sections, and a second scan that calculates the values for $f_3(y)$.

The determination of the lower envelope is done by a sequential process of computing the lower envelope of the first k parabolas. We calculate the y-coordinates of the intersection between two parabolas. Let y_s be the abscissa of the intersection and let $y_1 < y_2$. From the equation

$$(g(y_1))^2 + (y_s - y_1)^2 = (g(y_2))^2 + (y_s - y_2)^2 \tag{4.34}$$

for the intersection $y_s = y_s(\gamma_1, \gamma_2)$ of any two parabolas γ_1 and γ_2, we derive

$$y_s = y_2 + \frac{(g(y_2))^2 - (g(y_1))^2 - (y_2 - y_1)^2}{2(y_2 - y_1)} \tag{4.35}$$

or

$$y_s = \frac{(g(y_2))^2 - (g(y_1))^2 + y_2^2 - y_1^2}{2(y_2 - y_1)} \tag{4.36}$$

We give an informal description of the basic algorithm. The data structure to store the information is a stack. Only parabolas which contribute to the lower envelope stay in the stack, and all the others are eliminated from the stack. A straightforward algorithm is the following (see the sketch in Figure 4.4):

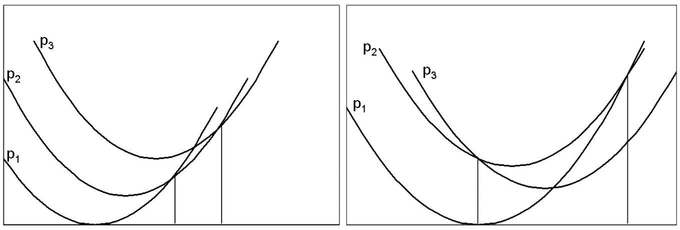


Figure 4.4: *Left sketch:* $y_s(\gamma_2, \gamma_3) > y_s(\gamma_1, \gamma_2)$, *right sketch:* $y_s(\gamma_2, \gamma_3) < y_s(\gamma_1, \gamma_2)$

Each stack item stores a pair of real values (b, e) for the start and end of the section of a parabola which contributes to the lower envelope.

(b_t, e_t) belongs to the top parabola of the stack, and (b_f, e_f) is the pair associated with the following parabola in the sequential process. The first item stores the start and end of the section for the first parabola. It is initialized with $(1, n)$, the lower envelope would consist of one segment if all the following parabolas have no common intersections between $(1, n)$.

Note that the parabolas are ordered according to their y-values in the picture. For each sequential step, we evaluate the intersection for the top item of the stack representing γ_t and the next following parabola γ_f. There are three cases:

1. $y_s(\gamma_t, \gamma_f) > n$: γ_f does not contribute to the lower envelope; do not change the stack, take the subsequent parabola;
2. $y_s(\gamma_t, \gamma_f) \leq b_t$: remove γ_t from the stack; evaluate the intersection of the new top item with γ_f (right case in the sketch in Figure 4.4), if the stack is empty, add the item for γ_f to the stack
3. $y_s(\gamma_t, \gamma_f) > b_t$: adjust γ_t with $e_t = y_s(\gamma_t, \gamma_f)$, add the item for γ_f to the stack, with $b_f = e_t, e_f = n$ (left case in sketch in Figure 4.4).

The procedure continues until the last parabola has been evaluated with the top item of the stack. At the end, only sections of the lower envelope are registered in the stack, and they are used for calculating the values for $f_3(x, y)$ in an additional scan.

Some authors reduced the number of computations by analyzing certain properties. For example, in (Bailey 2004) the author avoids a direct computation of the lower envelope to save divisions in Equation (4.36). If $(g(y_2))^2 - (g(y_1))^2 \leq (y_2 - y_1)^2$ [see Equation (4.35)] then $y_s \leq y_2$, and if $(g(y_2))^2 - (g(y_1))^2 > (y_2 - y_1)^2$ then $y_s > y_2$. These comparisons are sufficient to know whether the intersection is before or after y_2. The property is used for a direct computation of the squared Euclidean distance map in two scans.

An optimal (in asymptotic time) algorithm is presented in (Meijster et al. 2000). Properties are used to eliminate parabolas from a stack (where the parabolas for the lower envelope are stored) in order to reduce the number of calculations for intersections. The algorithm works in linear time. The same paper demonstrates the generality of the algorithm with respect to different metrics. The minimization problem for fixed x (note that y and j represent pixels or voxels (x, y) and (x, j), respectively, in the picture) of Equation (4.32) can be expressed by the following:

$$f_3(y) = min\{d(y, j) : P(j) = 0 \ \wedge \ j = 1, ..., n\} \tag{4.37}$$

with

$$d(y, j) = (g(j))^2 + (y - j)^2 \text{ for } d_e^2 \tag{4.38}$$

$$d(y, j) = |y - j| + g(j) \text{ for } d_4 \tag{4.39}$$

$$d(y, j) = \max(|y - j|, g(j)) \text{ for } d_8 \tag{4.40}$$

Equation (4.37) is equivalent to the general definition for distance transforms in Section 4.1. It shows that the same method is applicable for grid metrics d_4 and d_8. Computations for each dimension are done independently. The result of calculating a minimum distance for one dimension is an integer value for each grid point which will be used for the computation in the next dimension. The 2D distance transform can be expressed as follows:

$$t(x,y) = \min\{(x-i)^2 + (y-j)^2 : i = 1,...,n \ \wedge \ j = 1,...,n\} \tag{4.41}$$

Because i does not depend on j, we have:

$$t(x,y) = \min\{\min((x-i)^2) + (y-j)^2 : i = 1,...,n \ \wedge \ j = 1,...,n\} \tag{4.42}$$

The minimum calculation $\min((x-i)^2) = g(j)$ in Equation (4.42) corresponds to the row scans in the first part of the algorithm. We can rewrite the equation (for fixed x) as follows:

$$t(x,y) = \min\{g(j) + (y-j)^2 : j = 1,...,n\} \tag{4.43}$$

Let p be at one of the 3D locations (x,y,k), $k = 1,...,n$, associated with $f_3(x,y,k) = h(k)$ [for fixed (x,y)], and let

$$t(x,y,z) = \min\{h(k) + (z-k)^2 : k = 1,...,n\} \tag{4.44}$$

This can be done for arbitrary dimensions. A generalization of this approach (from binary pictures to real valued pictures) is discussed in (Felzenszwalb and Huttenlocher 2004). Values per element represent functions which can express special features (e.g., edges). Those values may be computed in a preprocessing step. In fact, this only adds one more independent dimension to the computations.

4.2 Reverse distance transform

The reverse distance transform considers a set of values associated with their coordinates as a result of a distance transform. It recovers the original binary picture if a minimum subset of elements associated with the distance transform value is given.

The reverse distance transform, for an element p and a given distance value $t(p) > 0$, identifies a finite number of elements q_l, $0 \leq l < m$, which have a distance to p smaller than $t(p)$ as object elements. In other words, all pixels or voxels closer than $t(p)$ to p belong to the same connected component as p in the picture.

Now we consider a finite set of elements $S = \{p_k, 0 \leq k < m\}$ and we define the reverse distance transform [see for example (Coeurjolly n.d.)]:

$$t_r(S) = \{q_l : \exists k, t(p_k) - d(p_k, q_l) > 0\} \tag{4.45}$$

An element q_l belongs to $t_r(S)$ if there exists a pixel or voxel $p_k \in S$ such that the distance between those pixels or voxels is smaller than $t(p_k)$. Equation (4.45) is equivalent to the following [see (Saito and Toriwaki 1994)]:

$$t_r(S) = \{q_l : \max(t(p_k) - d(p_k, q_l)) > 0\} \tag{4.46}$$

If we replace $t(p_k) = t(x, y)$, $q = (i, j)$, and we consider the squared Euclidean distance, then we derive:

$$t_r(S) = \{(i, j) : \max(t(x, y) - (x - i)^2 - (y - j)^2) > 0\} \tag{4.47}$$

Note that $t(x, y)$ is a given integer value, and value $\max(x - i)^2$ is (for a fixed row) independent from j. We can calculate $t_r(S)$ in independent steps for each dimension, in a similar way as for the distance transform. The row scans (y fixed) compute $f_1(x, y)$, and the algorithm stores all non-negative values in an array:

$$f_1(x, y) = \max\{t(i, y) - (x - i)^2 : i = 1, ..., n\} \tag{4.48}$$

The column scans (x fixed) compute $f_2(x, y)$, and the algorithm stores all non-negative values in an array. All resulting pixels or voxels, associated with positive values, belong to the objects in the original picture before the distance transform

$$f_2(x, y) = max\{f_1(x, j) - (y - j)^2 : j = 1, ..., n\} \tag{4.49}$$

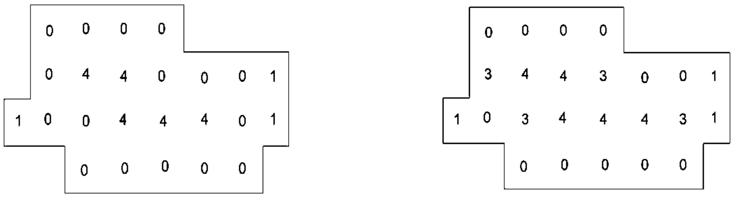
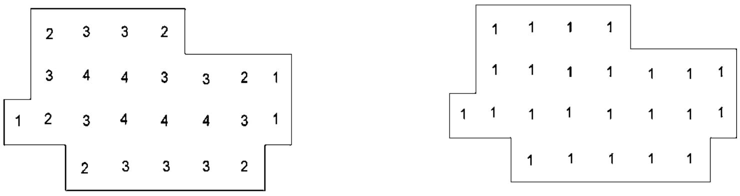


Image T

Result after row scan



Result after column scan

Recovered binary image

Figure 4.5: *Calculation of the reverse distance transform.*

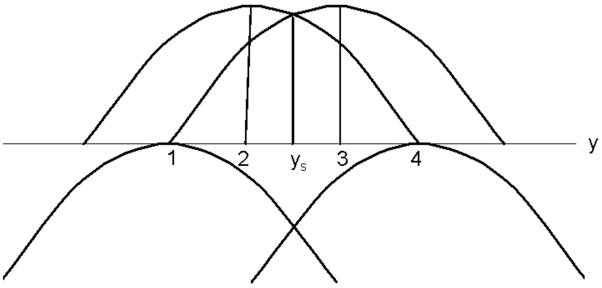


Figure 4.6: *Family of parabolas for one column.*

was carried out. Figure 4.5 shows an illustration of the algorithm for a simple 2D example.

We note that we restore the original object, but we do not receive the values of $t(p)$ in elements p. An optimal implementation is based on the same ideas as we have used for the distance transform. The geometric interpretation, that illustrates the idea, can be given in the following way: for a fixed column (i.e., x is constant, $f_1(x, y) = g(y)$) and a fixed row, any of the equations

$$\gamma_y(j) = (g(y))^2 - (y - j)^2, \; j = 1, ..., n \tag{4.50}$$

represents a single parabola with a local maxima at $y = j$.

Equation (4.50) defines thus a family of n parabolas. The upper envelope of this family of parabolas represents the reverse distance transform at integer coordinates. The algorithm calculates the upper envelope of those parabolas and then it determines the values for each element in an additional scan. The parabolas in Figure 4.6 correspond to the fourth column in Figure 4.5 after row scans.

In the sketch in Figure 4.6, the upper envelope consists of two segments with intersection at y_s. Parabolas γ_1 and γ_4 do not contribute to the upper envelope. We use the first parabola in the upper envelope to compute the values $\gamma_2(1) = 3$ and $\gamma_2(2) = 4$, and use the second parabola for the remaining values $\gamma_3(3) = 4$ and $\gamma_3(4) = 3$.

Let $y_1 < y_2$. It is straightforward to compute y_s as follows:

$$y_s = \frac{(g(y_2))^2 - (g(y_1))^2 + y_1^2 - y_2^2}{2(y_1 - y_2)} \tag{4.51}$$

The computation of the upper envelope can be done in a similar way as for the lower envelope. The cost for the upper envelope extraction is $\mathcal{O}(n)$. The split of a multi-dimensional transform into a "product" of 1D transforms, each with linear complexity, results into a total cost of $\mathcal{O}(n^k)$, for k dimensions.

4.3 Medial axis extraction

In Chapter 3 we introduced the mathematic model for skeletons in a closed subset of the Euclidean space. We defined the medial axis A as a set of center points of maximal disks (or balls).

Exact definitions for the digital space are not consistent in the literature. Related methods for determining connected arcs or curves are (normally) divided into two processing steps. Elements of the medial axis are extracted from a given distance map. Those sets are in general not connected. For this reason, a second step connects the elements of the medial axis to create arcs or curves. Common informal definitions for elements of A can be summarized as follows:

1. A is the smallest number of elements which is required to reconstruct the original picture.

2. A is the set of local maxima of a given distance map.

3. A is the set of centers of maximal disks or balls which is equivalent to the set of symmetric elements.

4. A is a 1D curve located at "the center" of objects in the picture.

Precise implementations of those definitions deliver different sets of skeletal pixels or voxels. A broad variation of published topologic thinning algorithms generates 1D curves. The location cannot be always exactly at central positions because there are cases where the center (defined in real coordinates) is actually located between grid points.

We discuss topologic skeletons in the next chapter. The first informal definition above does not deliver unique results. As we discussed already in Chapter 3, the second and third definition are equivalent if Equation (3.1) is valid. In those cases algorithms evaluate local neighborhoods for extracting skeletal elements. For example, a precise definition of medial axis for the d_4-metric is given in (Rosenfeld and Pfaltz 1966). A picture $T^* = t^*(x, y)$ is called medial axis picture of T if T is a result of a distance transform and

$$\begin{aligned} &t^*(x,y) = t(x,y) \text{ if } t(x-1,y), t(x+1,y), t(x,y-1), t(x,y+1) < t(x,y)+1 \\ &t^*(x,y) = 0 \text{ otherwise} \end{aligned} \tag{4.52}$$

All pixels with $t^*(x, y) \neq 0$ belong to the medial axis. The condition for identifying points of the medial axis is equivalent to $t(q) < t(p) + 1$, for all $q \in A_4(p)$, which corresponds to Equation (3.1). Local maxima coincide with centers of maximal disks for the d_4-metric.

The medial axis picture T^* is easy to compute in a third pass. (Rosenfeld and Pfaltz 1966) proved that the original picture can be reconstructed from the coordinates and values of pixels with $t^*(x, y) \neq 0$ in the distance skeleton picture. This principle can be applied in 3D for weighted distance metrics. Then an element p belongs to the medial axis if $t(q) < t(p) + c$, for all $q \in A_{26}(p)$, where c is the weight for the corresponding adjacency type.

For the squared Euclidean distance or the Euclidean distance, the centers $p \in M$ of maximal disks (or balls) do not correspond to local maxima in the distance transform picture of $\langle P \rangle$ in general. Examining local maxima in a distance map does not deliver exactly centers of maximal disks. The result of the squared Euclidean distance transform in digital space is, for each object pixel or voxel p, an integer value $t(p)$ which represents the radius $r = \sqrt{t(p)}$ of the maximal disk (or ball) centered at p. The reverse distance transform delivers all elements which are located on the disk (or inside the ball), for given centers p and $t(p)$. An algorithm for the extraction of elements of the medial axis decides if a given disk (or ball) is included in a different disk (or ball). If this is not the case then the given disk is maximal, and it belongs to the set of medial axis elements. These inclusion tests can be costly and difficult [see(Ge and Fitzpatrick 1996)]. Some authors [see, e.g., (Remy and Thiel 2005)] use look-up tables to implement the tests.

Let all nonzero values in picture T (see Figure 4.5) represent the set A of skeletal pixels. Set A is not the smallest set of pixels to recover the original picture. If the 4 in the fifth column would be 0, then the object is still reconstructible. Finding the minimum number of skeletal pixels or voxels is important for picture compression, but less interesting for the process of generating arcs and curves.

In (Toriwaki and Mori 2001, Coeurjolly and Montanvert 2005), an exact definition is based on 2D elliptic paraboloids. The following equation is an expression for an elliptic paraboloid, with center $p = (x, y)$ and height $t(x, y)$:

$$0 \leq z < t(x, y) - (x - i)^2 - (y - j)^2 \qquad (4.53)$$

The intersection of the paraboloid and the plane $z = 0$ is a disk with radius $\sqrt{t(p)}$. In continuous space, maximal disks and maximal elliptic paraboloids (not entirely contained in a different paraboloid) are equivalent.

Given is the distance map $T = \{t(p) : p \in M\}$ for d_e^2 in an arbitrary dimension k, for picture P and $q \in M$, $r_i \in M$, and $1 \leq i \leq m$.

14. DEFINITION. *A set A of elements of the medial axis of M is defined by:*

$$A = \{p : \exists q, [d_e^2(p,q) < t(p) \wedge \max_i\{t(r_i) - d_e^2(r_i,q)\} = t(p) - d_e^2(p,q)]\} \tag{4.54}$$

For two dimensions, elements (x,y) of a distance map T, with $t(x,y) > 0$, belong to A if there exists an element $(i,j) \in \mathbb{Z}^2$ with the following properties:

1. $(x-i)^2 + (y-j)^2 < t(x,y)$ and
2. $\max_{(u,v)}\{t(u,v) - (u-i)^2 - (v-j)^2\} = t(x,y) - (x-i)^2 - (y-j)^2$

In (Coeurjolly and Montanvert 2005) the authors have proved that A is a subset of the medial axis for the continuous case, and the original object can be reconstructed from A. Geometrically, the maximum calculation represents the upper envelope of a family of elliptic paraboloids in 2D, or a family of parabolas for each dimension as described for the reverse distance transform. If there are points in the upper envelope of elliptic paraboloids which coincide with points on the elliptic paraboloid with height $t(x,y)$ and center (x,y), then $p = (x,y)$ belongs to A.

The first condition ensures that all elements (i,j) are located on the disk with radius $\sqrt{t(x,y)}$. This definition leads to a large number of elements in A which do not contribute to the representation of shapes of objects (see, e.g., the sphere and its set A in Figure 4.7).

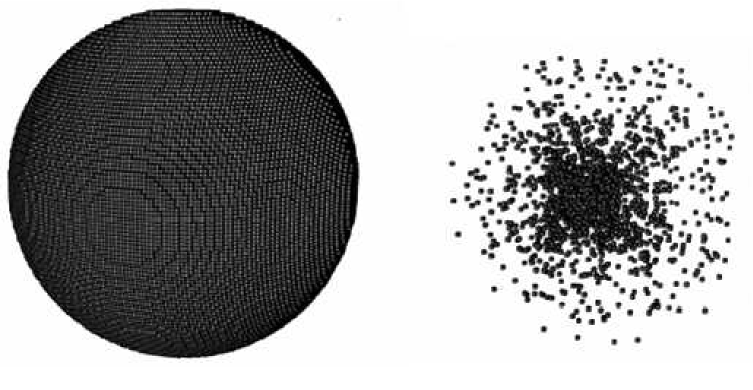


Figure 4.7: *The medial axis of a sphere.*

We apply a second definition to identify elements of set A. Only those elements (x,y) belong to A where local maxima of the upper envelope of the family of elliptic paraboloids coincide with $t(x,y)$.

15. DEFINITION. *Elements (x, y) of a distance map T with $t(x, y) > 0$ belong to A if there exists an element (i, j) with the following properties:*

1. $(i, j) \in \mathbb{Z}^2, (x-i)^2 + (y-j)^2 < t(x, y)$ *and*
2. $\max_{(u,v)}\{t(u, v) - (u-i)^2 - (v-j)^2\} = t(x, y)$

To extract elements of the medial axis for the discrete case we can use the same algorithm. We compute the upper envelope of u parabolas ($1 \leq u \leq n$) for fixed rows v one by one:

$$\gamma_u(i) = t(u, v) - (u-i)^2,\ 1 \leq i \leq n \tag{4.55}$$

For example, we consider the family of nine parabolas as shown in the sketch of Figure 4.8 for $v = 2$. Parabola $\gamma_1(i) \leq 0$, for all i, is not associated with an object element. Parabola $\gamma_2(i)$ is entirely included in parabola $\gamma_3(i)$ and it does not contribute to the upper envelope of the family of parabolas. The value of parabola $\gamma_3(3) = 4$ is a local maxima in the upper envelope, and it is equal to the value $t(3, 2)$ in the distance map. Pixel $p = (3, 2)$ is labeled. Note that the computation of parabolas for non-object elements is unnecessary.

The next step computes the upper envelope for the family of parabolas for fixed columns u as follows:

$$\gamma_v(j) = t(u, v) - (v-j)^2,\ 1 \leq j \leq n \tag{4.56}$$

We identify all pixels p, where $t(p)$ corresponds to local maxima on the upper envelope and where the pixel has been marked during the row scan. For example,

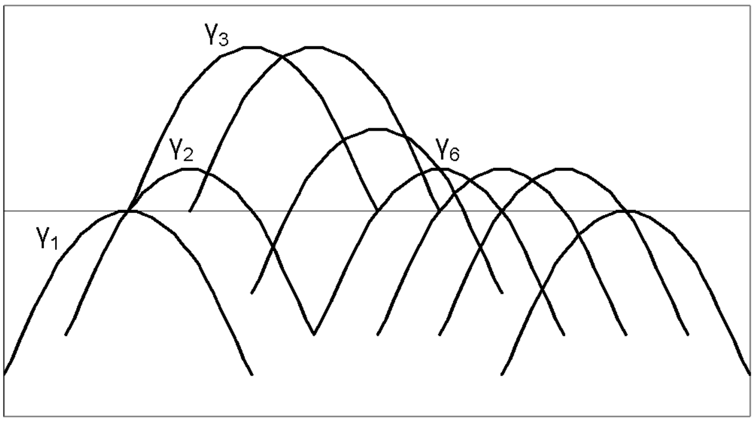


Figure 4.8: *Family of parabolas for row [0,1,4,4,2,1,1,1,0].*

$\gamma_2(2) = 4$ (for $u = 3$) is a local maxima in the upper envelope, and it is equal to the value at pixel $p = (3, 2)$ which has a label. Element $p = (3, 2)$ belongs to the set A.

