Topology Preservation and Thinning

Kálmán Palágyi

Department of Image Processing and Computer Graphics
Institute of Informatics, University of Szeged, Hungary

Dissertation for the Doctoral Degree of the
Hungarian Academy of Sciences

2019

Preface

The two principal application areas of digital image processing are improvement of visual information for human interpretation, and processing of image data for autonomous computer vision. The generic model of a modular computer vision system comprises of image acquisition, preprocessing (i.e., image restoration and enhancement), segmentation (i.e., subdividing an image into its objects), object or shape representation (i.e., extracting shape features from segments), classification and/or interpretation. In this dissertation, our attention is focussed on shape representation, particularly computationally efficient extraction of 'reliable' (i.e., geometrically and topologically correct) skeleton-like features from 2D and 3D objects.

Skeletonization (i.e., extracting skeleton-like features from discrete objects in a topology-preserving ways) goes back to Blum's suggestion. He defined the skeleton (i.e., a region-based shape feature of a continuous object) via the medial axis transform. There is a fairly general agreement that skeletonization procedures play a key role in a broad range of problems in image processing and computer vision.

Digital topology deals with the 'topological' nature (particularly, properties of 2D and 3D objects that involve the concept of connectedness, but do not depend on size or shape), and with algorithms that compute or preserve such properties. The importance of topological properties and algorithms show an upward tendency in the analysis of 2D and 3D digital images.

This dissertation presents a selection of my results concerning digital topology, topology-preserving thinning (i.e., a skeletonization technique), and a complex medical application of one of my 3D thinning algorithms. The results included in this work all originate from a period well after defending my PhD dissertation in the year 2000.

The introductory chapter (i.e., Chapter 1) presents the underlying results, and reviews the most important applications of skeletonization. It is followed by three chapters (i.e., Chapters 2–4) that summarize my results.

Chapter 2 reviews some theoretical results concerning diversified topological problems. These results were achieved by the author and Péter Kardos (i.e., the author's former PhD student and colleague now). Note that the reported results are absolutely not autotelic, they provide methods of verifying that an operator preserves the topology, allow us to generate topology-preserving operators, and provide computationally efficient thinning algorithms.

In Chapter 3, only some selected results concerning thinning are presented. The collaborators were my former PhD students and colleagues now: Péter Kardos and Gábor Németh. This chapter presents a computationally efficient general implementation scheme for thinning algorithms, a safe technique for designing topology-preserving parallel thinning algorithms, four pairs of equivalent sequential and parallel 3D subiteration-based surface-thinning algorithms, and two maximal sequential 3D curve-thinning algorithms.

Chapter 4 describes our complex method for quantitative analysis of pulmonary airway trees that is based on one of the author's 3D curve-thinning algorithm. Among others, the collaborators were Eric A. Hoffman and Milan Sonka (i.e., outstanding researchers at The University of Iowa, Iowa City, IA, USA).

To my beloved wife and to the memory of Attila Kuba

Acknowledgements

The results reviewed in this dissertation could not have been achieved without the contributions of the coauthors: Emese Balogh (*Univ. of Szeged*), Kenneth C. Beck (*The Univ. of Iowa, Iowa City, IA, USA*), Reinhard Beichel (*Graz Univ. of Technology, Austria*), Horst Bischof (*Graz Univ. of Technology, Austria*), Alexander Bornik (*Graz Univ. of Technology, Austria*), Deokiee Chon (*The Univ. of Iowa, Iowa City, IA, USA*), Balázs Erdőhelyi (*Univ. of Szeged*), Attila Fazekas (*Univ. of Debrecen*), Gerhard Friedrich (*Univ. Hospital Graz, Austria*), Bernhard Geiger (*Siemens Corporate Research Princeton Inc., Princeton, NJ, USA*), Junfeng Guo (*The Univ. of Iowa, Iowa City, IA, USA*), Csongor Halmai (*Univ. of Szeged*), Klaus Hausegger (*Univ. Hospital Graz, Austria*), Eric A. Hoffman (*The Univ. of Iowa, Iowa City, IA, USA*), Christian Janko (*Graz Univ. of Technology, Austria*), Péter Kardos (*Univ. of Szeged*), Karl Kiesler (*Univ. Hospital Graz, Austria*), Domagoj Kovacevic (*Univ. of Zagreb, Croatia*), György Kovács (*Univ. of Debrecen*), Attila Kuba (*Univ. of Szeged*), Franz Lindbichler (*Univ. Hospital Graz, Austria*), Sven Loncaric (*Univ. of Zagreb, Croatia*), László Martonossy (*Univ. of Szeged*), Brian Matejek (*Harvard Univ., Cambridge, MA, USA*), Darius Mohadjer (*Medical Univ. of Graz, Austria*), Geoffrey McLennan (*The Univ. of Iowa, Iowa City, IA, USA*), Gábor Németh (*Univ. of Szeged*), László G. Nyúl (*Univ. of Szeged*), Krisztián Ollé (*Univ. of Szeged*), Hanspeter Pfister (*Harvard Univ., Cambridge, MA, USA*), Thomas Pock (*Graz Univ. of Technology, Austria*), Joseph M. Reinhardt (*The Univ. of Iowa, Iowa City, IA, USA*), Bernhard Reitinger (*Graz Univ. of Technology, Austria*), Andrea Ruppert (*Univ. Hospital Graz, Austria*), László Ruskó (*Univ. of Szeged*), Osama Saba (*The Univ. of Iowa, Iowa City, IA, USA*), Shaher Samrah (*The Univ. of Iowa, Iowa City, IA, USA*), Hidenori Shikata (*The Univ. of Iowa, Iowa City, IA, USA*), Brett A. Simon (*The Johns Hopkins Univ., Baltimore, MD, USA*), Milan Sonka (*The Univ. of Iowa, Iowa City, IA, USA*), Erich Sorantin (*Univ. Hospital Graz, Austria*), Marco Subasic (*Univ. of Zagreb, Croatia*), Juerg Tschirren (*The Univ. of Iowa, Iowa City, IA, USA*), Anna Vilanova i Bartrolí (*Vienna Univ. of Technology, Austria*), Xueying Wang (*Harvard Univ., Cambridge, MA, USA*), Donglai Wei (*Harvard Univ., Cambridge, MA, USA*), Georg Werkgartner (*Univ. Hospital Graz, Austria*), Jinglin Zhao (*Harvard Univ., Cambridge, MA, USA*), Roman Zotter (*Graz Univ. of Technology, Austria*).

I gratefully acknowledge their support.

Contents

1 Introduction **1**
1.1 Basic Concepts of Digital Topology 2
1.1.1 Grids 2
1.1.2 Adjacency Relations 3
1.1.3 Binary Digital Pictures 5
1.2 Topology-Preserving Operators 7
1.2.1 Operators 7
1.2.2 Topology Preservation in Reductions 7
1.2.3 Simple Points 9
1.2.4 Sufficient Conditions 11
1.3 Skeletonization 16
1.3.1 Skeleton-Like Features in 2D and 3D 18
1.3.2 Skeletonization Techniques 19
1.3.3 Applications of Skeletonization 23

2 Topology Preservation **25**
2.1 Sufficient Conditions for 2D Pictures 26
2.1.1 Configuration-Based Conditions for Reductions 26
2.1.2 Point-Based Conditions for Reductions 28
2.2 Point-Based Sufficient Conditions for 3D Reductions 32
2.3 Equivalent Sequential and Parallel Operators 34
2.4 Relationships Among Conditions for Reductions 40
2.4.1 Hereditarily Simple Sets and P-Simple Sets 40
2.4.2 Configuration-Based and Point-Based Conditions 40
2.4.3 Configuration-Based Conditions and P-Simple Sets 41
2.4.4 P-Simple Sets and General-Simple Deletion Rules 42
2.4.5 Summarized Relationships 43

3 Advanced Thinning **45**
3.1 A General Implementation Scheme 48

3.2 3D Parallel Thinning Algorithms Derived from Sufficient Conditions for Topology Preservation 51
3.2.1 Fully Parallel Algorithms 52
3.2.2 Subiteration-Based Algorithms 53
3.2.3 Subfield-Based Algorithms 54
3.2.4 Implementation and Results 57
3.3 Equivalent Thinning 62
3.4 Maximal 3D Curve-Thinning Algorithms 67
3.4.1 An Endpoint-Based Sequential Curve-Thinning Algorithm 67
3.4.2 Curve-Thinning with Endpoint-Rechecking 68

4 Quantitative Analysis of Pulmonary Airway Trees 71
4.1 Airway Segmentation 73
4.2 Correction of the Segmented Tree 74
4.3 Root Detection 75
4.4 Producing Centerline 76
4.5 Pruning 77
4.6 Smoothing 80
4.7 Branch-point Identification 81
4.8 Generating Formal Tree Structure 82
4.9 Tree Partitioning 83
4.10 Calculating Associated Measures 85
4.11 Tree Matching 87
4.12 Re-sampling 2D Slices 89

5 Summary of New Scientific Results 91

Bibliography 95

Chapter 1

Introduction

This chapter reviews the fundamental concepts and surveys the major theoretical results that we need later on.

Section 1.1 discusses the basic concepts of digital topology including discrete grids, adjacency relations in regular planar grids and the 3D cubic grid, and binary digital pictures.

Section 1.2 surveys sufficient conditions to prove that an algorithm (or an operator) preserves the topology for all possible pictures.

Lastly, in Section 1.3, we review the skeleton of a continuous object, the skeleton-like features to be extracted from 2D and 3D digital objects, the major skeletonization techniques, and the applications of skeletonization.

1.1 Basic Concepts of Digital Topology

Digital topology deals with the topological properties of *digital pictures* [137]. Here, we apply the fundamental concepts of digital topology as reviewed by Kong and Rosenfeld [137, 139]. Note that there are other approaches that are based on cellular/cubical complexes [142] or polytopal complexes [141], but we insist on the 'historical paradigm'.

Our attention is focussed on the digital topology of *binary pictures* that assign a color of *black* (value 1) or *white* (value 0) to each point of the given grid [137, 184].

1.1.1 Grids

A number of different *grids* (i.e., sampling schemes) have been considered for performing picture processing operations. The *Voronoi diagram* [13] generated by the given grid points is a partitioning of the continuous space into regions that are called *cells*. The cell associated with a grid point p is composed of all points in the Euclidean space that are at least as close to p as to any other grid point. Note that a cell is always a closed convex polygon (in 2D) or a polyhedron (in the 3D case).

A *regular Voronoi diagram* of the 2D Euclidean space is formed by tiling the plane with uniform cells having equal angles, and sides of the same length. There are exactly three polygons that can form such regular Voronoi diagrams the equilateral triangle, the square, and the regular hexagon [133, 137, 138, 184] (see Figure 1.1). The points in the *triangular grid*, the *square grid*, and the *hexagonal grid* are the centroids of the cells/polygons and they are denoted by $\mathcal{T}$, $\mathbb{Z}^2$, and $\mathcal{H}$, respectively.

Throughout this dissertation, the three regular planar grids are illustrated by their associated Voronoi diagrams (see Figure 1.1), and the notation $\mathcal{V}$ means that $\mathcal{V}$ belongs to $\{\mathcal{T}, \mathbb{Z}^2, \mathcal{H}\}$.

Note that the dual of a 2D Voronoi diagram is formed by taking the center of each cell as a vertex and joining the centers of cells that share an edge. It can be easily seen that the triangular and the hexagonal grids are duals of each other (see Figure 1.2), and the square grid is equal to its own dual. Although 2D digital pictures sampled on the square grid are generally assumed, triangular and hexagonal grids also attract remarkable interest [21, 43, 75, 81, 133, 137, 184, 190, 200, 287, 296, 346, 357].

In the 3D case, our attention has been focused only on the *cubic grid* denoted by $\mathbb{Z}^3$ — i.e., the grid points are the points (x, y, z) with integer coordinates, hence each associated Voronoi cell is a cube in 3D Euclidean space, whose edges are of length 1 and parallel to the coordinate axes, and

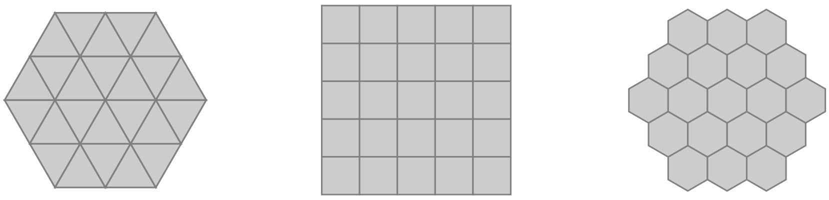


Figure 1.1: The three regular planar Voronoi diagrams associated with the three possible regular planar grids.

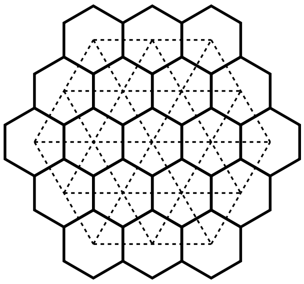


Figure 1.2: Duality between the triangular and the hexagonal grids.

whose center is a grid point (see Figure 1.3).

There are two additional remarkable 3D grids: The grid points in the *face-centered cubic grid* are the points with coordinates (x, y, z), where x, y, and z are integers such that $x + y + z$ is even, therefore the Voronoi cell associated with each grid point is a rhombic dodecahedron [137, 138] (see Figure 1.3). Many authors have suggested the use of the *body-centered cubic grid* [137, 138] whose grid points are the points with (x, y, z) in which x, y, and z are integers and $x \equiv y \equiv z \pmod 2$, in consequence the Voronoi cell associated with each grid point is a truncated octahedron (see Figure 1.3).

Note that some further interesting grids (e.g., Khalimsky grids) are also discussed in the seminal work of Kong and Rosenfeld [137].

1.1.2 Adjacency Relations

Let us focus on the grid $\mathcal{V}$ ($\mathcal{V} \in \{\, \mathcal{T}, \mathbb{Z}^2, \mathcal{H}\}$). Two points in $\mathcal{V}$ are 1-*adjacent* if the associated cells (i.e., regular polygons) share an edge, and they are 2-*adjacent* if those cells share an edge or a vertex. Note that both binary relations are reflexive and symmetric. Let us denote the set of points being j-adjacent to a point $p \in \mathcal{V}$ by $N_j^{\mathcal{V}}(p)$, and let $N_j^{*\mathcal{V}}(p) = N_j^{\mathcal{V}}(p) \setminus \{p\}$ $(j = 1, 2)$ (see Figure 1.4). It is obvious that $N_1^{\mathcal{T}}(p) \subset N_2^{\mathcal{T}}(p)$, $N_1^{\mathbb{Z}^2}(p) \subset N_2^{\mathbb{Z}^2}(p)$, and $N_1^{\mathcal{H}}(p) = N_2^{\mathcal{H}}(p)$.

In the triangular grid, $N_1^{*\mathcal{T}}(p)$ and $N_2^{*\mathcal{T}}(p)$ contain 3 and 12 elements, re-

Figure 1.3: A 3D object sampled on the cubic grid (left), the face-centered cubic grid (middle), and the body-centered cubic grid (right).

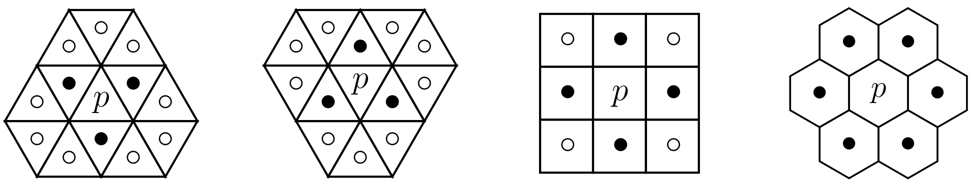


Figure 1.4: Adjacency relations on the three possible regular planar grids. Points that are 1-adjacent to the central point p are marked '•', while points being 2-adjacent but not 1-adjacent to p are marked '◦'. (Note that relations on the triangular grid for 'Δ' and '∇' points are differentiated.)

spectively. Hence, 1-adjacency and 2-adjacency are also called, respectively, 3-adjacency and 12-adjacency [133, 184]. Note that the triangular grid is formed by two kinds of points: Δ-*triangles* and ∇-*triangles* (see Figure 1.4). Without loss of generality, throughout this dissertation the topological properties of the triangular grid are illustrated for 'Δ' points.

In the conventional square grid, $N_1^{*\mathbb{Z}^2}(p)$ contains 4 elements, and $N_2^{*\mathbb{Z}^2}(p)$ is formed by 8 elements. That is why they are often referred to as 4-adjacency and 8-adjacency, respectively [133, 137, 184]. In the case of hexagonal grid, $N_1^{*\mathcal{H}}(p) = N_2^{*\mathcal{H}}(p)$ contains 6 elements, hence these two relations are also called 6-adjacency [184].

In the 3D cubic grid $\mathbb{Z}^3$ two points are 6-adjacent, 18-adjacent, and 26-adjacent if the associated Voronoi cells (i.e., unit cubes) share a face, share a face or an edge, and share a face, an edge, or a vertex, respectively.

Note that all the three binary relations are reflexive and symmetric. Let us denote the set of points that are j-adjacent to a point $p \in \mathbb{Z}^3$ by $N_j^{\mathbb{Z}^3}(p)$, and let $N_j^{*\mathbb{Z}^3}(p) = N_j^{\mathbb{Z}^3}(p) \setminus \{p\}$ $(j = 6, 18, 26)$ (see Figure 1.5).

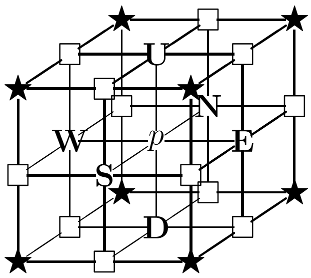


Figure 1.5: Frequently used adjacency relations in $\mathbb{Z}^3$. The set $N_6^{\mathbb{Z}^3}(p)$ contains point p and the six points marked **U**, **D**, **N**, **E**, **S**, and **W**. The set $N_{18}^{\mathbb{Z}^3}(p)$ contains $N_6^{\mathbb{Z}^3}(p)$ and the twelve points marked '□'. The set $N_{26}^{\mathbb{Z}^3}(p)$ contains $N_{18}^{\mathbb{Z}^3}(p)$ and the eight points marked '★'.

1.1.3 Binary Digital Pictures

Let us consider the adjacency relation j ($j = 1, 2$) on a regular planar grid.

A point $p \in \mathcal{V}$ is *j-adjacent* to a set of points $X \subseteq \mathcal{V}$ if there is a point $q \in X$ such that $q \in N_j^{\mathcal{V}}(p)$ [137].

A sequence of distinct points $\langle p_0, p_1, \ldots, p_m \rangle$ in $\mathcal{V}$ is called a *j-path* from p_0 to p_m in a non-empty set of points $X \subseteq \mathcal{V}$ if each point of the sequence is in X and p_i is j-adjacent to p_{i-1} for each $i = 1, 2, \ldots, m$. Note that a single point is a j-path of length 0.

Two points are said to be *j-connected* in a set X if there is a j-path in X between them. A set of points X is *j-connected* in the set of points $Y \supseteq X$ if any two points in X are j-connected in Y. A *j-component* of a set of points X is a maximal (with respect to inclusion) j-connected subset of X. Figure 1.6 illustrates the components of a set $X \subset \mathbb{Z}^2$.

In their seminal work [137], Kong and Rosenfeld defined a *binary digital picture* as follows:

Definition 1.1.1 [137] *A* $(k, \bar{k})$ binary digital picture *(or, in short* picture*) on grid $\mathcal{V}$ is a quadruple $(\mathcal{V}, k, \bar{k}, B)$, where $B \subseteq \mathcal{V}$ is the set of black points (consequently, $\mathcal{V} \setminus B$ is the set of white points), k-adjacency and $\bar{k}$-adjacency are used for B and $\mathcal{V} \setminus B$ ($k, \bar{k} \in 1, 2$), respectively.*

For practical purposes, we assume that all pictures are *finite* (i.e. they contain finitely many black points).

A *black component* or *object* is a k-component of B, while a *white component* is a $\bar{k}$-component of $\mathcal{V} \setminus B$. In a finite picture there is a unique infinite

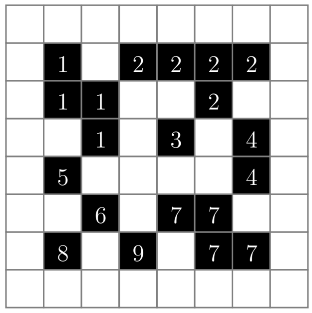


Figure 1.6: The set of points $X \subset \mathbb{Z}^2$ depicted in black. There are nine 1-components of X, where all points labeled by 'i' belong to the i-th component ($i = 1, \ldots, 9$). Note that X forms just one 2-component.

white component, which is said to be the *background*, and a finite white component is called a *cavity*[1].

A black point p is called an *interior point* if all points in $N_{\bar{k}}^{*\mathcal{V}}(p)$ are black. A black point is said to be a *border point* if it is not an interior point (i.e., it is $\bar{k}$-adjacent to at least one white point). A (black) border-point p is called an *isolated point* if all points in $N_{\bar{k}}^{*\mathcal{V}}(p)$ are white (i.e., $\{p\}$ is a singleton object).

In order to avoid *connectivity paradoxes* [137], and verify the *discrete Jordan's theorem* [184], different adjacency relations are usually taken into consideration for the black and white points (i.e., $k \neq \bar{k}$).

All concepts reviewed in the 2D case can be extended to the cubic grid $\mathbb{Z}^3$. In this work, our attention has been focussed on $(26, 6)$ pictures.

A black point p in picture $(\mathbb{Z}^3, 26, 6, B)$ is called an *interior point* if $N_6^{*\mathbb{Z}^3}(p) \subset B$ (i.e., all 6-neighbors of p are black). A black point p in this picture is said to be a *border point* if it is not an interior point (i.e., it is 6-adjacent to at least one white point). A black point p in picture $(\mathbb{Z}^3, 26, 6, B)$ is called a *d-border point* if its 6-neighbor marked d in Figure 1.5 is a white point ($d \in \{$ **U**, **D**, **N**, **E**, **S**, **W** $\}$).

[1]Some researchers use the term *hole* to refer to finite white components in 2D pictures, and the 'tunnel' that a 3D doughnut (torus) has is said to be a hole, as well. To avoid possible confusion, we reserve the term *cavity* to refer to finite white components in arbitrary dimensions.

1.2 Topology-Preserving Operators

Topology preservation is a major concern in topological algorithms. It is crucial to guarantee that an operator preserves the topology for all possible pictures.

This section defines the concepts of an operator and a simple point, and reviews some sufficient conditions for topology preservation.

1.2.1 Operators

Recall that only two possible values are assigned to each grid point in a (binary digital) picture. It is assumed that these two values are black (1) and white (0).

Let T be an *operator* that transforms the input picture with the set of black points B into the output picture in which the set of black points is denoted by $\mathsf{T}(B)$. A *local operator* gives each point p a 'new' value that depends only on the 'old' values of the *local neighborhood* or *support* of p. If the support contains n points, local operator T is specified by its *alteration rule* $f_{\mathsf{T}}: \{0,1\}^n \to \{0,1\}$ that is a Boolean function of n variables.

Operator T is called a *reduction* if $\mathsf{T}(B) \subseteq B$ for each possible set of black points B. Hence, reductions transform pictures only by changing some black points to white ones which is referred to as *deletion*. Operator T is said to be an *addition* if $B \subseteq \mathsf{T}(B)$ for each B. Thus additions change only some white points which is referred to as *filling*. An operator which does not fall into either of these two categories is called a *mixed operator*.

Note that reductions play a key role in various topological algorithms, e.g., *thinning* [89, 147, 324] (i.e., iterative object reduction to extract skeleton-like features) or *reductive shrinking* [90] (i.e., that is capable of producing a minimal structure that is topologically equivalent to the original object).

*Parallel operator*s can change a set of points simultaneously, while *sequential operator*s traverse the points of a picture, and focus on the actually visited single point for possible alteration [90]. Evidently, the result of a sequential operator may depend on the visiting order which is applied.

1.2.2 Topology Preservation in Reductions

In [319], Stefanelli and Rosenfeld laid down the following criterion for topology-preserving 2D reductions:

Criterion 1.2.1 [319] *A reduction (acting on 2D pictures) preserves the topology if and only if the following conditions all hold:*

1. *It never splits an object into two or more.*

2. *It never deletes completely any object in the input picture.*

3. *It never merges a cavity in the input picture with the background or another cavity.*

4. *It never creates a cavity where there was none in the input picture.*

Figure 1.7 illustrates a 2D reduction that is not topology-preserving.

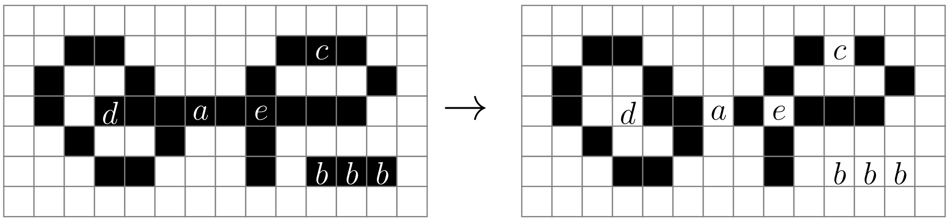


Figure 1.7: A reduction for a (2, 1) picture on $\mathbb{Z}^2$ that is not topology-preserving. Deletion of the point marked 'a' splits the larger object into two; the smaller object is completely deleted by deleting the points marked 'b'; deletion of the point marked 'c' merges a cavity with the background; the remaining two cavities are merged with each other by deleting the point marked 'd'; deletion of the point marked 'e' creates a brand new cavity.

In effect, Criterion 1.2.1 was rephrased as follows:

Criterion 1.2.2 [137] *Let $(\mathcal{V}, k, \bar{k}, B)$ be a (2D) picture. The deletion of the set of points $D \subset B$ preserves the topology if and only if*

1. *each k-component of B contains exactly one k-component of $B \setminus D$, and*

2. *each $\bar{k}$-component of $\mathcal{V} \setminus (B \setminus D)$ contains exactly one $\bar{k}$-component of $\mathcal{V} \setminus B$.*

Note that Criteria 1.2.1 and 1.2.2 are only valid in 2D.

There is an additional concept called a *hole* (or a *tunnel*) in 3D pictures [137, 139]. Holes (of the kind that doughnuts have) are formed by white points, but they are not white components. Topology preservation implies that creating, eliminating, and merging holes are not allowed (see Figure 1.8). That is why the proposed criteria for topology-preserving 3D reductions are rather complicated [137]. They are based on *digital deformation of closed paths* and a *digital fundamental group* introduced by Morgenthaler [193] and Kong [136], respectively.

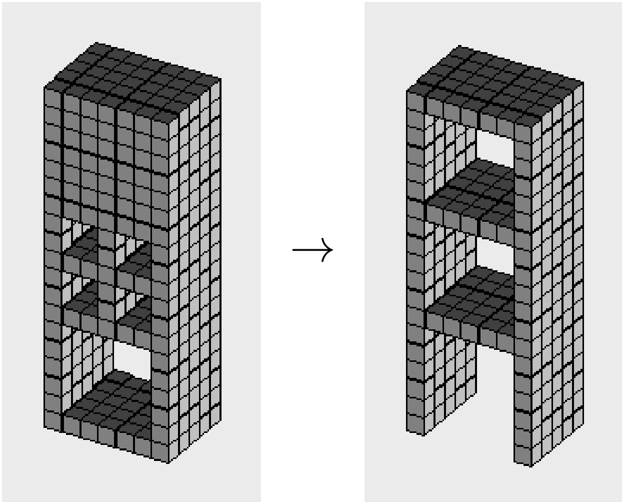


Figure 1.8: A 3D reduction that satisfies all conditions of Criteria 1.2.1, but is not topology-preserving on $(26, 6)$ pictures, since a hole not present in the input picture is created, a hole is eliminated, and four holes are merged with each other.

1.2.3 Simple Points

General conditions for topology preservation were obtained with the help of the concept of a *simple point* [137]. Since we have dealt with all the three kinds of operators (i.e., reductions, additions, and mixed operators), we have respect for the following definition[2]:

Definition 1.2.1 *A black point in an arbitrary picture is a* simple *point if and only if its deletion preserves the topology (i.e., it is a topology-preserving reduction). Likewise, a white point in an arbitrary picture is a* simple *point if and only if its filling preserves the topology (i.e., its filling is a topology-preserving addition).*

Kardos and Palágyi gave unified formal characterizations of simple (black or white) points in the given five types of 2D pictures:

Theorem 1.2.1 [122, 126] *Let p be a point in a picture $(\mathcal{V}, k, \bar{k}, B)$. Then p is simple if and only if the following conditions hold:*

1. *p is k-adjacent to exactly one k-component of $N_2^{*\mathcal{V}}(p) \cap B$.*
2. *p is $\bar{k}$-adjacent to exactly one $\bar{k}$-component of $N_2^{\mathcal{V}}(p) \setminus B$.*

[2]Since the attention of most researchers has been focused only on reductions, it is usually assumed that a simple point may only be black.

Theorem 1.2.1 shows that simplicity of a point p is a local property: it can be decided by examining the set $N_2^{*\mathcal{V}}(p)$ containing just 12, 8, and 6 points for $\mathcal{T}$, $\mathbb{Z}^2$, and $\mathcal{H}$, respectively. As a straightforward consequence of the above theorem we note that if a black point is an isolated or interior point then it is not simple (i.e., some border points may be simple). Another immediate consequence of Theorem 1.2.1 is the following duality theorem:

Theorem 1.2.2 *A point p is simple in picture $(\mathcal{V}, k, \bar{k}, B)$ if and only if p is simple in picture $(\mathcal{V}, \bar{k}, k, \mathcal{V} \setminus B)$ (i.e., in the picture that is obtained from the former by swapping the black and white sets of points and their associated adjacency relations).*

Figure 1.9 gives some examples of simple and non-simple points in $(1, 2)$ and $(2, 1)$ pictures on $\mathbb{Z}^2$.

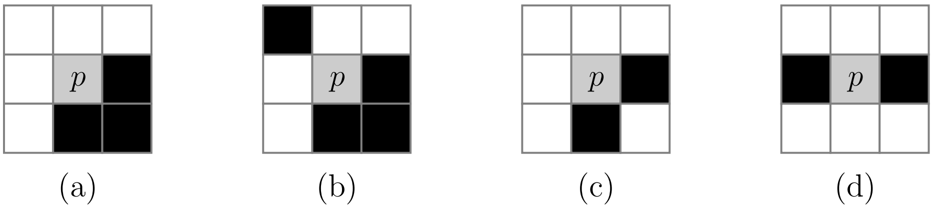


Figure 1.9: Examples of simple and non-simple points in $(1, 2)$ and $(2, 1)$ pictures on $\mathbb{Z}^2$. Point p is simple in both $(1, 2)$ and $(2, 1)$ pictures (a); p is simple in $(1, 2)$ pictures and it is non-simple in $(2, 1)$ pictures (b); p is non-simple in $(1, 2)$ pictures and it is simple in $(2, 1)$ pictures (c); p is non-simple in both $(1, 2)$ pictures and $(2, 1)$ pictures (d). Note that p may be either a black or a white point.

In [139], Kong proposed an easily visualized characterization of simple points on $(2, 1)$ pictures on $\mathbb{Z}^2$ by using the concept of an *attachment set*. Kardos and Palágyi adapted Kong's model for all the three regular planar grids [120, 124].

We should add that Kardos and Palágyi illustrated simple points in all the given five types of 2D pictures by a few configurations (so-called *matching templates*), which make an efficient implementation of the verification of simplicity possible [122, 124, 126].

Malandain and Bertrand established the following characterization of simple points on the 3D $(26, 6)$ pictures on $\mathbb{Z}^3$:

Theorem 1.2.3 [179] *Let p be a point in a picture $(\mathbb{Z}^3, 26, 6, B)$. Then p is simple in that picture if and only if all the following conditions hold:*

1. $N_{26}^{*\mathbb{Z}^3}(p) \cap B$ *contains exactly one* 26*-component.*

2. $N_{6}^{*\mathbb{Z}^3}(p) \cap (\mathbb{Z}^3 \setminus B) \neq \emptyset$ *(i.e., p is* 6*-adjacent to at least one white point).*

3. *Any two points in* $N_{6}^{*\mathbb{Z}^3}(p) \cap (\mathbb{Z}^3 \setminus B)$ *is* 6*-connected in the set* $N_{18}^{*\mathbb{Z}^3}(p) \cap (\mathbb{Z}^3 \setminus B)$.

Similarly to the examined 2D cases (see Theorem 1.2.1), the simplicity of a point p in $(26, 6)$ pictures is a local property: it can be decided by examining the set $N_{26}^{*\mathbb{Z}^3}(p)$. Figure 1.10 gives some examples of simple and non-simple points in $(26, 6)$ pictures on $\mathbb{Z}^3$.

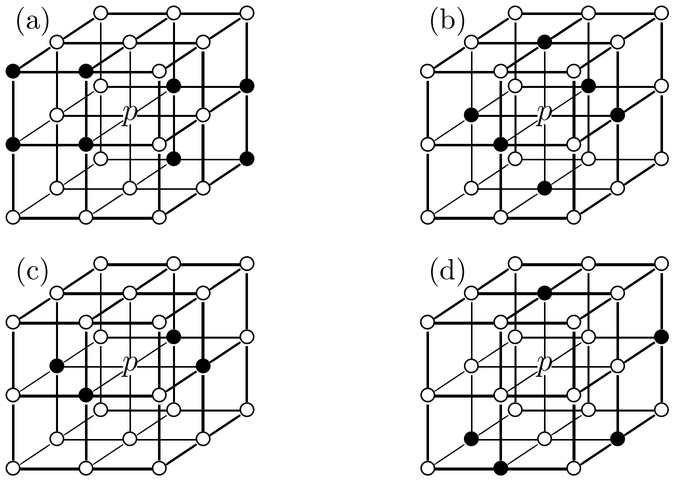


Figure 1.10: Examples of simple and non-simple points in $(26, 6)$ pictures on $\mathbb{Z}^3$. In configuration (a), p is not simple (since Condition 1 of Theorem 1.2.3 is violated); in configuration (b), p is not simple (since Condition 2 of Theorem 1.2.3 is violated); in configuration (c), p is not simple (since Condition 3 of Theorem 1.2.3 is violated); in configuration (d), p is simple since all the three conditions of Theorem 1.2.3 are satisfied.

Note that Kong extended his characterization of simple points in $(2, 1)$ pictures on $\mathbb{Z}^2$ by using attachment sets to 3D $(26, 6)$ pictures [139].

1.2.4 Sufficient Conditions

Let us see first three general sufficient conditions for topology preservation (i.e., conditions that are valid for arbitrary binary pictures).

Altering a single point p in a picture preserves the topology if and only if p is simple in that picture (see Subsection 1.2.3). Parallel operators can change a set of points at a time. Hence, we need a precise definition of what is meant by topology preservation when a number of points are changed (deleted or filled) simultaneously.

Definition 1.2.2 [139, 168] *Let* $\mathbf{P}$ *be an arbitrary picture. A set of* n *points* Q *is a* simple set[3] *in* $\mathbf{P}$ *if it is possible to arrange the elements of* Q *in a sequence* $\langle q_1, \ldots, q_n \rangle$ *such that* q_1 *is a simple point in* $\mathbf{P}$ *and each* q_i *is simple after the set of points* $\{q_1, \ldots, q_{i-1}\}$ *is changed* $(i = 2, \ldots, n)$. *Such a sequence is called a* simple sequence. *(And let the empty set be called simple.)*

Figure 1.11 gives examples of simple and non-simple sets of black points in a $(2, 1)$ picture on the grid $\mathbb{Z}^2$.

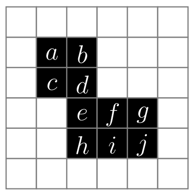


Figure 1.11: Examples of simple and non-simple sets in picture $(\mathbb{Z}^2, 2, 1, \{a, \ldots, j\})$. The set of black points $\{a, b, c, d\}$ is simple since all the 12 sequences (of the possible 24 ones) $\langle a, b, c, d\rangle$, $\langle a, b, d, c\rangle$, $\langle a, c, b, d\rangle$, $\langle a, d, b, c\rangle$, $\langle b, a, c, d\rangle$, $\langle b, a, d, c\rangle$, $\langle b, c, a, d\rangle$, $\langle b, d, a, c\rangle$, $\langle c, a, b, d\rangle$, $\langle c, b, a, d\rangle$, $\langle d, a, b, c\rangle$, and $\langle d, b, a, c\rangle$ are simple. The set of black points $\{f, i\}$ is non-simple, since both sequences $\langle f, i\rangle$ and $\langle i, f\rangle$ are non-simple. (Note that each black point is simple. Hence, all the 10 singleton sets $\{a\}, \ldots, \{j\}$ are simple sets.)

