



HAL
open science

Discrete topology and geometry algorithms for quantitative human airway trees analysis based on computed tomography images

Michal Postolski

► **To cite this version:**

Michal Postolski. Discrete topology and geometry algorithms for quantitative human airway trees analysis based on computed tomography images. Other. Université Paris-Est; Uniwersytet łódzki, 2013. English. NNT: 2013PEST1094. pastel-00977514

HAL Id: pastel-00977514

<https://pastel.hal.science/pastel-00977514>

Submitted on 11 Apr 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



A thesis submitted in partial fulfillment for the
degree of **Doctor of Philosophy in Computer Science**

Presented by

Michał Postolski

Supervised by

Michel Couprie and Dominik Sankowski

**Discrete Topology and Geometry
Algorithms for Quantitative Human
Airway Trees Analysis Based on Computed
Tomography Images**

December 18, 2013

Committee in charge:

Piotr Kulczycki (*reviewer*)
Nicolas Normand (*reviewer*)
Nicolas Passat (*reviewer*)
Jan Sikora (*reviewer*)
Michel Couprie
Yukiko Kenmochi
Andrzej Napieralski
Dominik Sankowski

To my brilliant wife.

Title: *Discrete Topology and Geometry Algorithms for Quantitative Human Airway Trees Analysis Based on Computed Tomography Images.*

Abstract: Computed tomography is a very useful technique which allow non-invasive diagnosis in many applications for example is used with success in industry and medicine. However, manual analysis of the interesting structures can be tedious and extremely time-consuming, or even impossible due its complexity. Therefore in this thesis we study and develop discrete geometry and topology algorithms suitable for use in many practical applications, especially, in the problem of automatic quantitative analysis of the human airway trees based on computed tomography images.

In the first part, we define basic notions used in discrete topology and geometry then we showed that several class of discrete methods like skeletonisation algorithms, medial axes, tunnels closing algorithms and tangent estimators, are widely used in several different practical application.

The second part consist of a proposition and theory of a new methods for solving particular problems. We introduced two new medial axis filtering method. The hierarchical scale medial axis which is based on previously proposed scale axis transform, however, is free of drawbacks introduced in the previously proposed method and the discrete adaptive medial axis where the filtering parameter is dynamically adapted to the local size of the object. In this part we also introduced an efficient and parameterless new tangent estimators along three-dimensional discrete curves, called 3D λ maximal segment tangent direction.

Finally, we showed that discrete geometry and topology algorithms can be useful in the problem of quantitative analysis of the human airway trees based on computed tomography images. According to proposed in the literature design of such system we applied discrete topology and geometry algorithms to solve particular problems at each step of the quantitative analysis process. First, we propose a robust method for segmenting airway tree from CT datasets. The method is based on the tunnel closing algorithm and is used as a tool to repair, damaged by acquisition errors, CT images. We also proposed an algorithm for creation of an artificial model of the bronchial tree and we used such model to validate algorithms presented in this work. Then, we compare the quality of different algorithms using set of experiments conducted on computer phantoms and real CT dataset. We show that recently proposed methods which works in cubical complex framework, together with methods introduced in this work can overcome problems reported in the literature and can be a good basis for the further implementation of the system for automatic quantification of bronchial tree properties.

Keywords: digital topology; medial axis; skeleton; thinning; cubical complexes; tunnel closing; digital geometry; tangent estimators; airway tree; quantitative analysis;

Titre: *Topologie discrète et algorithmes géométriques pour l'analyse quantitative de l'arbre bronchique humain, basée sur des images de tomodensitométrie.*

Résumé: La tomodensitométrie est une technique très utile qui permet de mener avec succès des analyses non-invasives dans plusieurs types d'applications, par exemple médicales ou industrielles. L'analyse manuelle des structures d'intérêt présentes dans une image peut prendre beaucoup de temps, être laborieuse et parfois même impossible à faire en raison de sa complexité. C'est pour cela que dans cette thèse, nous proposons et développons des algorithmes nécessaires à cette analyse, basés sur la géométrie discrète et la topologie. Ces algorithmes peuvent servir dans de nombreuses applications, et en particulier au niveau de l'analyse quantitative automatique de l'arbre bronchique humain, sur la base d'images de tomodensitométrie.

La première partie introduit les notions fondamentales de la topologie et de la géométrie discrètes utiles dans cette thèse. Ensuite, nous présentons le principe de méthodes utilisées dans de nombreuses applications : les algorithmes de squelettisation, de calcul de l'axe médian, les algorithmes de fermeture de tunnels et les estimateurs de tangentes.

La deuxième partie présente les nouvelles méthodes que nous proposons et qui permettent de résoudre des problèmes particuliers. Nous avons introduit deux méthodes nouvelles de filtrage d'axe médian. La première, que nous appelons "hierarchical scale medial axis", est inspirée du "scale axis transform", sans les inconvénients qui sont propres à la méthode originale. La deuxième est une méthode nommée "discrete adaptive medial axis", où le paramètre de filtrage est adapté dynamiquement aux dimensions locales de l'objet. Dans cette partie, nous introduisons également des estimateurs de tangente nouveaux et efficaces, agissant sur des courbes discrètes tridimensionnelles, et que nous appelons "3D lambda maximal segment tangent direction".

Enfin, nous avons montré que la géométrie discrète et les algorithmes topologiques pouvaient être utiles dans le problème de l'analyse quantitative de l'arbre bronchique humain à partir d'images tomodensitométriques. Dans une chaîne de traitements de structure classique par rapport à l'état de l'art, nous avons appliqué des méthodes de topologie et de géométrie discrète afin de résoudre des problèmes particuliers dans chaque étape du processus de l'analyse quantitative. Nous proposons une méthode robuste pour segmenter l'arbre bronchique à partir d'un ensemble de données tomographiques (CT). La méthode est basée sur un algorithme de fermeture de tunnels qui est utilisé comme outil pour réparer des images CT abîmées par les erreurs d'acquisition. Nous avons aussi proposé un algorithme qui sert à créer un modèle artificiel d'arbre bronchique. Ce modèle est utilisé pour la validation des algorithmes présentés dans cette thèse. Ensuite nous comparons la qualité des différents algorithmes en utilisant un ensemble de test constitué de fantômes

(informatiques) et d'un ensemble de données CT réelles. Nous montrons que les méthodes récemment présentées dans le cadre des complexes cubiques, combinées avec les méthodes présentées dans cette thèse, permettent de surmonter des problèmes indiqués par la littérature et peuvent être un bon fondement pour l'implémentation future des systèmes de quantification automatique des particularités de l'arbre bronchique.

Mots Clés: topologie discrète; axe médian; squelette; amincissement; complexes cubiques; fermeture de tunnels; géométrie discrète; estimateurs de tangente; arbre bronchique; analyse quantitative;

Tytuł: *Algorytmy Topologii i Geometrii Dyskretnej do Ilościowej Analizy Drzew Oskrzelowych Człowieka na Podstawie Obrazów Tomografii Komputerowej*

Streszczenie: Tomografia Komputerowa jest bardzo użyteczną metodą diagnostyczną, która pozwala na nieinwazyjne postawienie diagnozy w wielu różnych zastosowaniach, na przykład jest z powodzeniem stosowana w przemyśle i medycynie. Niestety, manualna analiza interesujących struktur widocznych w obrazach tomograficznych może być trudna i ekstremalnie czasochłonna, a czasem nawet niemożliwa ze względu na ich złożoność. Dlatego w tej pracy studiowane są obecne, a także proponowane są nowe, algorytmy topologii i geometrii dyskretnej, które mogą zostać zastosowane do rozwiązania pojawiających się problemów w wielu zastosowaniach praktycznych. W szczególności, problemów związanych z automatyczną ilościową analizą drzew oskrzelowych człowieka na podstawie obrazów Tomografii Komputerowej.

W pierwszej części pracy, zdefiniowane zostały podstawowe pojęcia używane w topologii i geometrii dyskretnej. Następnie pokazane zostały różne możliwości praktycznego zastosowania podstawowych algorytmów dyskretnych takich jak np. algorytmów szkieletyzacji, osi centralnych, zamykania tuneli oraz estymatorów stycznych.

Druga część pracy zawiera szczegółowe propozycje nowych metod przeznaczonych do rozwiązywania konkretnych problemów. Najpierw zaproponowane zostały dwie nowe metody filtracji punktów osi centralnej. Pierwsza metoda "Hierarchical Scale Medial Axis" rozwiązuje problemy pojawiające się we wcześniej zaproponowanej metodzie "Scale Axis Transform", na której bazuje. Metoda druga "Discrete Adaptive Medial Axis" pozwala na dynamiczne dostosowanie parametru filtracji bazując na lokalnej wielkości obiektu. W tej części pracy zaproponowana została również nowa metoda estymacji stycznej wzdłuż trójwymiarowej krzywej dyskretnej nazwana "3D λ Maximal Segment Tangent Direction". Nowa metoda jest efektywna i bezparametrowa.

Na koniec, pokazana została możliwość zastosowania algorytmów topologii i geometrii dyskretnej w problemie ilościowej analizy drzew oskrzelowych człowieka na bazie obrazów tomografii komputerowej. Bazując na zaproponowanym w literaturze przedmiotu projekcie systemu analizy ilościowej, zastosowane zostały metody dyskretne do rozwiązania pojawiających się problemów. Najpierw, zaproponowano nową niezawodną metodę segmentacji drzewa oskrzelowego z obrazów tomografii komputerowej. Zaproponowana metoda wykorzystuje algorytm zamykania tuneli jako narzędzia do poprawy obrazów zniekształconych błędami akwizycji. W pracy zaproponowano również algorytm generacji komputerowego modelu drzewa oskrzelowego, który następnie zastosowano do walidacji zaproponowanych metod. Wyniki działania różnych metod dyskretnych porównano w eksperymentach przeprowadzonych zarówno na wygenerowanych modelach jak i na prawdziwych obrazach tomografii komputerowej. Uzyskane wyniki dowiodły, że nowe

metody bazujące na teorii kompleksów sześciennych oraz metody zaproponowane w tej pracy pozwalają rozwiązać problemy pojawiające się w literaturze przedmiotu. Ich zastosowanie może stanowić solidną podstawę późniejszej implementacji systemu do automatycznej ilościowej analizy właściwości drzew oskrzelowych.

Słowa kluczowe: topologia dyskretna; oś centralna; szkielet; pocienianie; zamykanie tuneli; geometria dyskretna; estymatory stycznej; drzewo oskrzelowe; analiza ilościowa;

Acknowledgements

Niniejsza rozprawa doktorska powstała dzięki pomocy wielu osób, którym jestem za to niezmiernie wdzięczny.

Serdeczne podziękowania składam Promotorom prezentowanej pracy Panu Profesorowi Michel Couprie oraz Panu Profesorowi Dominikowi Sankowskiemu za ogromne wsparcie, inspirację oraz za poniesiony trud opieki naukowej.

Dziękuję również moim Współpracownikom, a także innym osobom, którzy bezpośrednio lub pośrednio przyczynili się do powstania niniejszej rozprawy doktorskiej. Na szczególne podziękowania zasługują Marcin Janaszewski, Laurent Babout, Mariusz Jędrzejczyk oraz Yukiko Kenmochi.

Chciałby również podziękować moim najbliższym – mojej Żonie, moim Rodzicom, Siostrze, i Teściom za troskę, wyrozumiałość i motywację.

Contents

List of Figures	xii
List of Tables	xx
List of Algorithms	xxi
Nomenclature	xxii
1 Introduction	1
1.1 Background	1
1.2 Aims and Objectives	2
1.3 Contributions and Contents	2
1.4 Main Thesis of This Work	3
I State of The Art	5
2 Basic Notions	6
2.1 The Digital Topology Framework	6
2.1.1 Discrete objects	6
2.1.2 Euclidean distance and distance transform	6
2.1.3 Adjacency relations and neighborhood of a point	7
2.1.4 Discrete binary image	8
2.1.5 Connected components, cavities, holes and tunnels	9
2.1.6 Euclidean balls	10
2.1.7 Simple points in 2D	11
2.2 The Cubical Complex Framework	11
2.2.1 Elements in cubical complex framework	12
2.2.2 Cells and cubical complex sets	13
2.2.3 Facets in complexes	13
2.3 From Binary Images to Cubical Complex	13
3 Simplification of Binary Shapes	15
3.1 Skeleton Characteristics	18
3.2 Skeletonization Methodologies	21
3.3 Thinning in The Digital Topology Framework	23
3.3.1 Sequential thinning algorithms	24
3.3.2 Fully parallel thinning algorithms	26

3.3.3	Directional thinning algorithms	28
3.3.4	Subfield thinning algorithms	29
3.4	Thinning in The Cubical Complex Framework	30
3.4.1	Critical kernels based thinning	30
3.4.2	Collapse based thinning	31
4	Preserving Visual Aspects of a Binary Shape	32
4.1	Medial Axis	32
4.2	Medial Axis Filtering	34
4.2.1	Filtering based on medial ball size	35
4.2.2	Filtering based on ball covering	35
4.2.3	λ -medial axis	36
4.2.4	Filtering based on the bisector angle	39
4.2.5	Scale axis	40
4.3	Euclidean Opening Function	41
5	Visual and Topological Correction of Binary Shapes	43
5.1	The Tunnel Closing Algorithm	45
5.1.1	Topological hull	46
5.1.2	Topological hull directed by the distance transform	46
5.1.3	ϵ -closing algorithm	47
5.2	Properties and Limitations of the Tunnel Closing Algorithm	49
5.3	Application of The Tunnel Closing Algorithm	51
6	Extracting Geometrical Properties of Binary Shapes	54
6.1	Discrete Curves	57
6.2	Continuous Tangent Estimators	58
6.3	Discrete Tangent Estimators	60
6.3.1	Local discrete tangent estimators	60
6.3.2	Local adaptive discrete tangent estimators	62
6.3.3	2D λ -maximal segment tangent direction	64
II	Theory and Algorithms	67
7	Medial Axis Filtering with Features at Different Scales	68
7.1	Discrete Adaptive Medial Axis	68
7.2	Discrete Scale Axis	73
7.3	Hierarchical Scale Medial Axis	74
7.3.1	The s -scale filtered Euclidean medial axis	74
7.3.2	Algorithm for computing the hierarchical scale medial axis	77
7.4	Comparison of Medial Axis Filtering Methods	81
7.4.1	The reconstruction error	82
7.4.2	The size of medial axis	85
7.4.3	The border dissimilarity	87
7.4.4	The complexity of the skeleton	90
7.4.5	The multiscale representations	92
7.4.6	Discussions	93

7.5	Conclusions	95
8	Tangent Estimator Along 3D Discrete Curve	97
8.1	3D Discrete Line Segment	97
8.2	3D Tangential Cover Construction	98
8.2.1	Complexity evaluation	98
8.3	Three-dimensional λ -MSTD Estimator	99
8.4	Experiments on Parametric Curves	101
8.4.1	Multigrid convergence	101
8.4.2	Results and discussion	101
III	Verification and Application	111
9	Quantitative Analysis of The Human Airway Trees	112
9.1	Introduction	112
9.2	System Design	114
9.2.1	Airway tree identification	114
9.2.2	Airway tree simplification	116
9.2.3	Generation of the formal tree structure	118
9.2.4	Quantitative analysis	120
10	Three dimensional model of The Human Airway Trees	122
10.1	Basic Model of Bronchial Trees	123
10.2	Extended Algorithm of Bronchial Tree Modeling	127
10.2.1	Geometric deformations of surface model	127
10.2.2	Transformations of the surface model to volumetric model	128
10.2.3	Distortions introduced to in a volumetric model of bronchial tree	128
11	Robust Segmentation of The Human Airway Trees	131
11.1	Airway Tree Segmentation Based on Tunnel Closing	131
11.1.1	Histogram analysis and preliminary wall extraction	131
11.1.2	3D tunnel closing in airway walls	133
11.1.3	Merging and final segmentation	133
11.2	Segmentation Algorithms Comparison	134
12	Robust 3D Skeletonisation of Pulmonary Airway Tree Structures	139
12.1	Skeleton Centricity	140
12.2	Robustness to Noise	142
12.3	Discussion	143
13	Accurate Cross Sections Generation of The Human Airways Trees	150
13.1	Methodology of Tangent Estimators Comparison	150
13.2	Results and Discussion	152
14	Conclusions	156
14.1	Contribution of this thesis	157
	Bibliography	159

List of Figures

2.1	An example of a 2D binary image in (a) and its squared Euclidean distance map in (b).	7
2.2	An example of a 2D shape in (a) and 3D shape in (b).	9
2.3	Hollow torus and two loops which determine two tunnels. The loops γ_1 and γ_2 are generators of the torus fundamental group.	10
2.4	(a): The object X with $ C_8[X] = 2$ and $ C_4[\overline{X}] = 2$. The point p can be removed without changing the number of connected components of X and \overline{X} . Therefore, point p is simple; (b): The point x is not simple $ C_4[\overline{X} \cup \{p\}] = 1$; (c): The point y is not simple $ C_8[X \setminus \{y\}] = 3$; (d): The point z is not simple $ C_8[X \setminus \{z\}] = 1$	11
2.5	An example where removing point x from object X on (a) does not change number of connected component of X and \overline{X} , however a tunnel appear in X on (b). (Chaussard, 2010)	12
2.6	Graphical representations of faces in the cubical complex framework. (Chaussard, 2010)	12
2.7	The square "a" is a facet, as well as the edge "b". The edge "c" is not a facet, as it is included in a square. The vertex "d" is not a facet, as it is included in two edges. (Chaussard, 2010)	13
2.8	Some examples of sets of faces. Let A (resp. B, C, D) be the set of faces represented on a (resp. b, c, d). The set B is not a complex, as two squares miss an edge, and four edges miss a vertex. The set D is neither a complex, but A and C are complexes. Moreover, $A = B^-$. The complex A is pure as all its facets are squares, and the complex C is not pure as some of its facets are squares while some others are edges. We have $D = C^+$. The complexes A and C have a dimension equal to 2. (Chaussard, 2010)	14
2.9	An example of a 2D binary image in (a) and its cubical complex representation in (b).	14
3.1	An example of two 3D objects and corresponding skeletons (in yellow).	15
3.2	An example of object (ellipse) with set of centers of medial balls (in black). In red, medial ball centered at point x	16
3.3	An example of Virtual Colonoscopy. 3D rendering of the human colon reconstructed from CT data set with its skeleton superimposed.	17
3.4	An example of matching. Comparing skeleton of two objects, some correspondence can be found which allow to match similar objects (Sundar et al., 2003).	17
3.5	An example of 3D object decomposed into components. Each component has different color (Chaussard, 2010).	18
3.6	An example of (a) surface and (b) curve-skeleton of the same object.	19

3.7	An example of skeleton of: (a) smooth and (b) noisy object.	20
3.8	An example of the object with its skeletons on different level of hierarchy (Cornea et al., 2005b).	21
3.9	An example of potential field function, visible as vectors, of the zoomed region marked by red box. (Cornea et al., 2005a)	22
3.10	An example of Voronoi diagram (in light gray) of the object (in yellow) in respect to selected set of surface point (red dots). Corresponding skeleton is superimposed (in black).	23
3.11	Various removal orders of simple points lead to different skeletons.	25
3.12	Basic partition of the space into directions. (a): 6-directions; (b): 8-directions; (c): 12-directions. Each number i represents a voxel which has to be a background voxel to allow the deletion of p in the i -th sub-iteration.	28
3.13	Partition of the space into subfields. (a): 2 subfields; (b): 4 subfields; (c): 8 subfields. In i -th sub-iteration only points marked i will be considered.	30
3.14	(a) An object (a complex) and (b) its Critical Kernel (in red) (Bertrand, 2007)	30
3.15	Collapse of complexes - Let A (resp. B, C, D) be the set of faces represented on a (resp. b, c, d). In the complex A , the edge b is only included in the square a : the pair (a, b) is free for A . The edge c is included in two squares of A , the pair (a, c) is therefore not free for A . The complex $B = A \setminus \{a, b\}$, is an elementary collapse of A . In B , the vertex e is included in only one edge: the pair (d, e) is free for B , and $C = B \setminus \{e, g\}$ is an elementary collapse of B . The complex D is an elementary collapse of C , and therefore, D is a collapse of A . (Chaussard, 2010)	31
4.1	(a): a shape (in gray) and its Euclidean medial axis (in black); (b): the same shape, but with small amount of noise added on the contour. The medial axis of the shape (b) is much more complicated than the medial axis of the shape (a).	34
4.2	(a): Example of object (in gray) and its medial axis (in black); (b): Filtered medial axis with RFMA gives poor results; (c): Filtering based on bisector angle, preserve some points close to the border; (d): Filtering based on bisector angle and RFMA as a complementary gives much better results; (e,f,g,h): an ultimate homotopic skeleton of (a,b,c,d resp.) with the constraint of retaining the points of its filtered medial axis.	36
4.3	Medial axis filtering based on ball covering. The medial ball which uncovered area a (in red) is smaller than threshold parameter t will be removed.	36
4.4	An example, of medial axis filtering based on ball covering (in black) in which the method failed due to features at different scale in the input object (in gray). On (a) insufficient filtering, spurious branches exist. On (b) spurious branches still exist but right ear is missing. On (c) one spurious branch still exist but ears were fully removed.	37
4.5	Illustration of the λ -medial axis. Left: Points x, x' and x'' and their respective closest border points. Top right: The λ -medial axis with $\lambda = \epsilon$, a very small positive real number. Bottom right: The λ -medial axis with $\lambda = d(a', b')/2 + \epsilon$	37

4.6	(a): The projection of the point x is the point y : by symmetry, all points of the ribbon would have a singleton as projection, producing an empty λ -medial axis for any $\lambda > 0$; (b): The extended projection of the point x is the set $\{y, a, b, c\}$: by symmetry, the points located near the center of the ribbon would all have an extended projection consisting of four points, leading to a non empty λ -medial axis. (Chaussard, 2010)	38
4.7	Example of the discrete λ -medial axis of the object consists features at different scale. (a): a shape X (in gray); (b): DLMA of X (in white). The medial axis is not sufficiently filtered in the middle of the shape. However, we already start to lose the tail; (c) A more filtered medial axis of X . Now, the middle of the shape is well filtered. However, we lost all information about the tail.	38
4.8	Illustration of the bisector angle. The bisector angle of x is α and the bisector angle of y is β . (Talbot and Vincent, 1992)	39
4.9	(a): a set X and its medial axis (in black); (b): the bisector function Θ_X (dark colors correspond to wide angles); (c): filtered medial axis, based on the values of Θ_X ; (d): detail of the non-filtered and filtered medial axis. (Talbot and Vincent, 1992)	40
4.10	Example of the scale axis transform. (a): The input shape X (in green) and its Euclidean medial axis; (b): The multiplicatively scaled medial balls; (c): The object X_s reconstructed from the scaled medial axis. Note, that reconstructed object is much simpler than original shape; (d): The Euclidean medial axis of X_s ; (e): The scale axis of X . Note that the features at low scale were well preserved.	41
4.11	Illustration of the opening function. (a): an input set of maximal balls; (b): its Euclidean opening function. Bigger balls are on top of smaller ones (Coerjolly, 2010).	42
5.1	A rendering of a crack portion inside a material (material is represented by the background). The crack does not partition the space into inside and outside (Janaszewski et al., 2011). <i>Note:</i> The raw images from where this portion of a crack was segmented were acquired at the ESRF (ID19beamline). Details of the experiment can be found in Babout et al. (2006).	45
5.2	An example of tunnel closing algorithm to the object representing a part of trachea. Red patches correspond to closed tunnels. Notice that on (a) all tunnels were closed. However, on (b) a tunnel which is an anatomical feature is left open.	45
5.3	Examples of topological hulls of the object. (b) and (d) are, respectively, topological hulls of (a) and (c). Notice that topological hull (d) is not centered. (Aktouf et al., 1996)	47
5.4	Application of the 3D tunnels closing algorithm to an 3D object with different ϵ . (a): initial image; (b): $\epsilon = 5$; (c): $\epsilon = 9$. (Aktouf et al., 2002)	48
5.5	Application of the 3D tunnels closing algorithm to tabular object. (a): Initial image with two tunnels marked in yellow; (b, c): Results of working of the tunnel closing algorithms using different priority function. Closing patches are marked in yellow.	49
5.6	Application of the 3D tunnels closing algorithm to object with irrelevant branches close to the tunnel. (a): initial image; (b): the tunnel closing patch is distorted by branches; (c): corrected result using strategy described in Janaszewski et al. (2011)	50

5.7	Example of case when the tunnel closing algorithm cannot be used.	50
5.8	An example of tunnel closing algorithm. (a): the crack. There are three tunnels (a large one and two small ones) inside the crack to be closed; (b): the crack with tunnels closed. Tunnel closing patches, with one voxel thickness, are represented with dark gray color (Janaszewski et al., 2011). <i>Note:</i> The raw images from where this portion of a crack was segmented were acquired at the ESRF (ID19beamline). Details of the experiment can be found in Babout et al. (2006).	51
5.9	An example of tunnel closing algorithm. 3D rendering of an auxetic foam (in light gray) on which the corresponding skeleton is superimposed (dark curves: edges - dark spheres: vertices). An open cell is also closed (depicted by a dark gray circle).	52
5.10	(a): 3D rendering of the aorta (in light gray); (b): Intimal flap inside the aorta on which the intimal tears (holes) are pointed by arrows. (Lohou and Miguel, 2011)	53
6.1	Example of two objects with exactly the same digitalization. (a): Euclidean disk and its digitalization; (b): Euclidean octagon and its digitalization.	55
6.2	Example of presence of noise in the curve. (a): Euclidean straight line and its digitalization; (b, c): Two possible skeletonisation results of (a). Note that (c) can be considered as not "straight".	55
6.3	Example of naive tangent estimator. Each arrow correspond to the local tangent direction.	61
6.4	Example of Matas et al. (1995) local discrete tangents estimator. (a): The curve with points (in yellow) in which the tangent direction where evaluated; (b): The results of tangent estimation (we set parameter $j = 2$). Note that almost straight line where achieved.	61
6.5	Example of a discrete curve and its segmentation into set of maximal segments (red rectangles).	63
6.6	Example of a discrete curve and its tangential cover (red rectangles).	65
6.7	Example of a discrete curve and the pencil of maximal segments (red rectangles) of selected point (in dark gray).	65
6.8	Example of a maximal segment M_i around the point x . The eccentricity $e_i(x)$ depends on the position of x in M_i . Values m_i and L_i denotes the first point and length of the segment M_i , respectively.	66
7.1	(a): a binary shape X (in gray); (b): The filtered λ -medial axis of X (in black) calculated by using algorithm (Chaussard et al., 2010). The medial axis is not sufficient filtered in the middle of the shape. However, we already start to loose the tail; (c): The more filtered medial axis of X . Now, the middle of the shape is well filtered. However, we lost all information about the tail.	69
7.2	(a) and (b): Two graphs representing points distribution for two different shapes according to theirs $\mathcal{F}_X(x)$ and $G_E^X(x)$ values.	70
7.3	Graph representing points distribution according to their $\mathcal{F}_X(x)$ and $G_E^X(x)$ values. We can also distinguish three important sets from filtration point of view (V, I and U respectively). By lines we mark thresholding value according to different λ values. Points which are below such line will be removed during filtering.	71

7.4	Graph representing points distribution according to their $\mathcal{F}_X(x)$ and $G_E^X(x)$ values. By the thick lines, we mark the thresholding value according to equation 7.1 for different $\lambda_\gamma(x)$. Points which are below such lines will be removed during filtering.	72
7.5	(a): a shape X (in gray) and its Euclidean medial axis $MA(X)$ (in black); (b): the multiplicatively 1.1-scaled shape and its 1.1-scaled axis; (c): the multiplicatively 1.3-scaled shape and its 1.3-scaled axis. In (b, c) we can see that scales axes are not subsets of $MA(X)$. In both cases, an additional branch even appears. (Miklos et al., 2010)	74
7.6	(a): a shape X (in gray) and its Euclidean medial axis (in white); (b, c, d): the same shape and its $SFEMA_{1.1}(X)$, $SFEMA_{1.4}(X)$, $SFEMA_{1.6}(X)$, respectively. In all cases the elephant's tail, trunk, tusks and legs were considered as important and were not removed.	76
7.7	(a): a set X (in green) and its $MA(X)$ (in black); (b): the 1.3-scale axis of X ; (c): the 1.3-scale filtered Euclidean medial axis of X	77
7.8	(a): a set X (in green) and its $MA(X)$ (red dots); (b): multiplicatively scaled medial balls. The smaller ball is not fully covered by the bigger one after scaling. In scale axis representation both balls will be preserved. However, the bigger ball includes the smaller one inside set X . Therefore, the smaller ball will not exist in $SFEMA_s(X)$; (c): multiplicatively scaled medial balls. The smaller ball is included in the bigger one. Therefore, it is neither in scale axis nor in $SFEMA_s(X)$	77
7.9	(a): a shape X (in gray) and two maximal balls $B(x_1, r_1), B(x_2, r_2) \in X$; (b): dashed circles represent scaled balls by $s = 1.4$. One can observe that $B^X(x_2, r_2 \cdot 1.4) \subset B^X(x_1, r_1 \cdot 1.4)$, so $(x_2, r_2) \notin SFEMA_{1.4}(X)$; (c): dashed circles represent scaled balls by $s = 1.6$, now $p \in B^X(x_2, r_2 \cdot 1.6)$ and $p \notin B^X(x_1, r_1 \cdot 1.6)$, so $(x_2, r_2) \in SFEMA_{1.6}(X)$, which finally leads to the conclusion that $SFEMA_s(X)$ does not hold hierarchy property.	78
7.10	A sample of the 18 representative classes of shapes of Kimia's database.	81
7.11	A sample of the selected 6 representative classes of shapes of Kimia's database used to illustrate results of our experiments.	82
7.12	Medial axes (in black) superimpose to input object (in grey). Consecutive rows (from left to the right) contain results for DLMA, BFMA, HSMA, RFMA, CFMA and DAMA respectively. Columns contain results for different values of normalised residuals: 0.01, 0.03, 0.05, 0.1 respectively.	83
7.13	Medial axes (in black) superimpose to input object (in grey). Consecutive rows (from left to the right) contain results for DLMA, BFMA, HSMA, RFMA, CFMA and DAMA respectively. Columns contain results for different values of normalised residuals: 0.01, 0.03, 0.05, 0.1 respectively.	84
7.14	Medial axes (in black) superimpose to input object (in grey). Consecutive rows (from left to the right) contain results for DLMA, BFMA, HSMA, RFMA, CFMA and DAMA respectively. Columns contain results for different values of normalised residuals: 0.01, 0.03, 0.05, 0.1 respectively.	85
7.15	Medial axes (in black) superimpose to input object (in grey). Consecutive rows (from left to the right) contain results for DLMA, BFMA, HSMA, RFMA, CFMA and DAMA respectively. Columns contain results for different values of normalised residuals: 0.01, 0.03, 0.05, 0.1 respectively.	86
7.16	Residuals as a function of normalised medial axis size for different filtering criterions. Results generated for 6 selected shapes (see Fig 7.11).	87

7.17	Example of the dissimilarity measure. (a): Input shape (in gray); (b, c): Some part of the shape was removed. Note that the size of removed part is exactly the same. In both cases the number of residuals will be the same, however, dissimilarity will be much higher in case of (c).	88
7.18	Dissimilarity as a function of normalised residuals for different filtering criterions. Results generated for 6 selected shapes (see Fig 7.11).	89
7.19	(a): A skeleton with complexity $c = 1$; (b): Another example of a skeleton with complexity $c = 1$; (c): A skeleton with complexity $c = 3$	90
7.20	(a): a set X (in white); (b) ultimate homotopic skeleton of X ; (c): medial axis of X ; (d): ultimate homotopic skeleton of X constrained by the medial axis of X . In (b,c,d) the original set X appears in gray for comparison. . .	91
7.21	Number of minimum residuals according to the complexity level, calculated for different filtering criterions. Results generated for 6 selected shapes (see Fig 7.11). Each marker (e.g. triangle, circle and square) represents parameters of one filtered medial axis. Lines has been added only to emphasize the trend of measurements.	92
7.22	Residuals as a function of hierarchy levels for different filtering criterions. Results generated for 6 selected shapes (see Fig 7.11).	94
8.1	An example of 3D curve projected on the basic planes O_{XY} , O_{XZ} and O_{YZ} . The curve forms the 3D straight line segment since two projection (O_{XY} , O_{XZ}) are 2D straight line segments.	98
8.2	A 3D digital curve and its tangential cover calculated by Algorithm 10. Three projection on the basic planes O_{XY} , O_{XZ} and O_{YZ} of the curve were also depicted.	100
8.3	A set of 3D parametric curves used for the experiments.	102
8.4	A three different proposition of the λ functions which can be used in λ -MSTD estimator.	103
8.5	An evolution of RMSE between real and estimated tangent with increasing resolution for different definition of λ function.	104
8.6	An evolution of maximal absolute error between real and estimated tangent with increasing resolution for different definition of λ function.	105
8.7	The speed of RMSE convergence with increasing resolution.	106
8.8	An expected (theoretical) and estimated tangent direction along 3D circle at fixed resolution, for three axes x, y and z , respectively.	107
8.9	An expected (theoretical) and estimated tangent direction along 3D helix at fixed resolution, for three axes x, y and z , respectively.	108
8.10	An expected (theoretical) and estimated tangent direction along 3D Viviani's curve at fixed resolution, for three axes x, y and z , respectively. . .	109
8.11	An expected (theoretical) and estimated tangent direction along 3D Trefoil's Knot curve at fixed resolution, for three axes x, y and z , respectively.	110
9.1	The human airway tree. (Gray, 1918)	113
9.2	Consecutive steps of the system for the quantitative analysis of the human bronchial trees. (a): A CT dataset; (b): segmented airway tree from (a); (c): Airway tree decomposed into set of segments (branches); (d) The quantitative measurements of the bronchial properties at the point of interest.	115

9.3	Example of a severe segmentation leak. Result of segmentation with the standard region growing algorithm.	116
9.4	Example of the skeleton on (b) of the human airway tree on (a).	117
9.5	(a): The structure of airway tree represented as a graph. In the graph nodes correspond to branch-points, and edges correspond to segments; (b): Cylindrical partitions in a human airway tree. The colors corresponds to labels assigned to each point of the tree according to the segment. (Palagyi et al., 2006)	119
9.6	Airway tree with assigned labels. Labels refer to segments, but are assigned to terminating branch-point of respective segment. (Boyden, 1955).	119
10.1	Single bifurcation. Dark grey color represents bifurcation plane, light grey color represents volume division plane. Both planes are spread out to the borders of the parent region. The normal of the bifurcation plane has been marked by arrow. This normal has been anchored in the bifurcation point.	125
10.2	The bounding surface which limits volume where the tree is generated. (a) View from the top; (b) Side view.	125
10.3	Outer surface of a volume supplied by the left daughter of trachea.	126
10.4	Part of the model with bended branches.	128
10.5	Exemplary results of TopoEDEN when applied to a bronchial tree; a): TopoEDEN which erodes input object; b): TopoEDEN which dilate input object.	129
10.6	The effect obtained by using TopoEDEN. The tree is dilated and smoothed by ASFT.	129
10.7	Bronchial trees phantoms generated with different models; a): bronchial tree obtained by segmentation of tomographic image; b): tree generated with the use of basic model; c): extended model with bended branches; d): extended model with bended branches and noise.	130
11.1	Airway tree wall extraction. (a): One 2D slice obtained from a 3D CT dataset; (b): Zoomed fragment of the trachea; (c): Extracted airway walls; (d): An input image (b) merged with the image (c).	132
11.2	A typical histogram generated from a 3D chest image. Marked ranges correspond to air, internal part of bronchial walls, soft tissue and blood.	132
11.3	Tunnel closing in the extracted bronchial walls. (a): The 2D binary image which represent extracted bronchial walls; (b): The 3D reconstruction of (a) with well visible tunnels. After application of the tunnel closing algorithm all tunnels were closed (red patches).	133
11.4	3D visualization of segmentation results. Left column - proposed method, right column - the region growing procedure (Mori and et al., 1995). In each row the same CT dataset were used.	136
11.5	3D visualization of segmentation results by proposed method.	137
11.6	An example of skeletonisation results (the same algorithm) on the tree segmented from the same CT dataset. (a): The skeleton of the tree segmented based on hole closing; (b): The skeleton of the tree segmented using simple region growing approach. Note that in this case the skeleton is much more complicated (especially in top of the trachea) even if the tree is segmented reasonably well.	138
12.1	The set of bronchial tree phantoms used for the experiments	140

12.2	The skeletonisation results of the basic phantom.	145
12.3	The skeletonisation results of the extended phantom.	146
12.4	The skeletonisation results of the extended phantom with 5% of noise.	147
12.5	The skeletonisation results of the extended phantom with 10% of noise followed by ASFT filtering.	148
12.6	The TCC skeletonisation results of the four segmented airways tree from the real CT dataset.	149
13.1	The 3D reconstruction of one of bronchial tree models with selected path (black curve) used for experiments	151
13.2	A comparison between reference value from the phantom and estimated area using different tangent estimators.	154
13.3	A comparison between reference value from the phantom and estimated area using different tangent estimators.	155

List of Tables

11.1 Fraction (in %) of average number of extracted branches at different levels of the airway tree. Results obtained using the proposed method (HC) and region growing method (RGA) with manually selected optimal parameters on the 30 cases with standard quality (low dose) and 30 cases with extended quality (high dose).	135
12.1 Average dissimilarity between DAMA of X and skeleton of X calculated for basic phantoms (10 phantoms in each class).	141
12.2 Average dissimilarity between DAMA of X and skeleton of X calculated for extended phantoms (10 phantoms in each class).	142
12.3 Average number of the spurious skeleton branches calculated for basic phantoms (10 phantoms in each class).	142
12.4 Average number of the spurious skeleton branches calculated for extended phantoms (10 phantoms in each class).	143
13.1 Average RMSE between reference lumen diameter and evaluated one calculated for basic and extended phantoms (10 phantoms in each class).	152

List of Algorithms

1	Basic Sequential Thinning(Input X Output Y)	25
2	Basic Thinning with Priority(Input X, W, D Output Y)	26
3	Generic Fully Parallel Thinning(Input X Output Y)	27
4	Generic n -Subiteration Thinning(Input X Output Y)	29
5	Naive Euclidean Opening Function(Input X Output Y)	42
6	Tunnel Closing Algorithm(Input X, B, Ψ, ϵ Output Y)	48
7	DiscreteScaleAxis(Input X, s Output $SAT_s(X)$)	73
8	SFEMA(Input X, s Output H)	75
9	ScaleMap1(Input $MA(X)$ Output Ψ_X)	78
10	ScaleMap(Input $MA(X)$ Output Ψ_X)	80
11	Tangential Cover (Input C ; Output \mathcal{M})	99

Nomenclature

\mathbb{Z}	The set of integers
\mathbb{N}	The set of positive integers
\mathbb{R}	The set of reals
\mathbb{Z}^n	The discrete grid of dimension n
X	A discrete object or shape ($X \subset \mathbb{Z}^n$)
$ X $	The cardinality of X
\bar{X}	The complementary of X ($\bar{X} = \mathbb{Z}^n \setminus X$)
$d(x, y)$	The Euclidean distance between x and y ($x, y \in \mathbb{Z}^n$)
$d(x, Y)$	The Euclidean distance between x and set Y ($x \in \mathbb{Z}^n, Y \subset \mathbb{Z}^n$)
D_X	The Euclidean distance map of X
D_X^2	The squared Euclidean distance map of X
$\mathcal{N}(x)$	The direct neighborhood of x
$\mathcal{N}^*(x)$	The strict direct neighborhood of x ($\mathcal{N}^*(x) = \mathcal{N}(x) \setminus \{x\}$)
$\Gamma_k(x)$	The k -neighborhood of x
$\Gamma_k^*(x)$	The strict k -neighborhood of x ($\Gamma_k^*(x) = \Gamma_k(x) \setminus \{x\}$)
$\mathcal{P} = (\mathbb{Z}^n, k, \bar{k}, X)$	A binary image (or discrete binary image)
k	An adjacency used for X
\bar{k}	An adjacency used for complementary of X
\mathcal{S}	A sequence of points
C_k	k -connected component of X ($C_k \subseteq X$)
$C_k[X]$	The set of all k -connected components of X
$C_k[x, X]$	The set of all k -connected components of X which are k -neighbors of x
$T_{26}(x, X)$	The number of 26-connected components of $(X \cap \Gamma_{26}^*(x))$
$T_6(x, X)$	The number of 6-connected components of $(X \cap \Gamma_6^*(x))$
$B(x, r)$	The Euclidean ball of radius r centered on x ($X \subset \mathbb{Z}^n, r \in \mathbb{R}$)
$B^<(x, r)$	The strict Euclidean ball of radius r centered on x
$B_X(x, r)$	The intersection of ball $B(x, r)$ with X
$R(S)$	The radius of the smallest ball enclosing S
$MA(X)$	The medial axis of X
$RFMA(X, t)$	The filtered medial axis of X based on ball radius
$CFMA(X, t)$	The filtered medial axis of X based on ball covering
$BFMA(X, \alpha)$	The filtered medial axis of X based on bisector angle

$DLMA(X, \lambda)$	The discrete λ -medial axis of X
$DAMA(X, \gamma)$	The discrete adaptive medial axis of X
$HSMA_s(X)$	The hierarchical scale medial axis of X
$SAT_s(X)$	The s -scale axis transform of X
$\Psi_X(x, r)$	The scale map of X
$\Pi_S(x)$	The projection of x on S
Θ_X	The bisector function of X
G_E^X	The Euclidean opening function of X
$\mathcal{F}_X(x)$	The λ function of X
$\mathcal{D}_2(a, b, \mu, e)$	The 2D digital line
$\mathcal{D}_3(a, b, c, \mu, \mu', e, e')$	The 3D digital line
\mathcal{M}	The tangential cover
$P(x)$	The pencil of maximal segments
$e_i(x)$	The eccentricity of a point x
$\vec{t}(x)$	Tangent vector at point x
$REDT_X(Y)$	The reverse Euclidean distance transform of Y
$R_X(Y)$	The normalised residuals of Y
$NS_X(Y)$	The normalised medial axis size
$R_X^c(Y)$	The minimum normalised residuals of Y at complexity c
Y_c	The filtered medial axis with complexity c
$d_D(X Y)$	The Dubuisson and Jain's dissimilarity

Chapter 1

Introduction

1.1 Background

Computed tomography is a very useful technique which allow non-invasive diagnosis in many applications for example is used with success in industry and medicine. The rapid development of the x-ray computed tomography started in the 70s of XX century and continues today.

Computed Tomography technique produce stack of two-dimensional slices which combined together can represent three-dimensional structures. Such images needs to be analyzed by specialists. With recent advances in CT technology, slice thickness can now be much less than 1mm which make a data sets to process very large. Manual analysis of the interesting three-dimensional anatomical structures can be tedious and extremely time-consuming, or even impossible due its complexity. Therefore, manual analysis for routine clinical use is impractical and cause strong demand from the medical community to develop new methods which allows to effectively process CT data.

In the 60s, it became realistic that computers could be used in order to automatically process, visualize and analyze data from digital images. Therefore, using this technique as a tool to analyze CT data sets was natural. As addition developing of three-dimensional computer graphics opened a whole set of new possibilities and makes this task much more efficient. It allowed to consider anatomical structures not only as a set of two-dimensional images but more naturally as a three-dimensional structures. Which will not only speed up the diagnostic process but also allow to propose many new methods to non-invasive diagnosis purposes like for example virtual endoscopy techniques.

However, to process such amount of data we need use non-trivial algorithms which needs to be efficient and robust enough to deal with images which can be corrupted for example by presence of noise. Algorithms should allow to extract interesting objects, analyze its properties, make some quantitative measurements or makes its visualization. These

steps consist of problems like object simplification, detection, decompositions, evaluation of its properties, and many more. Based on the discrete nature of the CT data a lot of these problems can be solved based on discrete topology and geometry theories and concepts. They provide tools and algorithms which can be effectively used to digital image processing. This is active research field used with success in many practical application.

1.2 Aims and Objectives

This thesis is initially aimed at studying and developing discrete geometry and topology algorithms suitable for use in the problem of quantitative analysis of the human airway trees based on computed tomography images. We look for methods which are reliable, robust and fast enough to effectively process large amount of data. Moreover, methods should gives results with high accuracy and which can be further used in clinical applications to help doctors in diagnostic process of many pulmonary disorders.

Therefore, the final objective of this thesis is a thorough review of the discrete and geometry algorithms and theirs desirable properties and proposition of a set of methods which used together can allow doctors to obtain precise quantitative information about particulars parts of the bronchial tree especially its wall thickness and diameter of the airway lumen.

To achieve this objective, all along the chapters of this work, we study well known discrete methods as well as a new methods which can be useful at different stages of the quantitative analysis process. To test algorithms we use computer phantoms of the airway trees and many CT scans of the real patients both healthy and affected by different pulmonary disease. As addition, examples and results will be also presented on other types of discrete datasets, in order to prove that the methods exposed here are not aimed at only one domain.

All computed tomography datasets are provided thanks to interdisciplinary cooperation with Department of Radiology and Diagnostic Imaging, Medical University of Lodz. Medical doctors are also responsible for further validate usefulness of developed methods in clinical applications.

1.3 Contributions and Contents

In this thesis, we propose digital topology and geometry algorithms suitable for use in many practical applications, especially, in the problem of quantitative analysis of the human airway trees based on computed tomography images.

In order to reach this goal, we have done some contributions in different fields of research:

Digital Topology - As the skeleton of the airway tree is a basis for many different task in bronchial trees analysis, like for instance, tree decomposition and branch labeling or cross section computation, we have worked on comparison skeletonisation algorithms based on desirable set of properties. We use quantitative comparison based on properties of medial axes of the tree. However, using the discrete medial axis directly is hard due several problems. Therefore, we have worked on filtering methods which are able to effectively prune medial axes points in object with features at different scales. We have also proposed a new methodology for fair comparison between different filtering methods.

Digital Geometry - As the accuracy of the quantitative measurements relay on the reliable perpendicular cross sections evaluation in each point of the bronchial tree, we proposed a new tangent estimators along three-dimensional discrete curves. Tangent vector is used as normal vector to the plane which used to define perpendicular cross sections.

Medical Image Processing - We also study the tunnel closing algorithm as a tool to repair, damaged by acquisition errors, CT datasets and we propose a new strategy for segmenting airway tree from CT datasets. We also proposed an artificial model of the bronchial tree which can be used to validate developed algorithms.

The structure of this manuscript is in three main parts: the first one introduces the state of the art of the digital topology and geometry algorithms which are essential in quantitative analysis process like tunnel closing, skeletonization, medial axes and tangent estimators. The second part consist of a proposition and theory of a new methods for solving particular problems like discrete medial axis filtering and three-dimensional tangent estimation. In this part we also verify proposed methods on the set of artificial objects. The last part of the work consist of state of the art about quantitative analysis of the human airway trees based on computed tomography. This part also consist of a proposition of a new airway tree segmentation algorithm and method for building three-dimensional model of the tree. Finally in this part we shows experiments and results performed on real tomographic data and computer phantoms to validate and compare all implemented methods in application to quantitative analysis of the human airway tress.

1.4 Main Thesis of This Work

In this work we set following thesis:

Main These:

Algorithms build on top of discrete topology and geometry like skeletons, medial axes, tunnel closing algorithms and tangent estimators can be used in many practical application which are based on image processing and analysing techniques. They still have many

limitations, however, the recently proposed methods overcome many of them which allow to use such tools effectively and fully automatically. Therefore, the discrete topology and geometry algorithms can be used in each step of quantitative human airway trees analysis based on computed tomography.

Supplementary These 1:

New medial axis filtering algorithms proposed in this manuscript allow to filter effectively medial axes of objects which consist features at different scales. Therefore, they can be used to make filtered medial axes of the human airway trees where the branches are with different diameter.