We prove a proposition (derived in this thesis) to justify the selection of elements of the medial axis.

2. PROPOSITION. *Let* $\max_u\{t(u, v) - (u - i)^2\} = env_r(i, v)$*, for a fixed row* v*, and* $\max_v\{env_r(u, v) - (v - j)^2\} = env_c(u, j)$*, for a fixed column* u*. A pixel* $p = (u, v)$ *is an element of the medial axis in a 2D discrete space (i.e.,* $p \in A$*) if* $env_r(i, v) = t(u, v)$*, for* $1 \leq i \leq n$*, and* $env_c(u, j) = t(u, v)$*, for* $1 \leq j \leq n$*.*

Proof. If $env_r(i, v) = t(u, v)$, for $1 \leq i \leq n \wedge env_c(u, j) = t(u, v), 1 \leq j \leq n$, then $u = i$ and $v = j$ per definition. We consider (u, v) as center of an elliptic paraboloid $\mathcal{P}$ with height $t(u, v)$ and show that this paraboloid is maximal.

We assume that $\mathcal{P}$ is not maximal. Then there exists an elliptic paraboloid such that $\mathcal{P}$ is entirely covered by another paraboloid. Then there exist two parabolas $\gamma_u(i) > t(u, v)$, for a fixed row v, and $\gamma_v(j) > t(u, v)$, for a fixed column u. This is a contradiction. □

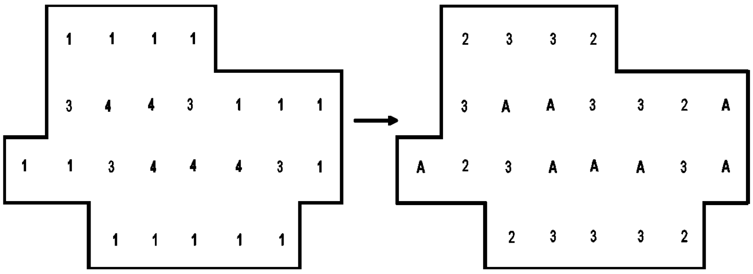


Figure 4.9: *Left: Bold numbers are marked after row scan. Right: Elements of medial axis are labeled with* ***A****.*

We consider the example in Figure 4.2. In Figure 4.9 on the left, numbers are values of the upper envelops of parabolas, row by row, and candidates for the medial axis are bold. The numbers on the right represent values of the upper envelopes of parabolas column by column and the A's represent elements of the medial axis.

We discuss some observations with respect to the second definition for a special example. We consider a digitized disk.

In Figure 4.10, all disks have the same size. Both drawings on the left show the squared Euclidean distance values. The drawings on the right show the result of the reverse distance transform. The grid constant c_B of disk B is half of the size of the grid constant of A (i.e., $c_B = c_A/2$). The geometric center of disk A coincides

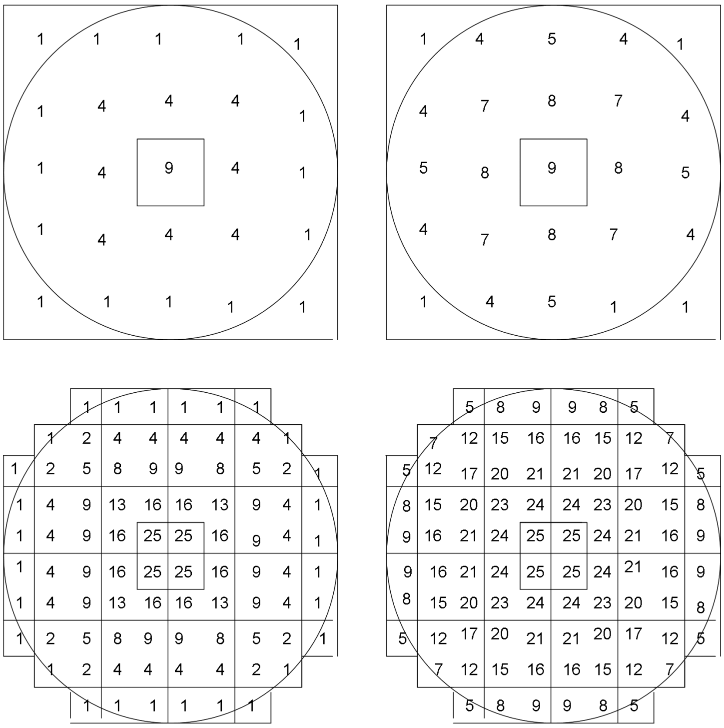


Figure 4.10: *Upper left: d_e^2-distance transform of disk A. Upper right: reverse distance transform of A. Lower right: d_e^2-distance transform of disk B. Lower right: reverse distance transform of B.*

with the center of one 2-cell. This 2-cell and the four border elements with value 1 in the distance map and in the reverse distance map constitute the medial axis according to the definition. The geometric center of disk B coincides with a 0-cell in the frontier of four 2-cells. Only these four 2-cells constitute the medial axis of disk B (no border elements are in the set of medial axis elements). For the same disk with a different grid resolution, the results are different in terms of the numbers of elements of the medial axis and the location of those elements. However, considering the real location, the 2-cell of the medial axis of disk A in the center coincides with the union of the four 2-cells of the medial axis of disk B. If we continue to increase the grid resolution by reducing the size of the grid constant (i.e., $c_C = c_B/2$) then the medial axis of disk C consists of the union of four 2-cells of half of the previous size. If the grid constant goes to 0 then the medial axis of the disk converges towards one real point, the geometric center of the disk in continuous space.

If the number of object cells in one row or in one column is even then the medial axis in continuous space is between adjacent 2-cells in the digital space.

The example shows as well that the reverse distance transform does not recover the values of the distance map, because the values of the reverse distance map are the ordinates of the upper envelope of families of parabolas for integer coordinates. However, all cells (apart from border elements) which have equal values in the distance map and in the reverse distance map constitute the medial axis. They are sufficient to recover the whole disks, because only reverse distance transform values larger than 0 belong to the medial axis.

The medial axis of a digital sphere based on the second definition consists of one element if the geometric center coincides with the center of one element. In all other cases, the medial axis of a sphere is a connected component of elements closely located to the geometric center.

Analogously to the discussions for the calculation of the distance transform and the reverse distance transform, we can process dimensions, one by one, for the extraction of the skeletal elements. The complexity of the algorithm is $\mathcal{O}(n^k)$ for k dimensions, analogously to the reverse distance transform.

(Hesselink et al. 2005) propose a new approach to calculate the medial axis transform. They introduce the so called integer medial axis transform (IMA). They use the concept of EVA-algorithms to compute the feature transform. The feature transform assigns to each object element the closest non-object elements. An object element belongs to the IMA if the distance between its first closest feature and the first closest feature of its adjacent elements is larger than a given value (normally 1). The authors compare some properties of IMA with properties of the medial axis transform defined as sets of centers of maximal disks. Elements of the integer medial axis are connected under certain conditions. More work needs to be done to understand topological properties of IMA.

4.4 The eccentricity transform

As an alternative to distance transforms, the eccentricity[1] transform of components in 2D images based on its adjacency graphs has been discussed in (Kropatsch et al. 2006) with the goal to explore properties of this transform for shape simplification. The eccentricity transform assigns to each object element p in a component B the lengths of the longest shortest α-path to any object element q in B. For example, in a digitized disk, elements with the maximum value of the d_4-distance map coincide

---

[1] Eccentricity is defined in graph theory, identifying the radius and the center of a finite connected graph.

with central vertices of the adjacency graph where the d_4 metric is used.

				A	1	1	1	A	A	A	A	A
				1	2	2	2	1				
				1	2	A	2	1				
				1	2	2	2	1				
A	A	A	A	A	1	1	1	A				

Figure 4.11: *d_4-distance map, local maxima labeled with A.*

				C	9	10	11	12	13	14	15	16
				9	C	9	10	11				
				10	9	C	9	10				
				11	10	9	C	9				
16	15	14	13	12	11	10	9	C				

Figure 4.12: *Eccentricity map, center vertices are labeled with C.*

In Figure 4.11 local maxima labeled with A are identified as being skeletal elements [Equation (4.52)]. Figure 4.12 shows the same object as in Figure 4.11 where each object pixel is considered to be a vertex of the adjacency graph; each object element is labeled with its eccentricity value (based on d_4-metric). Center vertices are labeled with C. This example shows that the set of center vertices (based on the eccentricity transform) and the set of local maxima of the d_4-distance transform differ in general. The location of the set of all center vertices does not identify the elongated shape of this particular region.

4.5 Generation of curve-like structures

Above definitions for elements of the medial axis based on distance maps generate sets of elements which are not connected. We consider, for example, two disjoint maximal d_E^2-disks or d_E^2-balls D_1 and D_2, associated with radii r_1 and r_2, respectively, which are adjacent. The distance between the elements of medial axis A depends on the sum of the radii $r_1 + r_2$. In our example (see Figure 4.9) a simple condition could exclude all extracted elements which are border pixels (maximal disks of one element, could be noise). The remaining elements of the medial axis would be connected but the original object is not reconstructible anymore.

This simple example shows that the computation of medial axis elements is very noise sensitive. In general, a postprocessing step is required to connect all elements of the medial axis. In 2D, the computation of saddle points is useful to generate connectivity (Niblack et al. 1992). The identification of such points in 3D is expensive. Some authors [e.g., in (Toriwaki and Mori 2001)] use the Euclidean distance transform image as input for a topologic thinning algorithm. Elements of the medial axis are considered to be non-simple.

Strategies adapted from graph theory could be used to connect the elements of the medial axis as well. Such strategies consider all elements of one object as a strongly connected graph and use Euclidean distances between them as weights, allowing to calculate the minimum spanning tree.

Squared Euclidean distances are used as attributes of vertices and edges in neighborhood graphs and dual crack graphs in (Glantz and Kropatsch 2001). A contraction of edges in the crack graph is defined that leads to an extended skeleton graph, and the contraction of the extended skeleton graph leads to an skeleton graph which is a strongly connected graph.

Even semiautomatic approaches have been discussed in the literature. For example, in (Gagvani and Silver 1997) endpoints for skeletal curves are selected manually and a divide-and-conquer recursion generates elements between those endpoints.

The result in Figure 4.9 is not a simple 1D 8-curve because the object includes a region with 3 columns of 4 pixels width. The intersection of the parabolas with positive values is on the edge of two pixels, but per definition all elements of a digital 1D curve are 2-cells. In other words, the resulting skeletal curve is not thin. This problem appears always if the object includes elongated parts with even column or row size. Heuristic approaches to solve this problem can be found in the literature. Normally, they define an additional condition to delete pixels from one direction from the set of skeletal pixels. The result is a union of simple arcs and curves which are not at the center. The requests to ensuring unique reconstruction, or to obtain thin curves are conflicting.

4.6 Summary

This chapter is a discussion about mathematic models for skeletons, and of the extraction of skeletal elements in binary pictures based on distance transforms. Distance transforms are linked to a specified metric. Euclidean distance transforms or squared Euclidean transforms deliver rotation invariant results. Approximations of the Euclidean metric are more time efficient but less accurate. EVA algorithms for the squared Euclidean distance transform, the reverse distance transform and the skeleton extraction are explained in detail. Main advantages in comparison to other algorithms are

- the option to calculate exact Euclidean distance values,
- the multidimensional transform can be split into separate runs for each dimension,
- implementations can be done in linear time per dimension; altogether the complexity is $\mathcal{O}(n^k)$ if n is the number of elements for each dimension and k the number of dimensions.

Skeleton definitions are not consistent with the medial axis definition for continuous space. Extracted skeletal elements are not subsets of the points located on the medial axis in continuous space in general (see Figure 4.10). The main disadvantage is that medial axes are not connected in discrete space. Further postprocessing steps are required to generate curve-like structures.

Chapter 5

Topologic skeletons

This chapter introduces topologic thinning as a general concept for the extraction of skeletons in a digital space. These algorithms iteratively delete simple elements until only non-simple elements or end-elements remain. The notion of simple elements (pixels or voxels) is of basic importance for topologic thinning algorithms. Detecting simple elements is crucial in all approaches where representations of the objects in the picture are topologically equivalent to the original picture. Characterizations of simple elements in two or three dimensions have been an active research area, and a diversity of papers has been published. (Recent publications also discuss methods for detecting simple elements in higher dimensions.)

Common methods are, for example, calculations of characteristic numbers, connected component analysis, template matching, or the computation of Euler characteristics. The construction of look-up tables is based on such characterizations. For higher dimensions, look-up tables are costly; for example, a four-dimensional cube has 2^{80} *possible neighborhood configurations. We review algorithms and some characterizations of simple elements and propose a new methodology for identifying non-simple elements.*

Parts of this chapter are published in (Klette 2003a) and (Klette and Pan 2004).

5.1 Topologic thinning

Thinning has been developed over the past forty years as an important preprocessing technique for picture analysis. The goal of thinning is to change connected components in a given picture P into "topologically equivalent components". We call those components *topologic skeletons*.

In general, thinning aims at finding topologic properties of digital objects which are not related to size or quantity. The results should be independent from the position of a set in the plane or space, grid resolution (for digitizing this set), or the geometric shape complexity of the given set.

In the literature, the term thinning is not used in a unique interpretation besides that it denotes a connectivity preserving reduction operation applied to digital pictures. We understand topologic thinning as an iterative process that changes specified border elements into non-object elements. A subset $M \subseteq P$ of object elements is reduced by a defined set D in one iteration, and the result $M' = M \setminus D$ becomes M for the next iteration.

Topologic thinning preserves the topology of the original picture. Changing the value of simple elements preserves adjacency or surroundedness relations between the connected components of 1's and 0's. For this type of deformations, a bijective mapping exists between the components of the original picture and the components of the deformed picture, and the adjacency relations are the same. Thinning is a one-way simple deformation that reduces the number of object elements.

The result of a thinning algorithm might be defined to be *ideally thin* if no element which is not an end element can be deleted without violating connectivity properties. It can happen that a thinning procedure results in digital arcs which intersect each other, generating *"branching elements"* or *"junctions"*. An ideally thin subset of $\langle P \rangle$ may contain a set of branching elements as in the following example in 2D:

$$\begin{array}{ccccccc}
1 & & & 1 & & & 1 \\
 & 1 & & 1 & & 1 & \\
 & & 1 & 1 & 1 & & \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & 1 & 1 & 1 & & \\
 & 1 & & 1 & & 1 & \\
1 & & & 1 & & & 1
\end{array}$$

Note that this set is not yet a digital arc or curve as defined above. In Chapter 6 we discuss how we can split sets, obtained as results of topologic thinning, into sets of disjoint digital arcs.

5.2 Simple elements

The topology of a picture is defined by its region adjacency tree (Rosenfeld 1974). Two pictures are topologically equivalent if their region adjacency trees are isomorphic (see Chapter 2). The basic notion of a *simple element* is used in topology preserving digital deformations in order to characterize a single element p of a digital picture P which can change the value $P(p)$ without changing the topology of the picture.

16. DEFINITION. *A* simple element *is a single element p of an nD digital picture that can change its value $P(p)$ without altering the topology of the picture.*

This definition is general for any dimension, and not useful for the identification of simple elements in thinning algorithms. Ways to characterize simple elements are discussed in topology and in picture processing papers. Especially in the 3D space it is important to find efficient ways of identifying simple voxels as part of algorithms that determine “topologic skeletons” (Kong 1995, Lohou and Bertrand 2002).

Some applications require even 4D spaces by assigning time to the fourth dimension (e.g., 3D picture sequences as in studies of the moving heart). The complexity of algorithms increases with each dimension. Simple elements are defined in any finite dimension; examples in this work are restricted to three dimensions focused on applications in 3D confocal microscope picture processing.

Simple elements have in common that changes from 1 to 0 (or vice versa) preserve the adjacency or surroundedness relations between connected components of 1's and 0's of the picture. There are different ways to define this important property. Some of the characterizations in the literature are abstract such as Kong's definition (Kong 1995, Fourey and Malgouyres 2003) using homotopy equivalence in order to include pixels of 3D or higher dimensional pictures. However for the design of algorithms it is necessary to find simple and time efficient ways to determine whether a given pixel of an picture P is simple in P. Different theoretic approaches and notions are used to describe simple elements in a large volume of publications and we prove that some of them are actually equivalent. We review a few characterizations which are easy to compute, or easy to visualize for 2D and 3D digital pictures.

5.2.1 Equivalent characterizations of simple elements in 2D

We use the grid point model and we define a simple 1 following (Rosenfeld et al. 1998) using a *good pair* α, α': if $\alpha = 4$ then $\alpha' = 8$, and if $\alpha = 8$ then $\alpha' = 4$.

17. DEFINITION. *A 1 of a picture P is called α-simple ($\alpha \in \{4, 8\}$) if it is α-adjacent to exactly one α-component of 1's in $A_8(p)$ and it is α'-adjacent to exactly one α'-component of 0's in $A_8(p)$.*

If p is a border pixel then this definition simplifies: $(p, P(p))$ is α-simple in P iff p is α-adjacent to exactly one α-component of $P(p)$ in $A_8(p)$. Note that a 4-simple pixel p can be 4-adjacent to exactly one 4-component of 1's in $A_8(p)$ and 8-adjacent to different 4-components of 1's in $A_8(p)$. The example below shows an element p which is either a 4-simple 1, or an 8-simple 0:

$$\begin{matrix} 1 & 1 & 1 \\ 0 & p & 0 \\ 1 & 0 & 1 \end{matrix}$$

We consider both "atomic operations", changing a 1 into 0, or vice versa. Changing an α-simple element p of an α-component U results into a non-empty α-component $U \setminus \{p\}$ and a non-empty α'-component $V \cup \{p\}$, and the adjacency relations to all other components remain the same. The rooted region adjacency trees of the original picture and of the resulting picture are isomorphic.

We extend the above definition: A pixel $(p, P(p))$ is called *α-simple* if it is α-adjacent to exactly one α-component in $A_8(p)$ and it is α'-adjacent to exactly one α'-component in $A_8(p)$. Changing the value of a single α-simple pixel p results into an α'-simple pixel.

A value change of a simple pixel delivers a topologically equivalent picture (Rosenfeld et al. 1998). Two pictures differ by *simple deformation* if one can be obtained from the other one by repeatedly changing simple pixels from 1 to 0, or vice versa. Changing simple pixels from 1 to 0, or vice versa is a two-way simple deformation. Thinning or shrinking procedures are one-way simple deformations. They are typically only formulated for changes of object elements to non-object elements.

Characterizations of simple pixel apply different assumptions (i.e., 8-connectivity is used for the 1's, and 4-connectivity for the 0's). Simple pixels are often restricted to simple 1's. We review a few earlier characterizations of simple 1's, but we consider general properties of simple pixels in order to use them for two-way simple deformations.

1. CHARACTERIZATION. *A 1 of a picture P is 4-simple in P iff $X_R(p) = 2$.*

Changing 1 to 0 of a picture P preserves 4-connectivity of P if there is exactly one change from 0 to 1 and exactly one change from 1 to 0 in $A_8(p)$. In $A_8(p)$ is exactly one 4-component of 1's and exactly one 4-component of 0's. This is a restriction compared to 4-simple pixels based on the above definition.

(Hilditch 1969) defined an 8-simple 1 as follows:

2. CHARACTERIZATION. *A 1 of a picture P is 8-simple in P iff $X_H(p) = 1$.*

This characterization is equivalent to the characterization in (Hall 1996) where a 1 is 8-simple iff there is exactly one distinct 8-component of 1's in $A_8(p)$ and p is a border 1.

The following well-known characterization of simple pixels (Rosenfeld 1970) is equivalent to Characterization 2.

3. CHARACTERIZATION. *A 1 of a picture P is 8-simple in P iff both of the following conditions hold: The union of all pixels in $P \setminus \{p\}$ that is 8-adjacent to p is nonempty and connected. p is 4-adjacent to a 0.*

(Kong 1995) used the concept of an P-attachment set of a pixel $(p, P(p))$, which is formulated for the grid cell model.