There is general agreement that the concept of a simple set trivially implies a sufficient condition for topology-preserving parallel operators:

Criterion 1.2.3 [139, 168, 272] *A parallel operator is* topology-preserving *if, for all possible pictures, it changes only simple sets.*

Bertrand [23] and Kong [139] reported alternative solutions to the problem: they introduced the notions of a *P-simple set* and a *hereditarily simple set*, respectively, whose simultaneous deletion is proved to be topology-preserving.

[3] Since most researchers have studied only reductions, it is usually assumed that a simple set is a subset of black points. We have dealt with all the three kinds of operators (i.e., reductions, additions, and mixed operators), hence there may be white points in 'our' simple sets.

Definition 1.2.3 [23] *Let* $\mathbf{P} = (\mathcal{V}, k, \bar{k}, B)$ *be an arbitrary picture. A set of black points* $Q \subset B$ *is a* P-simple set *in* $\mathbf{P}$ *if for any point* $q \in Q$ *and any set of points* $R \subseteq Q \setminus \{q\}$*,* q *is simple in picture* $(\mathcal{V}, k, \bar{k}, B \setminus R)$*. Each element of a P-simple set is called a* P-simple point.

Theorem 1.2.4 [23] *A reduction that deletes a subset composed solely of P-simple points is topology-preserving.*

Note that Bertrand gave a local characterization of P-simple points in $(26, 6)$ pictures on $\mathbb{Z}^3$ [23]. Later Bertrand and Couprie proposed a similar characterization in $(2, 1)$ pictures on $\mathbb{Z}^2$ [27], and Kardos and Palágyi presented both formal and easily visualized sufficient and necessary conditions of P-simple points in all the given five types of 2D pictures [125].

Figure 1.12 shows a P-simple set in a $(2, 1)$ picture on $\mathbb{Z}^2$.

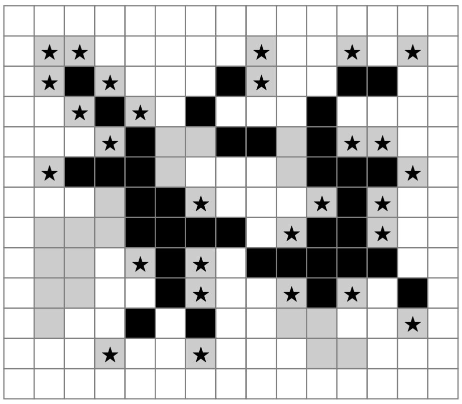


Figure 1.12: Example for a P-simple set in a $(2, 1)$ picture on $\mathbb{Z}^2$. Non-simple black points are depicted in black, simple black points are depicted in gray, elements in a P-simple set are marked '$\star$'. Note that all possible P-simple sets are subsets of simple points.

Definition 1.2.4 [139] *A set of black points* Q *is said to be* hereditarily simple *in a picture if all subsets of* Q *(including* Q *itself) are simple sets in that picture.*

Theorem 1.2.5 [139] *A reduction that deletes only hereditarily simple sets is topology-preserving.*

To ensure topology preservation for parallel reductions, Ronse gave the following specific sufficient condition:

Theorem 1.2.6 [272] *A parallel reduction* R *acting on* $(2,1)$ *pictures on* $\mathbb{Z}^2$ *is topology-preserving, if all the following conditions hold:*

1. *Only simple points are deleted by* R.

2. *For any two* 1*-adjacent black points* p *and* q *that are deleted by* R, $\{p,q\}$ *is a simple set.*

3. R *never deletes completely any object contained in a* 2×2 *square (i.e., a set of four mutually* 2*-adjacent points).*

Ma [168] and Kong [139] gave the foremost sufficient condition for topology-preserving reductions on 3D $(26,6)$ pictures. They introduced the concepts of a unit lattice square and a unit lattice cube: a *unit lattice square* in $\mathbb{Z}^3$ is formed by four mutually 18-adjacent points, and a *unit lattice cube* is a set of eight mutually 26-adjacent points (see Figure 1.13).

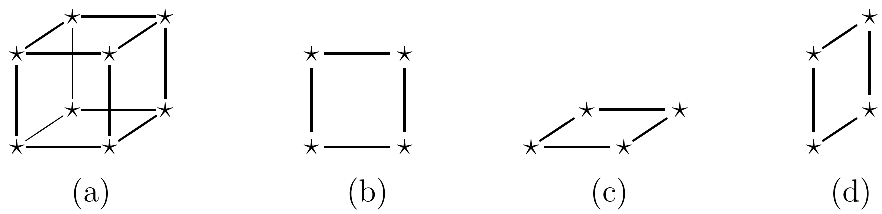


Figure 1.13: A unit lattice cube (a), and three unit lattice squares (b)-(d). Their corners (i.e., points in $\mathbb{Z}^3$) are marked '$\star$'. (Notice that each 'face' of a unit lattice cube is a unit lattice square.)

Theorem 1.2.7 [139, 168] *A 3D parallel reduction* R *acting on* $(26,6)$ *pictures is topology-preserving if all the following conditions hold:*

1. *Only simple points are deleted by* R.

2. *For any two black points* p *and* q *contained in a unit lattice square that are deleted by* R, $\{p,q\}$ *is a simple set.*

3. *For any three black points* p, q, *and* r *contained in a unit lattice square that are deleted by* R, $\{p,q,r\}$ *is a simple set.*

4. *For any four black points* p, q, r, *and* s *contained in a unit lattice square that are deleted by* R, $\{p,q,r,s\}$ *is a simple set.*

5. R *never deletes completely any object contained in a unit lattice cube.*

In order to verify the topological correctness of some 3D thinning algorithms, Palágyi and Kuba [221] proposed a simplified condition[4]:

Theorem 1.2.8 [221] *A 3D parallel reduction* R *acting on* $(26, 6)$ *pictures on* $\mathbb{Z}^3$ *is topology-preserving if all the following conditions hold:*

1. *Only simple points are deleted by* R.
2. *Let* $p \in B$ *be any point in a picture* $(\mathbb{Z}^3, 26, 6, B)$ *such that* p *is deleted by* R. *Let* $Q \subset B \setminus \{p\}$ *be any set of points that are deleted by* R *from picture* $(\mathbb{Z}^3, 26, 6, B)$ *such that* $Q \cup \{p\}$ *is contained in a unit lattice square. Then* p *is simple in picture* $(\mathbb{Z}^3, 26, 6, B \setminus Q)$.
3. R *never deletes completely any object contained in a unit lattice cube.*

[4] This result originated before defending the author's Ph.D. dissertation in the year 2000.

1.3 Skeletonization

In his seminal work [32], Blum introduced a shape feature called *skeleton* that jointly describes the topology and the geometry of a binary object. The skeleton of a continuous object in $\mathbb{R}^n$ was historically given by the *Medial Axis Transform* (also-called *Symmetry Axis Transform*), which tracks the points having at least two closest boundary points. It can be defined in two alternative ways:

- Blum's grassfire (or prairie-fire) propagation assumes that the object is a field of dry grass and its entire boundary is set on fire at a time. It is also supposed that the fire spreads in all directions with equal velocity. Then the skeleton is the set of points where the concurrent fire fronts meet and quench each other (see Figure 1.14).
- The skeleton is the loci of the centers of all maximal inscribed hyperspheres (i.e., disks in 2D, balls in 3D, etc.), where a hypersphere is inscribed if it is completely included by the object, and it is maximal if it is not covered by any other inscribed hypersphere (see Figure 1.15).

The equivalence of the above two approaches was proved by Calabi and Hartnett [46, 92].

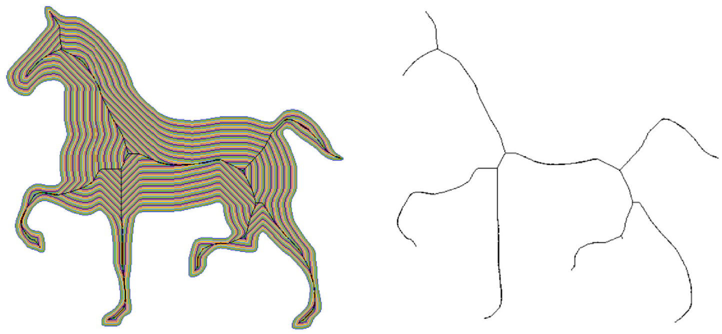


Figure 1.14: Illustrating the prairie-fire propagation for a horse. Colored layers removed by the isotropic object reduction process (left) and the skeleton (right).

The skeleton is a widely used shape descriptor due to its advantageous properties:

- It represents local object symmetries [39, 59, 79, 290].
- It reflects the topological structure of the object to be characterized [45, 188].

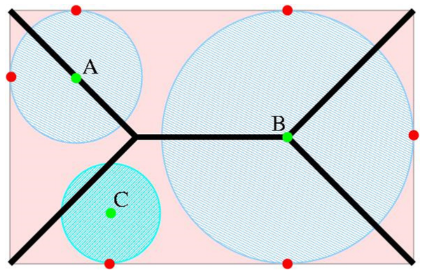


Figure 1.15: Skeleton of a solid (2D) rectangle. Both points 'A' and 'B' belong to the skeleton, since they have more than one closest boundary points and they are centers of maximal inscribed disks. On the contrary, point 'C' is not skeletal, since the corresponding disk is not maximal (i.e., it touches the boundary in just one point).

- It can be used to object decomposition (i.e., to partition an object into a set of primitives) [33, 34, 199].
- It reduces the dimensionality since a 2D object is reduced to 1D curves; the skeleton of a 3D object may contain just 2D and 1D structures (surfaces and curves) [59, 79, 199, 280] (see Figure 1.16).
- It is 'thin' (i.e., the skeleton contains 1-point width manifolds and offers dramatically less information than the original object) [59, 79].

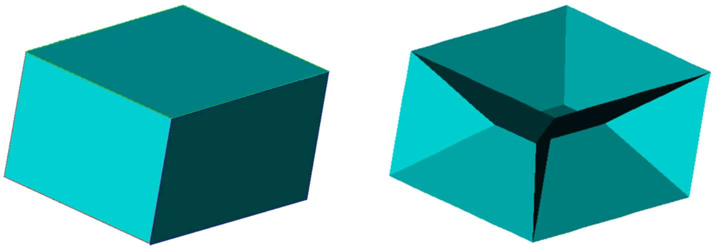


Figure 1.16: A solid 3D box (left) and its skeleton (right) that contains some 2D surface patches.

The above-defined skeleton assumes *continuous* approach. Since we deal with digital pictures, *skeletonization* means extraction of *skeleton-like features* from digital binary objects [42, 54, 280]. These features are presented in Subsection 1.3.1.

Subsection 1.3.2 describes the major skeletonization approaches, and applications are reviewed in Subsection 1.3.3.

1.3.1 Skeleton-Like Features in 2D and 3D

In 2D, two kinds of skeleton-like features are taken into consideration (see Figure 1.17): the *centerline* that approximates the continuous skeleton, and the *topological kernel* that is a minimal set of points being topologically equivalent [137] to the original object (i.e., if we remove any further point from it, then the topology is not preserved). If an object does not contain any cavity, its topological kernel is an isolated point. Otherwise, topological kernels are formed by 1-point thick closed curves.

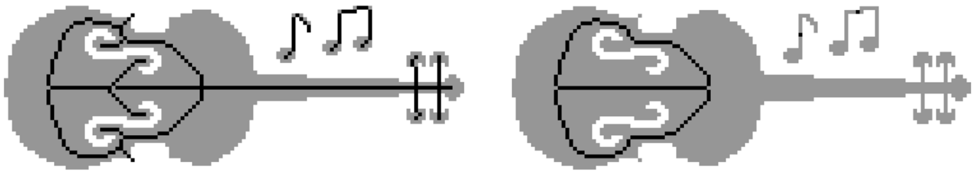


Figure 1.17: Centerline (left) and topological kernel (right) superimposed on a 2D (discrete) object.

In the 3D case, there are three types of skeleton-like features: the *centerline*, the *medial surface* (see Figure 1.18), and the *topological kernel* (see Figure 1.19).

The centerline in 3D is a line-like or stick-like 1D representation of objects that captures the part-whole structure of the object to be described [53, 307]. In many applications (see Subsection 1.3.3), it is a concise representation of tubular and tree-like 3D objects. The medial surface provides an approximation to the the continuous 3D skeleton, since it can contain 2D surface patches. Similarly to the 2D topological kernel, a topological kernel of a 3D object is a minimal set of points that is topologically equivalent to the original object. It is fairly useful in representing or checking the topological structure of the object to be described. Note that a topological kernel of a 3D object is an isolated point if and only if it does not contain any holes nor

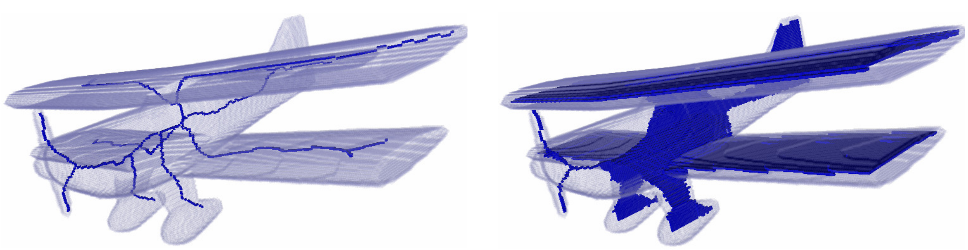


Figure 1.18: Centerline (left) and medial surface (right) superimposed on a 3D image of a biplane.

cavities.

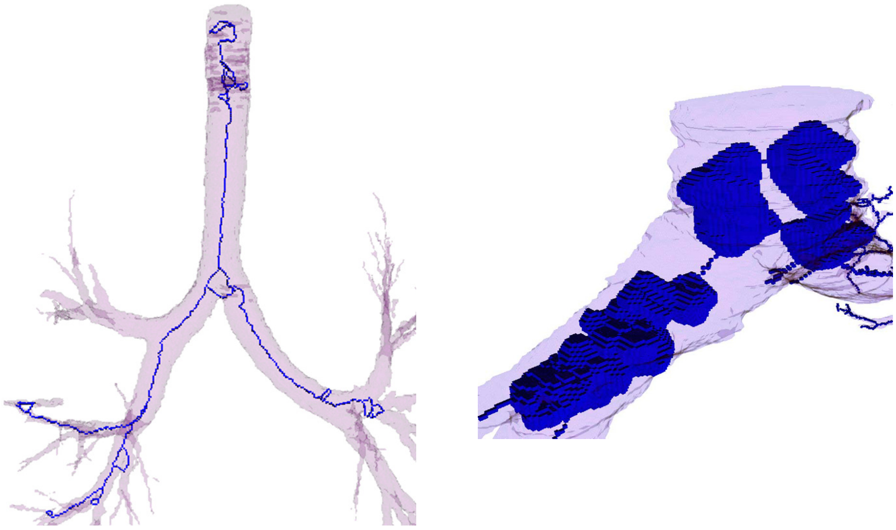


Figure 1.19: Topological kernels superimposed on incorrectly segmented 3D human airway trees. Since the segmented trees contain some holes (left) and some cavities (right), there are some closed 'thin' curves (loops) and some 'bubbles' (cavities) with one point thick walls in their topological kernels.

It is an important property of (2D and 3D) centerlines that they contain just three types of elements: *endpoints* (which have only one skeletal neighbor), *line-points* (that are 'adjacent' to exactly two skeletal points), and *branch-points* (that have more than two skeletal neighbors) that form junctions (bifurcations, trifurcations, etc.). Hence, a complex descriptor named *skeletal graph*[5] can be derived from centerlines. The set of vertices of a skeletal graph is formed by the endpoints and the branch-points, and there is an edge between two vertices if they are connected via 'adjacent' line-points [15, 16, 41, 62, 109, 167, 177, 231, 269, 273] (see Figures 1.20 and 1.21).

Lastly, we mention that Pizer et al. [256, 258] proposed a particular type of shape feature called *deformable m-reps* (a sampled medial object representation). It is derived from the medial surface of a 3D object, and represents 3D object boundaries using sheets of medial atoms [304].

1.3.2 Skeletonization Techniques

Several approaches have been proposed for producing skeleton-like features from (segmented) binary objects. Some authors presented comprehensive

[5]Some researchers use the term *skeleton graph* for the same shape descriptor. That is ambiguous, since the notion of skeleton graph also means a particular edge- and vertex-weighted simple graph introduced by Erdős et al. [66].

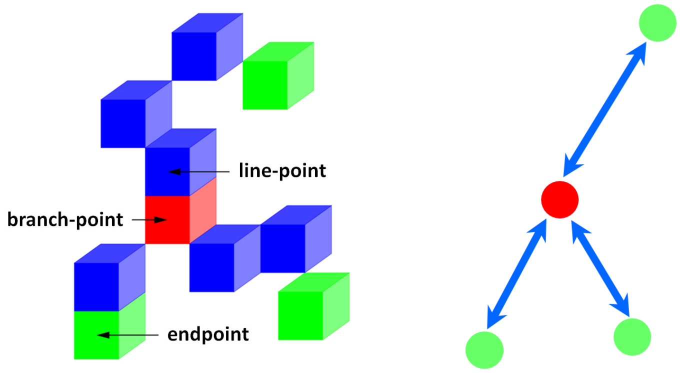


Figure 1.20: A 3D centerline with the three types of elements (left), the derived skeletal graph (right).

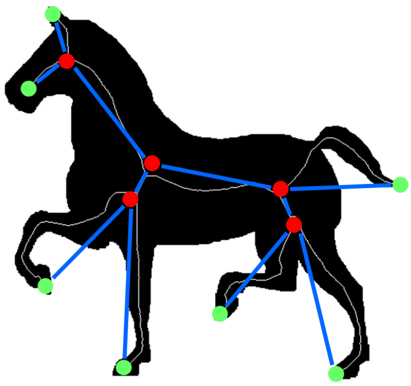


Figure 1.21: Centerline superimposed on a 2D image of a horse and the corresponding skeletal graph.

and concise surveys [278, 279, 280, 282, 304, 307, 330]. Here we summarize the three major skeletonization techniques: generation from distance maps, geometric approach based on Voronoi diagrams, and modeling the fire-front propagation (thinning).

Distance-based skeletonization refers to the definition of the (continuous) skeleton by the centers of all maximal inscribed hyperspheres. It uses *distance maps* that are the results of *distance transform* algorithms [35]. The input of a (discrete) distance transform algorithm is a binary picture in which white points form the set of feature points, and it is converted into a (non binary) array, where each element has a value that gives the distance to the 'nearest' feature point. The produced distance map strongly depends on the applied distance [86]. The most popular ones are distance family derived from adjacency relations [35, 321], weighted or chamfer distances [35, 322], quasi-Euclidean distances [35, 60, 191], and the exact (error-free) Euclidean distance [67, 189, 284, 353]. Note that computing distance transform takes linear time in arbitrary dimensions [35, 189].

If a distance transform is calculated from the white points in a binary picture, some *local maxima* in the distance map form centerlines in 2D, and medial surfaces in the 3D case. These 'ridges' have been identified by standard techniques of differential geometry and heuristic methods [5, 36, 270].

Note that distance-based skeletonization can produce geometrically accurate features (from exact Euclidean distance maps), but some ridge-detectors cannot provide topologically correct results.

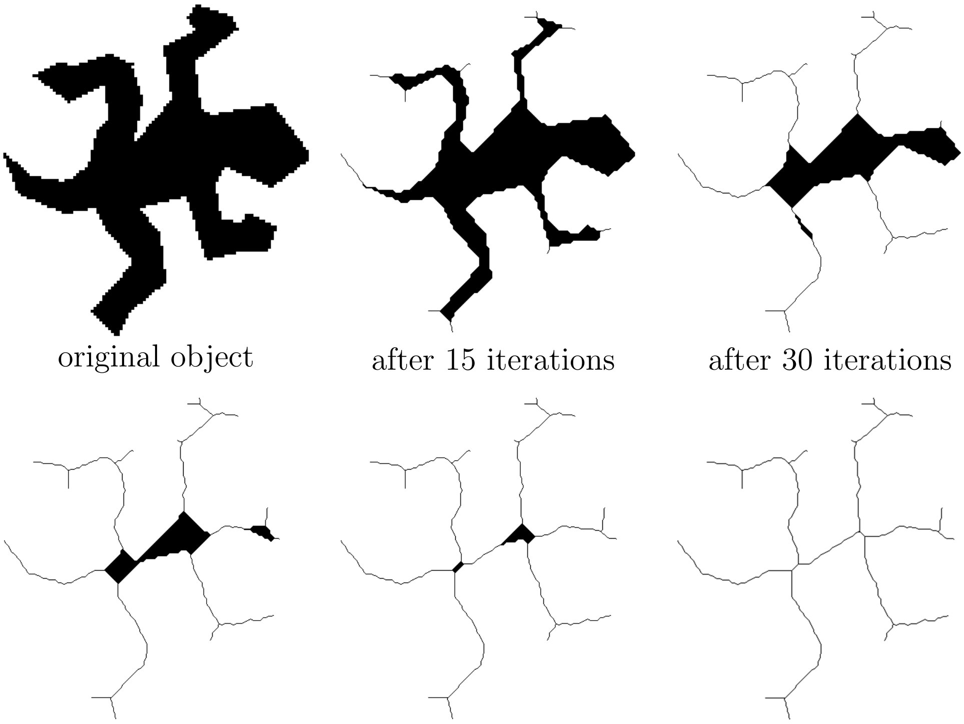
The *Voronoi skeleton* is obtained by computing the Voronoi diagram of sampled boundary points [3, 280, 329]. In [289], Schmitt has shown that if the density of the sampled boundary points uniformly goes to infinity, the 'internal' elements of the corresponding Voronoi diagram converges to the (exact continuous) skeleton. Several authors proposed computationally efficient algorithms for computing the Voronoi diagram [40, 131, 202, 216, 262]. The raw Voronoi skeleton may contain a large number of spurious branches in 2D, and unwanted surface patches in the 3D case. Thus the Voronoi skeletonization is to be paired with a proper *pruning* method [12, 202, 217, 218].

Although the Voronoi skeleton is correct in both geometrical and topological senses, this method is rather time-consuming and 'hybrid' (i.e., its input is a discrete set of points, but the output if formed by continuous segments).

The third major skeletonization technique is the digital simulation of the fire-front propagation. *Thinning* [89, 147, 280, 307, 324] is an iterative object-reduction process for producing any skeleton-like feature in a topology-preserving way: the outmost layer of an object is deleted, and the entire process is repeated until stability is reached (see Figure 1.22).

Most of the existing thinning algorithms are parallel, since the fire-front propagation is parallel by nature [8, 22, 25, 27, 28, 29, 56, 57, 58, 82, 88, 89, 113, 115, 120, 121, 147, 150, 162, 165, 166, 168, 169, 170, 171, 172, 174, 175, 180, 181, 203, 204, 205, 206, 208, 209, 210, 211, 219, 220, 221, 222, 225, 229, 232, 233, 234, 235, 236, 237, 242, 243, 244, 246, 248, 268, 275, 276, 324, 338, 354, 358]. We should add that some sequential algorithms have also been proposed [55, 112, 114, 116, 117, 147, 150, 197, 198, 224, 240, 242, 243, 244, 246, 252, 266, 320, 324].

The topology-oriented thinning pays less attention to the metric properties of the object to be represented, since the invariance under arbitrary rotation angles or scaling factors is not fulfilled. In spite of these drawbacks, our attention has been focussed on thinning, since thinning is the fastest skeletonization method, it can be implemented easily, it can produce all types of



original object — after 15 iterations — after 30 iterations

after 45 iterations — after 60 iterations — centerline

Figure 1.22: Thinning of Escher's (rep)tile. Some phases of the iterative object reduction are shown.

skeleton-like features, the topology-preservation is guaranteed, and thinning provides practically exquisite descriptors for a number of applications.

Note that there are some additional skeletonization approaches:

- Some attempts have been made to approximate the result of Medial Axis Transform in discrete spaces. The 'skeletons' are defined in terms of a new concept, called the Integer Medial Axis (IMA) transform [98, 186].

- Distance transform has been combined with thinning. Some authors proposed skeletonization algorithms in which the distance transform is followed by (non-iterative) sequential deletion of 'deletable' points in ascending distance order [263, 285, 309]. In [327], 'peak' points are detected in distance maps, then these points are preserved by an iterative reductive shrinking.

- The 'skeleton' can be expressed in terms of the basic morphological operations [69, 83, 296, 308], where hyperspheres are approximated by successive dilations, and the original object can be exactly reconstructed from the the skeletal subsets [149]. Note that some simple thinning algorithms can be described by the morphological hit-or-miss transform (as a basic tool for detecting the 'deletable' points of the applied reduction) [83, 133, 184, 296, 308].

- Some authors have used minimal cost method for generating 2D and 3D centerlines (i.e., path optimization can be carried out with graph algorithms) [107, 108, 161].

- Several techniques use continuous curve propagation for simulating the fire-front propagation: active contour [152], anisotropic partial differential equation [365], Gaussian affinity voting [371], principal curves [154], shock of boundary evolution [128, 302], normal field [104, 176], level set [129, 332, 351], flux [37, 255, 303], Markov framework (MRF) [2], absolute scale [11], potential field [1, 51], and gradient vector flow [93].

- Some authors have reported skeletonization algorithms acting on greyscale pictures and fuzzy objects [6, 7, 31, 106, 173, 183, 266, 283, 293, 297, 331, 363].

Numerous authors have proposed methods to evaluate the performance of skeletonization algorithms [9, 53, 85, 103, 145, 276, 286, 288, 306, 307, 330]. Due to the lack of definition of the 'true skeleton' for a discrete object, a widely accepted approach evaluating the goodness of skeletonization algorithms is yet absent [280]. That is why we have proposed a method for the quantitative comparison of 'skeleton' features [212]. The two key components of our approach are a specific similarity measure for 'skeletons' and a gold standard image database containing pairs of reference objects and their expected results.

1.3.3 Applications of Skeletonization

Skeleton-like features have been widely used in different image processing and computer vision applications. Here, we provide a list of these applications with no claim to completeness: animation [54, 61, 74, 347], chordal surface generation [146], computer graphics [53], coding [100, 158, 143, 182], design and engineering applications [264], fingerprint analysis [68, 84, 369], generating mesh sizing functions [265], measuring shape similarity [336], motion analysis [73, 74], multiscale shape analysis [335], object recognition and classification [11, 16, 62, 95, 291, 300, 362], off-line character recognition

[10, 65, 148], part-patch segmentation and object decomposition [135, 295], porous filter permeability, analysis of porous media, and pore space morphology [155, 156, 159, 261, 305, 368], raster-to-vector conversion [18, 253, 310], registration [345], segmentation [44, 53, 257, 259], shape deformation and morphing [30, 361], shape matching and retrieval [15, 41, 49, 53, 54, 70, 80, 96, 104, 167, 178, 213, 267, 325], shape modeling [71, 304, 347], terrain modeling [333], tracing and virtual navigation [53, 101, 104].

Skeletonization has been frequently applied in many medical image processing applications, including airway analysis and virtual bronchoscopy [19, 76, 107, 195, 196, 360, 363], brain tissue analysis [132], characterization of complex anatomic object [38, 202, 367], colonography and virtual colonoscopy [63, 111, 274, 102, 318, 348], estimating the motion of arteries [334], malignant tumor identification [134], medical image segmentation and registration [47, 254, 260, 364], mesh generation of tubular geometries [185], neuron traces [344], path planning [352], stenosis detection [50, 72, 85, 215, 288, 351], surgical planning [292], trabecular bone analysis [105, 277, 281, 283], vascular and endovascular surgery, vessel lumen segmentation, angiography, angiogenesis, and detection of aortic aneurism [4, 48, 78, 91, 151, 154, 157, 261, 194, 214, 271, 301, 337, 349, 350, 355, 356, 359, 366, 370, 371], virtual endoscopy [94, 161], and visualization of tomographic molecular imaging [17].

In [153], Leymarie and Kimia covered a wide spectrum of applications of medical symmetries (including applications in geography, cartography, wireless sensor networks, urbanism, architecture, archaeology, visual arts, motion analysis, body animation, robotics, machining, industrial design, registration, medicine, and biology).

The author's 3D thinning algorithms have been involved in the following biomedical applications: assessment of infrarenal aortic aneurysm [224], assessment of tracheal stenosis [224, 311, 313, 315, 316], unraveling and virtual dissection of the colon [224, 312, 314, 317], colorectal polyp detection [314], characterization of the interstitial lung diseases [99], matching and anatomical labeling of human airway tree [339, 341], quantitative analysis of pulmonary airway trees [226, 227, 228, 231], liver segmentation for surgical resection planning [20], and identifying synaptic connections [187].

For reasons of scope, in this dissertation, our method for quantitative analysis of pulmonary airway trees (see Chapter 4) is only described.

Chapter 2

Topology Preservation

This chapter is to summarize our theoretical results concerning diversified topological problems. These results provide methods of verifying that an operator preserves the topology, allow us to generate topology-preserving operators, and provide computationally efficient thinning algorithms.

For reasons of scope, in this chapter, only some selected results are presented.

Configuration-based and point-based sufficient conditions for topology-preserving reductions are presented for 2D pictures in Section 2.1.

In Section 2.2, both symmetric and asymmetric point-based conditions are given for topology-preserving reductions acting on 3D $(26, 6)$ pictures on $\mathbb{Z}^3$.

Instead of investigating the sets of altered points, the author proposed a novel sufficient condition for topology-preserving operators that takes the alteration rules of operators into consideration. In Section 2.3, it is described that the general-simple alteration rules provide pairs of equivalent and topology-preserving sequential and parallel operators.

In Section 2.4, the relationships among the five types of sufficient conditions for topology-preserving reductions are reviewed.

2.1 Sufficient Conditions for 2D Pictures

In this section sufficient conditions for topology-preserving reductions. These results are valid for all the given five types of 2D pictures. Configuration-based conditions provide methods of verifying that an operator preserves the topology, and point-based conditions allow us to generate directly topology-preserving operators.

For reasons of scope, our sufficient conditions for topology-preserving additions [119, 122, 126] and mixed operators [123, 126] are not presented here.

2.1.1 Configuration-Based Conditions for Reductions

In [272], Ronse gave a sufficient condition for topology-preserving reductions in $(2, 1)$ pictures on $\mathbb{Z}^2$ (see Theorem 1.2.6).

Condition 2 of Theorem 1.2.6 examines the simplicity of a set of two points. Thanks to the following (absolutely general and dimensionless) lemma stated by Kardos and Palágyi, it can be simplified:

Lemma 2.1.1 [122, 124] *Let p and q be two (black) simple points in an arbitrary picture. If p remains simple after the deletion of q, then q remains simple after the deletion of p.*

Lemma 2.1.1 can be rephrased as follows:

- If a simple set is formed by two simple points, then both possible sequences of its elements are simple sequences.
- If two black points p and q are simple in a picture, then the following two statements are equivalent:
  - p is simple after the deletion of q (i.e., $\langle q, p\rangle$ is a simple sequence).
  - q is simple after the deletion of p (i.e., $\langle p, q\rangle$ is a simple sequence).

In other words, the simplicity of a set of two simple points can be decided by testing just one sequence/permutation of its elements. Hence, no 'repechage' is needed.

Condition 3 of Theorem 1.2.6 examines objects contained in a 2×2 square. It is easy to check that there are exactly ten such objects (see Figure 2.1).

With the help of Lemma 2.1.1 and Figure 2.1, Ronse's condition (i.e., Theorem 1.2.6) can be rephrased as follows:

Theorem 2.1.1 [122, 124, 208] *A parallel reduction R acting on $(2, 1)$ pictures on $\mathbb{Z}^2$ is topology-preserving, if all the following conditions hold:*

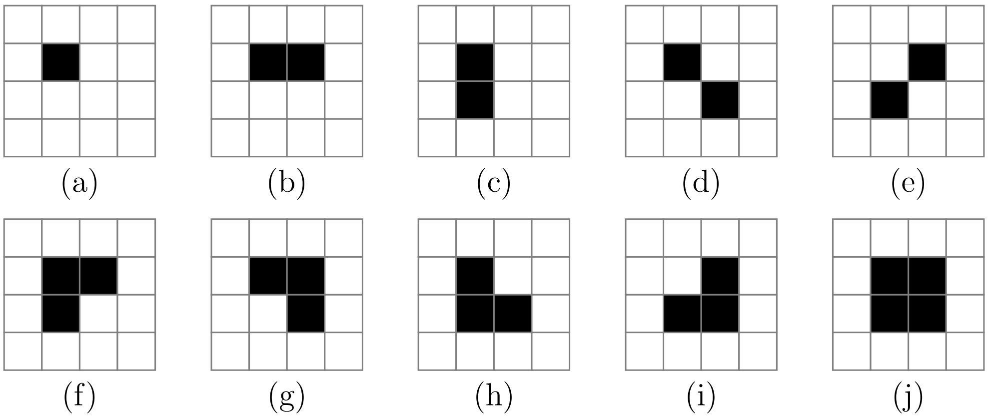


Figure 2.1: The ten possible objects contained in a 2×2 square.

1. *Only simple points are deleted by* R.

2. *For any two* 1*-adjacent black points p and q that are deleted by* R*, p is simple after the deletion of q.*

3. R *never deletes completely any objects shown in Figure 2.1(d) – Figure 2.1(j).*

Note that the object in Figure 2.1(a) is not deleted by Condition 1 of Theorem 2.1.1, and the next two objects shown in Figure 2.1(b) and Figure 2.1(c) cannot be deleted by Condition 2 of Theorem 2.1.1, therefore, Condition 3 of Theorem 2.1.1 is satisfied if the remaining seven objects shown in Figure 2.1(d) – Figure 2.1(j) are not deleted completely.

Kardos and Palágyi extended Theorem 2.1.1 to all the given five types of 2D pictures [122, 126]. They introduced two notions: the maximal set composed of mutually 2-adjacent points on $\mathcal{V}$ is called a *unit element* ($\mathcal{V} = \mathcal{T}, \mathbb{Z}^2, \mathcal{H}$) (see Figure 2.2). An object in a $(2, 1)$ picture on $\mathcal{V}$ is said to be a *small object* if it is contained in a unit element, it is not singleton, and it is not formed by two 1-adjacent points (see Figures 2.3-2.5).

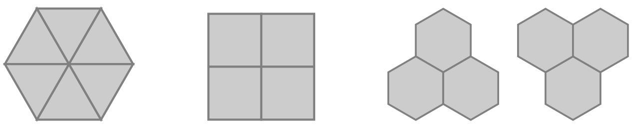


Figure 2.2: All the possible unit elements on grids $\mathcal{T}$, $\mathbb{Z}^2$, and $\mathcal{H}$.

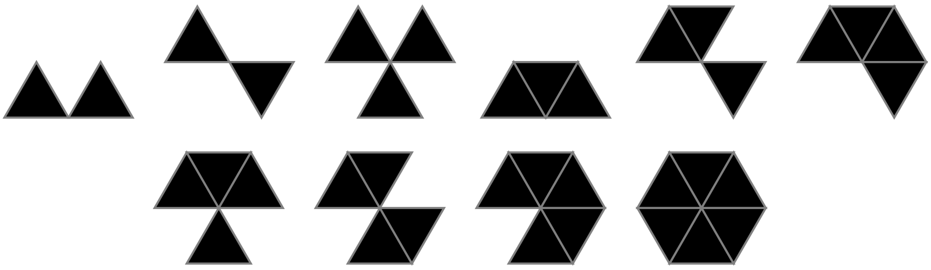


Figure 2.3: Base small objects in $(2,1)$ pictures on $\mathcal{T}$. All their rotated and reflected versions give all the possible 50 cases.

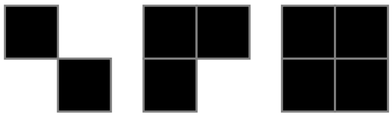


Figure 2.4: Base small objects in $(2,1)$ pictures grid $\mathbb{Z}^2$. All their rotated and reflected versions give all the possible 7 cases. (They correspond to the seven objects shown in Figure 2.1(d) – Figure 2.1(j).)