Supplementary These 2:

Skeletonisation algorithms based on thinning are sensitive to presence of noise in the input object which can cause skeletons to have some spurious branches or to be not well centered. Recently proposed methods based on cubical complexes are more robust to such kind of distortions. Therefore, they can be used effectively to generate the skeletons of the human airway trees with sufficient accuracy.

Supplementary These 3:

The new efficient tangent estimation algorithm along three-dimensional discrete curves proposed in this manuscript gives accuracy comparable to the continuous methods, without needs to set any parameter. Its accuracy will be improved with increasing resolution. Therefore, it can be used to define local direction of the airway tree and to generate perpendicular cross sections in each point of interest.

Part I

State of The Art

Chapter 2

Basic Notions

2.1 The Digital Topology Framework

In this section we give some basic notions of the main concepts used in the digital topology framework, necessary to understand the following parts of this work. We based on definitions presented in [Kong and Rosenfeld \(1989\)](#); [Bertrand \(1994\)](#); [Chassery and Montanvert \(1991\)](#).

2.1.1 Discrete objects

We denote by \mathbb{Z} the set of integers, by \mathbb{N} the set of positive integers, by \mathbb{R} the set of reals. The set \mathbb{Z}^n , where $n \in \mathbb{N}$, is called the discrete grid or space of dimension n . In this work we focus, mostly, on two-and three-dimensional cases, thus $n = 2$ or 3 . In case of $n = 2$ an element is called *pixel*, in case of $n = 3$ it is called *voxel*, and more generally, a *point*. Each point of \mathbb{Z}^n is defined by an n -tuple (x_1, \dots, x_n) , where $x_i \in \mathbb{Z}$, which represents its coordinates in discrete space. A subset X of \mathbb{Z}^n is called a *discrete object* of \mathbb{Z}^n , for simplicity we also called it shortly an *object* or a *shape*. The number of points in X (i.e. the *cardinality*) of X is denoted by $|X|$. For a given object $X \subset \mathbb{Z}^n$, we denote by \bar{X} the *complementary* of X and we define it by, $\bar{X} = \{y \in \mathbb{Z}^n \mid y \notin X\}$, we also call it the *background*. Each point $x \in X$ is also called a *black point* or an *object point* and each point $y \in \bar{X}$ is also called a *white point* or a *background point*.

2.1.2 Euclidean distance and distance transform

In the discrete space, one can find several definitions of distance ([Borgefors, 1984](#); [Chassery and Montanvert, 1991](#)). In this work we based on the exact Euclidean distance which is defined as follows.

Let $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$, where $x, y \in \mathbb{Z}^n$, we denote by d the function $d : \mathbb{Z}^n \times \mathbb{Z}^n \rightarrow \mathbb{R}$ which associates to x and y the (*Euclidean*) distance between x and y , where d is defined by:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.1)$$

Let $x \in \mathbb{Z}^n$ and $Y \subseteq \mathbb{Z}^n$, we denote by $d(x, Y)$ the (*Euclidean*) distance between x and the set Y , that is, $d(x, Y) = \min_{y \in Y} \{d(x, y)\}$. Given $X, Y \subseteq \mathbb{Z}^n$, we denote by $d(X, Y)$ the Euclidean distance between X and the set Y , that is, $d(X, Y) = \min_{x \in X} \{d(x, Y)\}$.

Let $X \subseteq \mathbb{Z}^n$, we denote by D_X the map from \mathbb{Z}^n to \mathbb{R} which associates, to each point x of \mathbb{Z}^n , the value $D_X(x) = d(x, \overline{X})$. The map D_X is called the (*Euclidean*) distance map of X or the (*Euclidean*) distance transform of X (see an example in Fig.2.1). In the same way we can define the *square of the (Euclidean distance)* denoted by d^2 and corresponding *squared (Euclidean) distance transform* denoted by D_X^2 .

In the literature different methods have been proposed to compute the exact squared Euclidean distance transform of an object X in linear time with respect to the size of the input, for example methods proposed by [Saito and Toriwaki \(1994\)](#); [Hirata \(1996\)](#); [Meijster et al. \(2000\)](#).

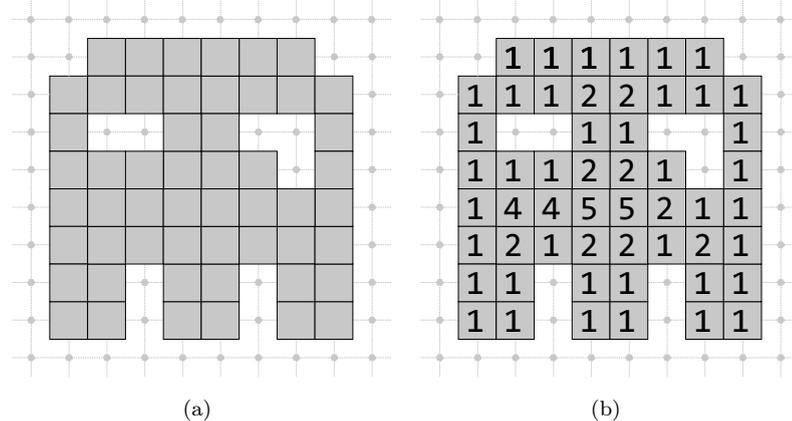


FIGURE 2.1: An example of a 2D binary image in (a) and its squared Euclidean distance map in (b).

2.1.3 Adjacency relations and neighborhood of a point

Now, we can define different adjacency relations between points in \mathbb{Z}^n . Let $x \in \mathbb{Z}^n$, the *direct neighborhood* of x , denoted by $\mathcal{N}(x)$ is the set of points of the discrete grid that are closest to x , defined by $\mathcal{N}(x) = \{y \in \mathbb{Z}^n \mid d(x, y) \leq 1\}$. The direct neighborhood comprises $2n + 1$ points.

We denote by Γ_k an *adjacency relation*, with $k = 4, 8$ in case of \mathbb{Z}^2 and $k = 6, 18, 26$ in case of \mathbb{Z}^3 . The adjacency relation defines which points are considered as neighbors of a given point x .

In the 2D case,

$$\Gamma_4(x) = \{y \in \mathbb{Z}^2 \mid d(x, y) \leq 1\} \quad (2.2)$$

$$\Gamma_8(x) = \{y \in \mathbb{Z}^2 \mid d(x, y) \leq \sqrt{2}\} \quad (2.3)$$

In the 3D case,

$$\Gamma_6(x) = \{y \in \mathbb{Z}^3 \mid d(x, y) \leq 1\} \quad (2.4)$$

$$\Gamma_{18}(x) = \{y \in \mathbb{Z}^3 \mid d(x, y) \leq \sqrt{2}\} \quad (2.5)$$

$$\Gamma_{26}(x) = \{y \in \mathbb{Z}^3 \mid d(x, y) \leq \sqrt{3}\} \quad (2.6)$$

A point y is k -neighbor of a point x , if $y \in \Gamma_k(x)$. It is k -neighbor of a subset X of \mathbb{Z}^n if it is a k -neighbor of at least one point of X . We denote by $\Gamma_k^*(x)$ the *strict k -neighborhood* of a point x , defined by, $\Gamma_k^*(x) = \Gamma_k(x) \setminus \{x\}$. The same way we define $\mathcal{N}^*(x) = \mathcal{N}(x) \setminus \{x\}$, called the *strict direct neighborhood*. Notice that $\mathcal{N}(x) = \Gamma_4(x)$ in 2D, and $\mathcal{N}(x) = \Gamma_6(x)$ in 3D.

In order to have coherent topological properties in object X and in its complementary \bar{X} , at the same time, we need to use different adjacency relations for both sets (Kong and Rosenfeld, 1989). That is, in case of \mathbb{Z}^2 when considering an object with the 8-adjacency relation its complementary should be considered with the 4-adjacency relation (and vice-versa). In \mathbb{Z}^3 when considering an object with the 26-adjacency relation its complementary should be considered with the 6-adjacency relation (and vice-versa). In the sequel of this work we assume that the adjacency relation pair (k, \bar{k}) has been chosen in case of \mathbb{Z}^2 equal to $(8, 4)$ and equal to $(26, 6)$ in case of \mathbb{Z}^3 . We do not write the subscript k, \bar{k} , unless necessary.

Let $X \subset \mathbb{Z}^n$. If a black point $x \in X$ has no white point as a neighbor, that is $\Gamma_{\bar{k}}(x) \cap \bar{X} = \emptyset$, it is called an *inside point*. Otherwise, that is, $\Gamma_{\bar{k}}(x) \cap \bar{X} \neq \emptyset$, it is called a *border point*. A background point $y \in \bar{X}$ is called an *outside point* if all its neighbors are background voxels, that is, $\Gamma_{\bar{k}}(y) \cap X = \emptyset$.

2.1.4 Discrete binary image

Taking all above into consideration a *binary image* (or *discrete binary image*), is denoted by \mathcal{P} and is defined as a quadruple $\mathcal{P} = (\mathbb{Z}^n, k, \bar{k}, X)$ where $X \subset \mathbb{Z}^n$, and where k and \bar{k} specify the adjacency relations used for the object X and its complementary, respectively, according to rules described above.

2.1.5 Connected components, cavities, holes and tunnels

Given $x, y \in \mathbb{Z}^n$, we define a k -path between x and y as a sequence of points $\mathcal{S} = (p_1, \dots, p_j)$ such that $x = p_1$, $y = p_j$ and, for each $i \in [1, j - 1]$, p_i and p_{i+1} are k -neighbors. If $p_1 = p_j$ then a k -path is a *closed k -path*. We say that a set $X \subset \mathbb{Z}^n$ is k -connected if, for each $x, y \in X$, there exist a k -path between x and y which entirely lying inside X . A non-empty subset $C_k \subseteq X$ is called a k -connected component of X if it is k -connected and if it is not strictly included in any other k -connected subset of X . The set composed of all k -connected components of X (respectively \bar{X}) is denoted by $C_k[X]$ (respectively $C_k[\bar{X}]$). The set composed of all k -connected components of X which are k -neighbors of a point x is denoted by $C_k[x, X]$.

In case of \mathbb{Z}^2 and \mathbb{Z}^3 any finite k -connected component of the background is called a *hole* (see Fig.2.2(a)) and a *cavity* (see Fig.2.2(b)), respectively.

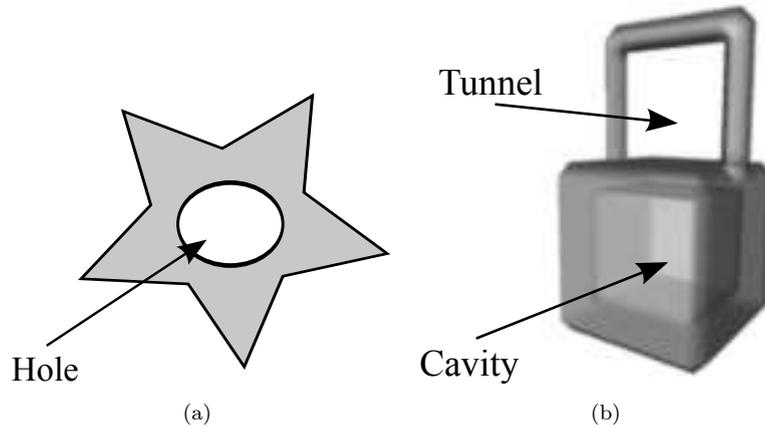


FIGURE 2.2: An example of a 2D shape in (a) and 3D shape in (b).

A *tunnel*, sometimes called a *3D hole*, in an object $X \subset \mathbb{Z}^3$ is detected whenever there is a loop in X which cannot be transformed into a single point by a sequence of elementary homotopic local deformations inside X (Kong and Rosenfeld, 1989). Figure 2.3 shows a hollow torus with two different loops which cannot be iteratively transformed into a single point.

Formal definition of the homotopic deformation introduced by Kong and Rosenfeld (1989) is as follows: Let $X \subseteq \mathbb{Z}^3$ and let $p \in X$ be a point, called the *base point*. Let γ and γ' be two closed k -paths composed of points of X and which have p as origin. We say that γ' is an *elementary k -deformation* of γ , or $\gamma \sim \gamma'$, if there are two k -paths π_1, π_2 , and two non-empty k -paths π, π' , such that γ and γ' are of the form $\gamma = \pi_1\pi\pi_2$, $\gamma' = \pi_1\pi'\pi_2$, and such that all points of π and π' are included in a little portion P of \mathbb{Z}^3 :

- for $k = 6$, P is a unit square (a 2×2 square);
- for $k = 26$, P is a unit cube (a $2 \times 2 \times 2$ cube).

We say that γ' is an k -deformation of γ or $\gamma \simeq \gamma'$ if there is a sequence of closed k -paths $\gamma_0 \dots \gamma_n$ such that $\gamma = \gamma_0$, $\gamma' = \gamma_n$ and $\gamma_{i-1} \sim \gamma_i$ for $i = 1 \dots n$.

Let $\gamma = px_0 \dots x_i p$ and $\gamma' = px'_0 \dots x'_j p$ be two closed k -paths composed of points from X and which have p as origin. The *product* of γ and γ' is the closed k -path $px_0 \dots x_i px'_0 \dots x'_j p$ obtained by concatenating γ and γ' .

Let us consider the classes of equivalence of the closed k -paths with origin p under the relation \simeq . We define the product of two such classes as the equivalence class of the product of two closed k -paths corresponding to the classes.

Under the product operation, these classes constitute a group $\Pi_k(p, X)$ which is the *fundamental group* (or Poincaré group) with base point p . As in the continuous spaces, the fundamental group reflects the structure of the tunnels in X . For example, the fundamental group of a hollow torus is a free abelian group on two generators (see Fig. 2.3). Note that, if p and q belong to the same k -connected component of X , then $\Pi_k(p, X)$ is isomorphic to $\Pi_k(q, X)$.

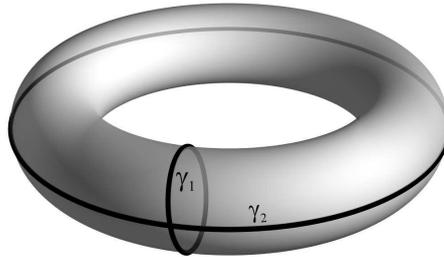


FIGURE 2.3: Hollow torus and two loops which determine two tunnels. The loops γ_1 and γ_2 are generators of the torus fundamental group.

2.1.6 Euclidean balls

Let $x \in \mathbb{Z}^n$ and $r \in \mathbb{R}$, we denote by $B(x, r)$ the (*Euclidean*) ball of radius r centered on x , defined by $B(x, r) = \{y \in \mathbb{Z}^n \mid d(x, y) \leq r\}$. We also define the *strict (Euclidean) ball of radius r centered on x* as the set $B^<(x, r) = \{y \in \mathbb{Z}^n \mid d(x, y) < r\}$.

Notice that, for any point x in X , the value $D_X(x)$ is precisely the radius of a strict ball centered on x and included in X , which is not included in any other ball centered on x and included in X .

Let $x \in X$ and $r \in \mathbb{R}$, we denote by $B_X(x, r)$ the *intersection of ball $B(x, r)$ with X* , defined by $B_X(x, r) = \{y \in X \mid d(x, y) \leq r\}$.

2.1.7 Simple points in 2D

Now we can introduce notions which are fundamental to the definition of topology preserving transformations in discrete spaces.

Given $p \in X$, intuitively, a point p is a *simple point* of the object X , if it is "inessential" to the topology of X (Rosenfeld, 1970), that is, we can remove p from X without changing the topology of X . Notice that, topology of the object depends on the chosen adjacency, therefore if we consider k -connected object we need to use k -simple points.

As reported in Lam et al. (1992) studies of two dimensional deletable points started in the 60s in A.E. Brain and Munson (1965); Golay (1969) where, in order to decide if a point p is simple, the number of connected components of whole object X (respectively \bar{X}), need to be evaluated. Then, a point p is simple if we have the same number of connected components of X (respectively \bar{X}), before and after removing a point p from X , that is, $|C_k[X]| = |C_k[X \setminus \{p\}]|$ and $|C_{\bar{k}}[\bar{X}]| = |C_{\bar{k}}[\bar{X} \cup \{p\}]|$ (see Fig.2.4).

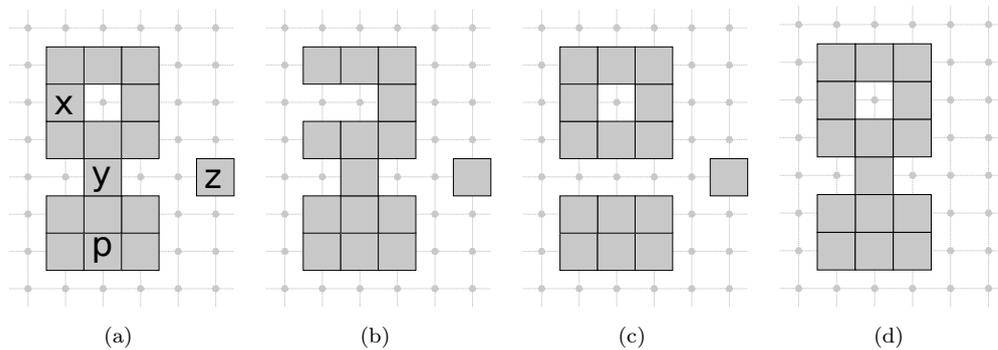


FIGURE 2.4: (a): The object X with $|C_8[X]| = 2$ and $|C_4[\bar{X}]| = 2$. The point p can be removed without changing the number of connected components of X and \bar{X} . Therefore, point p is simple; (b): The point x is not simple $|C_4[\bar{X} \cup \{p\}]| = 1$; (c): The point y is not simple $|C_8[X \setminus \{y\}]| = 3$; (d): The point z is not simple $|C_8[X \setminus \{z\}]| = 1$.

In three dimensional objects such strategy is not sufficient to decide if a point p is simple or not. The reason is in 3D removing a point from the object may also change the structure of tunnels in X . Therefore, it is possible to find an example where after point removal we does not change number of connected component of the object nor its complementary, however topology of the object is changed (see Fig.2.5).

2.2 The Cubical Complex Framework

In the following section we introduce the basis notions of the cubical complex framework. Abstract (cubical) complexes have been studied in particular by Kovalevsky (1989) in order to provide a sound topological basis for image analysis. We based on definition presented in Bertrand and Couprie (2008).

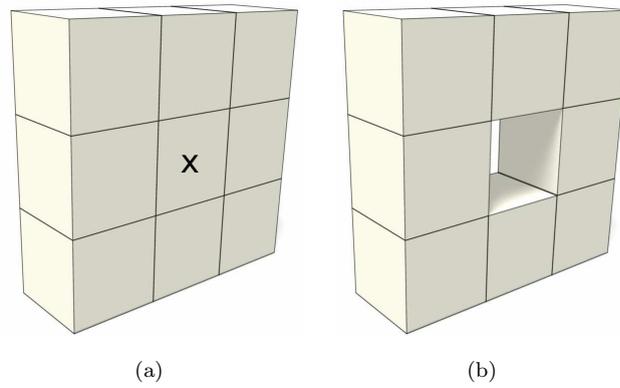


FIGURE 2.5: An example where removing point x from object X on (a) does not change number of connected component of X and \bar{X} , however a tunnel appear in X on (b). (Chaussard, 2010)

2.2.1 Elements in cubical complex framework

In the cubical complex framework we use a set of basic elements having various dimensions (e.g. vertices, edges, squares or cubes). Each such element is called a face and is defined as follows.

Definition 2.1 (Bertrand and Couprie (2008)). Let \mathbb{Z} be the set of integers, we consider the family of sets \mathbb{F}_0^1 and \mathbb{F}_1^1 , such that $\mathbb{F}_0^1 = \{\{a\} \mid a \in \mathbb{Z}\}$ and $\mathbb{F}_1^1 = \{\{a, a+1\} \mid a \in \mathbb{Z}\}$. Any subset f of \mathbb{Z}^n such that f is the cartesian product of m elements of \mathbb{F}_1^1 and $(n-m)$ elements of \mathbb{F}_0^1 is called a *face* or a *m-face* of \mathbb{Z}^n , m is the dimension of f , we write $\dim(f) = m$.

We denote by \mathbb{F}^n the set composed of all faces in \mathbb{Z}^n . Given $m \in \{0, \dots, n\}$, we denote by \mathbb{F}_m^n the set composed of all m -faces in \mathbb{Z}^n . When we consider a 3D cubical complex, that is, \mathbb{F}^3 , then a 0-face is called a *vertex*, a 1-face is an *edge*, a 2-face is a *square*, and a 3-face is a *cube* (see Fig.2.6, where graphical representation of these four kinds of faces are depicted).

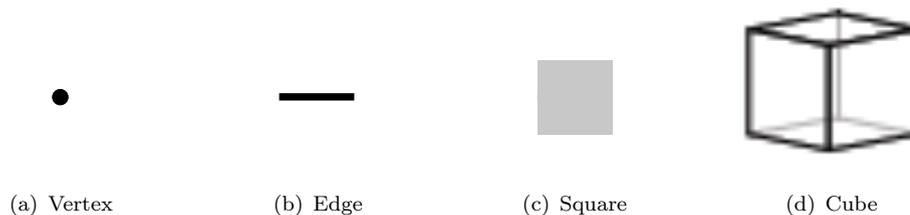


FIGURE 2.6: Graphical representations of faces in the cubical complex framework. (Chaussard, 2010)

Let $f \in \mathbb{F}^n$. We set $\hat{f} = \{g \in \mathbb{F}^n \mid g \subseteq f\}$, and $\hat{f}^* = \hat{f} \setminus \{f\}$. Any element of \hat{f} is a *face of f*, and any element of \hat{f}^* is a *proper face of f*. We call *star of f* the set

$\check{f} = \{g \in \mathbb{F}^n \mid f \subseteq g\}$, and we write $\check{f}^* = \check{f} \setminus \{f\}$: any element of \check{f} is a *coface* of f . It is plain that $g \in \hat{f}$ iff $f \in \check{g}$.

2.2.2 Cells and cubical complex sets

A set \mathcal{X} of faces in \mathbb{F}^n is a *cell*, or *m-cell*, if there exists an m -face $f \in \mathcal{X}$ such that $\mathcal{X} = \hat{f}$. The *closure* of a set of faces \mathcal{X} is the set $\mathcal{X}^- = \cup\{\hat{f} \mid f \in \mathcal{X}\}$. The set $\overline{\mathcal{X}}$ is $\mathbb{F}^n \setminus \mathcal{X}$.

Definition 2.2 (Bertrand and Couprie (2008)). A finite set \mathcal{X} of faces in \mathbb{F}^n is a *cubical complex* if $\mathcal{X} = \mathcal{X}^-$, and we write $\mathcal{X} \preceq \mathbb{F}^n$. Any subset Y of \mathcal{X} which is also a complex is a *subcomplex* of \mathcal{X} , and we write $Y \preceq \mathcal{X}$.

2.2.3 Facets in complexes

A face $f \in \mathcal{X}$ is a *facet* of \mathcal{X} if f is not a proper face of any face of \mathcal{X} (see Fig. 2.7). We denote by \mathcal{X}^+ the set composed of all facets of \mathcal{X} . A complex \mathcal{X} is *pure* if all its facets have the same dimension (see Fig. 2.8). The *dimension* of \mathcal{X} is $\dim(\mathcal{X}) = \max\{\dim(f) \mid f \in \mathcal{X}\}$. If $\dim(\mathcal{X}) = d$, then we say that \mathcal{X} is a *d-complex*.

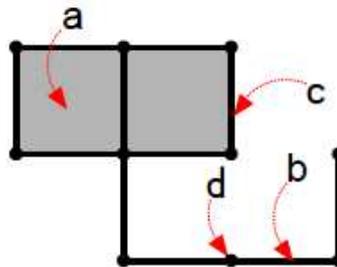


FIGURE 2.7: The square "a" is a facet, as well as the edge "b". The edge "c" is not a facet, as it is included in a square. The vertex "d" is not a facet, as it is included in two edges. (Chaussard, 2010)

2.3 From Binary Images to Cubical Complex

In this work we focus on images which are defined in discrete topology framework which is the most common representation of an image in the field of image processing. Therefore, to use operators which are designed to work with cubical complexes, we need to define a way to transpose a binary image to cubical complex framework, that is, two dimensional binary images are seen as pure 2-complexes (see Fig.2.9(b)) and three dimensional binary images as pure 3-complexes. To make such transposition we use the same strategy as presented in Chaussard (2010).

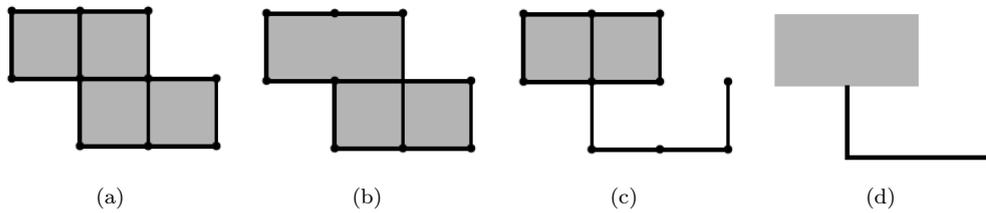


FIGURE 2.8: Some examples of sets of faces. Let A (resp. B , C , D) be the set of faces represented on **a** (resp. **b**, **c**, **d**). The set B is not a complex, as two squares miss an edge, and four edges miss a vertex. The set D is neither a complex, but A and C are complexes. Moreover, $A = B^-$. The complex A is pure as all its facets are squares, and the complex C is not pure as some of its facets are squares while some others are edges. We have $D = C^+$. The complexes A and C have a dimension equal to 2. (Chaussard, 2010)

Given a subset X of \mathbb{Z}^n and $x = (x_1, \dots, x_n) \in X$, to obtain a complex from the set X , first, we define the map Φ which associates to each element of X an n -face of \mathbb{F}^n , that is, $\Phi(x) = \{x_1, x_1 + 1\} \times \dots \times \{x_n, x_n + 1\}$. We can extend the map Φ straightforwardly to sets: $\Phi(X) = \{\Phi(x) \mid x \in X\}$. Then, the complex that we associate to a set X is $\Phi(X)^-$

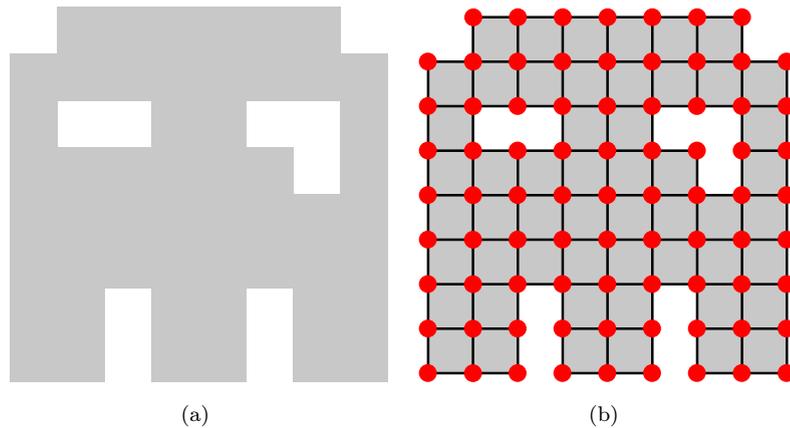


FIGURE 2.9: An example of a 2D binary image in (a) and its cubical complex representation in (b).

Chapter 3

Simplification of Binary Shapes

Three-dimensional volumetric objects are usually complicated and consist of large amounts of points to process. This fact, among others, make a difficulties to work effectively with such objects in several applications. For instance in object recognition and matching, object registration or animation is almost impossible to use directly volumetric data. As stated in [Cornea et al. \(2007\)](#) while the 3D volumetric representation is invaluable in such application, many of them require alternate simplified representation of 3D objects. The solution can be a skeleton of 3D object, which is a simple representation of the object geometry which preserves its topology (see Fig. 3.1).

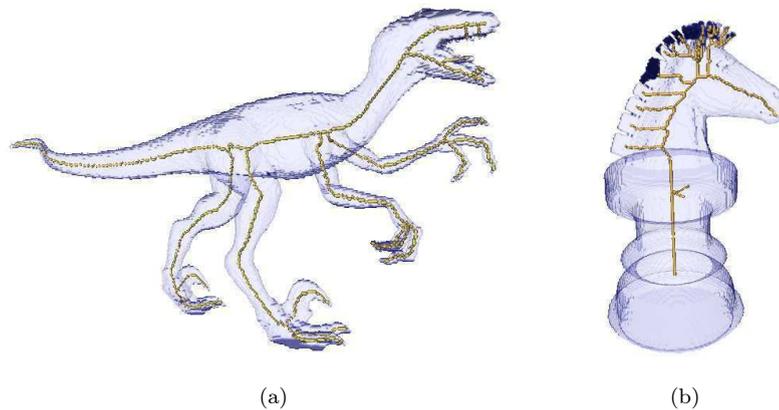


FIGURE 3.1: An example of two 3D objects and corresponding skeletons (in yellow).

The notion of the skeleton was originally defined by [Blum \(1962\)](#) through an analogy with a grassfire. If we imagine an object as a field of grass and we set on fire on the border of the field, the fire will start propagating inside the field and when fire fronts meet, they vanish. The meeting points of the flame fronts would constitute the skeleton of the object.

In the continuous framework, formally the skeleton of an object X is composed by the centers of the balls which are included in X but which are not fully included in any other

ball included in X (Blum, 1967; Calabi, 1965). This set of points is, by nature, centered in the object with respect to the distance which is used to define the notion of ball (see Fig. 3.2). However, the skeleton extraction and evaluation based on Blum's definition is very difficult in discrete framework. For instance, it is not possible in general to preserve topology.

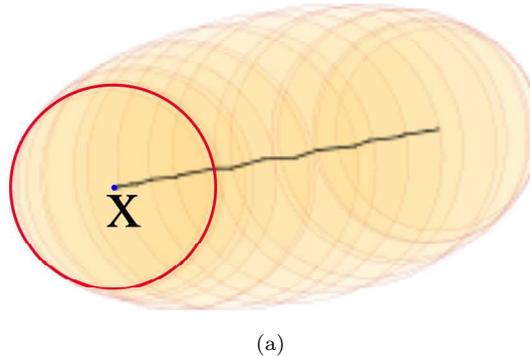


FIGURE 3.2: An example of object (ellipse) with set of centers of medial balls (in black). In red, medial ball centered at point x .

Since skeleton was first introduced it has found uses in many areas. For example, according to Cornea et al. (2007) it can be used in virtual navigation (D. Perchet, 2004; Wan et al., 2001) where skeleton define collision free paths through a scene or through an object. Producing skeleton of the background we can find collision free paths through the scene composed by many 3D objects. Virtual navigation guided by skeleton is also widely used in surgical planning or virtual endoscopy used in medical imaging to examine the interior structures of human organs (Janaszewski et al., 2009b; Nain et al., 2001; He et al., 2001). The technique is superior to the traditional fiberoptic endoscopy because of its non-invasivity, cost effectiveness and sufficient accuracy. Moreover it is free of risk and side effects like perforation or infection and can be applied for some special organs which are impossible to inspect using traditional endoscopy (e.g. blood vessels). Therefore, many prototype systems have been developed for a variety of clinical applications, including virtual colonoscopy (Janaszewski et al., 2009b; Wan et al., 2002; Jiang and Gu, 2005), virtual bronchoscopy (D. Perchet, 2004; Wegenkittl et al., 2000), virtual angiography (Bartz et al., 1999; Do Yeon and Jong Won, 2003) and others. After skeleton generation, the system is ready to perform real time navigation based on volume rendering on a personal computer. The inspection can be realized in an automatic manner, following the skeleton or by interactive navigation for more accurate study of suspicious areas (see Fig. 3.3).

Another common application of the skeleton is the registration of two images by aligning some objects that are visible in both images. For example, images from the same patient taken with different modality (CT, MRI). In this case a skeleton is used to reduce dimensionality of the problem since registration can be achieved by aligning skeletons (Aylward et al., 2003; Fritsch et al., 1994; Pizer et al., 1999).



FIGURE 3.3: An example of Virtual Colonoscopy. 3D rendering of the human colon reconstructed from CT data set with its skeleton superimposed.

Skeleton is also used for matching of 3D objects where the problem is to find in a database an object which are similar to a query object (Brennecke and Isenberg, 2004; Cornea et al., 2005a; Hilaga et al., 2001; Sundar et al., 2003). Matching can be done by comparing only skeletons. As addition such strategy can be also valid when only part of the skeleton of the query object is provided (see Fig. 3.4).

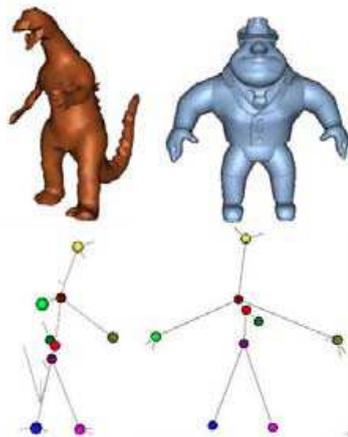


FIGURE 3.4: An example of matching. Comparing skeleton of two objects, some correspondence can be found which allow to match similar objects (Sundar et al., 2003).

There exists many more application of the skeleton (Cornea et al., 2007), for example analysis of scientific data, surface reconstruction (Amenta et al., 2000), hydrothermal plume visualisation (Santilli et al., 2004), vortex core extraction (Banks and Singer, 1994), feature tracking (Vrolijk et al., 2003), morphing (Blanding et al., 2000; Lazarus and Verroust, 1997) to cite only few, where skeletons allow to make complex topologies more easily tractable. Very interesting example of recent skeleton application is decomposing 3D objects into components (Chaussard, 2010), if a 3D object is to be treated as a sum of components it can be decomposed by analyzing its skeleton (see Fig. 3.5).

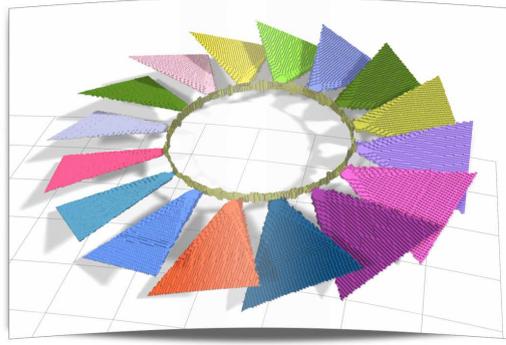


FIGURE 3.5: An example of 3D object decomposed into components. Each component has different color (Chaussard, 2010).

3.1 Skeleton Characteristics

Informally, a 3D skeleton can be a set of curves or surfaces or both, which passes through the center of an object. Some objects like for instance human colon (see Fig. 3.3) have only one curve sometimes called centerline. However, there are complicated objects with many branches, tunnels and cavities where a skeleton can be a very complicated structure. Therefore, in the literature different researchers propose a set of properties which are used to define a skeleton. Hilditch in 1969 gave first four properties that a skeleton in two-dimensional space should possess (Hilditch, 1969), that is, skeleton should be homotopic to the original object, it should be thin, it should be centered in the original object and making skeletonization of the skeleton should not change anything. Later, such set of properties was generalized to n -dimensional case and extended by adding more properties for instance smoothness, robustness, reliability, cost effectiveness (Cornea et al., 2007; Palágyi, 2002; Zhou and Toga, 1999). Some of these properties, for example, the property which states that skeleton should be smooth and reliable, are specially designed and used in particular applications (He et al., 2001; Kang and Ra, 2005). However, we can distinguish a subset of properties which is important in general case:

Singularity (Thinness) - a skeleton should be thin (Hilditch, 1969), that is, in n -dimensional space, the skeleton of an object should be at most $(n - 1)$ -dimensional. For example the skeleton of a three-dimensional object should be a set of curves or surfaces (see Fig. 3.6) and in two-dimensional case it should be only a set of curves. In some applications this requirement is even stronger, that is, a skeleton of an n -dimensional object should be always one-dimensional. Such skeleton is called a *curve skeleton* and it is generated during *curvilinear skeletonization* (Svensson et al., 2002; Dey and Sun, 2006).

Topology preserving (Homotopic) - a skeleton should have exactly the same topology as the original object (Kong and Rosenfeld, 1989; Lieutier, 2003; Saha and Chaudhuri, 1996). Intuitively, if the original object consist of one connected component without tunnels and cavities then a skeleton should have also one connected component without

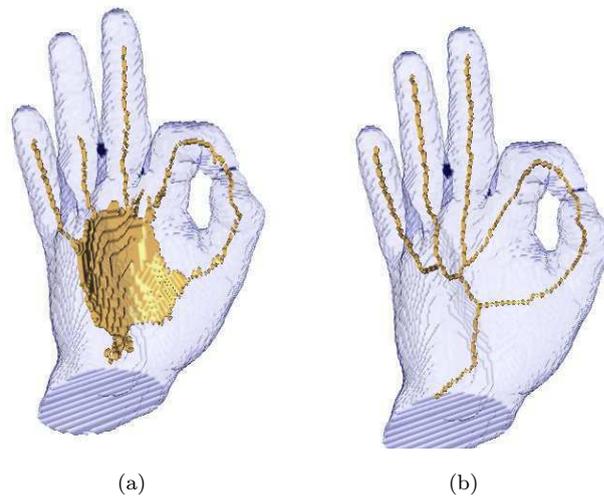


FIGURE 3.6: An example of (a) surface and (b) curve-skeleton of the same object.

loops. Moreover, as pointed in [Kong and Rosenfeld \(1989\)](#), a skeleton should be a subset of the original object, that is, during skeletonization process no points are added, only points removal are allowed. Another important problem related to the topology preservation property of the curve skeleton is a way of dealing with objects which have cavities. A curve skeleton is a one-dimensional representation of the object, therefore, it cannot preserve cavities. In [di Baja and Svensson \(2002\)](#) this problem is addressed and solution is proposed by using a modified definition of topology preservation during curvilinear skeletonization, that is, a curve-skeleton should have at least one loop around each cavity in the original object. In [Cornea et al. \(2007\)](#) this definition is extended to deal with tunnels and cavities in the original object.

Centricity (Centeredness) - a skeleton should cross the centre of an object interior ([Hilditch, 1969](#)). This postulate is essential in many applications, because it guarantees that a skeleton is a well-defined descriptor of an object. There also exist many applications where a perfect centeredness is not very important. However, the skeletonization algorithm should always produce skeletons which are centered as much as possible. More formally, to achieve perfect centeredness the skeleton should be the medial axis (or surface) of the object ([Dey and Sun, 2006](#); [di Baja and Svensson, 2002](#)).

Robustness - one of the biggest problem during skeletonization is sensitivity of skeletonization algorithms to noise on the object border which can significantly change generated skeleton (see [Fig. 3.7](#)). Therefore, the algorithm should not be sensitive to little changes in an object structure or geometric transformations such as translation or rotation. Robustness property is also related to property proposed by [Hilditch \(1969\)](#), that is, property which state that making skeletonization of the skeleton should not change anything. Furthermore, extracted skeleton should not depend on arbitrary decisions made by the skeletonization algorithm.

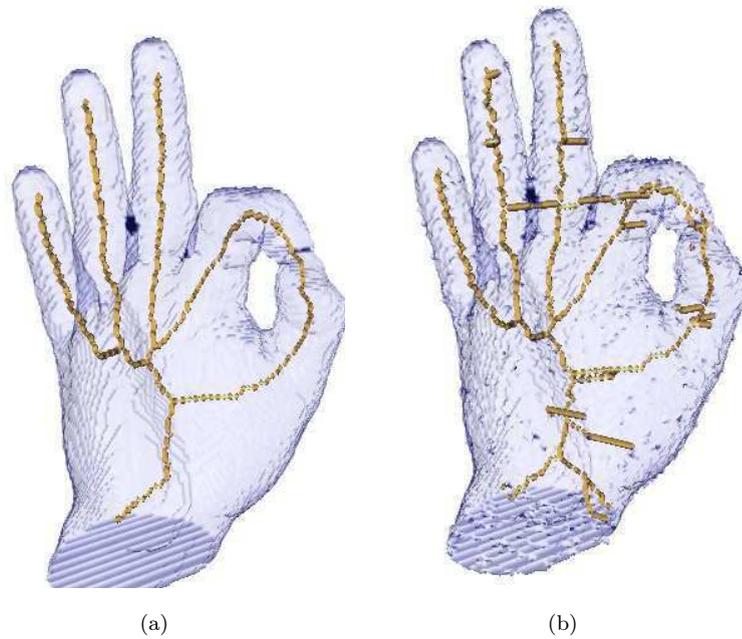


FIGURE 3.7: An example of skeleton of: (a) smooth and (b) noisy object.

Automation - the algorithm should extract a skeleton fully automatically without user interaction. This is also related to setting parameters of skeletonization algorithm if any. Algorithms which are parameter free are considered as algorithms which hold automation property. However, in some applications, algorithms which allow the user to tune some parameters are more interesting. As stated in [Cornea et al. \(2007\)](#) because the skeleton is an approximation of the complex components of an object, the skeletonization process and the skeleton itself should reflect the natural hierarchy of these complexities ([Cornea et al., 2005b](#); [Katz and Tal, 2003](#)). For example, during skeletonization, we can decide which level of hierarchy we would like to achieve (see Fig. 3.8). This property can also cover a problem of curve-skeleton automatic decomposition into different components like end-points, branches, loops or junction points. In other words, the curvilinear skeletonization process should allow to simply transform skeleton into graph-like structure ([Giblin and Kimia, 2004](#)).

Cost effectiveness (Efficiency) - for large and complicated data computational time and memory utilization are critical. Therefore algorithms should be fast enough to extract a skeleton for complicated data in seconds on standard PC computer. More formally speaking, algorithms should have linear complexity, in respect to the size of the input, which guarantees its efficiency.

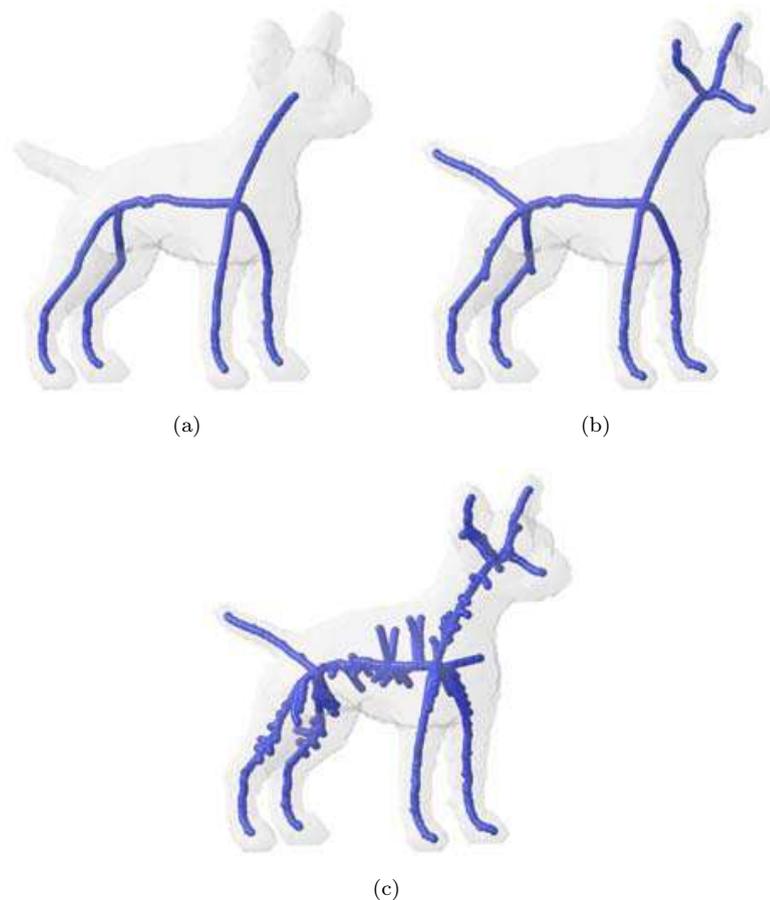


FIGURE 3.8: An example of the object with its skeletons on different level of hierarchy (Cornea et al., 2005b).

3.2 Skeletonization Methodologies

In the literature many skeletonization algorithms have been proposed. Skeletons produced using different skeletonization techniques, hold different sets of properties presented above. Some of them are specially designed to fit into particular application, others try to be as much general as possible. Some authors, for example Palágyi (2008); chun Ma et al. (2003); Telea and Vilanova (2003), categorize skeletonization algorithms into general groups based on a main concept used to compute skeletons. Following Cornea et al. (2007) the main skeletonization concepts can be presented as follows:

Manual extraction - these methods require significant manual work of a user who is responsible for marking a set of points in the centre of an object. Then the skeleton is linearly interpolated between consecutive marked points. Unfortunately the method is time consuming, sometimes difficult to perform and does not guarantee the centricity of marked points, not the homotopy equivalence, because of possible human mistakes. The quality of resulting skeleton also depends on the amount of points which were marked.

Topological thinning - these algorithms delete, at each iteration, some points satisfying both topological (simple points) and geometrical criteria, from the border of an object. The process stops when no more such points can be removed. See chapter 3.3 for more detailed review of this group of skeletonization algorithms.

Distance-based (Voxel coding based) transformation - a voxel coding scheme is a voxel by voxel recursive propagation and assignment of integer codes to object voxels starting from a set of voxels which are called seeds. Most of these algorithms use a special voxel coding to approximate distance transform where the seed set consists of object border voxels. Result image is called a distance field which has very useful property from skeleton generation point of view. Its ridges correspond to the voxels that are local centers in the object. Based on the ridges various algorithms use various approaches to build the skeleton. Usually the set of ridges is pruned and then remaining voxels are connected in order to form connected skeleton (di Baja and Thiel, 1996; Borgefors et al., 1999; Toriwaki and Mori, 2001b; Zhou and Toga, 1999).

General-field methods - methods based on potential field are similar to distance-based methods. However they use different kinds of vector field than those based on distance transform. For example electrostatic field function, radial basis functions, or generalized potential function where a generalization of the Newtonian (gravitational) potential field is used (see Fig. 3.9 for example) (Ahuja and Chuang, 1997; Rumpf and Telea, 2002; Chuang et al., 2000; Cornea et al., 2005a; Abdel-Hamid and Yang, 1994; Grigorishin et al., 1998; che Wu et al., 2003). In these methods, first, the potential in every interior point of the object is determined. For instance, a sum of potentials is generated by charges placed on the border of the object. Then, skeleton paths are determined using a force-following algorithm. Each path started at a seed point ends at a potential minimum detected by a major change in the vector direction. Final skeleton is computed by connecting extracted skeleton paths.

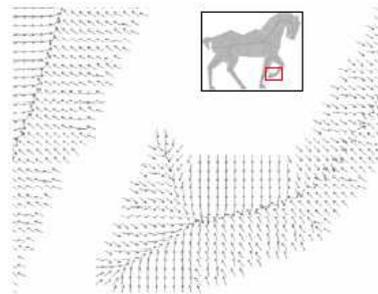


FIGURE 3.9: An example of potential field function, visible as vectors, of the zoomed region marked by red box. (Cornea et al., 2005a)

Geometric (Voronoi-based) transformations - there are several geometric approaches to compute skeletons in the continuous space or plane, however the most popular are methods which based on Voronoi diagrams (Brandt and Algazi, 1992; Näf et al., 1997; Attali

and Montanvert, 1997; Attali and Lachaud, 1999, 2001; Ogniewicz and Ilg, 1992; Ogniewicz and Kübler, 1995; Ogniewicz, 1994). The Voronoi diagram represents a subdivision of the space into regions that are closer to a generator element (a surface sample in the case of a 3D model) than to any other such element. These methods based on fact that the interior faces of the Voronoi diagram approach the medial axis as the density of surface samples increases. Therefore, the skeleton can be extracted directly from Voronoi-based transformation (see Fig. 3.10).

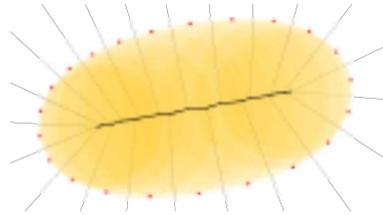


FIGURE 3.10: An example of Voronoi diagram (in light gray) of the object (in yellow) in respect to selected set of surface point (red dots). Corresponding skeleton is superimposed (in black).