4. CHARACTERIZATION. *A* 1 *at* p *of a picture* P *is* 0*-simple in* P *iff the* P*-attachment set of* p*, and the complement of that set in the frontier of* p *are non-empty and connected.*

We can simplify this for 2D pictures as follows:

5. CHARACTERIZATION. *A* 1 *at* p *of a picture* P *is* 0*-simple in* P *iff the* P*-attachment set of* p *is non-empty, connected, but not the entire frontier of the 2-cell* p*.*

(Kong 1995) shows that the last two characterizations are equivalent for 2D pictures. In order to determine whether a pixel is simple for two-way simple processes we show (Klette 2002):

2. THEOREM. *A* 1 *(or* 0*) of a picture* P *is 4-simple iff* $X_Y(p) = 1$ *(or* $\overline{X}_Y(p) = 1$*). A* 1 *(or* 0*) of a picture* P *is 8-simple iff* $\overline{X}_Y(p) = 1$ *(or* $X_Y(p) = 1$*).*

Proof. Let p be a 4-simple 1. First we assume that $X_Y(p) = 0$ (i.e. $a_i = 0$, for all $1 \leq i \leq 4$). Per definition we have $a_i = P(q_{2i-1}) - P(q_{2i-1}) * P(q_{2i}) * P(q_{2i+1})$. It follows that $a_i = 0$ iff $P(q_{2*i-1}) = 0$ or $P(q_{2*i-1}) = P(q_{2*i}) = P(q_{2*i+1}) = 1$. In the case that $P(q_{2*i-1}) = 0$, for all $1 \leq i \leq 4$, there is a contradiction to the property of having exactly one 4-adjacent 4-component of 1's. In case $P(q_{2*i-1}) = P(q_{2*i}) = P(q_{2*i+1}) = 1$, for all $1 \leq i \leq 4$, this is a contradiction to the property of having exactly one 8-component of 0's in $A_8(p)$.

Now assume $X_Y(p) > 1$. Then there exist at least two a_i's, with $a_i = a_j = 1$, for $i \neq j, 1 \leq i, j \leq 4$. It follows that there are at least two 4-adjacent 4-components of 1's and two 8-adjacent 8-components of 0's, which is a contradiction to the assumption that p is a 4-simple 1. It follows that $X_Y(p) = 1$.

Let $X_Y(p) = 1$ (i.e., one $a_i = 1$, say $a_1 = 1$ and $a_j = 0$, for all $2 \leq j \leq 4$). $a_1 = 1$ iff $P(q_1) = 1$ and $P(q_2) = 0$ or $P(q_3) = 0$. It follows that at least one 4-adjacent 4-component of 1's and at least one 8-adjacent 8-component of 0's exist. For all other terms is $a_j = 0$, for all $2 \leq j \leq 4$, that means $P(q_{2*j-1}) = 0$ or $P(q_{2*j-1}) = P(q_{2*j}) = P(q_{2*j+1}) = 1$.

$P(q_{2*j-1}) = 0$, for all $2 \leq j \leq 4$. It follows that there is exactly one 4-adjacent 4-component of 1's, and that there is exactly one 8-adjacent 8-component of 0's. In case $P(q_{2*j-1}) = P(q_{2*j}) = P(q_{2*j+1}) = 1$, for all $2 \leq j \leq 4$, it follows that $P(q_2) = 0$ is the only 8-adjacent 8-component of 0's. Now we consider $P(q_{2*j-1}) = 0$ and $P(q_{2*j+1}) = 1$, for all $j = 2, 3$ then $P(q_{2*j+1})$ is always 4-connected to $P(q_1) = 1$. The value of $P(q_{2*j})$ can be 0 or 1, it doesn't change the number of 4-adjacent 4-components of 1's and the number of 8-adjacent 8-component of 0's. In each case there is exactly one 4-adjacent 4-component of $1'$s and exactly one 8-adjacent 8-component of $0'$s. All other cases follow by symmetry.

Let p be an 8-simple 1. There is exactly one 4-adjacent 4-component of $0'$s and exactly one 8-adjacent 8-component of $1'$s. Based on the definition of $\overline{X_Y(p)}$ the proof is analogous. The proof for 4-(8)-simple $0'$s is analogous to the given proof for simple $1'$s. □

It follows that a pixel $(p, 1)$ is 4-simple in P iff $(p, 0)$ is 8-simple in P and a pixel $(p, 1)$ is 8-simple in P iff $(p, 0)$ is 4-simple in P. It follows as well that $(p, 0)$ is 8-simple in P iff $(p, 1)$ is 8-simple in $\overline{P}$ where $\overline{P}$ is the negative picture of P (all $1'$s in P are $0'$s in $\overline{P}$ and vice versa). The good pair concept for binary pictures (use of 8-connectivity for object components and 4-connectivity for non-object components or vice versa) provides this property of duality for simple pixels. For simple deformations the use of binary pictures is verified by applying the characteristic function after segmentation of grey value input pictures. The application of a sequential thinning algorithm (algorithm A in Chapter 5.4) on $\overline{P}$, for example, delivers a topologically equivalent picture to the original picture. The skeleton (see Figure 5.1) located in the non-object region is a representation of the original object. The skeleton picture is topologically equivalent to the original picture. It is not a magnification of the object by a certain scale factor because the non-object region is limited (the size of a digital picture is always finite) and the algorithm changes values based on conditions which are restricted to topology preservation and end-element stability.

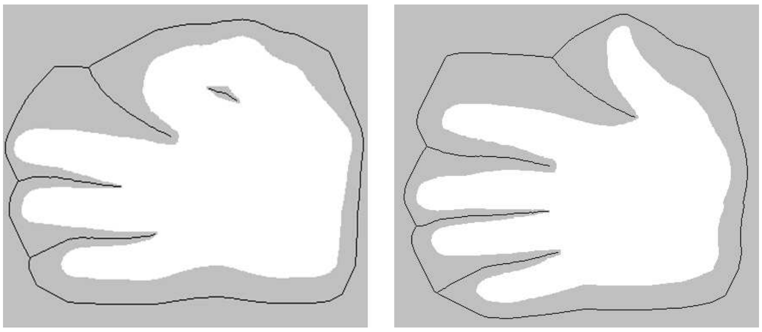


Figure 5.1: *Result of algorithm A applied on* $\overline{P}$.

Note that for a 1 at p of a picture P the crossing number $X_H(p)$ is always equal to $\overline{X}_Y(p)$. The characterization of an 8-simple 1 in the above theorem is equivalent to

Characterization 2. It follows that, if $X_H(p) = 1$, then p is an 8-simple 1 or a 4-simple 0 depending on the value $P(p) = 1(0)$.

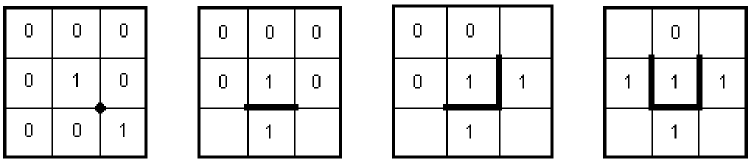
The following theorem shows that characterizations for 8-simple 1′s based on P-attachment sets are equivalent to Characterization 2 [see (Klette 2003a)].

3. THEOREM. *Let p be a 1 of a picture P. Then the P-attachment set of p is non-empty and connected and not the entire frontier of p iff $X_H(p) = 1$.*

Proof. Let p be 8-simple. Then we only have four ways in which the P-attachment set is non-empty and connected and not the entire frontier, in each case $X_H(p) = 1$. Let $X_H(p) = 1$. At least one 8-neighbor is a 1 per definition. It follows the P-attachment set is non-empty. At least one 4-neighbor is a 0 per definition, it follows the P-attachment set is not the entire frontier. It remains to show that the P-attachment set is connected. Assume the P- attachment set is not connected then $X_H(p) > 1$ which is a contradiction. □

The following theorem justifies the choice of masks for window matching algorithms.

4. THEOREM. *Let p be a 1 of a picture P. Per Characterization 5, p is 8-simple in P iff the neighborhood of p matches one of the following masks (simple element masks 1 to 4, from left to right, empty squares can be either 0 or 1)*



or one of their 90° rotations.

Proof. These masks represent the only four ways in which the P-attachment set is non-empty and connected and not the entire frontier. □

Previously published thinning algorithms have used Characterization 1 in order to preserve 4-connected subsets of the original picture. The following example shows two 4-components of 1′s that are 8-adjacent:

$$\begin{matrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix}$$

Based on Characterization 1, the central pixel p would be identified as a 4-simple 1 because $X_R(p) = 2$. If we would only use this criterion then a 4-component of a single 1, that is 8-adjacent to a component of 1's, would completely disappear. An algorithm based on this condition determines pixels as simple which have exactly one 4-component of 1's and exactly one 4-component of 0's in $A_8(p)$. The 4-component of 1's can be 4-adjacent or 8-adjacent to p. For $(\alpha, \alpha') = (8, 4)$ and $(\alpha, \alpha') = (4, 8)$ a thinning algorithm using Characterization 1 for connectivity preservation results in a 4-connected subset of the original 4-component. A 4-component disappears if it consists of a single pixel and is 8-adjacent to a disjoint 4-component.

Now let us consider the good pair $(\alpha, \alpha') = (4, 8)$. In case that two disjoint 4-components of 1's are 8-adjacent to each other, then there exists a 1 which is 4-simple and $X_R(p) \neq 2$. Evidently this 1 is not 4-simple based on Characterization 1. To avoid this conflict (Latecki and Eckardt 1995) introduced the concept of *well-composedness*. The following theorem includes an equivalent condition to well-composedness that ensures that the described critical configurations cannot occur [see (Klette 2003a)].

5. THEOREM. *Let all disjoint 4-components of 1's of a picture P be pairwise 8-separated by 0's to each other. Then a 1 is 4-simple in P iff $X_R(p) = 2$.*

Proof. Let $X_Y(p) = 1$ (p with $P(p) = 1$ is 4-simple), and any 8-path from a 1 of a 4-component to a disjoint 4-component of 1's must intersect a 0 (all disjoint 4-components of 1's in P are 8-separated by 0's). Then it follows that we have exactly one 4-adjacent 4-component of 1's, and exactly one 4-component of 0's in $A_8(p)$ and $X_R(p) = 2$.

Now we consider $X_R(p) = 2$ and $X_Y(p) = 0$. The only possible configuration would be $P(q_2) = 1$ and all other pixel values in $A_8(p)$ are 0 (and all symmetric cases). This is a contradiction to our precondition.

Let us consider $X_R(p) = 2$ and $X_Y(p) > 1$. A configuration for this condition does not exist. It follows that $X_Y(p) = 1$. $\square$

5.2.2 Characterizations of simple elements in higher dimensions

The literature offers a long list of definitions of simple elements in 3D pictures. The following questions arise: Can we find analogous characterizations for 3D pictures based on concepts used for the 2D case? Which characterizations are efficient to determine simple voxels in 3D? Are these existing characterizations equivalent to one-another? – First we review some definitions. In 1994, two independent publications (Bertrand and Malandain 1994, Saha and Chaudhuri 1994) proposed the following:

6. CHARACTERIZATION. *A voxel $(p, P(p))$ of a picture P is a 26-simple grid element iff it is 26-adjacent to exactly one 26-component of voxels in $A_{26}(p)$ and it is 6-adjacent to exactly one distinct 6-component of voxels in $A_{18}(p)$.*

A voxel $(p, P(p))$ of a picture P is a 6-simple grid element iff it is 6-adjacent to exactly one 6-component of voxels in $A_{18}(p)$ and it is 26-adjacent to exactly one distinct 26-component of voxels in $A_{26}(p)$.

The calculation of the numbers of components in the 26-neighborhood of p is time consuming for 3D pictures. It can be done by using algorithms derived from graph theory. The number of computations depends on the size of the components.

The following characterization of 26-simple object elements is due to (Malandain and Bertrand 1992, Saha et al. 1991):

7. CHARACTERIZATION. *An object voxel p of a 3D picture P is 26-simple in P iff*

- *(i)* *p is 26-adjacent to another object voxel q, and*
- *(ii)* *p is 6-adjacent to a non-object voxel q', and*
- *(iii)* *the set of object voxels which are 26-adjacent to p is connected, and*
- *(iv)* *every two non-object voxels that are 6-adjacent to p are 6-connected by non-object voxels that share at least one edge with p.*

In terms of topologic numbers we can express this characterization as follows (Bertrand 1994):

8. CHARACTERIZATION. *Let $p \in M$ and $(\alpha, \alpha') \in \{(6, 26), (26, 6)\}$. An object voxel p is α-simple iff $T_\alpha(p, M) = 1$ and $T_{\alpha'}(p, \overline{M}) = 1$.*

The equivalence of the last two characterizations follows from the applied definitions.

In Figure 5.2, the P-attachment set of p in the frontier of p is connected and the complement of the P-attachment set of p in the frontier of p is not connected. Element p is not simple. In terms of the previous characterizations, $T_{\alpha'}(p, \overline{M}) = 2$ or there exists a non-object voxel q that is 6-adjacent to p such that q is not 6-connected to any non-object voxel which shares at least one edge with p.

In the following characterizations we use the cell model. Properties of the P-attachment set of p for 3D pictures are easy to visualize [see, e.g., (Kong 1995)].

9. CHARACTERIZATION. *An object voxel p of a picture P is 0-simple in P iff the P-attachment set of p and the complement of that set in the frontier of p are both non-empty and connected.*

Analogously to the arguments in 2D, we could simplify this statement if p is a border voxel because the P-attachment set of p, and the complement of that set in the frontier of p, are both non-empty in this case. The above characterization is equivalent to the previous characterization for 26-simple voxels (Klette 2003b). As reviewed in Chapter 2, the Euler characteristic $\chi(K_p)$ of a P-attachment set is a useful tool to describe properties of this set.

10. CHARACTERIZATION. *A voxel p is 0-simple iff the P-attachment set K_p of p, and the complement of that set in the frontier of p are connected and $\chi(K_p) = 1$.*

Recent research (Gau and Kong 2003, Fourey and Malgouyres 2003, Niethammer et al. 2005) is often focused on simple element detection in dimensions $d \geq 3$. General definitions and characterizations are based on the fact that an element p is called simple if its change to a non-object element results in an topologically equivalent picture, independent from its dimension. A characterization of simple elements based on the α-fundamental group is given in (Fourey and Malgouyres 2003, Kong 1995).

11. CHARACTERIZATION. *Let $M \subset \mathbb{Z}^d$ and $p \in M$. An object element p is simple iff*

- *(i) M and $M \backslash p$ have the same number of connected components.*
- *(ii) $\bar{M}$ and $\bar{M} \cup p$ have the same number of connected components.*
- *(iii) For each voxel b (b is base element) in $M \backslash p$, the group map $i_* : E^\alpha(M \backslash p, b) \rightarrow E^\alpha(M, b)$ is an isomorphism.*

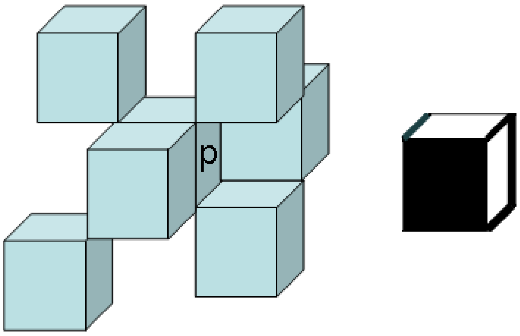


Figure 5.2: *The P-attachment set on the right is not empty, connected, and it is not the entire frontier, but p (center voxel on the left) is not simple.*

The map $i : M\backslash p \rightarrow M$ is called *inclusion map*. It induces the group isomorphism i_*. In (Fourey and Malgouyres 2003) the authors prove that these three properties are sufficient to characterize simple voxels. If a voxel satisfies these three conditions then $T_\alpha(p, M) = 1$ and $T_{\alpha'}(p, \overline{M}) = 1$, and, if $T_\alpha(p, M) = 1$ and $T_{\alpha'}(p, \overline{M}) = 1$, then these three conditions follow as well. In other words: this characterization and the characterization based on topologic numbers are equivalent in 3D space. In addition, from these three conditions it also follows that for each voxel b' in $\bar{M}$, the group map $i'_* : E^{\alpha'}(\bar{M}, b') \rightarrow E^{\alpha'}(\bar{M} \cup p, b')$, induced by the inclusion map $i' : \bar{M} \rightarrow \bar{M} \cup p$, is an isomorphism.

For the purpose of making the decision whether a given element is simple or not, those theoretic results need a "translation" into a form which allows computations.

5.2.3 Non-simple voxels

This section considers characterizations of non-simple voxels. Note that a vertex v (i.e., a 0-cell) of a voxel p shares its point with three 1-cells and three 2-cells in the frontier of p. An edge e (i.e., a 1-cell) shares its points with two 0-cells, four other 1-cells and four 2-cells in the frontier of p. A face f (i.e., a 2-cell) shares its points with four 0-cells, eight 1-cells and four other 2-cells.

Let x be a cell in the frontier of a voxel p. Let $C_x(p)$ be the set of all n-cells ($n \in 0, 1, 2$) in the frontier of p which have a non-empty intersection with x.

We say that a given n-cell x in the P-attachment set K_p of p is *isolated* if $C_x(p) \cap K_p$ = x. We say that a given 2-cell y in the complement of the P-attachment set $\bar{K_p}$ of p is *isolated* if all points in the frontier of y belong to the P-attachment set K_p of p.

If p is 0-simple (see the 10. Characterization) then $\chi(K_p) = 1$. But, if $\chi(K_p) = 1$ then it does not follow that p is 0-simple, because there are two additional conditions. We investigated cases where $\chi(K_p) = 1$ and p is not simple (see Figure 5.3) and we describe 0-non-simple voxels.

Let N_2 be the number of non-object voxels which are 2-*adjacent* to p. Configurations for $N_2 = 6$, where $\chi(K_p) = 1$ and p is not simple, are shown in Figure 5.3. Note that the two cases on the right of the upper row are symmetric. We show:

3. PROPOSITION. *Let $N_2 = 6$. An object voxel p is 0-non-simple and $\chi(K_p) = 1$ iff the P-attachment set K_p consists of two or three disjoint components. For two disjoint components, one component is a simple curve in the Euclidean space, and the other component is a single point or an arc in the Euclidean space. For three disjoint components, one of these components is a non-simple curve in the Euclidean space, and the other components are isolated points in the Euclidean space.*

Proof. Let p be an object voxel. Let $X_i(p), 1 \leq i \leq 3$ be a component of K_p such that $\bigcup_i X_i(p) = K_p$. If $X_i(p)$ is a simple curve in the Euclidean space, then $\chi(X_i(p) = 0$

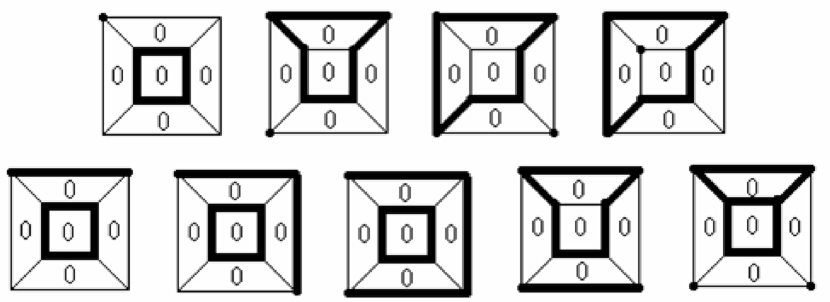


Figure 5.3: *Voxels are not simple, for $\chi(K_p) = 1$ and $N_2 = 6$.*

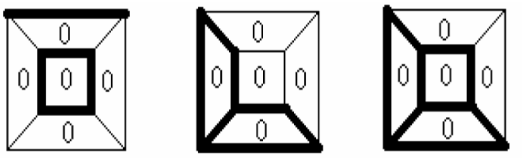


Figure 5.4: *(a) K_p consists of two disjoint sets. X_1 is a simple curve ($\chi(X_1(p)) = 0$, $n_0 = n_1$, $n_2 = 0$) and X_2 is an arc ($\chi(X_2(p)) = 1$, $n_0 - n_1 = 1$, $n_2 = 0$) (b) K_p is a non-simple curve and ($\chi(K_p) = -1$, $n_0 - n_1 = -1$, $n_2 = 0$) (c) K_p is a non-simple curve and ($\chi(K_p) = -2$, $n_0 - n_1 = -2$, $n_2 = 0$)*

($n_0 = n_1, n_2 = 0$) (see Figure 5.4.a). If $X_i(p)$ is an isolated point or an arc in the Euclidean space, then $\chi(X_i(p)) = 1$ ($n_0 - n_1 = 1, n_2 = 0$) (See Figure 5.4.a). If X_i is a non-simple curve in the Euclidean space, then $\chi(X_i(p)) \leq -1$ ($n_0 - n_1 \leq -1, n_2 = 0$) (see Figures 5.4.b and 5.4.c). – We consider the following two cases:

1. We assume that p is not simple and $\chi(K_p) = 1$ and $n_2 = 0$. Based on Characterization 10 we know that the P-attachment set K_p or the complement of K_p ($\bar{K_p}$) are not connected. $\chi(K_p) = 1$ if $\sum_{i=1}^{3} \chi(X_i(p)) = 1$. If K_p consists only of one component then K_p and $\bar{K_p}$ are connected. This is a contradiction to our assumption. One component out of two must have Euler characteristic 1 such that $\chi(X_1(p)) = 1$ and $\chi(X_2(p)) = 0$. $X_1(p)$ can be a 0-cell or an arc. It follows that $X_2(p)$ is a closed curve. For three components we have only the option that $n_0 - n_1 \leq -1$ for one component, and there must be two other components with $\chi(X_i(p)) = 1$. This is only possible if one component constitutes a non-simple curve and the other two are both isolated points.
2. Now we assume that K_p consists of two disjoint components, an arc or a single point $X_1(p)$ and a simple curve $X_2(p)$. Then it follows that $\chi(X_1(p)) = 1$,

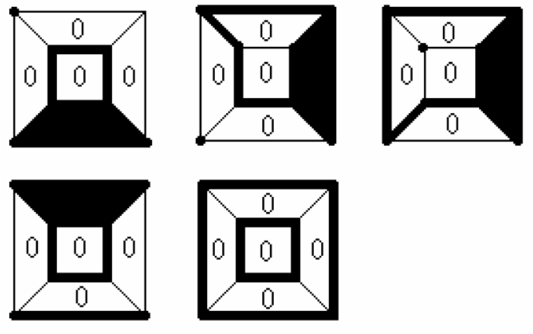


Figure 5.5: *These voxels are non-simple and* $\chi(K_p) = 1$*, for* $N_2 = 5$.