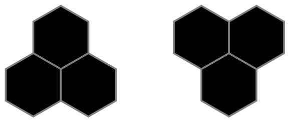


Figure 2.5: The two possible small objects in $(1,2) = (2,1)$ pictures on $\mathcal{H}$.

Theorem 2.1.2 [122, 126] *A parallel reduction* R *acting on* $(k, \bar{k})$ *pictures on* $\mathcal{V}$ *is topology-preserving, if all the following conditions hold:*

1. *Only simple points are deleted by* R.
2. *For any two* $\bar{k}$*-adjacent black points* p *and* q *that are deleted by* R, p *is simple after the deletion of* q.
3. *For the* $(k, \bar{k}) = (2,1)$ *case, no small object is deleted completely by* R.

2.1.2 Point-Based Conditions for Reductions

Condition 2 of Theorem 2.1.2 takes pairs of $\bar{k}$-adjacent deleted points into consideration, and Condition 3 deals with small objects. Hence, that theorem states a *configuration-based* condition, and just provides a method of verifying that a formerly constructed parallel reduction preserves the topology, rather than a methodology, for constructing topology-preserving reductions. That is why we proposed *point-based* conditions that directly provide deletion rules of topology-preserving reductions, and allow us to construct topology-preserving thinning algorithms.

2.1.2.1 Symmetric Conditions

Kardos and Palágyi proposed the following theorem that states the deletability of individual points:

Theorem 2.1.3 [120, 124, 126] *A parallel reduction acting on $(k, \bar{k})$ pictures on $\mathcal{V}$ is topology-preserving, if each point p deleted by that reduction satisfies the following conditions:*

1. *Point p is simple for B.*
2. *For any point $q \in N_{\bar{k}}^{*\mathcal{V}}(p) \cap B$ being simple for B, point p is simple for $B \setminus \{q\}$.*
3. *For the $(k, \bar{k}) = (2, 1)$ case, p is not an element of a small object.*

2.1.2.2 Asymmetric Conditions

Conditions of Theorem 2.1.3 may be viewed as *symmetric* since elements in pairs of $\bar{k}$-adjacent points and small objects are not distinguished.

Let us focus on the addressing schemes shown in Figure 2.6, which maps every point in $\mathbb{Z}^2$ and $\mathcal{H}$ to a pair of coordinates. The *lexicographical order relation* '$\prec$' between two distinct points $p = (p_x, p_y)$ and $q = (q_x, q_y)$ is defined as follows: $p \prec q \;\Leftrightarrow\; (p_y < q_y) \vee ((p_y = q_y) \wedge (p_x < q_x))$. Let Q be a finite subset of $\mathcal{V}$. Then, point $p \in Q$ is said to be the *smallest element* of Q if for any $q \in Q \setminus \{p\}$, $p \prec q$.

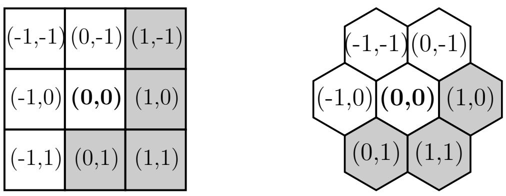


Figure 2.6: Feasible addressing schemes for the grids $\mathbb{Z}^2$ and $\mathcal{H}$. All points q in $N_2^{*\mathbb{Z}^2}(p)$ and $N_2^{*\mathcal{H}}(p)$ such that $p \prec q$ are depicted in gray, where p is the central point with coordinates $(0, 0)$.

With the help of the proposed ordering, Kardos and Palágyi gave the following *asymmetric point-based* condition for topology-preserving reductions:

Theorem 2.1.4 [120, 124, 248] *A parallel reduction* R *acting on* $(k, \bar{k})$ *pictures on* $\mathcal{V}$ *is topology-preserving, if the following conditions hold for each point* p *deleted by* R*:*

1. *Point* p *is simple for* B*.*
2. *For any point* $q \in N^{*\mathcal{V}}_{\bar{k}}(p) \cap B$ *that is simple for* B *and* $p \prec q$*, point* p *is simple for* $B \setminus \{q\}$*.*
3. *For the* $(k, \bar{k}) = (2, 1)$ *case,* p *is not the smallest element of a small object.*

Note that Kardos and Palágyi marked the smaller point in the possible pairs of $\bar{k}$-adjacent points, and the smallest point in the possible small objects on $\mathcal{T}$ [124]. Therefore relation '$\prec$' on the triangular grid is also defined.

Our symmetric and asymmetric point-based sufficient conditions (see Theorems 2.1.3 and 2.1.4) allow us to frame the following reductions:

Definition 2.1.1 *Let* $\mathsf{R}^{\mathcal{V},(k,\bar{k})}_{symm}$ *be the reduction acting on* $(k, \bar{k})$ *pictures on* $\mathcal{V}$ *that deletes all points satisfying all conditions of Theorem 2.1.3.*

Definition 2.1.2 *Let* $\mathsf{R}^{\mathcal{V},(k,\bar{k})}_{asymm}$ *be the reduction acting on* $(k, \bar{k})$ *pictures on* $\mathcal{V}$ *that deletes all points satisfying all conditions of Theorem 2.1.4.*

Note that all derived reductions are evidently topology-preserving.

The supports (i.e., the minimal set of points whose values determine the operator) of the five pairs of reductions $(\mathsf{R}^{\mathcal{V},(k,\bar{k})}_{symm}, \mathsf{R}^{\mathcal{V},(k,\bar{k})}_{asymm})$ are shown in Figures 2.7–2.9. It can readily be seen that the asymmetric reductions can delete much more points than the symmetric ones derived from point-based sufficient conditions for topology preservation.

Note that Kardos and Palágyi gave an advanced asymmetric sufficient condition for topology-preserving reductions acting on $(2, 1)$ pictures on $\mathbb{Z}^2$ [127]. It is based on an ordered classification of border points.

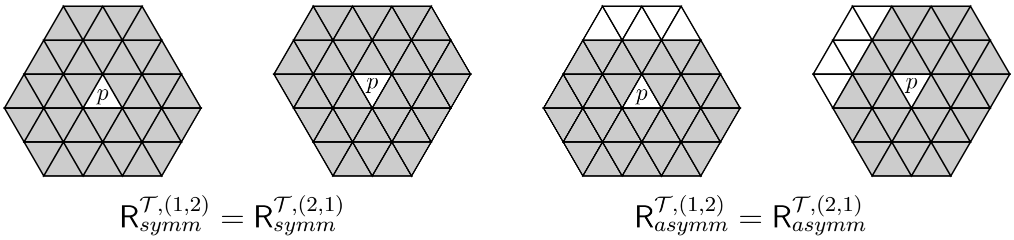


Figure 2.7: Supports of the reductions for 'Δ' and '∇' triangles on $\mathcal{T}$. The 36 and 31 points whose values determine the operators are depicted in gray.

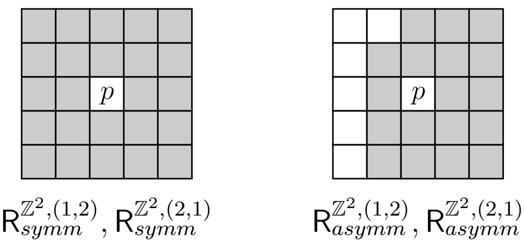


Figure 2.8: Supports of the derived reductions on $\mathbb{Z}^2$. The 24 and 18 points whose values determine the operators are depicted in gray.

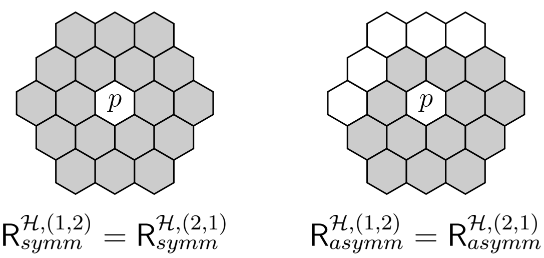


Figure 2.9: Supports of the derived reductions on $\mathcal{H}$. The 18 and 13 points whose values determine the operators are depicted in gray.

2.2 Point-Based Sufficient Conditions for 3D Reductions

Ma [168] and Kong [139] gave the foremost sufficient condition for topology-preserving reductions acting on 3D $(26, 6)$ pictures on $\mathbb{Z}^3$ (see Theorem 1.2.7). Later Palágyi and Kuba [221] proposed a simplified condition (see Theorem 1.2.8). Both conditions fall into the configuration-based category, and they provide general methods of verifying that a parallel reduction (or thinning algorithm) preserves the topology.

The following point-based sufficient conditions were derived from Theorem 1.2.8:

Theorem 2.2.1 *The parallel reduction* R *is topology-preserving for* $(26, 6)$ *pictures on* $\mathbb{Z}^3$ *if all the following conditions hold for any black point* p *in any picture* $\mathbf{P} = (\mathbb{Z}^3, 26, 6, B)$ *such that* p *is deleted by* R.

1. *Point* p *is simple in* $\mathbf{P}$.
2. *Let* $Q \subset B$ *be any set of simple points in* $\mathbf{P}$ *such that* $p \in Q$*, and* Q *is contained in a unit lattice square. Then* p *is simple in picture* $(\mathbb{Z}^3, 26, 6, B \setminus (Q \setminus \{p\}))$.
3. *Point* p *is not an element of any object* $C \subseteq B$ *in* $\mathbf{P}$ *such that* C *is contained in a unit lattice cube.*

It can readily be seen that if a reduction satisfies Condition i of Theorem 2.2.1 (i.e., the point-based result), Condition i of Theorem 1.2.8 (i.e., our former configuration-based result) holds for each $i \in \{1, 2, 3\}$.

Theorem 2.2.1 provides a symmetric point-based condition since elements contained in unit lattice squares and unit lattice cubes are not distinguished.

The *lexicographical order* relation '$\prec$' between two distinct points $p = (p_x, p_y, p_z)$ and $q = (q_x, q_y, q_z)$ in $\mathbb{Z}^3$ is defined as follows:

$$p \prec q \quad \Leftrightarrow \quad (p_z < q_z) \vee (p_z = q_z \wedge p_y < q_y) \vee (p_z = q_z \wedge p_y = q_y \wedge p_x < q_x).$$

Let $S \subset \mathbb{Z}^3$ be a finite set of points. Point $p \in S$ is the *smallest element* of S if for any $q \in S \backslash \{p\}$, $p \prec q$.

With the help of the lexicographical ordering we stated the following asymmetric point-based condition:

Theorem 2.2.2 [236] *A parallel reduction* R *is topology-preserving for* $(26, 6)$ *pictures on* $\mathbb{Z}^3$ *if all the following conditions hold for any black point* p *in any picture* $\mathbf{P} = (\mathbb{Z}^3, 26, 6, B)$ *such that* p *is deleted by* R.

1. *Point p is simple in $\mathbf{P}$.*

2. *Let $Q \subseteq B$ be any set of simple points in $\mathbf{P}$ such that $p \in Q$, and Q is contained in a unit lattice square. Then point p is simple in picture $(\mathbb{Z}^3, 26, 6, B\setminus(Q\setminus\{p\}))$, or p is not the smallest element of Q.*

3. *Point p is not the smallest element of any object $C \subseteq B$ in $\mathbf{P}$ such that C is contained in a unit lattice cube.*

2.3 Equivalent Sequential and Parallel Operators

All above mentioned sufficient conditions for topology-preserving operators examine some configurations of deletable points or individual deletable points. The author proposed a novel condition that takes the alteration rules of operators into consideration [238, 239, 241]. This approach is momentous since:

- it is universal (i.e., it is valid for arbitrary binary pictures);
- it provides a condition not only for topology-preserving reductions, but also for topology-preserving additions and mixed operators;
- it provides a verification method to design topology-preserving thinning algorithms;
- it allows us to implement parallel thinning algorithms directly on conventional sequential computers.

Here our attention is focussed on reductions that play a key role in various topological algorithms, e.g., thinning [89, 147, 324] or reductive shrinking [90].

*Parallel reduction*s can change a set of black points simultaneously, while *sequential reduction*s traverse the black points of a picture, and focus on the actually visited single point for possible deletion. These two absolutely dissimilar strategies are illustrated in Algorithm 1 and Algorithm 2.

Algorithm 1: Parallel reduction

Input: set of black points B,
constraint set $C \subset B$, and
deletion rule R
Output: set of black points PB
// selecting interesting points
$X \leftarrow B \setminus C$
// determining deletable points
$D \leftarrow \{\, p \mid p \in X \text{ and } \mathsf{R}(p, B, C) = \textbf{true} \,\}$
// deletion
$PB \leftarrow B \setminus D$

Thinning algorithms generally classify the set of black points in input pictures into two (disjoint) subsets: the deletion rule associated with a phase

Algorithm 2: Sequential reduction

Input: set of black points B,
constraint set $C \subset B$,
permutation (total ordering) Π of elements in $B \setminus C$
deletion rule R
Output: set of black points PB
// selecting interesting points
$X \leftarrow B \setminus C$
// setting initial black points
$SB \leftarrow B$
// traversal of X according to Π
foreach $p \in X$ **do**
  if $\mathsf{R}(p, SB, C) = \textbf{true}$ **then**
    // deletion
    $SB \leftarrow SB \setminus \{p\}$

of an algorithm is evaluated for the elements of its set of *interesting points*, and black points in its *constraint set* are not taken into consideration. That is why Algorithm 1 and Algorithm 2 examine a constraint set $C \subset B$ (as an input parameter) and its complementary $X = B \setminus C$ as a set of interesting points.

An interesting point $p \in X$ is *deletable*, if $\mathsf{R}(p, Y, C) = \textbf{true}$, where Y denotes the set of black points in the (actual) picture, i.e., $Y = SB \subseteq B$ in sequential reductions (see Algorithm 2), and $Y = B$ in the parallel case (see Algorithm 1). Therefore, in a parallel reduction, the initial picture is examined when the deletion rule is evaluated. In contrast, the picture is dynamically altered when a sequential reduction is performed. We should add that elements of the constraint set C are omitted when the deletion rule R is evaluated. For practical purposes we deal with finite pictures (i.e., B contains finitely many points).

The sequential approach suffers from the drawback that different visiting orders of interesting points may provide different results. A deletion rule R is said to be *order-independent* if the result of Algorithm 2 is uniquely specified by R (i.e., the result of Algorithm 2 does not depend on the order Π in which the interesting points are selected by the **foreach** loop) [114, 238, 266].

Definition 2.3.1 [239] *Two reductions are called* equivalent *if they produce the same result for each input picture. A deletion rule is said to be* equivalent *if it provides a pair of equivalent parallel and sequential reductions.*

Recall that the support of a deletion rule R applied at a point is a minimal set of points whose values determine whether the examined points are deleted by R from a picture. Note that thinning and reductive shrinking algorithms use local supports with 'small' diameters. Let us denote the support of the deletion rule R with respect to a point p by $\mathbf{S}_{\mathsf{R}}(p)$. It is clear that $\mathsf{R}(p, Y, C) = \mathsf{R}(p, Y \cap \mathbf{S}_{\mathsf{R}}(p), C \cap \mathbf{S}_{\mathsf{R}}(p))$.

The author introduced two special classes of deletion rules:

Definition 2.3.2 [239] *Let* R *be a deletion rule, let B be the set of black points in a picture, let $p \in B \setminus C$ be an interesting point with respect to the constraint set $C \subset B$, and let us assume that* $\mathsf{R}(p, B, C) = \textbf{true}$ *(i.e., p can be deleted by* R*). Then R is* general *if* $\mathsf{R}(q, B, C) = \mathsf{R}(q, B \setminus \{p\}, C)$ *for any point $q \in B \setminus C$.*

In other words, a deletion rule is general if the deletability of any point does not depend on the 'color' of any deletable point. It is obvious that a method of verifying that a deletion rule R is general may ignore each point $q \notin \mathbf{S}_{\mathsf{R}}(p)$.

Definition 2.3.3 [239] *A deletion rule is* general-simple *if it is general, and it deletes only simple points.*

The following theorem gives a necessary and sufficient condition for order-independent deletion rules:

Theorem 2.3.1 [239] *A deletion rule is order-independent if and only if it is general.*

Figure 2.10 presents an example of a non-general deletion rule. Hence, it is not order-independent by Theorem 2.3.1.

Let us see a useful property of general deletion rules.

Lemma 2.3.1 [239] *Let* R *be a general deletion rule. Then the parallel and the sequential reductions with* R *are equivalent.*

We are now ready to state a condition for equivalent deletion rules as an immediate consequence of Lemma 2.3.1.

Theorem 2.3.2 [239] *A deletion rule is equivalent if it is general.*

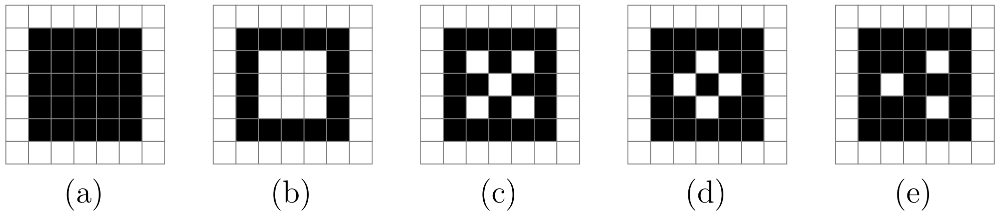


Figure 2.10: Example of a non-general deletion rule that removes interior points from $(2, 1)$ pictures on $\mathbb{Z}^2$. We can state that the parallel and the sequential reductions with that rule cannot produce the same result for the original picture (a). The result of the parallel reduction (b), and three of the possible pictures produced by the sequential reduction with various visiting orders (c)–(e).

Note that Theorem 2.3.2 gives a sufficient (but not necessary) condition for equivalent deletion rules, since a non-general deletion rule may specify a pair of equivalent parallel and sequential reductions. Examine the deletion rule on $\mathbb{Z}^2$ that deletes a black point if its southern neighbor is black. It is clear that it is not order-independent, hence it is not general by Theorem 2.3.1. It can readily be seen that if we apply the row-by-row visiting order, then that sequential reduction with that deletion rule is equivalent to the parallel reduction with the same rule.

The following theorem provides a novel sufficient condition for topology-preserving reductions in arbitrary pictures.

Theorem 2.3.3 [239] *A parallel reduction is topology-preserving if its deletion rule is general-simple.*

Examine the deletion rule $\mathsf{R}_{\text{border}}$ that deletes all border points from $(2, 1)$ pictures on $\mathbb{Z}^2$, and assume that the constraint set is formed by the interior points. It can readily be seen that $\mathsf{R}_{\text{border}}$ is general (and order-independent), hence it provides a pair of equivalent parallel and sequential reductions. Since some border points are not simple, $\mathsf{R}_{\text{border}}$ is not general-simple, and the specified parallel and sequential reductions are not topology-preserving (see Figure 2.11).

In the additional example, deletion rule $\mathsf{R}_{\text{simple}}$ deletes all simple points from $(2, 1)$ pictures on $\mathbb{Z}^2$, and the constraint set $C \subset B$ is formed by the interior points in B. In this case, the parallel reduction (see Algorithm 1) is not topology-preserving, since simple points may form non-simple sets. Notice that a black component is disconnected into three components and the three white components are merged. Algorithm 2 with respect to $\mathsf{R}_{\text{simple}}$ may specify numerous topology-preserving sequential reductions as it is illustrated

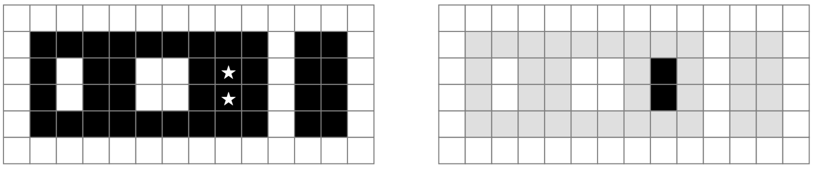


Figure 2.11: Example of a general deletion rule that is not general-simple. The sample original picture (left), where interior points (i.e., elements of the constraint set) are marked '$\star$'. The picture produced by the parallel and the (unique) sequential reductions (right) with the general deletion rule $\mathsf{R}_{\text{border}}$ (right). Deleted points are depicted in light gray. These reductions are not topology-preserving since one black component is completely deleted and the three white components are merged.

by Figure 2.12, hence deletion rule $\mathsf{R}_{\text{simple}}$ is not general (and it is not general-simple).

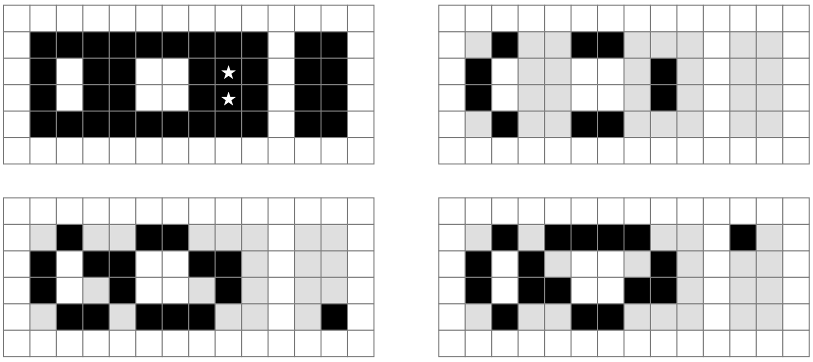


Figure 2.12: Example of a deletion rule that is not general-simple. The sample original picture (top-left), where interior points (i.e., elements of the constraint set) are marked '$\star$'. The picture produced by the parallel reductions (top-right) with deletion rule $\mathsf{R}_{\text{simple}}$. Two of the possible results generated by the sequential reductions with $\mathsf{R}_{\text{simple}}$ (bottom). The bottom-left picture is the result with respect to the row-by-row visiting order, and we got the bottom-right picture by applying the opposite ordering (i.e., scanning from the bottom upwards, and right to left on each row). Deleted points are depicted in light gray.

The following theorem summarizes our most important results concerning general-simple deletion rules:

Theorem 2.3.4 *Let* R *be general-simple deletion rule. Then the following conditions hold:*

1. *The parallel reduction with deletion rule* R *(see Algorithm 1) is topology-preserving.*

2. *The sequential reduction with deletion rule* R *(see Algorithm 2) is topology-preserving.*

3. *The parallel and the sequential reductions with deletion rule* R *are equivalent.*

We should add that the author extended those results to mixed operators (that also include reductions and additions), and he proposed an equivalent contour-smoothing algorithm [241].

In [239], the author gave a method of verifying that a deletion rule provides a pair of topology-preserving and equivalent parallel and sequential reductions. With the help of that method, we managed to prove that deletion rules of some 2D and 3D thinning algorithms are general-simple [242, 243, 244, 246].

2.4 Relationships Among Conditions for Reductions

Let us review the existing sufficient conditions for topology-preserving reductions that fall into the following five categories:

- configuration-based (see Theorems 1.2.6, 1.2.7, 1.2.8, and 2.1.2),
- point-based (see Theorems 2.1.3, 2.1.4, 2.2.1, and 2.2.2),
- P-simple sets (see Theorem 1.2.4),
- hereditarily simple sets (see Theorem 1.2.5), and
- general-simple deletion rules (see Theorem 2.3.4).

Next, the relationships among these five approaches are presented.

2.4.1 Hereditarily Simple Sets and P-Simple Sets

In [140], Kong and Gau proved that the two kinds of sufficient conditions for topology-preserving reductions based on P-simple sets (i.e., Theorem 1.2.4) and hereditarily simple sets (i.e., Theorem 1.2.5) are equivalent.

Theorem 2.4.1 [140] *A set of black points in a picture is hereditarily simple if and only if it is a P-simple set in that picture.*

2.4.2 Configuration-Based and Point-Based Conditions

Let us now examine the relationship between the point-based and the configuration-based conditions (see Sections 2.1 and 2.2):

- In the case of reductions acting on $(k, \bar{k})$ pictures on $\mathcal{V}$:
  It can readily be seen that if a reduction satisfies Condition i of Theorem 2.1.3 (i.e., the symmetric point-based result) or Theorem 2.1.4 (i.e., the asymmetric point-based result), Condition i of Theorem 2.1.2 (i.e., the configuration-based result) holds for each $i \in \{1, 2, 3\}$.
- In the case of reductions acting on $(26, 6)$ pictures on $\mathbb{Z}^3$:
  It is clear that if a parallel reduction satisfies Condition i of Theorem 2.2.1 (i.e., the symmetric point-based result) or Theorem 2.2.2 (i.e., the asymmetric point-based result), Condition i of Theorem 1.2.8 (i.e., the configuration-based result) holds for each $i \in \{1, 2, 3\}$.

Thus we can state the following theorem.

Theorem 2.4.2 *If a reduction satisfies a (symmetric or asymmetric) point-based condition, it satisfies the corresponding configuration-based condition as well.*

2.4.3 Configuration-Based Conditions and P-Simple Sets

In [248], Palágyi and Kardos proved the following theorem:

Theorem 2.4.3 [248] *If a reduction acting on* $(k, \bar{k})$ *pictures on* $\mathcal{V}$ *deletes only P-simple points, all conditions of Theorem 2.1.2 (i.e., the configuration-based result) are satisfied.*

It is clear that the analogous statement is valid for reductions acting on $(26, 6)$ pictures on $\mathbb{Z}^3$.

Theorem 2.4.4 *If a reduction acting on* $(26, 6)$ *pictures on* $\mathbb{Z}^3$ *deletes only P-simple points, all conditions of Theorem 1.2.8 (i.e., the configuration-based result) are satisfied.*

We managed to prove that the contrary of Theorem 2.4.3 holds for reductions acting on the given five types of pictures on the regular 2D grids:

Theorem 2.4.5 [248, 249] *If a reduction acting on* $(k, \bar{k})$ *pictures on* $\mathcal{V}$ *satisfies all conditions of Theorem 2.1.2, it deletes only P-simple sets.*

In [24], Bertrand proposed a two-step (topology-preserving) thinning technique that is based on P-simple points. One phase/reduction of the iterative thinning process is performed as follows:

1. A set of black points Q in the actual picture is (somehow) chosen and labeled.

2. All P-simple points in Q are deleted (simultaneously).

Note that Step 2 concerns *tricolor* pictures (say: the value '0' corresponds to white points, the value '1' is assigned to (black) points that are not in Q, and value '2' corresponds to (black) points in Q). Hence this two-step scheme is both space- and time-consuming.

Theorems 2.4.2 and 2.4.5 provide a single-step 2D thinning scheme that deletes P-simple points as well [248]. The deletion rule of a phase of the iterative thinning process can be directly constructed by combining the reduction

$\mathsf{R}_{symm}^{\mathcal{V},(k,\bar{k})}$ (see Definition 2.1.1) or $\mathsf{R}_{asymm}^{\mathcal{V},(k,\bar{k})}$ (see Definition 2.1.2) with different thinning strategies (i.e., *fully parallel*, *subiteration-based*, and *subfield-based* [89]) and various geometric constraints (say *endpoint*s [89] or *isthmus*es [26]). The generated deletion rule is a common *Boolean function* that is to be evaluated for the neighborhood of the points in question in binary (*two-level*) pictures. As this Boolean function can be stored in a pre-calculated *look-up-table*, the proposed single-step scheme can be implemented efficiently.

An immediate consequence of Theorem 2.4.5 is that the five pairs of reductions $(\mathsf{R}_{symm}^{\mathcal{V},(k,\bar{k})}, \mathsf{R}_{asymm}^{\mathcal{V},(k,\bar{k})})$ directly mark out P-simple sets in 2D pictures.

Note that there may be numerous P-simple sets in a picture. That is why Kardos and Palágyi introduced the concept of a maximal P-simple set [127]:

Definition 2.4.1 [127] *A P-simple set $Q \subset B$ is a* maximal P-simple set *for B if for any $q \in (B \setminus Q)$, $Q \cup \{q\}$ is not a P-simple set for B.*

In [127], we gave a novel asymmetric point-based sufficient condition for topology-preserving reductions acting on $(2, 1)$ pictures on $\mathbb{Z}^2$, and proved that this condition directly marks out a maximal P-simple set.

2.4.4 P-Simple Sets and General-Simple Deletion Rules

Let us consider two important properties of P-simple sets and general-simple deletion rules:

Proposition 2.4.1 [249] *Any set of points $Q \subset B$ is a P-simple set for B if and only if all possible permutations of Q form simple sequences.*

Proposition 2.4.2 [249] *All permutations of the elements in the set of points deleted by a (sequential or parallel) reduction with a general-simple deletion rule form simple sequences.*

The following theorem is an immediate consequence of Proposition 2.4.1 and Proposition 2.4.2.

Theorem 2.4.6 [249] *Reductions with general-simple deletion rules delete P-simple sets.*

Note that the contrary statement does not hold:

Theorem 2.4.7 [249] *The deletion rule of a reduction that deletes only P-simple sets may not be general-simple.*

In [245], the author constructed a special deletion rule that deletes only simple sets, and he proved that it is general-simple. In [249], the author showed that for each simple set Q in a picture, there is a general-simple deletion rule that deletes Q from this picture.

2.4.5 Summarized Relationships

We summarize the relationships among the five types of sufficient conditions for topology-preserving reductions below with the help of Figure 2.13. Note that three of them (namely: deletion of simple sets, deletion of hereditarily simple sets, and general-simple deletion rules) are absolutely universal, and the relationships among them are valid for arbitrary pictures.

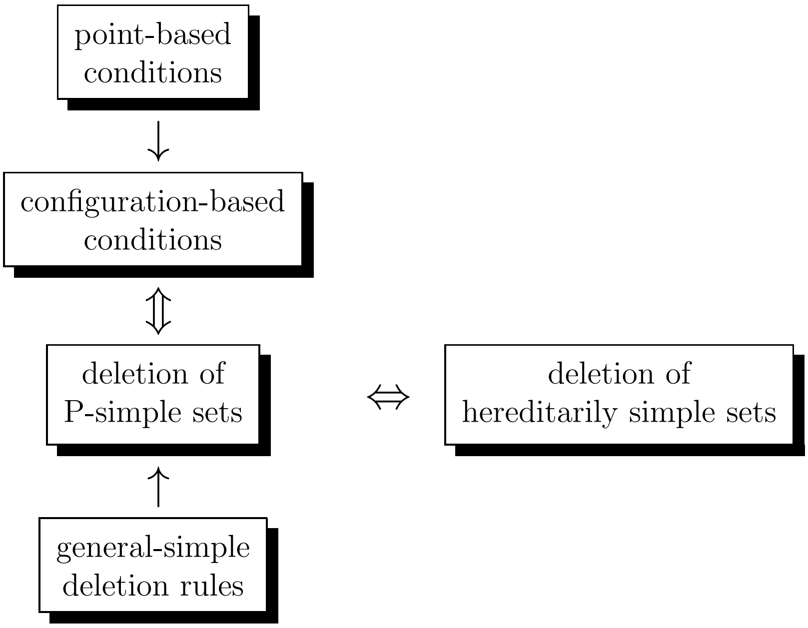


Figure 2.13: How the five kinds of sufficient conditions for topology-preserving reductions are related for the given five types of 2D pictures. Unfortunately, it is an open problem whether the contrary of Theorem 2.4.4 is valid for the examined type of 3D pictures.

Chapter 3

Advanced Thinning

As it is sketched in Subsection 1.3.2, thinning is a major skeletonization technique. It is an iterative object reduction process capable of producing all types of skeleton-like features in a topology-preserving way: some border points that satisfy certain topological and geometric constraints are deleted from the object, and the entire process is repeated until stability is reached (i.e., no points are deleted).

Parallel and sequential thinning algorithms are composed of parallel and sequential reductions, respectively. Parallel reductions (i.e., thinning phases) can delete all black points from a picture that satisfy their deletion rules simultaneously, while sequential reductions consider a single black point for possible deletion at a time.

From 2D and 3D pictures, *curve-thinning* algorithms are used to extract centerlines, and *kernel-thinning* algorithms are capable of extracting topological kernels. (Note that kernel-thinning algorithms are often referred to as *reductive shrinking* algorithms [90].) In the 3D case, *surface-thinning* algorithms produce medial surfaces.

Curve-thinning and surface-thinning algorithms do not delete some points that provide relevant geometrical information with respect to the shape of the object. *Endpoint-based* curve-thinning algorithms preserves *curve-endpoint*s, and surface-thinning algorithms never delete *surface-endpoint*s. *Isthmus-based* curve-thinning and surface-thinning algorithms preserve accumulated *isthmus*es [26] (i.e., generalization of curve- or surface-interior points). Kernel-thinning algorithms do not take geometric constraints into consideration.

Unfortunately, all skeletonization methods including thinning are rather sensitive to coarse object boundaries, hence the produced skeleton-like features generally contain some false segments. Unwanted skeletal parts are usually removed by a pruning process as a post-processing step [14, 87, 110, 144, 160, 192, 294, 298, 299, 323, 328]. In order to overcome this problem, Németh, Kardos, and Palágyi proposed a novel thinning scheme that uses iteration-by-iteration contour smoothing [207].

There are three major strategies for parallel thinning [89]: a *fully parallel* algorithm applies the same parallel reduction in each iteration step; a *subiteration-based* (also frequently referred to as *directional*) algorithm decomposes an iteration step into $k \geq 2$ successive parallel reductions according to k deletion directions, and a subset of border points associated with the actual direction can be deleted by a parallel reduction; and in *subfield-based* algorithms the digital space is partitioned into $s \geq 2$ subfields which are alternatively activated, at a given iteration step s successive parallel reductions assigned to these subfields are performed, and some black points in the active subfield can be designated for deletion.

We have constructed a number of 2D and 3D algorithms [112, 113, 114, 115, 116, 117, 118, 120, 121, 203, 204, 205, 206, 207, 208, 209, 210, 211, 219, 220, 221, 222, 224, 225, 229, 232, 233, 234, 235, 236, 237, 240, 243, 244, 246, 248, 252]. It should be stressed that all of them are topology-preserving.

In our 'conventional' parallel thinning algorithms [113, 219, 220, 221, 222, 225, 232, 233, 234], first the deletion rules were designed, then their topological correctness were verified with the help of configuration-based sufficient conditions for topology-preserving reductions (see Subsections 1.2.4 and 2.1.1). The remaining parallel algorithms [115, 120, 121, 203, 204, 205, 206, 208, 209, 210, 211, 229, 235, 236, 237, 243, 244, 246, 248] are based on a novel approach invented by the author: the deletion rules are derived from 'advanced' sufficient conditions (see Subsection 2.1.2, Section 2.2, and Section 2.3) combined with parallel thinning strategies and geometric constraints.

Thinning algorithms iterate reductions until stability is reached. If a border point is not deleted in an iteration step, it is examined again for possible deletion in the next step. Hence, it is important to characterize survival points whose rechecking is not needed in the remaining phases of the iterative process. Palágyi and Németh introduced the notions of a 2D-simplifier point [247], a weak-3D-simplifier point [251], and a fixpoint [250]. With the help of these concepts, computationally efficient implementation schemes were proposed for sequential endpoint-based thinning algorithms [251] and iterated equivalent reductions [250].

For reasons of scope, in this chapter, only some selected results concerning thinning are presented.

In Section 3.1, a general implementation scheme is presented. Although thinning is an iterative process, the author proposed an easy and computationally efficient implementation scheme for arbitrary (sequential and parallel) thinning algorithms.

Section 3.2 reviews a safe technique for designing topology-preserving parallel thinning algorithms. We describe 15 parallel 3D algorithms (5 curve-thinning, 5 surface-thinning, and 5 kernel-thinning ones) that are derived from our asymmetric point-based sufficient condition (see Theorem 2.2.2) combined with the three major parallel thinning strategies and three types of geometric constraints.

Four pairs of equivalent sequential and parallel subiteration-based 3D surface-thinning algorithms are reviewed in Section 3.3. They use the same deletion rule, but four types of constraints are taken into consideration.

Lastly, Section 3.4 presents two maximal 3D curve-thinning algorithms (i.e, algorithms that can produce centerlines containing only non-simple points and 3D-curve-endpoints). These algorithms have been successfully applied in several medical image processing applications.

3.1 A General Implementation Scheme

The author proposed a general and computationally efficient implementation scheme for arbitrary thinning algorithms [230, 231, 234]. His method utilizes the following properties of topology-preserving thinning:

- Only border points in the current picture are examined in each reduction / thinning phase (i.e., we do not have to evaluate the deletion rules for interior points).