Hybrid methods - these methods consist of different algorithms which cannot be simply categorized into previous groups. They work in different frameworks than voxel framework for example in cubical complex framework (see chapter 2.2), but use similar thinning scheme. It is also possible, in one algorithm, to use both thinning and distance transform technique. For instance, thinning guided by distance transform or constrained by medial axis to achieve better centeredness property.

3.3 Thinning in The Digital Topology Framework

Thinning is the most popular concept in skeletonization. Therefore, in this thesis we focus on and present more deeply this approach to compute skeleton.

As was mentioned in previous section thinning is a process of removing some simple points from the input object until stability, that is, until no more simple points exist or additional special conditions occur. Example of such additional condition is for example a set of points which are called an *inhibitor set*. Inhibitor set is a set of point which should not be removed from the object. A common example of an inhibitor set is a filtered medial axis of the input object (see chapter 4.2). Another example of constraints during thinning is preservation of points which are special kind, for instance, curve end points or surface points. Thanks to the notion of connectivity numbers such points and simple points as well, can be effectively detected in the input object only by analyzing the neighborhood of the point. These characterization yields efficient skeletonization strategies.

Proposition 3.1. *Rosenfeld (1979)* Let $X \subset \mathbb{Z}^2$, $p \in X$, and k be the adjacency relation chosen for X . If $X \cap \Gamma_8^*(p)$ has the same number of k -connected components as $X \cap (\Gamma_8^*(p) \cup \{p\})$, and that $\bar{X} \cap \Gamma_4^*(p) \neq 0$, then p is simple for X .

Proposition 3.2. *Bertrand and Malandain (1994)* Let $X \subseteq \mathbb{Z}^3$ and $x \in X$, let $T_{26}(x, X)$ be the number of 26-connected components of $(X \cap \Gamma_{26}^*(x))$, and let $T_6(x, X)$ be the number of 6-connected components of $(X \cap \Gamma_6^*(x))$.

In 26-connectivity, x is simple for X iff $T_{26}(x, X) = 1$ and $T_6(x, \bar{X}) = 1$.

In 6-connectivity, x is simple for X iff $T_6(x, X) = 1$ and $T_{26}(x, \bar{X}) = 1$

The possibility to decide which simple points will be removed during thinning gives us a tool to produce the different kinds of skeletons. The decision of what kind of skeleton should be used strongly depends on its application. We can distinguish different types of skeletons in respect to the dimension of the result:

- Ultimate skeleton - it is a type of skeleton which gives us information about topology of the object without care about its geometry. For example ultimate skeleton of a each solid connected component without holes and tunnels, consists of one point.
- Curvilinear skeleton - it is an one-dimensional representation of the object which preserves both topology and geometry of the object.
- Surface skeleton - In three-dimensional case we can also compute a two-dimensional skeleton. It will provide more geometrical information than curvilinear skeleton which can be useful in some applications.

The thinning process, that is, simple points removal can be done in two different way. Simple points can be remove sequentially or in parallel. By removing simple points sequentially, we can simply produce thin and topologically correct skeletons. However attention need to be payed to set proper order of points removal if we want to achieve well centered and smooth skeleton. From the other side, we can remove several simple points simultaneously and produce skeletons which should be well centered but not necessary thin. However, parallel removal of simple points can change topology therefore more attention need to be payed to produce topologically correct results.

A more comprehensive review about different thinning strategies is presented in the following sections.

3.3.1 Sequential thinning algorithms

The most obvious thinning strategy is based on a sequential removal of simple points from an object until no more simple point can be found. The most important problem during

sequential removal of points is a problem of points removal order. In the simplest case, simple points can be removed totally randomly which leads to produce ultimate skeleton which is not necessary centered in an object (see Fig. 3.11). To achieve well centered skeleton simple points should be removed following a precisely defined order. Intuitively they should be removed "layer by layer", from outside to inside of an object. For example [Arcelli \(1981\)](#); [Pavlidis \(1980\)](#), in the 80s, proposed sequential thinning algorithms which remove simple points from the contour of an object. This simple process can be described as presented in algorithm 1. First the set of border points S of an object is computed, and then simple points are removed one after the other from S . When no more simple points exist in S the new set of border points is generated.

Algorithm 1 Basic Sequential Thinning(**Input** X **Output** Y)

```

01.   $Y = X$ 
02.   $S = \{p \in Y \mid p \text{ is simple and belongs to the border of } Y\}$ 
03.  while  $S \neq \emptyset$  do
04.      foreach  $p \in S$  do
05.           $S = S \setminus \{p\}$ 
06.          if  $p$  is still simple for  $Y$  then
07.               $Y = Y \setminus \{p\}$ 
08.          end
09.      end
10.       $S = \{p \in Y \mid p \text{ is simple and belongs to the border of } Y\}$ 
11.  end
12.  return  $Y$ 

```

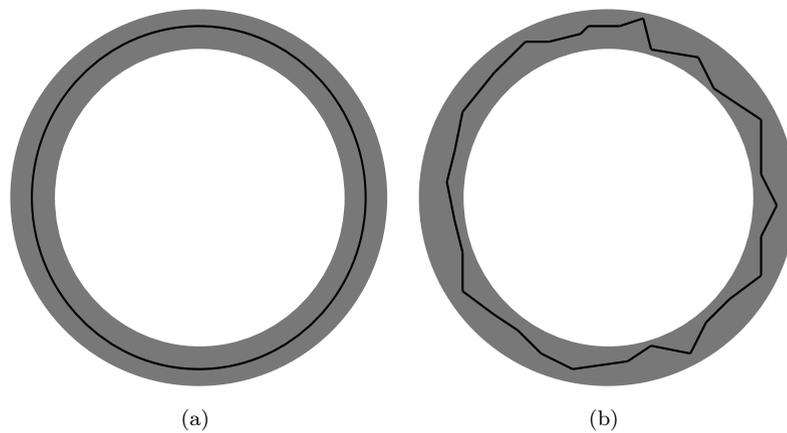


FIGURE 3.11: Various removal orders of simple points lead to different skeletons.

Another possibility to define an order of simple points removal is to use a priority function which is computed on the object. Then, simple points are removed according to their values ([Davies and Plummer, 1981](#)). In this case a "layer" of the object is considered as a set of points which have the same priority value. The most common choice for priority map is the Euclidean distance transform of an object ([Couprie et al., 2007](#); [Malandain and Fernández-Vidal, 1998](#); [Talbot and Vincent, 1992](#)) or other discrete distances ([Pudney,](#)

1998). In (Chaussard et al., 2010) is shown that using values of lambda function as a priority gives skeletons with very good properties.

Algorithm 2 implements a basic sequential thinning strategy which utilizes a priority function D . In each iteration only simple points which have the smallest priority are deleted. In addition to the priority function, the algorithm 2 uses inhibitor set W which permits to preserve points of W during thinning. In the literature, some authors report that using thinning guided by priority function and with defined inhibitor set can lead in some cases to poor quality skeletons. Solutions to this problem can be found in (Couprie et al., 2007; Talbot and Vincent, 1992) where the inhibitor set is dynamically modified during thinning or priority function is recalculated to avoid some effects of discretization.

Algorithm 2 Basic Thinning with Priority(**Input** X, W, D **Output** Y)

```

01.    $Y = X$ 
02.   while there exists a simple point in  $Y \setminus W$  do
03.        $A = \{y \in Y \setminus W \mid y \text{ is simple for } Y\}$ 
04.        $B = \{x \in A \mid \text{for all } y \in A, D(x) \leq D(y)\}$ 
05.       Let  $z \in B$ 
06.        $Y = Y \setminus \{z\}$ 
07.   end
08.   return  $Y$ 

```

Many thinning algorithms which sequentially remove simple points were proposed. A very exhaustive survey of thinning methods in 2D before 1992 can be found in (Lam et al., 1992).

3.3.2 Fully parallel thinning algorithms

In the previous section we discussed the thinning strategy where at each iteration only one simple point can be removed. Based on the priority function the order of points removal was strictly defined. However, even so, the sequential algorithms are order-dependent, that is, some arbitrary choices have to be made during thinning since several simple points can have the same priority.

The order-independent algorithms should remove the set of all points in a layer at once in one iteration. Moreover, the decision about which points can be removed should not depend on other decisions made in the same iteration. The thinning strategy which follows such rules is called fully parallel thinning.

In the fully parallel thinning scheme, a set of simple points is removed simultaneously. This process is more similar to the layer-by-layer points removal model. Therefore, the priority function is generally not needed. The set of simple points which can be removed simultaneously is computed based on a criterion C which is constant during thinning, and which must ensure topology preservation.

The generic fully parallel thinning algorithm can be presented as in algorithm 3. Where, in each iteration all simple points which satisfy a criterion C are removed. The process ends when there is no more simple points exist or none of them satisfy C .

Algorithm 3 Generic Fully Parallel Thinning(**Input** X **Output** Y)

```

01.    $Y = X$ 
02.   repeat
03.        $A = \{x \in Y \mid x \text{ is simple for } Y \text{ and } C(x) \text{ is true}\}$ 
04.        $Y = Y \setminus A$ 
05.   until  $A = \emptyset$ 
06.   return  $Y$ 

```

However, removing several simple points simultaneously may not guarantee that topology will be preserved. This fact makes the problem of designing a topology-preserving parallel thinning algorithm very complex. The first fully parallel thinning algorithm was proposed by Rutovitz (1966). It has been proved that Rutovitz's algorithm does not always preserve topology. Some modifications exist, for example proposed in (Couprie, 2006; Eckhardt, 1988). In 3D the first fully parallel algorithm was proposed by Manzanera et al. (1999).

During the last decades some theories have been proposed in order to characterize sets of simple points which can be removed at the same time without changing the topology. For example, in the discrete topology framework Ronse (1988) define the notion of set D which consist only of simple points and that there exist a discrete process of removing simple points which allow to transform X into $X \setminus D$, such set we called a *simple set of X* or we say that D is *simple for X* . For example in two dimensional space, given an object $X \subset \mathbb{Z}^2$, $|C_k[X]| = 1$ and a set $D \subset X$, that, if D consist only of simple points of X and $|C_k[X]| = |C_k[X \setminus D]|$ and $|C_{\bar{k}}[\bar{X}]| = |C_{\bar{k}}[\bar{X} \cup D]|$, then D is simple for X .

Another notion presented by Ronse (1988) which can help to construct an topology preserving algorithm is a notion of minimal sets of points which can not be removed. Otherwise, topology of an object will change. Such sets are called *minimal non-deletable sets*. Minimal non-deletable sets were used to topologically validate some parallel thinning algorithms in 2D (Hall, 1992), 3D (Kong, 1993, 1995; Ma, 1994) and 4D cases (Gau and Kong, 2003; Kong and Gau, 2004).

Proposition 3.3. *Ronse (1988) Let $X \subset \mathbb{Z}^2$, and let k be equal to 4 or 8. A set $U \subseteq X$ is a minimal non- k -deletable set from X iff one of the following holds:*

- U consists of a single pixel that is not k -simple for X (in the sense of the k -connectivity).
- U consists of a pair of 8-adjacent pixels which are k -simple for X , but U is not k -simple for X .
- If $k = 8$, U consists of a triple or quadruple of pairwise 8-adjacent pixels, and U is an 8-connected component of X .

In three-dimensional space, [Bertrand \(1995\)](#) introduces the notion of *P-simple points* in order to characterize which simple points could be removed simultaneously. Based on *P-simple points* another set of computer test of 3D parallel thinning algorithms or new algorithms where proposed ([Bertrand, 1995](#); [Lohou and Bertrand, 2004a, 2005a](#)).

Most fully parallel thinning algorithms are based on similar strategies to define which points can be removed simultaneously. They utilize a set of masks which define how the neighborhood of a point should look to allow such point to be removed. Only points which match at least one mask can be removed. The number of masks, their size, and points configurations strictly depends on the algorithm and has direct impact to the quality of the resulting skeleton. Using proper set of masks it is possible to propose an algorithm which produces skeletons with particular properties like for example curvilinear or surface skeletons. The 2D fully parallel algorithm proposed by [Pavlidis \(1981\)](#) uses four masks of size 3x3 points and produces a symmetric skeleton. Several other parallel thinning algorithms which produce symmetric skeletons have been proposed based on a set of masks ([Hall, 1989](#); [Eckhardt and Maderlechner, 1993](#); [Bernard and Manzanera, 1999](#); [Ma, 1994](#)). [Guo and Hall \(1992\)](#) proposed an algorithm which produces asymmetric results and is based on masks which contain at least 11 points.

3.3.3 Directional thinning algorithms

The second possibility to remove a set of simple points simultaneously is to use a strategy where each iteration is divided into several sub-iterations. It permits to use a different set of deleting criterions in each sub-iteration. In the directional thinning scheme points are divided in respect to their orientation. The orientation of a point is defined by a set of complementary points in its neighborhood (see [Fig. 3.12](#)).

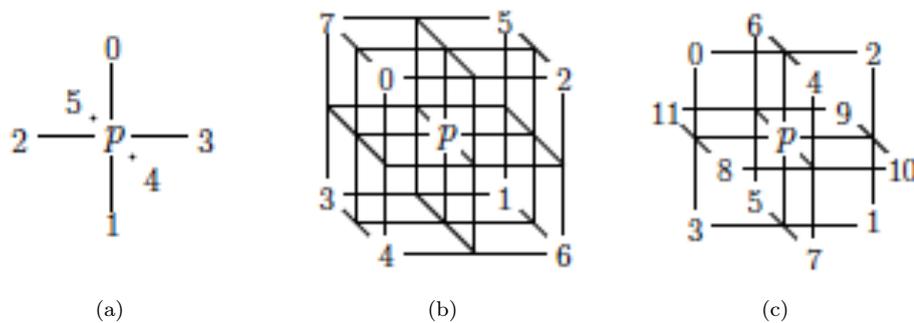


FIGURE 3.12: Basic partition of the space into directions. (a): 6-directions; (b): 8-directions; (c): 12-directions. Each number i represents a voxel which has to be a background voxel to allow the deletion of p in the i -th sub-iteration.

Algorithm 4 shows the generic directional thinning scheme. It performs n sub-iterations and removes only simple points which satisfy i -th condition C_i during i -th sub-iteration. It is important to notice that the condition of orientation alone is not sufficient to guarantee topology preservation, some additional conditions are necessary.

The first proposed directional thinning algorithm (Stefanelli and Rosenfeld, 1971) was designed for 2D and use 4 sub-iterations.

In the 3D case, most proposed algorithms use 6 sub-iterations (Bertrand, 1995; Gong and Bertrand, 1990; Lee et al., 1994; Lohou and Bertrand, 2005b; Ma and Wan, 2000; Mukherjee et al., 2000; Palagyi and Kuba, 1998; Xie et al., 2003) or 3 sub-iterations (Palagyi, 2002, 2007a,b). There exist also algorithms based on 8 (Palagyi and Kuba, 1999a) or 12 sub-iterations (Lohou and Bertrand, 2004b; Palagyi and Kuba, 1999b).

Algorithm 4 Generic n -Subiteration Thinning(**Input** X **Output** Y)

```

01.    $Y = X$ 
02.    $remove = \text{TRUE}$ 
03.   while  $remove$  do
04.      $remove = \text{FALSE}$ 
05.     for  $i : 1 \rightarrow n$  do
06.        $A = \{x \in Y \mid x \text{ is a simple point and } C_i(x) \text{ is true}\}$ 
07.       if  $A \neq \emptyset$  then
08.          $remove = \text{TRUE}$ 
09.       end
10.        $Y = Y \setminus A$ 
11.     end
12.   end
13.   return  $Y$ 

```

3.3.4 Subfield thinning algorithms

The third parallel strategy is based on grouping all points of \mathbb{Z}^n into m disjoint sets S_0, \dots, S_{m-1} , for $m \geq 2$ and $\bigcup_{i=0}^{m-1} S_i = \mathbb{Z}^n$. Such sets are called subfields and can be processed separately, one after another. Only simple points which belongs to the actually processed subfield can be removed. The main idea of dividing space into subfield allow to delete points with controlled impact on its neighborhood and in consequence preserve topology.

The number of subfields depends on used algorithm. The common choices are 2, 4 and 8 subfields respectively (see Fig. 3.13). The same generic algorithm (see Algorithm 4) can be used. In the case of 8 subfields, the condition $C_i(x) = \{x \in S_i\}$ is sufficient to ensure topology preservation. However for 2 and 4 subfields, additional conditions are necessary.

In the literature several subfield thinning algorithms have been proposed using 2 subfields (Kardos et al., 2010; Ma and Wan, 2001; Ma et al., 2002a), 4 subfields (Ma et al., 2002b; Nemeth et al., 2010) and 8 subfields (Nemeth et al., 2010).

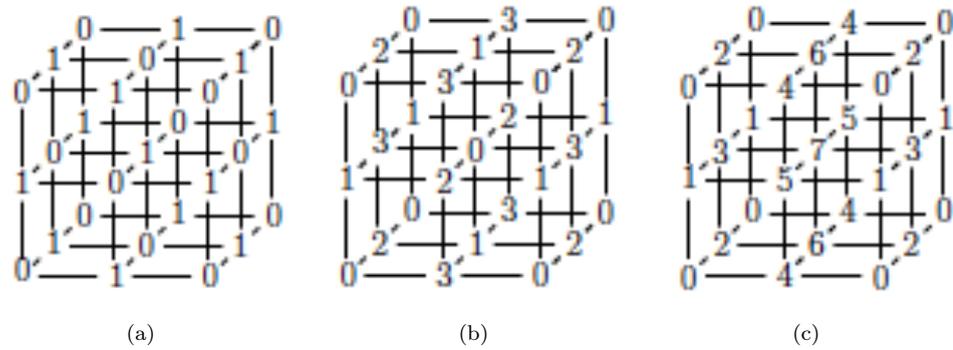


FIGURE 3.13: Partition of the space into subfields. (a): 2 subfields; (b): 4 subfields; (c): 8 subfields. In i -th sub-iteration only points marked i will be considered.

3.4 Thinning in The Cubical Complex Framework

3.4.1 Critical kernels based thinning

Recently, [Bertrand \(2007\)](#) introduced the notion of critical kernels in the framework of cubical complexes. The critical kernels theory led to a new definition and new characterizations of simple points in 2D, 3D and 4D (see ([Coupric and Bertrand, 2009](#))). The links between this framework, P-simple points and minimal non-deletable sets were established in ([Coupric and Bertrand, 2008](#)). Critical kernel is a powerful tool to study parallel homotopic thinning and can be used to check the topological validity of existing algorithms ([Coupric, 2006](#); [Bertrand and Coupric, 2008, 2006](#)). Moreover, critical kernels may also be used to design new algorithms.

Thinning algorithms based on critical kernels consist of iteratively computing the critical kernel of the result of the previous step (see Fig. 3.14), and removing any subset of the object, under the condition that the critical kernel is preserved.

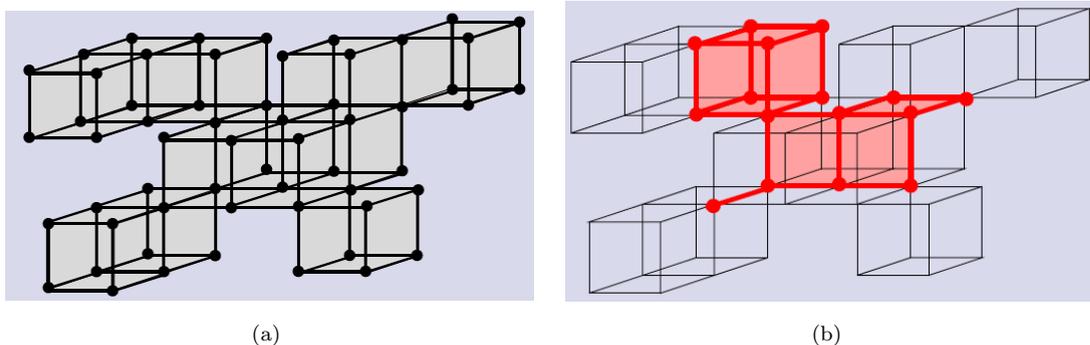


FIGURE 3.14: (a) An object (a complex) and (b) its Critical Kernel (in red) ([Bertrand, 2007](#))

3.4.2 Collapse based thinning

Another thinning scheme defined in the cubical complex framework relies on the *collapse* operation which is an elementary topology-preserving transformation that has been introduced by [Whitehead \(1939\)](#). Collapse can be seen as a discrete analogue of a retraction, that is, a continuous deformation of an object onto itself and plays an important role in combinatorial topology.

Let $\mathcal{X} \preceq \mathbb{F}^n$, and let f, g be two faces of \mathcal{X} . The face g is free for \mathcal{X} , and the pair (f, g) is a free pair for \mathcal{X} if f is the only face of \mathcal{X} such that g is a proper face of f .

Let $\mathcal{X} \preceq \mathbb{F}^n$, and let (f, g) be a free pair for \mathcal{X} . The complex $\mathcal{X} \setminus \{f, g\}$ is an elementary collapse of \mathcal{X} .

Let $\mathcal{X} \preceq \mathbb{F}^n$, the complex \mathcal{X} collapses onto Y if there exists a sequence of complexes $(\mathcal{X}_0, \dots, \mathcal{X}_l)$ of \mathbb{F}^n such that $\mathcal{X} = \mathcal{X}_0, Y = \mathcal{X}_l$ and for all $i \in \{1, \dots, l\}$, \mathcal{X}_i is an elementary collapse of \mathcal{X}_{i-1} . We also say, in this case, that Y is a collapse of \mathcal{X} .

In other words, the collapse operation consists of removing two distinct elements (f, g) from a complex \mathcal{X} under the condition that g is contained in f and is not contained in any other element of \mathcal{X} . This operation may be repeated several times and is the basic operation for performing homotopic thinning of a complex (see [Fig. 3.15](#) for example).

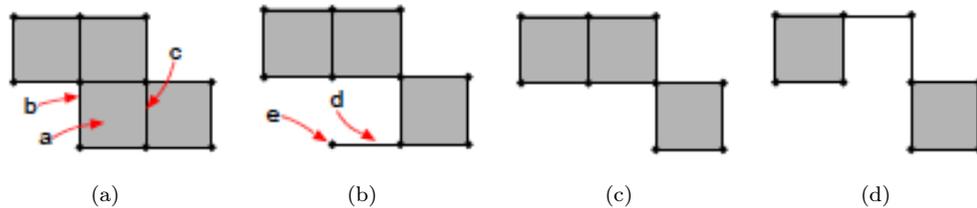


FIGURE 3.15: Collapse of complexes - Let A (resp. B, C, D) be the set of faces represented on a (resp. b, c, d). In the complex A, the edge b is only included in the square a: the pair (a, b) is free for A. The edge c is included in two squares of A, the pair (a, c) is therefore not free for A. The complex $B = A \setminus \{a, b\}$, is an elementary collapse of A. In B, the vertex e is included in only one edge: the pair (d, e) is free for B, and $C = B \setminus \{e, d\}$ is an elementary collapse of B. The complex D is an elementary collapse of C, and therefore, D is a collapse of A. ([Chaussard, 2010](#))

Based on collapse several thinning algorithms, which can produce curvilinear or surface skeletons, were proposed ([Chaussard and Couprie, 2009](#)). A recent review is presented in ([Chaussard, 2010](#)).

Chapter 4

Preserving Visual Aspects of a Binary Shape

Skeletonization algorithms presented in the previous chapter permit to simplify objects and preserve their topology. Some of them try also to preserve some visually important features of the object. However, preserving geometry of the object is not always guaranteed. For example, based on certain kinds of skeletons it is not possible, in general, to reconstruct original object. We say that these skeletons do not hold reconstruction property, which can be defined as ability to fully recover the input object from its skeleton.

In this chapter we introduce the notion of medial axis of the object. It is an object representation which holds reconstruction property and is suitable to preserving all visual aspects of an input shape.

4.1 Medial Axis

The notion of medial axis has been introduced by Blum in the 60s ([Blum, 1961, 1967](#)). In the continuous space it can be formalized by two following definitions:

- *Interpretation (a)* - Let $X \subset \mathbb{R}^n$. The medial axis of an object X is composed by the centers of the balls which are included in X but which are not fully included in any other ball included in X .
- *Interpretation (b)* - Let $X \subset \mathbb{R}^n$. The medial axis of an object X is composed by the points $x \in X$ that have at least two nearest points on the border of X .

These interpretations gives medial axes which consist of different set of points, however, difference concerns only a negligible set of points (see [Matheron \(1988\)](#)) and in general

case, the set produced based on interpretation (b) is a strict subset of the set produced based on interpretation (a).

Notice that in some works, the medial axis is also called "skeleton", especially in continuous framework. In deed, in the continuous space, the medial axis has all the desired properties of a skeleton: it is thin, centered and homotopy-equivalent to the shape. However, that is no longer true in the discrete space. In this thesis, we shall restrict the use of term "skeleton" to the case where the skeleton is topologically equivalent to the original shape, and we will not require the strict reconstruction property.

In this thesis we focus on the discrete medial axis based on the Euclidean metric. Therefore we need to recall the formal definition of notion of the medial axis in discrete framework (see also (Rosenfeld, 1982; Talbot and Vincent, 1992)). The following definition is based on Euclidean balls and interpretation (a).

Definition 4.1. Let $X \subseteq \mathbb{Z}^n$ and D_X its Euclidean distance transform. A ball $B(x, r) \subseteq X$, with $x \in X$ and $r \in \mathbb{R}$, is *maximal for X* if it is not strictly included in any other ball included in X . The *medial axis of X*, denoted by $MA(X)$, is the set of the all couples (x, r) such that $B(x, r)$ is a maximal ball for X , that is,

$$MA(X) = \{(x, D_X(x)) \mid \forall y \in X \setminus \{x\}, B^<(x, D_X(x)) \not\subseteq B^<(y, D_X(y))\}$$

In the literature different methods have been proposed to compute the medial axis approximately or exactly, for instance methods relying on discrete geometry (Borgefors et al., 1991; Ge and Fitzpatrick, 1996; Malandain and Fernández-Vidal, 1998; Remy and Thiel, 2003, 2005; Hesselink and Roerdink, 2008; Chaussard et al., 2010), digital topology (Davies and Plummer, 1981; Vincent, 1991; Talbot and Vincent, 1992; Pudney, 1998), mathematical morphology (Serra, 1982; Soille, 1999), computational geometry (Attali and Lachaud, 2001; Ogniewicz and Kübler, 1995; Attali and Montanvert, 1996), partial differential equations (Siddiqi et al., 1999), or level-sets (Kimmel et al., 1995).

The medial axis is a very useful representation of the object and similarly to skeleton plays a major role in shape analysis in numerous applications, for example object recognition, registration or compression. A study of the various properties of medial axes in the discrete framework is done in (Hulin, 2009). We can also characterize medial axes following the same set of properties as skeleton (see section 3.1).

The main property of the medial axis is reconstruction property. From the medial axis points and associated ball radii, one can exactly reconstruct the original shape. Another interesting property is centerdness. Medial axes are centered in the object in respect to the used distance. However, in consequence in the discrete framework, the Euclidean medial axis may not be thin: in some parts, it can be two pixel thick. To cope with this, Saúde et al. (2006); Saúde (2011), propose an Euclidean medial axis on higher resolution and give some thinness properties of this axis.

However it can be hard or even impossible to use this tool effectively without first dealing with some problems, especially in discrete spaces and with noisy objects.

Firstly, the medial axis in discrete spaces has not, in general, the same topology as the original object. Solutions to this problem have been proposed by several authors, for instance (Davies and Plummer, 1981; Talbot and Vincent, 1992; Couprie and Bertrand, 2012). They use discrete homotopic transformations guided and constrained by the medial axis, to obtain homotopic skeleton which contains the medial axis.

The second problem is sensitivity of the medial axis to small contour perturbations (robustness property) (see, for example, Fig. 4.1). In other words, the medial axis is not stable under small perturbations of a shape: modifying a shape slightly (for example in terms of Hausdorff distance) can result in substantially different medial axes. This is a major difficulty when the medial axis is used in practical applications (e.g. shape recognition). A recent survey which summarizes selected relevant studies dealing with this topic is presented in (Attali et al., 2009). This fact, among others, explains why it is usually necessary to add a filtering step (or pruning step) to any method that aims at computing the medial axis, and why a nonreversible but simplified description of binary objects is of interest.

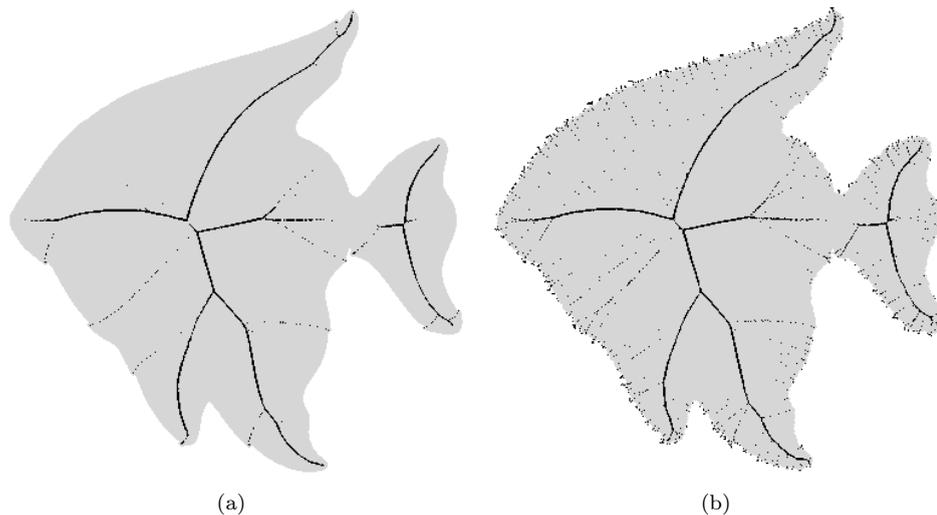


FIGURE 4.1: (a): a shape (in gray) and its Euclidean medial axis (in black); (b): the same shape, but with small amount of noise added on the contour. The medial axis of the shape (b) is much more complicated than the medial axis of the shape (a).

4.2 Medial Axis Filtering

Different criteria can be used to locally detect and discard spurious medial axis points or branches.

Usually methods proposed in the literature are based on a single parameter, and they have the property that any filtered medial axis of a shape X for a parameter value λ is a subset of any filtered medial axis of X for parameter value μ , whenever $\lambda \geq \mu$. In other words, the set of all filtered medial axes of a shape forms a hierarchy.

In the following sections we review the most popular concepts to filter medial axes.

4.2.1 Filtering based on medial ball size

The simplest strategy to filter the medial axis is to keep only couples which are centers of maximal balls of at least a given diameter (Coeurjolly and Montanvert, 2007).

Definition 4.2. Let $X \subseteq \mathbb{Z}^n$ and $MA(X)$ its Euclidean medial axis. Let t be a filtering parameter. The filtered medial axis, denoted by $RFMA(X, t)$, is the set of the all couples (x, r) such that the radius r of the ball $B(x, r)$ is not smaller than filtering parameter t , that is,

$$RFMA(X, t) = \{(x, r) \in MA(X) : r \geq t\}$$

This simple strategy alone gives poor results in general (see Fig. 4.2(b) for example) and is not suitable to deal with object with features at different scales. However, it can be used as a supplementary filtering criterium in addition to a more robust one. For example it can be used with bisector angle (see section 4.2.4), since, the noisy objects have usually many medial balls which are very small (e.g single points) and can be simply removed using this strategy (see Fig. 4.2(d)) .

4.2.2 Filtering based on ball covering

A more complex criterion was proposed by Coeurjolly and Montanvert (2007), they utilize information about ball importance in the shape with respect to all other balls by counting the number of object points inside a ball which are not covered by other balls. The medial axis point will be removed if the uncovered area of corresponding ball is too small (see Fig. 4.3).

Definition 4.3 (Coeurjolly and Montanvert (2007)). Let $X \subseteq \mathbb{Z}^n$ and $MA(X)$ its Euclidean medial axis. Let t be a filtering parameter. We denoted by $CFMA(X, t)$, the filtered medial axis defined by

$$CFMA(X, t) = \{(x, r) \in MA(X) : |B(x, r) \setminus \bigcup_{(y, r_2) \in MA(X)} B(y, r_2)| \geq t\}$$

Such filtering strategy gives much better results than previous one since it permits to remove not only small medial balls but also very large ones which are not important to

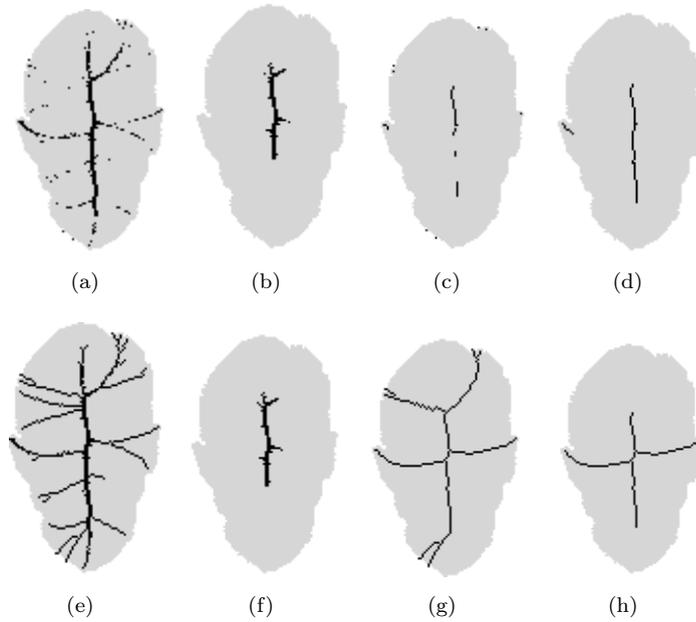


FIGURE 4.2: (a): Example of object (in gray) and its medial axis (in black); (b): Filtered medial axis with RFMA gives poor results; (c): Filtering based on bisector angle, preserve some points close to the border; (d): Filtering based on bisector angle and RFMA as a complementary gives much better results; (e,f,g,h): an ultimate homotopic skeleton of (a,b,c,d resp.) with the constraint of retaining the points of its filtered medial axis.

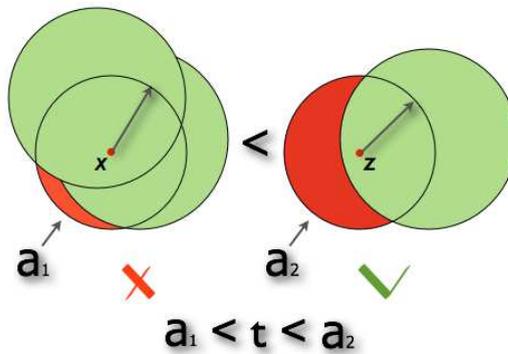


FIGURE 4.3: Medial axis filtering based on ball covering. The medial ball which uncovered area a (in red) is smaller than threshold parameter t will be removed.

the object understanding or reconstruction. However, when the object consists of features at low scale, this method can fail to generate suitable results (see Fig. 4.4).

4.2.3 λ -medial axis

This filtering method proposed by [Chazal and Lieutier \(2005\)](#) in the continuous framework, later transposed to discrete framework, utilizes information about circumradius of the closest points on the shape border, to the given medial axis point.

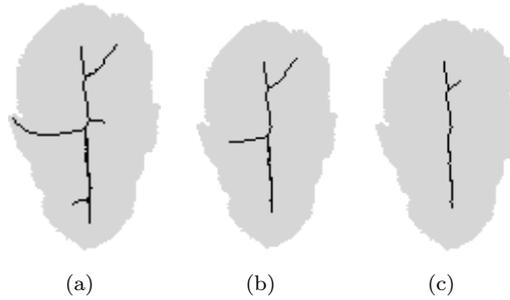


FIGURE 4.4: An example, of medial axis filtering based on ball covering (in black) in which the method failed due to features at different scale in the input object (in gray). On (a) insufficient filtering, spurious branches exist. On (b) spurious branches still exist but right ear is missing. On (c) one spurious branch still exist but ears were fully removed.

Let S be a non-empty subset of \mathbb{R}^n , and let $x \in \mathbb{R}^n$. The *projection of x on S* , denoted by $\Pi_S(x)$, is the set of points y of S which are at minimal distance from x , defined by,

$$\Pi_S(x) = \{y \in S \mid \forall z \in S, d(y, x) \leq d(z, x)\}$$

If X is a subset of \mathbb{R}^n , the *projection of X on S* is defined by $\Pi_S(X) = \bigcup_{x \in X} \Pi_S(x)$.

Let $S \subset \mathbb{R}^n$, we denote by $R(S)$ the radius of the smallest ball enclosing S , that is, $R(S) = \min\{r \in \mathbb{R} \mid \exists y \in \mathbb{R}^n, B(y, r) \supseteq S\}$.

The λ -medial axis may now be defined based on these notions.

Definition 4.4 (Chazal and Lieutier (2005)). Let X be an open bounded subset of \mathbb{R}^n , and let $\lambda \in \mathbb{R}^+$. The λ -medial axis of X is the set of points x in X such that $R(\Pi_{\bar{X}}(x)) \geq \lambda$.

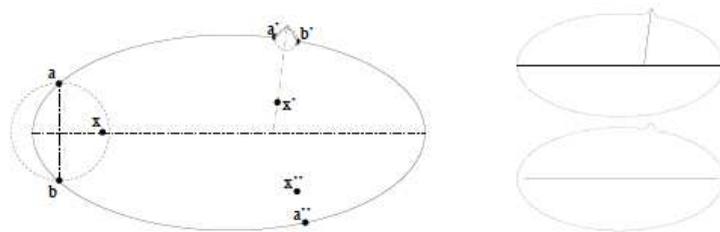


FIGURE 4.5: Illustration of the λ -medial axis. Left: Points x, x' and x'' and their respective closest border points. Top right: The λ -medial axis with $\lambda = \epsilon$, a very small positive real number. Bottom right: The λ -medial axis with $\lambda = d(a', b')/2 + \epsilon$.

Transposing the original definition of the λ -medial axis directly to \mathbb{Z}^n would lead to an unsatisfactory result as reported in (Chaussard et al., 2010) (see Fig. 4.6). In order to avoid such unwanted behavior, projection need to be replaced by so-called extended projection (Couprie et al., 2007).

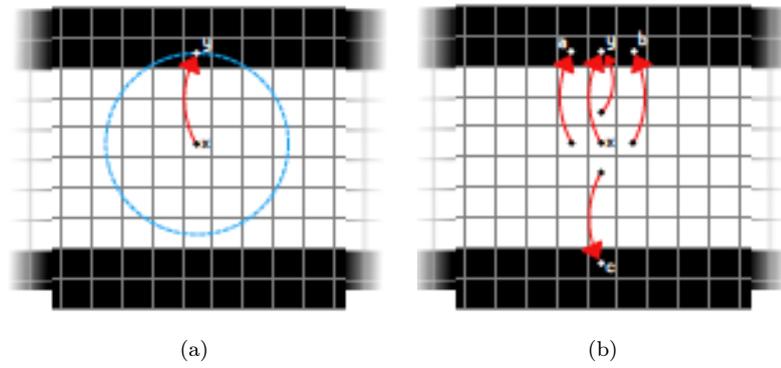


FIGURE 4.6: (a): The projection of the point x is the point y : by symmetry, all points of the ribbon would have a singleton as projection, producing an empty λ -medial axis for any $\lambda > 0$; (b): The extended projection of the point x is the set $\{y, a, b, c\}$: by symmetry, the points located near the center of the ribbon would all have an extended projection consisting of four points, leading to a non empty λ -medial axis. (Chaussard, 2010)

For each point $x \in \mathbb{Z}^n$, we define the *direct neighborhood* of x as $N(x) = \{y \in \mathbb{Z}^n \mid d(x, y) \leq 1\}$. The direct neighborhood comprises $2n + 1$ points.

Let $X \subseteq \mathbb{Z}^n$, and let $x \in X$. The *extended projection of x on \bar{X}* , denoted by $\Pi_{\bar{X}}^e(x)$, is the union of the sets $\Pi_{\bar{X}}(y)$, for all y in $N(x)$ such that $d(y, \bar{X}) \leq d(x, \bar{X})$.

We are now ready to introduce the definition of discrete λ -medial axis.

Definition 4.5 (Chaussard et al. (2010)). Let X be a finite subset of \mathbb{Z}^n , and let $\lambda \in \mathbb{R}^+$. We define the function \mathcal{F}_X which associates, to each point x of X , the value $\mathcal{F}_X(x) = R(\Pi_{\bar{X}}^e(x))$. The *discrete λ -medial axis (or DLMA)* of X is the set of points x in X such that $\mathcal{F}_X(x) \geq \lambda$.

This filtering strategy is very robust and gives very good results in general. However, it can be impossible to set proper value of the parameter λ if object consists of features at different scales (see an example on Fig. 4.7).

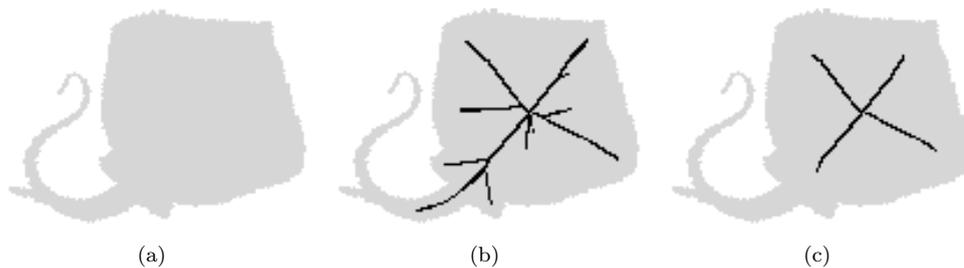


FIGURE 4.7: Example of the discrete λ -medial axis of the object consists features at different scale. (a): a shape X (in gray); (b): DLMA of X (in white). The medial axis is not sufficiently filtered in the middle of the shape. However, we already start to lose the tail; (c) A more filtered medial axis of X . Now, the middle of the shape is well filtered. However, we lost all information about the tail.

4.2.4 Filtering based on the bisector angle

Talbot and Vincent (1992) proposed a medial axis filtering based on a discrete version of the bisector function.

The bisector function of X is the function which associates to each point x of X , its bisector angle in X . The bisector angle of a point x in X can be defined, in the continuous framework, as the maximal unsigned angle formed by x and any two points in the projection of x on \bar{X} (Meyer, 1979; Talbot and Vincent, 1992) (see Fig. 4.8).

Definition 4.6. Let $X \subseteq \mathbb{Z}^n$. We denote by Θ_X the *bisector function* of X , defined by:

$$\Theta_X(x) = \max\{\widehat{yxz} : \forall y, z \in \Pi_{\bar{X}}(x)\}$$

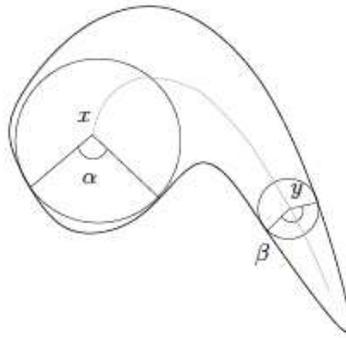


FIGURE 4.8: Illustration of the bisector angle. The bisector angle of x is α and the bisector angle of y is β . (Talbot and Vincent, 1992)

The point with higher bisector angle is considered as more important. Therefore to filter spurious medial axis point one should remove points with bisector angle below given filtering parameter (see Fig. 4.9).

Definition 4.7. Let $X \subseteq \mathbb{Z}^n$ and $MA(X)$ its Euclidean medial axis. Let α be a filtering parameter. We denoted by $BFMA(X, \alpha)$, the filtered medial axis defined by

$$BFMA(X, \alpha) = \{(x, r) \in MA(X) : \Theta_X(x) \geq \alpha\}$$

Methods which are based on the bisector function (Attali et al., 1995; Attali and Montanvert, 1996; Couprie et al., 2007) are effective in general. They can work with objects with features at different scales, since the bisector angle depends on ball position in the object and the size of the medial ball is not important.

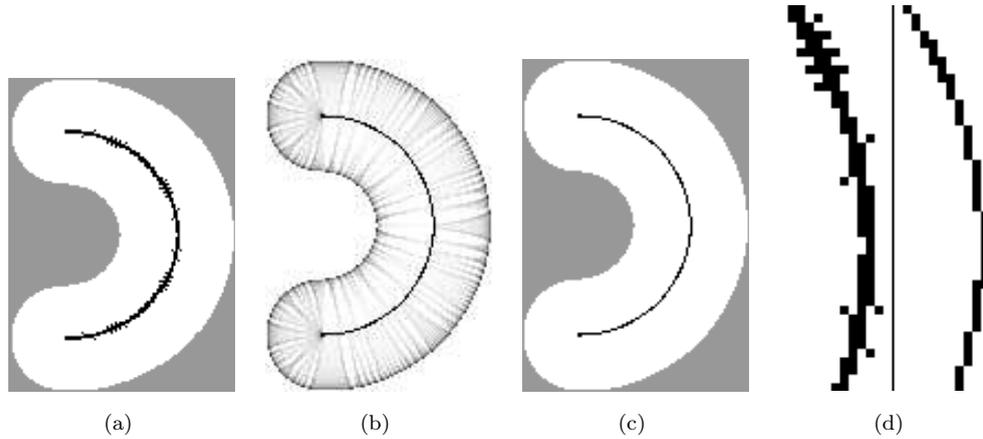


FIGURE 4.9: (a): a set X and its medial axis (in black); (b): the bisector function Θ_X (dark colors correspond to wide angles); (c): filtered medial axis, based on the values of Θ_X ; (d): detail of the non-filtered and filtered medial axis. (Talbot and Vincent, 1992)

4.2.5 Scale axis

In previous presented methods, except for ball covering, a local information (that is, geometric information extracted from a single medial ball) is compared to a global parameter value to determine the importance of the corresponding medial axis point. However, it is well known that this local filtering can lead to remove small branches which might be important for the shape understanding, especially for shapes with features at different scales (Attali et al., 2009).

In (Miklos et al., 2010), the authors address this issue and propose an approach that put in relation local information and regional information, that is, the status a ball is only influenced by the one of neighboring balls. Their method is based on the theory of the scale axis transform (Giesen et al., 2009), and defines a whole family of medial representations at different levels of abstraction, called *scale axis representations*. The original algorithm to compute scale axis, given by Miklos et al. (2010) was defined in the framework of union of balls in the continuous space. However it can be straightforwardly adapted to the case of \mathbb{Z}^n as presented in chapter 7.2.

For objects or scenes that include parts showing different scales, this method gives very good results in general (see Fig. 4.10). However, the scale axis representation is not free of drawbacks. The most important one is that the scale axis is not necessarily a subset of the Euclidean medial axis, it is even not necessarily a subset of the original shape. More generally, the family of all scale axes of a shape is not, in general, a hierarchy. This can strongly limit its practical application, since in most applications the medial axis or corresponding skeleton cannot be defined outside the original object. In chapter 7.3.1 we propose a solution to this problem and a new medial axis filtering strategy which holds hierarchy property.