$\chi(X_2(p)) = 0$, $\chi(K_p) = 1$ and p is not simple. If K_p consists of one non-simple curve $X_1(p)$ in the Euclidean space, and one single point $X_2(p)$ and a different single point $X_3(p)$, then it follows that $\chi(X_1(p)) = -1$, $\chi(X_2(p)) = 1$, $\chi(X_3(p)) = 1$, $\chi(K_p) = 1$, and p is not simple.

This concludes the proof of the proposition. □

All configurations shown in Figure 5.3 are non-simple, with $\chi(K_p) = 1$ and $N_2 = 6$. For $N_2 = 5$, non-simple voxels with $\chi(K_p) = 1$ are shown in Figure 5.5. The right-hand case in the upper row is a rotation of the case at the middle of the upper row.

For $N_2 \leq 4$, the Euler characteristic of the configuration in Figure 5.6 (and of all of its rotations) is $\chi(K_p) = 1$, and this voxel is 0-non-simple.

To summarize those findings, we give a new description of non-simple voxels as follows:

3. COROLLARY. *For voxel* p*, let* N_2*, with* $1 < N_2 \leq 6$*, be the number of 2-adjacent non-object voxels, and the number of adjacent object voxels is larger than 1. Voxel* p *is 0-*

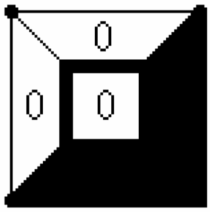


Figure 5.6: *This voxel is not simple and* $\chi(K_p) = 1$*, for* $N_2 = 1, 2, 3, 4$.

non-simple iff $\chi(K_p) \neq 1$*, or* $\chi(K_p) = 1$ *and* K_p *includes an isolated 0-cell or an isolated 1-cell or* $\bar{K_p}$ *includes an isolated 2-cell.*

Proof. We assume that p is 0-non-simple. According to Characterization 10 it follows that $\chi(K_p) \neq 1$ or $\chi(K_p) = 1$ and K_p or $\bar{K_p}$ are not connected. We consider two cases:

1. First, assume that $\chi(K_p) = 1$ and K_p is not connected. Then K_p has at least two components. For $N_2 = 6$ and two components in K_p, we have proved that one component is an isolated 0-cell (a point in the Euclidean space) or a simple arc. If it is a simple arc then the second component must be a simple curve that isolates a 2-cell in $\bar{K_p}$. If it is a simple arc which coincides with one single 1-cell, then this 1-cell is isolated in K_p. For $N_2 = 6$ and three components in K_p, we have shown two of them are isolated 0-cells.

   For $N_2 = 5$ and two components $X_1(p)$ and $X_2(p)$ in K_p, the Euler characteristic of one of those components, say $X_1(p)$ is $\chi(X_1(p)) = n_0 - n_1 + 1$, and for the second component we have $\chi(X_2(p)) = n_0 - n_1 + 0$. If $\chi(X_1(p)) = 1$ then $n_0 = n_1$ which is an isolated 2-cell in K_p, and $\chi(X_2(p)) = 0$ which is a simple curve in K_p and an isolated 2-cell in $\bar{K_p}$ (see the right case in the bottom row in Figure 5.5). If $\chi(X_1(p)) = 0$ then $n_0 = n_1 - 1$ and $\chi(X_2(p)) = 1$ which is an isolated 0-cell or an arc (see for example the left case in the bottom row in Figure 5.5). The arc coincides with only one 1-cell, otherwise it would be connected with X_1. If $\chi(X_1(p)) > 1$ then $n_0 > n_1$ and $\chi(X_2(p)) < 0$. This is impossible, because $X_1(p)$ would be connected with $X_2(p)$.

   For $N_2 = 4$ and two components $X_1(p)$ and $X_2(p)$ in K_p, we have two 2-cells in K_p. If they are opposite then $\chi(K_p) = 2$, $\bar{K_p}$ is connected. If $X_1(p)$ has two 2-cells in K_p and $\chi(X_1(p)) = 1$ then $\chi(X_2(p)) = 0$. Then $X_2(p)$ must be a simple curve, this is impossible because it would be connected with $X_1(p)$. If $X_1(p)$ has two 2-cells in K_p and $\chi(X_1(p)) = 0$ then $\chi(X_2(p)) = 1$. Then $X_2(p)$ must be a 0-cell or an arc. An arc is impossible because the components would be connected. If $X_1(p)$ has two 2-cells in K_p and $\chi(X_1(p)) > 1$ then $\chi(X_2(p)) < 0$. Then $X_2(p)$ must be a non-simple curve, but this is impossible.

   For $N_2 = 3$ and two components $X_1(p)$ and $X_2(p)$ in K_p, all three 2-cells must be connected. It follows that $\chi(X_1(p)) = n_0 - n_1 + 3$. The only possible second disjoint component is a 0-cell. It follows that $\chi(X_2(p)) = 1$ and then the first component must be $\chi(X_1(p)) = 0$. But for this configuration $\chi(X_1(p)) = 1$ which is a contradiction.

   For $N_2 = 2$, only one component in K_p is possible, which is a contradiction to our assumption.

   A voxel is always simple for $N_2 = 1$.

2. Second, assume that $\chi(K_p) = 1$ and $\bar{K_p}$ is not connected. We can apply analogous conclusions as in the first case.

Now we assume that $\chi(K_p) \neq 1$, or $\chi(K_p) = 1$ and K_p includes an isolated 0-cell or an isolated 1-cell or $\bar{K_p}$ includes an isolated 2-cell. If $\chi(K_p) \neq 1$ then p is not 0-simple. If $\chi(K_p) = 1$ and K_p includes an isolated 0-cell or 1-cell then there exists a second component and K_p is not connected and p is not 0-simple. If $\chi(K_p) = 1$ and $\bar{K_p}$ includes an isolated 2-cell then $\bar{K_p}$ is not connected and p is not simple. □

Note that these descriptions are independent from rotations. All 0-non-simple voxels with $\chi(K_p) = 1$ have four or five or six 0-adjacent non-object voxels. Simple element detection is always a crucial part in iterative thinning algorithms. A 26-neighborhood of an object voxel p has a total number of

$$2^{26} = 67,108,864$$

possible configurations in a binary 3D picture. A voxel is simple in

$$25,985,144$$

cases [see, e.g., (Lohou 2001)]. It follows that a voxel is non-simple in

$$41,123,720$$

cases. We prove the following new proposition:

4. PROPOSITION. *The total number of non-simple voxels with* $\chi(K_p) = 1$ *equals* $434,304$.

Proof. We consider the left sketch in the upper row of Figure 5.3. Eight different 0-cells can be the isolated 0-cell in the P-attachment set K_p of p. For each case, the simple curve in K_p can be located in three different locations. The picture value of the four 0-cells on this closed curve can be 1 or 0. The number of configurations with one isolated 0-cell and one simple curve, as shown in Figure 5.3, is equal to $8 * 3 * 2^4 = 384$. The numbers of configurations for all other cases are calculated in an analogous way. □

It follows that in $40,689,416$ cases, voxels are 0-non-simple and $\chi(K_p) \neq 1$.

5.2.4 Characterizations of simple sets

Changing the value of one simple element changes the actual configuration for all adjacent elements and may create new simple elements. Simple deformation algorithms have in common that a set of elements satisfies a number of tests, and these elements change the values $P(p)$ simultaneously.

For example, a cube of $2 \times 2 \times 2$ object elements would disappear because each single element is simple. However, the whole set of elements is not simple. This is the reason that tests whether a single element is simple or not, are insufficient to preserve the topology of a picture. Similar to tests of single elements to be simple or not we are interested in characterizing sets of elements to be simple in order to show that specified algorithms for simple deformations preserve the topology of the picture. Analogously to (Ronse 1986), we define simple sets. Let D be the set of elements in the original picture P that values have changed during deformation; let P^* be the picture after deformation, $D = \{p : p \in P \wedge P(p) \neq P^*(p)\}$, $D \subseteq P$. Let S be the set of object elements in P, let B be the set of non-object elements in P, let S^* be the set of object elements in P^* and B^* the set of non-object elements in P^*. Then $S^* = S \setminus D$ or $S^* = S \cup D$ and $B^* = B \cup D$ or $B^* = B \setminus D$. $C_\alpha(P)$ is the number of α-connected components of object elements in P and $C_{\alpha'}(P)$ is the number of α'-connected components of non-object elements in P.

18. DEFINITION. *D is* α-simple *iff $C_\alpha(P) = C_\alpha(P^*)$ and $C_{\alpha'}(P) = C_{\alpha'}(P^*)$. D is* strongly α-simple *iff (i) for each α-connected component of object elements exists exactly one α-connected component of object elements after deformation and vice versa, and (ii) for each α'- connected component of non-object elements exists exactly one α'-connected component of non-object elements after deformation and vice versa.*

Analogously, we say that a picture P is α-deformable by set D iff $C_\alpha(P) = C_\alpha(P^*)$, $C_{\alpha'}(P) = C_{\alpha'}(P^*)$. A picture P is strongly α-deformable by set D iff (i) for each α-connected component exists exactly one α-connected component in the resulting picture and vice versa, and (ii) for each α'- connected component exists exactly one α'-connected component after deformation and vice versa.

In other words, a subset D of a digital picture P is strongly α-simple iff two bijective maps exist: one between α-connected components of object elements of P and α-connected components of object elements after deformation of P by D, and the other one between the α'-connected components of non-object elements of P and the α'-connected components after deformation of P by D.

We consider the following definition of a *simple sequence* (Ma 1994, Kong 1995):

19. DEFINITION. *Let $P = P_0$ be the original picture and P_i the result of deformation i. A sequence $q_1, q_2, ...q_n$ of distinct elements of an α-connected component in a digital picture P is called an α-*simple sequence *of P if q_1 is α-simple in P, and q_i is α-simple in $P_{i-1}, 2 \leq i \leq n$. A set D of elements in a digital picture is called* simple *in P if D is empty or if D is finite and the elements of D can be ordered as an α-simple sequence of P.*

We present the following theorem to show that above definitions are equivalent.

6. THEOREM. *A picture P is strongly α-deformable by D iff D is simple in P.*

Proof. First, we assume that P is strongly α-deformable. Per definition, no α-connected component and no α'-connected component can completely disappear. For each α-connected component of elements S' exists a proper subset D' with $D' \subset S'$, $(S')^* = S' \setminus D'$ or $(S')^* = S' \cup D'$, and D is the union of all these subsets D'. P is a digital picture, it follows that D is finite or empty. If D is empty then no pixel value has been changed, and the α-connected components in the original picture are unchanged. In case that D is finite then all subsets D' are finite or empty. For each α-connected component in the original picture exists exactly one α-connected component after deformation. For each α-connected component exists a set D' which changed the value during deformation. This set is empty or finite. Each non-empty D' has at least one element that must be simple otherwise the change of the value would split the α-connected component in P or the α'-connected component in $\overline{P}$. In case that D' has more than one element, all these elements can be ordered in an α-simple sequence. All these sequences can be ordered in one sequence D one after another and this new sequence is α-simple. That means that D is simple in P.

Now we assume that D is simple. Per definition D must be empty or finite. In case that D is empty, then the α-connected components of P stay exactly the same and all properties of strongly α-deformable are valid. If D is finite, then the elements of D can be ordered as a simple sequence of P. Now we consider all elements of D' of the same α-connected component with $D' \subseteq D$. That means no α-connected component can be split or vanish and no α'-connected component can be split or vanish. It follows that two bijective maps exist between α-connected components before and after simple deformation, and between α'-connected components before and after simple deformation. It follows that D is strongly α-deformable. ☐

As a consequence, a thinning algorithm needs to test whether a marked set D is simple in P or not. We are interested in using local operations to show that the implemented algorithms preserve the topology of the picture.

5.3 Criteria for topology preservation

An algorithm preserves the topology of a 2D picture if it satisfies the following properties:

1. It must not split an α-connected component of P into two or more α-connected components of P.

2. It must not completely delete an α-connected component of P.

3. It must not split an α'-connected component of $\overline{P}$.
4. It must not completely delete an α'-connected component of $\overline{P}$.

An algorithm preserves the topology if it (for any input picture) changes only the values of a simple set D. For practical reasons, it is more interesting to find test criteria based on local neighborhoods. Two more definitions are required to report about conditions for topology preservation tests.

20. DEFINITION. *A set of elements is* small *if every 2 elements of the set are α-adjacent to each other. An α-*deformable set *is such a set which can be deformed while preserving α-connectivity.*

Obviously, every small set of elements is connected. A pair of α-simple elements p, q is α-deformable iff q is α-simple after p is deformed. For example, if p and q are 4-neighbors, then the pair p, q is 8-deformable iff the number of 8-connected components in the adjacency set of p, q is one. The following examples show configurations where p and q are 4-adjacent and the number of 8-connected components is larger than one:

$$\begin{array}{cccc} 0 & 1 & 0 & 0 \\ 0 & p & q & 0 \\ 0 & 0 & 1 & 0 \end{array} \qquad \begin{array}{cccc} 1 & 1 & 0 & 0 \\ 1 & p & q & 0 \\ 0 & 0 & 1 & 1 \end{array} \qquad \begin{array}{cccc} 0 & 1 & 1 & 0 \\ 0 & p & q & 0 \\ 0 & 1 & 1 & 0 \end{array}$$

Let A be a thinning algorithm for 2D pictures. The following conditions are sufficient to show that A preserves topology (topology preservation test):

1. If an object element has been changed by A then it must be simple.
2. If two 4-neighbors in P have been changed by A then they must constitute a simple set.
3. No small set of object elements has vanished by A.

Analogously (Ma 1994, Kong 1995), we can verify that a 3D thinning algorithm A preserves topology. Let P_j, with $0 \leq j \leq m$, be the resultant picture after j iterations; let A_j, with $0 < j \leq m$, be the application of algorithm A after $(j-1)$ iterations. A parallel 3D thinning algorithm A preserves topology if the following conditions hold, for every iteration j and every picture P_j:

1. A_j changes only simple voxels in P_{j-1}.
2. Let T be a set of two or more voxels contained in a block of 1x2x2 or 2x1x2 or 2x2x1 voxels. If each voxel of T is simple in P_{j-1}, then each voxel is still simple after changing the other voxels in T.

3. If a small component in P_{j-1} exists then A_j does not change at least one voxel of this component.

We observe that an α-path π with fixed end elements crossing T is homotopic to an α-path π' with the same fixed end elements crossing T after one thinning iteration.

5.4 Examples of iterative thinning algorithms

Researchers developed different strategies to ensure that only elements of a simple set D change their values $P(p)$ in one iteration. There are three main strategies:

1. The investigation of large (more than 26 0-adjacent n-cells) adjacency sets.
2. Iterations are divided into sequential directional subiterations.
3. Division into disjoint subsets where only specified subsets are considered for tests in one subiteration.

The first strategy operates in parallel. The approach applies the idea of the grassfire transform which assumes that the fire starts at every border element at the same time. Tests of simplicity for a set D in one iteration step are done by examining neighborhoods of the 0-adjacent elements of $p \in D$. For example, in (Ma 1995) an algorithm for 3D pictures is published where the test of simplicity for a single element depends on 30 elements, and an additional test is required to guarantee topology preservation. In 2D, sequential thinning algorithms incorporate additional conditions to ensure that only simple sets are deleted in one iteration or they operate in sequential directional subiterations. Such strategies have been extensively developed for 3D pictures. Different adjacency relations (0-, 1-, 2) define different types of directions and the number of subiterations (6, 8 or 12). Tests of simplicity are restricted to all 0-adjacent elements, as for example in (Palagyi and Kuba 1998, Palagyi and Kuba 1999). The location of the resulting set of skeletal curves depends on the defined sequence of subiterations. Results vary under rotation. The third strategy is applied in (Bertrand and Aktouf 1994, Saha et al. 1997).

Non-end elements of ideally thin curves do not satisfy pixel deletion criteria in the sense that they are non-simple (for p with exactly one 0-adjacent object voxel is $\chi(K_p) = 1$ and K_p and $\overline{K_p}$ are connected). An efficient thinning algorithm stops if all remaining elements are non-simple. This must be different for end elements to ensure Condition 3 of the criteria for topology preservation. Several thinning algorithms stop these iterative procedure if all remaining elements are ”multiple” [see for example (Pavlidis 1980, Borgefors et al. 1999)]. Multiple elements are not uniquely defined. The identification of all multiple elements is not enough to generate digital curves or arcs. Normally, one postprocessing step is required to determine non-simple elements in the set of multiple elements.

An example of a sequential algorithm in 2D is given in (Hilditch 1969) (Algorithm A). The following criteria are tested, for all elements $p \in P$, and pixels are marked in standard scan order. Let L_α the number of α-adjacent object voxels.

A1: p is an object pixel.

A2: p is a border pixel, that means at least one 4-neighbor is a non-object pixel.

A3: p is not isolated or an end pixel, that means $L_8(p) > 1$.

A4: At least one object pixel in $A_8(p)$ is unmarked.

A5: $X_H(p) = 1$.

A6: If q_3 is marked, setting $P(q_3) = 0$ does not change $X_H(p) = 1$.

A7: If q_5 is marked, setting $P(q_5) = 0$ does not change $X_H(p) = 1$.

At the beginning, let M be the set of all object pixels (1's) and B the set of all non-object pixels (0's). After one iteration, all marked pixels (set D) are changed into non-object pixels. The result $M' = M \setminus D$ becomes M, and $B' = B \cup D$ becomes B for the next iteration and so on, until no simple pixel is left.

Based on criteria (as stated in the previous subsection), we can verify that this algorithm preserves topology. Condition 5 implies that p is a simple pixel, at least one 4-neighbor is a 0 and p is not isolated. Conditions 1, 2 and the end element condition are checked at the beginning to be computationally efficient. Conditions 6 and 7 ensure that, if two 4-neighbors in P are changed, then they constitute a simple set. If q_3 or q_5 are simple and marked then Criterions A6 and A7 guarantee that p can only be marked if p is simple in $P \setminus \{q_3\}$ and p is simple in $P \setminus \{q_5\}$. No small sets of object elements can change because of Conditions 3 and 4. See Figure 5.7 for results.

These examples show that the algorithm is sensitive regarding small holes or sharp peaks. After pre-processing, applying the morphological operations *closing* and *opening* [see. e.g., (Gonzalez and Woods 2002) for those], the results (see Figure 5.8) show improvements regarding robustness against noise.

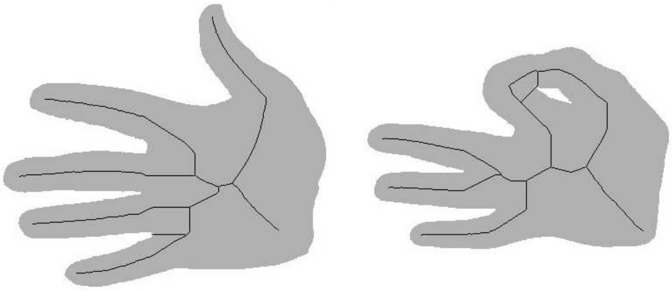


Figure 5.7: *Results of algorithm A without morphologic pre-processing.*

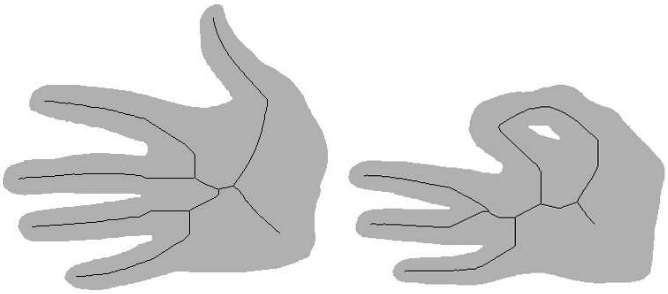


Figure 5.8: *Results of algorithm A with morphologic pre-processing (closing followed by opening).*

The result of preprocessing and thinning delivers a set of connected digital arcs that is topologically equivalent to the smoothed picture. It approximates the medial axis. End elements are preserved.

One standard example of using directional subiterations is the four-subiteration algorithm by Rosenfeld (Rosenfeld 1975) (Algorithm B). A pixel p is deleted if

B1: p is an object pixel.

B2: p is not isolated or an end pixel, that means $L_8(p) > 1$.

B3: $X_H(p) = 1$.

B4: $P(q_{2i+1}) = 0$, where $i = 1, ..., 4, 1, ...$, at successive iterations.

The first of these four subiterations changes in parallel only border pixels where $P(q_3) = 0$, the second subiteration changes only border pixels where $P(q_5) = 0$

and so on. The algorithm terminates when no changes occur during four successive iterations. The location of the resulting skeleton depends on the termination point. For example, if the algorithm does not identify a changeable element after two subiterations then elements of the skeleton are not located on a center line.

Algorithms A and B deliver the basic ideas for directional subiteration algorithms in 3D.

Computationally more efficient are parallel algorithms with two or only one subcycle. A typical example for a 2-subcycle algorithm (Zhang and Suen 1984) (Algorithm C) is the following: The first subcycle only deletes a pixel p if:

C1: p is an object pixel.

C2: $1 < L_8(p) < 6$.