- Only some simple points in the current picture may be deleted in each thinning phase (since all existing sufficient conditions for topology-preserving reductions require that).

- Since deletion rules of thinning algorithms use local supports, the corresponding Boolean functions can be evaluated for all possible configurations, and the results can be stored in (pre-calculated) look-up-tables.

The proposed method uses a list for storing the border points in the current picture, thus the repeated scans/traverses of the entire array (that stores the picture) are avoided. The pseudocode of collecting border points in the input picture (i.e., the initialization step of the thinning process) is described by Algorithm 3.

Algorithm 3: Collecting border points

Input: array A storing the picture to be thinned
Output: list *border_list* storing the border points in that picture
border_list $\leftarrow$ $<$ empty list $>$
foreach element p in array A **do**
    if $A[p] = 1$ and p is a border point **then**
        border_list $\leftarrow$ *border_list* $+ < p >$
        $A[p] \leftarrow 2$

In input array A, the value '1' corresponds to black points in the picture to be thinned, and the value '0' is assigned to white ones. In order to avoid storing more than one copy of a border point in *border_list*, a three-color picture is assumed in which the value '2' corresponds to border points to be checked in the forthcoming reductions / thinning phases.

Algorithm 4 and Algorithm 5 describe one phase of arbitrary sequential and parallel thinning algorithms, respectively.

Algorithm 4: Sequential thinning phase

```
Input:  array A storing the (input or interim) (m, n)-picture and
        list border_list storing the border points in that picture
Output: array A containing the result of the reduction and
        the updated border_list
foreach point p in border_list do
    if p is 'deletable' then
        // deletion
        A[p] ← 0
        border_list ← border_list − < p >
        // updating the list of border points
        foreach point q being n̄-adjacent to p do
            if A[q] = 1 then
                A[q] ← 2
                border_list ← border_list + < q >
```

Algorithm 5: Parallel thinning phase

```
Input:  array A storing the (input or interim) (m, n)-picture and
        list border_list storing the border points in that picture
Output: array A containing the result of the reduction and
        the updated border_list
// collecting deletable points
deletable_list ← < empty list >
foreach point p in border_list do
    if p is 'deletable' then
        border_list ← border_list − < p >
        deletable_list ← deletable_list + < p >
foreach point p in deletable_list do
    // deletion
    A[p] ← 0
    // updating the list of border points
    foreach point q being n̄-adjacent to p do
        if A[q] = 1 then
            A[q] ← 2
            border_list ← border_list + < q >
```

In parallel case (see Algorithm 5), a second list is used for storing the 'deletable' points of the current iteration (see Figure 3.1). In order to ensure that both the input and the output pictures of an iteration can be stored in a single array, the evaluation and the deletion phases are separated: first all 'deletable' points are added to the *deletable_list*, then they are deleted and *border_list* is updated accordingly.

In both the sequential and the parallel cases, if a border point is deleted, all interior points that are n-adjacent to it become border points. These brand new border points of the resulted picture are added to the *border_list*.

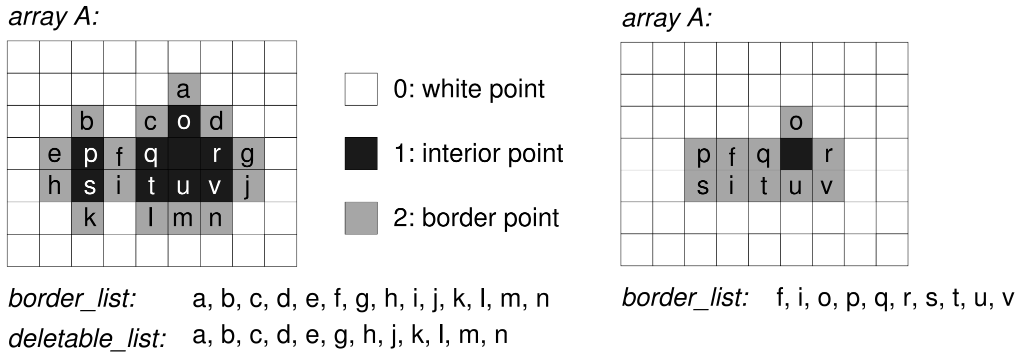


Figure 3.1: A thinning phase of an imaginary 2D fully parallel thinning algorithm acting on $(2, 1)$ pictures on $\mathbb{Z}^2$. Array A contains 14 border points and 12 of them are deletable points (left). There are 10 border points in the resultant array A after this reduction, because 8 interior points turn into border points (right).

In Algorithm 4 and Algorithm 5, the thinning process terminates when no more points can be deleted (i.e., stability is reached). After thinning, all points having a nonzero value in array A belong to the produced skeleton-like feature.

3.2 3D Parallel Thinning Algorithms Derived from Sufficient Conditions for Topology Preservation

A crucial issue in producing skeleton-like features is to ensure topology preservation. In [56], Couprie examined fifteen frequently cited 2D parallel thinning algorithms (acting on $(2, 1)$ pictures on $\mathbb{Z}^2$). He showed that five among these fifteen algorithms are not topology-preserving.

In order to prove that a parallel 2D or 3D thinning algorithm preserves the topology for all possible pictures, some configuration-based sufficient conditions for topology-preserving reductions have been proposed, see Theorem 1.2.6 [272], Theorem 2.1.1 [122, 124], Theorem 2.1.2 [122, 126], Theorem 1.2.7 [139, 168], and Theorem 1.2.8 [221]. The proofs in the literature that show that some thinning algorithms satisfy these conditions are generally combinatorial. Despite complex proofs, Palágyi [223], Lohou [163], just as Wang and Basu [354] detected that Ma and Sonka's 3D fully parallel curve-thinning algorithm [170] does not preserve the topology. Furthermore, Lohou [164] stated that Ma's 3D fully parallel surface-thinning algorithm [169] is not topology-preserving, and Lohou and Dehos [165] showed that the correction of Ma and Sonka's algorithm [170] proposed by Wang and Basu [354] does not preserve the topology, either.

That is why we proposed a safe technique for designing topology-preserving parallel thinning algorithms [208, 209, 236]. It is based on our (symmetric and asymmetric) point-based sufficient conditions for topology-preserving reductions (see Theorem 2.1.3, Theorem 2.1.4, Theorem 2.2.1, and Theorem 2.2.2). Our conditions directly provide deletion rules of topology-preserving reductions, and allow us to generate topology-preserving thinning algorithms by combining these reductions with parallel thinning strategies and geometric constraints (i.e., preserving endpoints or isthmuses) [115, 120, 121, 124, 203, 204, 205, 206, 208, 209, 210, 235, 236].

Next, we present 15 parallel 3D thinning (and shrinking) algorithms that are derived from the asymmetric point-based sufficient condition (see Theorem 2.2.2) combined with the three major strategies for parallel thinning (i.e., fully parallel, subiteration-based, and subfield-based) and three types of endpoints [236].

These endpoint-preserving thinning algorithms do not delete some border points that provide relevant geometrical information with respect to the shape of the object. We focused on the following three types of endpoints:

Definition 3.2.1 *There is no* endpoint of type **TK**.

To standardize the notations, reductive shrinking algorithms capable of producing topological kernels are examined as kernel-thinning algorithms, where no endpoint is preserved, hence we apply endpoints of type **TK** (i.e., the empty set of the endpoints).

Definition 3.2.2 *A black point p in picture $(\mathbb{Z}^3, 26, 6, B)$ is an* endpoint of type **CE** *(in short* 3D-curve-endpoint*) if $(N_{26}^{\mathbb{Z}^3}(p)\setminus\{p\}) \cap B$ contains exactly one point (i.e., p is 26-adjacent to exactly one further black point).*

Endpoints of type **CE** have been taken into consideration by numerous existing 3D curve-thinning algorithms [204, 205, 206, 219, 220, 222, 221, 231].

Definition 3.2.3 *A black point p in picture $(\mathbb{Z}^3, 26, 6, B)$ is an* endpoint of type **SE** *(in short* surface-endpoint*) if there is no interior point in $N_{26}^{\mathbb{Z}^3}(p) \cap B$.*

Note that endpoints of type **SE** are preserved by some existing surface-thinning algorithms [8, 180, 204, 205, 206, 233, 234, 235].

Throughout this section, the notation ε means that ε belongs to { **TK**, **CE**, **SE** }.

3.2.1 Fully Parallel Algorithms

In fully parallel algorithms, the same parallel reduction is applied in each iteration step [8, 165, 169, 170, 234, 354].

Algorithm 6 shows the scheme of the fully parallel thinning algorithm **3D-FP-ε**.

Algorithm 6: Fully parallel thinning algorithm **3D-FP-ε**

```
Input: picture (Z^3, 26, 6, X)
Output: picture (Z^3, 26, 6, Y)
Y ← X
repeat
    // one iteration step
    D ← { p | p is 3D-FP-ε-deletable in Y }
    Y ← Y \ D
until D = ∅;
```

'3D-FP-ε-deletable' points are defined as follows:

Definition 3.2.4 *A black point is* 3D-FP-ε-deletable *if it is not an endpoint of type ε, and all conditions of Theorem 2.2.2 hold.*

It is obvious that Algorithm **3D-FP-**ε is topology-preserving. (Note that all objects contained in a unit lattice cube are formed of endpoints of type **SE**. Hence, condition 3 of Theorem 2.2.2 can be ignored in algorithm 3D-FP-**SE**.)

Note that in [235], Palágyi and Németh proposed three fully parallel 3D surface-thinning algorithms, but those are based on an alternative point-based sufficient condition that is 'weaker' than Theorem 2.2.2.

3.2.2 Subiteration-Based Algorithms

In subiteration-based thinning algorithms, an iteration step is decomposed into $d \geq 2$ successive parallel reductions according to d deletion directions, and only border points of a certain kind can be deleted in each subiteration [89]. Since there are six kinds of major directions in 3D cases, 6–subiteration algorithms were generally proposed [23, 82, 150, 171, 197, 206, 219, 268, 338, 358]. Moreover, we have also proposed 3–subiteration [225, 233, 232], 8–subiteration [222], and 12–subiteration [221, 237] algorithms.

Here we present three examples of parallel 3D 6–subiteration thinning algorithms. Algorithm 7 sketches the scheme of algorithm **3D-6-SI-**ε that preserves endpoints of type ε.

Algorithm 7: Subiteration-based thinning algorithm **3D-6-SI-**ε

Input: picture $(\mathbb{Z}^3, 26, 6, X)$
Output: picture $(\mathbb{Z}^3, 26, 6, Y)$
$Y \leftarrow X$
repeat
    `// one iteration step`
    foreach $i \in \{$ **U**, **D**, **N**, **E**, **S**, **W** $\}$ **do**
        `// subiteration for deleting some` i`-border points`
        $D(i) \leftarrow \{\, p \mid p$ is a 3D-6-SI-i-ε-deletable point in $Y \,\}$
        $Y \leftarrow Y \setminus D(i)$
until $D(\mathbf{U}) \cup D(\mathbf{D}) \cup D(\mathbf{N}) \cup D(\mathbf{E}) \cup D(\mathbf{S}) \cup D(\mathbf{W}) = \emptyset$;

The ordered list of deletion directions $\langle \mathbf{U}, \mathbf{D}, \mathbf{N}, \mathbf{E}, \mathbf{S}, \mathbf{W} \rangle$ is examined in the proposed algorithm **3D-6-SI-**ε. Note that subiteration-based thinning algorithms are not invariant under the order of deletion directions (i.e., choosing different orders may provide various results).

In the first subiteration of our 6-subiteration thinning algorithms, all '3D-6-SI-**U**-ε-deletable' points are deleted simultaneously, and the set of '3D-6-

SI-$\mathbf{W}$-ε-deletable' points are deleted in the last (i.e., the 6th) subiteration. Now we lay down '3D-6-SI-$\mathbf{U}$-ε-deletable' points.

Definition 3.2.5 *A black point p in picture $(\mathbb{Z}^3, 26, 6, X)$ is* 3D-6-SI-$\mathbf{U}$-ε-deletable *if all of the following conditions hold:*

1. *Point p is a simple and $\mathbf{U}$-border point, but it is not an endpoint of type ε in picture $(\mathbb{Z}^3, 26, 6, X)$.*

2. *Let $\mathcal{A}(p)$ be the family of the following 13 sets (see Figure 3.2b):*
   $\{e\}$, $\{s\}$, $\{se\}$, $\{sw\}$, $\{dn\}$, $\{de\}$, $\{ds\}$, $\{dw\}$,
   $\{e, s\}$, $\{e, se\}$, $\{s, se\}$, $\{s, sw\}$,
   $\{e, s, se\}$.
   For any set A in family $\mathcal{A}(p)$ composed of simple and $\mathbf{U}$-border points, but no endpoints of type ε in picture $(\mathbb{Z}^3, 26, 6, X)$, point p remains simple in picture $(\mathbb{Z}^3, 26, 6, X \backslash A)$.

3. *Let $\mathcal{B}(p)$ be the family of the following 32 objects in picture $(\mathbb{Z}^3, 26, 6, X)$ (see Figure 3.2c):*
   $\{a, h\}$, $\{b, g\}$, $\{c, f\}$, $\{d, e\}$,
   $\{a, h, b\}$, $\{a, h, c\}$, $\{a, h, f\}$, $\{a, h, g\}$, $\{b, g, a\}$, $\{b, g, d\}$, $\{b, g, e\}$, $\{b, g, h\}$, $\{c, f, a\}$, $\{c, f, d\}$, $\{c, f, e\}$, $\{c, f, h\}$, $\{d, e, b\}$, $\{d, e, c\}$, $\{d, e, f\}$, $\{d, e, g\}$,
   $\{a, h, b, c\}$, $\{a, h, b, g\}$, $\{a, h, c, f\}$, $\{a, h, f, g\}$, $\{b, g, a, d\}$, $\{b, g, d, e\}$, $\{b, g, e, h\}$, $\{c, f, a, d\}$, $\{c, f, d, e\}$, $\{c, f, e, h\}$, $\{d, e, b, c\}$, $\{d, e, f, g\}$.
   Point p is not the smallest element of any object in $\mathcal{B}(p)$.

Note that the deletable points at the remaining five subiterations can be derived from '3D-6-SI-$\mathbf{U}$-ε-deletable' points (assigned to the deletion direction $\mathbf{U}$, see Definition 5) by reflections and rotations.

In [236], it is proved that Algorithm **3D-6-SI-ε** is topology-preserving.

3.2.3 Subfield-Based Algorithms

The third type of parallel thinning algorithms applies subfield-based technique [89]. In existing subfield-based parallel 3D thinning algorithms, $\mathbb{Z}^3$ is partitioned into two [172, 174, 205], four [175, 204], and eight [22, 204] subfields which are alternatively activated. At a given iteration step of an s-subfield algorithm, $s \geq 2$ successive parallel reductions associated to the s subfields are performed. In each parallel reduction, some border points in the active subfield can be designated to be deleted.

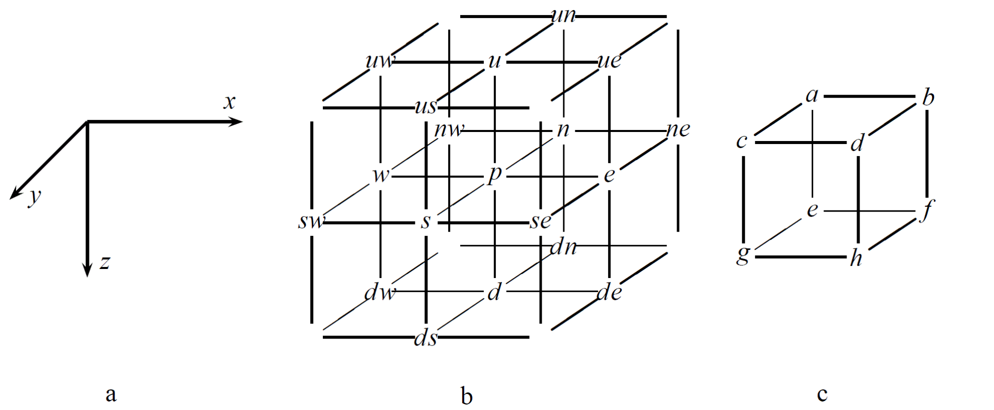


Figure 3.2: The examined right-handed 3D coordinate system (a). Notation for the points in $N_{18}^{\mathbb{Z}^3}(p)$ (b). Notation for the points in a unit lattice cube (c).

Let us denote $SF_s(i)$ the i-th subfield if $\mathbb{Z}^3$ is partitioned into s subfields ($s = 2, 4, 8;\ i = 0, \ldots, s-1$). $SF_s(i)$ is defined formally as follows:

$$
\begin{aligned}
SF_2(i) &= \{\ (p_x, p_y, p_z) \mid (p_x + p_y + p_z \bmod 2) = i\ \}, \\
SF_4(i) &= \{\ (p_x, p_y, p_z) \mid (p_x{+}1 \bmod 2) \cdot [2 \cdot (p_y \bmod 2) + (p_z \bmod 2)] + \\
&\qquad (p_x \bmod 2) \cdot [2 \cdot (p_y{+}1 \bmod 2) + (p_z{+}1 \bmod 2)] = i\ \}, \\
SF_8(i) &= \{\ (p_x, p_y, p_z) \mid 4 \cdot (p_x \bmod 2) + 2 \cdot (p_y \bmod 2) + (p_z \bmod 2) = i\ \}
\end{aligned}
$$

The examined divisions are illustrated in Figure 3.3.

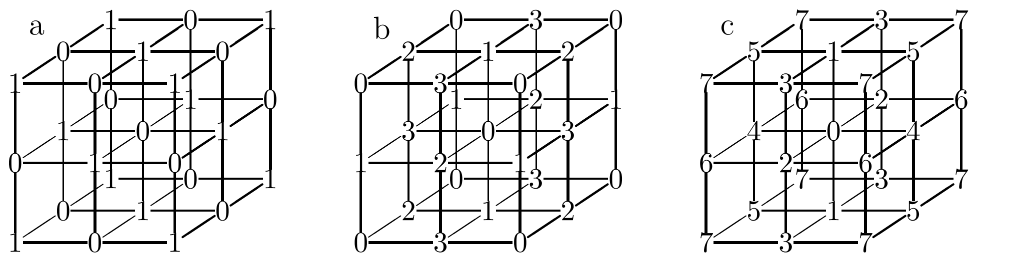


Figure 3.3: The divisions of $\mathbb{Z}^3$ into 2 (a), 4 (b), and 8 (c) subfields. If partitioning into s subfields is considered, then points marked 'i' are in the subfield $SF_s(i)$ ($s = 2, 4, 8;\ i = 0, 1, \ldots, s-1$).

Proposition 3.2.1 *For the 2-subfield case (see Figure 3.3a), two points p and $q \in N_{26}^{\mathbb{Z}^3}(p)$ are in the same subfield, if $q \in N_{18}^{\mathbb{Z}^3}(p) \backslash N_6^{\mathbb{Z}^3}(p)$.*

Proposition 3.2.2 *For the 4-subfield case (see Figure 3.3b), two points p and $q \in N_{26}^{\mathbb{Z}^3}(p)$ are in the same subfield, if $q \in N_{26}^{\mathbb{Z}^3}(p) \backslash N_{18}^{\mathbb{Z}^3}(p)$.*

Proposition 3.2.3 *For the 8-subfield case (see Figure 3.3c), two points p and $q \in N_{26}^{\mathbb{Z}^3}(p)$ are* not *in the same subfield.*

In order to reduce the noise sensitivity and the number of skeletal points (without overshrinking the objects), Németh, Kardos, and Palágyi introduced a novel subfield-based thinning scheme [205]. It takes the endpoints into consideration at the beginning of iteration steps, instead of preserving them in each parallel reduction as it is done in the conventional subfield-based thinning scheme.

Here we present nine parallel 3D subfield-based thinning algorithms. The scheme of the subfield-based parallel thinning algorithm **3D-s-SF-ε** with iteration-level endpoint checking using endpoint of type ε is sketched by Algorithm 8 ($s = 2, 4, 8$).

Algorithm 8: Subfield-based thinning algorithm **3D-s-SF-ε**

Input: picture $(\mathbb{Z}^3, 26, 6, X)$
Output: picture $(\mathbb{Z}^3, 26, 6, Y)$
$Y \leftarrow X$
repeat
    `// one iteration step`
    $E \leftarrow$ { $p \mid p$ is a border point, but not an endpoint of type ε in Y }
    for $i \leftarrow 0$ **to** $s-1$ **do**
        `// subfield` $SF_s(i)$ `is activated`
        $D(i) \leftarrow$ { $q \mid q$ is a 3D-SF-s-deletable point in $E \cap SF_s(i)$ }
        $Y \leftarrow Y \setminus D(i)$
until $D(0) \cup D(1) \cup \ldots \cup D(s-1) = \emptyset$;

The 3D-SF-s-deletable points are defined as follows ($s = 2, 4, 8$):

Definition 3.2.6 *A black point p is* 3D-SF-s-deletable *($s = 2, 4, 8$) in picture $(\mathbb{Z}^3, 26, 6, X)$ if all of the following conditions hold:*

1. *Point p is simple in $(\mathbb{Z}^3, 26, 6, X)$.*
2. *If $s = 2$, then point p is simple in picture $(\mathbb{Z}^3, 26, 6, X \backslash \{q\})$ for any simple point q such that $q \in N_{18}^{\mathbb{Z}^3}(p) \backslash N_{6}^{\mathbb{Z}^3}(p)$ and $p \prec q$.*
3. • *If $s = 2$, then point p is not the smallest element of the ten objects shown in Figure 3.4.*

- *If $s = 4$, then point p is not the smallest element of the four objects shown in Figure 3.5.*

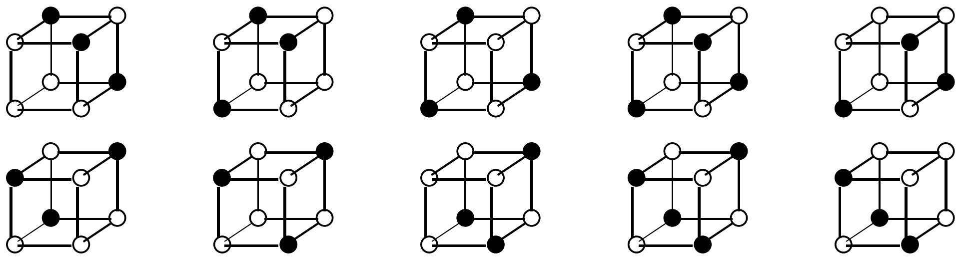


Figure 3.4: The ten objects that are taken into consideration by 2-subfield algorithms. Notations: each point marked '●' is a black point; each point marked '○' is a white point. (Note that each of these objects is contained in a unit lattice cube.)

Figure 3.5: The four objects examined by 4-subfield algorithms. Notations: each point marked '●' is a black point; each point marked '○' is a white point. (Note that each of these objects is contained in a unit lattice cube.)

In [236], it is proved that Algorithm **3D-s-SF-ε** ($s = 2, 4, 8$) is topology-preserving.

3.2.4 Implementation and Results

Here we outline a method for implementing the presented parallel thinning algorithms on conventional sequential computers. The proposed computationally efficient method follows our general framework (see Algorithm 3 and Algorithm 5), and uses a pre-calculated look-up-table to encode simple points. In addition, two lists are used to speed up the process: one for storing the border points in the current picture (since thinning can only delete border points, thus the repeated scans/traverses of the entire array storing the picture are avoided); the other list is to collect all deletable points in the current phase of the process. At each iteration, the deletable points are detected and deleted, and the list of border points is updated accordingly.

For simplicity, the pseudocode of the proposed 3D fully parallel thinning algorithms is given (see Algorithm 9). The subiteration-based and the subfield-based variants can be implemented in similar ways.

Algorithm 9: Fully parallel thinning

```
Input: array A and endpoint characterization ε
Output: array A
border_list ←< empty list >
foreach p in A do
    if p is border point then
        border_list ← border_list + < p >; A[p] ← 2;
repeat
    deleted ← 0
    deletable_list ← < empty list >
    foreach point p in border_list do
        if p is a simple point and not an endpoint of type ε then
            deletable_list ← deletable_list + < p >; A[p] ← 3;
    foreach point p in deletable_list do
        if deletion p does not satisfy Cond. 2 of Th. 2.2.2 then
            deletable_list = deletable_list − < p >; A[p] ← 2;
    foreach point p in deletable_list do
        if deletion p does not satisfy Cond. 3 of Th. 2.2.2 then
            deletable_list = deletable_list − < p >; A[p] ← 2;
    foreach point p in deletable_list do
        A[p] ← 0; deleted ← deleted+1;
        border_list ← border_list − < p >
        foreach point q that is 6-adjacent to p do
            if A[q] = 1 then
                border_list ← border_list + < q >; A[q] ← 2;
until deleted = 0;
```

The two input parameters of the process are array A which stores the input picture to be thinned and the type of the endpoint ε. In input array A, the value '1' corresponds to black points and the value '0' denotes white ones. According to the proposed scheme, the input and the output pictures can be stored in the same array, hence array A will contain the resultant structure.

First, the input picture is scanned and all the border points are inserted into the list *border_list*. Since only a small part of points in a usual picture belongs to the objects, the thinning process is much faster if we just deal with the set of border points in the actual picture. This subset of object points is stored in *border_list*. The *border_list* is then updated: if a border point is deleted, then all interior points that are 6-adjacent to it become border points. These new border points are added to the *border_list*. In order to avoid storing more than one copy of a border point in *border_list*, array A represents a four-colour picture during the thinning process: the value of '0' corresponds to the white points, the value of '1' corresponds to (black) interior points, the value of '2' is assigned to all (black) border points in the actual picture (added to *border_list*), and the value of '3' corresponds to points that are added to the *deletable_list* (i.e., a sublist of *border_list*).

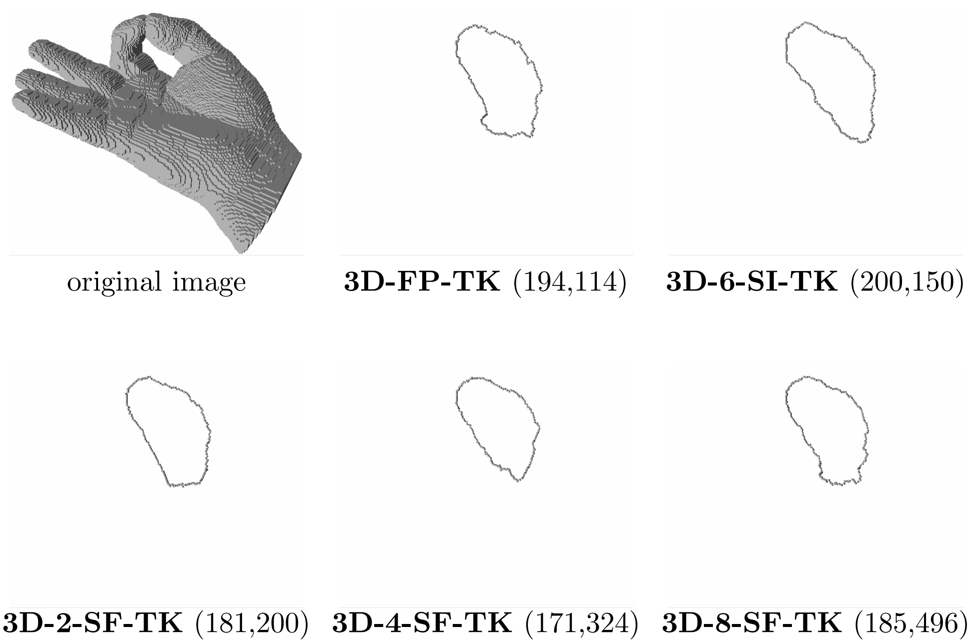
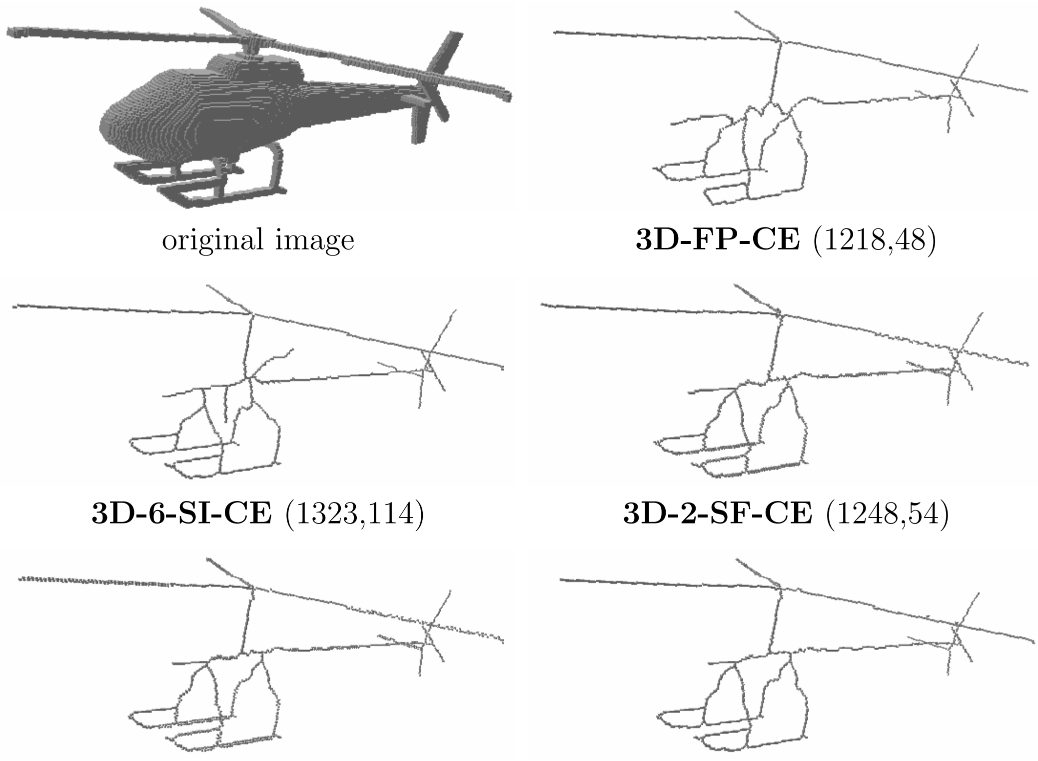


Figure 3.6: A $191 \times 96 \times 114$ image of a hand and its topological kernels produced by the five proposed parallel 3D kernel-thinning algorithms. The original image contains 455 295 black points. Since the original object contains a hole, there are holes in its topological kernels, too.

The kernel of the **repeat** cycle corresponds to one iteration step of the thinning process. The number of deleted points is stored in the variable called *deleted*. The process terminates when no more points can be deleted (i.e., no further changes occur). After thinning, all points having a nonzero value belong to the produced skeleton-like feature.



original image | **3D-FP-CE** (1218,48)

3D-6-SI-CE (1323,114) | **3D-2-SF-CE** (1248,54)

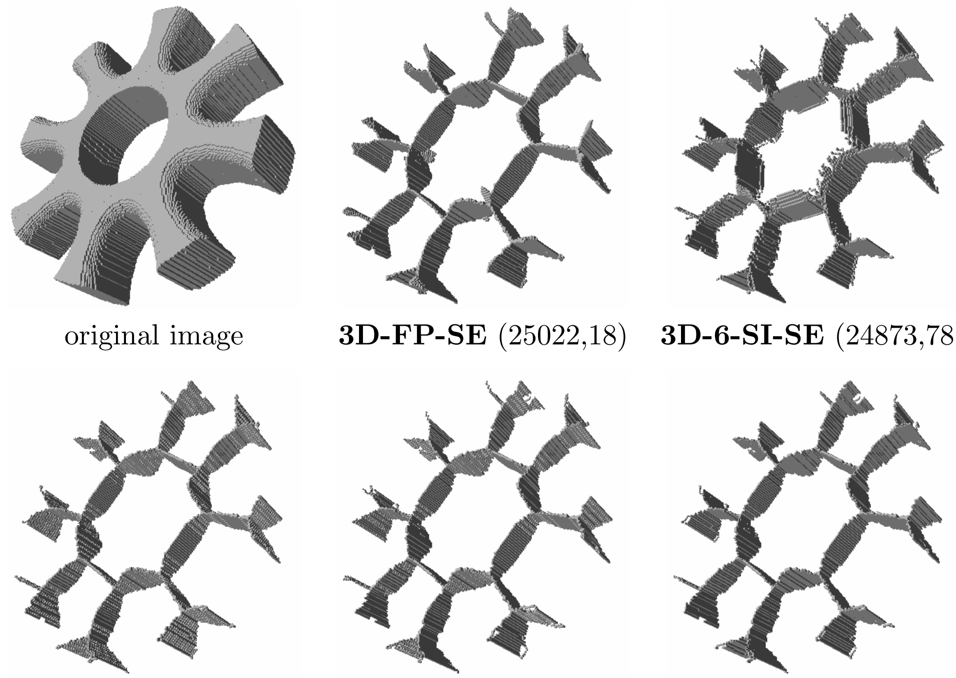
3D-4-SF-CE (1241,108) | **3D-8-SF-CE** (1230,208)

Figure 3.7: A $103 \times 381 \times 255$ image of a helicopter and its centerlines produced by the five proposed parallel 3D curvethinning algorithms. The original image contains 273 743 black points.

Simple points in $(26, 6)$ pictures can be locally characterized; the simplicity of a point p can be decided by examining the set $N_{26}^{\mathbb{Z}^3}(p)$ [137]. There are 2^{26} possible configurations in the $3 \times 3 \times 3$ neighborhood if the central point is not taken into consideration. Hence, we can assign an index (i.e., a non-negative integer code) for each possible configuration and address a pre-calculated (unit time access) look-up-table having 2^{26} entries of 1 bit in size, therefore, it requires only 8 megabytes storage space in memory.

By adapting this implementation method, our algorithms can be well applied in practice: they are capable of extracting skeleton-like features from large 3D pictures containing 1.000.000 object points within one second on a usual PC.

The presented $5 + 5 + 5 = 3 + 3 + 3 * 3 = 15$ algorithms were tested on objects of different shapes. Here we present some of them in Figures 3.6-3.8. The pairs of numbers in parentheses are the counts of object points in the produced skeleton-like feature and the *parallel speed* (i.e., the number of the required parallel reductions [89]).



original image | **3D-FP-SE** (25022,18) | **3D-6-SI-SE** (24873,78)

3D-2-SF-SE (17825,34) | **3D-4-SF-SE** (17235,72) | **3D-8-SF-SE** (17527,136)

Figure 3.8: A $45 \times 191 \times 191$ image of a gear and its medial surfaces produced by the five proposed parallel 3D surface-thinning algorithms. The original image contains 596 360 black points.

3.3 Equivalent Thinning

Equivalent thinning algorithms are composed of reductions with equivalent deletion rules (see Definition 2.3.1).

In [242], the author proved that the deletion rule of the 2D fully parallel thinning algorithm proposed by Manzanera et al. [181] is equivalent, and Palágyi, Németh, and Kardos proposed a pair of 2D equivalent sequential and parallel 4-subiteration (subiteration-based) thinning algorithms [243]. In the 3D case, Palágyi, Németh, and Kardos proposed four pairs of equivalent sequential and parallel 6-subiteration surface-thinning algorithms [244], and Palágyi and Németh presented a pair of equivalent sequential and fully parallel surface-thinning algorithms [246]. Furthermore, the author also showed that each fully parallel thinning algorithm with an equivalent deletion rule provides various subfield-based algorithms that are equivalent to the original fully parallel one [239].

For reasons of scope, only four pairs of equivalent sequential and parallel 3D subiteration-based surface-thinning algorithms (**S-6-SI-**i, **P-6-SI-**i) ($i = 1, 2, 3, 4$) acting on $(26, 6)$ pictures are presented in this section. All of these algorithms use the same deletion rule, but diverse pairs of them apply different constraint sets.

Thinning algorithms generally classify the set of black points of the input picture into two (disjoint) subsets: the set of *interesting point*s (i.e., potentially deletable points) for which the deletion rule associated with a thinning phase is evaluated, and the *constraint set* whose black points are not taken into consideration (i.e., safe points that cannot be deleted). Since a phase of a subiteration-based algorithm cannot delete interior points and border points that do not fall into the actual type, these points are certainly in the constraint set.

Our algorithms preserve some points that provide relevant geometrical information with respect to the shape of the object to be thinned. Here we consider the following four characterizations of protected points that are elements in the constraint set of a subiteration.