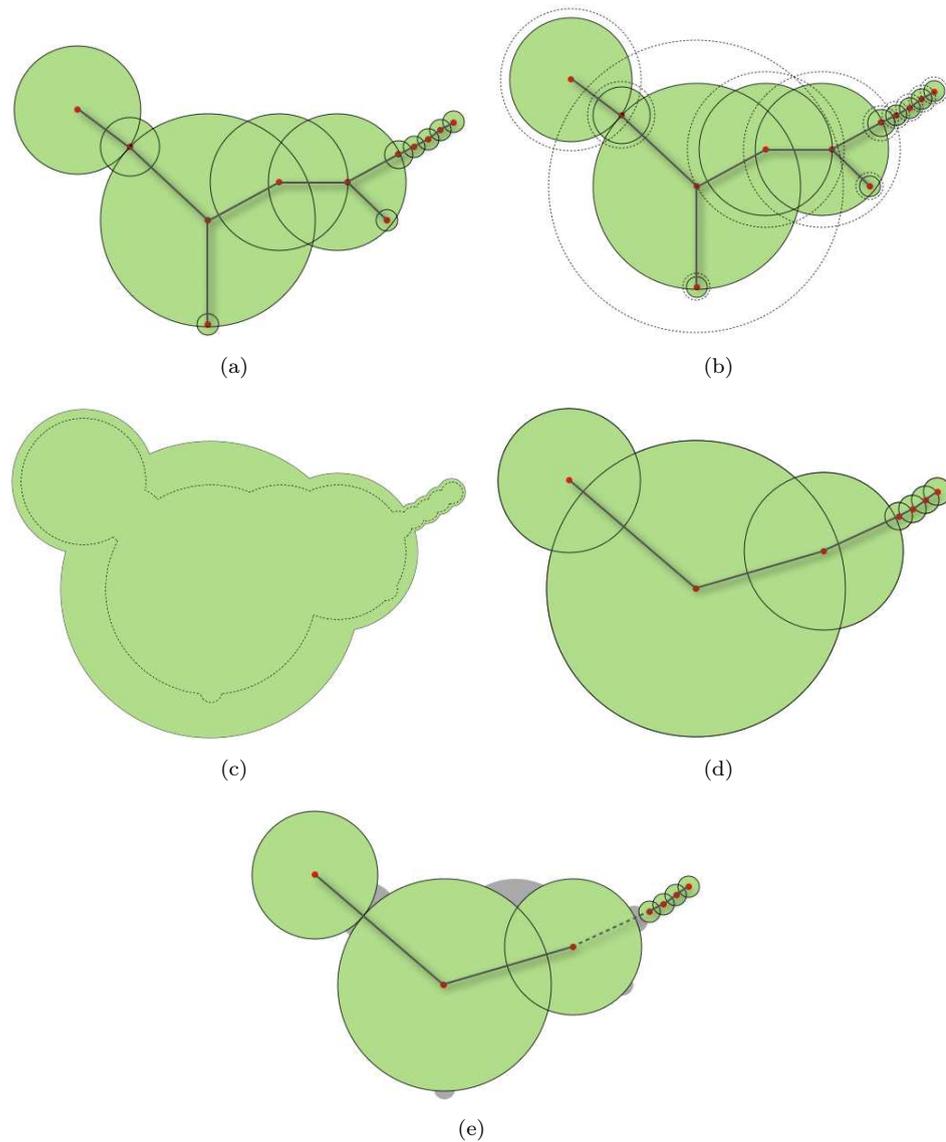


FIGURE 4.10: Example of the scale axis transform. (a): The input shape X (in green) and its Euclidean medial axis; (b): The multiplicatively scaled medial balls; (c): The object X_s reconstructed from the scaled medial axis. Note, that reconstructed object is much simpler than original shape; (d): The Euclidean medial axis of X_s ; (e): The scale axis of X . Note that the features at low scale were well preserved.

4.3 Euclidean Opening Function

Medial axis is not only used as a simplification of the object, since, it carries information about radius of the maximal balls which can be included in the object. For example [Hildebrand and R̃egsegger \(1997\)](#) propose a method to compute local thickness of the object based on its medial axis. Such method is called the Euclidean opening function or thickness distribution. In chapter 7.1 we will use this information to propose a new adaptive medial axis filtering strategy.

The Euclidean opening function associates to each point x of a shape X the radius of the biggest maximal ball $B(y, r)$ contained in X , such that, x is in $B(y, r)$ (see Fig. 4.11).

Definition 4.8 (Hildebrand and Regsegger (1997)). Let X be a finite subset of \mathbb{Z}^n , and let $(y, r) \in MA(X)$. The Euclidean opening function, denoted by, G_E^X is a function which associates, to each point x of X , the value

$$G_E^X(x) = \max\{r \mid \exists(y, r) \in MA(X), B(y, r) \subset X, x \in B(y, r)\}$$

Algorithm 5 shows a naive method to compute the Euclidean opening function.

Algorithm 5 Naive Euclidean Opening Function(**Input** X **Output** Y)

```

01.  foreach  $p \in X$  do  $Y(p) = 0$  end
02.   $MA(X) \leftarrow \mathbf{EuclideanMedialAxis}(X)$ 
03.  foreach  $(x_i, r_i) \in MA(X)$  do
04.    foreach  $p \in B(x_i, r_i)$  do
05.      If  $r_i > Y(p)$  then
06.         $Y(p) = r_i$ 
07.      end
08.    end
09.  end
10.  return  $Y$ 

```

The Euclidean opening function G_E^X can be effectively calculated in arbitrary dimensions thanks to notion of the power diagram as proposed in Coeurjolly (2010).

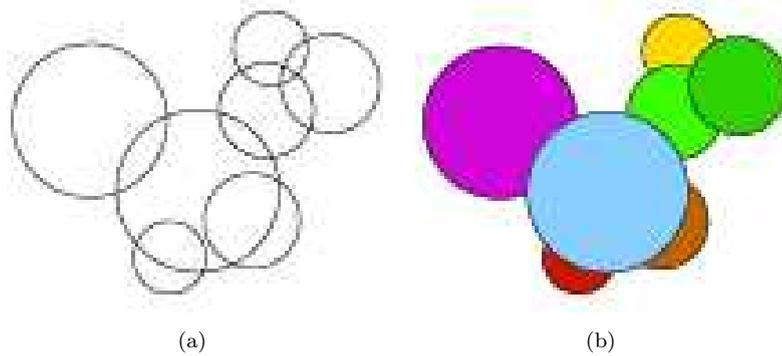


FIGURE 4.11: Illustration of the opening function. (a): an input set of maximal balls; (b): its Euclidean opening function. Bigger balls are on top of smaller ones (Coeurjolly, 2010).

Chapter 5

Visual and Topological Correction of Binary Shapes

The problem of mesh repairing and tunnel closing is considered in literature as part of a more general topic dealing with reparation of an object 3D model, which has many applications in laser scanning and bio-medical imaging. Therefore, the topic has led to many researches in the last decade and there are many articles presenting different approaches for 3D model reparation as presented in recent surveys ([Ju, 2009](#); [Breckon and Fisher, 2005](#)).

The related works can be divided into three groups. The first group represents methods of geometric hole (void) closing in a polygonal mesh defined as a set of vertices and a set of oriented polygons that join these vertices. Geometric holes (voids) in meshes are generally easily detected by searching for loops of boundary edges, that is, edges contained in only one polygon ([Bøhn and Wozny, 1992](#); [Mäkelä and Dolenc, 1993](#)). However, in some cases, especially when noisy objects are in interest, additional step, that is, manual or automated cleaning is necessary before loops can be properly detected ([Patel et al., 2005](#); [Pernot et al., 2006](#); [Lévy, 2003](#)). Once a closed boundary loop is identified the heuristic searches can be applied to incrementally triangulate the loop. For example, see methods proposed in ([Wagner et al., 2003](#); [Varnuska et al., 2005](#); [Barequet and Sharir, 1995](#); [Bøhn and Wozny, 1992](#); [Mäkelä and Dolenc, 1993](#); [Liepa, 2003](#); [Attene and Falcidieno, 2006](#)). More sophisticated methods use post-processing to obtain smooth continuation of the object surface in missing areas, see for example methods proposed in ([Zhao et al., 2001](#); [Pernot et al., 2006](#); [Liepa, 2003](#); [Zhao et al., 2007](#); [Bac et al., 2008](#); [Clarenz et al., 2004](#)) or methods based on adopting scattered data fitting techniques ([Wang and Oliveira, 2007](#); [L.S. Tekumalla, 2004](#); [Branch et al., 2006](#); [Chen and Cheng, 2008](#); [Kumar et al., 2007](#); [Jia and Tang, 2004](#)). There are also methods which restore surface relief in closing patches by finding and extending similar geometry in nearby regions for example ([Sharf et al., 2004](#); [Xiao et al., 2007](#); [Breckon and Fisher, 2005](#); [Bendels et al., 2005](#); [Park](#)

et al., 2006; Xu et al., 2006; Pauly et al., 2008). Sometimes the needed geometry is not represented in the input object itself, then the missing geometry is restored based on matching templates from a given library for example (Anguelov et al., 2005; Blanz and Vetter, 1999; Kähler et al., 2002; Allen et al., 2003; Blanz et al., 2004; Pauly et al., 2005; Savchenko and Kojekine, 2002; Yunbao et al., 2006). All cited methods represents geometric approaches to void closing and they do not work for closing tunnels which are considered as topological errors in 3D models (Jun, 2005). For example, a solid torus does not have any geometric holes in it as the triangulation of its outer surface does not have any boundary edges. On the other hand a tunnel closing algorithm does not close geometric holes in the outer surface of a 3D objects. Consider a (thin) sphere with a small disc removed. From the topological point of view there is no tunnel in such object.

The second group consists of methods which convert open polygonal surface into volumetric grid, where each point of the grid has positive or negative value denoting whether it is inside or outside the input object. then the signed grid can be easily converted into a waterproof triangulated outer surface which separates points of different signs, with the use of methods such as Marching Cubes algorithm (Lorensen and Cline, 1987; Nielson and Hamann, 1991; Montani et al., 1994; van Gelder and Wilhelms, 1994). The key issue in the procedure is to make proper generation of signed grid in areas nearby geometric holes in the input polygonal surface. Recent works in this domain have used the orientation of polygons in the input surface to obtain smooth void closing geometry. There are methods which classify each point of the grid based on the signed distance from oriented polygons for example (Curless and Levoy, 1996; Furukawa et al., 2007; Sagawa and Ikeuchi, 2008, 2003). Other methods perform propagation of the signed field from geometrically correct regions into void areas, where zero level set is the intended surface (Davis et al., 2001; Guo et al., 2006; Verdera et al., 2003). Signed distances can be also generated from line-of-sight direction of a scanner (Davis et al., 2001; Curless and Levoy, 1996). The methods from the second group do not close tunnels in objects which do not partition the space into inside and outside and, as stated by Ju (2009)(last paragraph), the task of tunnel closing for such type of objects is a challenge.

The last type of approach and the most interesting from our point of view is a method proposed by Aktouf et al. (1996, 2002). They constructed fast algorithm which closes tunnels directly in discrete topology framework taking advantages of local topological characteristic of points. This method could be applied with success for objects which do not partition the space into inside and outside for example 3D objects representing cracks in material (Janaszewski et al., 2011)(see Fig.5.1). Moreover, this approach can be applied directly to 3D objects obtained through a segmentation process from biomedical images, for example CT or MRI datasets.

During segmentation some unwanted holes can occur. These holes can be considered as noise, and may not be avoided by segmentation (see Fig.5.2(a)). On the other hand some tunnels are important to object understanding, for example anatomical features of

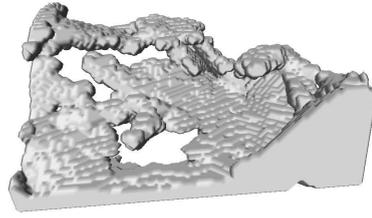


FIGURE 5.1: A rendering of a crack portion inside a material (material is represented by the background). The crack do not partition the space into inside and outside (Janaszewski et al., 2011). *Note:* The raw images from where this portion of a crack was segmented were acquired at the ESRF (ID19beamline). Details of the experiment can be found in Babout et al. (2006).

the human organs (see Fig.5.2(b)). Such kind of tunnels should be preserved. Therefore, tunnel closing algorithm should allow to automatically decide which tunnels should and which should not be closed. In algorithm proposed by Aktouf et al. (2002) such decision is made by analyzing size of the tunnel based on assumption that small tunnels are usually results of noise where large tunnels are important features of the object. However, the original algorithm can be applied to correctly close tunnels in bronchial trees as a part of segmentation procedure (see chapter 11.1). This approach is presented in more details in the next subsection.

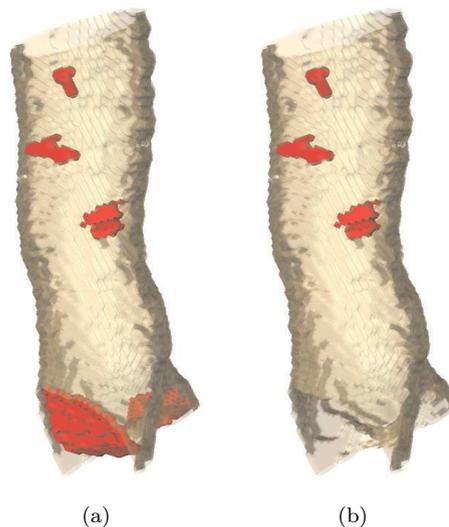


FIGURE 5.2: An example of tunnel closing algorithm to the object representing a part of trachea. Red patches correspond to closed tunnels. Notice that on (a) all tunnels where closed. However, on (b) a tunnel which is an anatomical feature is left open.

5.1 The Tunnel Closing Algorithm

In this section we present the procedure and properties of the original algorithm of tunnel (3D hole) closing presented in (Aktouf et al., 1996, 2002).

5.1.1 Topological hull

Let X be a subset of \mathbb{Z}^3 . We denote by $S(X)$ the class composed of all the subsets of \mathbb{Z}^3 containing X . Following [Aktouf et al. \(1996\)](#), first, we introduce the notion of topological hull of X and a binary relation on the set $S(X)$ which are used to define a tunnel closing algorithm.

Definition 5.1 ([Aktouf et al. \(1996\)](#)). Let $X \subset \mathbb{Z}^3$ and $Y \in S(X)$. We say that Y is a *topological hull* of X , if Y has no tunnels and no cavities and if, $\forall y \in Y \setminus X$, $Y \setminus \{y\}$ has a tunnel or a cavity.

The following theorem, proposed and proved in [Aktouf et al. \(1996\)](#), gives a local characterization of the class of sets which are topological hulls relatively to the class of sets which have no holes and no cavities:

Theorem 5.2 ([Aktouf et al. \(1996\)](#)). Let $X \subset \mathbb{Z}^3$ and $Y \in S(X)$. Suppose that Y has no cavities and no holes. Then Y is a topological hull of X if and only if for each y of $Y \setminus X$, y is an interior point or a 2D isthmus for Y , that is, if and only if $T_k(y, \bar{Y}) \neq 1$.

Direct consequence of the proof of [Theorem 5.2](#) is a corollary which leads to the following definition and method for extracting a topological hull of a set X .

Definition 5.3 ([Aktouf et al. \(2002\)](#)). Let $X \subset \mathbb{Z}^3$. We define the *binary relation* h on $S(X)$: $\forall U, V \in S(X)$, $V \in h(U)$ if $\exists u \in U \setminus X$ such that $V = U \setminus \{u\}$ and $T_k(u, \bar{U}) = 1$. Let $B \subset \mathbb{Z}^3$ with $X \subset B$ and $Y \subset \mathbb{Z}^3$. We say that $Y \in h^\infty(B)$ if there is a sequence $B = Y^0, Y^1, \dots, Y^n = Y$, such that, $Y^i \in h(Y^{i-1})$ and such that $h(Y) = \emptyset$. Let $B \in S(X)$ such that B has no cavities and no holes. If $Y \in h^\infty(B)$, then Y is a topological hull of X .

Now, to compute a topological hull of the set X , we first compute a bounding box B , that is, solid cube which has no cavities and no tunnels and which contains X . Then, we iteratively remove points of B which do not belong to X and which are not interior points and not 2D isthmuses. The procedure is repeated until stability, that is, until no more points can be removed.

5.1.2 Topological hull directed by the distance transform

As stated in [Aktouf et al. \(2002\)](#) in general a given object has several topological hulls which may be not centered. In [Fig. 5.3](#) an example of topological hulls is given and the

problem is shown. To avoid such drawback by eliminating tendency to build a topological hull which is not centered, a topological hull directed by the distance transform is introduced which leads to the new definition of topological hull as presented in [Aktouf et al. \(2002\)](#).

Definition 5.4 ([Aktouf et al. \(2002\)](#)). Let $X \subset \mathbb{Z}^3$ and let Ψ be a function from \mathbb{Z}^3 to \mathbb{N} . We define the binary relation h on $S(X) : \forall U, V \in S(X), V \in h(U)$ if $\exists u \in U \setminus X$ with $T_k(u, \bar{U}) = 1$ such that:

- (i) $V = U \setminus \{u\}$ and
- (ii) $\Psi(u) = \max\{\Psi(y), \forall y \in U \setminus X \text{ with } T_k(y, \bar{U}) = 1\}$

Let $B \in S(X)$ be a set with no holes and no cavities. if $Y \in h^\infty(B)$, we say that Y is a Ψ -topological hull of X .

In the sequel of this work the function Ψ is set to the squared (Euclidean) distance transform of X . The distance transform is used as a constraint in order to restrict the choice of the possible points to process in each iteration. The points are sorted according to their decreasing distance from the object. Now the procedure to calculate topological hull can be applied. The furthest points from the object are going to be processed first. By processing the points in such way, the points which are going to survive to the elimination process will be the closest to the object. Therefore the obtained topological hull is centered according to the object.

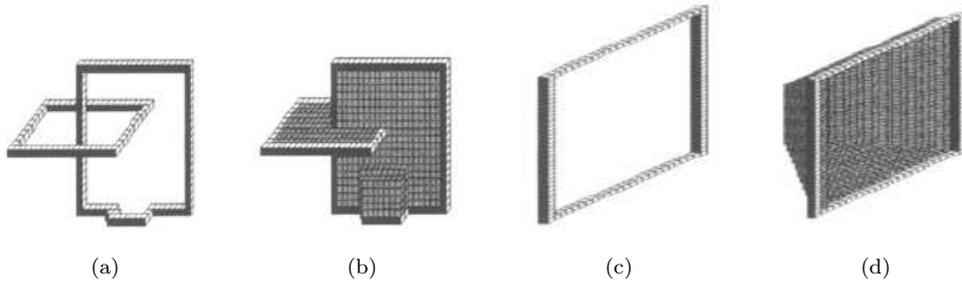


FIGURE 5.3: Examples of topological hulls of the object. (b) and (d) are, respectively, topological hulls of (a) and (c). Notice that topological hull (d) is not centered. ([Aktouf et al., 1996](#))

5.1.3 ϵ -closing algorithm

The definition of the topological hull directed by distance transform (Definition 5.4) gives also another possibility to control the resulting topological hull, that is, the ability to select the tunnels which are going to be closed. By setting a parameter called distance threshold denoted by ϵ we can simply remove all points which have distance to the object

greater than ϵ . Such strategy is called ϵ -closing. Examples with tunnel closing according to different threshold values are presented on Fig. 5.4.

Algorithm 6 proposed in [Aktouf et al. \(1996, 2002\)](#) allows one to compute topological hulls directed by distance transform image Ψ . It also allows one to define maximal size of tunnels which should be closed by setting the distance threshold ϵ . It takes as an input the object X and its bounding box B and the results is a centered topological hull Y . In the algorithm a binary image E which associates to each point a boolean flag (TRUE or FALSE) is used to let us know if the corresponding point was already processed. Points which are processed in each iteration are stored in hierarchical list L in decreasing order of distance.

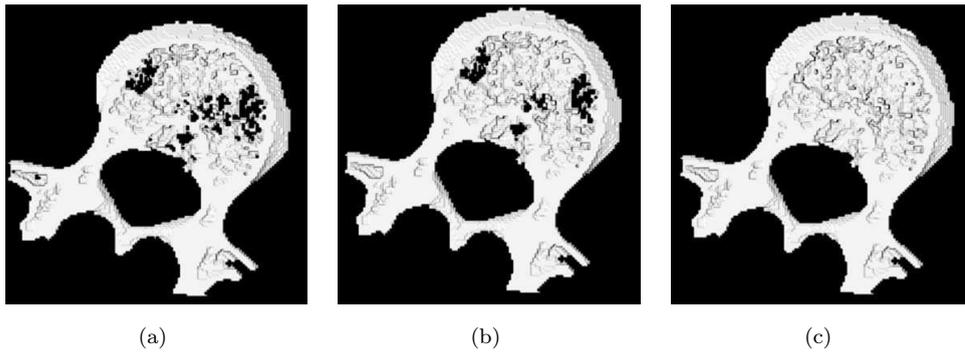


FIGURE 5.4: Application of the 3D tunnels closing algorithm to a 3D object with different ϵ . (a): initial image; (b): $\epsilon = 5$; (c): $\epsilon = 9$. ([Aktouf et al., 2002](#))

Algorithm 6 Tunnel Closing Algorithm(**Input** X, B, Ψ, ϵ **Output** Y)

```

01.    $Y = B$ 
02.   foreach  $x \in \overline{X} \cap B$  such that  $\Gamma_k^*(x) \cap \overline{B} \neq \emptyset$  do
03.        $L[\Psi(x)] = L[\Psi(x)] \cup \{x\}$ 
04.        $E(x) = \text{TRUE}$ 
05.   end
06.    $i = \text{Max}_{x \in B} \Psi(x)$ 
07.   while  $i \geq 0$  do
08.       while  $L[i] \neq \emptyset$  do
09.            $x = \text{FIRST}(L[i])$ 
10.            $E(x) = \text{FALSE}$ 
11.           if  $(T_{\overline{k}}(x, \overline{X}) = 1)$  or  $(\Psi(x) > \epsilon)$  then
12.                $Y = Y \setminus \{x\}$ 
13.               foreach  $y \in \Gamma_{26}^*(x) \cap \overline{X} \cap Y$  such that  $E(y) = \text{FALSE}$  do
14.                    $E(y) = \text{TRUE}$ 
15.                    $L[\Psi(y)] = L[\Psi(y)] \cup \{y\}$ 
16.                   if  $(\Psi(y) > i)$  then  $i = \Psi(y)$  end
17.               end
18.           end
19.       end
20.        $i = i - 1$ 
21.   end
22.   return  $Y$ 

```

5.2 Properties and Limitations of the Tunnel Closing Algorithm

Algorithm 6 is linear in time and space complexity. It builds a topological hull of the input object which closes all tunnels which are smaller than a given diameter. In some applications, ϵ parameter can be considered as optional and can be omitted. In such case all tunnels will be closed. However, the tunnel closing algorithm closes not only tunnels but also cavities in an input image. In some applications this property of the tunnel closing algorithm is very useful especially when cavities are unwanted in an output image for instance when they are the results of acquisition errors. However, cavity filling can also be considered as a drawback where only tunnels should be closed. To simply eliminate this drawback, fast cavities filling algorithm can be applied before using tunnel closing. The cavities filling algorithm gives useful information where all cavities occur in an input image, which can be simply used to let cavities preserved if necessary (Janaszewski et al., 2011).

Another property of the tunnel closing algorithm is ability to produce centered topological hull in respect to used priority function, for example the Euclidean distance map may be a good choice as priority function in general case. However, the result may not be suitable for some applications. As example, consider a three-dimensional tubular object X (see Fig. 5.5(a)). Such object has two tunnels which will be closed by the tunnel closing algorithm. If we use the Euclidean distance map of X as priority, the tunnel closing patches will be situated in the middle of the tubes (see Fig. 5.5(b)) or in the most narrow part of the object if we introduce some distortion. However, by changing the priority we can cause the tunnel closing algorithm to close tunnels, for example in different parts of the object (see Fig. 5.5(c)).

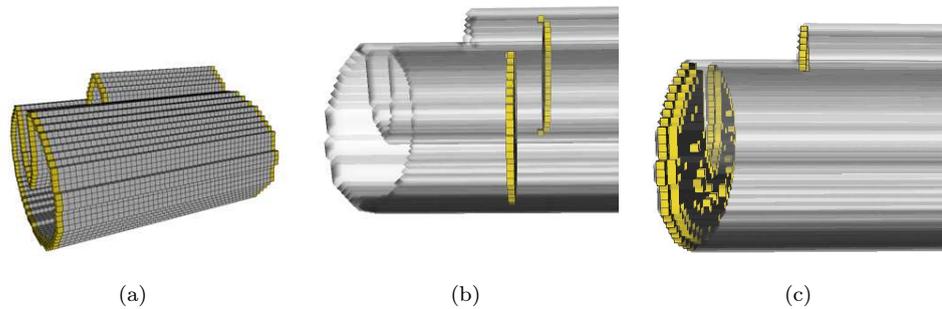


FIGURE 5.5: Application of the 3D tunnels closing algorithm to tubular object. (a): Initial image with two tunnels marked in yellow; (b, c): Results of working of the tunnel closing algorithms using different priority function. Closing patches are marked in yellow.

The algorithm is also sensitive to other type of distortions. This drawback occurs, for example, when in an object some branches or object parts are situated in the close vicinity of a tunnel. In this case the shape of the tunnel closing patch may be significantly

influenced by these branches. Such situation is presented in Fig. 5.6. This artificial object presents one large tunnel in the middle, surrounded by three branches. After applying the tunnel closing algorithm, the tunnel closing patch, which is represented with dark grey color, goes up to the branch, so it does not correspond to the "geometry of the tunnel" (that is nearly planar). It is important when we further want to analyze e.g. the geometry of the tunnel closing patch. See for example (Janaszewski et al., 2009a, 2010, 2011), for more detailed discussion about this limitation and improvements designed to solve this particular problem.

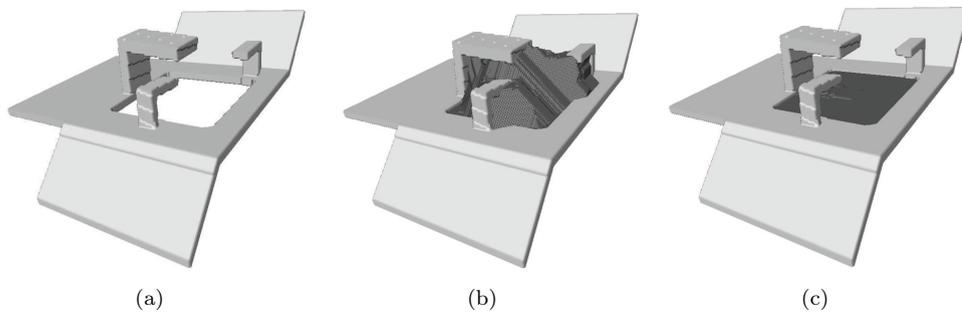


FIGURE 5.6: Application of the 3D tunnels closing algorithm to object with irrelevant branches close to the tunnel. (a): initial image; (b): the tunnel closing patch is distorted by branches; (c): corrected result using strategy described in Janaszewski et al. (2011)

The tunnel closing algorithm is designed to correct only topological errors in an input object. Therefore, it is not suitable to correct geometrical errors unless they are caused by presence of tunnels or cavities. For example we can consider an object representing a three-dimensional sphere. We can alter its geometry by removing some connected set of points (see Fig. 5.7). In such case there is no tunnel in our object. Therefore, tunnel closing algorithm will not correct our object. In such application different algorithm needs to be considered.

We can also notice the problem if we want to close some tunnels in a connected component which is nested inside another one. For example any object with tunnels which is inside empty cube. In such case, for example, some preprocessing step is necessary to extract interesting connected component from the image before applying tunnel closing.

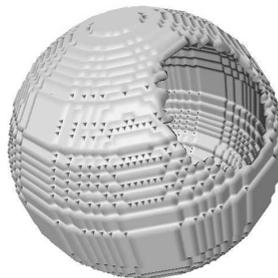


FIGURE 5.7: Example of case when the tunnel closing algorithm cannot be used.

5.3 Application of The Tunnel Closing Algorithm

The tunnel closing algorithm has several documented potential applications in materials science and medicine. An important example is stress corrosion cracking in stainless steel (Scott, 2000). X-ray microtomographic studies have observed the initiation and propagation of intergranular cracks along the grain boundaries of this polycrystal material (Babout et al., 2006). These observations have revealed local ligaments of material which span the developed crack, and create tunnels in the crack path (see Fig. 5.8). These ligaments are called bridges, and generally correspond to special grain boundaries with higher resistance to damage. The measurements of the spatial distribution and the size of these bridges during in situ observations of cracking is one of the important characterizations to carry out towards the prediction and optimization of the material resistance to stress corrosion cracking (Marrow et al., 2006). Detection and analysis of bridges has usually been done manually by visual inspection of 3D data set. Such a strategy is time consuming and potentially unreliable, so an automatic procedure of tunnel detection for volumetric images seems to be a valuable alternative.

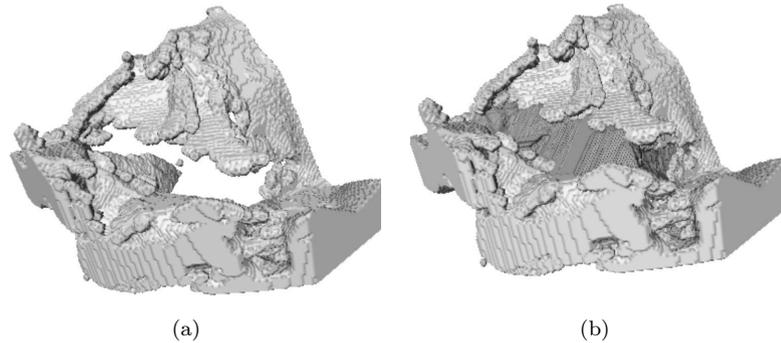


FIGURE 5.8: An example of tunnel closing algorithm. (a): the crack. There are three tunnels (a large one and two small ones) inside the crack to be closed; (b): the crack with tunnels closed. Tunnel closing patches, with one voxel thickness, are represented with dark gray color (Janaszewski et al., 2011). *Note:* The raw images from where this portion of a crack was segmented were acquired at the ESRF (ID19beamline). Details of the experiment can be found in About et al. (2006).

Another useful application of tunnel closing is for cellular materials. It can be used to close defects in foam walls, but it can also be used, as shown in Fig. 5.9 to close joints-ribs cycles in open cell foams. This typical example corresponds to an auxetic polyurethane foam, a new generation of foam that has a tendency to expand laterally when stretched and contract laterally when compressed. Recent studies have intended to better understand the bulk tensile deformation response of auxetic foam in situ by coupling X-ray tomography with digital image correlation (McDonald et al., 2009). This has also enabled to estimate for each joint that composes cycles the rotation matrix that needs to be applied in order to recover the standard polyurethane foam microstructure, that is, cycle with a perfect polygonal shape. The tunnel closing algorithm is very useful to identify the cycles and to represent 3D maps of deviation to perfect polygon. In the

present case shown in Fig. 5.9, the tunnel closing was performed after 3D curvilinear skeletonisation of the structure and each cycle was closed separately after identifying each vertices and edges of the skeleton.

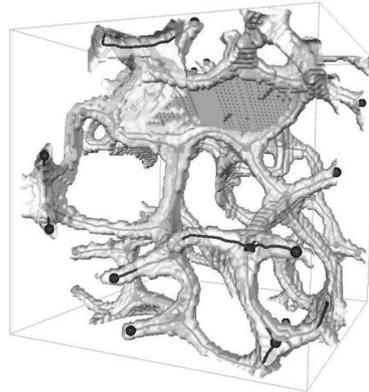


FIGURE 5.9: An example of tunnel closing algorithm. 3D rendering of an auxetic foam (in light gray) on which the corresponding skeleton is superimposed (dark curves: edges - dark spheres: vertices). An open cell is also closed (depicted by a dark gray circle).

An example of medical application of the tunnel closing algorithm is a aortic dissection problem. According to [Lohou and Miguel \(2011\)](#) it is a medical emergency which may lead to death. Aortic dissection is caused by aortal tissue perforation due of blood pressure. Such perforation causes presence of tears or holes in the intimal tissue inside lumens (see Fig. 5.10). They are usually detected by radiologists by examining successive CT slices. Manual detection is a very difficult and error-prone task. Using the tunnel closing algorithm these intimal tears can be detected automatically or semi-automatically and their quantitative properties can be evaluated which can help cardiac surgeons in making diagnosis.

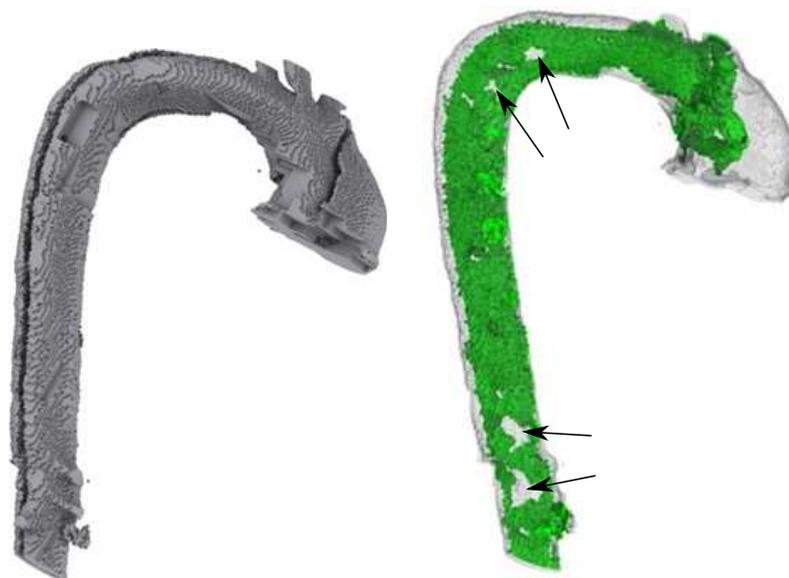


FIGURE 5.10: (a): 3D rendering of the aorta (in light gray); (b): Intimal flap inside the aorta on which the intimal tears (holes) are pointed by arrows. (Lohou and Miguel, 2011)

Chapter 6

Extracting Geometrical Properties of Binary Shapes

One classical problem in discrete geometry is to estimate geometrical properties of digitized shapes without any knowledge of the underlying continuous shape. Algorithms used to perform such estimation are called *digital geometric estimators*. Many vision, image analysis and pattern recognition applications rely on the estimation of the geometry of the discrete shape or the curve which can be either a contour of the shape or result of skeletonization process. Therefore, the accurate estimation of geometric parameters is an essential step in many practical applications and it is an active research field.

According to [De Vieilleville and Lachaud \(2008\)](#) estimating geometrical quantities on digitalized shapes is a difficult task for at least four major reasons:

- There exist infinitely many continuous Euclidean shapes that have the same digitization. This problem implies that, it is impossible to estimate proper geometrical quantities of the underlying Euclidean shape without additional hypotheses about reference shape (see [Fig. 6.1](#)). For example properties such as smoothness, compactness, convexity, minimal perimeter or maximal area are common choices.
- Determining the required size of a computational window around a point of interest to achieve a good estimation is not straightforward. Moreover, size of a computational window has a strong impact on multigrid convergence property of the estimator (see [de Vieilleville et al. \(2007\)](#) and [section 8.4.1](#)). The size of a computational window should be adapted to local shape geometry, for example parts of the shape with high curvature variation need relatively small computational window.
- The digitized shape can be noisy or damaged. This is a common problem, especially when analyzed shapes are results of image segmentation algorithms or skeletonization strategies (see [Fig. 6.2](#)). This problem is addressed and efficient solutions are

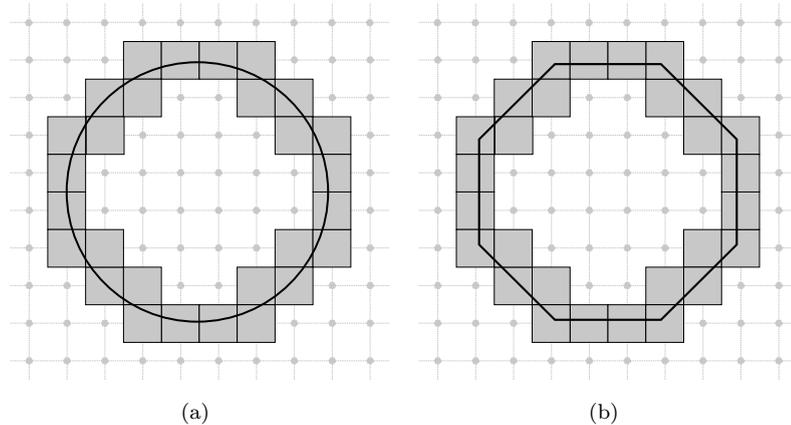


FIGURE 6.1: Example of two objects with exactly the same digitalization. (a): Euclidean disk and its digitalization; (b): Euclidean octagon and its digitalization.

proposed in continuous framework, but lacks proper definition in the discrete framework. Presence of noise in digitalized shapes is the reason why continuous methods are generally preferred for extraction of geometric properties.

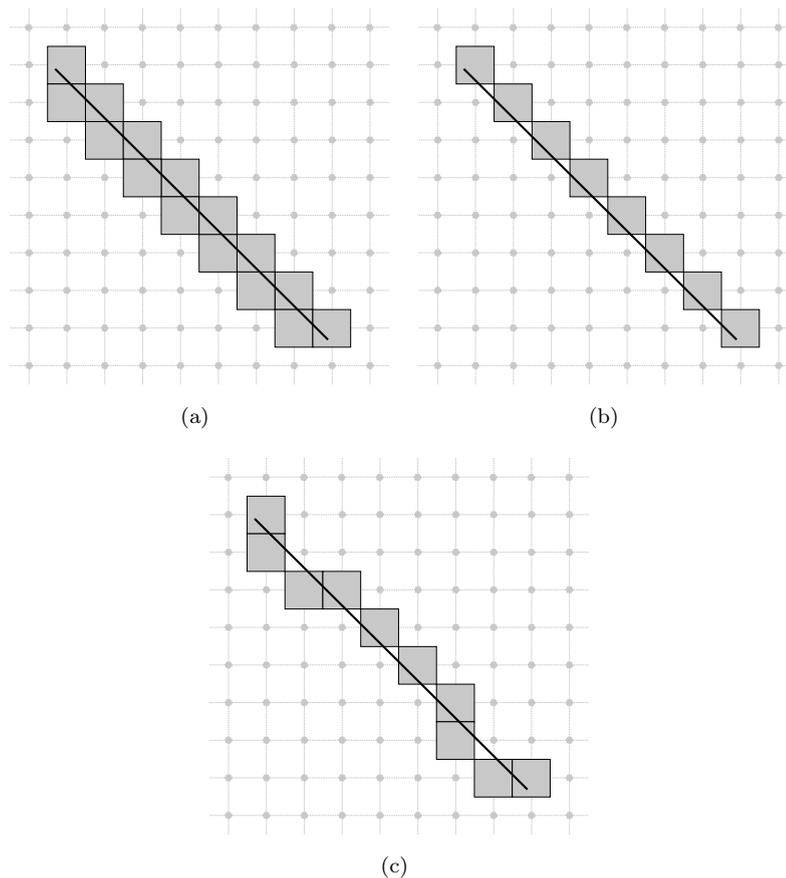


FIGURE 6.2: Example of presence of noise in the curve. (a): Euclidean straight line and its digitalization; (b, c): Two possible skeletonisation results of (a). Note that (c) can be considered as not "straight".

- The time spent on computations may be limited. This problem occurs when estimators based on optimization techniques are used, or when the computational window is too large. On the other hand, small size of computational window (which may speed up the computation) has a direct negative impact of the precision of the estimator.

In the literature numerous algorithms were proposed to estimate different geometrical properties. They can be divided into two main groups. A first group consists of algorithms which estimate the global geometrical features for example area and perimeter (Yang et al., 1994; Kulpa, 1977; Gray, 1971; de Vieilleville and Lachaud, 2009; Klette and Rosenfeld, 2004), moments (Yang and Albregtsen, 1994a,b, 1996) or length (Dorst and Smeulders, 1987; Tajine and Daurat, 2003; Coeurjolly et al., 2001a; Stojmenovic and Tasic, 1991). The second group consists of local geometrical features estimators like normals (Lenoir et al., 1996), tangents (presented in more details in this chapter), or curvatures.

In the three-dimensional space important local geometric properties are curvature and torsion. Several methods have been proposed for curvature estimation, which can be divided in three categories according to (Worring and Smeulders, 1993; Vialard, 1996a; Hermann and Klette, 2007). In the first group are methods based on the inverse of the osculating circle radius (Coeurjolly et al., 2001b; Coeurjolly and Svensson, 2003; Hermann and Klette, 2007; fleischmann et al., 2010). The second group consists of methods which try to reconstruct locally the curve with some polynomial of given degree (Marji, 2003; Hermann and Klette, 2007) or try to find the curve, among all with the same digitalization, which minimizes its squared curvature (Kerautret and Lachaud, 2009). The third group estimates curvature based on the convolution of the curve by some derivative of Gaussian kernel (Worring and Smeulders, 1993; Vialard, 1996a; Malgouyres et al., 2008; Esbelin and Malgouyres, 2009; Fiorio et al., 2010). Recently proposed methods, which belong to the first group, based on notion of maximal digital circular arcs (Roussillon and Lachaud, 2011) which is a natural extension of the notion of the maximal digital straight segments (presented later in this chapter in section 6.3.2).

Several methods for torsion estimation were also proposed. It can be calculated directly from torsion formula based on Gaussian (Mokhtarian, 1997) or B-spline smoothing (Kerhtarnavaz and de Figueiredo, 1988). Another methods are based on least-squares fitting techniques (Lewiner et al., 2005; Medina et al., 2004).

Local geometric quantities can also be used naturally as methods for the detection of dominant points on discrete curves (Ray and Pandyan, 2003; Marji, 2003; Kerautret et al., 2008). However, in this thesis we focus mainly on tangent estimators along discrete curves, since, tangent directions at each point of a discrete curve give us very useful information. In the three-dimensional space the normal vector field can be estimated from tangent directions (Lachaud and Vialard, 2003; Lenoir et al., 1996). We can accurately estimate

length of the curve by simple tangent integration (Lenoir et al., 1996; Coeurjolly and Klette, 2004). Derivating the tangent direction provides an estimation of the curvature (Worring and Smeulders, 1993; Vialard, 1996b; Lenoir et al., 1996). We utilize tangent to compute cross sections of the airway tree, which allow for reliable measurements (see chapter 13).

Different tangent estimators have been already studied in the literature. In the framework of digital geometry, there exist few studies on 3D discrete curves, while there are numerous methods performing in 2D. However, all methods can be categorized into two main groups, that is, the continuous tangent estimators (chapter 6.2) and discrete tangent estimators (chapter 6.3).

6.1 Discrete Curves

Several authors have proposed definitions of discrete curves, for example (Kiryati and Kubler, 1992; Jonas and Kiryati, 1997). However, not all of them are suitable for our purposes. In this thesis we utilize curves which are result of skeletonization process, either in two-dimensional or three-dimensional space. Therefore, we based our work on the following definition of a discrete curve:

Definition 6.1. Given $X \subset \mathbb{Z}^n$, any set $S \subseteq X$ such that $|S| \geq 2$ is called a k -curve iff there are exactly two points $p, q \in S$ such that $|\Gamma_k^*(p) \cap S| = |\Gamma_k^*(q) \cap S| = 1$, and for any $x \in S \setminus \{p, q\}$ we have $|\Gamma_k^*(x) \cap S| = 2$.

Let C be a k -curve, where $k \in \{4, 8\}$ or $k \in \{6, 26\}$ in 2D and 3D respectively. Based on selected connectivity k we can distinguish two types of discrete curve C :

- for $k = 4, 6$ - C is called the *standard discrete curve* or *standard digital curve*
- for $k = 8, 26$ - C is called the *naive discrete curve* or *naive digital curve*

There are methods which can work only with one type of discrete curves. To simplify writing in the following parts of this work we assume that the adjacency relation k in case of \mathbb{Z}^2 equal to 8 and equal to 26 in case of \mathbb{Z}^3 , has been chosen and we do not write the prefix k , unless necessary. Therefore, it means that presented algorithms are capable to work with the naive discrete curves.

All points in a discrete curve C can be totally ordered to form a sequence of points (p_1, \dots, p_ℓ) such that p_1 is the first point of C and p_ℓ is the last point of C , where $\ell = |C|$. We denote by C_i the i -th point of C where $1 \leq i \leq |C|$. If $C_1 = C_\ell$ we say that curve is closed, or open otherwise.

6.2 Continuous Tangent Estimators

First group of tangent estimators along discrete curves consists of algorithms which are based on approximation techniques in the continuous Euclidean space. The continuous methods can be of various types, for example they can rely on interpolation, reconstruction or fitting techniques. The numerical methods, required to extract the chosen solution can be costly and use non trivial optimization and may even require to set proper parameters themselves.

The most common continuous methods are based on least square fitting using polynomials (Lewiner et al., 2005; Cazals and Pouget, 2005). The aim of these methods is to find a polynomial of finite degree which minimizes a positional squared error from a set of samples (de Vieilleville and Lachaud, 2009).

For example, in case of planar curve parametrized as $y = f(x)$, we can obtain a set of M samples, denoted by $(s_i = (x_i, y_i))_{1 \leq i \leq M}$. Then we need to minimize the functional

$$E(a_0, \dots, a_N) = \sum_{i=1}^M (y_i - \sum_{j=0}^N a_j x_i^j)^2$$

At least one solution exist and can be efficiently computed using QR factorization (Press et al., 1992). Once the optimal polynomial for E is determined, it can be used to estimate the tangent orientation.

In the practical applications to estimate tangent orientation in the point of interest C_0 considered as the origin. We set computational window ranging from C_{-q} to C_q and use low order polynomials. For example, the linear regression, implicit parabola fitting and explicit parabola fitting are common choices.

Least square fitting method can be extended by adding some variable importance w_i in the fitting process, to each sample s_i . A sample with heavier weight is more important to fit.

$$E(a_0, \dots, a_N) = \sum_{i=1}^M w_i (y_i - \sum_{j=0}^N a_j x_i^j)^2$$

Such method is called weighted least square fitting. However, the biggest drawback of this method is an assumption that the weights are known exactly. In the real application the exact weights are almost never known, so they must to be estimated which is not easy in general.

Another tangent estimator which is an extension to least square fitting is called independent coordinates fitting (de Vieilleville and Lachaud, 2009). In this estimator the fit is done on each coordinates with respect to a given parametrization of the curve.

Let C_0 is a point of interest and $M = 2q + 1$ is the size of the computational window, range from C_{-q} to C_q . We also define the arc-length l_i from C_0 to C_i computed as $l_i = \sum_{k=0}^{i-1} \|C_{k+1} - C_k\|_1$ if $i > 0$ or $l_i = -\sum_{k=0}^{i-1} \|C_{k+1} - C_k\|_1$ otherwise. Now, we minimize the two following quantities:

$$\sum_{i=1}^M (x_{C_i} - \sum_{j=0}^N a_j l_i^j)^2$$

$$\sum_{i=1}^M (y_{C_i} - \sum_{j=0}^N b_j l_i^j)^2$$

However, as stated in de Vieilleville and Lachaud (2009), independent coordinates fitting does not yield much better results than presented above.

The second commonly used continuous tangent estimators are method based on Gaussian filtering (Mokhtarian and Mackworth, 1986; Witkin, 1983). The Gaussian filters has been used in pattern recognition for almost 30 years for improving the quality of noisy images. When applied to discrete curves they produce its smoothed continuous approximation. Then its first and second order derivatives can be easily computed which yields to simple geometric quantities estimation.

We denote by C'_0 the estimated derivative at the discrete point C_0 defined as

$$C'_0 = \sum_{i=-q}^q G'_{\sigma_q}(-i) C_i$$

where $G'_\sigma(t)$ is the first derivative of the Gaussian function

$$G_\sigma(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{t^2}{2\sigma^2}\right)$$

The biggest problem when using this class of method is a choice of the parameter σ . Which is often chosen globally for the whole curve. However, it is well known that setting this parameter globally may lead to over-smoothing for some regions and under-smoothing for others, especially for curves with large curvature variations.

6.3 Discrete Tangent Estimators

The second group of tangent estimators consist of methods which work in the discrete space directly. They are usually more sensitive to presence of noise in the input shape than continuous methods. However, discrete methods are usually much faster than approximation methods, and they do not need to set any parameter. Only in some algorithms the size of computational windows need to be defined. We can consider such methods as *local discrete tangent estimators*. On the other hand, there exists a set of local tangent estimators where size of computational window is determined automatically and can adapt to local geometry of the shape. Such methods are called *local adaptive discrete tangent estimators*.

In the following subsection we present basic concepts of the discrete tangent estimators.