C3: $X_R(p) = 2$.

C4: $P(q_1) \cdot P(q_3) \cdot P(q_7) = 0$.

C5: $P(q_1) \cdot P(q_7) \cdot P(q_5) = 0$.

Equations C4 and C5 are both equal to 0 if $P(q_1) = 0$ or $P(q_7) = 0$ or $P(q_3) = 0$ and $P(q_5) = 0$. In the second subcycle, the last two conditions are replaced by equations $P(q_3) \cdot P(q_5) \cdot P(q_7) = 0$ and $P(q_1) \cdot P(q_3) \cdot P(q_5) = 0$. The algorithm terminates when no changes occur after a full cycle. The algorithm delivers a union of 4-connected arcs. In contrast to algorithms A and B, the location of the skeletons (see figures below) is different.

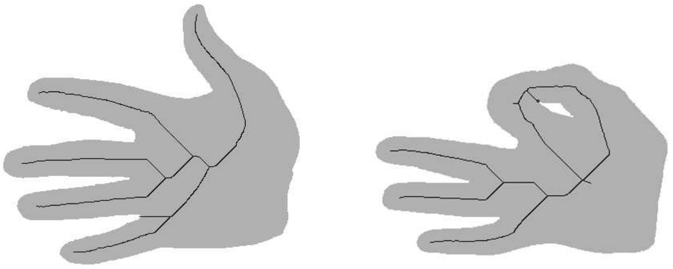


Figure 5.9: *Results of algorithm C without morphologic pre-processing.*

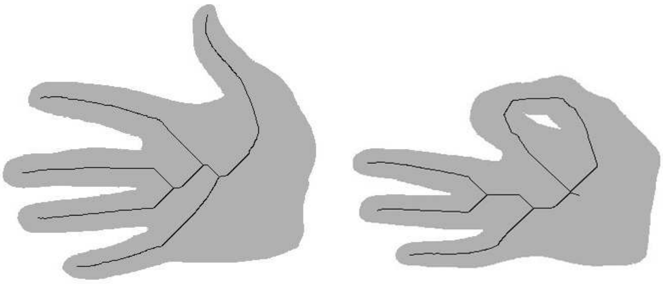


Figure 5.10: *Results of algorithm C with morphologic pre-processing (closing followed by opening).*

Algorithm D (Pavlidis 1980) applies the idea of the grassfire transform. Border pixels are traced and labeled per iteration. Only multiple pixels are retained. Iterations continue until all object elements are multiple. Pixel p is *multiple* if at least one of the following conditions is true:

D1: $X_H(p) > 1$

D2: p has no 4-adjacent pixel which is an inner pixel (all 4-adjacent pixels are in the object).

D3: p has at least one 4-adjacent border pixel but it is not traced immediately before or after p.

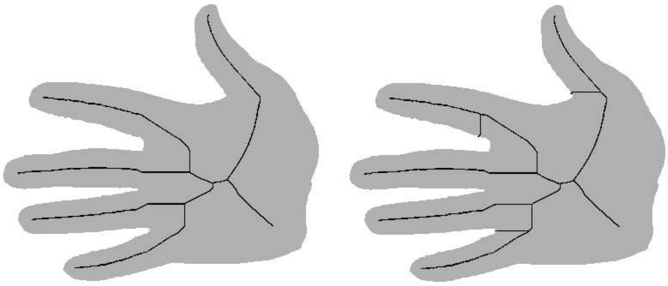


Figure 5.11: *Result of algorithm D with and without pre-processing.*

Note that the set of non-simple pixels is a subset of multiple pixels.

All four algorithms preserve the topology of the smoothed picture (original after morphological preprocessing). Comparing the examples, algorithms A and D deliver the same number of arcs and the same number of branch elements (more than two 8-adjacent object elements, see the next chapter for exact definitions).

Algorithm C generates, after the same preprocessing steps for an equal input picture, a 4-connected skeleton (based on C3) with 7 arcs and 3 branch elements. It shows how important conditions for simplicity of single elements are for skeleton approximations and particularly for features such as number of branch elements.

We can also verify that algorithm D preserves topology. D1 guarantees that p is simple for deletion, p is not an isolated object element (all 8-neighbors are non-object elements) and p is not an inner object element (all 8-neighbors are object elements). A small set of object elements cannot vanish because of condition D2. Two 4-neighbors can only change their values if they constitute a simple set because of D3. Results of algorithm D are not unions of connected simple arcs, in some locations they are 2 pixels width. For example, we consider a rectangle with 2 pixels height and 10 pixels width. Such an object has no inner pixel, all pixels are retained. A postprocess would be required to generate a union of simple arcs. The basic concept of algorithm D is extended to the 3D case in (Borgefors et al. 1996, Borgefors et al. 1999). In addition, border voxels are labeled with the actual iteration number in such a way that those labels represent the distance to the nearest non-object element based on the d_6-metric.

Our main focus of those expectations on skeletons, as described in Chapter 3, is the generation of topologically equivalent curve-like structures for classification. As a conclusion from above studies in 2D we selected an algorithm for 3D which applies the ideas of algorithms A and B. Figure 5.12 shows the results of algorithm A, C and D for the same picture, and it shows the result of a distance skeleton where skeletal elements are calculated as local maxima of a d_4 distance map.

Algorithm A generates topologically equivalent skeletons which are unions of 8-connected digital arcs or curves, located close to the center of elongated parts, post-processing steps for the deletion of unnecessary elements or to connect elements are not required. Algorithm C generates 4-connected skeletons, algorithm D generates ”thick” skeletons and distance skeletons are not connected.

The algorithm in (Palagyi et al. 2001, Palagyi and Kuba 1998) uses 6 subiterations (U, D, N, S, W, E) (see Figure 5.13). The algorithm follows the concept of directional thinning as algorithm B in 2D and the authors in (Palagyi and Kuba 1998) have proved that it preserves topology. It is a curve thinning algorithm for 3D pictures as required for astrocytes pictures. End element conditions are different for curve thin-

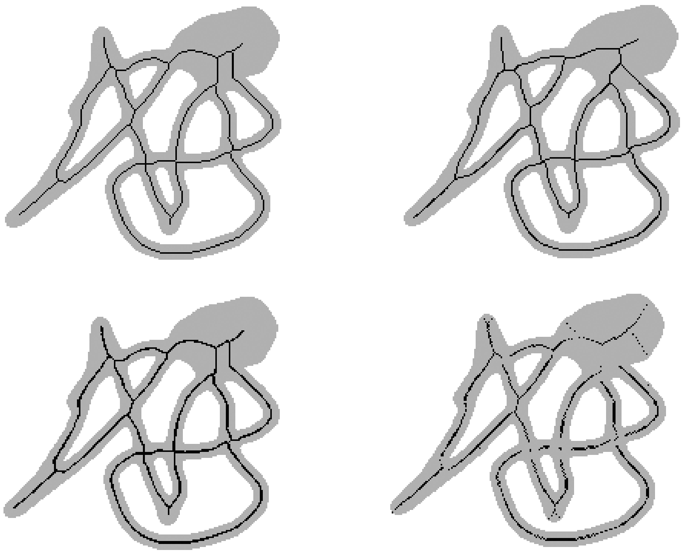


Figure 5.12: *Upper left: result of algorithm A, upper right: result of algorithm C, lower left: result of algorithm D, lower right: distance skeleton.*

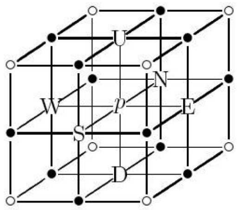


Figure 5.13: *Directions U, D, N, S, E and W for a 6-subiteration algorithm.*

ning and surface thinning. The algorithm has been successfully applied in medical image processing.

We apply the same concept and we introduce some adjustments (Klette and Pan 2004). As in 2D for directional thinning procedures; only border voxels in one direction are considered for further tests in each subiteration. Then simple voxels that are not end elements are marked. All marked voxels have a second test to secure that they are still simple and not end elements after some adjacent voxels have been processed. The number of changeable voxels in the 2-adjacency set is limited to a threshold $t, 1 \leq t \leq 6$. The pseudo code [see Algorithm 5.1] shows the main processing steps.

```
initialize $P$ {load input data into $P$}
initialize $T$ {load lookup table into the array $T$}
initialize $L$ {add all border voxels into the list $L$}
repeat {one iteration step}
  $n \leftarrow 0$ {counter for changeable voxels per iteration}
  for all directions $i \in \{U, D, N, S, E, W\}$ do {one subiteration step}
    $V \leftarrow \emptyset$ {$V$ stores changeable voxels for one subiteration}
    for all $p \in L$ do {test voxels}
      if $p$ is $i$-border voxel and $p$ is changeable then
        add $p$ to $V$
    for all $p$ in $V$ do {recheck changeable voxels}
      if $p$ is simple then
        if $p$ is not end element or $m_2(p) \geq t$ then
          $P(p) \leftarrow 0$ {$m_2(p)$ is the number of 2-adjacent voxels in $V$}
          $n \leftarrow n + 1$
          for all $q \in A_2(p)$ do
            if $P(q) = 1$ then
              add $q$ to $L$ {change of $p$ generates new border voxels}
until $n = 0$
```

Algorithm 5.1: *Algorithm G: 3D sequential 6-subiteration thinning algorithm.*

In (Palagyi and Kuba 1998), simple element configurations are described by a set of masks. A Boolean lookup table indexed by the values of p's 26 0-adjacent elements is used. Characterization 7 (or equivalent characterizations using masks) of simple elements in the grid point model are applied to label each configuration. The actual tests during thinning are done by comparing the configurations of a voxel p in the picture with rows in the lookup table.

For computational efficiency, the construction of Boolean lookup tables is a common way. A Boolean lookup table stores the 2^{26} configurations of all 26 0-adjacent elements in a defined order. Each table entry (row) is a binary word, 27 bits long. Each bit represents the binary value of 26 adjacent elements of p and one bit (the first or the last) a label $l \in \{0, 1\}$. If a voxel in $A_0(p)$ is an object voxel, then its corresponding bit is set to 1. Otherwise, the bit is set to 0. In this way, each binary word of 26 bits represents a unique configuration of $A_0(p)$. For each binary word,

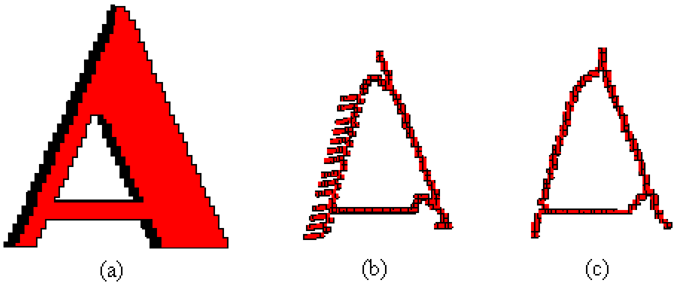


Figure 5.14: *Left: Synthetic 3D object "A". Middle: 3D skeleton of "A" generated by original version. Right: 3D skeleton of "A" generated by algorithm G.*

if p is simple, the first bit is marked by 1. The new characterization of non-simple voxels developed in this thesis [see also (Klette and Pan 2004)] is useful for setting the labels. An efficient algorithm executes successive the following steps for each configuration:

1. Calculate N_2 (number of 2-adjacent non-object voxels) and $\chi(K_p)$
2. If $\chi(K_p) \neq 1$ then $l = 0$ (40,689,416 cases)
3. If $N_2 \leq 3$ and $\chi(K_p) = 1$ then $l = 1$.
4. If $N_2 > 3$ and $\chi(K_p) = 1$ then if an isolated 0-cell is in K_p then $l = 0$ (297,600 cases); otherwise, if an isolated 2-cell is in $\bar{K}_p$ then $l = 0$ (133,632 cases); otherwise, if an isolates 1-cell is in K_p then $l = 0$ (3,072 cases); otherwise $l = 1$.

Experimental results have shown that this construction process is very fast (four times faster than tests for simple voxels based on other equivalent characterizations). The size of the lookup table can dramatically be reduced by removal of all rows starting with 0 such that the table stores only configurations for simple elements. The stored lookup tables are used during the thinning process to compare actual configurations in the picture with rows in the table for the simplicity test.

A direct application of the above algorithm during thinning without lookup tables is a different option. The number of computations is reduced because of preprocessing steps using morphological operations.

The algorithm in (Palagyi and Kuba 1998) can generate artificial arcs in 3D objects because the change of 2-adjacent object voxels can produce new end elements [see Figure 5.14]. The original algorithm creates arcs and branching elements [see Figure 5.14(b)] which do not describe the shape of the object. This type of problems are discussed in (Lee et al. 1994, Palagyi et al. 2003). The modified version of the original algorithm (algorithm G) counts the number of 2-adjacent voxels which have been changed in previous subiterations. If this number is larger than a certain threshold value t, $1 \leq t \leq 6$, then $P(p) = 0$ independent from the end-element condition. Experimental results have shown that $t = 1$ generates a union of arcs which are an adequate representation of shape [see Figure 5.14(c)].

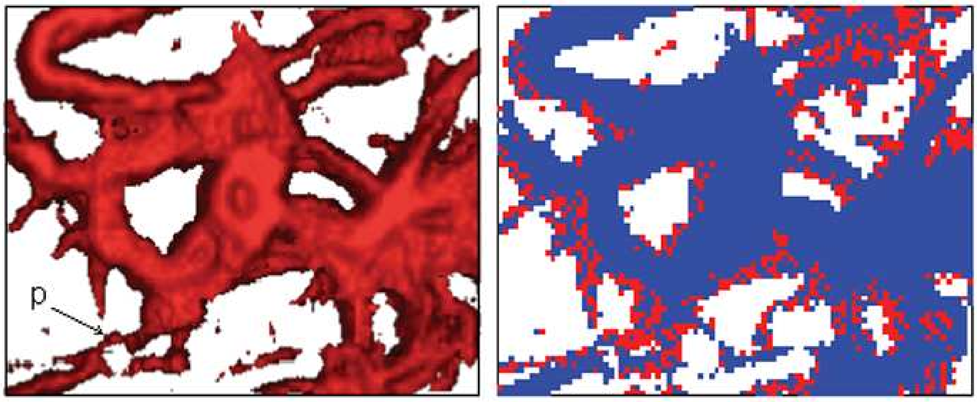
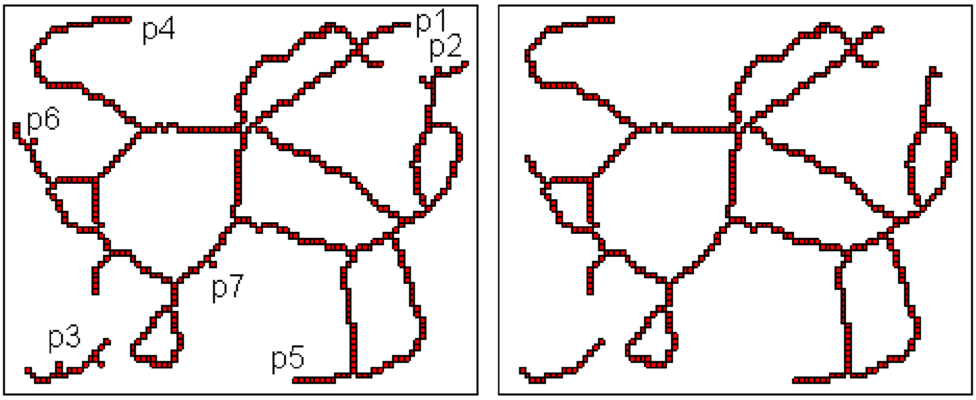


Figure 5.15: *Left: sub-volume, object is not connected at p, overlapping. Right: sub-volume as voxel view (OpenGL).*



(a) (b)

Figure 5.16: *Left: skeleton produced by original version. Right: skeleton produced by algorithm G.*

A sub-volume (see Figure 5.15) of the 3D picture in Figure 1.4 has been processed by morphological operations ($closing_{(S_1)} + opening_{(S_2)}$), where structuring elements S_α coincide with neighborhoods N_α) followed by the original algorithm [see Figure 5.16 (a)] and exactly the same preprocessing steps have been done followed by algorithm G [see Figure 5.16 (b)].

For example, parts of the skeleton labeled with p_2, p_3, p_6 and p_7 in Figure 5.16 (left) disappear in Figure 5.16 (right). Obviously, the number of branch elements is different. Parts labeled with p_1, p_4 and p_5 in Figure 5.16 (left) have a reduced number of voxels [see Figure 5.16 (right)], the arc lengths are different. In those cases, the number of branch elements does not change.

5.5 Summary

This chapter reviews characterizations of simple elements which are a fundamental concept for simple deformations of binary pictures. Different models are used to describe simple elements and simple sets. The chapter shows that characterizations are equivalent. The definition of non-simple elements is new, and it is used in a topologic thinning algorithm. We report about existing thinning algorithms in 2D and as a conclusion of those studies we modify one existing algorithm for 3D which delivers curve-like structures. The skeletons follow the requirements 1, 4 and 5 as stated in Chapter 3. The chapter describes that small variations of conditions in the thinning algorithm deliver different results. Branching elements can be produced by unstable thinning and they have no relationship to the shape of objects.

Chapter 6

Abstract curve graphs

In practice, distance skeletons, topologic skeletons and combinations of those are calculated and used for picture analysis. The definition of the branching index of points on curves in Euclidean space (introduced by Urysohn and Menger) is fundamental in curve theory. We discuss the derivations of analogous notions such as branching index, branch element and junction for the digital space, and we illustrate the potentials of using junctions in 3D picture analysis. Parts of this chapter are published in (Klette 2006).

6.1 Definitions in Euclidean space

The branching index of a point on a continuous curve classifies points into isolated points, end points, curve points and branching points. In digital space, points are replaced by elements. We consider each 2-cell (pixel) or 3-cell (voxel) as an open set of points in continuous space and any two elements form disjoint open sets. If elements are connected then they share certain cells which belong to the closure of both elements. We discuss approaches of defining the notion of branching index of elements in the digital space. The notion of a junction will play a crucial role. We illustrate the potentials of using junctions in 3D picture analysis focused on the description of skeletal curves. Concepts for describing curve points in a continuous space are known for more than 80 years [see (Klette and Rosenfeld 2004) for a review]. P. Urysohn in 1923 and K. Menger in 1932 defined (independently) *simple curves (arcs)* based on the notion of the branching index of points on curves (arcs). The *branching index* of a point on a curve was defined as follows:

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

Figure 6.1 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

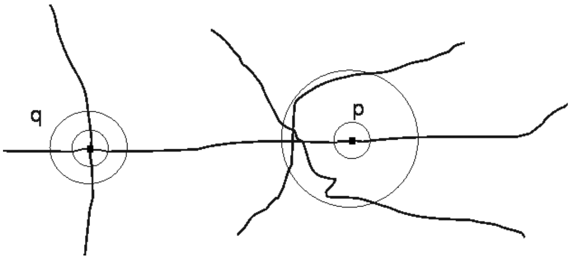


Figure 6.1: *$F(U_\varepsilon(q)) \cap \gamma = 4$ and $F(U_\varepsilon(p)) \cap \gamma = 2$, assuming that ε is sufficiently small.*

Euclidean space is the number of crossings of a circle (with radius $\varepsilon < r$ and center point p) with curve γ. For all circles close enough to p, the branching index is a constant integer value.

An isolated point has branching index 0, and an end point has branching index 1. 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, or a simple curve with one of its points labeled as an endpoint. A branching point has a branching index larger than 2.

6.2 Digital curve elements

We consider curve-like structures produced by a 3D topologic thinning algorithm. A 3D curve skeleton ρ is a digital curve, which we consider with respect to 0-adjacency. A translation of the concepts in the Euclidean space to the 3D digital space leads to the following definitions:

- A digital curve ρ has *branching index* $m > 0$ at voxel $p \in \rho$ iff exactly m 0-adjacent voxels are elements of ρ;
- an element $p \in \rho$ is *regular* iff p has branching index 2;
- $p \in \rho$ is a *branch element* iff p has a branching index of at least 3;
- $p \in \rho$ is an *end element* iff p has branching index 1;
- $p \in \rho$ is a *singular element* iff p is either a branch element or an end element;
- the digital curve ρ in 3D space is *simple* iff every element in ρ is regular; and

- ρ is a *simple arc* iff it is either a curve in which every element p is regular except for two end elements e_1 and e_2, or a simple curve where one of its elements is labeled to be a double[1] end element.

In a 3D curve skeleton S, we may have 0-adjacent branch elements. We like to extract a graph where the nodes represent branch elements or end elements, and edges represent digital curves. The construction of a weighted graph with centers of single elements (cell model) as nodes would be an advantage for subsequent length measurements. The branching index could be a useful tool to describe the complexity of a curve structure in pictures. The number of 0-adjacent elements of a single element is not an analogous concept to the definition of the branching index in continuous space. Above definitions result into limitations of branching indices at elements which restricts the generality of the concept. For ensuring unlimited branching indices we introduce specific clusters of branch elements.

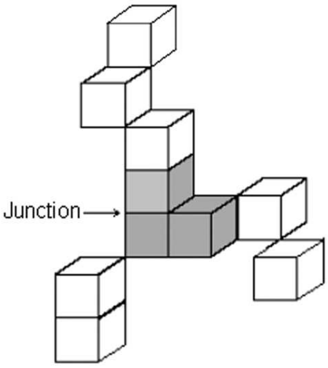


Figure 6.2: *Junction with cardinality* $C_J = 3$.