Definition 3.3.1 *A black point p in picture $(\mathbb{Z}^3, 26, 6, B)$ is a* protected point of type 1 *if there is no interior point in $N_6^{\mathbb{Z}^3}(p) \cap B$ (i.e., p is not 6-adjacent to any interior point).*

Note that the concept of the protected points of type 1 coincides with surface-endpoints of type **SE** (see Definition 3.2.3).

Definition 3.3.2 [221] *A black point p in picture $(\mathbb{Z}^3, 26, 6, B)$ is a* protected point of type 2 *if at least one of the three pairs of points in $N_6^{\mathbb{Z}^3}(p)$ (**U**,**D**), (**N**,**S**), and (**E**,**W**) (see Figure 1.5) is formed by two white points.*

Definition 3.3.3 [82] *A black point p in picture $(\mathbb{Z}^3, 26, 6, B)$ is* **not** *a* protected point of type 3 *if $\|N_{26}^{*\mathbb{Z}^3}(p) \cap B\| \geq 8$, or $4 \leq \|N_{26}^{*\mathbb{Z}^3}(p) \cap B\| \leq 7$ and $N_6^{*\mathbb{Z}^3}(p) \cap B$ contains three mutually 26-adjacent points. ($\|S\|$ stands for the counts of elements in set S).*

Definition 3.3.4 [26] *A black point p in a $(26, 6)$ picture is a* protected point of type 4 *if p is a non-simple border point (i.e., Condition 1 of Theorem 1.2.3 or Condition 3 of Theorem 1.2.3 is violated).*

Definitions 3.3.1-3.3.3 make it possible for us to specify endpoint-based surface-thinning algorithms, and Definition 3.3.4 is the basis of isthmus-based surface-thinning algorithms

The proposed sequential algorithms **S-6-SI-**i and the parallel algorithms **S-6-SI-**i are given by Algorithm 10 and Algorithm 11, respectively ($i = 1, 2, 3, 4$).

Algorithm 10: Sequential algorithm **S-6-SI-**i ($i = 1, 2, 3, 4$)

Input: set of black points B
Output: set of black points S
$S = B$
$I = \emptyset$
repeat
    foreach $d \in \{\mathbf{U}, \mathbf{D}, \mathbf{N}, \mathbf{E}, \mathbf{S}, \mathbf{W}\}$ **do**
        if $i = 4$ **then**
            $I = I \cup \{\ p \mid p \in S$ is a protected point of type $i\ \}$
        $C = I \cup \{\ p \mid p \in S$ is not a d-border point $\}$
        if $i < 4$ **then**
            $C = C \cup \{\ p \mid p \in S$ is a protected point of type $i\ \}$
        $X = S \setminus C$
        $del(d) = 0$
        foreach $p \in X$ **do**
            if point p is d-DELETABLE in S **then**
                $S = S \setminus \{p\}$
                $del(d) = del(d) + 1$
until $del(\mathbf{U}) + del(\mathbf{D}) + del(\mathbf{N}) + del(\mathbf{E}) + del(\mathbf{S}) + del(\mathbf{W}) = 0$;

We can state that the pair of sequential and parallel algorithms (**S-6-SI-**i, **P-6-SI-**i) never deletes any protected points of type i.

Algorithm 11: Parallel algorithm **P-6-SI-**i ($i = 1, 2, 3, 4$)

Input: set of black points B
Output: set of black points P
$P = B$
$I = \emptyset$
repeat
  foreach $d \in \{\mathbf{U}, \mathbf{D}, \mathbf{N}, \mathbf{E}, \mathbf{S}, \mathbf{W}\}$ **do**
    if $i = 4$ **then**
      $I = I \cup \{\ p \mid p \in P$ is a protected point of type $i\ \}$
    $C = I \cup \{\ p \mid p \in P$ is not a d-border point $\}$
    if $i < 4$ **then**
      $C = C \cup \{\ p \mid p \in P$ is a protected point of type $i\ \}$
    $X = P \setminus C$
    $del(d) = 0$
    $D_d = \{\ p \mid p \in X$ is a d-DELETABLE point in $P\ \}$
    $P = P \setminus D_d$
until $D_{\mathbf{U}} \cup D_{\mathbf{D}} \cup D_{\mathbf{N}} \cup D_{\mathbf{E}} \cup D_{\mathbf{S}} \cup D_{\mathbf{W}} = \emptyset$;

By comparing the sequential algorithm **S-6-SI-**i (see Algorithm 10) and the parallel algorithm **P-6-SI-**i (see Algorithm 11), we can state that in the parallel case the initial set of black points P is considered when the deletability of all the interesting points are investigated. On the contrary, the set of black points S is dynamically altered when a sequential reduction/subiteration is performed; the deletability of the actual point is evaluated in a modified picture (in which some previously visited interesting points are white).

The applied deletion rules that specify d-DELETABLE points ($d = \mathbf{U}$, $\mathbf{D}$, $\mathbf{N}$, $\mathbf{E}$, $\mathbf{S}$, $\mathbf{W}$) are given by $3 \times 3 \times 3$ matching templates depicted in Figure 3.9. Note that the six deletion rules were originally proposed by Gong and Bertrand [82] in their endpoint-based 6-subiteration surface-thinning algorithm with respect to surface-endpoints of type 3 (see Definition 3.3.3). An interesting black point ($p \in X$) is d-DELETABLE if template T_d matches it ($d = \mathbf{U}$, $\mathbf{D}$, $\mathbf{N}$, $\mathbf{E}$, $\mathbf{S}$, $\mathbf{W}$). Note that the templates assigned to the deletion direction d give the condition to delete certain d-border points, and templates associated with the last five deletion directions can be obtained by proper rotations of the template $T_{\mathbf{U}}$.

A period (i.e., the kernel of the **repeat** cycle in Algorithm 10 and Algorithm 11) is decomposed into six successive subiterations according to the six

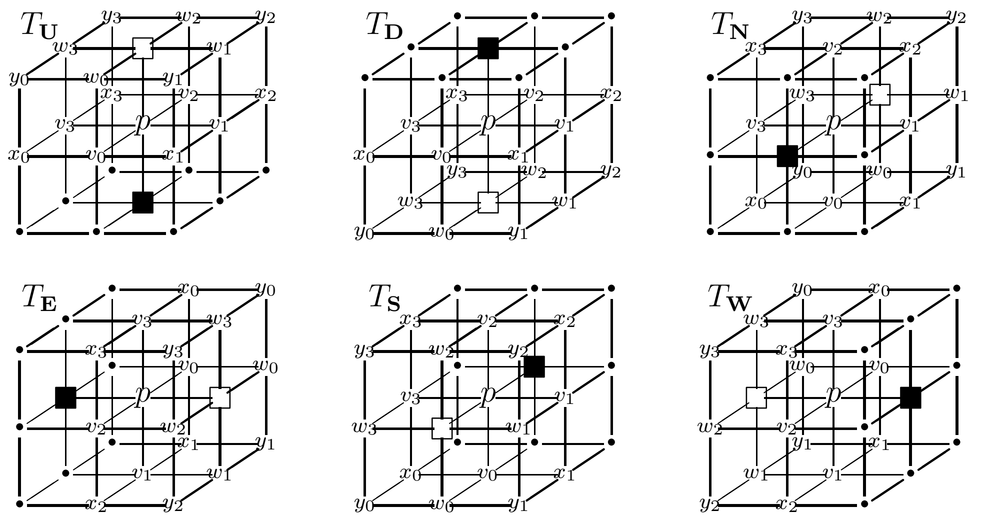


Figure 3.9: Matching template T_d associated with d-DELETABLE points ($d = \mathbf{U}$, $\mathbf{D}$, $\mathbf{N}$, $\mathbf{E}$, $\mathbf{S}$, $\mathbf{W}$). Notations: the central position marked p matches an interesting (black) point; the position marked '■' matches a (black) point in the constraint set; the position marked '□' matches a white point; if the position marked 'v_k' coincides with a white point, then the position marked 'w_k' coincides with a white point ($k = 0, 1, 2, 3$); if all the three positions marked 'v_k', '$x_{(k+1) \bmod 4}$', and '$v_{(k+1) \bmod 4}$' coincide with white points, then the position marked '$y_{(k+1) \bmod 4}$' coincides with a white point ($k = 0, 1, 2, 3$); each '·' (don't care) matches either a black or a white point.

main directions in 3D, and this period is repeated until stability is reached (i.e., no point is deleted within the last six reductions).

In [244], we proved that the deletion rule that deletes all d-DELETABLE points (with arbitrary constraint sets) is general simple ($d = \mathbf{U}$, $\mathbf{D}$, $\mathbf{N}$, $\mathbf{E}$, $\mathbf{S}$, $\mathbf{W}$). Hence, by Theorems 2.3.2 and 2.3.3, the pair of algorithms (**P-6-SI-**i,**S-6-SI-**i) are equivalent and topology-preserving ($i = 1, 2, 3, 4$).

It is important to emphasize that the parallel algorithm **P-6-SI-**3 coincides with the 6-subiteration 3D parallel surface-thinning algorithm proposed by Gong and Bertrand in 1990 [82]. We showed that algorithm **P-6-SI-**3 is equivalent to the sequential algorithm **S-6-SI-**3. In addition, the topological correctness of an existing parallel thinning algorithm is also confirmed. Note that (in 1990) Gong and Bertrand could not apply the very first sufficient conditions for topology-preserving 3D parallel reductions introduced by Ma (in 1994) [168].

In experiments the proposed four pairs of equivalent parallel and sequential surface-thinning algorithms were tested on objects of different shapes.

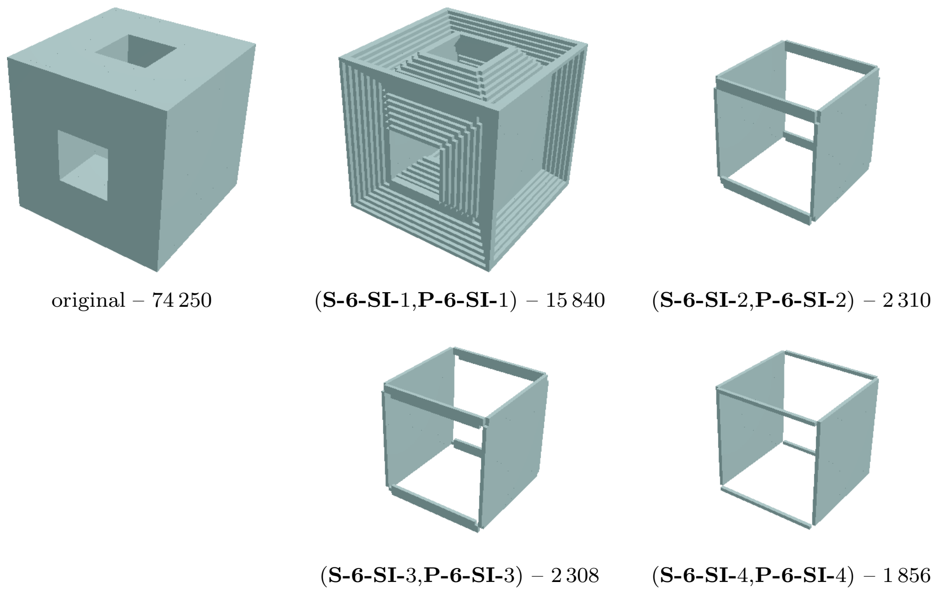


Figure 3.10: The original $45 \times 45 \times 45$ image of a cube with two tunnels and its medial surfaces produced by the proposed four pairs of equivalent surface-thinning algorithms.

Here we present two illustrative examples, see Figures 3.10 and 3.11.

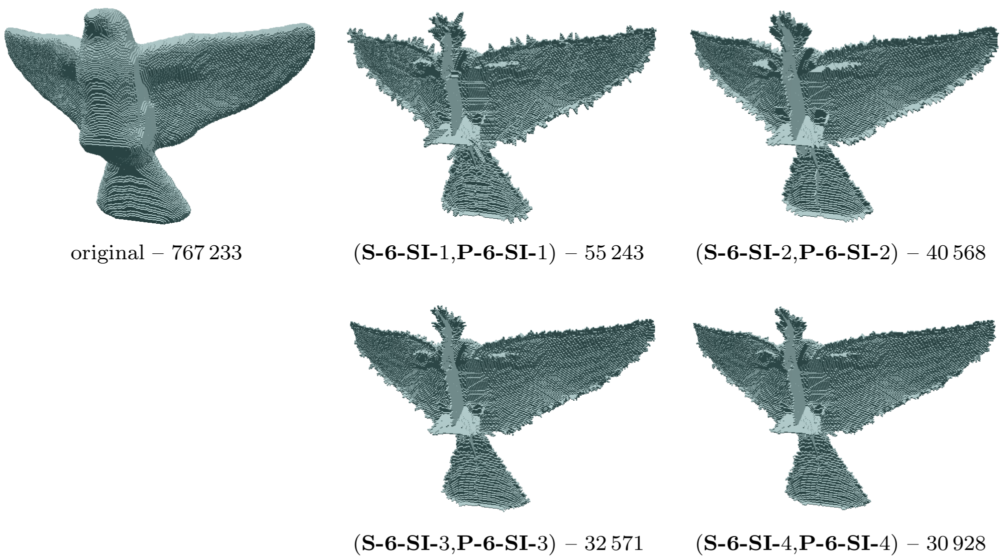


Figure 3.11: The original $321 \times 153 \times 227$ image of a bird and its medial surfaces produced by the proposed four pairs of equivalent surface-thinning algorithms.

3.4 Maximal 3D Curve-Thinning Algorithms

First, let us introduce the concept of a maximal 3D curve-thinning algorithm.

Definition 3.4.1 *A 3D curve-thinning algorithm is* maximal *if each point in the centerline produced by this algorithm contains only non-simple points and (simple) 3D-curve-endpoints (see Definition 3.2.2).*

Producing such centerline is crucial in many applications (e.g. raster-to-vector conversion [18, 253, 310] and skeletal graph construction [15, 16, 41, 62, 109, 167, 177, 231, 269, 273]).

Next, we present two maximal 3D curve-thinning algorithms acting on $(26, 6)$ pictures. Both of them fall into the category of sequential and subiteration-based. The first algorithm named **6SI-S-CT** [224] is endpoint-based, since it never deletes 3D-curve-endpoints. The second thinning algorithm called **6SI-S-CT-EPRC** [227, 231] may be viewed as the advanced version of the first one. It uses a novel endpoint-rechecking technique for reducing the number of unwanted side-branches. Thus it is a unique algorithm, since it is neither endpoint-based nor isthmus-based. Note that the author constructed two further maximal curve-thinning algorithms based on isthmuses [240] and anchored shrinking [252].

3.4.1 An Endpoint-Based Sequential Curve-Thinning Algorithm

The algorithm **6SI-S-CT** [224] is outlined by Algorithm 12.

One iteration step of the sequential object reduction process (i.e., the kernel of the **repeat** cycle) is decomposed into six successive sub-iterations according to the six main directions in 3D. In this way, the objects are shrunk uniformly in each direction. Each subiteration consists of two phases; first the border points of the actual type that are simple and not 3D-curve-endpoints are marked as potential deletable points. This marking phase can be performed in parallel, but the forthcoming deletion phase must be sequential. The algorithm terminates if stability is reached.

It is obvious that algorithm **6SI-S-CT** deletes all simple points that are not 3D-curve-endpoints. Thus the produced centerline may contain only non-simple points and (simple) 3D-curve-endpoints. Therefore, algorithm **6SI-S-CT** is maximal.

Algorithm **6SI-S-CT** can be implemented according to the general (and computationally efficient) scheme given by Algorithm 3 and Algorithm 4. Two linked lists are used; the first one stores all border points and the second

Algorithm 12: Endpoint-based 6-subiteration sequential curve-thinning

```
repeat
    // one iteration step
    foreach d ∈ { U, D, N, E, S, W } do
        // subiteration associated with deletion direction d
        mark all simple d-border points that are not
        3D-curve-endpoints
        // deletion of some d-border points
        foreach marked point p do
            if p is simple and not a 3D-curve-endpoint in the actual
            image then
                delete p
until no points are deleted;
```

one contains the marked points of the actual subiteration. The deletability of a point is decided by determining an integer code corresponding to the $3 \times 3 \times 3$ neighborhood of the point in question and addressing a single pre-calculated (unit time access) 8MB look-up-table (LUT) containing the answers for all possible $3 \times 3 \times 3$ configurations [230].

Algorithm **6SI-S-CT** has been successfully applied in several medical image processing problems, including assessment of infrarenal aortic aneurysms [224], assessment of tracheal stenosis [224, 311, 313, 315, 316], unraveling and virtual dissection of the colon [224, 312, 314, 317], and colorectal polyp detection [314].

3.4.2 Curve-Thinning with Endpoint-Rechecking

Next, the advanced version of the endpoint-based 6-subiteration sequential curve-thinning algorithm **6SI-S-CT** is presented. It uses *endpoint-rechecking* that compares the situation at some stage of the thinning process with the previous object configuration. The modified algorithm **6SI-S-CT-EPRC** [227, 231] is outlined by Algorithm 13.

During a (sequential) subiteration, a marked point is deleted if it remains simple and is not a 3D-curve-endpoint after the deletion of some previously visited marked points. In addition, in some special cases, some points are also deleted if they have become 3D-curve-endpoints (see lines 13-15 in Algorithm 13). The algorithm uses an extra parameter $t \in \{0, 1, 2, 3, 4, 5, 6\}$ and

Algorithm 13: 6-subiteration sequential curve-thinning with endpoint-rechecking

```
repeat
    // one iteration step
    foreach i ∈ { U, D, N, E, S, W } do
        // subiteration associated with deletion direction d
        mark all simple d-border points that are not
        3D-curve-endpoints
        // deletion of some d-border points
        foreach marked point p do
            if p is simple in the actual image then
                if p is not a 3D-curve-endpoint then
                    // 'conventional' deletion
                    delete p
                else if #( deleted 6-neighbors of p) ≥ t then
                    // deletion of a 3D-curve-endpoint
                    delete p
until no points are deleted;
```

a marked (simple and 3D-curve-endpoint) point can be deleted if at least t points of its 6-neighbors have been deleted during the actual subiteration. This additional condition identifies the configurations that are likely to produce 'spurious' side branches. Note that if $t = 6$, the endpoint-rechecking has no effect (since a point is not simple if all of its 6-neighbors are black). In that case, algorithm **6SI-S-CT-EPRC** produces the same result as algorithm **6SI-S-CT**. According to our experience, setting $t = 1$ or $t = 2$ is suggested for human airway trees [231]. See Figure 3.12 for an example of the usefulness of the endpoint-rechecking.

Since the centerline produced by algorithm **6SI-S-CT-EPRC** contains less points than the maximal thinning algorithm **6SI-S-CT**, curve-thinning algorithm **6SI-S-CT-EPRC** is also maximal.

Algorithm **6SI-S-CT-EPRC** has been applied in some biomedical image processing applications, including characterization of the interstitial lung diseases [99], matching and anatomical labeling of human airway tree [339, 341], quantitative analysis of pulmonary airway trees [227, 231], liver segmentation for surgical resection planning [20], and identifying synaptic connections [187].

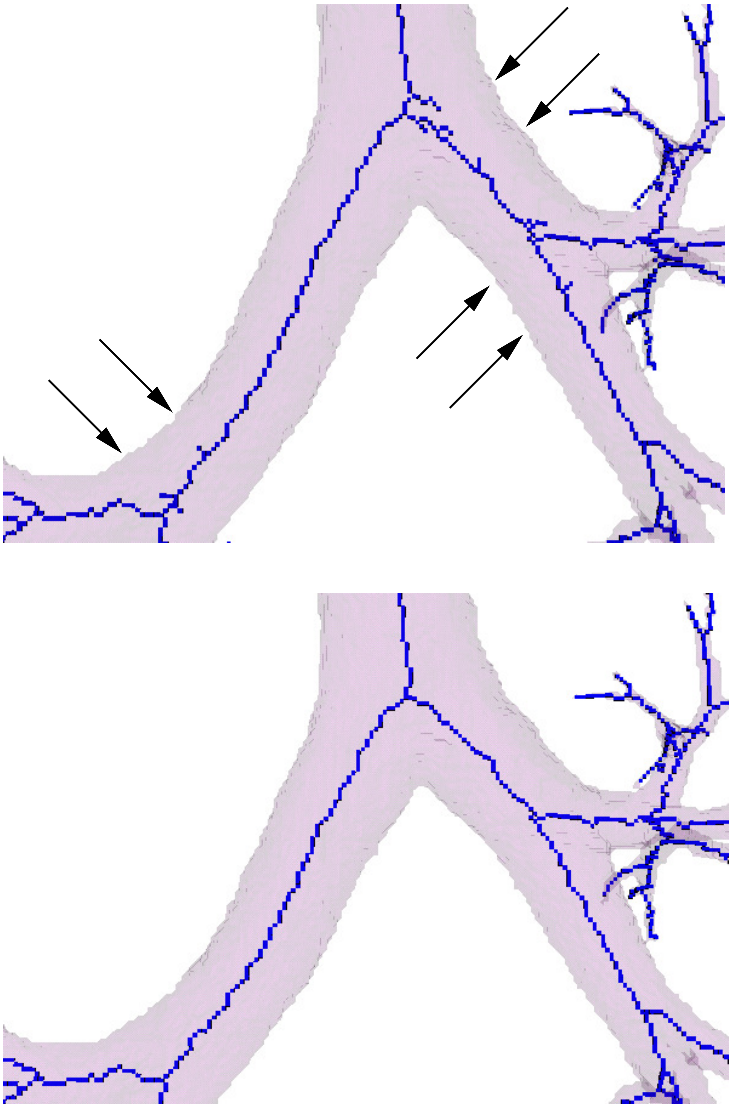


Figure 3.12: A part of a segmented human airway tree and its centerline produced by algorithm **6SI-S-CT** or algorithm **6SI-S-CT-EPRC** with $t = 6$ (top). The centerline made by algorithm **6SI-S-CT-EPRC** with $t = 1$ (down). The centerline produced by endpoint-rechecking contains only 125 (true) branch-points (junctions) and 128 3D-curve-endpoints. There are 167 branch-points and 176 3D-curve-endpoints in the centerline generated without endpoint-rechecking. Several of the unwanted branches are marked by arrows.

Chapter 4

Quantitative Analysis of Pulmonary Airway Trees

The author's 3D curve-thinning algorithms have been applied in several biomedical applications, including assessment of infrarenal aortic aneurysm [224], assessment of tracheal stenosis [224, 311, 313, 315, 316], unraveling and virtual dissection of the colon [224, 312, 314, 317], colorectal polyp detection [314], characterization of the interstitial lung diseases [99], matching and anatomical labeling of human airway tree [339, 341], quantitative analysis of pulmonary airway trees [227, 231], liver segmentation for surgical resection planning [20], and identifying synaptic connections [187].

For reasons of scope, in this chapter, only our method for quantitative analysis of pulmonary airway trees is presented[6].

Tubular structures are frequently found in living organisms. The tubes – e.g., arteries or veins are organized into more complex structures. Trees consisting of tubular segments form the arterial and venous systems, intrathoracic airways form bronchial trees, and other examples can also be found. Computed tomography (CT) or magnetic resonance (MR) imaging provides volumetric image data allowing identification of such tree structures. Frequently, the trees represented as contiguous sets of points must be quantitatively analyzed. The analysis may be substantially simplified if the point-level tree is represented in a formal tree structure consisting of a set of nodes and connecting arcs. To build such formal trees, the point-level tree object must be transformed into a set of interconnected single-point centerline representing individual tree branches. Therefore, the aim of our work was to develop a robust method for identification of centerline and bifurcation (trifurcation, etc.) points in segmented tubular tree structures acquired

[6]Thanks to the invitation by Milan Sonka, the author could work on this topic during his visits at the Departments of Electrical and Computer Engineering, The University of Iowa, IA, USA.

in vivo from humans and animals using volumetric CT or MR scanning, rotational angiography, or other volumetric imaging means.

There are many reasons why identifying tree centerlines is important. Centerlines can serve as one-dimensional structures allowing guidance for orderly exploration of the entire tree, they can serve as viewpoint trajectory for navigation purposes in virtual bronchoscopy or angioscopy. To facilitate quantitative analysis of the vascular or bronchial tree, e.g., luminal area or wall thickness, measurements must be obtained in cross-sections perpendicular to the long axis of the tree segments. Clearly, planes normal to the tree centerlines must be identified and centerline correctness is of paramount importance. As such, quantitative assessment of asthma or cystic fibrosis from pulmonary CT images depend on the performance of the centerline extraction method. Similarly, accuracy and reproducibility of arterial plaque thickness measurements from coronary CT or intravascular ultrasound depends on the ability to producing centerlines of tubular structures.

In this chapter, our method for quantitative analysis of pulmonary airway trees is presented. The developed method was validated in 343 computer phantom instances subjected to changes of its orientation, in a rigid plastic phantom CT-scanned under 9 orientations, in a rubber plastic phantom CT-scanned under 9 orientations, and in 54 in vivo scans of human lungs. The validation studies demonstrated sub-voxel accuracy of branch point positioning, insensitivity to changes of object orientation, and high reproducibility of derived quantitative indices of the tubular structures offering a significant improvement over previously reported methods [99, 227, 231, 341].

Assuming that an imperfectly segmented tree was obtained from volumetric data in the previous stages, our method allows us to obtain a single-point centerline of the tree while overcoming many segmentation imperfections, yields formal tree representation, and performs quantitative analysis of individual tree segments on a tree-branch basis. The input of our method is a 3D binary image representing a segmented point-level tree object. All main components of our method were specifically developed to deal with imaging artifacts typically present in volumetric medical image data. As such, the method consists of the following main steps: airway segmentation, correction of the segmented tree, identification of the tree root, producing centerline, pruning, smoothing, identification of branch-points, generating a formal tree structure, tree partitioning, calculating associated measures, tree matching, and re-sampling 2D slices perpendicular to airway segments.

The rest of this chapter describes the main steps in more detail.

4.1 Airway Segmentation

The employed airway segmentation method is based on fuzzy connectivity [97, 343]. After automatically identifying a seed point in the upper trachea, two regions – foreground and background – are grown simultaneously and are competing against each other during the segmentation process. To minimize occurrence of leaks into the surrounding lung parenchyma during the region-growing process, the root of a leak is identified once the leak is observed and the segmentation parameters are modified. To accomplish such behavior, a relatively small adaptive region of interest (ROI) is defined in an iterative fashion. The ROI follows the airway tree branches as they are segmented. The ROI has a cylindrical shape and adapts its geometrical dimensions, its orientation, and position to the predicted size, orientation, and position of the airway branch to be segmented, see Figure 4.1a. Using a cylindrically shaped ROI (versus the more common cubical ROI used in other 3D image segmentation tasks) has the advantage of the better adaptation of the ROI to the target shape, which is close to cylindrical. This means less 'useless' background points have to be analyzed and the computing time can be shortened. (Note that a similar approach was independently used by Kitasaka et al. [130].) After the series of iteration steps, an airway tree is segmented, see Figure 4.1b.

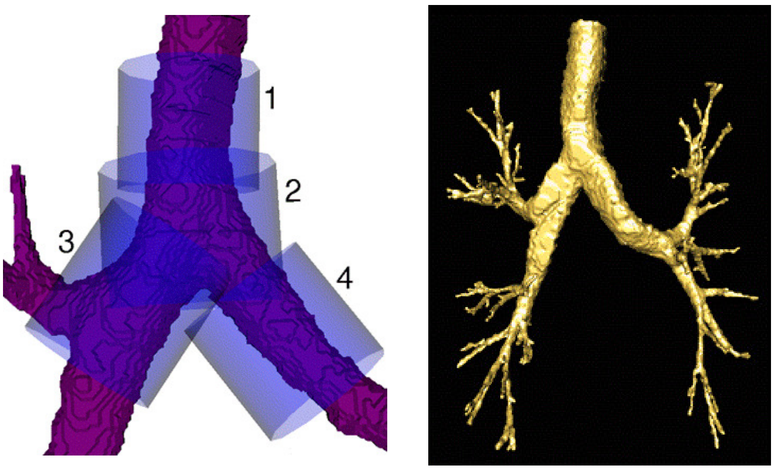


Figure 4.1: Adaptive cylindrical regions of interest (ROIs) follow airway tree branches as the segmentation proceed (left). Segmentation result using the proposed method (right).

The segmentation algorithm was shown to be robust in low dose and regular dose scans from normal and diseased subjects when compared to previous airway segmentation approaches [340].

4.2 Correction of the Segmented Tree

When applied to clinical volumetric images, segmentation algorithms may produce imperfect results in which the segmented objects contain internal cavities (i.e., 6-connected set of white points surrounded by 26-connected set of black points), holes (i.e., white points forming tunnels), and bays (i.e., disturbances without a topological change). Some of such imperfections cause unwanted changes of the underlying topology, and all of them disturb the centerline detection process and consequently yield an incorrect centerline and thus incorrect formal tree representation, see Figure 4.2. To overcome the effects of artifactual cavities, the white points connected to the frame of the volume are first labeled by sequential forward and backward scanning (instead of the conventional object labeling). Then, all unlabeled white points are changed to black points. The applied method is similar to the linear-time Chamfer distance mapping [35]. As a result, all cavities are filled with no connectivity alteration.

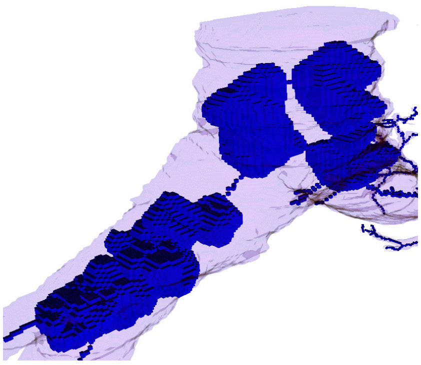


Figure 4.2: Part of a segmented human airway tree containing some cavities. Its topological kernel shown superimposed. Each cavity of the structure corresponds to a cavity in the topological kernel thus appearing as an enlarged 'bubble' with unit wide hull. Therefore, topologically correct centerline of any structure with a false cavity yields an incorrect formal representation since it cannot deal with cavity elimination.

Holes and bays are removed by applying morphological closing [83] (i.e., a dilation followed by an erosion with an experimentally determined structuring element). Note that the closing is a double-edged sword; it is suitable for filling small gaps, holes, and cavities, but new holes may be created in the process. This side effect can be handled by a post-processing pruning process. In this application, however, the object of interest is a single tubular structure and the potential negative effect of the closing operation is maximal.

4.3 Root Detection

Our work deals with analysis of intrathoracic airway trees from volumetric CT (or MR) image data. Consequently, a priori knowledge of the data set is used to identify the tree root. In other applications, a different root identification approach may be needed. The root detection is not a critical phase of the process. It can be identified interactively or automatically [196, 349].

In pulmonary CT images, the center of the topmost nonzero 2D slice in direction z (detected by 2D reductive shrinking [90]) defines the root of the formal tree to be generated and belongs to the trachea, see Figure 4.3. The detected root point acts as an anchor point during the centerline extraction (i.e., it cannot be deleted by the forthcoming iterative peeling process).

Figure 4.3: The detected tree root (as an anchor point) is shown in green.

4.4 Producing Centerline

The well-known approach to represent a 3D tubular structure is to construct its centerline by a skeletonization algorithm. However, some of the properties of these algorithms are undesirable. Specifically, we do not need the presence of surface patches (i.e., branched 2D manifolds [202]) nor 'thick' curves (i.e., line segments that are wider than one point). As a solution, we applied the maximal 3D curve-thinning algorithm **6SI-S-CT-EPRC**, see Algorithm 13 in Section 3.4.2. The produced centerline of a segmented human airway tree is illustrated in Figure 4.4.

Note that the selected 6-subiteration sequential curve-thinning algorithm with endpoint-rechecking was modified, since the detected root of the tree is to be preserved during the iterative process (i.e., we applied anchor-preserving thinning [326, 358]).

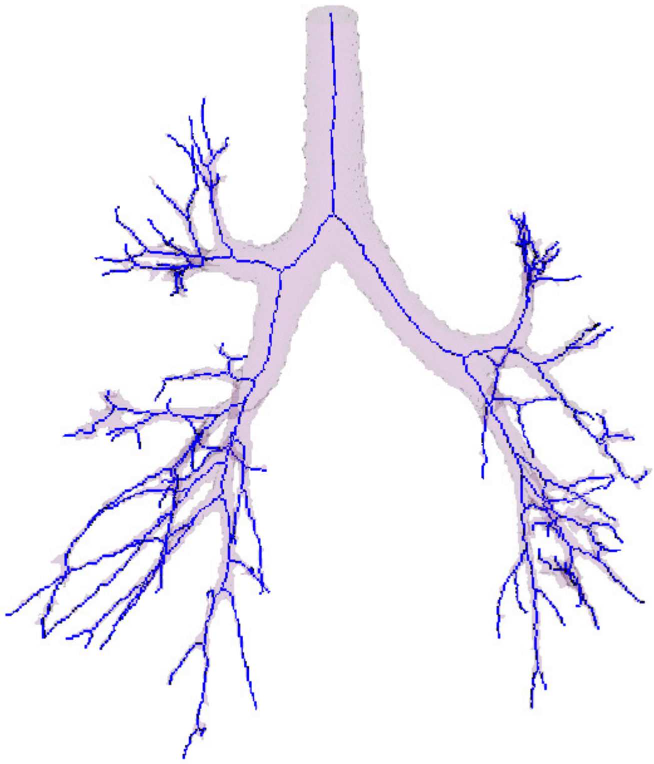


Figure 4.4: The produced centerline superimposed on the segmented tree.

4.5 Pruning

Unfortunately, each skeletonization algorithm (including ours) is sensitive to coarse object boundaries. As a result, the produced (approximation to the) 'skeleton' generally includes false segments that must be removed by a pruning step [14, 87, 110, 144, 160, 192, 294, 298, 299, 323, 328]. The simplest pruning method is the *morphological pruning* [64, 83]. It removes all side branches from the raw centerline that are shorter than a predefined threshold. This method necessarily fails in structures consisting of tubular segments of varying thickness.

Applying a proper pruning method that would yield reliable centerline is critical in all tree-representation applications. An unwanted branch causes false generation numbering and consequently false measurements corresponding to the individual segments of the tree (including length, volume, surface area, etc.). Therefore, we have developed a method capable of removing 'long' parasitic branches from 'thick' parts and preserving 'short' correctly determined branches in 'thin' segments. Spurious branches are identified by assessing the distance-from-surface function and the branch length. Our pruning process consists of the following two phases:

- cutting holes that remain after the morphological closing, and
- deleting side branches using both the length and depth information.

At first, the centerline is converted into a graph structure (each point corresponds to a graph node/vertex and there is an edge between two nodes if the corresponding points are 26-adjacent). Then, Dijkstra's algorithm is applied to solve the single-source shortest-paths problem [52] that maintains a rooted tree from a source node (i.e., the root detected in the first nonzero 2D slice in direction z). Since the result of Dijkstra's algorithm is always a (cycle-free) tree, we can detect and cut holes (loops) in the centerline easily: a point is to be deleted if it is not an endpoint (i.e., 3D-curve-endpoint, see Definition 3.2.2) and is not the parent of any other point in the Dijkstra's tree. This heuristic hole-cutting approach works well, although counter-examples can be given in which the heuristic approach does not apply, see Figure 4.5.

After the hole cutting, the parasitic side branches shall be removed. We have developed a pruning method for centerline pruning that uses both the branch length and the distance-from-surface (depth) information for the identification of a pruning candidate, see Figure 4.6. We delete all branches if their lengths are shorter than a given threshold t_l *and* their branch-points are not closer to the border/surface of the spacious tree (after topological correction) than a given threshold t_d. Note, that a similar, but not identical

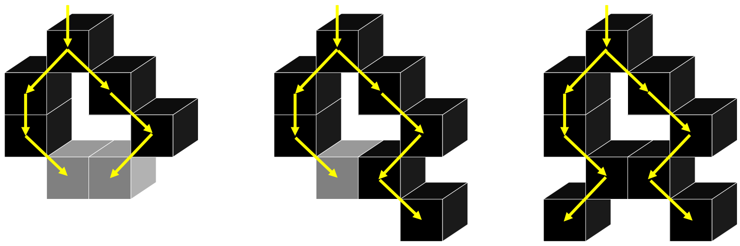


Figure 4.5: Examples of the hole-cutting method. Dark points are preserved, bright ones are deleted. Arrows correspond to the edges in the Dijkstra's tree. Holes are successfully eliminated in most cases (left, middle), but counter-examples can be found (right).

pruning rule has been introduced by Mori et al. [196]. They delete a branch if its length is shorter than a given threshold t_l *or* it starts from a branch whose diameter is larger than a given threshold t_d. Our method with repeated pruning steps (applying different pairs of thresholds (t_l, t_d)) is more general and can also delete unwanted subtrees as it is shown in Figure 4.7.