6.3.1 Local discrete tangent estimators

The local discrete tangent estimators evaluate tangent direction along all points of the discrete curve using a finite set of curve points around the point of interest. The size of such a finite set of neighboring points is fixed globally by users.

For example, consider a discrete curve C . A naive method can use only one neighboring point to the actual point C_i which can be either point C_{i-1} or C_{i+1} to estimate tangent direction. In such case the tangent direction in C_i is derived from a vector pointing from point C_i to C_{i-1} or C_{i+1} (see Fig. 6.3).

Such simple strategy is based on differential geometry where the tangent direction t of a curve $c(x)$ at the point $P_{x_0} = c(x_0)$ is defined as:

$$c'(x_0) = \lim_{h \rightarrow 0} \frac{c(x_0 + h) - c(x_0)}{h} = \lim_{h \rightarrow 0} \frac{\overrightarrow{P_{x_0} P_{x_0+h}}}{h}$$

Therefore, in discrete space, given the discrete curve C , the tangent direction at the point C_i can be estimated as:

$$t(\vec{i}) = \frac{c(i+1) - c(i)}{1} = \frac{\overrightarrow{C_i C_{i+1}}}{1} = \overrightarrow{C_i C_{i+1}}$$

The above naive method gives very poor results since it is extremely sensitive to noise and it has very small finite number of possible results. Therefore it has very limited practical applications.

To achieve better accuracy and robustness to noise, in general, methods proposed in the literature utilize a finite set of $2n + 1$ curve points around the point of interest [Matas](#)

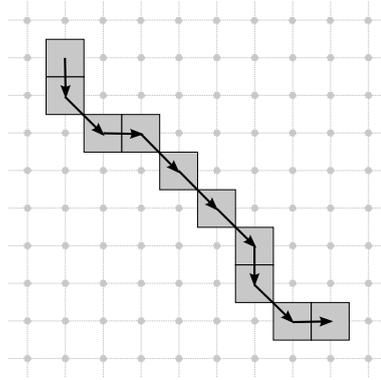


FIGURE 6.3: Example of naive tangent estimator. Each arrow correspond to the local tangent direction.

et al. (1995); Worring and Smeulders (1993); Anderson and Bezdek (1984); Lenoir et al. (1996). In the sequel we call such set *computational window*.

For example, a more robust to noise method can use 3 points, that is, both previous and next point plus the actual one, and then tangent direction in point C_i is derived from a vector pointing from point C_{i-1} to C_{i+1} .

Matas et al. (1995) compute the tangent direction as the median direction of the vectors linking the point C_i to the j next points and to the j previous points on the curve C . Such strategy permits to avoid small noise impact in the discrete curve since it averages tangent directions (see Fig. 6.4). Increasing number of points lead to stronger averaging of results but it loses accuracy and has a very poor behavior at sharp corners.

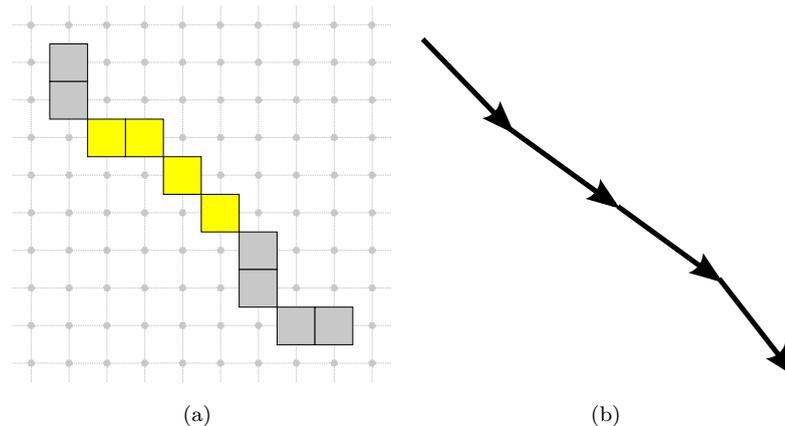


FIGURE 6.4: Example of Matas et al. (1995) local discrete tangents estimator. (a): The curve with points (in yellow) in which the tangent direction where evaluated; (b): The results of tangent estimation (we set parameter $j = 2$). Note that almost straight line where achieved.

The local discrete tangent estimators discussed in this chapter are very fast and simple. However, they have two important drawbacks. The first one is the need to set properly the size of the computational window. The second one is the consequence of the first one, that is, these methods cannot converge asymptotically (see section 8.4.1) to the

continuous shape because the computational window is fixed globally and is not adapted to the local shape geometry. Therefore, these methods can be used when the accuracy of the estimator is not critical.

6.3.2 Local adaptive discrete tangent estimators

Local adaptive discrete tangent estimators are free of the drawbacks presented above. The methods of this class can automatically adapt to the local shape geometry obtained by recognizing digital straight line segments around the point of interest (Feschet and Tougne, 1999; de Vieilleville and Lachaud, 2009; Lachaud et al., 2005a). According to (Lachaud et al., 2005b) exact tangent estimation for digitizations of straight lines can thus be achieved. Note that most of presented methods are designed for 2D discrete curves. Therefore, some of them cannot work straightforwardly with 3D curves. Even if the extension looks straightforward they often have unexpected behavior for some special configurations of curve points or they do not preserve all properties.

Local adaptive tangent estimator along discrete curves relies on the following arithmetical definitions of digital lines (Coeurjolly et al., 2000; Debled-Rennesson and Reveillès, 1995).

Definition 6.2 (Réveilles (1991)). A 2D digital line, with direction vector $(b, a) \in \mathbb{Z}^2$, shift μ and thickness e , where $a, \mu \in \mathbb{Z}$ and $b, e \in \mathbb{Z}_+$ such that $\text{nwd}(a, b) = 1$, is defined as the set of points $(x, y) \in \mathbb{Z}^2$ which satisfy the diophantine inequality:

$$\mu \leq ax - by < \mu + e,$$

and is denoted by $\mathcal{D}_2(a, b, \mu, e)$.

Local adaptive methods make segmentation (see Fig. 6.5) of the discrete curve into a set of maximal discrete straight segments which are defined as follows:

Definition 6.3. Given a discrete curve C , a set of its consecutive points $C_{i,j}$ where $1 \leq i \leq j \leq |C|$ is said to be a digital straight segment (DSS) iff there exists a digital line \mathcal{D}_2 containing all the points of $C_{i,j}$. In this case, we say that the property $S(i, j)$ holds.

Definition 6.4. Any subset $C_{i,j}$ of C is called a maximal segment iff $S(i, j)$ and $\neg S(i, j+1)$ and $\neg S(i-1, j)$.

Based on local DSS recognition, several tangent estimators at a digital curve point have been proposed. The main difference between them is a different balance between longest and most centered DSS around the point of interest. The following DSSs and their corresponding tangent estimators may be defined around any point C_i of the digital curve C :

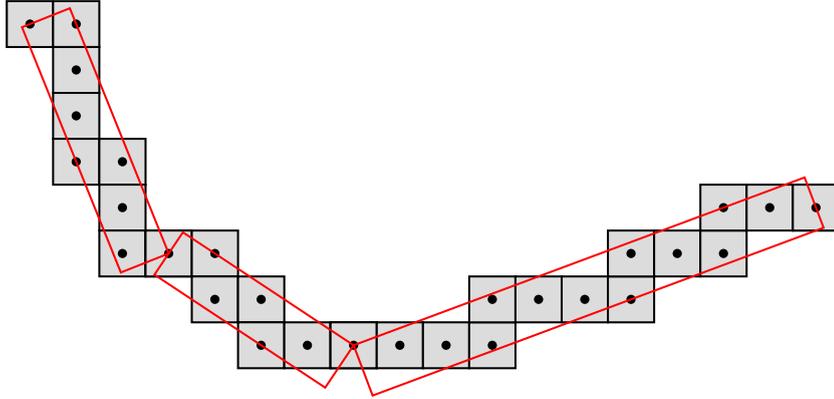


FIGURE 6.5: Example of a discrete curve and its segmentation into set of maximal segments (red rectangles).

- Symmetric tangent (ST) - the symmetric tangent at curve point C_i is the subset $C_{i-l,i+l}$ of C with $S(i-l, i+l)$ and $\neg S(i-l-1, i+l+1)$ (Lachaud and Vialard, 2003)
- Feschet-Touagne tangent (FTT) - the Feschet-Touagne tangent at curve point C_i is the maximal segment with biggest indices that includes the symmetric tangent at C_i (Feschet and Touagne, 1999)
- Extended tangent (ET) - the extended tangent at curve point C_i is the symmetric tangent $C_{i-l,i+l}$ which can be extended in the two following cases: (i) if $S(i-l, i+l+1)$ and $\neg S(i-l-1, i+l)$; (ii) if $S(i-l-1, i+l)$ and $\neg S(i-l, i+l+1)$
- Forward half-tangent (FHT) - the forward half-tangent at curve point C_i is the subset $C_{i,i+l}$ of C with $S(i, i+l)$ and $\neg S(i, i+l+1)$
- Backward half-tangent (BHT) - the backward half-tangent at curve point C_i is the subset $C_{i-l,i}$ of C with $S(i-l, i)$ and $\neg S(i-l-1, i)$
- Median half-tangent (HT) - the median half-tangent at curve point C_i is the arithmetical line median to the two half-tangents (FHT and BHT)

Then tangent direction is directly estimated based on angle between carrying standard line of the DSS and the x -axis, and denoted by θ .

Thanks to methods proposed in the literature designed to efficient recognition of the straight line segment, or computing tangential cover of the discrete curve (Debled-Rennesson and Reveillès, 1995; Feschet and Touagne, 1999; Coeurjolly et al., 2001a), adaptive tangent estimators have linear in time and space complexity which make them much faster than continuous methods and simple to use. They produce, in general, much better results than methods which are based on fixed size of computational windows. However, the presence of noise or corrupted curves can have negative impact on the results, since they make a problem to robust DSS recognition.

To deal with this problem the notion of blurred segments was introduced by [Debled-Rennesson et al. \(2006\)](#). Introduced strategy permits to change thickness of the straight lines which need to be recognized (note that an algorithm proposed by [Buzer \(2005\)](#) can be used for the same purpose). By changing vertical thickness it is possible to accommodate local perturbations in the discrete curve. However, selecting a large thickness can cause algorithm to lose important details of the curve. [Kerautret and Lachaud \(2012\)](#) recently addressed this problem and proposes a reliable method which automatically adapts thickness of used straight line. Such thickness can be different in each part of the curve, which gives a methodology to effectively deal with noise when methods based on DSS recognition are used.

6.3.3 2D λ -maximal segment tangent direction

[Lachaud et al. \(2007\)](#) recently proposed a tangent estimator called λ -maximal segment tangent direction (or λ -MSTD), which is considered as the best among digital ones at rough scale ([Lachaud et al., 2005b, 2007](#)). The accuracy of this estimator is as good as standard continuous methods ([De Villeville and Lachaud, 2008](#)). It is a simple method based on maximal straight segments recognition [Feschet and Tougne \(1999\)](#) along digital contour, and has very good properties such as linear computation complexity and accurate results. Moreover, it has multigrid convergence proven in [Lachaud et al. \(2007\)](#). Therefore, it is interesting to present λ -MSTD in more details.

λ -MSTD estimator was originally designed for estimating tangents on 2D digital contours. Its extension to three-dimensional discrete curves is proposed in chapter 8.

The main part of the λ -MSTD estimator consists in obtaining the tangential cover \mathcal{M} of a given k -curve C ,

Proposition 6.5. *For any discrete curve C , there is a unique set \mathcal{M} of its maximal segments, called the tangential cover.*

This property is derived by using the notion of the maximality of saturated set presented in [Feschet \(2010\)](#). Figure 6.6 illustrates an example of C and its tangential cover \mathcal{M} .

Then the estimator at a point x of a k -curve C should depend on the set of all maximal segments going through x . Let us number all the maximal segments of the tangential cover \mathcal{M} of C by increasing indices such that $M_i \in \mathcal{M}$ for $i = 1, 2, \dots$. Then, such a set is defined by $P(x) = \{M_i \in \mathcal{M}, x \in M_i\}$, called the *pencil of maximal segments* around x . Figure 6.7 illustrates an example of C and pencil of maximal segments $P(x)$ around point $x \in C$. As any point x of a k -curve C is covered by at least one maximal segment, we have the next property [Feschet \(2010\)](#).

Proposition 6.6. *The pencil of maximal segments $P(x)$ of any point x of a k -curve C is never empty.*

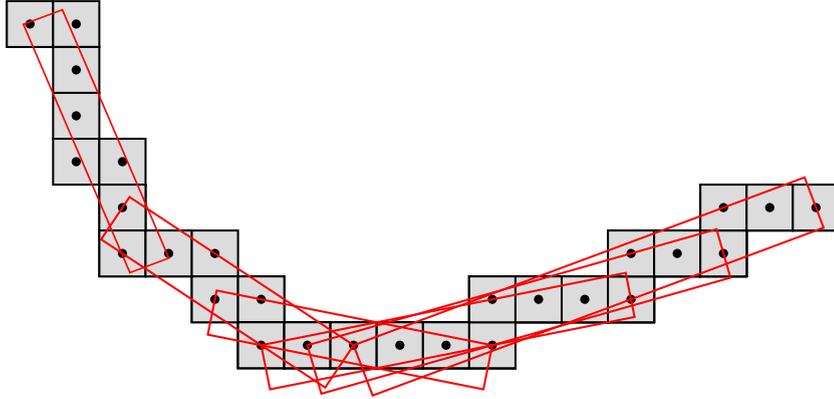


FIGURE 6.6: Example of a discrete curve and its tangential cover (red rectangles).

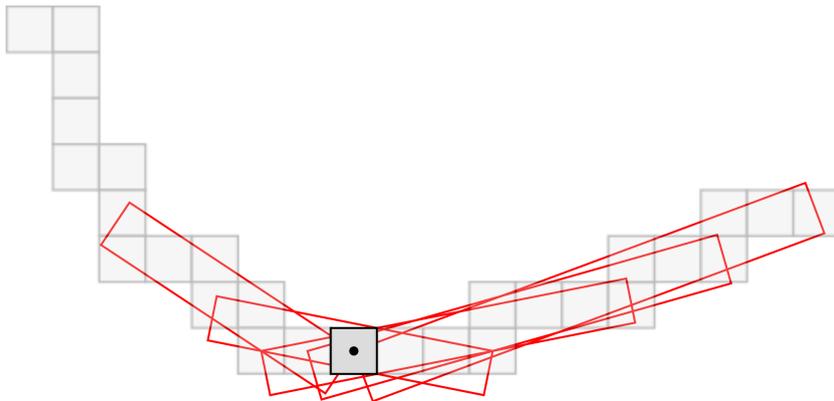


FIGURE 6.7: Example of a discrete curve and the pencil of maximal segments (red rectangles) of selected point (in dark gray).

In addition, as noted in [Feschet and Tougne \(1999\)](#), several successive points may have the same pencil. Therefore, the tangent estimator should take also into account the position of the point x within the pencil (see Fig. 6.8). More specifically, each point x has the eccentricity with respect to each maximal segment. Let us denote by $L_i = \|n_i - m_i\|_1$ the length of each $M_i = C_{m_i, n_i} \subset C$. Then, the *eccentricity* $e_i(x)$ of a point x with respect to a maximal segment M_i is its relative position between the extremities of M_i such that

$$e_i(x) = \begin{cases} \frac{x - m_i}{L_i} & \text{if } M_i \in P(x), \\ 0 & \text{otherwise.} \end{cases} \quad (6.1)$$

The tangent direction is thus estimated by a combination of the directions of maximal segments weighted by a function of the corresponding eccentricity. The function λ maps from $[0, 1]$ to \mathbb{R}_+ with $\lambda(0) = \lambda(1) = 0$ and $\lambda > 0$ elsewhere. For example, a C^2 function $64(-x^6 + 3x^5 - 3x^4 + x^3)$ can be used. Different proposition of several λ functions and theirs impact on the results are shown in chapter 8.4.1. More detailed information about the required properties of the λ function can be found in [Lachaud et al. \(2007\)](#).

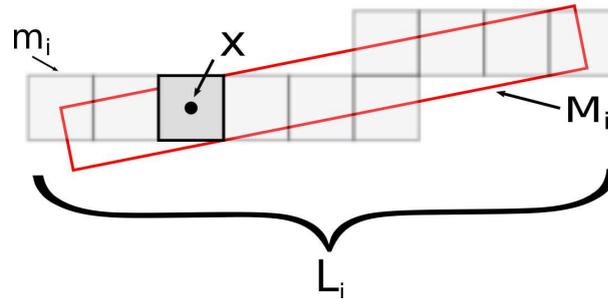


FIGURE 6.8: Example of a maximal segment M_i around the point x . The eccentricity $e_i(x)$ depends on the position of x in M_i . Values m_i and L_i denotes the first point and length of the segment M_i , respectively.

Definition 6.7. The λ -MST direction $\theta(x)$ at point x of a k -curve C is defined as a weighted combination of the directions θ_i of the covering maximal segments M_i such that

$$\theta(x) = \frac{\sum_{M_i \in P(x)} \lambda(e_i(x)) \theta_i}{\sum_{M_i \in P(x)} \lambda(e_i(x))}. \quad (6.2)$$

From Property 6.6 and the nature of the eccentricity, this value is always defined and computed with a linear time complexity.

Part II

Theory and Algorithms

Chapter 7

Medial Axis Filtering with Features at Different Scales

7.1 Discrete Adaptive Medial Axis

The major drawback of the discrete λ -medial axis method is a problem of filtering shapes which have important features at different scales (Attali et al., 2009). In such case, when we increase parameter λ to make a stronger filtration in thick parts, we lose information in the thin parts of the shape (see Fig. 7.1).

This can be simply explained by the fact that to threshold function \mathcal{F}_X , by using the inequality $\mathcal{F}_X(x) \geq \lambda$, we based only on the parameter λ , which is set globally, independently of any information derived from x like e.g. local thickness of the object (which can be simply evaluated by the Euclidean opening function, presented in chapter 4.3).

This fact motivates us to investigate more deeply on how function \mathcal{F}_X is related to local size of the shape defined by function G_E^X . It is very interesting to see visually this relation by plotting distribution of points of a shape X according to the values $\mathcal{F}_X(x)$ and $G_E^X(x)$. The resulting graphs for different families of shapes are presented on Fig. 7.2.

After analysis of how points aggregate in several cases, one can notice interesting patterns. We can divide the graph \mathcal{F}_X vs. G_E^X into three sets.

- The first set, denoted by V , consists of points $x \in X$, such that, $\mathcal{F}_X(x) \approx G_E^X(x)$. Points in this set are very important, since, they have value \mathcal{F}_X close to a local maximum. Therefore, points from such set should be preserved during filtering.
- The second set, denoted by U is formed by points $y \in X$, such that, $\mathcal{F}_X(y) \ll G_E^X(y)$. From filtering point of view, such points in general, belong to unstable part of λ -medial axis and should be removed.

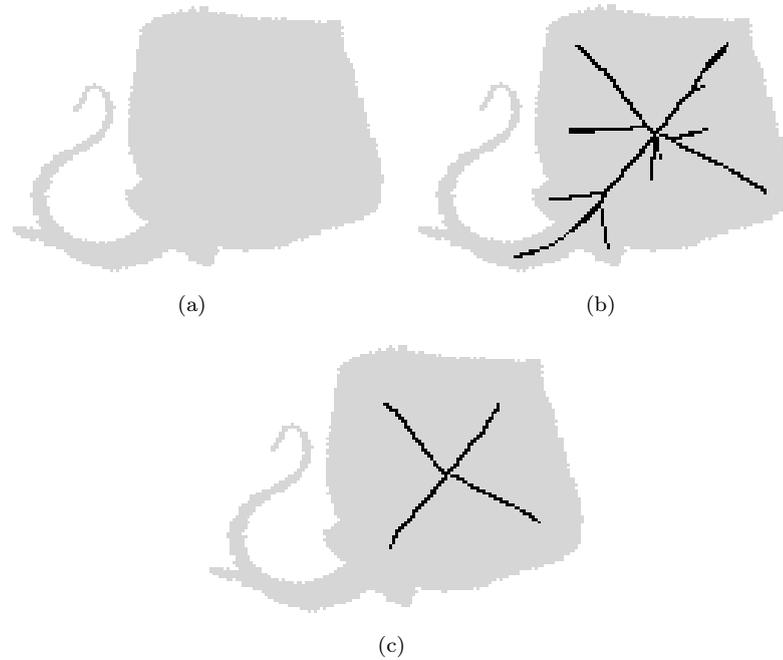


FIGURE 7.1: (a): a binary shape X (in gray); (b): The filtered λ -medial axis of X (in black) calculated by using algorithm (Chaussard et al., 2010). The medial axis is not sufficient filtered in the middle of the shape. However, we already start to lose the tail; (c): The more filtered medial axis of X . Now, the middle of the shape is well filtered. However, we lost all information about the tail.

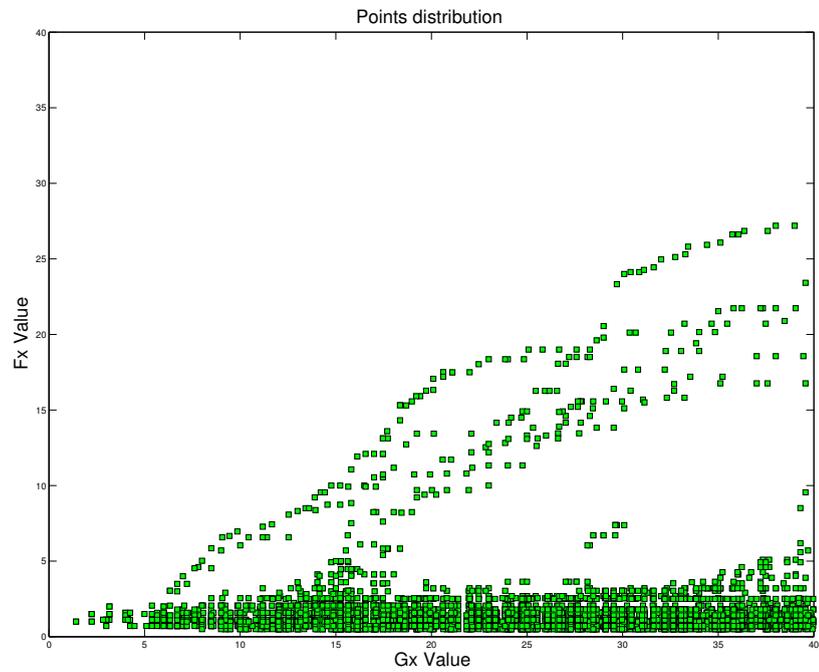
- The last set, denoted by I , consist of points $z \in X$, such that, $\mathcal{F}_X(z) < G_E^X(z)$. Points in set I , generally should be also preserved. However, their importance can be different according to level of the λ -medial axis simplification we would like to achieve. Notice that definition of function \mathcal{F}_X makes situation where $\mathcal{F}_X(x) > G_E^X(x)$ as impossible to achieve.

On the Fig. 7.3 sets I , U and V are sketched. We can also see how thresholding of function \mathcal{F}_X works in practice, according to different parameter λ . Now it is easy to see that using only a globally defined parameter λ in inequality $\mathcal{F}_X(x) \geq \lambda$, we are unable to well separate set U from sets V and I , which is crucial for preserving thin parts of the shape and allow effective filtering in the thick ones at the same time.

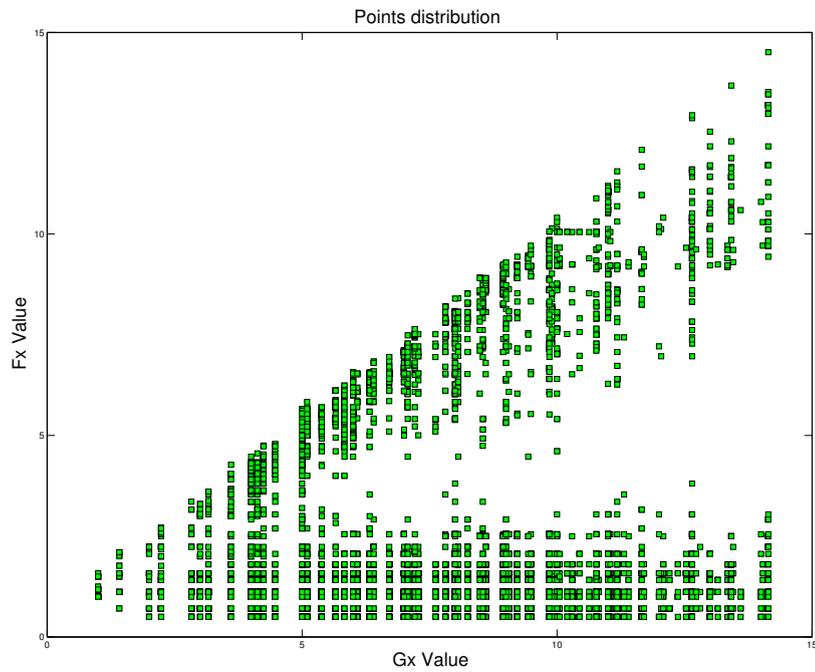
Our proposition to solve such problem is a new notion derived from the discrete λ -medial axis, where we introduce a new parameter γ in thresholding inequality. We call this new medial axis as the discrete adaptive medial axis, and define it as follows.

Definition 7.1. Let X be a finite subset of \mathbb{Z}^n . Let $\gamma \in \mathbb{R}$. The *discrete adaptive medial axis (or DAMA)* of X is the set of couples $(x, D_X(x))$ for x in X such that $\mathcal{F}_X(x) \geq \lambda_\gamma(x)$ where,

$$\lambda_\gamma(x) = \frac{\ln(G_E^X(x) + 1)}{\ln(\gamma)} \quad (7.1)$$



(a)



(b)

FIGURE 7.2: (a) and (b): Two graphs representing points distribution for two different shapes according to their $\mathcal{F}_X(x)$ and $G_E^X(x)$ values.

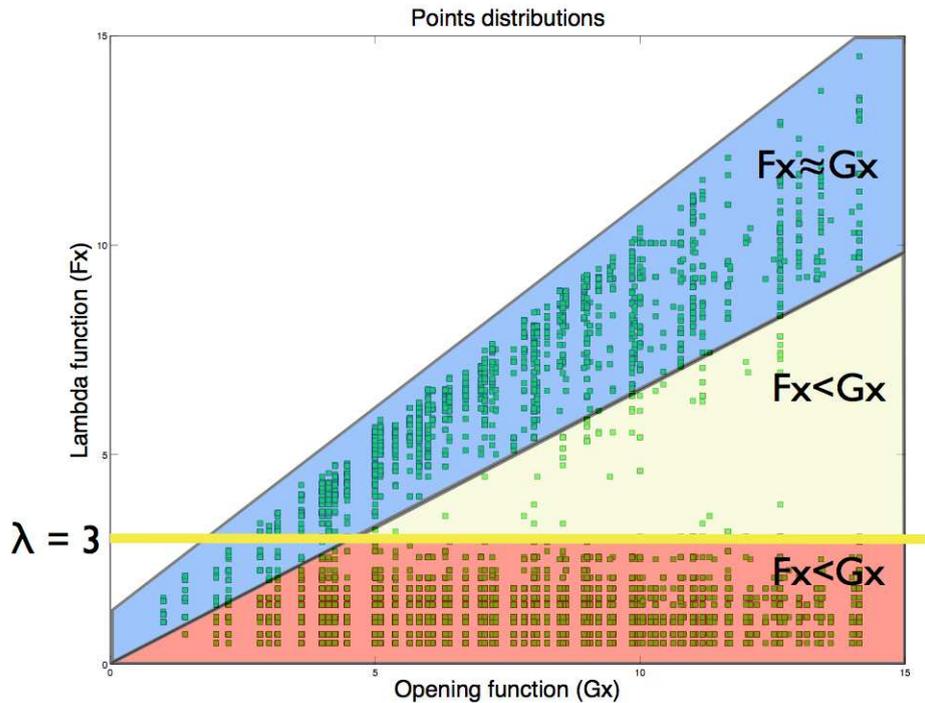


FIGURE 7.3: Graph representing points distribution according to their $\mathcal{F}_X(x)$ and $G_E^X(x)$ values. We can also distinguish three important sets from filtration point of view (V , I and U respectively). By lines we mark thresholding value according to different λ values. Points which are below such line will be removed during filtering.

In comparison with the DLMA where $\mathcal{F}_X(x)$ is compared to a scalar value λ , in definition 7.1 the filtering criterion depends on two values, as we included in function λ information about opening value $G_E^X(x)$. Which allows us to adapt filtering criterion to the local size of the shape. It is also possible to modify strength of filtration by setting a proper single parameter γ in equation 7.1.

One can also propose different families of functions $\lambda_\gamma(x)$ to make filtering with different behaviors by changing equation 7.1. However, we experimentally tested several functions and selected the one which gives the most interesting results from our point of view. Using this function allows us to well separate sets V and U and gives us some flexibility for dividing set I , thanks to parameter γ (see Fig. 7.4).

Similarly to the original λ -medial axis and some other previously proposed methods e.g. based on bisector angle, any adaptive medial axis DAMA of a shape X for a parameter value γ is a subset of any filtered medial axis DAMA of X for parameter value μ , whenever $\lambda_\gamma(x) \geq \lambda_\mu(x)$. In other words, the set of all filtered medial axes of a shape forms a hierarchy. On the Fig. 7.4 we can see the function $\lambda_\gamma(x)$ for three different values of the parameter γ . All points which are below line defined by $\lambda_\gamma(x)$ will be removed during filtering.

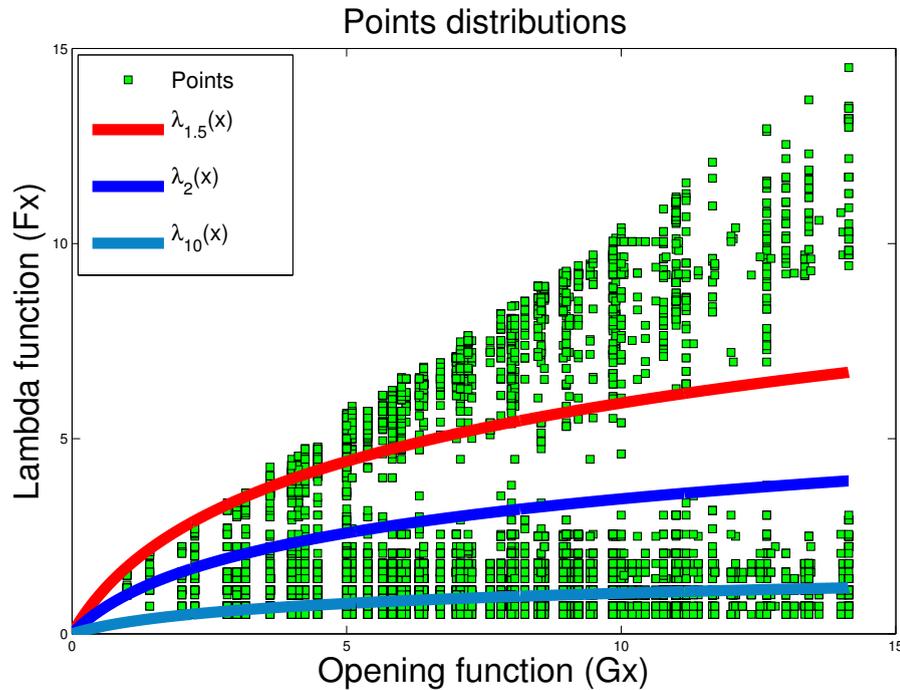


FIGURE 7.4: Graph representing points distribution according to their $\mathcal{F}_X(x)$ and $G_E^X(x)$ values. By the thick lines, we mark the thresholding value according to equation 7.1 for different $\lambda_\gamma(x)$. Points which are below such lines will be removed during filtering.

Is very important to see that meanings of the parameter λ in DLMA and γ in DAMA are different. Moreover, note, that condition $\lambda_\gamma(x) \geq \lambda_\mu(x)$ is satisfied if $\mu \geq \gamma$ which is opposite to the other methods presented in this work where by increasing filtering parameter we cause stronger filtration. To remove such inconsistency and to make hierarchy property better visible, we present the definition of the discrete adaptive medial axis in a different form (see definition 7.2).

Definition 7.2. Let X be a finite subset of \mathbb{Z}^n . We denote by $DAMA(X, \gamma)$ the discrete adaptive medial axis of X with parameter γ , defined by

$$DAMA(X, \gamma) = \{x \in X \mid \mathcal{F}_X^*(x) \geq \gamma\}$$

where,

$$\mathcal{F}_X^*(x) = \begin{cases} 0 & \text{if } \mathcal{F}_X(x) = 0 \\ e^{\frac{\mathcal{F}_X(x)}{\ln(G_E^X(x)+1)}} & \text{otherwise} \end{cases} \quad (7.2)$$

Now, according to definition 7.2, we compare $\mathcal{F}_X^*(x)$ to the single scalar value γ , similarly to other filtering criteria. Moreover, increasing this parameter cause stronger filtration.

7.2 Discrete Scale Axis

In this section, we adapt the notion of scale axis (see chapter 4.2.5), originally introduced in the continuous space and implemented in a framework of unions of balls, to the case of discrete grids. Let $X \subset \mathbb{Z}^n$, $x \in X$, $r \in \mathbb{R}_+$ and $s \in \mathbb{R}_+$. The parameter s is called the *scale factor*. We denote by X_s the *multiplicatively s -scaled shape*, defined by $X_s = \bigcup_{(x,r) \in MA(X)} B(x, rs)$. For $s \geq 1$, we denote by $SAT_s(X)$ the *s -scale axis transform* of X , defined by

$$SAT_s(X) = \{(x, r/s) \mid (x, r) \in MA(X_s)\}$$

An algorithm to compute this discrete version of scale axis can be now presented as follows.

First, calculate the Euclidean medial axis of X and then multiply radius of each medial ball by the chosen scaling factor s .

Next step is to reconstruct object X_s based on scaled radius values. Reconstruction can be made efficiently by reverse Euclidean distance transform (see, for example, (Coeurjolly and Montanvert, 2007)). Now, computing the medial axis of X_s achieves the simplification.

The final step of the algorithm consists of rescaling the medial balls of $MA(X_s)$ by a factor $1/s$ to obtain the scale axis of X . Finally, discrete scale axis algorithm can be presented in the following pseudocode:

Algorithm 7 DiscreteScaleAxis(**Input** X, s **Output** $SAT_s(X)$)

```

01.   $SAT_s(X) \leftarrow \emptyset$ ;  $MA'(X) \leftarrow \emptyset$ 
02.   $MA(X) \leftarrow \mathbf{EuclideanMedialAxis}(X)$ 
03.  foreach  $(x_i, r_i) \in MA(X)$  do
04.       $MA'(X) \leftarrow MA'(X) \cup \{(x_i, r_i s)\}$ 
05.  end
06.   $X_s \leftarrow \mathbf{ReverseEuclideanDT}(MA'(X))$ 
07.   $MA(X_s) \leftarrow \mathbf{EuclideanMedialAxis}(X_s)$ 
08.  foreach  $(x_i, r_i) \in MA(X_s)$  do
09.       $SAT_s(X) \leftarrow SAT_s(X) \cup \{(x_i, r_i/s)\}$ 
10.  end
11.  return  $SAT_s(X)$ 

```

All steps of DiscreteScaleAxis algorithm can be calculated in linear time in relation to $|X|$ (line 02), $|MA(X)|$ (line 03), $|X_s|$ (lines 06-07) and $|MA(X_s)|$ (line 08) respectively. Therefore, overall computational complexity of the algorithm is $O(|X|)$.

7.3 Hierarchical Scale Medial Axis

7.3.1 The s -scale filtered Euclidean medial axis

The crucial part of the method presented in the previous section, which is a source of problems ($MA(X_s) \not\subseteq MA(X)$), is the reconstruction part after medial balls scaling and need for new medial axis generation of the scaled object (see Fig. 7.5). On the other hand, at the first look, this is the most important part of the algorithm since the medial axis simplification occurs in this part.

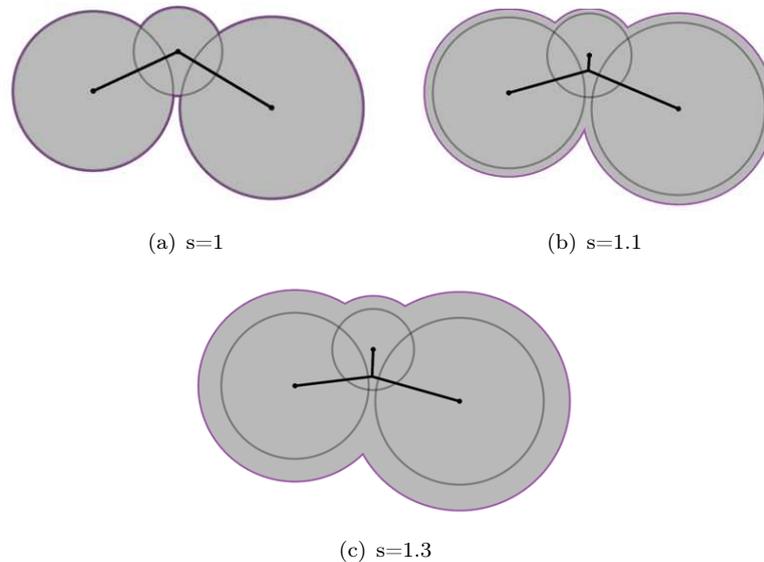


FIGURE 7.5: (a): a shape X (in gray) and its Euclidean medial axis $MA(X)$ (in black); (b): the multiplicatively 1.1-scaled shape and its 1.1-scaled axis; (c): the multiplicatively 1.3-scaled shape and its 1.3-scaled axis. In (b, c) we can see that scaled axes are not subsets of $MA(X)$. In both cases, an additional branch even appears. (Miklos et al., 2010)

To filter $MA(X)$ by removing centers of unimportant medial balls and preserving a set which is a subset of the $MA(X)$, one must avoid reconstruction part and hold simplification property at the same time. To solve this problem we make the assumption that to make efficient filtration we just need to decide which of $MA(X)$ points are not important and should be removed. Therefore, we do not generate a new object X_s , and its $MA(X_s)$ set. With this assumption we are sure that filtered Euclidean medial axis of X is a subset of $MA(X)$.

This informal discussion motivates the following definition of the s -Scale Filtered Euclidean Medial Axis (SFEMA).

Let $x \in X, r \in \mathbb{R}_+$. We denote by $B^X(x, r)$ the intersection of $B(x, r)$ with X , that is, $B^X(x, r) = \{y \in X \mid d(x, y) \leq r\}$.

Definition 7.3. Let $X \subset \mathbb{Z}^n$, and $s \in \mathbb{R}, s \geq 1$. We denote by $SFEMA_s(X)$ the *Scale Filtered Euclidean Medial Axis* of X defined by

$$SFEMA_s(X) = \{(x, r) \in MA(X) \mid B^X(x, rs) \not\subseteq \bigcup_{(y,t) \in MA(X), t > r} B^X(y, ts)\}$$

Below, we give an algorithm to compute $SFEMA_s(X)$ of a given object $X \subset \mathbb{Z}^n$.

Algorithm 8 SFEMA(**Input** X, s **Output** H)

```

01.    $H \leftarrow \emptyset$ 
02.    $MA(X) \leftarrow \mathbf{EuclideanMedialAxis}(X)$ 
03.   Let  $(x_1, r_1), \dots, (x_n, r_n)$  denote the elements of  $MA(X)$ 
      sorted in decreasing order of radii, that is,  $r_1 \geq \dots \geq r_n$ 
04.   foreach  $p \in X$  do
05.        $i \leftarrow 1$ 
06.       while  $d(x_i, p) > sr_i$  do  $i \leftarrow i + 1$  end
07.        $H \leftarrow H \cup \{(x_i, r_i)\}$ 
08.   end
09.   return  $H$ 

```

Theorem 7.4. *Algorithm SFEMA computes exactly the s -Scale Filtered Euclidean Medial Axis, as defined by Def. 7.3.*

Proof. Consider any point p in X (after line 04). The assertion: “for all $k \in \{1, \dots, i\}, d(x_k, p) > sr_k$ ” holds true during the execution of the loop at line 06. After execution of this loop (line 07) we know that for all $k \in \{1, \dots, i-1\}, d(x_k, p) > sr_k$, and $d(x_i, p) \leq sr_i$. Thus, at line 08, we know that p belongs to $B^X(x_i, r_i s)$ and that it does not belong to any ball $B^X(x_k, r_k s)$, with $r_k > r_i$. From this we deduce that (x_i, r_i) belongs to $SFEMA_s(X)$.

This proves that all detected pairs indeed belong to $SFEMA_s(X)$. Now consider any pair (x, r) of $SFEMA_s(X)$, there exists at least one point p in $B^X(x, rs)$ that is not included in $\bigcup_{(y,t) \in MA(X), t > r} B^X(y, ts)$. As all points p of X are considered by the algorithm (line 04), the pair (x, r) will be detected. \square

SFEMA in line 02 performs sorting of medial axis elements. It is possible to implement sorting algorithm in linear time with the use of counting sort and integer values of radius e.g. when the square of Euclidean distance transform is used. In the following lines the algorithm performs two loops, the second nested in the first. The first one starts in line 04 and does $|X|$ iterations. The next nested loop starts in line 06 and in worst case performs $|MA(X)|$ iterations. Summarizing, computational complexity of SFEMA is $O(|X||MA(X)|)$ where $|MA(X)| \ll |X|$ (usually).

Examples of $SFEMA_s(X)$ with different scale factors s_i , calculated by using the method described above, are shown in Fig. 7.6

Let us analyze properties and the major differences between the Miklos et al. (2010) s -scale axis and our s -scale filtered Euclidean medial axis. The most important property is

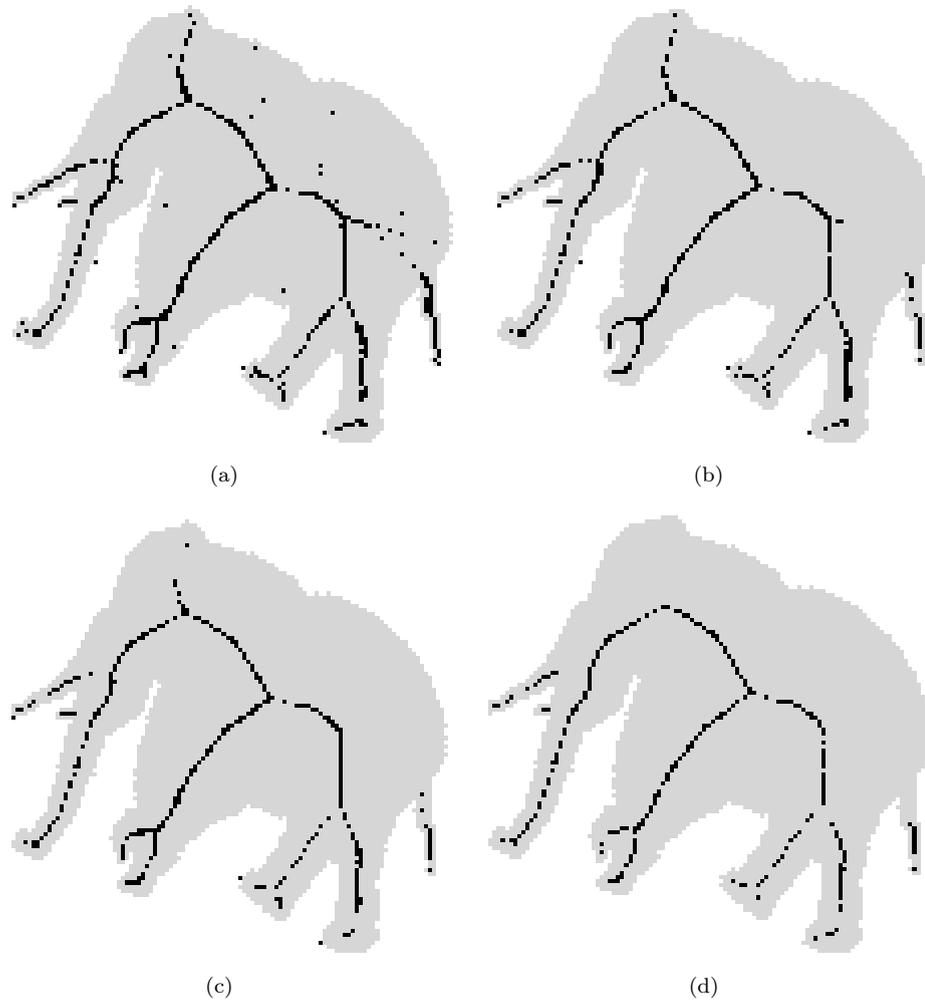


FIGURE 7.6: (a): a shape X (in gray) and its Euclidean medial axis (in white); (b, c, d): the same shape and its $SFEMA_{1.1}(X)$, $SFEMA_{1.4}(X)$, $SFEMA_{1.6}(X)$, respectively. In all cases the elephant's tail, trunk, tusks and legs were considered as important and were not removed.

$SFEMA_s(X)$ consists only of couples which are also in non-filtered $MA(X)$, that is, for all $s \geq 1$: $SFEMA_s(X) \subset MA(X)$. Lacking this property (inclusion property, for short) has very strong negative impact to final results. In Fig.7.5 we have shown example of the Miklos's scale axis where additional branch even appear after filtering. In the Fig.7.7 opposite example is shown. The scale axis is too much simplified and loses important features of the object. However, s -scale filtered medial axis is holding inclusion property and permits to reconstruct almost entirely the original object.

The second interesting property relies on the notion of ball intersection with object. In the Miklos's scale axis, if we want to simplify the object, for example, by removing medial ball $B(x, r)$, $x \in X$, the scale factor should be set to big enough value which cause ball $B(x, rs)$ to be included in one of other medial ball, that is, $B(x, rs) \subset B(y, rs)$, $y \in X$ (see Fig.7.8(c)) or in a union of such balls. In our algorithm, since we use intersection of scaled balls with X , we only test inclusion inside X (see Fig.7.8(b)). This allows us to use

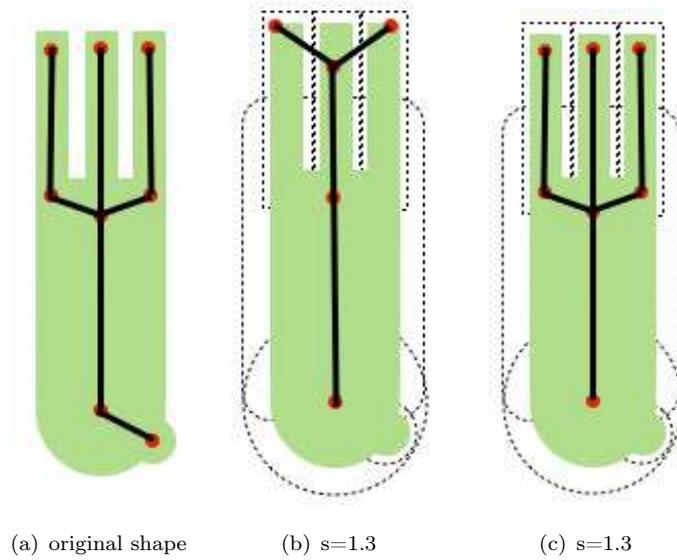


FIGURE 7.7: (a): a set X (in green) and its $MA(X)$ (in black); (b): the 1.3-scale axis of X ; (c): the 1.3-scale filtered Euclidean medial axis of X .

smaller scale factor. Therefore, we have better ability to control resulting s -scale filtered Euclidean medial axis.