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

The cardinality of a junction C_J is the number of 0-connected branch elements. Figure 6.2 illustrates a junction which consists of three branch elements. Note that a junction is a nonempty 0-connected set of branch elements. A single branch element also represents a junction ($C_J = 1$).

---

[1] One labeled element represents two end elements $e_1 = e_2$.

It follows that a junction has a branching index greater than 2. For example, the branching index of the junction shown in Figure 6.2 is 3. We identify the geometric location of a junction in the Euclidean space with its centroid. It is required for the calculation of Euclidean distances. Let J be a junction, n be the number of branch elements p_i constituting J, with $p_i = (x_i, y_i, z_i),\ 1 \leq i \leq n$. The *centroid* $c(J)$ of J is a 3D point with coordinates

$$x = \frac{\sum_{i=1}^{n} x_{p_i}}{n}, \quad y = \frac{\sum_{i=1}^{n} y_{p_i}}{n}, \quad z = \frac{\sum_{i=1}^{n} z_{p_i}}{n} \tag{6.1}$$

If the junction consists of one grid element then the centroid coincides with the center of this element.

6.3 Construction of abstract curve graphs

Complete junctions or end elements are mapped into nodes, and simple arcs are mapped into edges of an undirected graph. The algorithms for those mappings influence subsequent lengths measurements.

23. DEFINITION. *An abstract curve graph corresponding to a digital curve ρ is an undirected graph G, where a node of G represents either a junction or an end element of ρ. Two nodes in G are connected by an edge iff the corresponding junctions or end elements are 0-connected in ρ. G is the abstract curve graph of ρ.*

The abstract curve graph G is uniquely defined by the chosen adjacency (0-adjacency in our case).

An actual mapping can be done by a labeling procedure which is divided into two steps. The first step assign labels to all branch elements and end elements of skeleton picture S. Branch elements $p_i \in J,\ 1 \leq i \leq n$, of one junction get the same label. Let A be an array which stores the labels and corresponding coordinates. Object elements in S are tested in standard scan order and labels are defined as follows:

1. Initialize label: $l = 1$,
2. If p is an end element ($m = 1$), then assign label l to p in A, increase l,
3. If p is a branch element ($m > 2$) and no processed 0-adjacent element of p has a label in A then assign l to p in A, increase l,
4. If p is a branch element ($m > 2$) and one processed 0-adjacent element q has already a label l in A then assign the same label l to p in A.

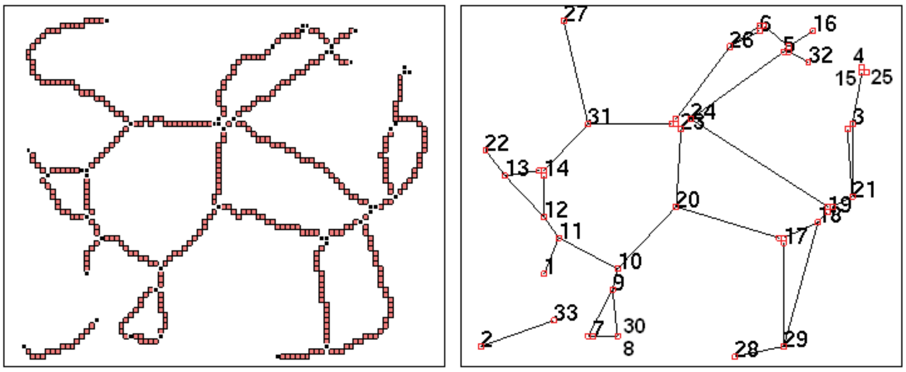


Figure 6.3: *Left: a skeleton (junctions are black). Right: an abstract curve graph (singular elements are labeled).*

The singular element labeling process could also start at a randomly chosen object element p, and labels are propagated to singular elements in connected components recursively.

The second step of the labeling process is designed in a way that all elements of one arc between two singular elements get the same label. If a singular element is a junction then any, not yet labeled branch element of this junction is assigned to exactly one outgoing arc for subsequent length measurements. One option of the arc labeling algorithm starts at the first element (registered in A) with label i. The algorithm selects the first item in A and it issues a new label k for this starting element, registered in another array B. It checks adjacent elements until it finds 0-adjacent unlabeled regular elements. It labels successive 0-adjacent regular elements with the same label k until it reaches a labeled element j in A. This element j in A gets label k in B and represents the end element of an arc. A loop is an arc which starts and ends in the same junction ($i = j$). In case of a loop, we register the last labeled regular element in B as a new end element of an arc in A with l and we increase l. All elements of the traversed arc get the same new label k in B. The process continues until all object elements have a new label in B. In this way all object elements in the skeleton picture are divided into a set of connected arcs which are pairwise disjoint.

Figure 6.4 shows again the skeleton of Figure 6.3; different labels are represented by different colors except the endpoints of arcs.

An undirected graph is completely defined by its adjacency matrix. Parallel to the second labeling step, we construct this matrix. The number of rows and columns is defined by the number of singular elements which corresponds to the value of l in

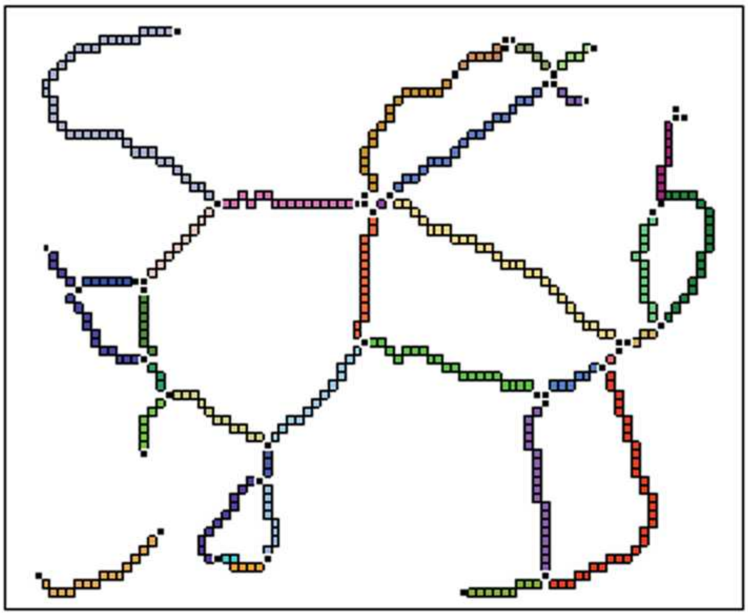


Figure 6.4: *Arcs shown in different colors.*

A at the end of step 2 of the labeling process. We construct a symmetric $l * l$ matrix M where all positions get labels of arcs which connect two singular elements. If there is an arc between two singular elements, say $i, 1 \leq i \leq l$ and $j, 1 \leq j \leq l$, then the arc label will be inserted at (i, j) and at (j, i). If $i = j$ then we insert 0 into the matrix. Otherwise, we set ∞ at the corresponding positions to indicate that there is no edge. The substitution of those arc labels by length measurements[2] creates a cost matrix which will be useful for extractions of properties.

The so constructed adjacency matrix M (or undirected graph) is uniquely defined. However, the number of equally labeled voxels of a detected arc and the length of the arc may differ depending where the algorithm starts. One reason is the fact that the cardinality of a junction is not restricted.

---

[2] It will be discussed in the following chapter.

6.4 Properties of junctions

The branching index in 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 elements into a single node in the abstract curve graph G. Theoretically, the cardinality of a junction can grow without limit.

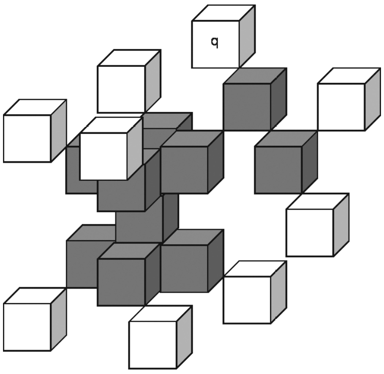


Figure 6.5: *A junction with cardinality 10 and* $m = 9$.

The cardinality of a junction can grow if the picture size or the grid resolution grows. Let us consider Figure 6.5.

- All black cubes are branch elements with a branch index $m \geq 3$, and all white cubes are end elements (if the white cubes would be regular then the junction would not change).
- If end element q (a cube) would share two more vertices with two more elements, then q would change into a branch element. We could continue this process of adding two more elements to one of the new end elements. 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 6.6. We use the concept of an attachment set to separate branch elements into two types.

24. DEFINITION. *Let m be the number of elements in $A_0(p)$ and n the number of components in the P-attachment set of p.* [3]

---

[3] Remember that $A_0(p)$ is the set of all 0-adjacent elements of p.

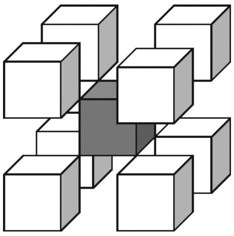


Figure 6.6: *A junction with cardinality 1 and* $m = 8$.

- *A branch element* p *is a* proper branch element *if* $m = n$,
- *A branch element* p *is a* normal branch element *if* $m > n$.
- *A junction is either a 0-region of normal branch elements, or a proper branch element.*

Based on the above definition we conclude (directly from definitions and the example in Figure 6.7) the following:

4. COROLLARY. *(i) A proper branch element is a junction of cardinality one.*

(ii) Two proper branch elements can be adjacent to each other.

(iii) A junction with cardinality two does not exist after topologic thinning.

This definition splits the large junction in Figure 6.5 (for example) into three disjoint junctions. The black elements (see Figure 6.7) 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. It leads to a slightly different definition of graph G. We identify the centroid of each junction J with a node in the abstract curve graph G, and each center of a 0-connected regular element or end element to J is a start (or end) element for the length measurement of a digital arc between two nodes. We do not use elements in junctions with cardinality larger than 1 as start or end elements for lengths measurements. The length between two nodes in the graph is defined by the calculated length of the arc (using different metrics) between two identified start or end elements plus the Euclidean distance from the start element 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 element of the arc to the centroid of the 0-connected junction (if it is connected to a junction).

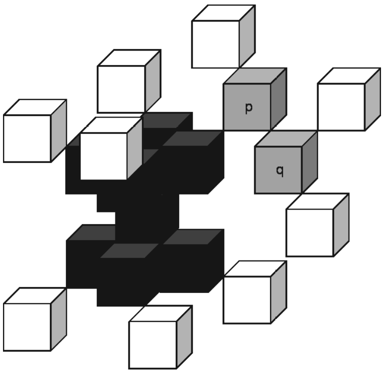


Figure 6.7: *Three 0-connected junctions, one formed by the dark gray elements, and two defined by single elements each (elements p and q).*

We can connect the centroid of a junction with all centers of 0-adjacent regular elements or end elements by continuous straight lines. This approach allows a direct application of the classic definition of branching index from continuous space. Similar to the global idea for EVA-algorithms[4], we determine a set of continuous curves using discrete data with the goal to use existing concepts.

6.5 Summary

The main contribution of the thesis in this Chapter are proposals for the adaption of fundamental concepts on curves in continuous space to digital curves. We propose new methods for the construction of abstract curve graphs which will be used for the extraction of properties based on lengths measurements. Nodes of abstract curve graphs are junctions or end elements, edges are arcs between 0-connected singular elements. We demonstrate that the cardinality of junctions, defined as 0-connected regions of branch elements, can grow without limit. For that reason we introduce a distinction between normal branch elements and proper branch elements as a possibility to restrict the cardinality of junctions. We discuss some properties of junctions generated by topologic thinning.

---

[4] We discussed EVA-algorithms as an efficient way to compute Euclidean distance maps in Chapter 4 of this thesis.

Chapter 7

Shape description

Graph theory is a traditional discipline in mathematics; complex problems are mapped onto graph representations which allow efficient solutions. We have mapped the skeletons onto abstract curve graphs. Those graphs are finite because the pictures are finite. This approach provides options to take advantage of existing graph algorithms for the description of those skeleton images. In this chapter we define a list of properties for those graphs which can be divided into three groups of features. One group describes the structure of skeletal curves without any length measurements. For the second group we discuss lengths measurements and we split components into disjoint simple arcs and compute the lengths of those arcs. In a third group we adapt features designed for graphs. Parts of this chapter are published in (Klette 2006).

7.1 Junctions for the description of volume images

The number of junctions, branching indices and cardinality of junctions are examples of properties to express the complexity of curve structures. The adjacency matrix M of dimension $l * l$, as a result of the labeling algorithms, can be used for the computation of properties which are independent from any lengths measurements. The number of rows or columns represents the number of singular elements, each singular element is a node of a graph. In graph theory, the *degree of a node* v (or valency) of an undirected graph G is the number of edges incident with v. The degree of node v in G is equivalent to the branching index of a singular element labeled with v. If all nodes in G have the same degree then G is regular.

5. COROLLARY. *The degree (branching index)* m *of a node* i, $1 \leq i \leq l$, *in an abstract curve graph* G, *represented by adjacency matrix* M, *is the number* N_i *of labels* j, $1 \leq j \leq l$, *in row* i *of matrix* M.

We can derive a set of features (adapted from graph theory) such as the following:

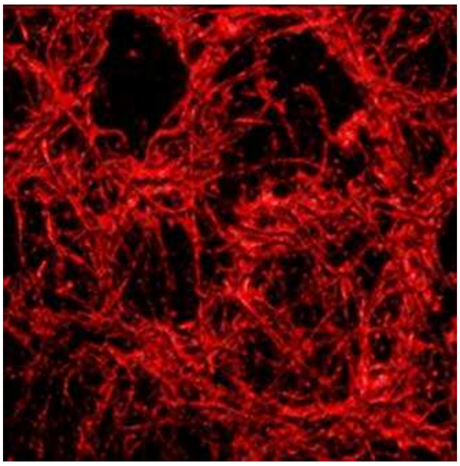


Figure 7.1: *A 3D view of brain tissue (35 slices) of a patient with epilepsy.*

- the *maximum degree (branching index)*, given by $MAX_m = max\{N_i, 1 \leq i \leq l\}$,
- the *minimum degree (branching index)*, given by $MIN_m = min\{N_i, 1 \leq i \leq l\}$,
- the *average degree (branching index)*, given by $A_m = \frac{\sum N_i}{l}, 1 \leq i \leq l$,
- the *number of singular voxels with a fixed branching index* $k, 0 \leq k \leq M_{max}$,
- the *order* (total number of nodes), or
- the *number of junctions* in S, which is equal to the number of rows with $m > 2$.

7.2 Application to astrocyte analysis

Long term observations have shown that astrocytes in human brain tissue (which we related to junctions in the skeleton images) tend to increase and tend to move closer together with increasing degrees of epilepsy. It also appears in some example images that parts of the volume show no astrocytes at all (see Figure 7.1).

For those reasons we are interested in the number of junctions in subvolumes. If we divide the volume into equally sized cubes then some of those cubes have no junctions and others have a high number of junctions. In other words, it is interesting to compute properties which express the *density* and *distribution* of junctions in the whole volume. The higher the number of junctions in a subvolume the higher

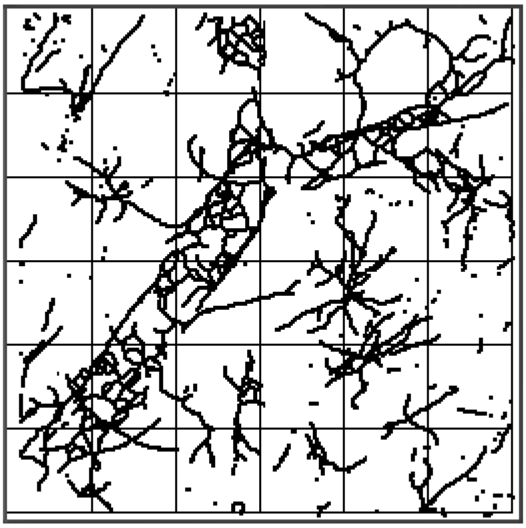


Figure 7.2: *3D skeleton of the binarized volume shown in Figure 1.4.*

is the density relative to the whole volume. The number of junctions in subvolumes is needed to express the *uniformity* of junctions. The volume data are divided into a set of subvolumes $s_l, 1 \leq l \leq S_v$ (small cubes of identical size, S_v is the total number of those cubes). For a fixed branching index j, we count the number of junctions N_{s_l} in each cube having branching index j.

25. DEFINITION. *Junctions of branching index j are uniformly distributed in a volume picture if the number of such junctions is equal in every subvolume. The deviation from this ideal case characterizes the degree of non-uniformity.*

We can also derive easily basic features such as the following:

- the maximum number of junctions with the same branching index in one subvolume $MAX_{sv} = max\{N_{s_l}, 1 \leq l \leq S_v\}$,
- the minimum number of junctions with the same branching index in one subvolume $MIN_{sv} = min\{N_{s_l}, 1 \leq l \leq S_v\}$,
- the average number of junctions with the same branching index in one subvolume $A_{sv} = \frac{\sum N_{s_l}}{S_v}$.

We illustrate the case of a fixed segmentation using pairwise disjoint, uniformly sized cubes of voxels. The data set shown in Figure 1.4 is divided into 36 subcubes,

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 7.1: *Number of junctions per branching index in the red cubes (volume V_1) and in the total volume (volume V_2).*

all of size 42^3. (This also generates some redundant data.) We have chosen this subdivision based on the sizes of given data sets and we had in mind that experts in the school of medicine have the hypothesis that there is a relationship between the number of astrocytes close to the main blood vessels and stages of epilepsy. See Figure 7.2 for the resulting curve (3D skeleton). We demonstrate the approach by results for this example data set.

All identified junctions have branching indices between 3 and 7. The shaded cubes in Figure 7.3 correspond to the location of the main blood vessel, and they contain in total more than 50% of all junctions, for each branching index between three and seven.

Table 7.1 presents the total number of junctions per branching index for the red 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. Table 7.2 shows the total number of junctions per cube. Obviously, they are not (ideally) uniformly distributed in the whole volume. Most of them are located close to the blood vessel.

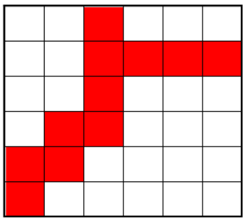


Figure 7.3: *Location of a main blood vessel (shown as red cubes) detected by analyzing the 3D skeleton shown in Figure 7.2.*

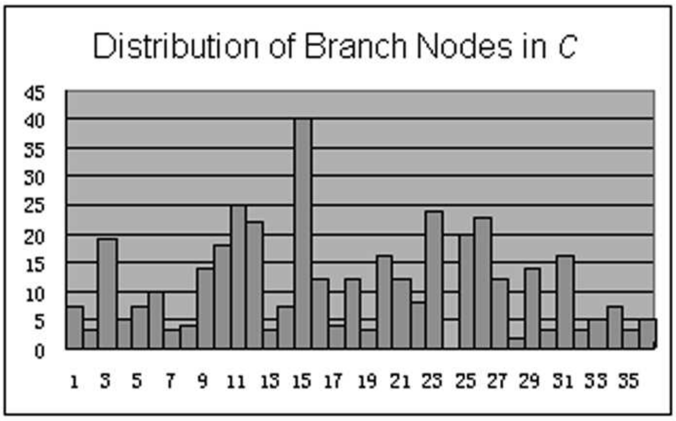


Table 7.2: *Distribution of junctions in subcubes: The horizontal axis represents the ordinal numbers of the subcubes, and the vertical axis represents the number of all junctions in a subcube.*

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 the cardinality and the branching index remaining at low values. Theoretically, this is not always the case as discussed in Chapter 6.

7.3 Length measurements in volume images

In this section we use the length of curves in subvolumes to derive features which are measures for the density and the distribution of curves in subvolumes relative to the whole volume. Figure 7.1 shows subvolumes where blood vessels are very dense, and subvolumes without vessels. This is reflected in the skeleton image which shows very densely located arcs in some subvolumes and only some arcs in others. The total length of those curves per subvolume is a measure for the density of arcs and curves. In Chapter 6 we derived the abstract curve graph expressed in an adjacency matrix M. Length measurements between any two singular elements (nodes of the graph) are required for the construction of a weighted cost matrix, where the weight is the distance between nodes. Accurate length measurements for digital curves is an active research area since a long time. The Euclidean distance is a measure for the shortest continuous straight line between vertices. It is an indication how far away or how close vertices are to each other. Having the coordinates of centroids of all singular elements allows the computation of the Euclidean distance between any two of them.

The Euclidean distance $d_e(p,q)$ between two nodes is a measure for the length of a continuous straight line between those nodes p and q, and it is not a measure for the length of the digital curve between those nodes. However, the Euclidean distance is useful to describe the discrepancy between the straight line and the actual curve. An easy approach to describe the density of those arcs in subvolumes is counting the number of regular voxels between end voxels during the labeling process as described in the previous chapter. To reduce the error for junctions we can add the Euclidean distance between the centroid and the center of the starting voxel to the number of regular voxels. This estimation is very rough. A different possibility is the use of chamfer distances (see Chapter 2). For example, we could add weights 1 for 2-adjacent neighbors, $\sqrt{2}$ for 1-adjacent neighbors and $\sqrt{3}$ for 0-adjacent neighbors. For a given 0-connected arc γ in a 3D volume image and a grid of resolution h the length L of the arc is defined by

$$L_h(\gamma) = \frac{1}{h}(n_1 + \sqrt{2}n_2 + \sqrt{3}n_3). \tag{7.1}$$

An advantage of this approach is that it can be done during the labeling process. In general, those algorithms have linear time complexity. Those local estimators take only distances of local neighbors into account. They have the disadvantage that they do not converge to the true length of the arcs with increasing grid resolution (Klette and Rosenfeld 2004).