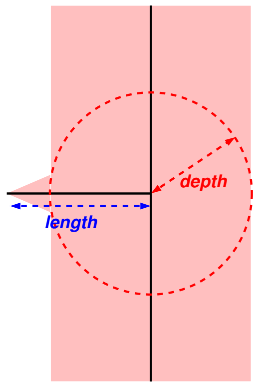


Figure 4.6: Pruning centerline based on the branch length and the distance-from-surface (depth) information. A side branch is to be deleted if it is shorter than t_l (length) and its branch-point is not closer to the border/surface than t_d (depth).

The following algorithm is applied:

1. Calculate the linear time (3,4,5)–Chamfer distance map [35] for the spacious tree (after topological correction) in which the feature points are formed by the white points in the picture. The resulting 'distance-from-surface' map DSM is a non-binary array containing the distance to the closest feature point.

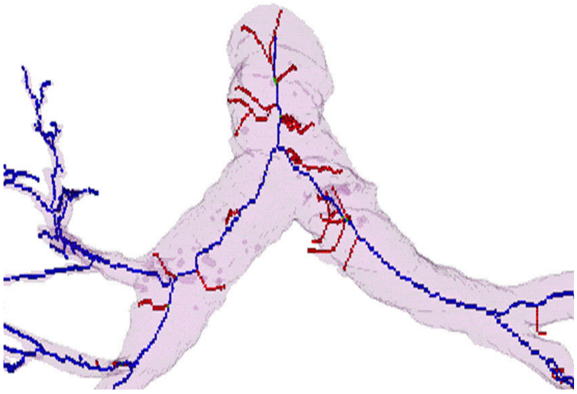


Figure 4.7: Example of removing some unwanted subtrees. (Deleted points are depicted in red.)

2. Initialization of the 'skeletal distance map' SD:

$$SDM(v) = \begin{cases} 0 & \text{if } v \text{ is a branch-point and } DSM(v) \geq t_d \\ \mathrm{B} & \text{if } v \text{ is a branch-point and } DSM(v) < t_d \\ \mathrm{B} & \text{if } v \text{ is the root of the tree} \\ \mathrm{m} & \text{otherwise} \end{cases},$$

   where values 'B' and 'm' should be larger than the maximal length in the tree. The 'B' points are 'bumpers' during the forthcoming distance-propagation step while the 'm' points (assigned to line-points and end-points in the centerline) are to be changed.

3. Distance propagation in SDM according to the (3,4,5)–Chamfer distance can be performed similarly to the linear time Chamfer distance mapping. Note that the 'B' points remain unchanged during this step.

4. Branch deletion: a side branch with an associated endpoint v is deleted if $SDM(v) \leq t_l$. It can be done easily by using the Dijkstra's tree or in the following way:

   for $i = t_l$ **downto** 1 **do**
     for each endpoint v in the centerline **do**
       if $SDM(v) = i$ **then** delete v from the centerline

Steps 2–4 of the above process can be repeated k–times for different pairs of thresholds: $(t_{l_1}, t_{d_1}), (t_{l_2}, t_{d_2}), \ldots, (t_{l_k}, t_{d_k})$,. In our experience, 2 to 4 iterations typically provide satisfactory results for in vivo airway trees. Note, that the thresholds are to be experimentally determined.

4.6 Smoothing

The pruned centerline may be rough, therefore, a smoothing step is applied. It alters some one-, two-, and three-point long sequences of line-points. The smoothing rules are given by a set of matching templates. For brevity, only the templates capable of altering one point long segments are presented in Figure 4.8. The smoothing process is illustrated in Figure 4.9.

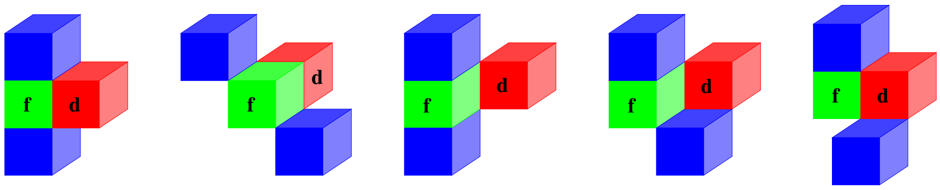


Figure 4.8: The five base matching templates of our smoothing process capable of altering one point long parts. If the two blue points and the red one are line-points in the centerline, then the red point is deleted (i.e., changed to white) and the white point shown in green is filled (i.e., changed to black). Note, that all rotations of these base matching templates (where the rotation angles are 90°, 180°, and 270°) are also matching templates.

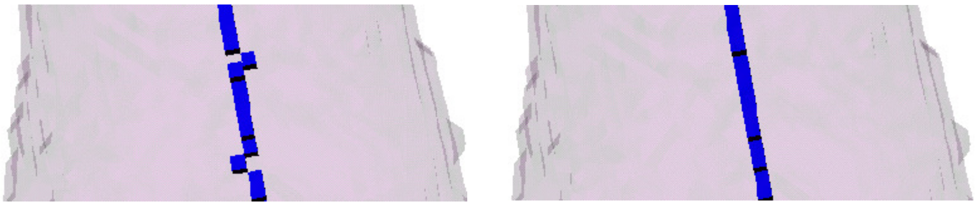


Figure 4.9: Example of the smoothing process: the same part of the centerline before smoothing (left) and after smoothing (right).

4.7 Branch-point Identification

There are three types of points in a centerline produced by a maximal 3D curve-thinning algorithm: endpoints (which have only one 26-neighbor), line-points (which have exactly two 26-neighbors), and branch-points (which have more than two 26-neighbors) that form junctions (bifurcations, trifurcations, etc.), see Figure 4.10. Clearly, branch-identification in a maximally thinned centerline is trivial. One problem is that more than one branch-point may form a junction. In that case, the branch-point closest to the root of the tree is assigned the junction label. The branch-point identification is illustrated in Figure 4.11.

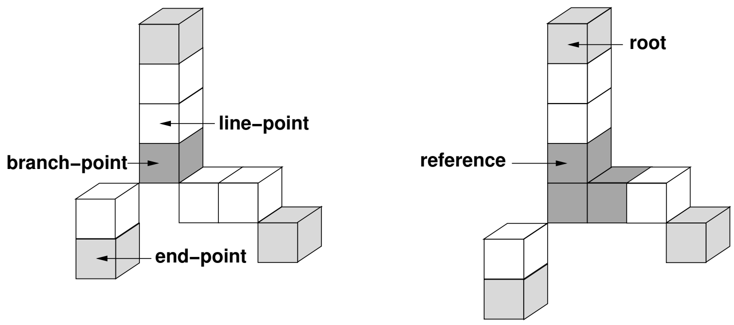


Figure 4.10: The three types of points in a centerline (left). Sometimes, a junction is formed by more than one branch-point (right). In that case, the branch-point closest to the root is the reference point of the junction.

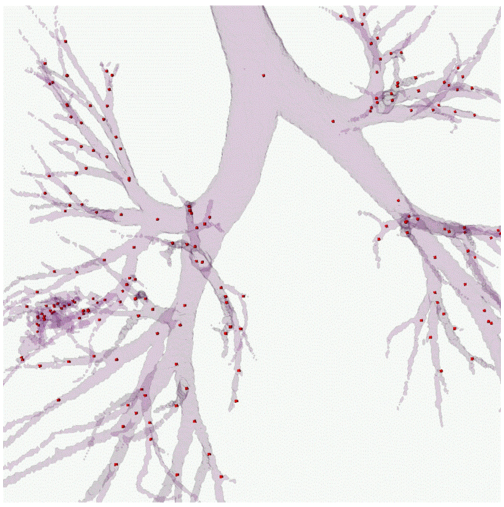


Figure 4.11: Identified branch-points in a human airway tree.

4.8 Generating Formal Tree Structure

The formal tree structure (i.e., skeletal graph) assigned to the pruned centerline is based on the updated Dijkstra's tree (after pruning). It is stored in an array of n elements for a centerline containing n points. Each element of that array stores the coordinates of a point, its depth in the volume, and the index of the element that corresponds to the parent/predecessor point in the tree. This internal data structure is suitable for the forthcoming measurements, and provides an efficient coding of the resulting binary image. A similar structure is assigned to the branch-points. In the formal tree, a path between two branch-points is replaced by a single edge, see Figure 4.12. Note, that similar methods were already presented by several authors [77, 195, 196, 349]. The skeletal graph is stored in an XML file, which provides associated measures for each part of the partitioned tree.

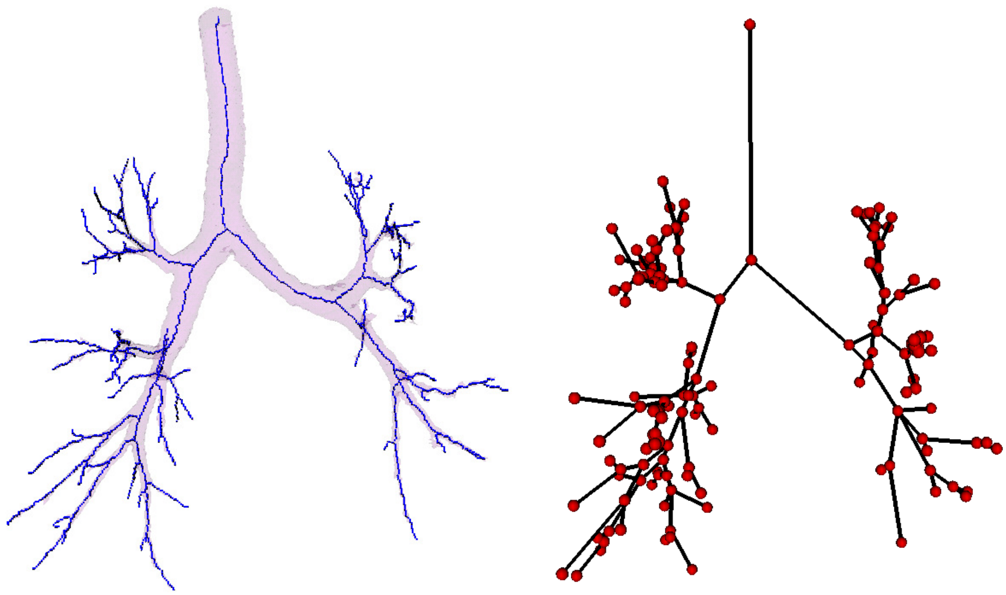


Figure 4.12: A human airway tree and its centerline (left). The corresponding formal tree structure, in which a path between two branch-points is replaced by a single edge (right).

4.9 Tree Partitioning

The aim of the partitioning procedure is to partition all points of the segmented tree into branches (i.e., a branch-specific label is assigned to each). There are two inputs into the process – the segmented tree after topological corrections, and the formal tree structure corresponding to the centerline. The output is a gray-level image, in which value '0' corresponds to the background and different non-zero values are assigned to the points belonging to different tree branches/partitions.

The automated partitioning consists of two steps, see Figure 4.13. First, only the points in the centerline are partitioned so that each branch/partition of the centerline has a unique label. Points of the segmented tree that are not in the centerline are then partitioned by label propagation – each point in the tree gets the label of the closest point in the centerline.

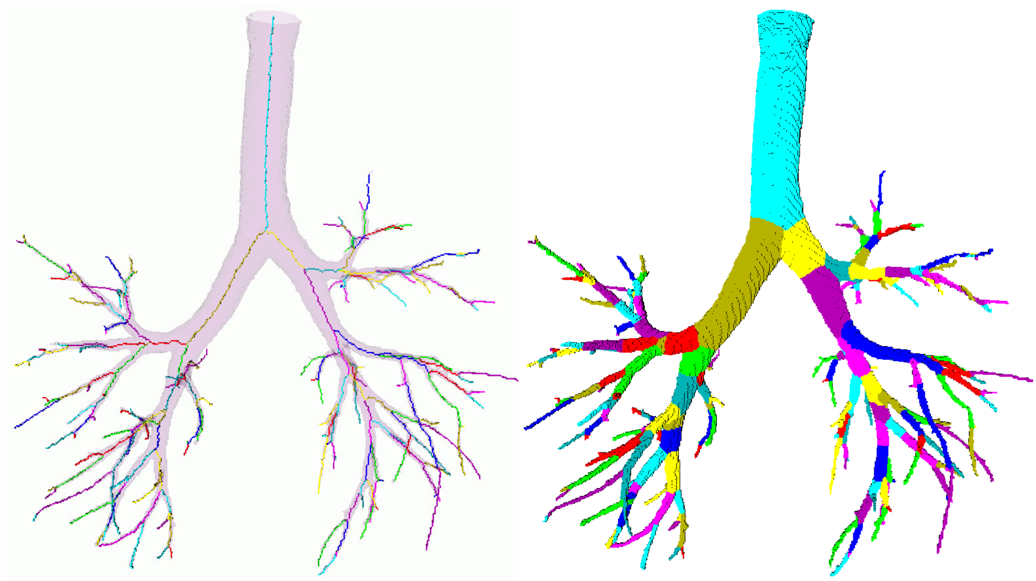


Figure 4.13: Partitioning process: the segmented volume and the partitioned centerline (left) and the partitioned volume after label-propagation (right). (Note, that we used only 9 colors in displaying these trees, therefore, the same color was assigned to multiple branches.)

The first step (i.e., centerline partitioning) uses a queue Q (first-in-first-out data structure) and assigns label $l(v)$ and generation number $g(v)$ to the point in the centerline v. That process is outlined by Algorithm 14.

At first, the starting label is set and queue Q is initialized to contain the root of the skeletal tree. Each time through the outer **while** loop, a vertex v (i.e., the root or a branch-point) is extracted from the queue and labeled

Algorithm 14: Centerline partitioning

```
label ← starting_label
g(tree_root) ← 1; ENQUEUE(Q, tree_root);
while NONEMPTY(Q) do
    v ← DEQUEUE(Q); l(v) ← label; generation ← g(v);
    while v has only one child do
        u ← ONLYCHILD(v); l(u) ← label;
        g(u) ← generation; v ← u;
    label ← NEXT(label); generation ← generation + 1;
    for each child u of v do
        g(u) ← generation; ENQUEUE(Q,u);
```

by the current label. Then all vertices (i.e., line-points) along the path from point v to the next branch-point are labeled by the same value (see the inner **while** loop). After labeling an entire branch, the current label is modified and the starting points of all adjacent branches are placed at the tail of the queue (see the **for** loop).

The algorithm – as implemented – is more difficult than as was presented. To deal with trifurcations, it is also capable to merge two branch-points if the distance between them is less than a given threshold. Trifurcations routinely appear in human airway trees, represented by a sequence of two bifurcations in a close proximity along the identified centerline, see Figure 4.14.

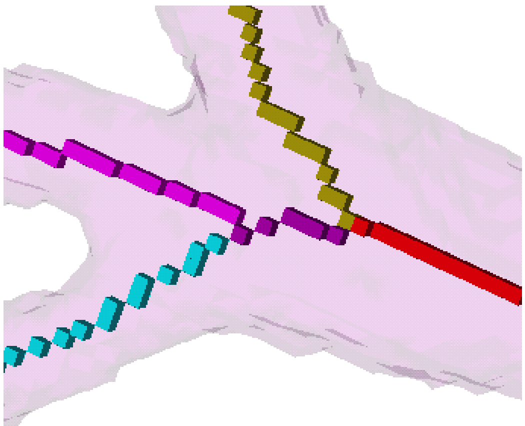


Figure 4.14: Two close branch-points represent a trifurcation. They are merged in our formal tree.

4.10 Calculating Associated Measures

For each partition/branch of the tree, the following measures/indices are calculated:

- branch length – defined as a Euclidean distance between the parent and child branch-points (in mm),
- branch volume – defined as a volume of all voxels belonging to the branch (in mm^3),
- branch surface area – defined as a surface area of all boundary voxels belonging to the branch (in mm^2),
- branch radius – derived from the branch length and the branch volume assuming 'cylindrical' partition (in mm):

$$radius = \sqrt{\frac{volume}{\pi \cdot length}}.$$

Determining the first three indices is fairly straightforward, but calculating a reliable approximation to the branch radius is rather complicated. The two ends of a partition/branch are 'conic', therefore, they must be suppressed to get measurements only from the 'cylindrical' partitions.

First, points in the centerline are re-labeled as follows:

$$l(v) \begin{cases} l(v) + max\_label & \text{if voxel } v \text{ belongs to a `branch-point area'} \\ l(v) & \text{otherwise} \end{cases},$$

where $l(v)$ is the original label of point v and max_label is greater than the largest label assigned during the centerline partitioning process (therefore, $l(v) + max\_label$ is a brand new label). 'Branch-point areas' (i.e., the set of points to be re-labeled) are determined by the values in the distance map corresponding to the branch-points.

During the label propagation step (i.e., when each point in the tree gets the label of the closest centerline point), the new labels are propagated. Determining branch volume and branch surface is based on labels $l(v)\ mod\ max\_label$. Calculating the branch radius is based on 'cylindrical' branch volume and 'cylindrical' branch surface. Those indices are determined by labels that are less than max_label. (Note that branch radius is derived from the 'cylindrical' branch length that is shorter than the calculated branch length.) The method is illustrated in Figures 4.15 and 4.16.

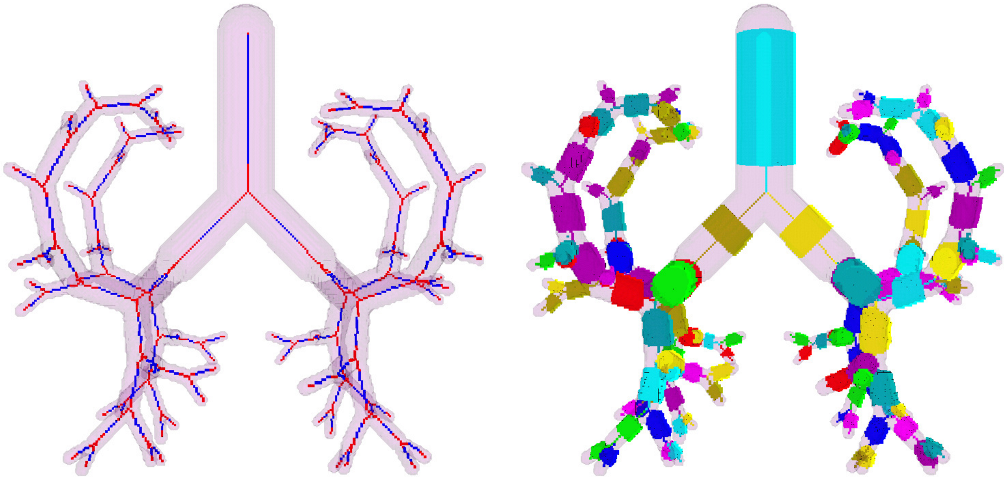


Figure 4.15: Calculating branch radius in a computer phantom. Centerline points belonging to the excluded branch- and end-areas are in red (left), 'cylindrical' partitions for estimating branch radius (right). (Note, that the partitioned trees are displayed by using only 9 colors, therefore, the same color may have been assigned to some adjacent partitions.)

Figure 4.16: 'Cylindrical' partitions in a human airway tree.

Lastly, we can state that the described automated method for pruned and smoothed centerline extraction, branch-point identification and quantitative analysis of tubular tree structures is robust, efficient, and highly reproducible. It facilitates calculation of a number of morphologic indices described above as well as indices not considered in this work – branch angle, curvature, and many others.

4.11 Tree Matching

The goal of matching human intrathoracic airway trees is to find anatomically corresponding branch-points in two different trees, see Figure 4.17. Two types of matching are of utmost interest: intra-subject and inter-subject matching. In the first case, trees coming from different scans of the same subject are matched. In the second case, two or more trees are matched originating from different subjects. The latter case only allows matching of the primary branch-points (the first three or four generations). These primary branch-points are frequently (although not universally) identical among humans. The branching pattern of higher airway generations varies from subject to subject, much like fingerprints do.

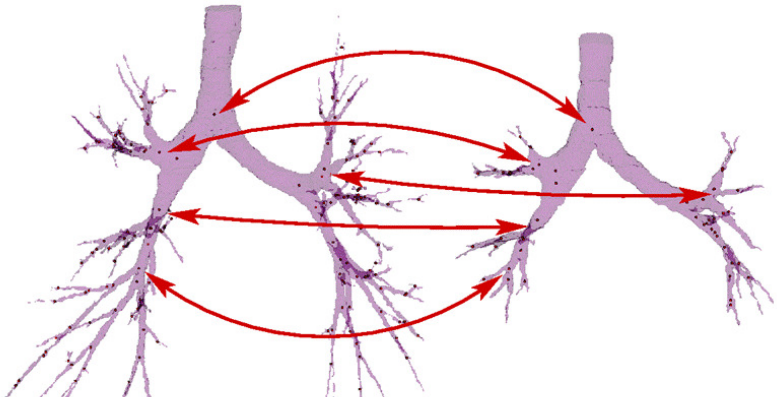


Figure 4.17: Finding correspondences among branch-points in two airway trees.

Our matching algorithm [339, 341] performs the following three main steps:

- Aligns the two input trees by performing a rigid registration.
- Finds and match major branch-points.
- Matches subtrees underneath major branch-points, one pair of subtrees at a time.

The proposed branch-point matching allowed us to match pairs of volumetric high resolution in vivo CT scans. Each subject was scanned twice, one scan at functional residual capacity (FRC, 55% lung volume), and one scan at total lung capacity (TLC, 85% lung volume). After matching a pair of

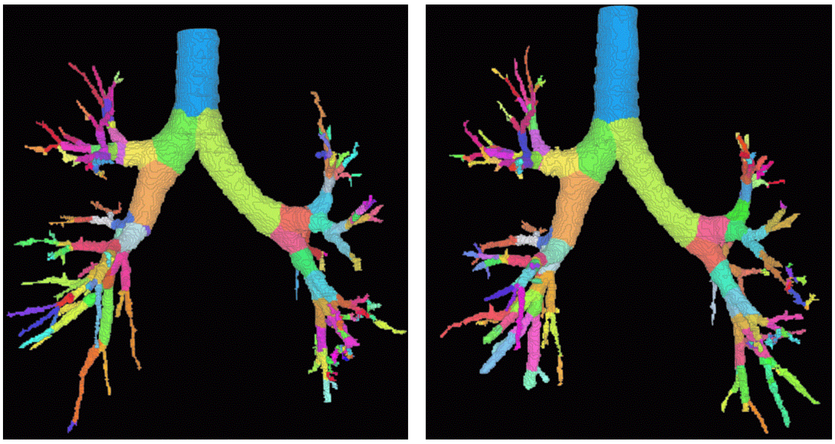


Figure 4.18: Matched partitioned trees: FRC (left) and TLC (right).

FRC/TLC tree (see Figure 4.18) we could analyze the changes of the quantitative indices (i.e., length, volume, surface area, and radius) of the paired individual branches.

Assigning anatomical names to the segments and branch-points of the human airway tree, is of significant interest for clinical applications and physiological studies. Our anatomical labeling algorithm aims to assign 32 anatomical names to their respective segments (the left inferior bronchus was divided into two parts, which accounts for the one additional segmental name compared to the standard nomenclature), see Figure 4.19.

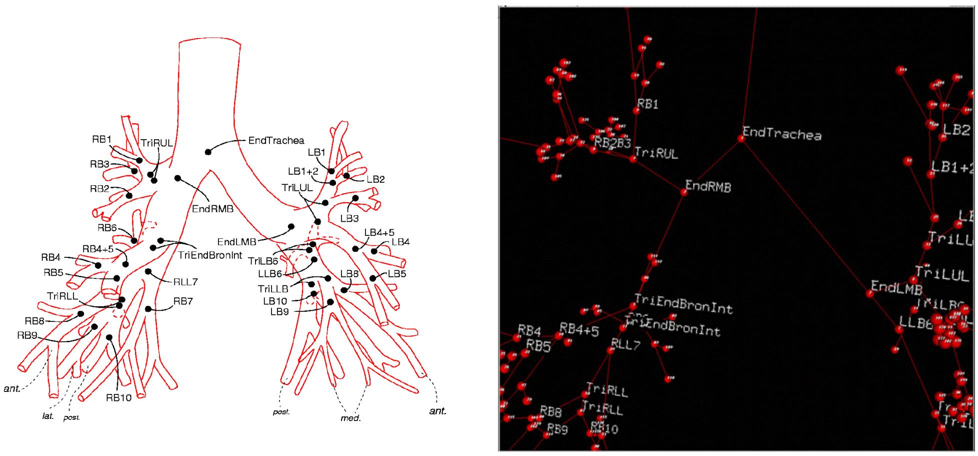


Figure 4.19: Anatomical labeling of a human airway tree. Anatomical atlas (i.e., airway tree with assigned labels) in which labels refer to segments, but are assigned to the terminating branch-point of the respective segment (left). The labeled formal tree (right).

4.12 Re-sampling 2D Slices

The quantitative analysis of airway trees is based on the segmentation of the airway tree. Consequently, approximate surfaces of the airway tree segments together with the centerline can be used to guide the accurate detection of the airway walls [342]. A 2D slice is re-sampled from the original gray-level volume (i.e., a high resolution in vivo CT scan) using centerline, and a 2D slice is re-sampled from the earlier airway segmentation result, see Figure 4.20.

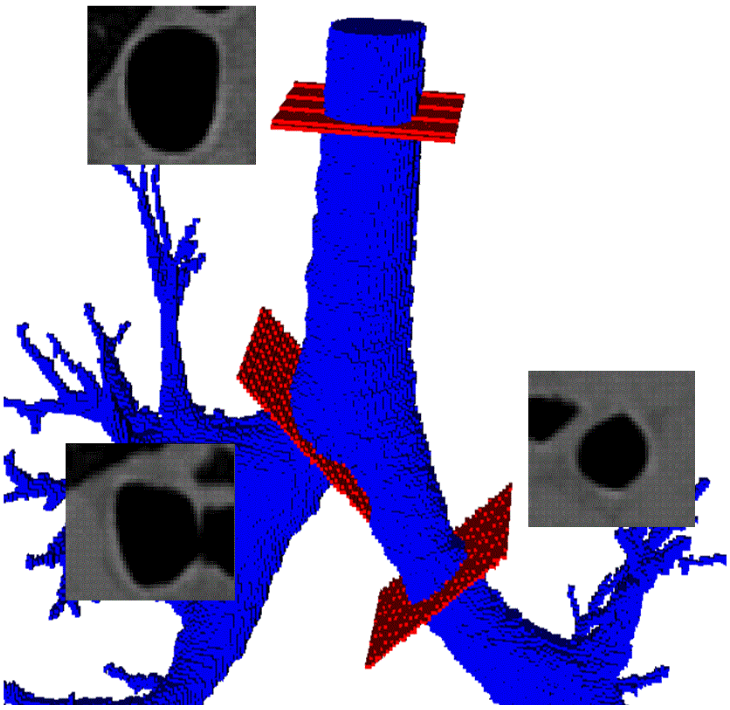


Figure 4.20: A segmented airway tree with three 2D slices perpendicular to airway segments in the original CT scan.

Chapter 5

Summary of New Scientific Results

This dissertation presents a selection of my results all originate from a period well after defending my PhD dissertation in the year 2000. The three theses are related to Chapters 2-4, and include only results in which my contribution were essential.

Thesis 1: Topology Preservation

This thesis summarizes our theoretical results concerning diversified topological problems:

1.1: Kardos, Németh and me proposed configuration-based and point-based sufficient conditions for topology-preserving operators for 2D pictures.

See Section 2.1. *Related publications*: [119, 120, 122, 123, 124, 126, 208, 248].

1.2: I gave both symmetric and asymmetric point-based sufficient conditions for topology-preserving reductions acting on 3D $(26, 6)$ pictures on $\mathbb{Z}^3$.

See Section 2.2. *Related publication*: [236].

1.3: Instead of investigating the sets of altered points, I proposed a novel sufficient condition for topology-preserving operators that takes the alteration rules of operators into consideration. I proved that the general-simple alteration rules provide pairs of equivalent and topology-preserving sequential and parallel operators.

See Section 2.3. *Related publications*: [238, 239, 241].

1.4: Kardos and me disclosed the relationships among the different types of sufficient conditions for topology-preserving reductions.

See Section 2.4. *Related publications*: [245, 248, 249]

Thesis 2: Advanced Thinning

Thinning is an iterative object reduction process capable of producing all types of skeleton-like features in a topology-preserving way. For reasons of scope, this thesis contains only some selected results concerning thinning:

2.1: Although thinning seems to be a time-consuming process, I proposed an easy and computationally efficient implementation scheme for arbitrary (sequential and parallel) thinning algorithms.

See Section 3.1. *Related publications*: [230, 231, 234].

2.2: I proposed a safe technique for designing topology-preserving parallel thinning algorithms.

Kardos, Németh, and me generated some families of 2D and 3D topology-preserving parallel thinning algorithms. For reasons of scope, this dissertation describes only 15 parallel 3D algorithms (5 curve-thinning, 5 surface-thinning, and 5 kernel-thinning ones) that are derived from our asymmetric point-based sufficient condition (for topology-preserving 3D parallel reductions) combined with the major parallel thinning strategies and some types of geometric constraints.

See Section 3.2. *Related publications*: [206, 208, 209, 211, 235, 236].

2.3: I proved that the deletion rules of two existing parallel thinning algorithms (i.e., a fully parallel 2D algorithm [181], and a 6-subiteration 3D surface-thinning algorithm [82]) are general-simple, thus they are equivalent. I constructed a pair of topology-preserving equivalent sequential and parallel 4-subiteration 2D thinning algorithms and a pair of topology-preserving equivalent sequential and parallel fully parallel 3D surface-thinning algorithms. These algorithms were implemented and tested by Kardos and Németh.

For reasons of scope, only four pairs of equivalent sequential and parallel subiteration-based 3D surface-thinning algorithms are described in this dissertation.

See Section 3.3. *Related publications*: [242, 243, 244, 246].

2.4: I proposed two maximal 3D curve-thinning algorithms (i.e, algorithms that can produce centerlines containing only non-simple points and 3D-curve-endpoints). These topology-preserving algorithms have been successfully applied in several medical image processing applications.

See Section 3.4. *Related publications*: [224, 227, 231].

Thesis 3: Quantitative Analysis of Pulmonary Airway Trees

My 3D thinning algorithms have been involved in several biomedical applications. For reasons of scope, in this dissertation, my method for quantitative analysis of pulmonary airway trees is only described.

3.1: I proposed, implemented, tested, and validated a complex method for extracting reliable centerlines from segmented human airway trees. It consists of the following phases: correction of the segmented tree, identification of the tree root, producing centerline, centerline pruning, and centerline smoothing.

3.2: I proposed, implemented, tested, and validated algorithms for symbolic description of airway trees and characterizing the individual branches. My algorithms solve the following problems: identification of branchpoints, generating a formal tree structure, tree partitioning, calculating associated measures (i.e., length, radius, volume, and surface area) for individual branches, and re-sampling 2D slices perpendicular to airway segments.

Among others, the collaborators were Eric A. Hoffman and Milan Sonka (i.e., outstanding researchers at The University of Iowa, Iowa City, IA, USA).

See Chapter 4. *Related publications*: [99, 227, 231, 341].

Bibliography

[1] N. Ahuja, J-H. Chuang: Shape representation using a generalized potential field model, IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 169–176, 1997.

[2] B. Altinoklu, I. Ulusoy, S. Tari: A probabilistic sparse skeleton based object detection, Pattern Recognition Letters 83, 243–250, 2016.

[3] N. Amenta, S. Choi: Voronoi methods for 3D medial axis, In: K. Siddiqi, S. Pizer (Eds.): Medial representations – Mathematics, algorithms and applications, Computational Imaging and Vision 37, Springer, 223-239, 2008.

[4] L. Antiga, B. Ene-Iordache, G. Remuzzi, A. Remuzzi: Automatic generation of glomerular capillary topological organization, Microvascular Research 62, 346–354, 2001.

[5] C. Arcelli, G. Sanniti di Baja: Ridge points in Euclidean distance maps, Pattern Recognition Letters 13, 237–243, 1992.

[6] C. Arcelli, G. Ramella: Finding grey-skeletons by iterated pixel removal, Image and Vision Computing 13, 159–167, 1995.

[7] C. Arcelli, L. Serino: Parallel reduction operators for gray-tone pictures, Int. J. Pattern Recognition and Artificial Intelligence 14, 281–295, 2000,

[8] C. Arcelli, G. Sanniti di Baja, L. Serino: New removal operators for surface skeletonization, In: Proc. 13th Int. Conf. Discrete Geometry for Computer Imagery, DGCI 2006, Lecture Notes in Computer Science 4245, Springer, 555–566, 2006.

[9] C. Arcelli, G. Sanniti di Baja, L. Serino: A parallel algorithm to skeletonize the distance transform of 3D objects, Image and Vision Computing 27, 666–672, 2009.

[10] N. Arica, F.T. Yarman-Vural: An overview of character recognition focused on off-line handwriting, IEEE IEEE Transactions on Systems Man and Cybernetics 31, 216–233, 2001.

[11] C. Aslan, A. Erdem, E. Erdem, S. Tari: Disconnected skeleton: shape at its absolute scale, IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 2188–2203, 2008.

[12] D. Attali, A. Montanvert: Computing and simplifying 2D and 3D continuous skeletons, Computer Vision and Image Understanding 67, 261–273, 1997.

[13] F. Aurenhammer, R. Klein, D-T. Lee: Voronoi diagrams and Delaunay triangulations, World Scientific Publishing Company, 2013.

[14] X. Bai, L.J. Latecki, W-Y. Liu: Skeleton pruning by contour partitioning with discrete curve evolution, IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 449–462, 2007.

[15] X. Bai, L.J. Latecki: Path similarity skeleton graph matching, IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 1282–1292, 2008.

[16] X. Bai, X. Yang, D. Yu, L.J. Latecki: Skeleton-Based Shape Classification Using Path Similarity, Int. J. Pattern Recognitionand Artificial Intelligence 22, 733–746, 2008.

[17] C. Bajaj, Z. Yu, M. Auer: Volumetric feature extraction and visualization of tomographic molecular imaging, J. Structural Biology 144, 132–143, 2003.

[18] P. Barreira-González, M. Gómez-Delgado, F. Aguilera-Benavente: From raster to vector cellular automata models: A new approach to simulate urban growth with the help of graph theory, Computers, Environment and Urban Systems 54, 119131, 2015.

[19] C. Bauer, R. Adam, D.A. Stoltz, R.R. Beichel: Computer-aided analysis of airway trees in micro-CT scans of ex vivo porcine lung tissue, Computerized Medical Imaging and Graphics 36, 601–609, 2012.

[20] R. Beichel, T. Pock, Ch. Janko, B. Zotter, B. Reitinger, A. Bornik, H. Bischof, K. Palágyi[7], E. Sorantin, G. Werkgartner, M. Sonka: Liver segment approximation in CT data for surgical resection planning, In: Medical Imaging 2004: Image Processing, Proceedings of SPIE Vol. 5370, 1435–1446, 2005.

[21] S.B.M. Bell, F.C. Holroyd, D.C. Mason: A digital geometry for hexagonal pixels, Image and Vision Computing 7, 194–204, 1989.

[22] G. Bertrand, Z. Aktouf: A 3D thinning algorithm using subfields, In: SPIE Proc. of Conf. Vision Geometry, 113-124, 1994.

[23] G. Bertrand: On P-simple points, Compte Rendu de l'Académie des Sciences de Paris, Série Math. I(321), 1077–1084, 1995.

[24] G. Bertrand: P-simple points: A solution for parallel thinning, In: Proc. 5th Int. Conference on Discrete Geometry for Computer Imagery, DGCI 2005, 233–242, 1995.

[25] G. Bertrand, M. Couprie: A new 3D parallel thinning scheme based on critical kernels, In: Proc. 13th Int. Conf. Discrete Geometry for Computer Imagery, DGCI 2006, Lecture Notes in Computer Science 4245, Springer, 580–591, 2006.