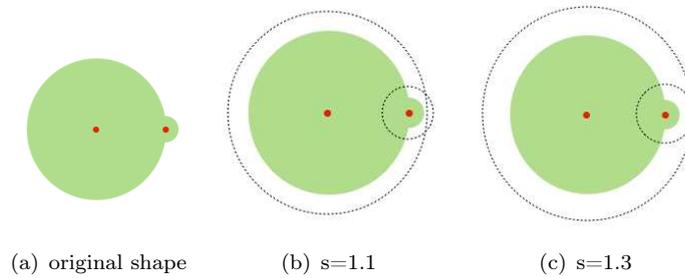


FIGURE 7.8: (a): a set X (in green) and its $MA(X)$ (red dots); (b): multiplicatively scaled medial balls. The smaller ball is not fully covered by the bigger one after scaling. In scale axis representation both balls will be preserved. However, the bigger ball includes the smaller one inside set X . Therefore, the smaller ball will not exist in $SFEMA_s(X)$; (c): multiplicatively scaled medial balls. The smaller ball is included in the bigger one. Therefore, it is neither in scale axis nor in $SFEMA_s(X)$.

7.3.2 Algorithm for computing the hierarchical scale medial axis

Definition 7.5. We denote by $MA_s(X)$ the filtered medial axis of $X \subset \mathbb{Z}^n$ for $s \in S \subseteq \mathbb{R}_+$, where S represents the set of all possible values of parameter s of considered filtering algorithm. The family $H = \{MA_s(X) \mid s \in S\}$ is a *hierarchy* if $\forall s_1, s_2 \in S \mid s_2 > s_1 \Rightarrow MA_{s_2}(X) \subseteq MA_{s_1}(X) \subseteq MA(X)$.

Unfortunately, SFEMA does not hold hierarchy property, it means that there is $X \subset \mathbb{Z}^n$ such that $\{SFEMA_s(X) \mid s \geq 1\}$ is not a hierarchy. The counter-example to the hierarchy property is presented in the Fig. 7.9.

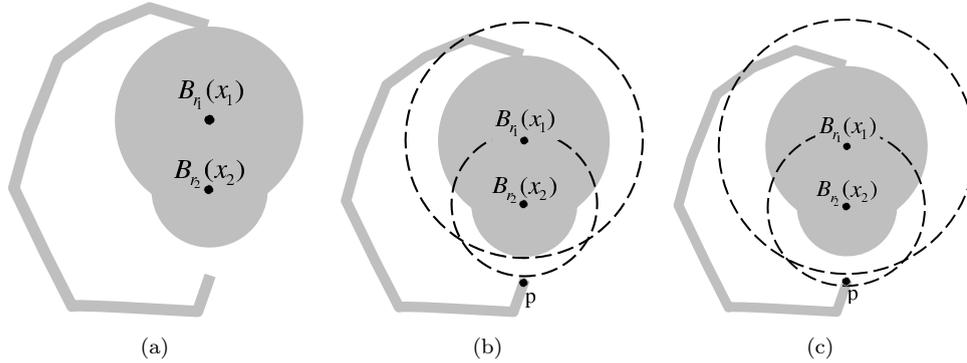


FIGURE 7.9: (a): a shape X (in gray) and two maximal balls $B(x_1, r_1), B(x_2, r_2) \in X$; (b): dashed circles represent scaled balls by $s = 1.4$. One can observe that $B^X(x_2, r_2 \cdot 1.4) \subset B^X(x_1, r_1 \cdot 1.4)$, so $(x_2, r_2) \notin SFEMA_{1.4}(X)$; (c): dashed circles represent scaled balls by $s = 1.6$, now $p \in B^X(x_2, r_2 \cdot 1.6)$ and $p \notin B^X(x_1, r_1 \cdot 1.6)$, so $(x_2, r_2) \in SFEMA_{1.6}(X)$, which finally leads to the conclusion that $SFEMA_s(X)$ does not hold hierarchy property.

This fact motivates the following definition:

Definition 7.6. Let $X \subset \mathbb{Z}^n$ and $(x, r) \in MA(X)$. We denote by $\Psi_X : MA(X) \rightarrow [1, \infty) \cup \{\infty\}$ the *scale map* of X defined by

$$\Psi_X(x, r) = \min\{t \geq 1 \mid \forall s \geq t, (x, r) \notin SFEMA_s(X)\}$$

We first present an algorithm that is easier to understand and prove that our final one, but less efficient.

Algorithm 9 ScaleMap1(**Input** $MA(X)$ **Output** Ψ_X)

```

01.  foreach  $(x, r) \in MA(X)$  do  $\Psi_X(x, r) \leftarrow 0$  end
02.  foreach  $p \in X$  do
03.    foreach  $(x_i, r_i) \in MA(X)$  do
04.       $t_{min} \leftarrow +\infty$ 
05.      for  $j = 1$  to  $n$  do
06.         $t \leftarrow \frac{d(x_j, p)}{r_j}$ 
07.        If  $t > 1$  and  $t < t_{min}$  and  $r_j > r_i$  then  $t_{min} \leftarrow t$ 
08.      end
09.      If  $t_{min} > \Psi_X(x_i, r_i)$  then  $\Psi_X(x_i, r_i) \leftarrow t_{min}$ 
10.    end
11.  end

```

Before proving the validity of the algorithm, let us first remark that, in the definition of Ψ , it is not necessary to consider all possible values of t or s , but only the ones of the form $\frac{d(x, p)}{r}$, with $p \in X$ and $(x, r) \in MA(X)$, and $+\infty$. Thus, we define:

$$S_p = \left\{ s = \frac{d(x_i, p)}{r_i}; (x_i, r_i) \in MA(X) \text{ and } s > 1 \right\} \cup \{+\infty\}$$

$$S = \bigcup_{p \in X} S_p$$

And we have:

$$\Psi_X(x, r) = \min\{t \in S \mid \forall s \in S, s \geq t, (x, r) \notin SFEMA_s(X)\} \quad (7.3)$$

Now, for any p in X and any (x, r) in $MA(X)$, let us define:

$$E(p, x, r) = \min\{t \in S_p \mid \forall s \in S, s \geq t, p \notin B^X(x, sr) \text{ or } p \in \bigcup_{r_j > r} B(x_j, sr_j)\}$$

Proposition 7.7. *For any (x, r) in $MA(X)$, we have $\Psi_X(x, r) = \max_{p \in X} E(p, x, r)$.*

Proof. From the definition of $SFEMA$ and from (7.3), we have

$$\Psi_X(x, r) = \min\{t \in S \mid \forall s \in S, s \geq t, \forall p \in B^X(x, sr), p \in \bigcup_{r_j > r} B(x_j, sr_j)\}$$

That is,

$$\Psi_X(x, r) = \min\{t \in S \mid \forall s \in S, s \geq t, \forall p \in B^X(x, sr), C(p, x, sr)\}$$

where $C(p, x, sr)$ stands for “there exists j such that $r_j > r$ and $p \in B(x_j, sr_j)$ ”.

Let $t_1 = \Psi(x, r)$, thus

$$\forall s \in S, s \geq t_1, \forall p \in B^X(x, sr), C(p, x, sr) \quad (7.4)$$

Now, suppose that there is a p in X such that $t_1 < E(p, x, r)$, i.e.,

$$t_1 < \min\{t \in S_p \mid \forall s \in S, s \geq t, p \notin B^X(x, sr) \text{ or } C(p, x, sr)\}.$$

Take $s = t_1$, we thus have $s \in S_p$, $p \in B^X(x, sr)$ and not $C(p, x, sr)$, a contradiction with (7.4).

This proves that for any p in X , $\Psi_X(x, r) \geq E(p, x, r)$, hence $\Psi_X(x, r) \geq \max_{p \in X} E(p, x, r)$.

From the definitions of Ψ_X and E , we can deduce that there exists a p in X such that $\Psi_X(x, r) = E(p, x, r)$, hence $\Psi_X(x, r) = \max_{p \in X} E(p, x, r)$. \square

Theorem 7.8. *Algorithm ScaleMap calculates exactly scale map, as defined in the Def. 7.6*

Proof. Let us first remark that $E(p, x, r) = \min\{t \in S_p \mid \forall s \in S, s \geq t, p \notin B^X(x, sr)\} \cup \{t \in S_p \mid \forall s \in S, s \geq t, p \in \bigcup_{r_j > r} B(x_j, sr_j)\}$, and that $\{t \in S_p \mid \forall s \in S, s \geq t, p \notin B^X(x, sr)\}$ is always empty. Hence we have:

$$E(p, x, r) = \min\{t \in S_p \mid \forall s \in S, s \geq t, p \in \bigcup_{r_j > r} B(x_j, sr_j)\}.$$

Remark also that the number t , lines 6-7, is such that for any $s \geq t, p \in B(x_j, sr_j)$. Thus it is easy to see that the quantity t_{min} represents $\min\{t \in S_p \mid \forall s \in S, s \geq t, p \in \bigcup_{r_j > r_i} B(x_j, sr_j)\}$, that is, $E(p, x_i, r_i)$. By proposition 7.7, line 9 correctly computes the value of $\Psi_X(x_i, r_i)$. \square

Now, let us present a variant of the previous algorithm, that computes the same result but with a lower time complexity.

Algorithm 10 ScaleMap(**Input** $MA(X)$ **Output** Ψ_X)

```

01.  foreach  $(x, r) \in MA(X)$  do  $\Psi_X(x) \leftarrow 0$  end
02.  Let  $(x_1, r_1), \dots, (x_n, r_n)$  be elements in  $MA(X)$  sorted such that  $r_i \geq r_{i+1}$ 
03.  foreach  $p \in X$  do
04.       $t_{min} \leftarrow +\infty$ 
05.      for  $i = 1$  to  $n$  do
06.           $t \leftarrow \frac{d(x_i, p)}{r_i}$ 
07.          If  $i = 1$  or  $r_i \neq r_{i-1}$  then  $t_{min}^r \leftarrow t_{min}$ 
08.          If  $t > 1$  and  $t < t_{min}$  then  $t_{min} \leftarrow t$ 
09.          If  $\Psi_X(x_i, r_i) < t_{min}^r$  then  $\Psi_X(x_i, r_i) \leftarrow t_{min}^r$ 
10.      end
11.  end

```

Now, let us define a new medial axis filtering strategy, which holds hierarchy property, based on the scale map calculated by algorithm 9.

Definition 7.9. Let $X \subset \mathbb{Z}^n$, $s > 1$ and Ψ_X is a scale map of X . We define by $HSMA_s(X)$ the *hierarchical scaled medial axis* of X by the following formula:

$$HSMA_s(X) = \{(x, y) \in MA(X) \mid \Psi_X(x, r) > s\}$$

By conclusion, we have the following property, which is straightforwardly consequence of definition 7.5 and 7.9.

Theorem 7.10. *Let $X \subset E$. The family $H = \{HSMA_s(X) \mid s > 1\}$ is a hierarchy.*

The computational complexity of algorithm 10 is the same as algorithm 8, that is, $O(|X||MA(X)|)$. The algorithm in line 02 performs sorting of medial axis elements. It is possible to implement sorting algorithm in linear time with the use of a counting sort and integer radius. In the following lines the algorithm performs two loops, the second nested in the first. The first one starts in line 03 and does $|X|$ iterations. The next

nested loop starts in line 05 and performs $|MA(X)|$ iterations, where $|MA(X)| \ll |X|$ (usually).

7.4 Comparison of Medial Axis Filtering Methods

In this section, we compare qualitatively and quantitatively properties of six medial axis filtering algorithms. In the comparison we include algorithms introduced in chapter 4.2, that is, discrete λ -medial axis (DLMA) (Chaussard et al., 2010), the Euclidean medial axis filtered with the use of bisector function (BFMA) (Couprie et al., 2007), filtering based on medial ball size (RFMA) (Coeurjolly and Montanvert, 2007) and filtering based on ball covering (CFMA) (Coeurjolly and Montanvert, 2007). We also include two filtering methods proposed in this work: the hierarchical scaled medial axis (HSMA) and the discrete adaptive medial axis (DAMA).

In our experiments we use all shapes from Kimia's database of 216 shapes (Sharvit et al., 1998). Fig. 7.10 shows a sample of these shapes, each shape represents one of 18 classes of objects in the database. Note that, in the sequel we limit presented examples and illustrations only to 6 carefully selected class of objects to simplify reading (see Fig. 7.11). Four of selected objects: "Ray", "Elephant", "Camel" and "Fountain", have many important features at different scales in contrast to: "Classic" and "Misk" (the used names correspond to the names in Kimia's database).

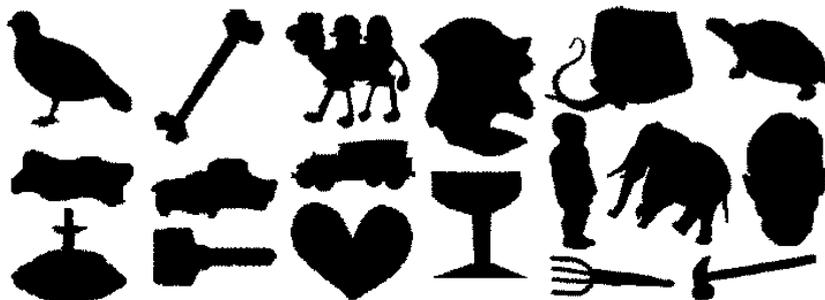


FIGURE 7.10: A sample of the 18 representative classes of shapes of Kimia's database.

In comparison of different medial axes, especially comparison between methods, for which the filtering parameter does not have the same meaning is hard to justify why one filtering criterion is better than another. Therefore, we need to define strategies which allow us to ensure a fair comparison.

In the literature we can find some interesting proposition. Chaussard (2010) defined, so called, *visual quality factor* which is combination of reconstruction error with information derived from skeleton of the shape, that is, number of skeleton points. Medial axes are then compared in respect to achieving the lowest visual quality factor. Van Eede et. al (Eede et al., 2006) propose solution based on weighted reconstruction error, combined

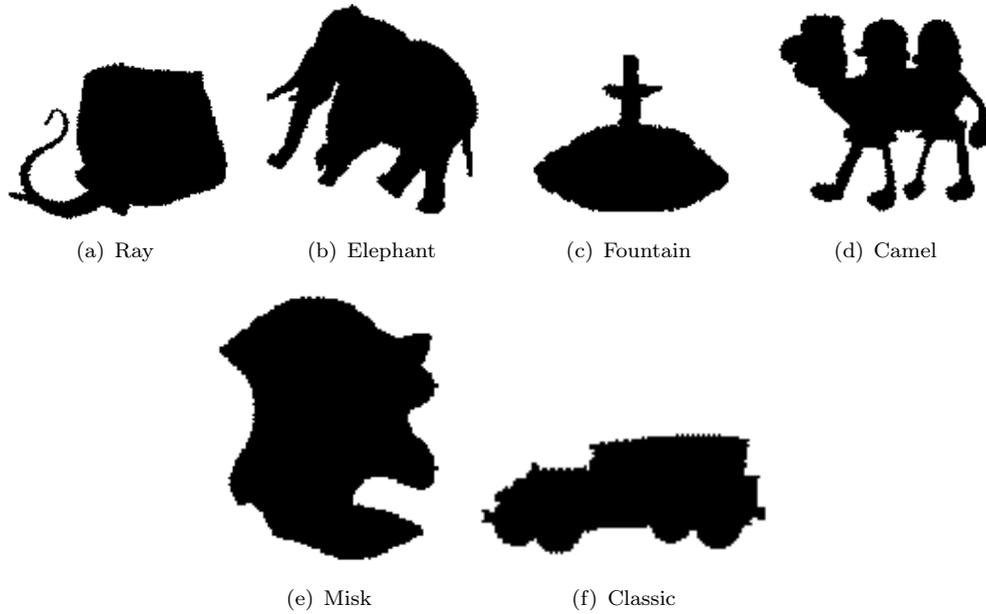


FIGURE 7.11: A sample of the selected 6 representative classes of shapes of Kimia's database used to illustrate results of our experiments.

with skeleton complexity defined as number of skeleton branches. However, quality of the medial axis depends of the reconstruction weight which need to be set arbitrary. Setting this parameter properly is not obvious.

Therefore, in following subsections we introduce several notions that will help us to make comparison between methods. Then we conduct our experiments based on the proposed methodology.

7.4.1 The reconstruction error

Let $X \subset \mathbb{Z}^n$, $Y \subset X$, we denote by $REDT_X(Y)$ the *reverse Euclidean distance transform* (Coeurjolly and Montanvert, 2007), defined by

$$REDT_X(Y) = \bigcup_{y \in Y} B(y, r)$$

For exact and unfiltered $MA(X)$ we have $X = REDT_X(MA(X))$. However, this property is no longer true if we consider filtered medial axes e.g. HSMA, DLMA, BFMA, RFMA, CFMA or DAMA. Therefore, it is interesting to measure how much information about the original object is lost when we perform filtering. We can define

$$R_X(Y) = \frac{|(X \cup REDT_X(Y)) \setminus (REDT_X(Y) \cap X)|}{|X \cup REDT_X(Y)|}$$

We call $R_X(Y)$ the *(normalised) residuals of Y*. Residuals give us a numerical evaluation of *reconstruction error*. Now we can set Y to different filtered medial axes e.g. by using different method or filtering parameter and then evaluate which one filtration is better in respect of ability to original object reconstruction. The results $R_X(Y)$ is a real value between 0 (perfect reconstruction) and 1 (bad reconstruction).

Figure 7.12, 7.13 presents HSMA, DLMA, BFMA, RFMA, CFMA and DAMA of an exemplary shape, extracted for several values of normalised residuals. Note that we compare the results of methods for approximately equal values of their residuals, rather than for equal values of their parameters.

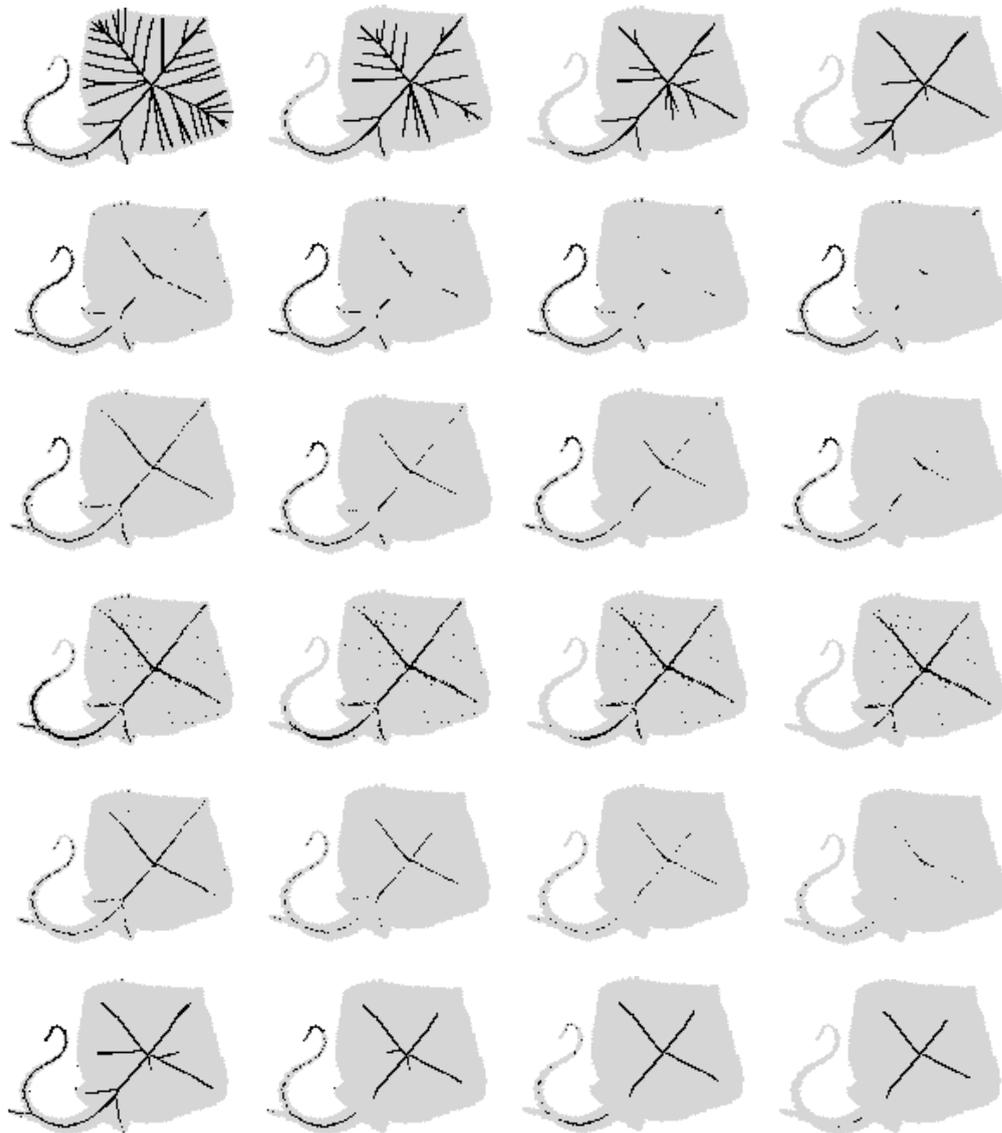


FIGURE 7.12: Medial axes (in black) superimpose to input object (in grey). Consecutive rows (from top to bottom) contain results for DLMA, BFMA, HSMA, RFMA, CFMA and DAMA respectively. Columns contain results for different values of normalised residuals: 0.01, 0.03, 0.05, 0.1 respectively.

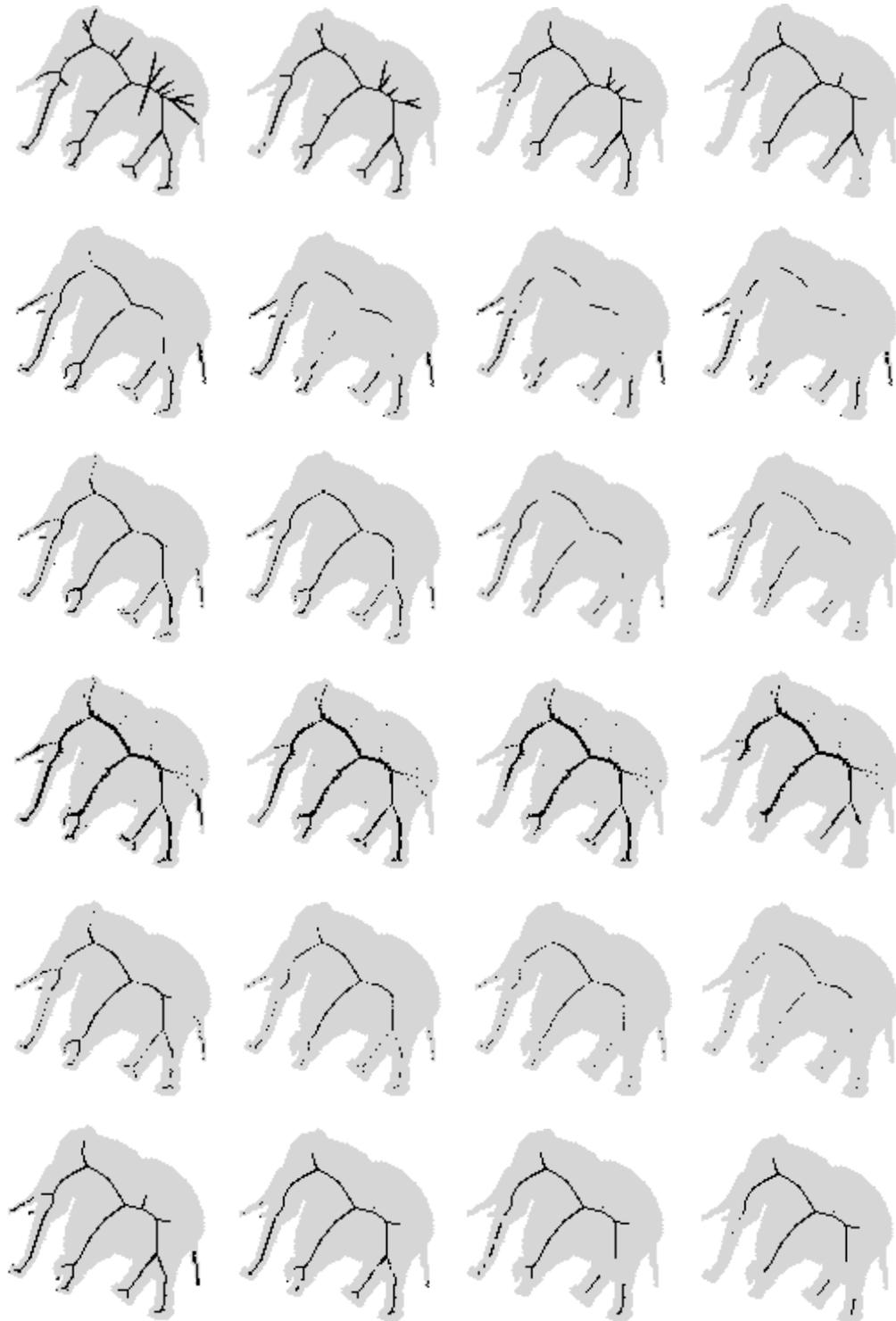


FIGURE 7.13: Medial axes (in black) superimpose to input object (in grey). Consecutive rows (from top to bottom) contain results for DLMA, BFMA, HSMA, RFMA, CFMA and DAMA respectively. Columns contain results for different values of normalised residuals: 0.01, 0.03, 0.05, 0.1 respectively.

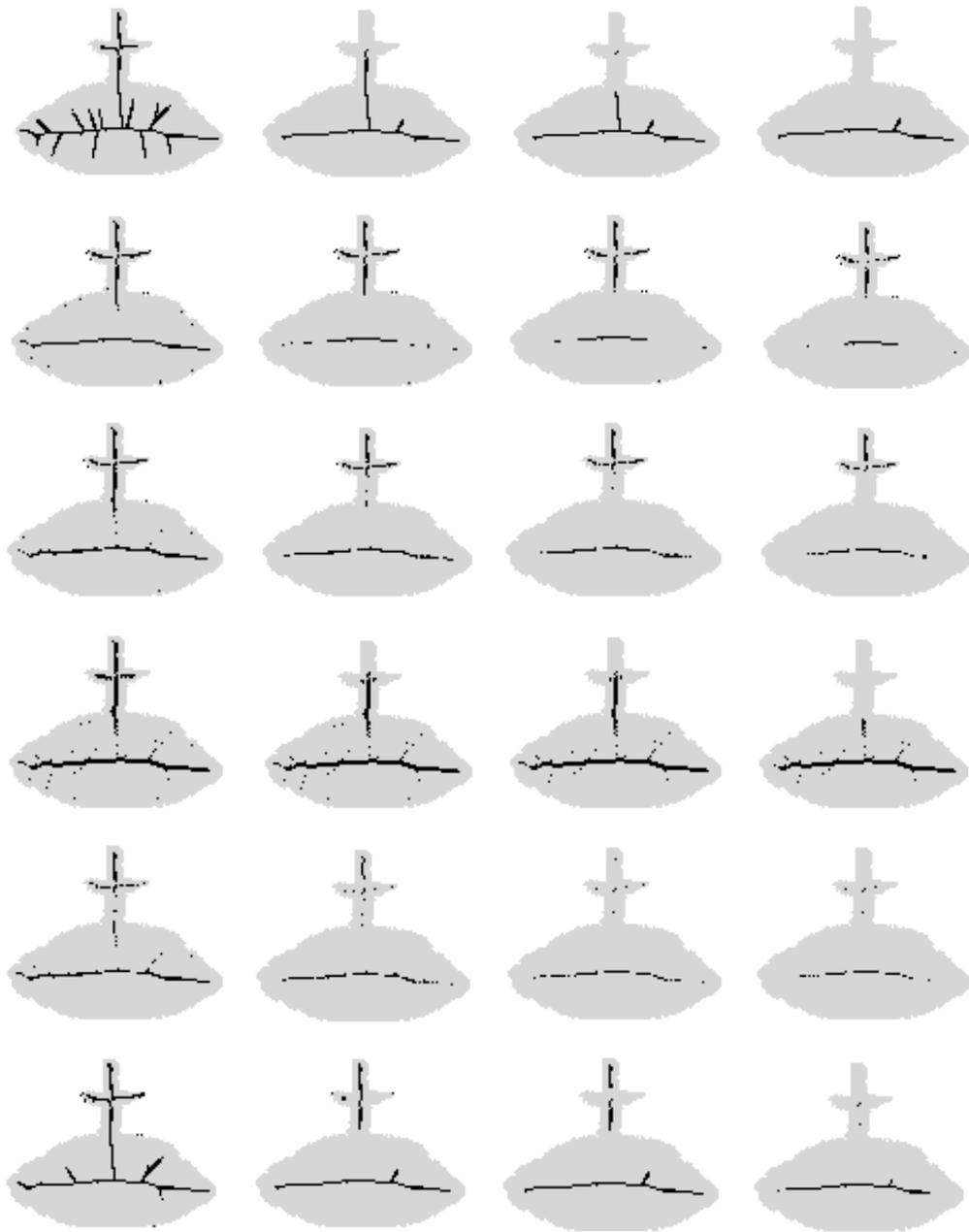


FIGURE 7.14: Medial axes (in black) superimpose to input object (in grey). Consecutive rows (from top to bottom) contain results for DLMA, BFMA, HSMA, RFMA, CFMA and DAMA respectively. Columns contain results for different values of normalised residuals: 0.01, 0.03, 0.05, 0.1 respectively.

7.4.2 The size of medial axis

We call $NS_X(Y)$ *normalised medial axis size* defined as a ratio of number of the medial axis points to number of object points:

$$NS_X(Y) = |Y|/|X|$$

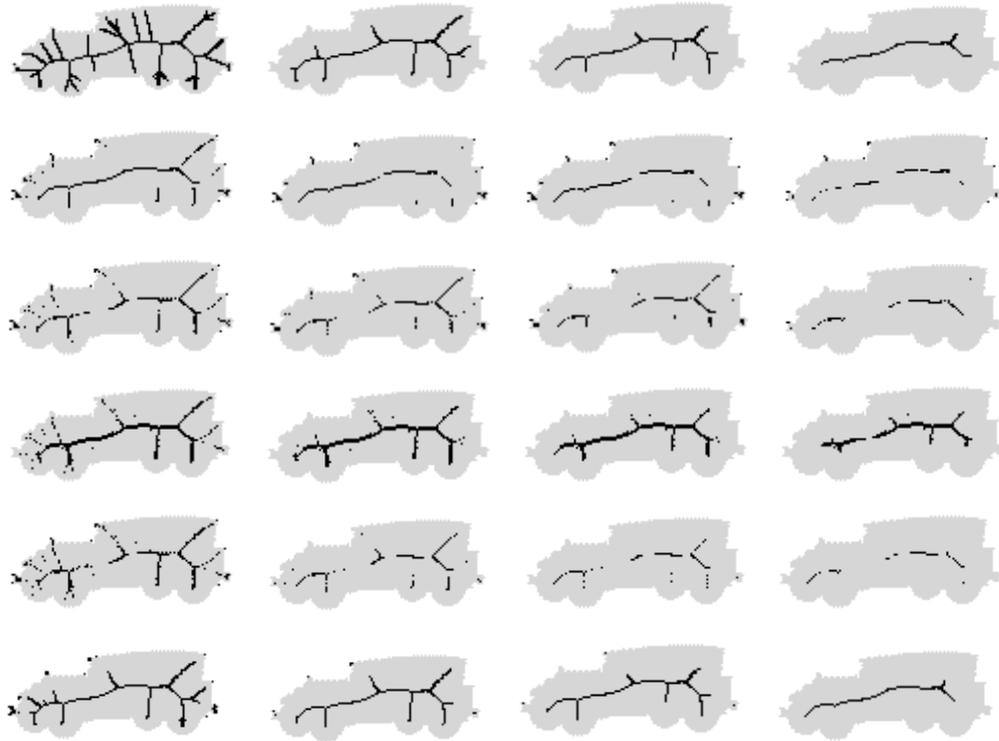


FIGURE 7.15: Medial axes (in black) superimpose to input object (in grey). Consecutive rows (from top to bottom) contain results for DLMA, BFMA, HSMA, RFMA, CFMA and DAMA respectively. Columns contain results for different values of normalised residuals: 0.01, 0.03, 0.05, 0.1 respectively.

Now we can compare normalised residuals obtained using different methods for the same NS . In other words, we replace the parameters of the different medial axis filtering algorithms by only one parameter (NS), which ensures a fair comparison.

In Fig. 7.16 we show the normalised residuals R as a function of normalised medial axis size NS for an exemplary shapes and different filtering methods. The results shows that the CFMA obtains the smallest residual values for the very small normalised sizes of filtered medial axis. When the size of the medial axis is large enough the CFMA behave similarly to BFMA and HSMA. It means that CFMA, BFMA and HSMA, respectively, at the first place remove points which not affect the reconstruction error. In contrast to RFMA and DLMA where the important, for reconstruction, points are removed even for very large size of medial axes. This superiority of CFMA is expected according to the definition. Note that DAMA fares better than original DLMA for shapes with features at low scale, where DAMA behavior is comparable to the methods yielding the best results in this experiment.

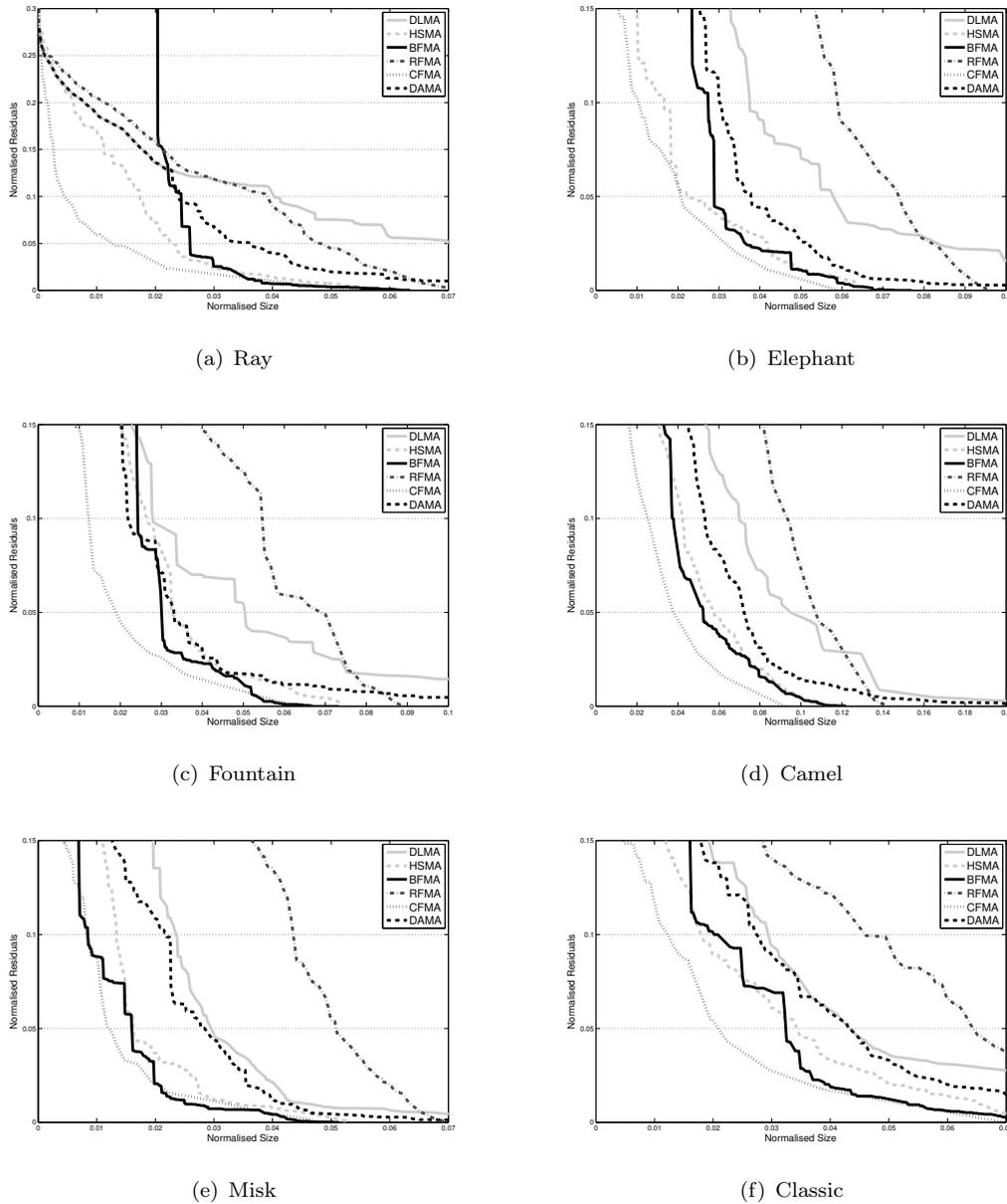


FIGURE 7.16: Residuals as a function of normalised medial axis size for different filtering criteria. Results generated for 6 selected shapes (see Fig 7.11).

7.4.3 The border dissimilarity

From the shape understanding point of view, preserving visually important segments and removing unimportant ones, at the same time, is even more important than achieve very good reconstruction property. Based on already presented comparison methodology, is hard to observe that small visually important object features disappear due its relative small area. Therefore, in this section, we introduce another comparison strategy which allow us to observe how different filtering strategies preserve visually important features.

Globally, the filtering algorithms will be tested by measuring the "difference" between the border of an object X , and the border of the reconstructed shape after filtering. The "difference" will be measured using the dissimilarity measure proposed by Dubuisson-Jolly and Jain (1994) and defined as follows:

Let X, Y be two subsets of \mathbb{R}^n . We set

$$D(X | Y) = \frac{1}{|X|} \sum_{x \in X} \min_{y \in Y} \{d(x, y)\}$$

and $d_D(X | Y) = \max\{D(X | Y), D(Y | X)\}$ is the *Dubuisson and Jain's dissimilarity measure between X and Y* (called shortly *dissimilarity* in the sequel).

The dissimilarity allow us to increase importance of the small features of the object in our experiments. As example consider the case when we "shrink" the large object with one long and thin branch, for example, by downscale it by factor 2. The results number of residuals will be high, however the dissimilarity will be relatively low, and the object will be still fully understandable. On the other hand when we remove only the thin branch, the residuals will be relatively low, however the dissimilarity will be high due to large difference in object geometry. In Fig. 7.17 we show even simpler example where after filtering the number of residuals will be exactly the same in both cases, however the dissimilarity will be very different.

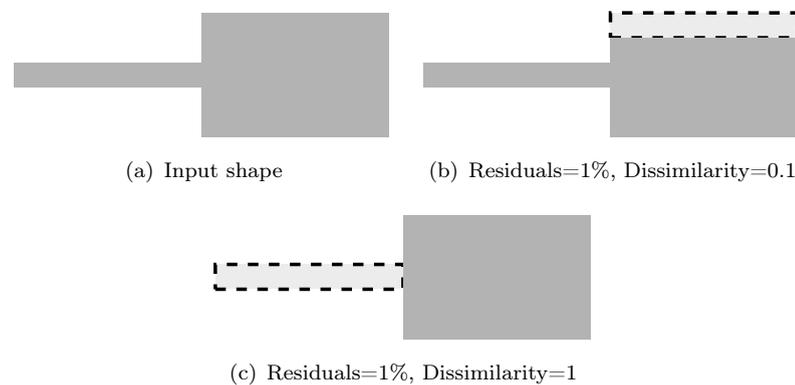
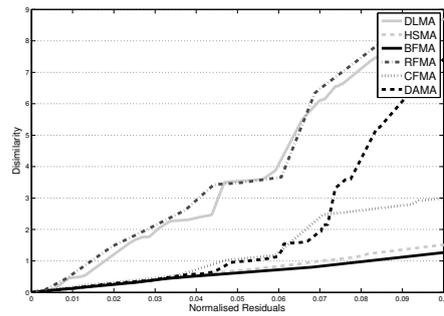


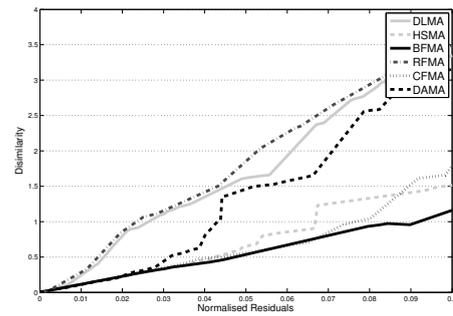
FIGURE 7.17: Example of the dissimilarity measure. (a): Input shape (in gray); (b, c): Some part of the shape was removed. Note that the size of removed part is exactly the same. In both cases the number of residuals will be the same, however, dissimilarity will be much higher in case of (c).

In Fig. 7.18 we show the dissimilarity as a function of normalised residuals for an exemplary shapes and different filtering methods. The results shows that in all cases BFMA produce lower dissimilarity than other filtering criterions for the same number of normalised residuals. It means that BFMA preserve more visually important features. The HSMA produce comparable results to BFMA. We can also observe good behavior of the CFMA and DAMA up to some level of residuals. After reaching this level we start to

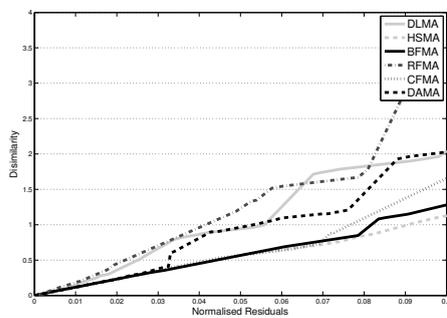
removing features at low scale which cause dissimilarity to rapidly increase. On the other hand, DLMA and RFMA gives only comparable results for the object without features at different scales and very poor results for another. Which is expected behavior, since it is well known that DLMA and RFMA do not preserve visually important features at low scale.



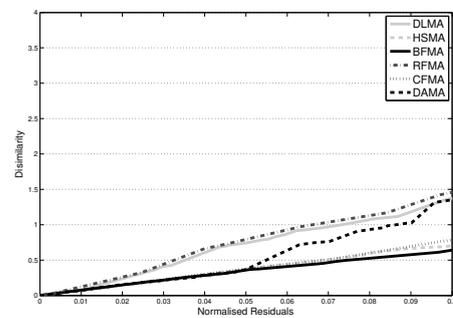
(a) Ray



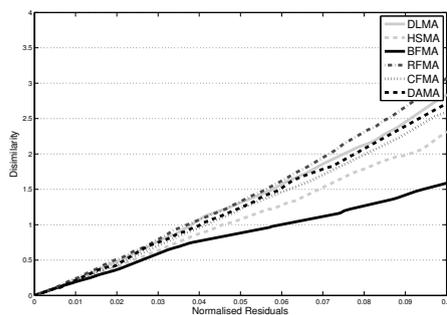
(b) Elephant



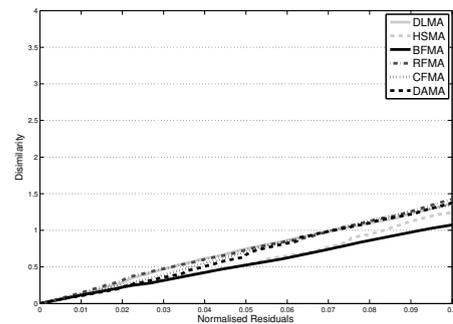
(c) Fountain



(d) Camel



(e) Misk



(f) Classic

FIGURE 7.18: Dissimilarity as a function of normalised residuals for different filtering criterions. Results generated for 6 selected shapes (see Fig 7.11).

7.4.4 The complexity of the skeleton

Using medial axis filtering strategies we can observe that some unimportant points (close to the border) or spurious branches left unfiltered in addition to fully removes element at low scale. Therefore, it is interesting to compare the ability of filtering methods to filter such unimportant points and preserve features at different scale at the same time.

We propose another comparison strategy which relay on similar strategy to [Eede et al. \(2006\)](#). However, instead of using weighted reconstruction error, we simply evaluate residuals at different level of skeleton complexity, defined as number of skeleton end points. More precisely, at each level of the complexity (see [Fig. 7.19](#)) we look for the lowest possible reconstruction error.

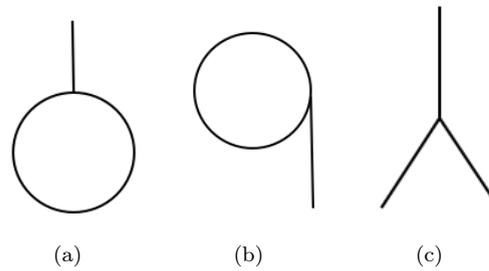


FIGURE 7.19: (a): A skeleton with complexity $c = 1$; (b): Another example of a skeleton with complexity $c = 1$; (c): A skeleton with complexity $c = 3$.

To evaluate skeleton complexity, first we need to generate ultimate homotopic skeleton of the shape X based on information derived from its filtered medial axis Y . In order to guarantee topology preservation, we can perform an ultimate homotopic thinning of X with the constraint of retaining the points of its filtered medial axis Y (constraint set), that is, we can iteratively remove simple points from X that do not belong to Y (see [Fig. 7.20](#)). A priority function is needed in order to specify which points must be considered at each step of the thinning. In the general case, the choice of this priority function is not obvious as reported in ([Couprie et al., 2007](#); [Talbot and Vincent, 1992](#)). For example, taking the exact Euclidean distance map as a priority function may cause "extra branches" to appear which change the skeleton complexity and unable proper comparison. Taking this fact into consideration we choose function \mathcal{F}_X defined in ([Chaussard et al., 2010](#)) as priority. Now, based on skeletonisation results, we can evaluate number of skeleton end points. It can be done simply, for example, thanks to notion of topological numbers introduced in ([Bertrand, 1994](#)).

We denote by $Y(X, p)$ *filtered medial axis at value p* . Notice that, meaning of parameter p can be different for each medial axis extraction algorithms (for our approach, p can be equivalent to, for example, scale factor s or the ball radius r). Let $c \in \mathbb{N}_+$, we denote by $Y_c(X, p)$ *filtered medial axis at value p with complexity c* . Notice that, in some cases set

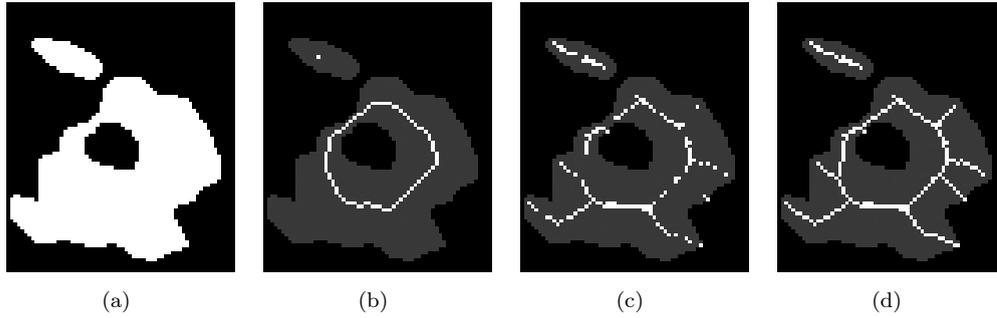


FIGURE 7.20: (a): a set X (in white); (b) ultimate homotopic skeleton of X ; (c): medial axis of X ; (d): ultimate homotopic skeleton of X constrained by the medial axis of X .

In (b,c,d) the original set X appears in gray for comparison.

$Y_c(X, p) = \emptyset$, it means that, parameter p which produce the medial axis with complexity c does not exist .

We call $R_X^c(Y(X, p))$ the *minimum residuals of $Y(X, p)$ at complexity level c* , and defined by

$$R_X^c(Y(X, p)) = \min\{R_X(Y_c(X, p))\}$$

Now we can simply compare medial axes at each complexity level independently by changing the parameter c without cares about meaning of used filtering parameter p . We can also make comparison by focus only on the most suitable level for whole family of shapes which are of interest. For example if we take into consideration shape similar to the letter "p" or "b" the most interesting complexity level is one, because both shapes have one visually important branch. Therefore, to compare filtered medial axes of such family of shapes we use value of $R_X^1(Y(X, p))$. Notice that, if in results skeleton, additional branch appear in wrong place due to wrong filtering and the important one will disappear at the same time. We still have the complexity at level one. However, residuals should be higher due important part of the shape is missing after reconstruction. Therefore, we have more robust information about quality of the used filtering procedure.