A different approach is used in global length estimations for digital arcs or curves. One example for such a global length estimator is the so called *digital straight segment algorithm* (DSS-algorithm). The algorithm cuts an arc into a set of digital straight segments, and the total length is the sum of the lengths of those segments. An example for a linear time 3D DSS-algorithm (by I. Debled-Rennesson) is also contained in (Klette and Rosenfeld 2004).

Let a and b be relatively prime integers, and let μ and ω be integers:

26. DEFINITION. *$D = D_{a,b,\mu,\omega} \subset \mathbb{Z}^2$ is a 2D arithmetic line defined by integers a, b, μ, ω, if and only if the following is true:*

$$D_{a,b,\mu,\omega} = \{(x,y) \in \mathbb{Z}^2 : \mu \leq bx - ay < \mu + \omega\} \tag{7.2}$$

$D_{a,b,\mu,\omega}$ ia called an arithmetic line with slope $\frac{b}{a}$, approximate intercept $\frac{\mu}{a}$ and arithmetic width $|\frac{\omega}{a}|$. The parameter μ is the lower bound of D, and the parameter ω is the arithmetic thickness of D.

27. DEFINITION. *$D = D_{a,b,\mu_1,\mu_2,\omega_1,\omega_2} \subset \mathbb{Z}^3$ is a 3D arithmetic line defined by integers a, b, c, μ_1, μ_2, ω_1, and ω_2, if and only if the following is true:*

$$D_{a,b,\mu_1,\mu_2,\omega_1,\omega_2} = \{(x,y,z) \in \mathbb{Z}^3 : \mu_1 \leq cx - az < \mu_1 + \omega_1 \wedge \mu_2 \leq bx - ay < \mu_2 + \omega_2\} \tag{7.3}$$

The parameters $a, b,$ and c are relatively prime integers, $0 \leq c \leq b \leq a$, which represent x-, y-, and z-coordinates respectively, the parameters μ_1 and μ_2 are the lower bounds of D, and the parameters ω_1 and ω_2 are the arithmetic thickness of D.

Chain codes of DSS are defined as follows:

28. DEFINITION. *Let $\gamma_{a,b}$ be a digital ray such that*

$$\gamma_{a,b} = \{(x, ax + b) : 0 \leq x < \infty\}, \tag{7.4}$$

where $0 \leq a \leq 1$. It is enough to consider this choice of a due to the symmetry of the grid. $\gamma_{a,b}$ intercepts at a sequence of points p_n, $n \geq 0$, with the vertical grid lines. Let $(n, I_n) \in \mathbb{Z}^2$ be the grid point closest to p_n, and let the following be true:

$$I_{a,b} = \{(n, I_n) : n > 0 \wedge I_n = \lfloor an + b + 0.5 \rfloor\} \tag{7.5}$$

Chain codes $i_{a,b}$ are the differences between successive I_ns such that

$$i_{a,b}(n) = I_{n+1} - I_n = \begin{cases} 0 & if\ I_n = I_{n+1} \\ 1 & if\ I_n = I_{n+1} - 1 \end{cases} \tag{7.6}$$

A DSS is called α-DSS (grid point model), $\alpha \in \{4, 6, 8, 18, 26\}$, if A_α is used for the adjacency relation.

It has been proved that $w = max\{|a|, |b|\}$ defines 8-DSSs. In this thesis, we only consider 8-DSS to calculate the length of the digital arc.

7. THEOREM. *Any set of grid points $D_{a,b,\mu,max\{|a|,|b|\}}$ is the set of grid points of a digital straight line. Conversely, for any rational[1] digital straight line, there exist a, b, and μ such that the set of grid points of the given digital straight lines is $D_{a,b,\mu,max\{|a|,|b|\}}$.*

This is a theorem from (Debled-Rennesson and Reveilles 1995). The authors also proposed a linear online 2D 8-DSS segmentation algorithm.

A DSS algorithm is called online if the algorithm reads successive chain codes $i(0), i(1), ...$ and determines the maximum $k \geq 0$ such that $i(0), i(1), ..., i(k)$ is a DSS, but $i(0), i(1), ..., i(k), i(k + 1)$ is not a DSS.

We assume that the lines are in the first octant; other cases can be mapped into this case by reflection. Let a and b be relatively prime integers, and let μ and w

---

[1] The slope is a rational number.

be integers. We consider lines with slope $0 \leq b/a \leq 1$ $(0 \leq b \leq a)$. Therefore, for an 8-DSS, all the grid points in $D_{a,b,\mu,\omega}$ lie between or on a lower supporting line $bx - ay = \mu$ and an upper supporting line $bx - ay = \mu + max\{|a|, |b|\} - 1 = \mu + a - 1$. Two parallel lines are called a pair of supporting lines of $D_{a,b,\mu,\omega}$ if they are at minimum diagonal distance, and all grid points are located between those lines. Let u_1, u_2 and l_1, l_2 be the points on the upper and lower supporting line, respectively, where index 1 denotes the point q_i $(1 \leq i \leq n)$ with the smallest x-coordinate and index 2 denotes points with the largest x-coordinate. Let $r = bx_{n+1} - ay_{n+1}$ be the remainder of point q_{n+1} with respect to the slope b/a of the given 8-DSS $\{q_1, ..., q_n\} \subset D_{a,b,\mu,-a}$. The algorithm for the computation of digital straight segments in 3D applies two important theorems. The first theorem is published in (Debled-Rennesson and Reveilles 1995).

8. THEOREM. *If* $\mu \leq r < \mu + max\{|a|, |b|\}$, *then* $q_{n+1} \in D_{a,b,\mu,max\{|a|,|b|\}}$. *If* $r = \mu - 1$, *then* $\{q_1, ...q_n, q_{n+1}\}$ *is a segment of a native line with a slope that is defined by vector* u_1q_{n+1}. *If* $r = \mu + max\{|a|, |b|\}$, *then* $\{q_1, ...q_n, q_{n+1}\}$ *is a segment of a native line with a slope that is defined by vector* l_1q_{n+1}. *If* $r < \mu - 1$ *or* $r > \mu + max\{|a|, |b|\}$, *then* $\{q_1, ...q_n, q_{n+1}\}$ *is not a segment of a native line.*

In 1983, (Kim 1983) has proved that a 26-arc is a digital straight segment in 3D if only two projections are digital straight segments in 2D.

9. THEOREM. *A simple 26-arc is a 26-DSS if and only if two of its projections onto the* $(x = 0)$-, $(y = 0)$-, *and* $(z = 0)$-*planes are 8-DSSs.*

We apply above theoretical results to measure the length of arcs in volume images.

Procedure 7.1 gives the pseudo code of 26-DSS recognition. Let X, Y, and Z be the chain codes for x-, y-, and z-coordinates of a 26-arc respectively, N be the size of the chain codes, and L be a list of endpoints of 26-DSS. B stores a boolean value to indicate if the 8-DSS recognition is successful for a projection in one plane. Let p be a voxel of a 26-arc, and p^i, $i \in \{x, y, z\}$, be a pixel that is the projection of p on $(i = 0)$-plane. C is a counter.

Figure 7.4 illustrates an example of segmenting a 26-arc into a sequence of four maximum-length 26-DSSs using Algorithm 7.1 for the following chain codes:

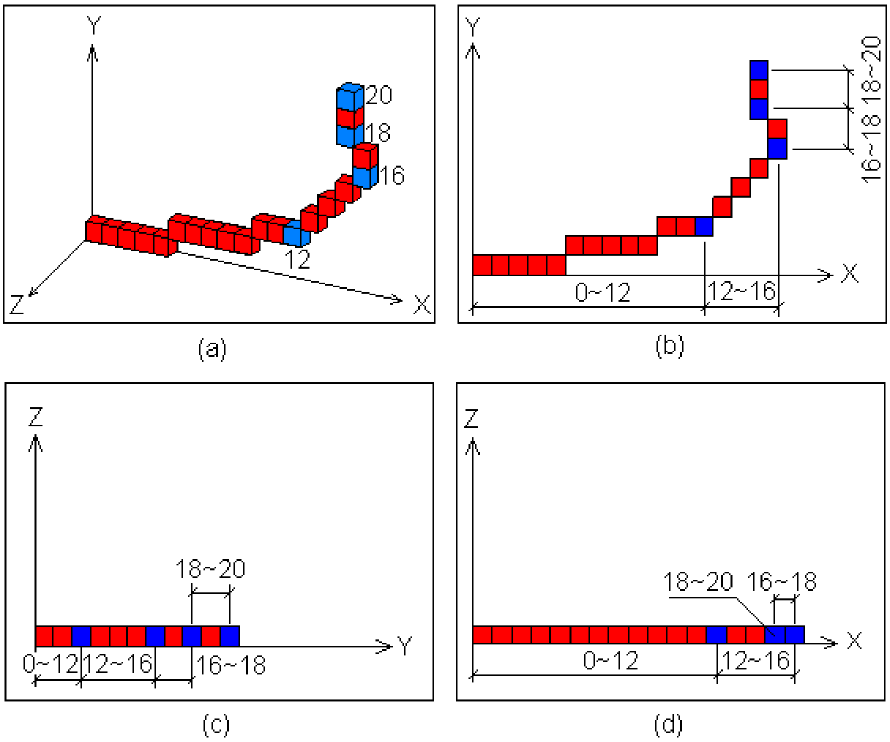
$$11111111111111110100,\ 00001000010011111111,\ 00000000000000000000$$

For the description of the *density of arcs* we calculate the total lengths of digital arcs in subvolumes (corresponding to equally sized subvolumes) and for the whole image based on three different length measurements, the Euclidean distance

```
Require: $X, Y, Z$ {stores chain codes in $x$-axis, $y$-axis, and $z$-axis, respectively.}
Require: $N$ {is the size of the chain code.}
$L \leftarrow \emptyset$ {will store endpoints of 26-DSSs}
for $i = 0$ to $N - 1$ do
    $C \leftarrow 0$ {counts the number of planes where 8-DSS recognition failed.}
    $B_z, B_x, B_y \leftarrow true$ {indicate if 8-DSS recognition is successful in $(z = 0)$-plane, $(x = 0)$-plane, and $(y = 0)$-plane, respectively.}
    if $X[i] = 0$ and $Y[i] = 0$ then {$p_i^z$ overlaps $p_{i-1}^z$}
        $B_z \leftarrow false$ {$(z = 0)$-plane is not considered.}
        $C \leftarrow C + 1$
    if $Y[i] = 0$ and $Z[i] = 0$ then {$p_i^x$ overlaps $p_{i-1}^x$}
        $B_x \leftarrow false$ {$(x = 0)$-plane is not considered.}
        $C \leftarrow C + 1$
    if $X[i] = 0$ and $Z[i] = 0$ then {$p_i^y$ overlaps $p_{i-1}^y$}
        $B_y \leftarrow false$ {$(y = 0)$-plane is not considered.}
        $C \leftarrow C + 1$
    while $true$ do
        if $i = N$ then
            break
        if $B_z$ and $p_i^z$ does not belong to the current 8-DSS in $(z = 0)$-plan then
            $B_z \leftarrow false$
            $C \leftarrow C + 1$
        if $B_x$ and $p_i^x$ does not belong to the current 8-DSS in $(x = 0)$-plane then
            $B_x \leftarrow false$
            $C \leftarrow C + 1$
        if $B_y$ and $p_i^y$ does not belong to the current 8-DSS in $(y = 0)$-plane then
            $B_y \leftarrow false$
            $C \leftarrow C + 1$
        if $C < 2$ then
            $i \leftarrow i + 1$ {Move to next voxel}
        else
            break
    $i \leftarrow i - 1$
    insert $p_i$ into $L$ {take the previous voxel as the endpoint of the current 26-DSS}
```

Algorithm 7.1: *Pseudo code of 26-DSS.*

between nodes, a local estimator and a global estimator. Each different length measurement represents a new cost matrix per volume. The cost is generally (in graph



(a) (b)

(c) (d)

Figure 7.4: *Segmentation into four DSSs.*

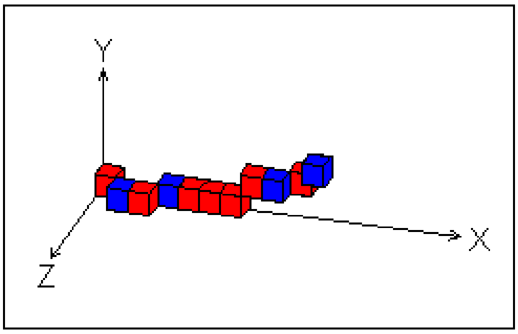
theory) a value which represents the cost of the relationships represented by an arc. In our context those costs are represented by different lengths estimations. The total length of all arcs in one subvolume relative to the total length in the whole image is a measure for the density of arcs in sub-volumes. We can derive features such as maximum, minimum, average, derivations similarly as in the previous section.

Figure 7.5 is an example for three different lengths measurements of one arc. The arc is segmented into four 26-DSSs. The sum of the Euclidean lengths of those four segments is the total DSS-length. The chain codes in x, y, z-direction are

$$1111111111,\ 0000001001,\ 1010000011$$

respectively. The number of voxels is eleven, the Euclidean distance is 10.2 and the 26-DSS length is 10.99.

We apply the above three lengths measurements for the calculation of the total length of arcs per subgraph G_i in subvolumes C_i $(1 \leq i \leq 36)$.



S1: (0, 0, 0) ~ (1, 0, 1)
S2: (1, 0, 1) ~ (3, 0, 0)
S3: (3, 0, 0) ~ (8, 1, 0)
S4: (8, 1, 0) ~ (10, 2, 0)

Figure 7.5: *DSSs along a 3D arc.*

In Figure 7.6, the x-axis corresponds to C_i, $1 \leq i \leq 36$. The y-axis corresponds to the total length of all arcs per subcube. The digital curves in those subvolumes must be relatively straight because there is not much discrepancy between the three lengths measurements.

Figure 7.7 shows the result of computing the total arc length in each subvolume by 26-DSS. If the total length is greater than 195, then subcubes are shaded. The total length of all arcs in those 15 shaded cubes is 65.09% of the total arc length in the whole volume. In eleven subcubes (light gray; close to the main blood vessel)

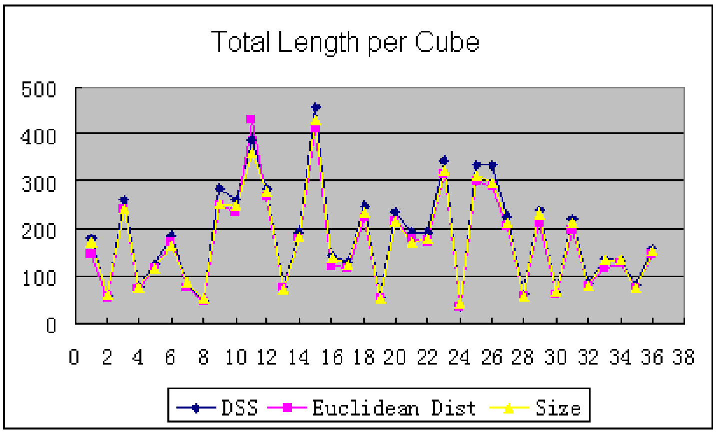


Figure 7.6: *Different length measurements per sub-volume. (Size stands for the number of voxels)*

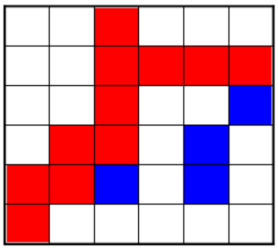


Figure 7.7: *Colored cubes with DSS-length larger than* 195. *Red cubes coincide with the location of the main blood vessel.*

the total arc length is 75.38% of the length in the whole volume.

7.4 Features derived from graph theory

We gave informal descriptions of calculating weights for a given adjacency matrix in the previous section.

Basic definitions such as path or connectedness in graph theory are analogous to definitions in digital picture processing if we map singular elements onto nodes and arcs consisting of regular elements onto edges in an undirected graph.

A path $P_{u,v}$ on a graph $G = (V, E)$ is a set of n nodes $\{p_i, 1 \leq i \leq n\}$ such that $p_1 = u, p_n = v$ and $(p_i, p_{i+1}), 1 \leq i \leq n-1$ are edges. The path is called simple if $p_i \neq p_j$ for any two nodes (i.e., with $i \neq j$) in $P_{u,v}$.

The length $l(p_{u,v})$ of a path $P_{u,v} = p_1, ..., p_n$ is defined by the sum of all lengths $l(p_{u,v}) = \sum_{i=1}^{n-1} l(p_i, p_{i+1})$.

A connected component G' of a graph $G = (V, E)$ is a maximal connected subgraph of G.

The bijective map between connected components in G and connected components in the approximated skeleton pictures suggests that we subdivide the skeleton picture into connected components instead of equally sized cubes to obtain a third group of properties. We can easily derive properties that are well known in graph theory as, for example, the following:

- the *number of connected components* in S equals the number of components in graph G,
- the *radius for each component* is the minimum eccentricity; the eccentricity $e(v)$

of a node v in a finite connected graph G is the greatest distance from a node v to any node of G,

- the *diameter* of a finite connected graph is the maximum distance between any two nodes,
- a closed path is called a *cycle*; the minimum length of all cycles in a graph is called *girth* of the graph $g(G)$, the maximum length of all cycles is its *circumference*,
- a spanning tree is a subgraph of an undirected graph which connects all nodes of the graph and it has no cycles (a tree); an undirected graph has a forest of spanning trees; the minimum spanning tree has a total weight (sum of all weights) smaller or equal to all the total weights of different spanning trees of the same graph; the total weight of the *minimum spanning tree* is a feature for each connected component.

Efficient algorithms for computations of those features are developed and published [see, e.g., (Lee et al. 2001)].

For example, we have identified the location of junctions, the branching index of those junctions and three types of positive real weights for an uniquely defined undirected graph. One version of the *shortest path problem* is finding the shortest path between two distinguished junctions with the same branching index. Algorithms to solve this problem are Dijkstra's algorithm or Bellman-Ford's algorithm [see, e.g., (Brassard and Bratley 1996)]. Both algorithms are applicable for directed graphs. We have to modify the given undirected graph accordingly.

7.5 Summary

Chapter 7 proposes three groups of features for characterizing skeleton pictures represented by abstract curve graphs. The first group describes the density, complexity and uniformity of junctions in equally sized sub cubes. The second group proposes different length measurements of digital arcs to describe the density and uniformity of digital arcs in subvolumes. The third group consists of suitable features which have been used for the description of weighted graphs and which can be adapted.

Chapter 8

Conclusions

This chapter concludes this thesis by presenting conclusions and ideas for improvements.

8.1 Conclusions

This thesis reviews and extends two approaches for the generation of curve-like structures in digital pictures inspired by 3D confocal microscope images of human brain tissue.

We discuss distance transforms as a classic tool for the generation of geometric skeletons. Envelope algorithms are explained in more detail because they calculate exact Euclidean distance maps in optimal time and in arbitrary dimensions. New details for the computation of the reverse distance transform and the skeleton extraction are explained. We demonstrate the strong relationship between the reverse distance transform and the medial axis extraction. Geometric skeleton definitions are not consistent in the literature. In general, a main disadvantage is that geometric skeletons are not connected. The example of a sphere shows that in one extreme, the extracted medial axis can be an unorganized cloud of elements. We propose a method to reduce the number of those elements. The result is not satisfactory because in some cases elements of the medial axis have a large Euclidean distance to each other. The discoveries for the computation of medial axis do not change the fact that the topologic structure has not been effectively captured in a suitable format such as an abstract curve graph. For the generation of curve-like structures, postprocessing algorithms are required. Medial axis extractions based on distance transforms are successful in well defined applications, where additional conditions support the generation of curves and arcs.

The thesis demonstrates also that envelope algorithms, for the computation of geometric skeletons, are an interesting example for the application of classical mathematic tools from the continuous space in digital picture processing and vice versa.

We have given our preference to topologic skeletons above the geometric skeletons for our specific application because our emphasis is the detection of the branching structure for subsequent analysis. For this purpose we review topology preserving thinning methods as a general concept. Simple element detection or simple set detection are central parts of topology preserving thinning. The literature offers a long list of characterizations of simple elements or simple sets based on different models. One of the key contributions of this thesis is that we prove the equivalence of different characterizations, and we use the Euler characteristic of the P-attachment set to characterize non-simple elements in 3D space. The application of this new characterization leads to a time efficient construction of look-up-tables. We describe the modification of an existing six subiteration sequential thinning algorithm.

The classic definition of branching indices in Euclidean space has been adapted for the 3D orthogonal grid. We propose a classification of elements in 3D topologic skeletons and we introduce junctions. The number of branch elements is unstable for noisy surfaces. Morphologic operators (sequences of opening and closing) as preprocessing procedures for topologic thinning provide better accuracy for the computation of the number and complexity of branch elements. The classification of skeletal elements is required for the unique construction of abstract curve graphs. We describe labeling algorithms for subsequent length measurements which lead to weighted undirected graphs. We propose the centroid of a junction to be the geometric location in combination with a 3D DSS-algorithm as a global length estimator.

The introduction of new features such as uniformity and density is based on the subdivision of volumes into equally sized cubes. They describe the branching structure in the whole 3D picture and how densely distributed curves are located in pairwise disjoint subcubes.

The representation of skeleton pictures as abstract curve graphs allows the adaption of graph algorithms. We propose the subdivision of skeleton pictures into connected components for the extraction of graphtheoretic properties.