[26] G. Bertrand, M. Couprie: Transformations topologiques discrètes. In: Coeurjolly, D., Montanvert, A., Chassery, J. (Eds.), Géométrie discrète et images numériques, Hermès Science Publications, 187–209, 2007.

[27] G. Bertrand, M. Couprie: Two-dimensional parallel thinning algorithms based on critical kernels, J. Mathematical Imaging and Vision 31, 35–56, 2008.

[28] G. Bertrand, M. Couprie: On parallel thinning algorithms: Minimal non-simple sets, P-simple points and critical kernels, J. Mathematical Imaging and Vision 35, 23–35, 2009.

[29] G. Bertrand, M. Couprie: Powerful parallel and symmetric 3D thinning schemes based on critical kernels, J. Mathematical Imaging and Vision 48, 134–148, 2014.

[30] R.L. Blanding, G.M. Turkiyyah, D.W. Storti, M.A. Ganter: Skeleton-based three-dimensional geometric morphing, Computational Geometry 15, 129–148, 2000.

[31] I. Bloch: Fuzzy skeleton and skeleton by influence zones: A review, In: P.K. Saha, G. Borgefors, G. Sanniti di Baja (Eds.): Skeletonization: Theory, methods and applications, Academic Press, 71–88, 2017.

[7] In the Bibliography, the name of the author of this dissertation is typeset in cyan.

[32] H. Blum: A transformation for extracting new descriptors of shape, In: W. Wathen-Dunn (Ed.), Models for the perception of Speech and visual form, MIT Press, 362-380, 1967.

[33] H. Blum: Biological shape and visual science (Part I), J. Theoretical Biology 38, 205–287, 1973.

[34] H. Blum, R.N. Nagel: Shape description using weighted symmetric axis features, Pattern Recognition 10, 167–180, 1978.

[35] G. Borgefors: Distance transformations in arbitrary dimensions, Computer Vision, Graphics, and Image Processing 27, 321–345, 1984.

[36] G. Borgefors, I. Nyström, G. Sanniti di Baja: Discrete skeletons from distance transforms in 2D and 3D, In: K. Siddiqi, S. Pizer (Eds.): Medial representations – Mathematics, algorithms and applications, Computational Imaging and Vision 37, Springer, 155-190, 2008.

[37] S. Bouix, K. Siddiqi, A. Tannenbaum: Flux driven automatic centerline extraction, Medical Image Analysis 9, 209-221, 2005.

[38] S. Bouix, J.C. Pruessner, D.L. Collins, K. Siddiqi: Hippocampal shape analysis using medial surfaces, NeuroImage 25, 1077–1089, 2005.

[39] J.W. Brandt: Describing a solid with the three-dimensional skeleton, In: Proc. SPIE Conf. Curves and Surfaces in Computer Vision and Graphics III, Vol. 1830, 258–269, 1992.

[40] J.W. Brandt, V.R. Algazi: Continuous skeleton computation by Voronoi diagram, CVGIP: Image Understanding 55, 329–337, 1992.

[41] A. Brennecke, T. Isenberg: 3D shape matching using skeleton graphs, In: Proc. Conf. Simulation and Visualization, SimVis 2004, 299–310, 2004.

[42] M. Breuss, A. Bruckstein, P. Maragos (Eds.): Innovations for shape analysis, Springer-Verlag, 2013.

[43] V.E. Brimkov, R.P. Barneva: Analytical honeycomb geometry for raster and volume graphics, The Computer Journal 48, 180–199, 2005.

[44] D. Brunner, G. Brunnett: Mesh segmentation using the object skeleton graph, In: Proc. Int. Conf. Computer Graphics and Imaging, CGIM 2004, 48–55, 2004.

[45] L. Calabi, J.A. Riley: The skeletons of stable plane sets, Technical Report AF 19 (628-5711), Parke Mathematical Laboratories, 1967.

[46] L. Calabi, W.E. Hartnett: Shape recognition, prairie fires, convex deficiencies and skeletons, The American Mathematical Monthly 75, 335–342, 1968.

[47] H-M. Chan, A.C.S. Chung: Efficient 3D-3D vascular registration based on multiple orthogonal 2D projections, In: Proc. Int. Workshop on Biomedical Image Registration, WBIR 2003, Lecture Notes in Computer Science 2717, Springer, 301–310, 2003.

[48] R-F. Chang, S-F. Huang, Y-H. Lee, D-R. Chen, W.K. Moon: Breast tumor angiogenesis analysis using 3D power Doppler ultrasound, In: Medical Imaging 2006: Ultrasonic Imaging and Signal Processing, Proc. of the SPIE 6147, 37–44, 2006.

[49] H. Chatbri, K. Kameyama, P. Kwan: A comparative study using contours and skeletons as shape representations for binary image matching, Pattern Recognition Letters 76, 59–66, 2016.

[50] S.Y. Chen, J.D. Carroll, J.C. Messenger: Quantitative analysis of reconstructed 3-D coronary arterial tree and intracoronary devices, IEEE Transactions on Medical Imaging 21, 724–740, 2002.

[51] J-H. Chuang, C-H. Tsai, M-C. Ko: Skeletonization of three-dimensional object using generalized potential field, IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 1241–1251, 2000.

[52] T. Cormen, C. Leiserson, R. Rivest: Introduction to algorithms, MIT Press, 1990.

[53] N.D. Cornea, D. Silver, P. Min: Curve-skeleton properties, applications, and algorithms, IEEE Transactions on Visualization and Computer Graphics 13, 530–548, 2007.

[54] L.F. Costa, R.M. Cesar: Shape classification and analysis: Theory and practice (2nd edition), CRC Press, 2009.

[55] M. Couprie, G. Bertrand: A 3D sequential thinning scheme based on critical kernels, In: 12th Int. Symposium Mathematical Morphology, ISMM 2015, Lecture Notes in Computer Science 9082, Springer, 549–560, 2015.

[56] M. Couprie: Note on fifteen 2D parallel thinning algorithms, Internal Report, Université de Marne-la-Vallée, IMG2006-01, 2006.

[57] M. Couprie, A.V. saude, G. Bertrand: Euclidean homotopic skeleton based on critical kernels, In: Proc. 19th Brazilian Symposium on Computer Graphics and Image Processing, SIBGRAPI'06, 307–314, 2006.

[58] M. Couprie, G. Bertrand: Asymmetric parallel 3D thinning scheme and algorithms based on isthmuses, Pattern Recognition Letters 76, 22–31, 2016.

[59] J. Damon: Geometry and medial structure, In: K. Siddiqi, S. Pizer (Eds.), Medial representations – Mathematics, algorithms and applications, Computational Imaging and Vision 37, Springer, 69–123, 2008.

[60] P.-E. Danielsson: Euclidean distance mapping, Computer Graphics and Image Processing 14, 227–248. 1980.

[61] E. de Aguiar, C. Theobalt, S. Thrun, H-P. Seidel: Automatic conversion of mesh animations into skeleton-based animations, Computer Graphics Forum (Proc. Eurographics EG'08) 27, 389–397, 2008.

[62] C. Di Ruberto: Recognition of shapes byattributed skeletal graphs, Pattern Recognition 37, 21–31, 2004.

[63] M. Ding, R. Tong, S-H. Liao, J. Dong: An extension to 3D topological thinning method based on LUT for colon centerline extraction, Computer Methods and Programs in Biomedicine 94, 39–47, 2009.

[64] E.R. Dougherty, R.A. Lotufo: Hands-on morphological image processing, SPIE Press, 2003.

[65] O. Due Trier, A.K. Jain, T. Taxt: Feature extraction methods for character recognition-a survey, Pattern Recognition 29, 641–662, 1996.

[66] P.L. Erdős, S.G. Hartke, L. van Iersel, I. Miklós: Graph realizations constrained by skeleton graphs, Electronic Journal of Combinatorics 24, P2.47, 2017.

[67] R. Fabbri, L.F. Costa, J.C. Torelli, O.M. Bruno: 2D Euclidean distance transform algorithms: A comparative survey, ACM Computing Surveys 40, 2:1–2:44, 2008.

[68] A. Farina, Z.M. Kovacs-Vajna, A. Leone: Fingerprint minutiae extraction from skeletonized binary images, Pattern Recognition 32, 877–889, 1999.

[69] A. Fazekas, K. Palágyi, Gy. Kovács, G. Németh: Skeletonization based on metrical neighborhood sequences, In: Proc. 6th Int. Conf. Computer Vision Systems, ICVS 2008, Lecture Notes in Computer Science 5008, Springer, 333–342, 2008.

[70] C. Feng, A. Jalba, A. Telea: A descriptor for voxel shapes based on the skeleton cut space, In: Proc. Eurographics 2016 Workshop on 3D Object Retrieval, 13–20, 2016.

[71] T. Fletcher, S. Pizer, A. Thall, A. Gash: Shape Modeling and Image Visualization in 3D with M-Rep Object Models, Technical Report, University of North Carolina at Chapel Hill Chapel Hill, 2000.

[72] H. Frimmel, J. Näppi, H. Yoshida: Centerline-based colon segmentation for CT colonography, Medical Physics 32, 2665–2672, 2005.

[73] H. Fujiyoshi, A.J. Lipton, T. Kanade: Real-time human motion analysis by image skeletonization, IEICE Transactions on Information and Systems 87, 113–120, 2004.

[74] N. Gagvani, D. Silver: Animating volumetric models, Graphical Models 63, 443–458, 2001.

[75] F.J. Gaspar, J.L. Gracia, F.J. Lisbona, C. Rodrigo: On geometric multigrid methods for triangular grids using three-coarsening strategy, Applied Numerical Mathematics 59, 1693–1708, 2009.

[76] H. Geng, J. Yang, W. Tan, Y. Yuan, F. Han, D. Zhao: Fast 3D skeleton extraction of airways and applications to virtual bronchoscopy, In: Proc. 26th Chinese Control and Decision Conference, CCDC 2014, 3879–3884, 2014.

[77] G. Gerig, T. Kollera, G. Székely, C. Brechbuhler, O. Kubler: Symbolic descriptions of 3-D structures applied to cerebral vessel tree obtained from MR angiography volume data, In: Proc. 13th Int. Conference on Information Processing in Medical Imaging, IPMI'93, Lecture Notes in Computer Science 687, Springer, 94–111, 1993.

[78] A. Giachettia, M. Tuveria, G. Zanettia, E. Piccinini: Web-based 3D quantitative measurements of abdominal aortic aneurysms, International Congress Series 1230, 395–400, 2001.

[79] P.J. Giblin, B.B. Kimia: Local forms and transitions of the medial axis, In: K. Siddiqi, S. Pizer (Eds.), Medial representations – Mathematics, algorithms and applications, Computational Imaging and Vision 37, Springer, 37–68, 2008.

[80] B-W. Goh: Strategies for shape matching using skeletons, Computer Vision and Image Understanding 110, 326–345, 2008.

[81] M. Golay: Hexagonal parallel pattern transformation, IEEE Transactions on Computers 18, 733–740, 1969.

[82] W.X. Gong, G. Bertrand: A simple parallel 3d thinning algorithm. In: Proc. 10th IEEE Int. Conf. Pattern Recognition, ICPR'90, 188–190, 1990.

[83] R.C. Gonzalez, R.E. Woods: Digital Image Processing (3rd Edition), Prentice Hall, 2008.

[84] S. Greenberg, M. Aladjem, D. Kogan, I. Dimitrov: Fingerprint image enhancement using filtering techniques, In: Proc. IEEE Int. Conf. Pattern Recognition, ICPR 2000, 322–325, 2000.

[85] H. Greenspan, M. Laifenfeld, S. Einav, O. Barnea: Evaluation of centerline extraction algorithms in quantitative coronary angiography, IEEE Transactions on Medical Imaging 20, 928–941, 2001.

[86] G.J. Grevera: Distance transform algorithms and their implementation and evaluation, In: A.A. FaragJasjit, S. Suri (Eds.), Deformable Models I., Topics in Biomedical Engineering, International Book Series, Springer, 33–60, 2007.

[87] G. Guo, X. Wang, W. Zhang, P. Li, X. Dai: A new pruning method for medial axis of planar free-form shape, In: Proc. IEEE Int. Conference on Progress in Informatics and Computing, PIC 2014, 265–268, 2014.

[88] Z. Guo, Z., R.W. Hall: Fast fully parallel thinning algorithms, CVGIP: Image Understanding 55, 317–328, 1992.

[89] R.W. Hall: Parallel connectivity-preserving thinning algorithms, In: T.Y. Kong, A. Rosenfeld (Eds.), Topological Algorithms for Digital Image Processing, Elsevier Science B. V., 145–179, 1996.

[90] R.W. Hall, T.Y. Kong, A. Rosenfeld: Shrinking binary images, In: T.Y. Kong, A. Rosenfeld (Eds.), Topological Algorithms for Digital Image Processing, Elsevier Science B. V., 31–98, 1996.

[91] S.M. Han, R.S. Elsayed, S.W. Ham, A. Mahajan, F. Fleischman, V.L. Rowe, M.J. Cunningham, F.A. Weaver: Comparison of intravascular ultrasound- and centerline computed tomography-determined aortic diameters during thoracic endovascular aortic repair, J. Vascular Surgery 66, 1184–1191, 2017.

[92] W.E. Hartnett: Plane figures: their skeletons and quench functions, Technical Report PML No 60429 SR-3, Parke Mathematical Laboratories, 1965.

[93] M.S. Hassouna, A.A. Farag: Variational curve skeletons using gradient vector flow, IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 2257–2274, 2009.

[94] T. He, L. Hong, D. Chen, Z. Liang: Reliable path for virtual endoscopy: ensuring complete examination of human organs, IEEE Transactions on Visualization and Computer Graphics 7, 333–342, 2001.

[95] L. He, C.Y. Han, B. Everding, W.G. Wee: Graph matching for object recognition and recovery, Pattern Recognition 37, 1557–1560, 2004.

[96] J. Hedrich, C. Yang, C. Feinen, S. Schäfer, D. Paulus, M. Grzegorzek: Extended investigations on skeleton graph matching for object recognition, In: Proc. 8th Int. Conf. Computer Recognition Systems, CORES 2013, 371–381, 2013.

[97] G.T. Herman, B.M. Carvalho: Multiseeded segmentation using fuzzy connectedness, IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 460–474, 2001.

[98] W. Hesselink, J. Roerdink: Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform, IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 2204–2217, 2008.

[99] E.A. Hoffman, J.M. Reinhardt, M. Sonka, B.A. Simon, J. Guo, O. Saba, D. Chon, S. Samrah, H. Shikata, J. Tschirren, K. Palágyi, K.C. Beck, G. McLennan: Characterization of the Interstitial Lung Diseases via Density-Based and Texture-Based Analysis of Computed Tomography Images of Lung Structure and Function, Academic Radiology 10, 1104–1118, 2003.

[100] L. Huang, A. Bijaoui: Astronomical image data compression by morphological skeleton transformation, Experimental Astronomy 1, 311–327, 1990.

[101] X. Huang, D. Wen, Y. Zhao, Q. Wang, W. Zhou, D. Deng: Skeleton-based tracing of curved fibers from 3D X-ray microtomographic imaging, Results in Physics 6, 170–177, 2016.

[102] G. Iordanescu, R.M. Summers: Automated centerline for computed tomography colonography, Academic Radiology 10, 1291–1301, 2003.

[103] M.Y. Jaisimha, R.M. Haralick, D. Dori: Quantitative performance evaluation of thinning algorithms under noisy conditions, In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, CVPR 1994, 678–683, 1994.

[104] A.C. Jalba, J. Kustra, A.C. Telea: Surface and curve skeletonization of large 3D models on the GPU, IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 1495–1508, 2013.

[105] R. Jennane, A. Almhdie, G. Aufort, E. Lespessailles: 3D shape-dependent thinning method for trabecular bone characterization, Medical Physics 39, 168–178, 2012.

[106] D. Jin, P.K. Saha: A new fuzzy skeletonization algorithm and its applications to medical imaging, In: Proc. 17th Int. Conf. Image Analysis and Processing, 662671, 2013.

[107] D. Jin, K.S.Iyer, C. Chen, E.A. Hoffman, P.K. Saha: A robust and efficient curve skeletonization algorithm for tree-like objects using minimum cost paths, Pattern Recognition Letters 76, 32–40, 2016.

[108] D. Jin, C. Chen, E.A. Hoffman, P.K. Saha: Curve skeletonization using minimum-cost path, In: P.K. Saha, G. Borgefors, G. Sanniti di Baja (Eds.): Skeletonization: Theory, methods and applications, Academic Press, 151–180, 2017.

[109] S. Jouili, S. Tabbone: Towards performance evaluation of graph-based representation, In: 8th Int. Workshop Graph-Based Representations in Pattern Recognition, GbRPR 2011, 72–81, 2011.

[110] T. Ju, M.L. Baker, W. Chiu: Computing a family of skeletons of volumetric models for shape description, Computer-Aided Design 39, 352–360, 2007.

[111] D.G. Kang, J.B. Ra: A new path planning algorithm for maximizing visibility in computed tomography colonography, IEEE Transactions on Medical Imaging 24, 957–968, 2005.

[112] P. Kardos, G. Németh, K. Palágyi: An order-independent sequential thinning algorithm, In: Proc. 13th Int. Workshop on Combinatorial Image Analysis, IWCIA 2009, Lecture Notes in Computer Science 5852, Springer, 162–175, 2009.

[113] P. Kardos, K. Palágyi: On topology preservation for hexagonal parallel thinning algorithms, In: Proc. 14th Int. Workshop on Combinatorial Image Analysis, IWCIA'2011, Lecture Notes in Computer Science 6636, Springer, 31–42, 2011.

[114] P. Kardos, K. Palágyi: Order-independent sequential thinning in arbitrary dimensions, In: Proc. Int. Conf. Signal and Image Processing and Applications, SIPA 2011, 129–134, 2011.

[115] P. Kardos, K. Palágyi: Hexagonal parallel thinning algorithms based on sufficient conditions for topology preservation, In: Proc. Int. Symposium of Computational Modeling on Objects Presented in Images: Fundamentals, Methods, and Applications, CompIMAGE'12, CRC Press, Taylor & Francis Group, 63–68, 2012.

[116] P. Kardos, K. Palágyi: Isthmus-based order-independent sequential thinning, In: Proc. 9th Int. Conf. Signal Processing, Pattern Recognition and Applications, SPPRA 2012, 28–34, 2012.

[117] P. Kardos, K. Palágyi: On order-independent sequential thinning, In: Proc. 3rd IEEE Int. Conf. Cognitive Infocommunications, CogInfoCom 2012, 149–154, 2012.

[118] P. Kardos, K. Palágyi: On topology preservation for triangular thinning algorithms, In: Proc. 15th Int. Workshop on Combinatorial Image Analysis, IWCIA 2012, Lecture Notes in Computer Science 7655, Springer, 128–142, 2012.

[119] P. Kardos, K. Palágyi: Sufficient conditions for topology preserving additions and general operators, In: Proc. 14th Int. Conf. Computer Graphics and Imaging, CGIM 2013, 107–114, 2013.

[120] P. Kardos, K. Palágyi: Topology-preserving hexagonal thinning, Int. J. Computer Mathematics 90, 1607–1617, 2013.

[121] P. Kardos, K. Palágyi: Parallel thinning on the triangular grid, In: Proc. 4th IEEE Int. Conf. Cognitive Infocommunications, CogInfoCom 2013, 277–282, 2013.

[122] P. Kardos, K. Palágyi: On topology preservation in triangular, square, and hexagonal grids, In: Proc. 8th Int. Symposium on Image and Signal Processing and Analysis, IEEE/EURASIP, ISPA 2013, 782–787, 2013.

[123] P. Kardos, K. Palágyi: Sufficient conditions for general 2D operators to preserve topology, In: Proc. 16th Int. Workshop on Combinatorial Image Analysis, IWCIA 2014, Lecture Notes in Computer Science 8466, Springer, 101–112, 2014.

[124] P. Kardos, K. Palágyi: Topology preservation on the triangular grid, Annals of Mathematics and Artificial Intelligence 75, 53–68, 2015.

[125] P. Kardos, K. Palágyi: Unified characterization of P-simple points in triangular, square, and hexagonal grids, In: Proc. Int. Symposium on Computational Modeling of Objects Presented in Images: Fundamentals, Methods, and Applications, CompIMAGE'16, Lecture Notes in Computer Science 10149, Springer, 79–88, 2016.

[126] P. Kardos, K. Palágyi: On topology preservation of mixed operators in triangular, square, and hexagonal grids, Discrete Applied Mathematics 216, 441–448, 2017.

[127] P. Kardos, K. Palágyi: Maximal P-simple sets on $(8, 4)$ pictures, In: Proc. 6th Int. Symposium on Computational Modeling of Objects Presented in Images: Fundamentals, Methods, and Applications, CompIMAGE'18, Lecture Notes in Computer Science 10986, Springer, 23–32, 2019.

[128] B.B. Kimia, A. Tannenbaum, S.W. Zucker: Shape, shocks, and deformations. I. The components of two-dimensional shape and the reaction-diffusion space, Int. J. Computer Vision 15, 189–224, 1995.

[129] R. Kimmel, D. Shaked, N. Kiryati: Skeletonization via distance maps and level sets, Computer Graphics and Image Processing 62, 382–391, 1995.

[130] T. Kitasaka, K. Mori, H.-I. Hasegawa, Y. Suenaga, J.-I. Toriwaki: Extraction of bronchus regions from 3D chest X-ray CT images by using structural features of bronchus, In: Computer Assisted Radiology and Surgery, CARS 2003, Int. Congress Series, vol. 1256, Elsevier, 240–245, 2003.

[131] F. Klein: Euclidean skeletons, In: Proc. 5th Scandinavian Conf. Image Analysis, SCIA'87, 443–450, 1987.

[132] G. Klette, M. Pan: Characterization of curve-like structures in 3D medical images, In: Proc. Image Vision Computing New Zealand, 164–175, 2005.

[133] R. Klette, A. Rosenfeld: Digital geometry: Geometric methods for digital picture analysis, Morgan Kaufmann, 2004.

[134] H. Kobatake, Y. Yoshinaga: Detection of spicules on mammogram based on skeleton analysis, IEEE Transactions on Medical Imaging 15, 235–245, 1996.

[135] J. Koehoorn., C. Feng, J. Kustra, A. Jalba, A. Telea: Unified part-patch segmentation of mesh shapes using surface skeletons, In: P.K. Saha, G. Borgefors, G. Sanniti di Baja (Eds.): Skeletonization: Theory, methods and applications, Academic Press, 89–122, 2017.

[136] T.Y. Kong: A digital fundamental group, Computers and Graphics 13, 159–166, 1989.

[137] T.Y. Kong, A. Rosenfeld: Digital topology: Introduction and survey, Computer Vision, Graphics, and Image Processing 48, 357–393, 1989.

[138] T.Y. Kong, A.W. Roscoe, A. Rosenfeld: Concepts of digital topology, Topology and its Applications 46, 219–262, 1992.

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

[140] T.Y. Kong, C-J. Gau: Minimal non-simple sets in 4-dimensional binary images with (8,80)-adjacency, In: Combinatorial Image Analysis, IWCIA 2004, Lecture Notes in Computer Science 3322, Springer, 318–333, 2004.

[141] T.Y. Kong: Critical kernels, minimal nonsimple sets, and hereditarily simple sets in binary images on n-dimensional polytopal complexes, In: P.K. Saha, G. Borgefors, G. Sanniti di Baja (Eds.), Skeletonization: Theory, Methods and Applications, Academic Press, 211–256, 2017.

[142] V.A. Kovalevsky: Geometry of locally finite spaces, Publishing House, 2008.

[143] R. Kresch, D. Malah: Skeleton-based morphological coding of binary images, IEEE Transactions on Image Processing 7, 1387–1399, 1998.

[144] S. Krinidis, M. Krinidis: Empirical mode decomposition on skeletonization pruning, Image and Vision Computing 31, 533–541, 2013.

[145] K.J. Kruszynski, R. van Liere, J.A. Kaandorp: Quantifying differences in skeletonization algorithms: A case study, In: Proc. Int. Conf. Visualization, Imaging, and Image Processing, VIIP 2005, 2005.

[146] K.Y. Kwon, B.C. Lee, S.W. Chae: Medial surface generation using chordal axis transformation in shell structures, Computers and Structures 84, 1673–1683, 2006.

[147] L. Lam, S.-W. Lee, C.Y. Suen: Thinning methodologies – A comprehensive survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 14, 869–885, 1992.

[148] L. Lam, C.Y. Suen: An evaluation of parallel thinning algorithms for character recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 17, 914–919, 1995.

[149] C. Lantuéjoul: Skeletonization in quantitative metallography, In: R. Haralick, J.-C. Simon (Eds.), Issues in Digital Image Processing, 107–135, 1980.

[150] T. Lee, R.L. Kashyap, C. Chu: Building skeleton models via 3–D medial surface/axis thinning algorithms, CVGIP: Graphical Models and Image Processing 56, 462–478, 1994.

[151] D. Lesage, E.D. Angelini, I. Bloch, G. Funka-Lea: A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes, Medical Image Analysis 13, 819–845, 2009.

[152] F. Leymarie, M.D. Levine: Simulating the grassfire transform using an active contour model, IEEE Transactions on Pattern Analysis and Machine Intelligence 14, 56–75, 1992.

[153] F. Leymarie, B.B. Kimia: From the infinitely large to the infinitely small, In: K. Siddiqi, S. Pizer (Eds.): Medial representations – Mathematics, algorithms and applications, Computational Imaging and Vision 37, Springer, 327-351, 2008.

[154] Z. Li, Y. Zhang, G. Li, H. Shao, W. Li, X. Tang: A robust coronary artery identification and centerline extractionmethod in angiographies, Biomedical Signal Processing and Control 16, 1–8, 2005.

[155] Z. Liang, M.A. Ioannidis, I. Chatzis: Geometric and topological analysis of three-dimensional porous media: Pore space partitioning based on morphological skeletonization, J. Colloid and Interface Science 221, 13–24, 2000.

[156] Z. Liang, M.A. Ioannidis, I. Chatzis: Permeability and electrical conductivity of porous media from 3D stochastic replicas of the microstructure, Chemical Engineering Science 55, 5247–5262, 2000.

[157] K. Lidayova, H. Frimmel, C. Wang, E. Bengtsson, O. Smedby: Skeleton-based fast, fully automated generation of vessel tree structure for clinical evaluation of blood vessel systems, In: P.K. Saha, G. Borgefors, G. Sanniti di Baja (Eds.): Skeletonization: Theory, methods and applications, Academic Press, 345–382, 2017.

[158] J.-M. Lien, G. Kurillo, R. Bajcsy: Skeleton-based data compression for multi-camera tele-immersion system, In: Proc. 3rd Int. Symposium on Visual Computing, ISVC 2007, Lecture Notes in Computer Science 4841, Springer, 714–723, 2007.

[159] C.L. Lin, J.D. Miller: Pore structure analysis of particle beds for fluid transport simulation during filtration, Int. J. Mineral Processing 73, 281–294, 2004.

[160] H. Liu, Z-H. Wu, X. Zhang, D.F. Hsu: A skeleton pruning algorithm based on information fusion, Pattern Recognition Letters 34, 1138–1145, 2013.

[161] B. Liu, A.C. Tele, J.B.T.M. Roerdink, G.J. Clapworthy, D. Williams, P. Yang, F. Dong, V. Codreanu, A. Chiarini: Parallel centerline extraction on the GPU, Computers and Graphics 41, 72–83, 2014.

[162] C. Lohou, G. Bertrand: A 3D 6-subiteration curve thinning algorithm based on P-simple points, Discrete Applied Mathematics 151, 198–228, 2005.

[163] C. Lohou: Detection of the non-topology preservation of Ma's 3D surface-thinning algorithm, by the use of P-simple points, Pattern Recognition Letters 29, 822–827, 2008.

[164] C. Lohou: Detection of the non-topology preservation of Ma and Sonkas algorithm, by the use of P-simple points, Computer Vision and Image Understanding 114, 384–399, 2010.

[165] C. Lohou, J. Dehos: Automatic correction of Ma and Sonkas thinning algorithm using P-simple points. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 1148–1152, 2010.

[166] C. Lohou, J. Dehos: An automatic correction of Ma's thinning algorithm based on P-simple points. J. Mathematical Imaging and Vision 36, 54–62, 2010.

[167] K. Lou, N. Iyer, S. Jayanti, Y. Kalyanaraman, S. Prabhakar, K. Ramani: Effectiveness and efficiency of three-dimensional shape retrieval, J. Engineering Design 16, 175–194, 2005.

[168] C.M. Ma: On topology preservation in 3D thinning, CVGIP: Image Understanding 59, 328–339, 1994.

[169] C.M. Ma: A 3D fully parallel thinning algorithm for generating medial faces, Pattern Recognition Letters 16, 83–87, 1995.

[170] C.M. Ma, M. Sonka: A fully parallel 3D thinning algorithm and its applications, Computer Vision and Image Understanding 64, 420–433, 1996.

[171] C.M. Ma, S-Y. Wan: Parallel thinning algorithms on 3D (18,6) binary images. Computer Vision Image Understanding **80**, 364–378 (2000)

[172] C.M. Ma, S.Y. Wan: A medial-surface oriented 3-d two-subfield thinning algorithm, Pattern Recognition Letters 22, 1439–1446, 2001.

[173] C.M. Ma, S.Y. Wan: A thinning algorithm on 2D gray-level images, In: SPIE Proceedings: Medical Imaging 4684, 893–900, 2002.

[174] C.M. Ma, S.Y. Wan, H.K. Chang: Extracting medial curves on 3D images, Pattern Recognition Letters 23, 895–904, 2002.

[175] C.M. Ma, S.Y. Wan, J.D. Lee: Three-dimensional topology preserving reduction on the 4-subfields, IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 1594–1605, 2002.

[176] J. Ma, S.W. Bae, S. Choi: 3D medial axis point approximation using nearest neighbors and the normal field, Visual Computer 28, 7–19, 2012.

[177] D. Macrini, K. Siddiqi, S.J. Dickinson: From skeletons to bone graphs: Medial abstraction for object recognition, In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, CVPR 2008, 324–332, 2008.

[178] A. Mademlis, P. Daras, A. Axenopoulos, D. Tzovaras, M.G. Strintzis: Combining topological and geometrical features for global and partial 3D shape retrieval, IEEE Transactions on Multimedia 10, 819–831, 2008.

[179] G. Malandain, G. Bertrand: Fast characterization of 3D simple points, In: IEEE Int. Conf. Pattern Recognition, 232–235, 1992.

[180] A. Manzanera, T.M. Bernard, F. Preteux, B. Longuet: Medial faces from a concise 3D thinning algorithm, In: Proc. 7th IEEE Internat. Conf. Computer Vision, ICCV'99, Vol. 1, 337–343, 1999.

[181] A. Manzanera, T.M. Bernard, F. Preteux, B. Longuet: n-dimensional skeletonization: a unified mathematical framework. J. Electronic Imaging 11, 25–37, 2002.

[182] P. Maragos, R. Schafer: Morphological skeleton representation and coding of binary images, IEEE Transactions on Acoustics, Speech, and Signal Processing 34, 1228–1244, 1986.

[183] J. Marchadier, D. Arques, S. Michelin: Thinning grayscale well-composed images, Pattern Recognition Letters 25, 581–590, 2004.

[184] S. Marchand-Maillet, Y.M. Sharaiha: Binary digital image processing – A discrete approach, Academic Press, 2000.

[185] E. Marchandise, C. Geuzaine, J.F. Remacle: Cardiovascular and lung mesh generation based on centerlines, Int. J. Numerical Methods in Biomedical Engineering 29, 665–682, 2013.

[186] R. Marie, O. Labbani-Igbida, E.M. Mouaddib: The Delta Medial Axis: A fast and robust algorithm for filtered skeleton extraction, Pattern Recognition 56, 26–39, 2016.

[187] B. Matejek, D. Wei, X. Wang, J. Zhao, K. Palágyi, H. Pfister: Synapse-aware skeleton generation for neural circuits, In: Proc. 22nd Int. Conf. on Medical Image Computing and Computer Assisted Intervention, MICCAI 2019, Lecture Notes in Computer Science 11764, Springer, 227–235, 2019.

[188] G. Matheron: Examples of topological properties of skeletons, In: J. Serra (Ed.), Image analysis and mathematical morphology Vol. 2, Theoretical Advances, Academic Press, 217–238, 1988.

[189] C.R. Maurer, R. Qi, V. Raghavan: A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions, IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 265–270, 2003.

[190] L. Middleton, J. Sivaswamy: Hexagonal image processing: A practical approach, Advances in Pattern Recognition, Springer-Verlag, 2005.

[191] U. Montanari: A method for obtaining skeletons using a quasi-Euclidean distance, J. Association for Computing Machinery 15, 600–624, 1968.

[192] A.S. Montero, J. Lang: Skeleton pruning by contour approximation and the integer medial axis transform, Computers and Graphics 36, 477–487, 2012.

[193] D.G. Morgenthaler: Three-dimensional simple points, serial erosion, parallel thinning, and skeletonization, Technical Report TR-1005, University of Maryland, 1981.

[194] M. Moreira-Soares, R. Coimbra, L. Rebelo, J. Carvalho, R.D.M. Travasso: Angiogenic Factors produced by Hypoxic Cells are a leading driver of Anastomoses in Sprouting Angiogenesis – a computational study, Scientific Reports 8, Article number: 8726, 2018.

[195] K. Mori, J. Hasegawa, J. Toriwaki, H. Anno, K. Katada: A fast rendering method using the tree structure of objects in virtualized bronchus endoscope system, In: Proc. 4th Int. Conference on Visualization in Biomedical Computing, VBC'96, Lecture Notes in Computer Science 1131, Springer, 33–42, 1996.

[196] K. Mori, J. Hasegawa, Y. Suenaga, J. Toriwaki: Automated anatomical labeling of the bronchial branch and its application to the virtual bronchoscopy system, IEEE Transactions on Medical Imaging 19, 103–114, 2000.

[197] J. Mukherjee, P.P. Das, B.N. Chatterjee: Thinning of 3-D images using the Safe Point Thinning Algorithm (SPTA), Pattern Recognition Letters 10, 167–173, 1989.

[198] N.J. Naccache, R. Shinghal: SPTA: a proposed algorithm for thinning binary patterns, IEEE Transactions on Systems, Man, and Cybernetics 14, 409–418, 1984.

[199] L.R. Nackman, S.M. Pizer: Three-dimensional shape description using the symmetric axis transform. I. Theory, IEEE Transactions on Pattern Analysis and Machine Intelligence 16, 187–202, 1985.

[200] B. Nagy: Characterization of digital circles in triangular grid, Pattern Recognition Letters 25, 1231–1242, 2004.

[201] B. Nagy: Digital disks by weighted distances in the triangular grid, In: 19th IAPR Int. Conf. Discrete Geometry for Computer Imagery, DGCI 2016, Lecture Notes in Computer Science 9647, Springer, 206–218, 2016.

[202] M. Näf, G. Székely, R. Kikinis, M.E. Shenton, O. Kb̈ler: 3D Voronoi skeletons and their usage for the characterization and recognition of 3D organ shape, Computer Vision and Image Understanding 66, 147–161, 1997.

[203] G. Németh, K. Palágyi: Parallel thinning algorithms based on Ronse's sufficient conditions for topology preservation, In: P. Wiederhold, R. Barneva (Eds.), Progress in Combinatorial Image Analysis, Research Publishing Services, 183–194, 2009.

[204] G. Németh, P. Kardos, K. Palágyi: Topology preserving 3D thinning algorithms using four and eight subfields, In: Proc. Int. Conf. Image Analysis and Recognition, ICIAR 2010, Lecture Notes in Computer Science 6111, Springer, 316–325, 2010.

[205] G. Németh, P. Kardos, K. Palágyi: Topology preserving 2-subfield 3D thinning algorithms, In: Proc. 7th Int. Conf. Signal Processing, Pattern Recognition and Applications SPPRA 2010, 310–316, 2010.

[206] G. Németh, P. Kardos, K. Palágyi: A family of topology-preserving 3D parallel 6-subiteration thinning algorithms, In: Proc. 14th Int. Workshop on Combinatorial Image Analysis, IWCIA 2011, Lecture Notes in Computer Science 6636, Springer, 17–30, 2011.