Figure 7.21 show the minimum residuals at different complexity level. We can notice that HSMA gives the lowest residual at the reasonably complexity according to the shape. The HSMA gives only poor results for very low complexity, since, the method very long preserve features at different scale. If they are finally removed, the resulting medial axes are almost fully filtered. Comparable results are generated by CFMA and DAMA which ranks second. The BFMA has the worst behavior in this experiment, since it preserve points which are close to the border. Therefore, the BFMA is unable to generate medial axes with low complexity which can strongly limits its practical application.

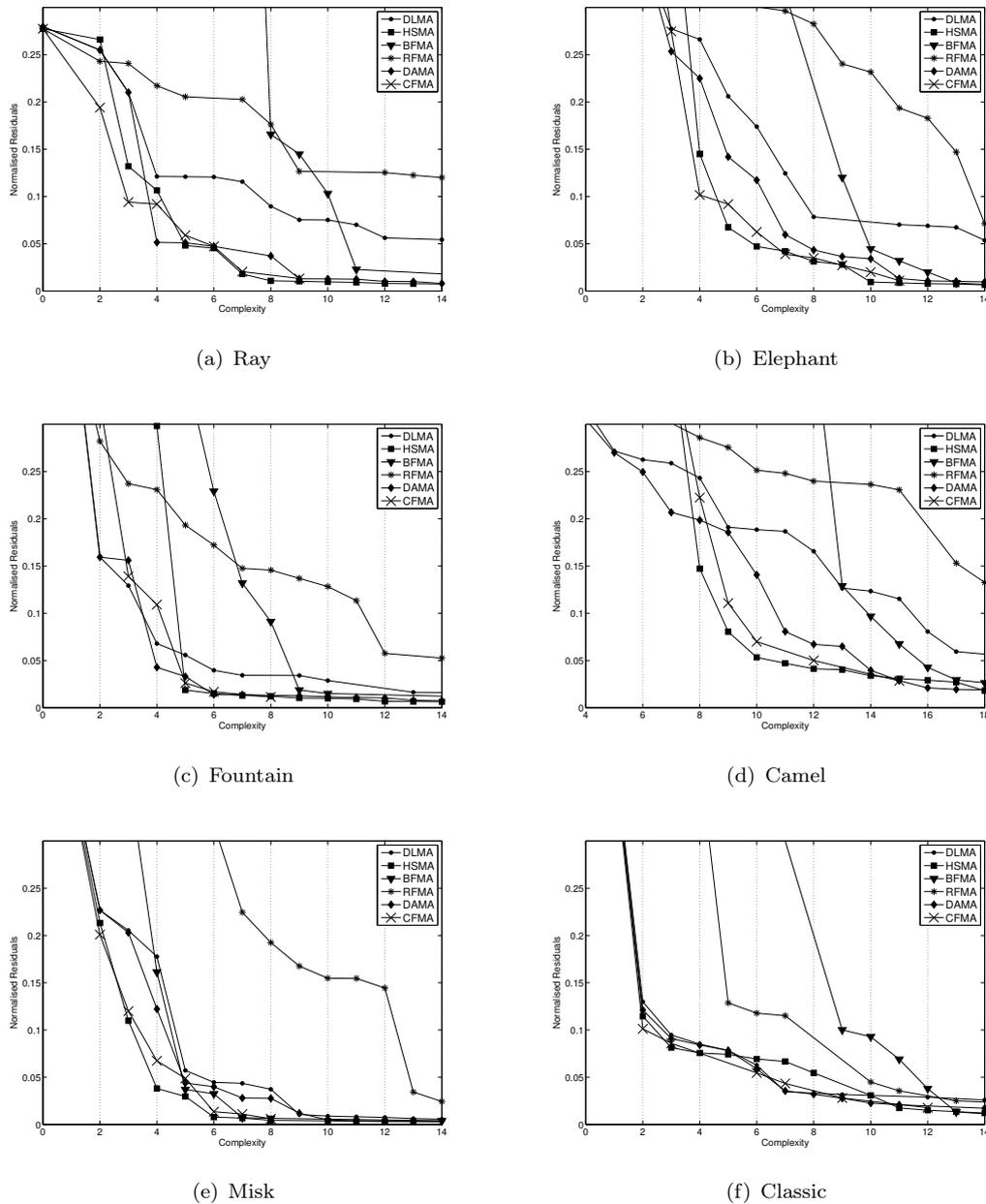


FIGURE 7.21: Number of minimum residuals according to the complexity level, calculated for different filtering criteria. Results generated for 6 selected shapes (see Fig 7.11). Each marker (e.g. triangle, circle and square) represents parameters of one filtered medial axis. Lines has been added only to emphasize the trend of measurements.

7.4.5 The multiscale representations

The last interesting property is related to the hierarchy property of the medial axes (see definition 7.5 in chapter 7.3.2). This property is important when we are interested in multiscale representation of an input object and strongly depends on application. For example, in some applications one can be more interested in filtering strategy which not

only remove large amount of points at once, but also allow to produce some intermediate results at different level of hierarchy.

Since all method, included in the comparison, holds the hierarchy property we can simply define the hierarchy level as follows:

Let $i \in \mathbb{Z}_+$ We denote by $MA_s(X)$ the filtered medial axis of $X \subset \mathbb{Z}^n$ for $s \in S \subseteq \mathbb{R}_+$, where S represents the set of all possible values of parameter s of considered filtering algorithm. The family H^i is called i -th hierarchy level, and defined by

$$H^i = \{MA_{s_i}(X) \mid s_i, s_{i-1} \in S, s_i > s_{i-1}, MA_{s_i}(X) \subset MA_{s_{i-1}}(X) \subset MA(X)\}$$

Then, we can simply evaluate reconstruction error at each hierarchy level independently. Note that we do not consider values of s which do not cause any filtering, since such s does not change hierarchy level. Therefore, the different meaning of the filtering parameter, in each method, does not cause the unfair comparison.

When we increase hierarchy level we cause the filtering algorithm to remove a set of medial axis points with the same "importance". The "importance" of point depends on used filtering criterium. The filtering method which removes smaller amount of points at each change of hierarchy, can produce more different representation of the object.

In the Fig. 7.22 we show achieved number of normalised residuals at each hierarchy level for all tested filtering methods and for different shapes. Based on results we can observe how many different medial axes can be produced by manipulating the filtering parameter and how the reconstruction error evolves.

For the first four objects which have many features at different scale the DAMA generate the highest number of multiscale representation of the object. The HSMA ranks second. For the last two classes of objects HSMA and DAMA yield comparable and globally the best result. In contrast to the CFMA which yield globally the smallest number of the hierarchies. In this case, the number of different filtered medial axes generated with CFMA, RFMA and DLMA might be not enough in some application. The relatively small number of medial axes are also generated by RFMA and DLMA.

7.4.6 Discussions

The usefulness of each filtering method is strongly application dependent. Therefore, each of presented experiment gives us the information which filtering method is the best in respect to evaluated property. However, each of presented methodology alone is not sufficient to clearly show which one filtering criterium is the best one at the global scope. In addition, some of evaluated properties are not compatible. For example, at the same time we would like to minimize reconstruction error, but filter us much point as possible

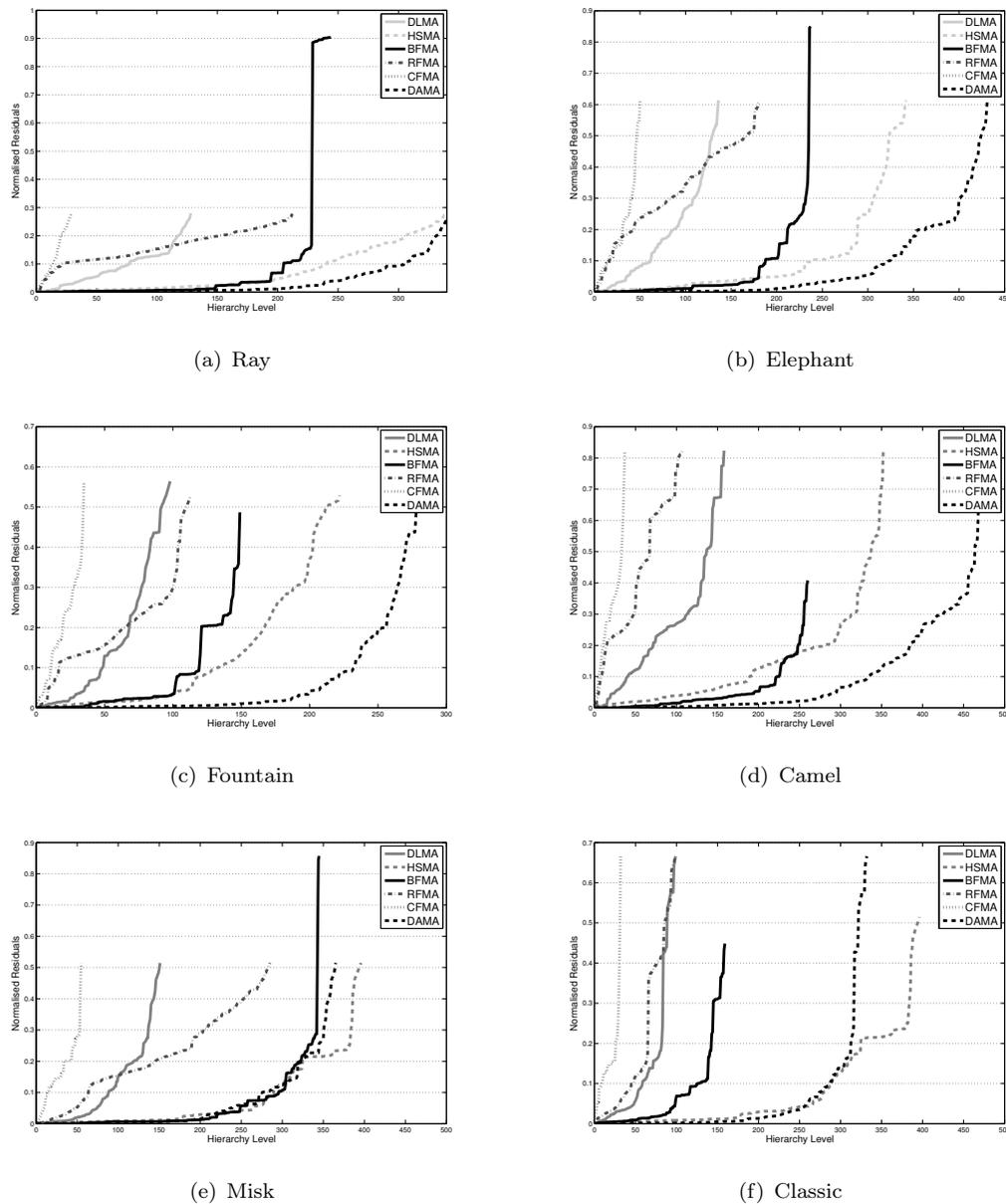


FIGURE 7.22: Residuals as a function of hierarchy levels for different filtering criteria. Results generated for 6 selected shapes (see Fig 7.11).

which can be done by removing relatively small medial balls at the first place. On the other hand, we would like to accurately reconstruct the geometry of the object, therefore we need to preserve some of them and allow to remove bigger balls first, but without preserving points close to the border which increase complexity. Such incompatibility is very well visible on the results achieved by BFMA where for medial axes of small size BFMA gives the lowest dissimilarity. Unfortunately, the complexity of BFMA is very high.

To objectively compare different filtering criteria at global scope we need to look at results of all experiments at the same time. Now, we can see that HSMA yields the

best or comparable results to the best one in all experiments. Moreover, HSMA allow to produce large set of hierarchies which makes it very flexible. We can say that HSMA give us the best balance between the reconstruction error, size of medial axis and complexity, and allow to preserve features at different scale. Therefore, HSMA can be considered as the best method in our comparison.

We can also notice the good behavior of the CFMA, only the lowest number of possible hierarchies, and high dissimilarity for larger residuals can limits its application. The DAMA in average also produce comparable results which are much better then original DLMA. At the same complexity level DAMA overcomes CFMA and allow to produce the largest set of hierarchies. In addition, at some point DAMA also start to removes features at low scale. This property of DAMA and CFMA can be considered as an asset if such features are irrelevant according to application. However, DAMA is much more flexible than CFMA, since it can produce more multiscale representations.

The last two methods in our comparison, that is, DLMA and RFMA, gives the worst results in our experiments. This is expected situation for both methods, since, the filtering relay on the ball size and the filtering parameter id defined globally. However, in contrast to RFMA, which is the simplest method and in practice is used only as a supplementary criterium, DLMA holds another interesting properties which are out of scope of this comparison. For example, stability to noise or rotation, makes DLMA very useful in practical application, interested reader can see ([Chaussard, 2010](#); [Chaussard et al., 2009](#)) for comprehensive evaluation of different DLMA properties.

7.5 Conclusions

In this chapter we propose two new medial axis filtering strategies. The first one HSMA, inspired by the scale axis method preserve important features of an input object from shape understanding point view even if they are at different scale. It also overcomes the most important drawbacks of scale axis: scale axis is not in general a subset of filtered medial axis. It is even not necessarily a subset of the original shape.

The second proposed method, DAMA based on DLMA utilize the information derived from the Euclidean opening function to dynamically adapt filtering parameter and allow to preserve important features at different scale.

The proposed methods and theirs properties where presented in 2D space, but it can be easily extended to any dimension. Experimental verification and comparison with previously introduced methods (RFMA, BFMA, DLMA, CFMA) shows that, at the global scope, HSMA is the best filtering method. DAMA produce results much better than DLMA and allow to use this medial axis in many more practical applications.

In this chapter, we also introduce an original methodology that ensures a fair comparison between different methods. This methodology could be of interest for comparing other medial axis filtering criterions.

Chapter 8

Tangent Estimator Along 3D Discrete Curve

8.1 3D Discrete Line Segment

Definition 8.1. A 3D digital line with main vector $(a, b, c) \in \mathbb{Z}^3$ such that $|a| \geq |b| \geq c$, shifts μ, μ' , and thicknesses e, e' , where $a, b, \mu, \mu' \in \mathbb{Z}$, $c, e, e' \in \mathbb{Z}_+$, is defined as the set of points $(x, y, z) \in \mathbb{Z}^3$ which satisfy the diophantine inequalities:

$$\mu \leq cx - az < \mu + e, \quad (8.1)$$

$$\mu' \leq bx - ay < \mu' + e', \quad (8.2)$$

and denoted by $\mathcal{D}_3(a, b, c, \mu, \mu', e, e')$.

The definition for coefficients ordered in ways different from $|a| \geq |b| \geq c$ may be obtained by permuting x, y, z as well as their coefficients.

It should be mentioned that the thicknesses e and e' control the adjacency relation of a 3D digital line (Coeurjolly et al., 2000). Hereafter we use one of the following three settings.

A 3D digital line $\mathcal{D}_3(a, b, c, \mu, \mu', e, e')$ is:

- a 6-curve if we set $e = |a| + c$ and $e' = |a| + |b|$,
- a 18-curve if we set either $e = |a| + c$ and $e' = |a|$, or $e = |a|$ and $e' = |a| + |b|$,
- a 26-curve if we set $e = e' = |a|$.

The points of any such k -curve C may be totally ordered by using the above property. So we can define a set of consecutive points of C from the i -th point to the j -th point, denoted

by $C_{i,j}$. With this definition of a part of a k -curve, we can define the following two notions, 3D digital straight segment and 3D maximal segment, which were originally defined for 2D (Debled-Rennesson and Reveillès, 1995; Feschet and Tougne, 1999; Lachaud et al., 2007) (see chapter 6.3.2) and can be extended to 3D.

Definition 8.2. Given a k -curve C , a set of its consecutive points $C_{i,j}$ where $1 \leq i \leq j \leq |C|$ is said to be a 3D digital straight segment if there exists a digital line \mathcal{D}_3 containing all the points of $C_{i,j}$. In this case, we say that the property $S(i,j)$ holds.

The next property (Coeurjolly et al., 2000) is led by Definitions 8.1 and 8.2.

Proposition 8.3. Property $S(i,j)$ is verified iff two of the three projections of $C_{i,j}$ on the basic planes O_{XY} , O_{XZ} and O_{YZ} are 2D straight line segments.

Figure 8.1 illustrates an example of curve C which verify the property $S(i,j)$.

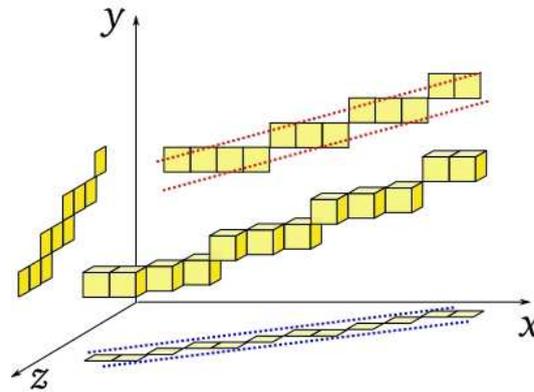


FIGURE 8.1: An example of 3D curve projected on the basic planes O_{XY} , O_{XZ} and O_{YZ} . The curve forms the 3D straight line segment since two projection (O_{XY} , O_{XZ}) are 2D straight line segments.

8.2 3D Tangential Cover Construction

The main part of the 3D λ -MSTD estimator consists in obtaining the tangential cover \mathcal{M} of a given k -curve C , similarly to the original 2D estimator (Lachaud et al., 2007). Thanks to Property 8.3, we give Algorithm 10, which finds the set of maximal segments by recognizing two 2D digital straight segments of the projections of C . Figure 8.2 illustrates an example of C and its tangential cover \mathcal{M} .

8.2.1 Complexity evaluation

Algorithm 10 performs two loops. The first one starts from line 04 and makes $|C|$ iterations. The next loop, nested in the first one, starts from line 07 and in worst case

Algorithm 11 Tangential Cover (**Input** C ; **Output** \mathcal{M})

```

01.  $\mathcal{M} \leftarrow \emptyset$ 
02.  $\mathcal{N} \leftarrow$  empty queue
03.  $\mathcal{E} \leftarrow$  NULL
04. foreach point  $p_i \in C$  do
05.    $M_i \leftarrow \emptyset$ 
06.   Push_element( $M_i, \mathcal{N}$ )
07.   foreach element  $M_j$  in  $\mathcal{N}$  do
08.      $s \leftarrow 0$ 
09.     If isDSS_XY( $M_j \cup \{p_i\}$ ) then  $s \leftarrow s + 1$  end
10.     If isDSS_XZ( $M_j \cup \{p_i\}$ ) then  $s \leftarrow s + 1$  end
11.     If isDSS_YZ( $M_j \cup \{p_i\}$ ) then  $s \leftarrow s + 1$  end
12.     If  $s \geq 2$  then  $M_j \leftarrow M_j \cup \{p_i\}$  end
13.     else If  $s < 2$  then
14.       If  $\mathcal{E}(i)$  is NULL then
15.          $\mathcal{E}(i) \leftarrow j$ 
16.          $\mathcal{M} \leftarrow \mathcal{M} \cup M_j$ 
17.       end
18.       Remove_first_element( $\mathcal{N}$ )
19.     end
20.   end
21. return  $\mathcal{M}$ 

```

Remark:

In Algorithm 10 we denote by $\mathcal{E}(i)$ the table which stores the first point of each maximal segment ending at the i -th point.

performs $|\mathcal{N}|$ iterations, where \mathcal{N} is a set of all maximal segments covering an actual point. The size of \mathcal{N} is bounded by a finite integer value; in fact $|\mathcal{N}| \leq 22$ on average in 2D de Vieilleville (2007). Thus, the presented algorithm is linear in number of points of the curve C if the incremental procedure "isDSS" in steps from 9 to 11 has a constant time complexity. Indeed, for this procedure, we can use one of the efficient methods for 2D digital straight line recognition in constant complexity, for example used in Faure et al. (2009), after projecting the current set of points on the planes O_{XY} , O_{XZ} , O_{YZ} .

8.3 Three-dimensional λ -MSTD Estimator

Similarly to the original 2D estimator Lachaud et al. (2007), the 3D estimator at a point x of a k -curve C should depend on the set of all maximal segments going through p . Let us number all the maximal segments of the tangential cover \mathcal{M} of C by increasing indices such that $M_i \in \mathcal{M}$ for $i = 1, 2, \dots$. Then, such a set is defined by $P(x) = \{M_i \in \mathcal{M}, x \in M_i\}$, and is called the *pencil of maximal segments* around x . As any point x of a k -curve C is covered by at least one maximal segment, we have the next property similarly to the 2D one (Feschet, 2010).

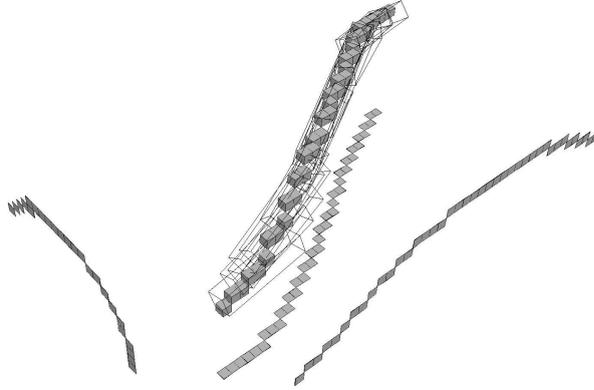


FIGURE 8.2: A 3D digital curve and its tangential cover calculated by Algorithm 10. Three projection on the basic planes O_{XY} , O_{XZ} and O_{YZ} of the curve were also depicted.

Proposition 8.4. *The pencil of maximal segments $P(x)$ of any point x of a k -curve C is never empty.*

In addition, as noted in [Feschet and Tougne \(1999\)](#) for 2D cases, several successive points may have the same pencil, and this is also observed for 3D cases. Therefore, the tangent estimator should take also into account the position of the point p within the pencil. More specifically, each point p has a certain eccentricity with respect to each maximal segment. Let us denote by $L_i = \|n_i - m_i\|_1$ the length of each $M_i = C_{m_i, n_i} \subset C$. Then, the *eccentricity* $e_i(x)$ of a point x with respect to a maximal segment M_i is its relative position between the extremities of M_i , that is,

$$e_i(x) = \begin{cases} \frac{x - m_i}{L_i} & \text{if } M_i \in P(x), \\ 0 & \text{otherwise.} \end{cases} \quad (8.3)$$

The tangent direction is thus estimated by a combination of the directions of maximal segments weighted by a function of the corresponding eccentricity. The function λ maps from $[0, 1]$ to \mathbb{R}_+ with $\lambda(0) = \lambda(1) = 0$ and $\lambda > 0$ elsewhere. For example, a C^2 function $64(-x^6 + 3x^5 - 3x^4 + x^3)$ can be used. Different proposition of several λ functions and theirs impact on the results are shown in the next section. More detailed information about the required properties of the λ function can be found in [Lachaud et al. \(2007\)](#).

Definition 8.5. The 3D λ -MSTD, that is, $\vec{t}(x)$ at point x of a k -curve C is defined as a weighted combination of the vectors \vec{t}_i of the covering maximal segments M_i such that

$$\vec{t}(x) = \frac{\sum_{M_i \in P(x)} \lambda(e_i(x)) \frac{\vec{t}_i}{|\vec{t}_i|}}{\sum_{M_i \in P(x)} \lambda(e_i(x))}. \quad (8.4)$$

From Property 8.4 and the nature of the eccentricity, this value is always defined and computed with a linear time complexity.

8.4 Experiments on Parametric Curves

8.4.1 Multigrid convergence

Since for a given shape there exists infinitely many shapes with the same digitization, it is difficult to compare objectively accuracy of different digital geometric estimators. In this thesis we evaluate quality of the estimator using one of the existing objective criteria, that is, we use *multigrid convergence* property. The estimator which hold this property ensures that the thinner digitization step brings a better approximation. Moreover, geometric estimator which converge faster should give more accurate results at the same resolution. This is very important in our application where the resolution is limited by the CT data.

In the literature many authors report about multigrid convergence property of the several digital geometric estimators like for example: area (Huxley, 1990), moments (Klette and Žunić, 2000), perimeter (Asano et al., 2001; Kovalevsky and Fuchs, 1992; Sloboda et al., 1998) and tangents (de Vieilleville and Lachaud, 2006; Coeurjolly, 2002; Lachaud et al., 2005a, 2007). Recently Roussillon and Lachaud (2011) propose the first curvature digital estimator which is also multigrid convergent.

Following the (Klette and Rosenfeld, 2004) the formal definition of the multigrid convergence property can be presented as follows:

Let D_h , be some digitization process of step h . Let \mathbb{X} be a family of shapes in \mathbb{R}^n and let G be a geometric quantity defined for all $X \in \mathbb{X}$. A *discrete estimator of G* is a map that associates to a digitization $D_h(X)$ an estimation of $G(X)$.

Definition 8.6. A discrete estimator E_G is *multigrid convergent* toward G for \mathbb{X} and D_h if and only if, for any $X \in \mathbb{X}$, there exists some $h_X > 0$ for which

$$\forall_{0 < h < h_X}, |E_G(D_h(X)) - G(X)| \leq r(h), \quad (8.5)$$

where $r : \mathbb{R}^+ \rightarrow \mathbb{R}^{+*}$ has limit value 0 at $h = 0$. This function defines the *speed of convergence*.

8.4.2 Results and discussion

To verify our tangent estimator and evaluate its multigrid convergence behavior, we experiment it on several families of 3D curves (see Fig. 8.3). First, we measure different errors between the expected theoretical tangent and the estimated one with increasing resolutions using several λ functions (see Fig. 8.4). On Fig. 8.5 and 8.6, we can see how the root mean square error (RMSE) and maximal absolute error evolves. We can observe that in both cases a choice of λ function is not critical.

Second, we measure the speed of the convergence. The evaluated average speed of the RMSE convergence (see Fig. 8.7) is likely $O(h^{-\frac{2}{3}})$ which is an expected value similar to the original 2D λ -MSTD estimator (see Lachaud et al. (2007)). The speed of the maximal absolute error convergence is limited by $O(h^{-\frac{1}{2}})$ which is also expected value and correspond to the 2D case. Therefore, experiments shows that 3D λ -MSTD estimator has multigrid convergence property similar to the 2D case, however, this still need to be theoretically proved.

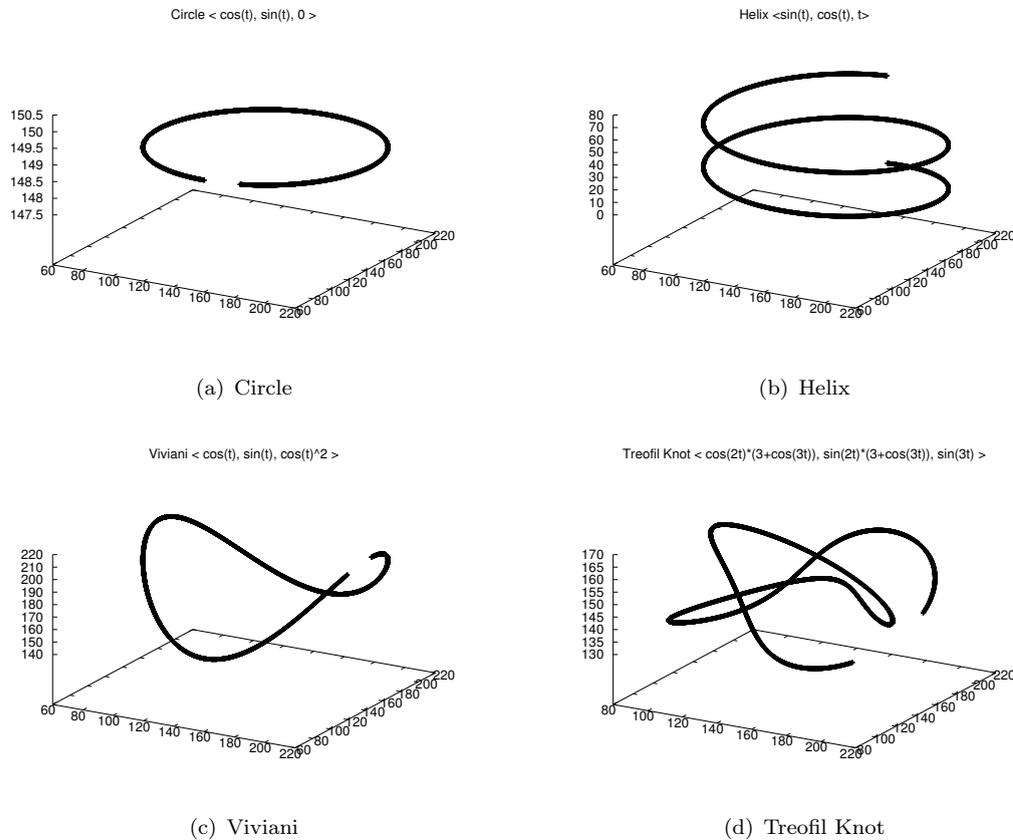
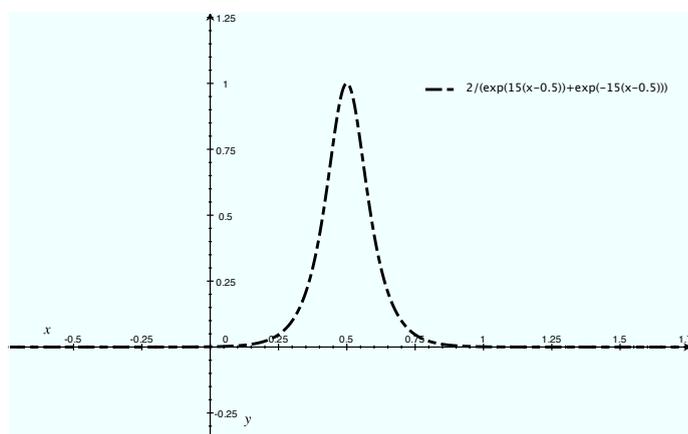


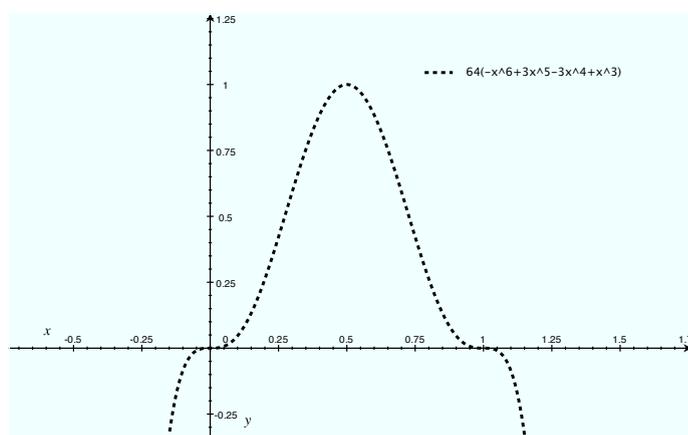
FIGURE 8.3: A set of 3D parametric curves used for the experiments.

To visualize the accuracy of the estimator, on Fig. 8.8, 8.9, 8.10 and 8.11 respectively, expected and estimated tangents in a fixed resolution along their principal x , y and z axis are shown. We can notice very good accuracy and we can expect based on the multigrid convergence property that if we increase resolution the result can be improved.

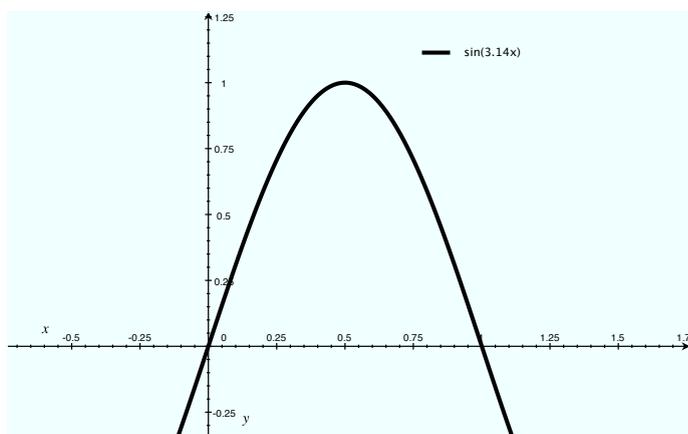
In the chapter 13 we compare the 3D λ -MSTD estimator with other tangent estimator (discrete and continuous) in practical application.



(a)



(b)



(c)

FIGURE 8.4: A three different proposition of the λ functions which can be used in λ -MSTD estimator.

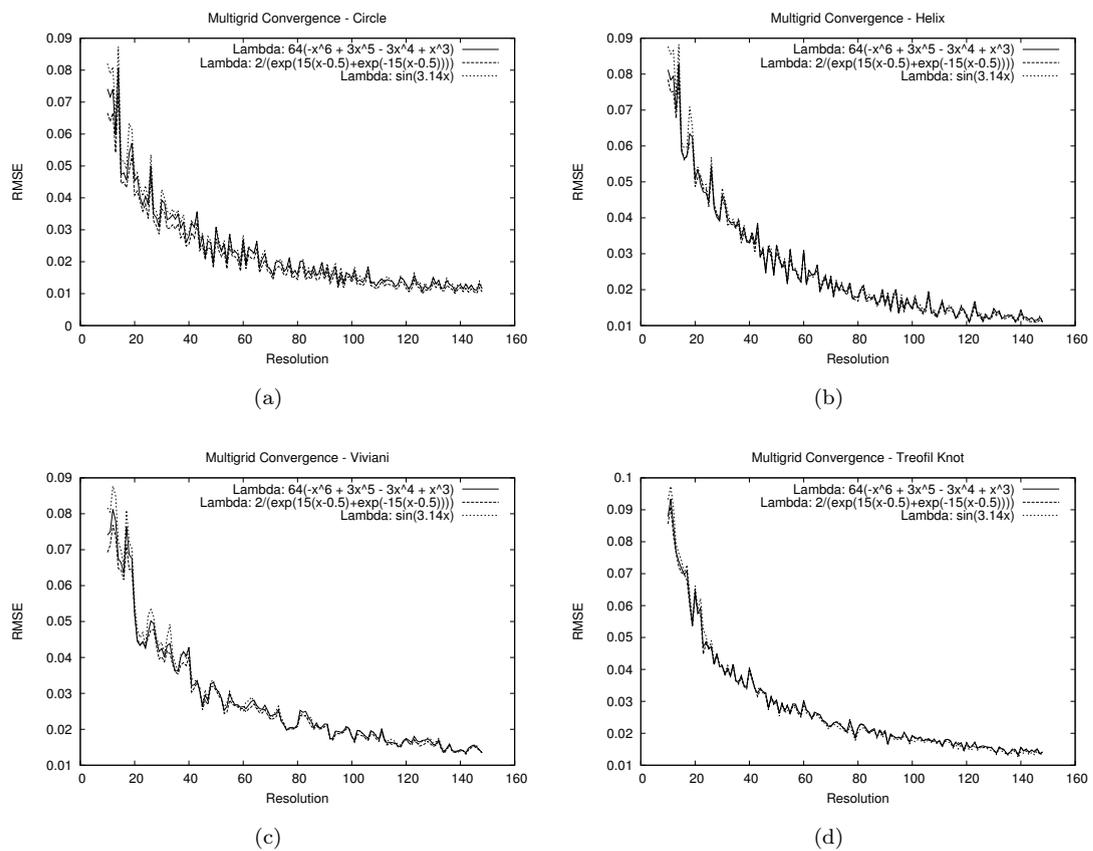


FIGURE 8.5: An evolution of RMSE between real and estimated tangent with increasing resolution for different definition of λ function.

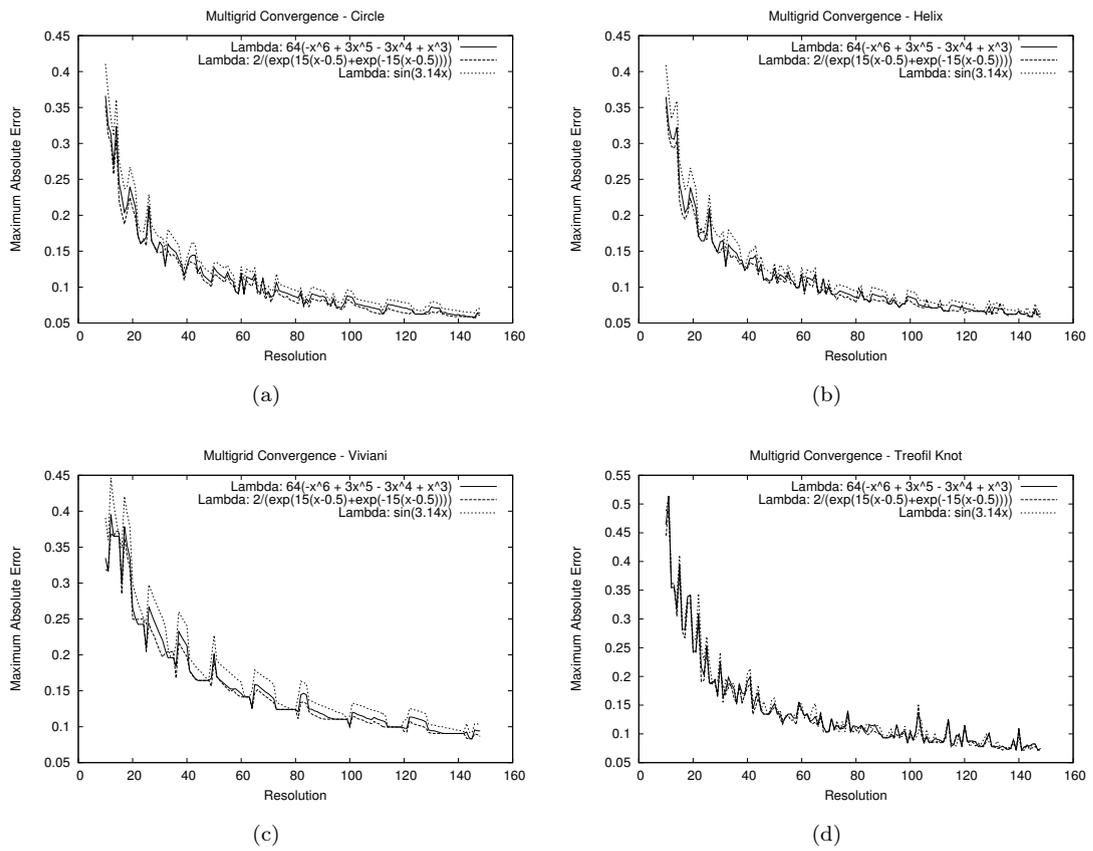


FIGURE 8.6: An evolution of maximal absolute error between real and estimated tangent with increasing resolution for different definition of λ function.

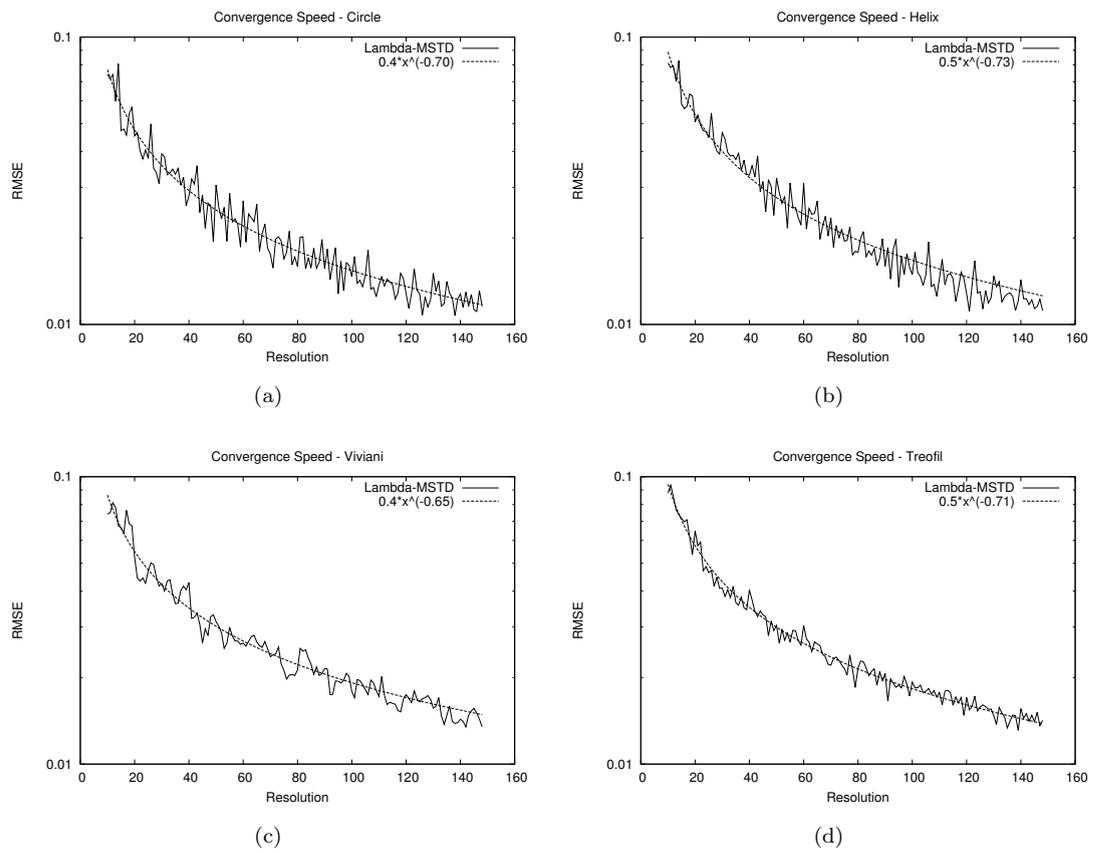
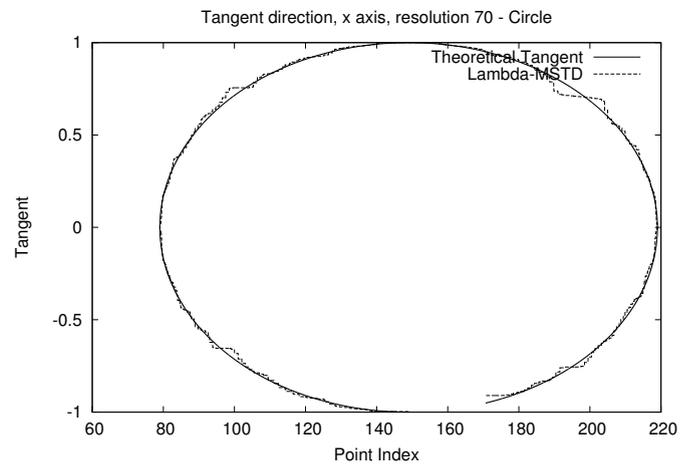
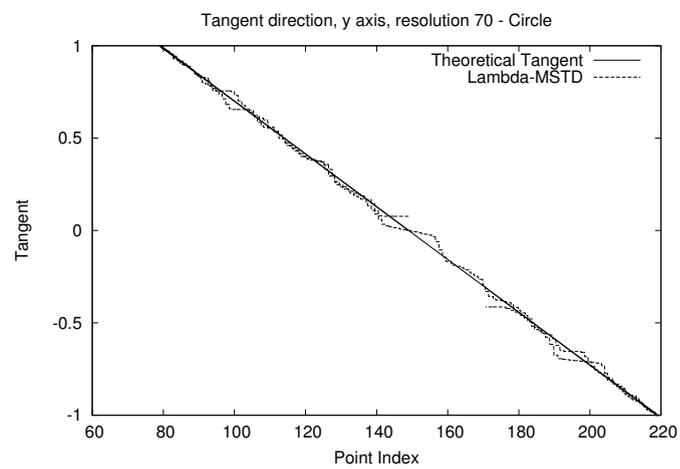


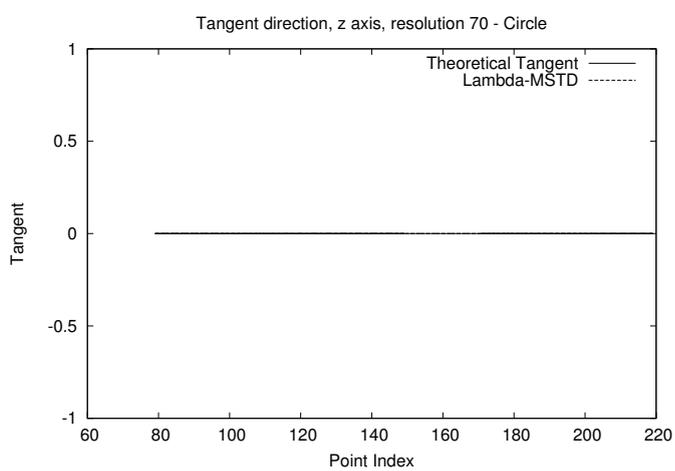
FIGURE 8.7: The speed of RMSE convergence with increasing resolution.



(a)

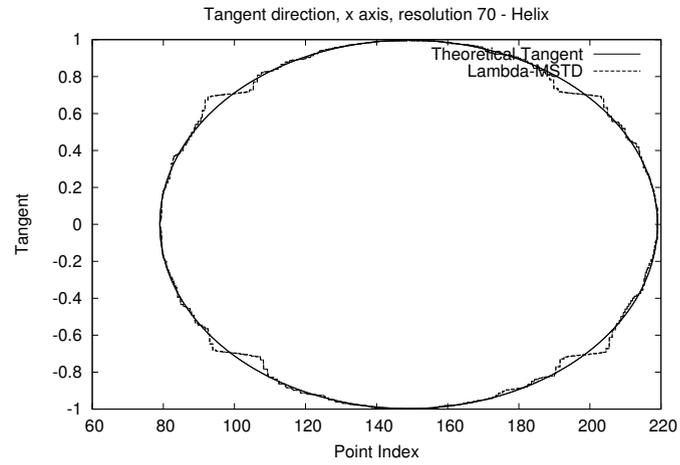


(b)

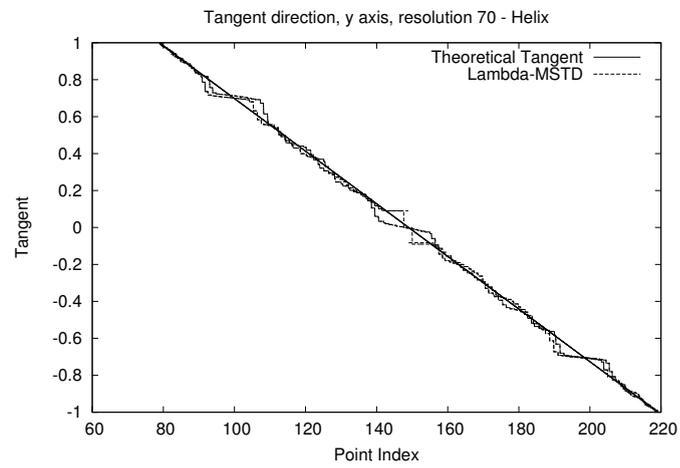


(c)

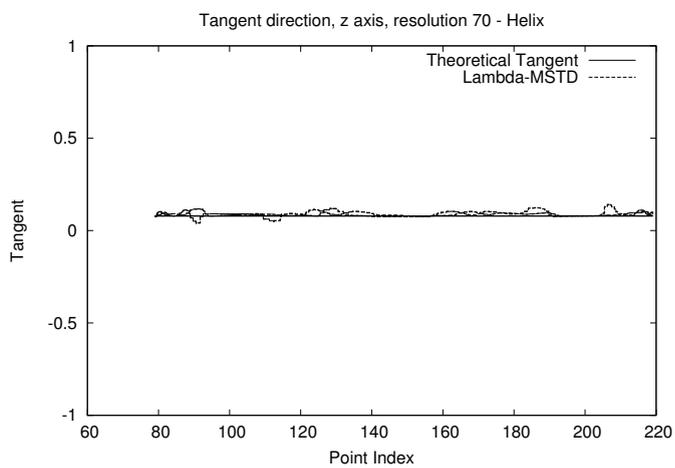
FIGURE 8.8: An expected (theoretical) and estimated tangent direction along 3D circle at fixed resolution, for three axes x , y and z , respectively.