8.2 Ideas

We have seen that EVA-algorithms are useful for the computation of the squared Euclidean distance transform, the reverse distance transform and the medial axis. Both presented medial axis definitions do not help to compute the “optimal” medial axis in a sense that only the minimum number of elements for reconstruction are determined. Further studies of the properties of the envelopes of the families of parabolas could lead to a solution for this problem. It is also interesting to study those properties with respect to connectedness of the resulting elements of the medial axis and subsequent classifications of those elements. We discussed properties of the medial axis of one disk (based on one definition) with respect to increasing grid resolution. An open question is: can we discover general theorems about properties of medial axes calculated with different methods with respect to increasing grid resolution? In other words: do existing methods improve with increased grid resolution and reduce problems which lie in the discrepancies between continuous maps and discrete picture maps?

We discussed some details of a directional 3D thinning algorithm. All directional thinning algorithms have the disadvantage that the locations of resulting skeletons are dependent on the start direction. Parallel thinning algorithms based on multiple element definitions generate sets of skeletal elements which are not ″thin″ in general. We should investigate the question how an increased grid resolution improves the location of skeletal curves for different thinning methods.

We analyzed some properties of junctions in skeletal curves after thinning. The study could continue with the final goal to find general dependencies between cardinalities and branching indices, or influences of specified morphologic operations as preprocessing steps for topologic thinning and number of junctions.

Appendix A

Selected research activities by the author

A.1 Visiting Positions

- General Member of the Institute of Mathematics and its Applications (IMA), University of Minnesota, Minneapolis, USA, February 2006 - May 2006.
- Exchange Scholar of the UoA Faculty of Sciences, National Taiwan Normal University, Taipei, Taiwan, November - December 2005.
- Visiting researcher at the Departamento de Control Automatico del CINVESTAV-IPN, Mexico City, February 2005.

A.2 Journal Articles

- Bodo Rosenhahn, Uwe Kersting, Katie Powell, Reinhard Klette, Gisela Klette, and Hans-Peter Seidel. A System for Articulated Tracking Incorporating a Clothing Model. (submitted to Machine Vision and Applications, the main contribution of this paper is not discussed in this thesis.)
- Gisela Klette: A Comparative Discussion of Distance Transformations and Simple Deformations in Digital Image Processing. Machine Graphics & Vision, 12:235-256, 2003.

A.3 Conference Publications (Talks)

- G. Klette: Branch Voxels and Junctions in 3D skeletons. Proc. Int. Conf. 10th IWCIA 2006, Berlin, Germany, Lecture Notes in Computer Science 4040, Springer, Berlin, pages 34-44,2006.
- G. Klette and M. Pan: Characterization of Curve-Like Structures in 3D Medical Images. Proc. Int. Conf. Image and Vision Computing New Zealand 2005, Dunedin, New Zealand, pages 357-362, 2005.

- G. Klette and M. Pan: 3D Topological Thinning by Identifying Non-simple Voxels. Proc. Int. Conf. 10th IWCIA 2004, Auckland, New Zealand, Lecture Notes in Computer Science 3322, Springer, Berlin, pages 164-174, 2004.

- A. Stilinovic, C. R. Green, R. Klette, S. Franke, G. Klette and D. L. Becker: Texture Analysis of Collagen Fibers in Scar Tissue". Proc. Int. Conf. Image and Vision Computing New Zealand" 2004, Akaroa, New Zealand, pages 185-191, 2004.

- X. Lin, G. Klette, R. Klette, J. Craig and S. Dean: Accurately Measuring the Size of the Pupil of the Eye. Proc. Int. Conf. Image and Vision Computing New Zealand 2003, Palmerston North, New Zealand, pages 221–226, 2003. (The subject of this paper is not discussed in this thesis.)

- G. Klette: Characterizations of simple pixels in binary images. Proc. Int. Conf. Image and Vision Computing New Zealand 2002, Auckland, New Zealand, pages 227-232, 2002.

A.4 Conference Publications (Posters)

- G. Klette: Branch Voxels and Junctions in 3D Skeletons of Confocal Microscope Images of Human Brain Tissue. Int. Workshop Natural Images, University of Minnesota, Minneapolis, USA, March 2006.

- L. Deng, D. J. Bertinshaw, R. Klette, G. Klette and D. Jeffries: "Footprint identification of weta and other insects". Proc. Int. Conf. "Image and Vision Computing New Zealand", Akaroa, New Zealand, pages 191-196, 2004.

- G. Klette: "Simple Points in 2D and 3D Binary Images". Proc. Int. Conf. CAIP 2003, Groningen, The Netherlands, Lecture Notes in Computer Science 2756, Springer, Berlin, pages 57-64, 2003.

A.5 Invited Talks

- University of Tennessee Knoxville, USA, 21 April 2006: Curve structures in brain tissue.

- Institute Colloquium at NCCU (National Chung Cheng University), Chia-Y, Taiwan December 2005: 3D curve analysis.

- Departmental seminar at NTU (National Taiwan University), Taipei, Taiwan, November 2005: 3D image analysis of human brain tissue.
- Departmental seminar at ICI at NTNU (National Taiwan Normal University), Taipei, Taiwan November 2005: 3D image analysis of human brain tissue.
- IPCV research group seminar, NTNU (National Taiwan Normal University), Taipei, Taiwan, November 2005: 3D image analysis of human brain tissue.
- Multimedia University Cyberjaya, Malaysia, 8.6.2005: Shape simplification methods.
- CIMAT (Center for Mathematical Research) Guanajuato, Mexico, 17 February 2005: Comparisons between thinning methods.
- Departamento de Control Automatico del CINVESTAV-IPN, Mexico City, 14 February 2005: Algorithms for shape simplifications in digital image processing.
- NTHU, Taiwan, 8 September 2004: Comparisons of skeletonization methods.
- NCCU, Taiwan, 1 September 2004: Measuring the pupil of the human eye.

A.6 Seminars

- Open day, CITR, University of Auckland, New Zealand, 15 October 2005: 3D skeletons for the analysis of human brain tissue.
- University Groningen, The Netherlands, 18 July 2005: Identification and analysis of branching points based on 3D skeletons
- Seminar, CITR, University of Auckland, New Zealand, 24 May 2004: The definition of the digital linking number - a topological invariant.
- Open day, CITR, University of Auckland, New Zealand, 20 February 2004: Measuring the size of pupils.
- Seminar, CITR, University of Auckland, New Zealand, 17 November 2003: Accurate measurements of the size of the pupil of the eye.
- Technical University Dresden, 8 September 2003: Simple points in 3D images.
- German Air and Space Institute (DLR), Berlin, Germany, 1 September 2003: Simple points in 3D images.

- Seminar, CITR, University of Auckland, New Zealand, 4 August 2003: Simple points in 3D binary images.
- University Goettingen, Germany, 1 July 2002: Characterization Theorems for thinning approaches.
- University Groningen, The Netherlands, 14 June 2002: Distance transforms and simple transformations for 2D images
- University Kiel, Germany, 2 June 2002: Distance transforms and simple transformations for 2D images
- Seminar, CITR, University of Auckland, New Zealand, 5 April 2002: Skeletons in digital image processing.

Bibliography

Aleksandrov, P.: 1956, *Combinatorial Topology, Volume 1.*, Graylock Press, New York.

Bailey, D.: 2004, An efficient euclidean distance transform., *in: Int. Workshop on Combinatorial Image Analysis, LNCS 3322* pp. 394–408.

Bertrand, G.: 1994, Simple points, topological numbers and geodesic neighborhoods in cubic grids., *Pattern Recognition Letters* **15**, 1003–1011.

Bertrand, G. and Aktouf, Z.: 1994, A 3d thinning algorithm using subfields., *in: Proc. SPIE Conf. on Vision Geometry III* **2356**, 113–124.

Bertrand, G. and Malandain, G.: 1994, A new characterization of three-dimensional simple points., *Pattern Recognition Letters* **15**, 169–175.

Blum, H.: 1962, An associative machine for dealing with the visual field and some of its biological implications., *in: Biological Prototypes and Synthetic Systems, volume 1, ed.: Bernard,E.E. and Kate,M.R.* pp. 244–260.

Blum, H.: 1967, A transformation for extracting new descriptors of shape., *in: Models for the Perception of Speech and Visual Form, ed.: Wathen-Dunn,W.* pp. 362–380.

Borgefors, G.: 1984, Distance transform inarbitrary dimensions., *Computer Vision Graphics Image Processing* **27**, 321–345.

Borgefors, G.: 1996, On digital distance transforms in three dimensions., *Computer Vision Image Understanding* **64**, 368–376.

Borgefors, G.: 2005, Digital distance transforms in 2d, 3d, and 4d., *in: Handbook of Pattern Recognition and Computer Vision* pp. 157–177.

Borgefors, G., Nystroem, I. and Sanniti Di Baja, G.: 1996, Surface skeletonization of volume objects., *in: Advances in Structural and Syntactical Pattern Recognition* pp. 251–259.

Borgefors, G., Nystroem, I. and Sanniti Di Baja, G.: 1999, Computing skeletons in three dimensions., *Pattern Recognition* **32**, 1225–1236.

Brassard, G. and Bratley, P.: 1996, *Fundamentals of algorithmics.*, Prentice-Hall, New Jersey.

Breu, H., Gil, J., Kirkpatrick, D. and Werman, M.: 1995, Linear-time euclidean distance transform algorithms., *pami* **17**, 529–533.

Calabi, L. and Hartnett, W.: 1968, Shape recognition, prairie fires, convex deficiencies and skeletons., *American Mathematical Monthly* **75**, 335–342.

Coeurjolly, D.: n.d., D-dimensional reverse euclidean distance transformation and euclidean medial axis extraction in optimal time., *in: Discrete Geometry for Computer Imagery, LNCS 2886* .

Coeurjolly, D. and Montanvert, A.: 2005, Optimal separable algorithms to compute the reverse euclidean distance transformation and discrete medial axis in arbitrary dimension., *accepted for publication in pami* .

Danielsson, P.: 1980, Euclidean distance mapping., *Computer Graphics Image Processing* **14**, 227–248.

Debled-Rennesson, I. and Reveilles, J.: 1995, A linear algorithm for segmentation of digital curves., *tpami* **9**, 635–662.

Felzenszwalb, P. and Huttenlocher, D.: 2004, Distance transforms of sampled functions., *Cornell Computing and Information Science Technical Report* .

Fourey, S. and Malgouyres, R.: 2003, A concise characterization of 3d simple points., *Discrete Applied Mathematics* **125**, 59–80.

Freeman, H.: 1961, Techniques for the digital computer analysis of chain-encoded arbitrary plane curves., *in: Nat. Electronics Conference.* **17**, 421–432.

Gagvani, N. and Silver, D.: 1997, Parameter controlled skeletonization of three dimensional objects., *Technical Report CAIP-TR-216* pp. 1–19.

Gau, C. and Kong, T.: 2002, 4d minimal non-simple sets., *in: Discrete Geometry for Computer Imagery, LNCS 2301* pp. 81–91.

Gau, C. and Kong, T.: 2003, Minimal non-simple sets in 4d binary images., *Graphical Models* **65**, 112–130.

Ge, Y. and Fitzpatrick: 1996, On the generation of skeletons from discrete euclidean distance maps., *pami* **18**, 1055–1066.

Glantz, R. and Kropatsch, W.: 2001, Skeletons in the framework of graph pyramids., *in: Visual Form, LNCS 2059* pp. 186–195.

Gonzalez, R. and Woods, R.: 2002, *Digital Image Processing.*, Prentice-Hall, New Jersey.

Hall, R.: 1996, Parallel connectivity-preserving thinning algorithms., *in: Topological Algorithms for Digital Image Processing (eds.: Kong,T.Y. and Rosenfeld,A.)* pp. 145–179.

Haralick, R. and Shapiro, L.: 1992, *Computer and Robot Vision Vol. 1.*, Addison-Wesley, New York.

Haxhimusa, Y. and Kropatsch, W.: 2003, Hierarchy of partitions with dual graph contraction., *in: Pattern Recognition Symposium, LNCS 278* pp. 338–345.

Hesselink, W., Visser, M. and Roerdink, J.: 2005, Euclidean skeletons of 3d data sets in linear time by the integer medial axis transform., *in: Mathematical Morphology: 40 Years On* pp. 259–268.

Hilditch, C.: 1969, Linear skeletons from square cupboards., *in: Machine Intelligence (eds.: Meltzer, B. and Mitchie, D.)* pp. 403–420.

Hirata, T.: 1996, A unified linear-time algorithm for computing distance maps., *Information Processing Letters* **58**, 129–133.

Khalimsky, E.: 1986, Pattern analysis of ?-dimensional digital images., *in: In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics* pp. 1559–1562.

Kim, E.: 1983, Three-dimensional digital line segments., *tpami* **5**, 231–234.

Klette, G.: 2002, Characterizations of simple pixels in binary images., *in: Image and Vision Computing New Zealand 2002* pp. 227–232.

Klette, G.: 2003a, A comparative discussion of distance transforms and simple deformations in digital image processing., *Machine Graphics & Vision* **2**, 235–256.

Klette, G.: 2003b, Simple points in 2d and 3d binary images., *in: Computer Analysis of Images and Patterns, LNCS 2756* pp. 57–64.

Klette, G.: 2006, Branch voxels and junctions in 3d skeletons., *in: Int. Workshop on Combinatorial Image Analysis, LNCS 4040* pp. 34–44.

Klette, G. and Pan, M.: 2004, 3d topological thinning by identifying non-simple voxels., *in: Int. Workshop on Combinatorial Image Analysis, LNCS 3322* pp. 164–175.

Klette, R. and Rosenfeld, A.: 2004, *Digital Geometry.*, Morgan Kaufmann, San Francisco.

Kong, T.: 1989, A digital fundamental group., *Comput. and Graphics* **13**, 159–166.

Kong, T.: 1995, On topology preservation in 2-d and 3-d thinning., *Int. J. for Pattern Recognition and Artificial Intelligence* **9**, 813–844.

Kovalevsky, V.: 1989, Finite topology as applied to image analysis., *Computer Vision, Graphics, and Image Processing* **46**, 141–161.

Kropatsch, W., Ion, A., Haxhimusa, Y. and Flanitzer, T.: 2006, The eccentricity transform of a digital shape., *in: Discrete Geometry for Computer Imagery, LNCS 4245* pp. 437–448.

Latecki, L. and Eckardt, U.: 1995, Well-composed sets., *Computer Vision Image Understanding* **61**, 70–83.

Lee, L.-Q., Lumsdaine, A. and Siek, J.: 2001, *The Boost Graph Library.*, Addison Wesley Professional.

Lee, T., Kashyap, R. and Chu, C.: 1994, Building skeleton models via 3d medial surface/axis thinning algorithms., *gmip* **56**, 462–478.

Listing, J.: 1861, Der census räumlicher complexe oder verallgemeinerungen des euler'schen satzes von den polyëdern., *Abhandlungen der Mathematischen Classe der Königlichen Gesellschaft der Wissenschaften zu Göttingen* .

Lohou, C.: 2001, Contribution to the topological analysis of images: study of thinning algorithms for 2d or 3d images, according to either a digital topology approach or a discrete topology approach., *Thesis* pp. 1–365.

Lohou, C. and Bertrand, G.: 2002, A new 3d 6-subiteration thinning algorithm based on p-simple points., *in: Discrete Geometry for Computer Imagery, LNCS 2301* pp. 102–113.

Ma, C.: 1994, On topology preservation in 3d thinning., *CVGIP: Image Understanding* **59**, 328–339.

Ma, C.: 1995, A 3d fully parallel 3d thinning algorithm for generating medial faces., *Pattern Recognition Letters* **16**, 83–87.

Malandain, G. and Bertrand, G.: 1992, Fast characterization of 3d simple points., *in: Int. Conf. on Pattern Recognition* pp. 232–235.

Marchand-Maillet, S. and Sharaiha, Y.: 2000, *Binary Digital Image Processing.*, Academic Press, Cambridge.

Meijster, A., Roerdink, J. and Hesselink, W.: 2000, A general algorithm for computing distance transforms inlinear time., *in: Mathematical Morphology and its Applications to Image and Signal Processing* pp. 331–340.

Menger, K.: 1932, *Kurventheorie.*, Teubner, Leipzig, Germany.

Montanari, U.: 1968, A method of obtaining skeletons using a quasi-euclidean distance., *Journal of the ACM* **15**, 600–624.

Montanari, U.: 1969, Continious skeletons from digitized images., *Journal of the ACM* **16**, 534–549.

Mullikin, J.: 1992, The vector distance transform in two or three dimensions., *Graphical Models and Image Processing* **54**, 420–437.

Niblack, W., Gibbons, P. and Kapson, B.: 1992, Generating skeletons and center lines from the distance transform., *Graphical Models and Image Processing* .

Niethammer, M., Kalies, W., Mischaikow, K. and Tannenbaum, A.: 2005, On the dedection of simple points in higher dimensions using cubical homology., *Accepted and to appear in the IEEE Transactions on Image Processing* .

Palagyi, K. and Kuba, A.: 1998, A 3d 6-subiteration thinning algorithm for extracting medial lines., *Pattern Recognition Letters* **19**, 613–627.

Palagyi, K. and Kuba, A.: 1999, A parallel 3d 12-subiteration thinning algorithm., *Graphical Models and Image Processing* **61**, 199–221.

Palagyi, K., Sorantin, E., Balogh, E., Kuba, A., Halmai, B., Erdohelyi, B. and Hausegger, K.: 2001, A sequential 3d thinning algorithm and its medical applications., *in: Int. Conf. Information Processing in Medical Imaging, LNCS 2082* pp. 409–415.

Palagyi, K., Tschirren, J. and Sonka, M.: 2003, Quantitative analysis of intrathoracic airway trees: methods and validation., *in: Int. Conf. Information Processing in Medical Imaging, LNCS 2732* pp. 222–232.

Pavlidis, T.: 1980, A thinning algorithm for discrete binary images., *Computer Graphics Image Processing* **13**, 142–157.

Ragnemalm, I.: 1993, The euclidean distance transform., *Dissertation No.304, Linkoping University, Linkoping, Sweden* .

Remy, E. and Thiel, E.: 2005, Exact medial axis with euclidean distance., *Image and Vision Computing* **23**, 167–175.

Ronse, C.: 1986, A topological characterization of thinning., *Theoretical Computer Science* **43**, 31–41.

Rosenfeld, A.: 1970, Connectivity in digital pictures., *Comm.ACM* **17**, 146–160.

Rosenfeld, A.: 1974, Adjacency in digital pictures., *Information and Control* **26**, 24–33.

Rosenfeld, A.: 1975, A characterization of parallel thinning algorithms., *Information and Control* **29**, 286–291.

Rosenfeld, A. and Klette, R.: 2002, Topologies for binary or multi-level images - a review., *6th Joint Conf. Inform. Sciences, Research Triangle Park, North Carolina* .

Rosenfeld, A., Kong, T. and Nakamura, A.: 1998, Topology-preserving deformations of two-valued digital pictures., *Graphical Models and Image Processing* **60**, 24–34.

Rosenfeld, A. and Pfaltz, J.: 1966, Sequential operations in digital picture processing., *Comm.ACM* **13**, 471–494.

Rutovitz, D.: 1966, Pattern recognition., *J. Royal Statist. Soc.* **129**, 504–530.

Saha, P., Chanda, B. and Majumder, D.: 1991, Principles and algorithms for 2d and 3d shrinking., *Technical Report TR/KBCS/2/91, NCKBCS Library, Indian Statistical Institute, Calcutta, India* .

Saha, P. and Chaudhuri, B.: 1994, Detection of 3d simple points for topology preserving transformations with application to thinning., *tpami* **16**, 1028–1032.

Saha, P., Chaudhuri, B. and Majumder, D.: 1997, A new shape-preserving parallel thinning algorithm for 3d digital images., *Pattern Recognition Letters* **30**, 1939–1955.

Saito, T. and Toriwaki, J.: 1994, New algorithms for euclidean distance transformation of an n-dimensional digitized picture with applications., *Pattern Recognition* **27**, 1551–1565.

Sanniti di Baja, G.: 1994, Well-shaped, stable and reversible skeletons from the (3,4)-distance transform., *J. Visual Comm. Image Repres.* **5**, 107–115.

Serra, J.: 1982, *Image Analysis and Mathematical Morphology.*, Academic Press, New York.

Shapiro, B., Pisa, J. and Sklansky, J.: 1981, Skeleton generation from x,y boundary sequences., *Computer Graphics and Image Processing* **15**, 136–153.

Sintorn, I. and Borgefors, G.: 2004, Weighted distance transform for volume images digitized in elongated voxel grids., *Pattern Recognition Letters* **25**, 571–580.

Sonka, M., Hlavac, V. and Boyle, R.: 1999, *Image Processing, Analysis and Machine Vision, 2 nd Edition.*, PWS Publishing.

Toriwaki, J. and Mori, K.: 2001, Distance transformation and skeletonization of 3d pictures and their applications to medical images., *in: Digital and Image Geometry, LNCS 2243* pp. 412–429.

Urysohn, P.: 1923, Uber die allgemeinen cantorischen kurven., *in: Annual meeting, Deutsche Mathematiker Vereinigung* .

Yokoi, S., Toriwaki, J. and Fukumura, T.: 1975, An analysis of topological properties of digitized binary pictures using local features., *Computer Graphics and Image Processing* **4**, 63–73.

Zhang, T. and Suen, C.: 1984, A fast parallel algorithm for thinning digital patterns., *Comm. ACM* **27**, 236–239.

Index