[207] G. Németh, P. Kardos, K. Palágyi: Thinning combined with iteration-by-iteration smoothing for 3D binary images, Graphical Models 73, 335–345, 2011.

[208] G. Németh, P. Kardos, K. Palágyi: 2D parallel thinning and shrinking based on sufficient conditions for topology preservation, Acta Cybernetica 20, 125–144, 2011.

[209] G. Németh, K. Palágyi: Topology preserving parallel thinning algorithms, Int. J. Imaging Systems and Technology 21, 37–44, 2011.

[210] G. Németh, K. Palágyi: 2D parallel thinning algorithms based on isthmus-preservation, In: Proc. 7th Int. Symposium on Image and Signal Processing and Analysis, IEEE/EURASIP, ISPA 2011, 585–590, 2011.

[211] G. Németh, K. Palágyi: 3D parallel thinning algorithms based on isthmuses, In: Proc. 14th Int. Conf. Advanced Concepts for Intelligent Vision Systems, ACIVS 2012, Lecture Notes in Computer Science 7517, Springer, 325–335, 2012.

[212] G. Németh, Gy. Kovács, A. Fazekas, K. Palágyi: A method for quantitative comparison of 2D skeletons, Acta Polytechnica Hungarica 13, 123–142, 2016.

[213] T. Ngo, T-T. Nguyen, Q-T. Ngo, D-D. Nguyen, S-C. Chu: Similarity shape based on skeleton graph matching, J. Information Hiding and Multimedia Signal Processing 7, 1254–1264, 2016.

[214] I. Nyström: Skeletonization applied to magnetic resonance angiography images, In: Medical Imaging 1998: Image Processing, Proceedings of SPIE Vol. 3338, 693–701, 1998.

[215] I. Nyström, Ö. Smedby: Skeletonization of volumetric vascular imagesdistance information utilized for visualization, J. Combinatorial Optimization 5, 27–41, 2001.

[216] R.L. Ogniewicz, M. Ilg: Voronoi skeletons: Theory and applications, In: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 1992, 63–69, 1992.

[217] R.L. Ogniewicz: Skeleton-space: A multiscale shape description combining region and boundary information, In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, CVPR 1993, 746–751, 1994.

[218] R.L. Ogniewicz, O. Kübler: Hierarchic Voronoi skeletons, Pattern Recognition 28, 343–359, 1995.

[219] K. Palágyi, A. Kuba: A 3D 6-subiteration thinning algorithm for extracting medial lines, Pattern Recognition Letters 19, 613–627, 1998.

[220] K. Palágyi, A. Kuba: A hybrid thinning algorithm for 3D medical images, J. Computing and Information Technology 6, 149–164, 1998.

[221] K. Palágyi, A. Kuba: A parallel 3D 12-subiteration thinning algorithm, Graphical Models and Image Processing 61, 199–221, 1999.

[222] K. Palágyi, A. Kuba: Directional 3D thinning using 8 subiterations, In: Proc. 8th Int. Conf. Discrete Geometry for Computer Imagery, DGCI'99, Lecture Notes in Computer Science 1568, Springer, 325–336, 1999.

[223] Palágyi K.: Vékonyító algoritmusok 3D képekre, Doktori értekezés, Szegedi Tudományegyetem, 2000. (in Hungarian)

[224] K. Palágyi, E. Sorantin, E. Balogh, A. Kuba, Cs. Halmai, B. Erdőhelyi, K. Hausegger: A sequential 3D thinning algorithm and its medical applications, In: Proc. 17th Int. Conf. Information Processing in Medical Imaging, IPMI 2001, Lecture Notes in Computer Science 2082, Springer, 409–415, 2001.

[225] K. Palágyi: A 3-subiteration 3D thinning algorithm for extracting medial surfaces, Pattern Recognition Letters 23, 663–675, 2002.

[226] K. Palágyi, J. Tschirren, M. Sonka: Quantitative analysis of three-dimensional tubular tree structures, In: Medical Imaging 2003: Image Processing, Proceedings of SPIE Vol. 5032, 277–287, 2003.

[227] K. Palágyi, J. Tschirren, M. Sonka: Quantitative analysis of intrathoracic airway trees: methods and validation, In: Proc. 18th Int. Conf. Information Processing in Medical Imaging, IPMI 2003, Lecture Notes in Computer Science 2732, Springer, 222–233, 2003.

[228] K. Palágyi, J. Tschirren, E.A. Hoffman, M. Sonka: Assessment of intrathoracic airway trees: Methods and in vivo validation In: Proc. Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis: ECCV 2004 Workshops CVAMIA and MMBIA, Lecture Notes in Computer Science 3117, Springer, 341–352, 2004.

[229] K. Palágyi: A 2-subfield 3D thinning algorithm for extracting medial curves, In: Proc. Joint Hungarian-Austrian Conf. Image Processing and Pattern Recognition, HACIPPR 2005, Austrian Computer Society, 135–142, 2005.

[230] K. Palágyi: Computationally efficient thinning in 3D, In: Proc. ECCV 2006 Workshop on Computation Intensive Methods for Computer Vision, CIMCV 2006, 85–98, 2006.

[231] K. Palágyi, J. Tschirren, E.A. Hoffman, M. Sonka: Quantitative analysis of pulmonary airway tree structures, Computers in Biology and Medicine 36, 974–996, 2006.

[232] K. Palágyi: A subiteration-based surface-thinning algorithm with a period of three, In: Proc. 29th Annual Symposium of the German Association for Pattern Recognition, DAGM 2007, Lecture Notes in Computer Science 4713, Springer, 294–303, 2007.

[233] K. Palágyi: A 3-subiteration surface-thinning algorithm, In: Proc. 12th Int. Conf. Computer Analysis of Images and Patterns, CAIP 2007, Lecture Notes in Computer Science 4673, Springer, 628–635, 2007.

[234] K. Palágyi: A 3D fully parallel surface-thinning algorithm, Theoretical Computer Science 406, 119–135, 2008.

[235] K. Palágyi, G. Németh: Fully parallel 3D thinning algorithms based on sufficient conditions for topology preservation, In: Proc. 15th IAPR Int. Conf. Discrete Geometry for Computer Imagery, DGCI 2009, Lecture Notes in Computer Science 5810, Springer, 481–492, 2009.

[236] K. Palágyi, G. Németh, P. Kardos: Topology preserving parallel 3D thinning algorithms, In: V.E. Brimkov, R.P. Barneva (Eds.), Digital Geometry Algorithms. Theoretical Foundations and Applications to Computational Imaging, Springer, 165–188, 2012.

[237] K. Palágyi: Parallel 3D 12-subiteration thinning algorithms based on isthmuses, In: Proc. 9th Int. Symposium on Visual Computing, ISVC 2013, Lecture Notes in Computer Science 8033, Springer, 87–98, 2013.

[238] K. Palágyi: Deletion rules for equivalent sequential and parallel reductions, In: Proc. 18th Iberoamerican Congress on Pattern Recognition, Part I, CIARP 2013, Lecture Notes in Computer Science 8285, Springer, 17–24, 2013.

[239] K. Palágyi: Equivalent sequential and parallel reductions in arbitrary binary pictures, Int. J. Pattern Recognition and Artificial Intelligence 28, 1460009-1–1460009-16, 2014.

[240] K. Palágyi: A sequential 3D curve-thinning algorithm based on isthmuses, In: Proc. 10th Int. Symposium on Visual Computing, ISVC 2014, Lecture Notes in Computer Science 8888, Springer, 406–415, 2014.

[241] K. Palágyi: Topology-preserving general operators in arbitrary binary pictures, In: Proc. 19th Iberoamerican Congress on Pattern Recognition, CIARP 2014, Lecture Notes in Computer Science 8827, Springer, 22–29, 2014.

[242] K. Palágyi: Equivalent 2D sequential and parallel thinning algorithms, In: Proc. 16th Int. Workshop on Combinatorial Image Analysis, IWCIA 2014, Lecture Notes in Computer Science 8466, Springer, 91–100, 2014.

[243] K. Palágyi, G. Németh, P. Kardos: Topology-preserving equivalent parallel and sequential 4-subiteration 2D thinning algorithms, In: Proc. 9th Int. Symposium on Image and Signal Processing and Analysis, IEEE/EURASIP, ISPA 2015, 306–311, 2015.

[244] K. Palágyi, G. Németh, P. Kardos: Equivalent sequential and parallel subiteration-based surface-thinning algorithms, In: Proc. 17th Int. Workshop on Combinatorial Image Analysis, IWCIA 2015, Lecture Notes in Computer Science 9448, Springer, 31–45, 2015.

[245] K. Palágyi: P-simple points and general-simple deletion rules, In: Proc. 19th IAPR Int. Conf. Discrete Geometry for Computer Imagery, DGCI 2016, Lecture Notes in Computer Science 9647, Springer, 143–153, 2016.

[246] K. Palágyi, G. Németh: A pair of equivalent sequential and fully parallel 3D surface-thinning algorithms, Discrete Applied Mathematics 216, 348–361, 2017.

[247] K. Palágyi: Simplifier points in 2D binary images, In: Proc. 18th Int. Workshop on Combinatorial Image Analysis, IWCIA 2017, Lecture Notes in Computer Science 10256, Springer, 3–15, 2017.

[248] K. Palágyi, P. Kardos: A single-step 2D thinning scheme with deletion of P-simple points, In: Proc. 22nd Iberoamerican Congress on Pattern Recognition, CIARP 2017, Lecture Notes in Computer Science 10657, Springer, 475–482, 2018.

[249] K. Palágyi: How sufficient conditions are related for topology-preserving reductions, Acta Cybernetica 23, 939–958, 2018.

[250] K. Palágyi, G. Németh: Fixpoints of iterated reductions with equivalent deletion rules, In: Proc. 19th Int. Workshop on Combinatorial Image Analysis, IWCIA 2018, Lecture Notes in Computer Science 11255, Springer, 17–27, 2018.

[251] K. Palágyi, G. Németh: Endpoint-based thinning with designating safe skeletal points, In: Proc. 6th Int. Symposium on Computational Modeling of Objects Presented in Images: Fundamentals, Methods, and Applications, CompIMAGE'18, Lecture Notes in Computer Science 10986, Springer, 3–15, 2019.

[252] K. Palágyi, G. Németh: Centerline extraction from 3D airway trees using anchored shrinking, In: Proc. 14th Int. Symposium on Visual Computing, ISVC 2019, Lecture Notes in Computer Science 11845, Springer, 419–430, 2019.

[253] J.R. Parker: Extracting vectors from raster images, Computers and Graphics 12, 75–79, 1988.

[254] N. Passat, C. Ronse, J. Baruthio, J-P. Armspach, C. Maillot: Magnetic resonance angiography: From anatomical knowledge modeling to vessel segmentation, Medical Image Analysis 10, 259–274, 2006.

[255] P. Peter, M. Breuss: Refined homotopic thinning algorithms and quality measures for skeletonisation methods, In: M. Breuss, A. Bruckstein, P. Maragos (Eds.): Innovations for shape analysis, Springer-Verlag, 77–92, 2013.

[256] S.M. Pizer, A. Thall, D.T. Chen: M-reps: A new object representation for graphics, Technical Report, University of North Carolina at Chapel Hill Chapel Hill, 1999.

[257] S.M. Pizer, P.T. Fletcher, S. Joshi, A. Thall, J.Z. Chen, Y. Fridman, D.S. Fritsch, G. Gash, J.M. Glotzer, M.R. Jiroutek, C. Lu, K.E. Muller, G. Tracton, P. Yushkevich, E.L. Chaney: Deformable m-reps for 3D medical image segmentation, Int. J. Computer Vision 55, 85–106, 2003.

[258] S.M. Pizer, Q. Han, S. Joshi, P.T. Fletcher: Synthesis, deformation, and statistics of 3D objects via m-reps, In: K. Siddiqi, S. Pizer (Eds.), Medial representations – Mathematics, algorithms and applications, Computational Imaging and Vision 37, Springer, 241–266, 2008.

[259] S.M. Pizer, M. Styner, T. Terriberry, R. Broadhurst, S. Joshi, E. Chaney, P.T. Fletcher: Statistical applications with deformable m-reps – Anatomic object segmentation and discrimination, In: K. Siddiqi, S. Pizer (Eds.), Medial representations – Mathematics, algorithms and applications, Computational Imaging and Vision 37, Springer, 269–308, 2008.

[260] S.M. Pizer, D.S. Fritsch, P.A. Yushkevich, V.E. Johnson, E.L. Chaney: Segmentation, registration, and measurement of shape variation via image object shape, IEEE Transactions on Medical Imaging 18, 851865, 1999.

[261] L. Pothuaud, P. Porion, E. Lespessailles, C.L. Benhamou, P. Levitz: A new method for three-dimensional skeleton graph analysis of porous media: application to trabecular bone microarchitecture, Journal of Microscopy 199, 149–161, 2000.

[262] F.P. Preparata, M.I. Shamos: Computational geometry, Springer-Verlag, 1985.

[263] C. Pudney: Distance-ordered homotopic thinning: A skeletonization algorithm for 3D digital images, Computer Vision and Image Understanding 72, 404–413, 1998.

[264] W.R. Quadros, B. Gurumoorthy, K. Ramaswami, F.B. Prinz: Skeletons for representation and reasoning in engineering applications, Engineering with Computers 17, 186–198, 2001.

[265] W.R. Quadros, V. Vyas, M. Brewer, S.J. Owen, K. Shimada: A computational framework for automating generation of sizing function in assembly meshing via disconnected skeletons, Engineering with Computers 26, 231–247, 2010.

[266] V. Ranwez, P. Soille: Order independent homotopic thinning for binary and grey tone anchored skeletons, Pattern Recognition Letters 23, 687–702, 2002.

[267] B. Raynal, M. Couprie, V. Biri: Homeomorphic alignment of weighted trees, Pattern Recognition 43, 2937–2949, 2010.

[268] B. Raynal, M. Couprie: Isthmus-based 6-directional parallel thinning algorithms, In: Proc. 16th IAPR Int. Conf. Discrete Geometry for Computer Imagery, DGCI 2011, Lecture Notes in Computer Science 6607, Springer, 175–186, 2011.

[269] F. Reinders, M.E.D. Jacobson, F.H. Pos: Skeleton graph generation for feature shape description, In: Proc. of the Joint EUROGRAPHICS and IEEE TCVG Symposium on Visualization, Data Visualization 2000, 73–82, 2000.

[270] E. Remy, E. Thiel: Look-up tables for medial axis on squared Euclidean distance transform, In: Proc. 11th IAPR Int. Conf. Discrete Geometry for Computer Imagery, DGCI 2003, Lecture Notes in Computer Science 2886, Springer, 224–235, 2003.

[271] L. Risser, F. Plouraboué, P. Cloetens, C. Fonta: A 3D-investigation shows that angiogenesis in primate cerebral cortex mainly occurs at capillary level, Int. J. Developmental Neuroscience 27, 185–196, 2009.

[272] C. Ronse: Minimal test patterns for connectivity preservation in parallel thinning algorithms for binary digital images, Discrete Applied Mathematics 21, 67–79, 1988.

[273] C.D. Ruberto: Recognition of shapes by attributed skeletal graphs, Pattern Recognition 37, 21–31, 2004.

[274] R.J.T. Sadleir, P.F. Whelan: Fast colon centreline calculation using optimised 3D topological thinning, Computerized Medical Imaging and Graphics 29, 251–258, 2005.

[275] P.K. Saha, B.B. Chaudhury: A new shape-preserving parallel thinning algorithm for 3D digital images, Pattern Recognition 30, 1939–1955, 1997.

[276] P.K. Saha, B.B. Chaudhury, D.D. Majumder: A new shape-preserving parallel thinning algorithm for 3D digital images, Pattern Recognition 30, 1939–1955, 1997.

[277] P.K. Saha, Y. Xu, H. Duan, A. Heiner, G. Liang: Volumetric topological analysis: A novel approach for trabecular bone classification on the continuum between plates and rods, IEEE Transactions on Medical Imaging 29, 1821–1838, 2010.

[278] P.K. Saha, R. Strand, G. Borgefors: Digital topology and geometry in medical imaging: a survey, IEEE Transactions on Medical Imaging 34, 1940–1964, 2015.

[279] P.K. Saha, G. Borgefors (eds.): Special Issue on Skeletonization and its Application, Pattern Recognition Letters 76, 1–88, 2016.

[280] P.K. Saha, G. Borgefors, G. Sanniti di Baja: A survey on skeletonization algorithms and their applications, Pattern Recognition Letters 76, 3–12, 2016.

[281] P.K. Saha: Characterization of trabecular bone plate-rod micro-architecture using skeletonization and digital topologic and geometric analysis, In: P.K. Saha, G. Borgefors, G. Sanniti di Baja (Eds.): Skeletonization: Theory, methods and applications, Academic Press, 287–313, 2017.

[282] P.K. Saha, G. Borgefors, G. Sanniti di Baja (Eds.): Skeletonization: Theory, methods and applications, Academic Press, 2017.

[283] P.K. Saha, D. Jin, Y. Liu, G.E. Christensen, C. Chen: Fuzzy object skeletonization: Theory, algorithms, and applications, IEEE Transactions on Visualization and Computer Graphics 24, 2298–2314, 2018.

[284] T. Saito, J-I. Toriwaki: New algorithms for Euclidean distance transformation of an n-dimensional digitized picture with application, Pattern Recognition 27, 1551–1565, 1994.

[285] T. Saito, J. Toriwaki: A sequential thinning algorithm for three dimensional digital pictures using the Euclidean distance transformation, In: Proc. 9th Scandinavian Conf. Image Analysis, SCIA'95, 507–516, 1995.

[286] G. Sanniti di Baja: Well-shaped, stable, and reversible skeletons from the (3,4)-distance transform, J. Visual Communication and Image Representation 5, 107–115, 1994.

[287] A. Sarkar, A. Biswas, S. Mondal, M. Dutt: Finding shortest triangular path in a digital object, In: 19th IAPR Int. Conf. Discrete Geometry for Computer Imagery, DGCI 2016, Lecture Notes in Computer Science 9647, Springer, 385–397, 2016.

[288] M. Schaap, C.T. Metz, T. van Walsum, A.G. van der Giessen, A.C. Weustink, N.R. Mollet, C. Bauer, H. Bogunovic, C. Castro, X. Deng, E. Dikici, T. ODonnell, M. Frenay, O. Friman, M. Hernandez Hoyos, P.H. Kitslaar, K. Krissian, C. Kuhnel,

M.A. Luengo-Oroz, M. Orkisz, O. Smedby, M. Styner, A. Szymczak, H. Tek, C. Wang, S.K. Warfield, S. Zambal, Y. Zhang, G.P. Krestin, W.J. Niessen: Standardized evaluation methodology and reference database for evaluating coronary artery centerline extraction algorithms, Medical Image Analysis 13, 701–714, 2009.

[289] M. Schmitt: Some examples of algorithms analysis in computational geometry by means of mathematical morphological techniques, In: Lecture Notes in Computer Science 391, Springer, 225-246, 1989.

[290] G.L. Scott, S.C. Turner, A. Zisserman: Using a mixed wave/diffusion process to elicit the symmetry set, Image Vision Computing 7, 63–70, 1989.

[291] T.B. Sebastian, B.B. Kimia: Curves vs. skeletons in object recognition, Signal Processing 85, 247–263, 2005.

[292] D. Selle, B. Preim, A. Schenk, H.-O. Peitgen: Analysis of vasculature for liver surgical planning, IEEE Transactions on Medical Imaging 21, 1344–1357, 2002.

[293] L. Serino: Skeletonization of gray-tone images based on region analysis, In: Proc. 9th Iberoamerican Congress on Pattern Recognition, CIARP 2004, Lecture Notes in Computer Science 3287, Springer, 495–502, 2004.

[294] L. Serino, G. Sanniti di Baja: A new strategy for skeleton pruning, Pattern Recognition Letters 76, 41–48, 2016.

[295] L. Serino, C. Arcelli, G. Sanniti di Baja: Decomposing 3D objects in simple parts characterized by rectilinear spines, Int. J. Pattern Recognition and Artificial Intelligence 28, 1460010.119, 2014.

[296] J. Serra: Image Analysis and Mathematical Morphology, Academic Press, 1982.

[297] J. Shah: Gray skeletons and segmentation of shapes, Computer Vision and Image Understanding 99, 96–109, 2005.

[298] D. Shaked, A.M. Bruckstein: Pruning medial axes, Computer Vision Image Understanding 69, 156–169, 1998.

[299] W. Shen, X. Bai, R. Hu, H. Wang, L.J. Latecki: Skeleton growing and pruning with bending potential ratio, Pattern Recognition 44, 196–209, 2011.

[300] W. Shen, K. Zhao, J. Yuan, Y. Wang, Z. Zhang, X. Bai: Skeletonization in natural images and its application to object recognition, In: P.K. Saha, G. Borgefors, G. Sanniti di Baja (Eds.): Skeletonization: Theory, methods and applications, Academic Press, 259–286, 2017.

[301] H. Shin, A. Chavan, F. Witthus, D. Selle, G. Stamm, H-O. Peitgen, M. Galanski: Precise determination of aortic length in patients with aortic stent grafts: in vivo evaluation of a thinning algorithm applied to CT angiography data, European Radiology 11, 733–738, 2001.

[302] K. Siddiqi, S. Bouix, J. Shah: Skeletons via shocks of boundary evolution, In: K. Siddiqi, S. Pizer (Eds.), Medial representations – Mathematics, algorithms and applications, Computational Imaging and Vision 37, Springer, 127–154, 2008.

[303] K. Siddiqi, S. Bouix, A. Tannenbaum, S.W. Zucker: Hamilton-Jacobi Skeletons, Int. J. Computer Vision 48, 215-231, 2002.

[304] K. Siddiqi, S. Pizer (Eds.): Medial representations – Mathematics, algorithms and applications, Computational Imaging and Vision 37, Springer, 2008.

[305] D. Silin, T. Patzek: Pore space morphology analysis using maximal inscribed spheres, Physica A: Statistical Mechanics and its Applications 371, 336–360, 2006.

[306] A. Sobiecki, H.C. Yasan, A. Jalba, A. Telea: Qualitative comparison of contraction-based curve skeletonization methods, In: Int. Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing, ISMM 3013, Lecture Notes in Computer Science 7883, Springer, 425–439, 2013.

[307] A. Sobiecki, A. Jalba, A. Telea: Comparison of curve and surface skeletonization methods for voxel shapes, Pattern Recognition Letters 47, 147–156, 2014.

[308] P. Soille: Morphological image analysis: Principles and applications — 2nd edition, Springer-Verlag, 2004.

[309] C. Song, Z. Pang, X. Jing, C. Xiao: Distance field guided L1-median skeleton extraction, Visual Computer 34, 243–255, 2018.

[310] J. Song, F. Su, H. Li, S. Cai: Raster to vector conversion of construction engineering drawings, Automation in Construction 11, 597–605, 2002.

[311] E. Sorantin, Cs. Halmai, B. Erdőhelyi, L. Martonossy, K. Palágyi, B. Geiger: New advances for imaging of laryngotracheal stenosis by post processing of spiral-CT data, In: W. Hruby (Ed.), Digital (R)Evolution in Radiology, Springer, 275–285, 2000.

[312] E. Sorantin, E. Balogh, A. Vilanova i Bartrolí, K. Palágyi, L.G. Nyúl, S. Loncaric, M. Subasic, D. Kovacevic: Virtual Dissection of the Colon, In: D. Caramella, C. Bartolozzi (Eds.), 3D Image Processing – Techniques and Clinical Applications, Springer, 197–2009, 2002.

[313] E. Sorantin, Cs. Halmai, B. Erdőhelyi, K. Palágyi, L.G. Nyúl, K. Ollé, B. Geiger, F. Lindbichler, G. Friedrich, K. Kiesler: Spiral-CT-based assessment of tracheal stenoses using 3-D-skeletonization, IEEE Transactions on Medical Imaging 21, 263–273, 2002.

[314] E. Sorantin, G. Werkgartner, E. Balogh, A. Vilanova i Bartrolí, K. Palágyi, L.G. Nyúl, L. Ruskó: Virtual dissection and automated polyp detection of the colon based on spiral CT data - Techniques and preliminary experience on a cadaveric phantom, European Surgery – Acta Chirurgica Austriaca 34, 143–151, 2002.

[315] E. Sorantin, D. Mohadjer, F. Lindbichler, L.G. Nyúl, K. Palágyi, B. Geiger: Techniques in 3D Assessment of tracheal-stenosis by the mean of Spiral Computed Tomography (S-CT) and their applications, In: C.T. Leondes (Ed.), Medical Imaging Systems Technology – Modalities, World Scientific, 61–80, 2005.

[316] E. Sorantin, D. Mohadjer, L.G. Nyúl, K. Palágyi, F. Lindbichler, B. Geiger: New advances for imaging of laryngotracheal stenosis by post processing of spiral-CT data, In: W. Hruby (Ed.), Digital (R)Evolution in Radiology – Bridging the Future of Health Care, Springer, 297–308, 2006.

[317] E. Sorantin, E. Balogh, A. Vilanova i Bartrolí, K. Palágyi, L.G. Nyúl, F. Lindbichler, A. Ruppert: Virtual dissection of the colon based of spiral CT data, In: E. Neri, D. Caramella, C. Bartolozzi (Eds.), Image Processing in Radiology - Current Applications, Springer, 257–268, 2008.

[318] C. Spuhler, M. Harders, G. Székely: Fast and robust extraction of centerlines in 3D tubular structures using a scattered-snakelet approach, In: Proc. SPIE 6144, Medical Imaging 2006: Image Processing, 614442, 2006.

[319] R. Stefanelli, A. Rosenfeld: Some parallel thinning algorithms for digital pictures, J. Association for Computing Machinery 18, 255–264, 1971.

[320] S.N. Srihari, J.K. Udupa, M.M. Yau: Understanding the bin of parts, In: Proc. IEEE Conf. Decision Control, 44–49, 1979.

[321] R. Strand, G. Borgefors: Distance transforms for three-dimensional grids with non-cubic voxels, Computer Vision and Image Understanding 100, 294–311, 2005.

[322] R. Strand, N. Normand: Distance transform computation for digital distance functions, TheoreticalComputerScience 448, 80–93, 2012.

[323] A. Sud, M. Foskey, D. Manocha: Homotopy-preserving medial axis simplification, Int. J. Computational Geometry and Applications 17, 423–451, 2007.

[324] C.Y. Suen, P.S.P. Wang (Eds.): Thinning methodologies for pattern recognition, Series in Machine Perception and Artificial Intelligence 8, World Scientific, Singapore, 1994.

[325] H. Sundar, D. Silver, N. Gagvani, S. Dickinson: Skeleton based shape matching and retrieval, In: Proc. IEEE Conf. Shape Modeling International, 130–139, 2003.

[326] S. Svensson, G. Borgefors, I. Nyström: On reversible skeletonization using anchor-points from distance transforms, J. Visual Communication and Image Representation 10, 379–397, 1999.

[327] S. Svensson: Reversible surface skeletons of 3D objects by iterative thinning of distance transforms, In: G. Bertrand et al. (Eds.): Digital and Image Geometry, Lecture Notes in Computer Science 2243, Springer, 400–411, 2001.

[328] S. Svensson, G. Sanniti di Baja: Simplifying curve skeletons in volume images, Computer Vision and Image Understanding 90, 242–257, 2003.

[329] G. Székely: Voronoi skeletons, In: K. Siddiqi, S. Pizer (Eds.): Medial representations – Mathematics, algorithms and applications, Computational Imaging and Vision 37, Springer, 191-221, 2008.

[330] A. Tagliasacchi, T. Delame, M. Spagnuolo, N. Amenta, A. Telea: 3D skeletons: A State-of-the-art report, In: Proc. Conf. European Association for Computer Graphics, EG'16, 573–597, 2016.

[331] Z.S.G. Tari, J. Shah, H. Pien: Extraction of shape skeletons from grayscale images, Computer Vision and Image Understanding 66, 133–146, 1997.

[332] A. Telea, A. Vilanova: A robust level-set algorithm for centerline extraction, In: Proc. of the Symposium on Data Visualisation, VISSYM 2003, 185–194, 2003.

[333] D. Thibault, C.M. Gold: Terrain reconstruction from contours by skeleton construction, GeoInformatica 4, 349–373, 2000.

[334] B.C.S. Tom, S.N. Efstratiadis, A.K. Katsaggelos: Motion estimation of skeletonized angiographic images using elastic registration, IEEE Transactions on Medical Imaging 13, 450–460, 1994.

[335] R.S. Torresa, A.X. Falcao, L.F. Costa: A graph-based approach for multiscale shape analysis, Pattern Recognition 37, 1163–1174, 2004.

[336] A. Torsello, E.R. Hancock: A skeletal measure of 2d shape similarity, Computer Vision and Image Understanding 95, 1–29, 2004.

[337] J. Touati, M. Bologna, A. Schwein, F. Migliavacca, M. Garbey: A robust construction algorithm of the centerline skeleton for complex aortic vascular structure using computational fluid dynamics, Computers in Biology and Medicine 86, 6–17, 2017.

[338] Y.F. Tsao, K.S. Fu: A parallel thinning algorithm for 3–D pictures, Computer Graphics and Image Processing 17, 315–331, 1981.

[339] J. Tschirren, K. Palágyi, J.M. Reinhardt, E.A. Hoffman, M. Sonka: Segmentation, skeletonization, and branchpoint matching - A fully automated quantitative evaluation of human intrathoratic airway trees, In: Proc. 5th Int. Conf. Medical Image

Computing and Computer-Assisted Intervention, MICCAI 2002, Lecture Notes in Computer Science 2489, Springer, 12–19, 2002.

[340] J. Tschirren: Segmentation, anatomical labeling, branch-point matching, and quantitative analysis of human airway trees in volumetric CT images, Ph.D. dissertation, University of Iowa, IA, USA, 2003.

[341] J. Tschirren, G. McLennan, K. Palágyi, E.A. Hoffman, M. Sonka: Matching and anatomical labeling of human airway tree, IEEE Transactions on Medical Imaging 24, 1540–1547, 2005.

[342] J. Tschirren, E.A. Hoffman, G. McLennan, M. Sonka: Intrathoracic airway trees: Segmentation and airway morphology analysis from low-dose CT scans, IEEE Transactions on Medical Imaging 24, 1529–1539, 2005.

[343] J.K. Udupa, S. Samarasekera: Fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation, Graphical Models and Image Processing 58, 246–261, 1996.

[344] Z. Vasilkoski, A. Stepanyants: Detection of the optimal neuron traces in confocal microscopy images, J. Neuroscience Methods 178, 197–204, 2009.

[345] S. Vera, D. Gil, H.M. Kjer, J. Fagertun, R.R. Paulsen, M.A. Gonzalez Ballester: Medial structure generation for registration of anatomical structures, In: P.K. Saha, G. Borgefors, G. Sanniti di Baja (Eds.): Skeletonization: Theory, methods and applications, Academic Press, 313–344, 2017.

[346] L. Vincent: Efficient computation of various types of skeletons, In: Proc. SPIE 1445, Medical Imaging, 297–311, 1991.

[347] L. Wade, R.E. Parent: Automated generation of control skeletons for use in animation, Visual Computer 18, 97–110, 2002.

[348] M. Wan, Z. Liang, Q. Ke, L. Hong, I. Bitter, A. Kaufman: Automatic centerline extraction for virtual colonoscopy, IEEE Transactions on Medical Imaging 21, 1450–1460, 2002.

[349] S-Y. Wan, A.P. Kiraly, E.L. Ritman, W.E. Higgins: Extraction of the hepatic vasculature in rats using 3-D micro-CT images, IEEE Transactions on Medical Imaging 19, 964–971, 2000.

[350] S-Y. Wan, E.L. Ritman, W.E. Higgins: Multi-generational analysis and visualization of the vascular tree in 3D micro-CT images, Computers in Biology and Medicine 32, 55–71, 2002,

[351] T. Wan, H. Feng, C. Tong, D. Li, Z. Qin: Automated identification and grading of coronary artery stenoses with X-ray angiography, Computer Methods and Programs in Biomedicine 167, 13–22, 2018.

[352] J. Wang, T. Ohya, H. Liao, I. Sakuma, T. Wang, I. Tohnai, T. Iwai: Intravascular catheter navigation using path planning and virtual visual feedback for oral cancer treatment, Int. J. Medical Robotics and Computer Assisted Surgery 7, 214–224, 2011.

[353] J.W. Wang, Y. Tan: Efficient Euclidean distance transform algorithm of binary images in arbitrary dimensions. Pattern Recognition 46, 230–242, 2013.

[354] T. Wang, A. Basu: A note on 'a fully parallel 3D thinning algorithm and its applications', Pattern Recognition Letters 28, 501–506, 2007.

[355] D.R. Whittaker, J. Dwyer, M.F. Fillinger: Prediction of altered endograft path during endovascular abdominal aortic aneurysm repair with the Gore Excluder, Journal of Vascular Surgery 41, 575–583, 2005.

[356] W.C.K. Wong, A.C.S. Chung, S.C.H. Yu: Augmented vessels for pre-operative preparation in endovascular treatments, In: Proc. 7th Int. Conf. Medical Image Computing and Computer-Assisted Intervention, MICCAI 2004, Lecture Notes in Computer Science 3217, Springer, 602–609, 2004.

[357] C. Wuthrich, P. Stucki: An algorithm comparison between square- and hexagonal-based grids, Graphical Models and Image Processing 53, 324–339, 1991.

[358] W. Xie, R.P. Thompson, R. Perucchio: A topology-preserving parallel 3D thinning algorithm for extracting the curve skeleton, Pattern Recognition 36, 1529–1544, 2003.

[359] Y. Xu, H. Zhang, H. Li, G. Hu: An improved algorithm for vessel centerline tracking in coronary angiograms, Computer Methods and Programs in Biomedicine 88, 131–143, 2007.

[360] Z. Xu, U. Bagci, B. Foster, A. Mansoor, J.K. Udupa, D.J. Mollura: A hybrid method for airway segmentation and automated measurement of bronchial wall thickness on CT, Medical Image Analysis 24, 1–17, 2015.

[361] H-B. Yan, S-M. Hu, R.R. Martin, Y-L. Yang: Shape deformation using a skeleton to drive simplex transformations, IEEE Transactions on Visualization and Computer Graphics 14, 693–706, 2008.

[362] X. Yang, X. Bai, D. Yu, L.J. Latecki: Shape classification based on skeleton path similarity, In: Proc. Int. Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, EMMCVPR 2007, Lecture Notes in Computer Science 4679, Springer, 375–386, 2007.

[363] P.J. Yim, P.L. Choyke, R.M. Summers: Gray-scale skeletonization of small vessels in magnetic resonance angiography, IEEE Transactions on Medical Imaging 19, 568–576, 2000.

[364] Ye. Zeng, Y. Zhao, S. Liao, M. Liao, Y. Chen, X. Liu: Liver vessel segmentation based on centerline constraint andintensity model, Biomedical Signal Processing and Control 45, 192–201, 2018.

[365] F. Zhang, Z. Xiao, L. Geng, J. Wu, Z. Xu, D. Wang, J. Xi: Skeleton extraction based on anisotropic partial differential equation, Optik 126, 3692–3697, 2015.

[366] H. Zhang, V.O. Kheyfets, E.A. Finol: Robust infrarenal aortic aneurysm lumen centerline detection for rupture status classification, Medical Engineering and Physics 35, 1358–1367, 2013.

[367] W. Zhang, X. Wang, X. Li, J. Chen: 3D skeletonization feature based computer-aided detection system for pulmonary nodules in CT datasets, Computers in Biology and Medicine 92, 64–72, 2018.

[368] F. Zhao, H.R. Landis, S.J. Skerlos: Modeling of porous filter permeability via image-based stochastic reconstruction of spatial porosity correlations, Environmental Science and Technology 39, 239–247, 2005.

[369] F. Zhao, X. Tang: Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction, Pattern Recognition 40, 1270–1281, 2007.

[370] C. Zhu, X. Wang, S. Chen, M. Xia, Y. Huang, X. Pan: Automatic centerline extraction of cerebrovascular in 4D CTA based on tubular features, Physics Medicine and Biology 63, 125014, 2018.

[371] W-B. Zhu, B. Li, L-F. Tian, X-X. Li, Q-L. Chen: Topology adaptive vessel network skeleton extraction with novel medialness measuring function, Computers in Biology and Medicine 64, 40–61, 2015.