(a)

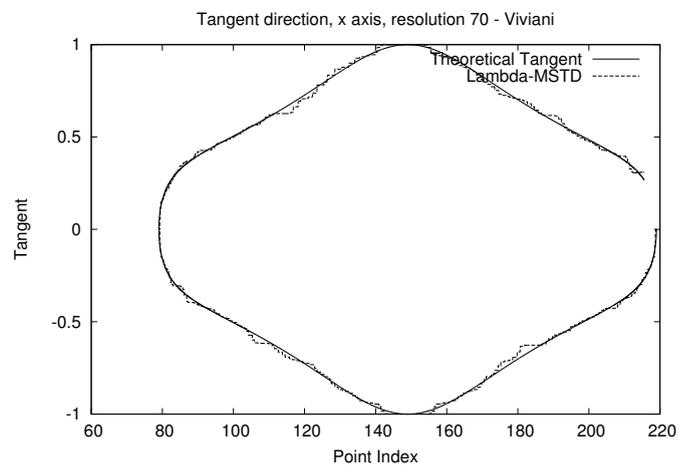


(b)

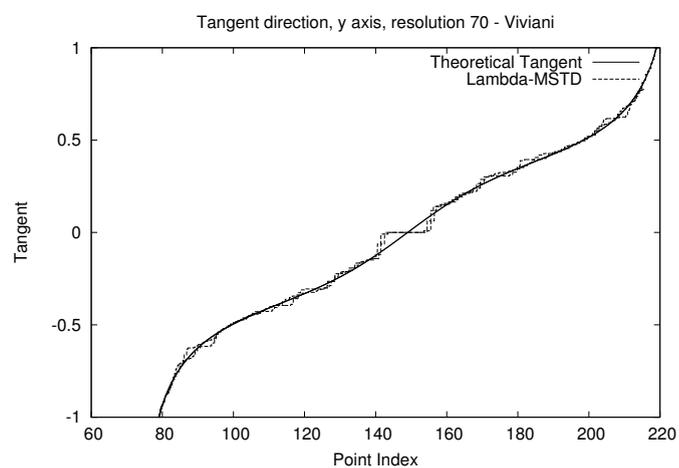


(c)

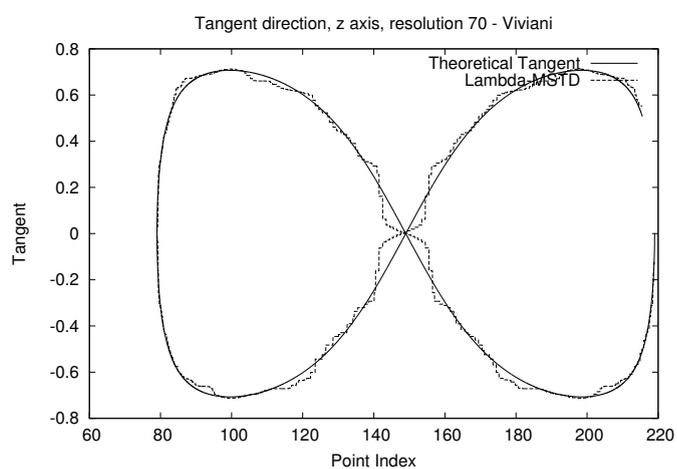
FIGURE 8.9: An expected (theoretical) and estimated tangent direction along 3D helix at fixed resolution, for three axes x , y and z , respectively.



(a)

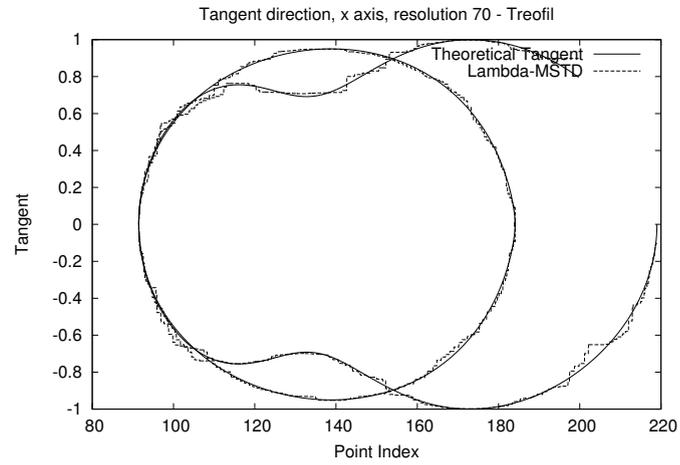


(b)

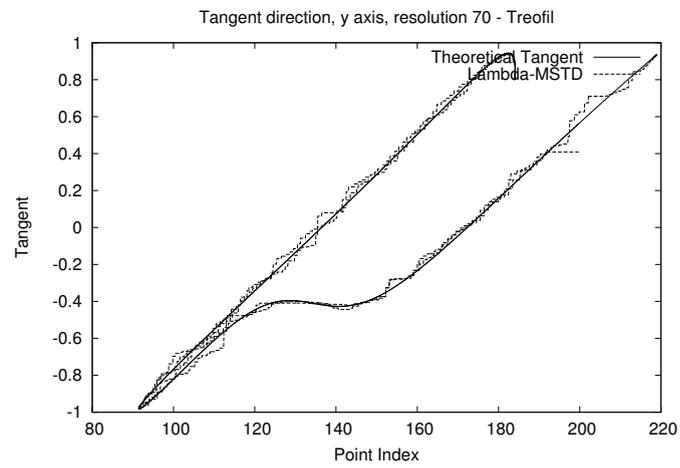


(c)

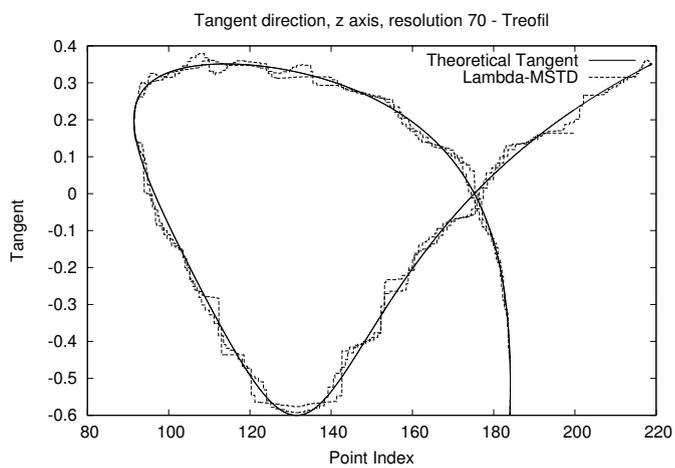
FIGURE 8.10: An expected (theoretical) and estimated tangent direction along 3D Viviani's curve at fixed resolution, for three axes x , y and z , respectively.



(a)



(b)



(c)

FIGURE 8.11: An expected (theoretical) and estimated tangent direction along 3D Trefoil's Knot curve at fixed resolution, for three axes x , y and z , respectively.

Part III

Verification and Application

Chapter 9

Quantitative Analysis of The Human Airway Trees

9.1 Introduction

Airway tree is a part of the lung and it is one of the most important organ of the human body (see Fig. 9.1). The function of airway trees is to transport inhaled air to the alveolus, where exchange between oxygen from air and carbon dioxide from blood takes place. Many diseases can disturb functionality of the airways. Therefore is very important to have a functional understanding of pulmonary anatomy and impact of respiratory disease like emphysema, cystic fibrosis, asthma or chronic obstructive pulmonary disease (COPD) which is common name for pathological changes characterized by airflow limitation due to different combinations of airways disease, and many more.

In the developing process of new methods for diagnosing and treatment of pulmonary diseases an important problem is to find sensitive and reliable methods which permit to detect abnormalities in regional lung structure and function. The previously proposed and still used methods are not sufficient due to two main reasons. First reason is connected to weak sensitivity of the existing methods such as global pulmonary function test or Anterior-Posterior chest roentgenogram. Modern medical computed tomography (CT) can overcome this problem. It uses multi-detector spiral scanners and can produce three-dimensional volumetric images of very high quality, and simply allows one to non-invasively look into inside of a human body. This is a very powerful and useful technique, being used in a variety of medical applications. 3D volumetric scans of a human organ provide an excellent basis for quantification of anatomical structures, for example airway trees. The second problem is caused by human's inability to repeatedly evaluate the same region of the lungs time after time and perform measurements which are accurate and reliable. Moreover, manual processing of data sets acquired using computed tomography is very time consuming and not free of human errors.

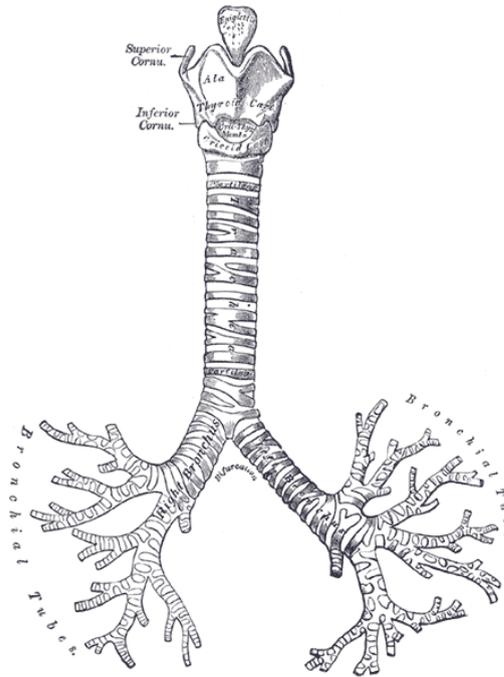


FIGURE 9.1: The human airway tree. (Gray, 1918)

Automatic quantitative description of an airway tree extracted from volumetric CT data set can be a useful information supporting the non-invasive diagnosis of bronchial tree pathologies, especially COPD which is among the leading causes of death in the world and tending to be even more widespread in the future (Murray and Lopez, 1997; Pauwels and Rabe, 2004) and asthma - one of the most widespread disease in the world (Bousquet et al., 2000).

Computer analysis of bronchial tree will allow a doctor to obtain precise data on the process called airway remodeling, which opens new possibilities such as: early identification of pathological changes, precise treatment control, diagnosis of the airway remodeling reason, development of new drugs, etc. Therefore, the goal of the research about automatic quantitative measurements of the pulmonary structure is to build a system for automatic measurements of diameter of an airway lumen and thickness of an airway wall. The system should be suitable for use in everyday clinical routine and due to application of modern effective image processing and analysis algorithms it should provide results of high accuracy.

In the literature there exist several work connected to this subject either in computer science field or in medicine where the study about design and possibilities of such system are reported. In this thesis we focus more on the subject from the computer science point of view. Interested reader can follow for example Matsuoka et al. (2005); King et al. (2000) to find more detailed discussion from the medical point of view.

In the following subsections, the design of such system, goals and advance in research of the quantitative analysis of the human airway trees analysis process are presented in more details.

9.2 System Design

Following [Palagyi et al. \(2006\)](#) the system for the quantitative analysis of the bronchial trees should consist of four main steps: tree identification, tree simplification, tree decomposition and finally quantitative measurements (see [Fig. 9.2](#)). Such steps can be later divided into several sub-steps according to particular application or used algorithms. For example, after skeleton generation step (see [subsection 9.2.2](#)) may be necessary to add automatic skeleton pruning step to correct skeletonization results. On the other hand, pruning can be avoided if proper skeletonization algorithm was used (see results in [chapter 12](#)) or we can manually select interesting part of the skeleton.

The quality and accuracy of the final measurements and in consequence usefulness of the system strongly depends on the used algorithms to solve many different problems. Therefore, each step in the whole process is very important and needs different sets of algorithms. Moreover, algorithms at each step should be as much insensitive as possible to errors in the input data, otherwise errors occurring in the previous steps can propagate and can have huge negative impact of the final measurements. Another important criterium connected to effectiveness of the system is its time and space complexity. Data sets are relatively large. Typical tomogram consists of 512^3 points which should be processed at each step by many different algorithms. Therefore, algorithms should be fast enough to make measurements in reasonable time.

9.2.1 Airway tree identification

The first step of the system for the quantitative airway trees analysis consist of application of segmentation strategies to extract bronchial tree structures from the CT datasets and produce three-dimensional representation of the tree. This task should be done automatically, since complexity of the tree makes manual segmentation tedious and extremely time-consuming (manual segmentation of the airway tree may require several hours of analysis ([Sonka et al., 1996](#))). Airway tree segmentation in CT images is a challenging problem due to the complex anatomy and the limitations in image quality.

Airway tree segmentation algorithms operate on a CT chest image represented by a large 3D array of points with associated values, which represents attenuation coefficient of a small portion of the tested object and are measured with Hounsfield units (HU). In well calibrated CT images, points which represent the interior of an airway tree should be at approximately -1000 HU (air) surrounded by walls which points having relatively high

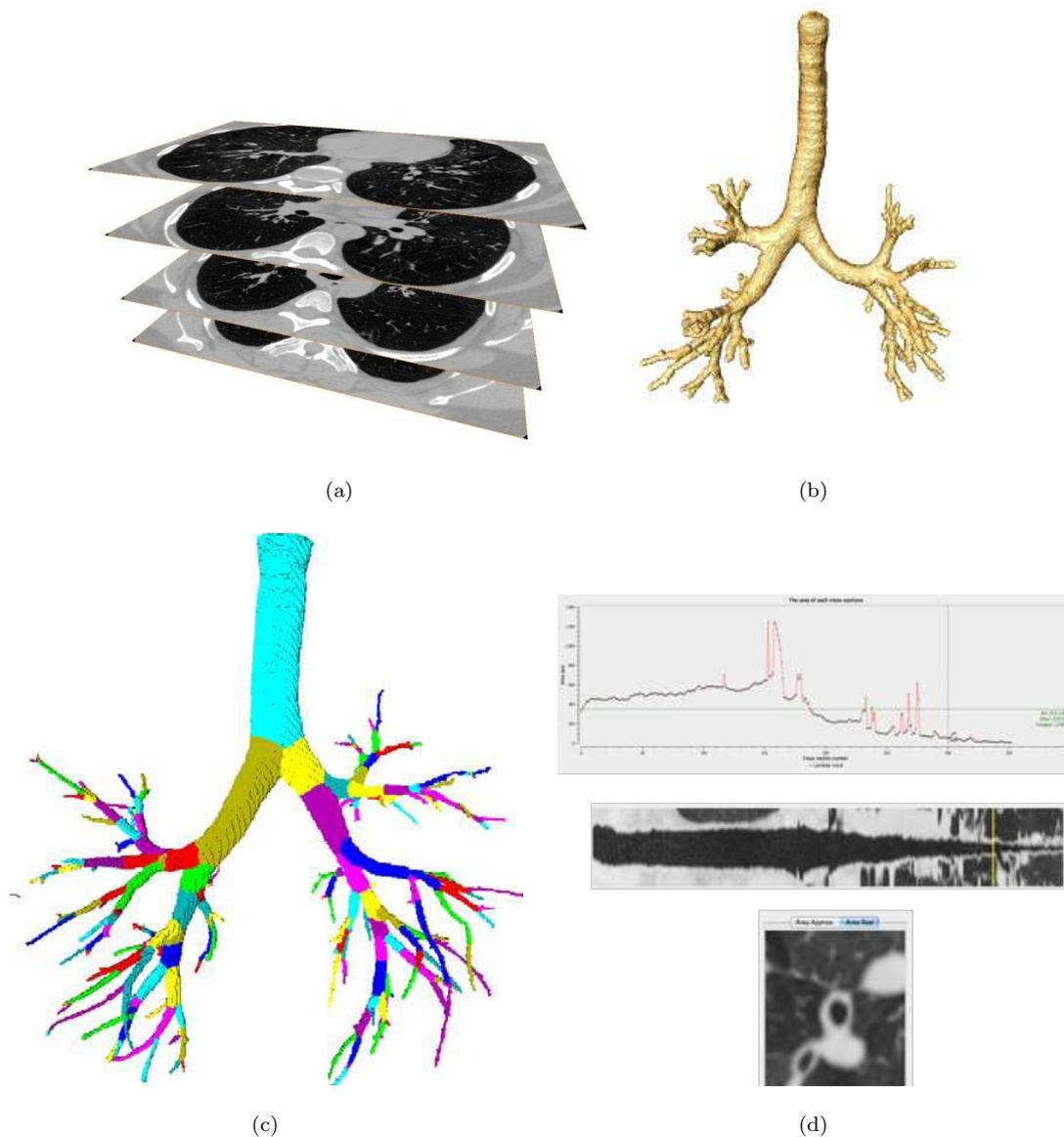


FIGURE 9.2: Consecutive steps of the system for the quantitative analysis of the human bronchial trees. (a): A CT dataset; (b): segmented airway tree from (a); (c): Airway tree decomposed into set of segments (branches); (d) The quantitative measurements of the bronchial properties at the point of interest.

value at approximately -100 to 200HU (soft tissues). Unfortunately, this situation is very rare in real applications. Noise, image reconstruction artifacts, movement artifacts (heart beat), non-standard patient anatomy, airway obstruction or partial volume effect (PVE) significantly decrease the difference between HU for bronchial wall points and points which represent surrounding air. Therefore, values of airway wall points at different bronchial parts can present different intensity values and can be similar to values of interior points of a tree, in particular for high order branches, that is, smallest ones. In addition, different reconstruction kernels (smooth kernels) can increase this effect. As a result small holes in a wall structure appear and high order segments of a tree disappear

which cause leakages of segmentation algorithms in surrounding parenchyma parts of a lung (see Fig. 9.3).

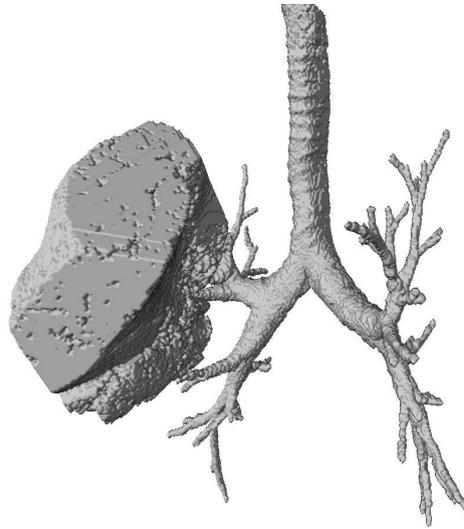


FIGURE 9.3: Example of a severe segmentation leak. Result of segmentation with the standard region growing algorithm.

Leakage is a main problem in segmentation of airway tree. A lung has a texture very similar to the small airways, which leads to a failure of simple segmentation algorithms like region growing (Mori and et al., 1995) or once based on global thresholding. Therefore, the problem is actively studied and many different algorithms were developed.

Previous work on airway segmentation can be divided into five groups: region growing based methods Chiplunkar et al. (1997)Tozaki et al. (1998)Law and Heng (2000), mathematical morphology based methods Pisupati et al. (1996)Preteux et al. (1998), and combination of the two Bilgen (2000), rule based methods Park et al. (1998) and energy function minimization Fetita and Preteux (2002).

Methods based on region growing are usually considered as the simplest and the fastest ones. However, due to the problem of leakage, the user has to adjust the algorithm parameters manually for each image separately, to use such method effectively. However, manual parameters adjustment is impractical and not reliable because it is very hard or, in some cases, even impossible to find suitable parameters.

9.2.2 Airway tree simplification

This step consist of generation of the simple representation of the tree which can be used as descriptor of the tree geometry (see Fig. 9.4). In practice the skeletonization (see chapter 3) or simple centerline extraction methods (Pisupati et al., 1995; Wood et al., 1995) were used to generate such representation automatically (Palagyi et al., 2006).

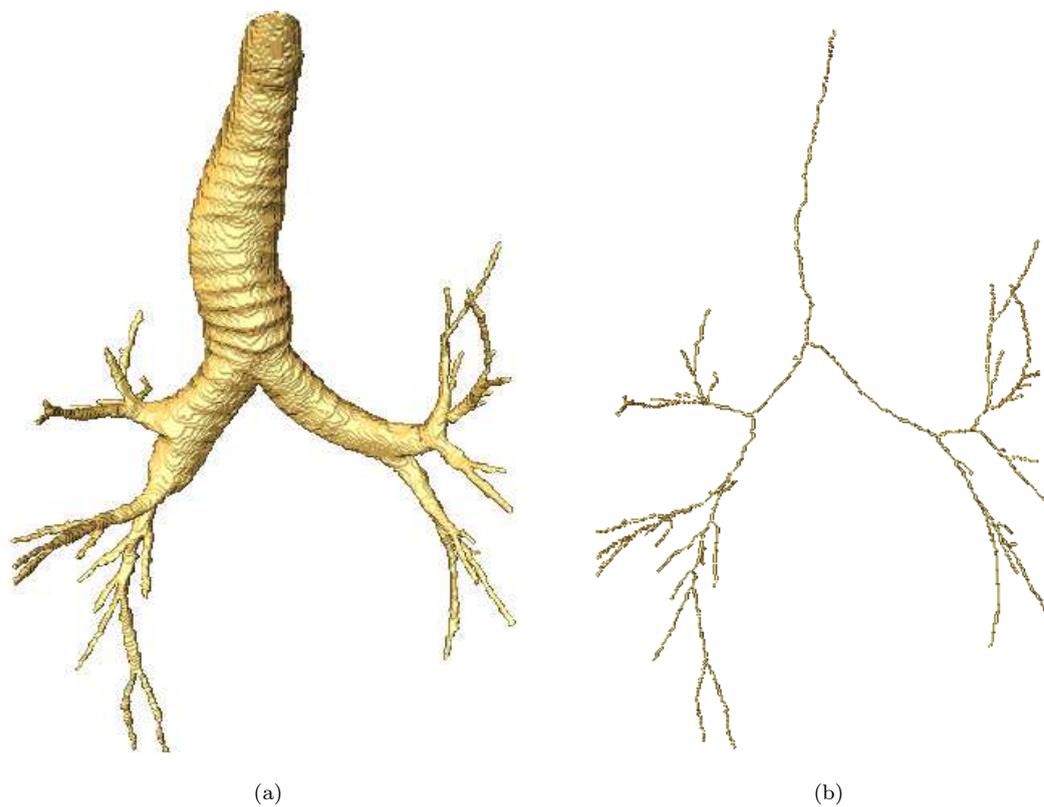


FIGURE 9.4: Example of the skeleton on (b) of the human airway tree on (a).

Then, the skeleton of the tree can be used as a tool to solve several different problems. It can be used for generating formal tree structure as presented in the next subsection. It is also used as descriptor of the airway tree center and utilized in the process of generating cross section of the tree (section 9.2.4). Based on the skeleton one can quantify some of geometrical properties of the tree, for example length or angle of branches (Pisupati et al., 1995; Wood et al., 1995). Another interesting application of the skeleton is a error detection during the segmentation process, for example, it can be used to detect leakage (Tschirren et al., 2005a), missed or spurious branches and anatomically implausible loops (Yu et al., 2007).

The bronchial tree is a tubular structure presenting the complex branching topology. In addition the segmented tree used as an input for the skeletonization can be corrupted by segmentation errors. For example, geometrical deformation (irregular surface of the tree) or topological errors like for instance presence of cavities or tunnels in the segmented tree. Since, the skeleton correctness is of paramount importance, the skeletonization of the bronchial tree is a very challenging task.

Several methods were already proposed to generate skeleton of the segmented tree. Most of them use one of the topological thinning methods (Palagyi et al., 2006; Gerig et al., 1993; Mori et al., 1996; Wan et al., 2000; Mori et al., 2000; Toriwaki and Mori, 2001a; Chen and Molloy, 2003) or distance transform based approach (Nystrom, 2003; Maddah

et al., 2003). However, methods based on Voronoi diagrams (Antiga et al., 2003) or general differential geometry (Krissian et al., 2000; Aylward and Bullitt, 2002) were also proposed. Based on our knowledge, no perfect skeletonization technique were successfully applied to skeletonization of the bronchial tree yet (in chapter 12 we will show that recently proposed skeletonization methods can be effectively used in this particular task). Thus in the most cases the generated skeleton need to be pruned (Shaked and Bruckstein, 1998; Svensson and di Baja, 2003) before the skeleton can be used. For example, the pruning methods can rely on mathematical morphology (Gonzales and Woods, 1992), geometry of branches (i.e. length), (Wan et al., 2000) or the position inside the tree (i.e. distance to surface function) (Palagyi et al., 2006).

To simplify the skeletonization problem the pre-processing steps, for instance, cavity filling, tunnel closing or morphological closing can also be useful. However, such tools have to be used with careful, since, they can change the geometry or the topology of the tree and they can be source of error in the further steps.

9.2.3 Generation of the formal tree structure

Next step of the system consist of application of methods which generate the formal tree structure of the segmented tree based on the generated skeleton. Note that the skeleton can be treated as a graph, therefore it can be straightforwardly used for this purpose. However, such formal structure can be represented in several different forms, depending on application. For example, based on generated skeleton we can simply divide the tree into set of segments and branch-points, that is, points where segments are dividing (see Fig. 9.5(a)). If we go further we can partition each point of the airway tree according to the corresponding branch (see Fig. 9.5(b)). The automated partitioning procedure were proposed, for example, in Palagyi et al. (2006). Such simple partitioning allow to perform another set of quantitative measurement, like for instance, branch surface area or branch volume evaluation.

Another useful and interesting possibility is to assign anatomical names to the segments and branch-points of the human airway tree. According to Tschirren et al. (2005b), the first six generations, roughly, of the human airway tree exhibit a relatively similar topology across subjects, and anatomical names exists for 31 segments and 42 sub-segments (Boyden, 1955) (see Fig. 9.6). Labeling an airway tree simplify the navigation and allow matching of branch-points across subjects.

In the past these task were often performed manually (Wood et al., 1995) due to the lack of automated algorithms that can tolerate false branches and anatomical variability typical for in vivo trees. The algorithms for automatic labeling exists, however, theirs number are very limited. Mori et al. (2000) presented a knowledge-based labeling algorithm which can be only applied to incomplete tree (about 30 branches per tree). Kitaoka

9.2.4 Quantitative analysis

The goal of the final step of the process is to measure the quantitative parameters of the selected branches or whole tree in general. As reported in the literature of the subject, the most interesting properties are local diameter of the airway lumen and the wall thickness which can be used as a basis for other parameters evaluations, for instance lumen area or wall area to lumen area ratio, and many more.

Based on the formal representation of the tree, from the previous step, we can simply find the interesting part of the airway in which we will perform measurements. Then, according to [Tschorren et al. \(2005a\)](#) the quantification process is, divided into the following three steps: Re-sample 2D slices perpendicular to airway segment. For each 2D slice separately, segment airway wall. Finally conduct measurements on segmentation results.

The resampled slices, in the first step in the process, need to be oriented perpendicular to the skeleton of the respective airway segment. One slice can be re-sampled for every skeleton voxel position. The perpendicular orientation is determined by computing the tangent to the skeleton. Re-sampling can be performed using, for example trilinear interpolation.

In the second step, the segmentation of the bronchial wall is performed independently on each cross sections image. According to [Saragaglia et al. \(2006\)](#), methods for airway wall segmentation can be roughly classified into three categories: full-width at half-maximum (FWHM) ([Matsuoka et al., 2005](#)), pattern-based ([Chabat et al., 2001](#); [King et al., 2000](#); [Wiemker et al., 2004](#)) and shape-independent approaches ([Saragaglia et al., 2006](#)). The method based on FWHM approach use the gray-level profile of rays cast from center of the lumen and assumption that maximum intensity along such profile is in the middle of the wall. However, this method can produce poor results for small airways or airways with thin walls ([Hoffman et al., 2003](#)). The second group of method assume that the cross section of the airway should be circular or ellipsoidal and try to detect such objects by matching predefined patterns. However, the cross sections of airways are normally not perfectly circular, therefore this method can not guarantee accurate results. The last group of methods try to detect wall location by using for example a contour matching technique, where two antagonist forces (external vs. internal) control the contour deformation.

The accurate segmentation of the bronchial walls allow quantitative measurements in the last step of the process. However, the quality of the final results depends on every single step in the whole process. Therefore, the problem of selecting the best set of algorithms is very complex. In the following chapters we present algorithms which relay on discrete geometry and topology and can be used with success at each step of the system for quantitative analysis of the human airway trees. We will start from the problem of airway

tree segmentation and we will finish at the problem of accurate cross section generation. However, this work does not cover the problem of wall segmentation.

Chapter 10

Three dimensional model of The Human Airway Trees

In recent years several researchers proposed some models of bronchial tree which vary in accuracy. [Weibel \(1963\)](#) and [Horsfield et al. \(1971\)](#) proposed structural models of airway that include airway dimensions and connectivity. However, these models do not include information about the spatial arrangement of the airway structure, and hence they are limited to modeling lung function in one dimension. Although there are several geometric airway models such as those proposed by [Nelson and Manchester \(1988\)](#) and [Martonen et al. \(1994\)](#), these models are still limited to 2D.

The rapid development of three-dimensional technology and demand from the medical community for 3D models of bronchial tree led to the construction of a number of three-dimensional models of the bronchial tree. The first 3D models were constructed in 80s of the XX century ([Chen et al., 1980](#)). One of the first 3D models presented by [Kitaoka et al. \(1999\)](#) is based on 9th main expert rules and 4th additional rules which deal with exceptional cases. The later model developed by [Tawhai et al. \(2000\)](#) uses a Monte Carlo method for growth of bifurcating systems in 3D space in response to host geometry. [Gillis and Lutchen \(1999\)](#) based on their 3D model have predicted images of ventilation distribution in asthmatic patients with potential clinical impacts. Other models e.g. ([van Ertbruggen et al., 2005](#)) have been used to predict aerosol deposition in the human respiratory track which allows to plan inhaled aerosol therapy. The models automatically generated by the algorithm consist of connected cylinders of different length and radius and have been constructed to study the structure-function relationship in the lung or to simulate flow and particle transport. Such models do not represent bronchial trees extracted from 3D CT datasets of a chest whose branches reveal noised and geometrically deformed tubes. Therefore the models are insufficient for testing algorithms for quantitative analysis of segmented trees.

On the other hand there is strong need for 3D models of CT trees. Algorithms for measurement of local lumen of bronchial trees have to deal with deformations and noise in CT images. However, it is difficult to test the algorithms on CT images because correct values of local lumen are unknown. Therefore, some researchers (Palagyi et al., 2006) have built simple phantom which consists of set of plastic tubes of constant diameter and smooth boundary surface. Then they have made CT of them and have compared quantitative results given by computer algorithms with known, correct values. Another strategy, presented in this chapter, might be to automatically generate digital models (phantoms), with the use of algorithm, which take into account noise and geometric deformations which occur in trees segmented from CT datasets. Correct quantitative parameters of such phantoms are known and can be compared with results of the algorithms being tested. Therefore, in this thesis we propose an extension of the model Kitaoka et al. (1999), by adding noise and geometric deformations to it. We developed the algorithm which results in volumetric model of airway trees useful for testing algorithms for quantitative analysis of the airways trees.

The model consists of two volumetric images: the first represents a tree map without noise and with geometric deformations, and the second contains a tree with noise and geometric deformations of branches. The proposed extension was developed on the following assumptions:

- *Constant diameter of a branch:* the analysis of a branch diameter allows us to obtain useful information about the tree being examined. Therefore it is very important to deform a branch in such a way that its diameter becomes constant. In this way correct value of a branch diameter is known which is necessary for testing of algorithms for quantitative analysis.
- *Noise and geometric deformations:* data obtained from CT images very often contain different types of noise. Therefore, the presented model simulates these disruptions. The issue is extensively explained in section 10.2.3.
- *Efficiency in use:* The generated bronchial tree model is planned to be used to test algorithms which measure local diameter of any branch of the generated tree. Therefore it is important to have fast access to correct local diameter for comparison. So, the value of local diameter is kept in each voxel of generated tree.

10.1 Basic Model of Bronchial Trees

This section presents short description of an algorithm proposed by Kitaoka et al. (1999) called basic algorithm, for automatic 3D bronchial tree generation. Interested reader can find detailed explanation in (Kitaoka et al., 1999).

Airways have a tree structure. The root of the tree correspond to the trachea. The trachea divides into two main bronchi: left and right. The main bronchi are further divided into next branches. Geometry of the bronchial tree is closely related to the spatial arrangement of the tree. Each branch is assigned to a particular part of a lung volume supplied in air by the branch and its descendants (in the tree hierarchy). This allows to correlate volume of air delivered through a branch with its spatial arrangement in a lung. Therefore, a branch size and direction is determined by size and geometry of the corresponding volume. More details on the rules of lung volume division are presented below.

The algorithm is based on three assumptions which make possible to realize regular and effective air transportation inside whole lungs.

- The first assumption declares that each branch is a circular, rigid tube with a constant diameter.
- The second assumption allows for correlation of the air transported through bronchi with its spatial position based on the following statement: the whole volume of air flowing through a branch is proportional to volume of a lung region that is supplied by this air. Furthermore, assume that for each bifurcation, children volume is proportional to the volume of its parent, which in turn means that the total air flow transported by the children is proportional to air flow transported by its parent.
- The third assumption says that the final branches, which correspond to final bronchioles in the human respiratory tree, are homogeneously arranged within the organ.

The above assumptions allow to define expert rules, described below, which are fundamental for the construction of the basic algorithm. Successive execution of these rules allows to generate model of the bronchial tree. The rules and the assumptions presented above define branching geometry and are based on earlier morphometric studies and flow rate analysis in tubular living organs e.g. [Weibel \(1963\)](#); [Horsfield et al. \(1971\)](#). [Figure 10.1](#) shows the basic parameters of a single bifurcation.

The algorithm requires input data that provide a set of parameters describing the root of the tree - trachea, and the surface (called boundary surface) that defines the space in which the tree is generated.

Trachea: the most important parameters of trachea (root) are: diameter, length, and position in space. Moreover the root should be placed in the proper position relative to the above mentioned surface. Parameters of the trachea should be chosen in accordance with the morphological studies.

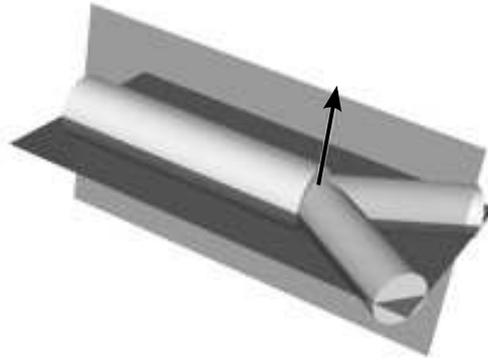


FIGURE 10.1: Single bifurcation. Dark grey color represents bifurcation plane, light grey color represents volume division plane. Both planes are spread out to the borders of the parent region. The normal of the bifurcation plane has been marked by arrow. This normal has been anchored in the bifurcation point.

Volume: it is limited by the bounding surface, which can be defined in several ways. However, to simplify this work and following (Kitaoka et al., 1999), we use a simple surface described by the equation 10.1 and shown in Fig. 10.2

$$z = 2 * 15^{-3}(x^2 + (1.5y)^2)^2 \quad (10.1)$$

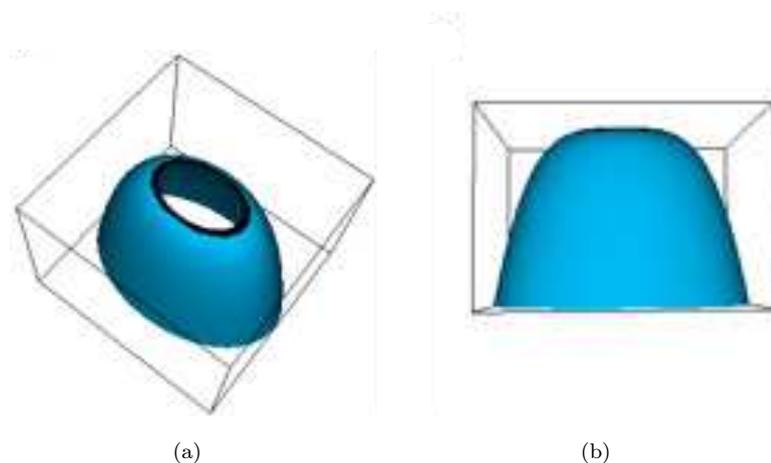


FIGURE 10.2: The bounding surface which limits volume where the tree is generated. (a) View from the top; (b) Side view.

The following rules define the next steps, of an algorithm which successive execution until the end condition defined in rule 9, allows to generate the bronchial tree model.

Rule 1: bifurcation is dichotomous, that is, each parent is divided into two children.

Rule 2: longitudinal sections of a parent and his children are on the same plane, called the plane of the bifurcation (see Fig. 10.1).

Rule 3: sum of the children flow is equal to the total flow of the parent: $d_0^n = d_1^n + d_2^n$, where d_0^n , d_1^n and d_2^n denote the diameter of the parent, first and second child respectively. The value of n is set to 2.8 based on statistical analysis of data from [Raabe et al. \(1976\)](#) and $d_1^n < d_2^n$.

Rule 4: volume, which is supplied by a parent is divided into two children volumes by the volume division plane. This plane is perpendicular to the plane of bifurcation and tends to the borders of the parent volume. Figure 10.3 shows the volume determined under this rule.



FIGURE 10.3: Outer surface of a volume supplied by the left daughter of trachea.

Rule 5: the flow-dividing ratio $r \in (0, 0.5)$ is equal to the volume-dividing ratio, defined as the ratio of the volume of the smaller child to volume of its parent. The algorithm for r calculation is presented in [Kitaoka et al. \(1999\)](#).

Rule 6: diameter of two children and the bifurcation angle between them are defined as a function of r and parent diameter ([Kitaoka et al., 1999](#)).

Rule 7: length of each branch is three times its diameter.

Rule 8: continuing to generate the new branches in a given direction causes the childrens become new parents and their bifurcation plane is perpendicular to the bifurcation plane of their parent.

Rule 9: The process of generating new branches in a given direction continues until the flow rate for a new branch is less than minimal a-priori defined flow rate or when the branch goes beyond defined region.

First two rules are based on observation of real lungs. The third rule is based on the modeling and analysis of inspiratory flow in bronchial trees. The fourth rule corresponds to the second assumption presented in this section. The sixth rule express the optimal

relationships between flow rate, diameter and branching angles presented earlier (Weibel, 1963; Horsfield et al., 1971). Rule seventh is a result of examination of data presented in Raabe et al. (1976) which led to the conclusion that the distribution of the length-to-diameter ratio is Gaussian-like distribution with a mean of 2.8 and standard deviation of 1.0. The additional four rules work in exceptional cases and are presented in details in Kitaoka et al. (1999).

The algorithm starts by determining the bounding space which defines the volume in which the bronchial tree is generated. Then it generates the root of the tree - trachea. The root and all branches are represented by cylinders. The use of a cylinder allows to obtain a uniform diameter of each branch, which was one of the crucial objectives of the implementation. However, there is a problem with cylindrical representation of branches. The connection of two cylinders of different radii generates unexpected sharp edges which do not occur in airways trees. The problem can be solved by the use of smoothing filter (see section 10.2.3), which smoothes the input object while preserving its topology. According to the rule 9 the process of a tree generation in a given direction is terminated when the generated branch exceeds the boundaries of its region.

10.2 Extended Algorithm of Bronchial Tree Modeling

This section presents algorithm for bronchial tree modeling which is an extension of the basic algorithm. First, the extended algorithm generates basic model with the use of basic algorithm, presented in the previous section. Then the branches generated in the first step are bent to obtain more "realistic" model. It returns triangulation of outer surface of the tree. Then, on the third step, the triangulated representation of the tree is transformed to voxel space. The fourth - last step consists in adding noise and smoothing to obtain a model which is more similar to real airways trees.

10.2.1 Geometric deformations of surface model

In the section we present branch bending procedure. The procedure makes the generated branches more similar to real segmented branches from CT images. The bending procedure generates a bended branch in two steps. First it draws a spiral with especially selected parameters and then the spiral is used to draw a "pipe". Drawing the spiral is realized according to equation 10.2 where f is a randomly selected function from the following trigonometric functions: \sin , $-\sin$, \cos , $-\cos$.

$$z_k = r f\left(2\pi \frac{k}{n-1}\right), x_k = r f\left(2\pi \frac{k}{n-1}\right), y_k = h \frac{k}{n} \quad (10.2)$$

Where, r is a radius of rotation. In our work we experimentally set to a branch radius divided by 3.5. The values n, h defines number of voxel forming the spiral and length of the spiral, respectively. The value k is an index of point in the spiral.

An example of the effect of bending procedure is shown in Figure 10.4.

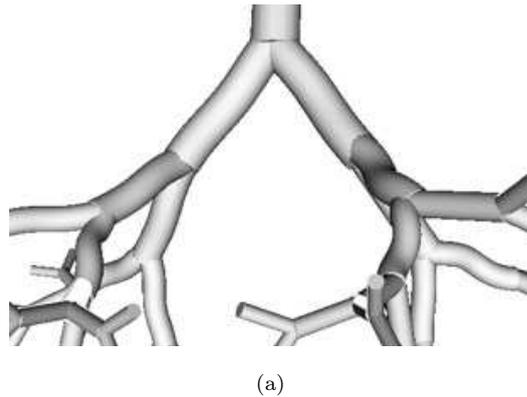


FIGURE 10.4: Part of the model with bended branches.

10.2.2 Transformations of the surface model to volumetric model

The bending procedure described in the previous section generates bronchial tree model in the form of triangulation of its outer surface. The procedure works in continuous space because geometrical deformations are easy to perform in the space. The procedure of noise generation for a model of bronchial tree works in discrete framework. Therefore after application of bending procedure the model is transformed to discrete representation. Conversion of the surface model into the volumetric model can be implemented in several ways. In the project only methods which fill inside of the converted object can be applied.

10.2.3 Distortions introduced to in a volumetric model of bronchial tree

Distortions in volumetric space are generated in two steps. The first step consists in addition and subtraction voxels from object with the use of modified EDEN algorithm (Eden, 1961) in the sequel called TopoEDEN. The very important feature of TopoEDEN algorithm is that it preserves topology of a modified object which can be guarantee by modification only simple points. In the case of bronchial trees the guarantee of topology preserving is very important because it ensures that during modifications of the tree any two branches do not merge or no one branch can be clogged. The TopoEDEN procedure can erode or dilate the input tree. In both cases the procedure makes a list of border voxels of the object or background. Then it randomizes a voxel from the list and if it is simple modifies it. There is also possible to make erosion and dilation alternately in TopoEDEN procedure.

Exemplary results that can be obtained with the use of TopoEDEN procedure are shown in figure 10.5.

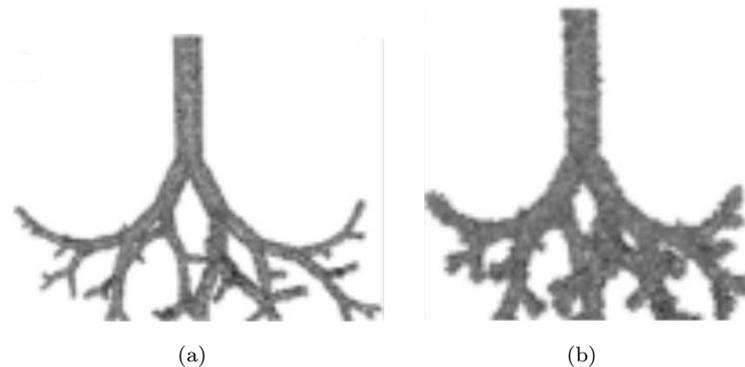


FIGURE 10.5: Exemplary results of TopoEDEN when applied to a bronchial tree; a): TopoEDEN which erodes input object; b): TopoEDEN which dilate input object.

Examples in figure 10.5 show that results of TopoEDEN are not rewarding in terms of similarity to real segmented bronchial trees from CT images. For that reason, at the next stage of processing the authors use iterative smoothing algorithm ASFT (Alternate Sequential Filter controlled by Topology) (Couprie and Bertrand, 2004). In this method smoothing is obtained by morphological open-close operation (Serra, 1982) with the use of sphere of variable radius. ASFT applied to result of TopoEDEN gives more realistic view of a tree surface. Exemplary effect is shown in figure 10.6.



FIGURE 10.6: The effect obtained by using TopoEDEN. The tree is dilated and smoothed by ASFT.

Finally, the algorithm which generates the extended model can be presented in the following several steps:

1. Generating of trachea and organ restricted area.
2. Generating of a tree branches based on basic algorithm rules.
3. Realization of individual tree branches bending procedure.
4. Conversion of the tree to the discrete framework.
5. Realization of TopoEDEN.
6. Realization of ASFT smoothing.

Figure 10.7 compares basic model with extended models. One can see that extended model is far more similar to real tree than basic model.

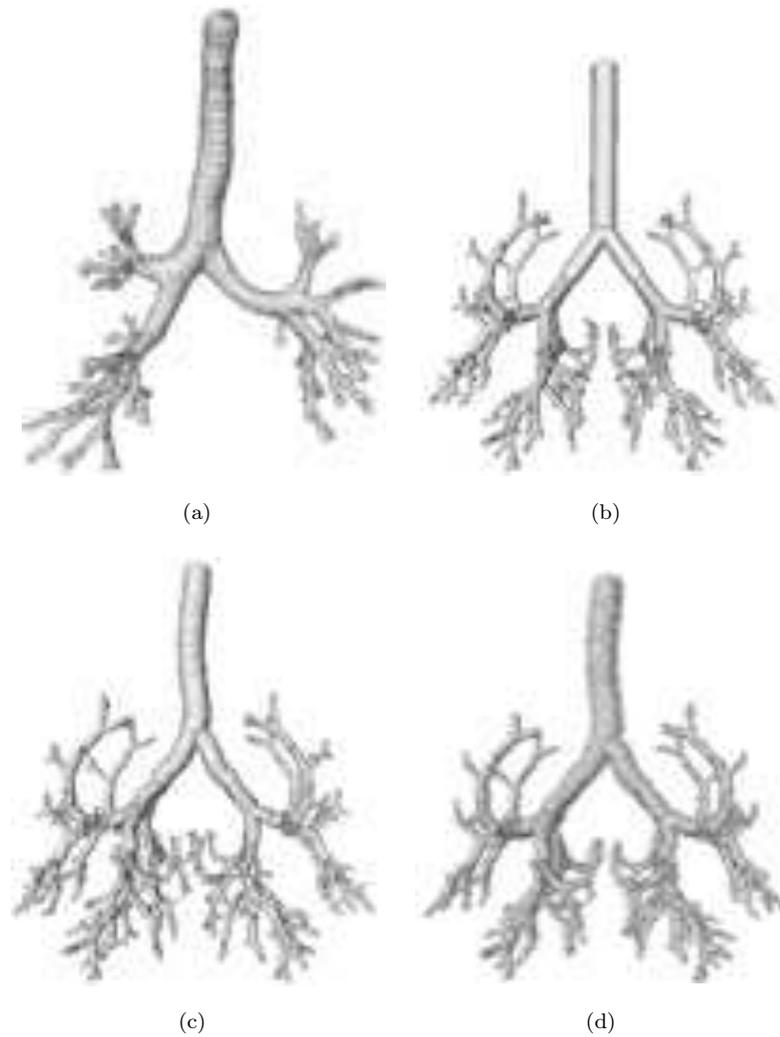


FIGURE 10.7: Bronchial trees phantoms generated with different models; a): bronchial tree obtained by segmentation of tomographic image; b): tree generated with the use of basic model; c): extended model with bended branches; d): extended model with bended branches and noise.

Chapter 11

Robust Segmentation of The Human Airway Trees

Previously published methods (see chapter 9.2.1) focus on how to detect and eliminate leakages when they occur or how to avoid them using complex rules. Some of these algorithms must be run several times with different parameters, another ones analyze very large sets of points using complicated nonlinear filters or semantic rules.

In this thesis we propose a new segmentation algorithm based on 3D tunnel closing in bronchial walls. The presented method eliminates the leakage problem by closing all tunnels in an airway tree wall and than performs the standard region growing algorithm. Thanks to the 3D tunnel closing algorithm (see chapter 5) our method is simple, reliable and is based on the basis of discrete topology. Moreover, the method is fast and fully automated. In the following section we present the method in details and we show experimental results of application of the method to the real CT datasets.

11.1 Airway Tree Segmentation Based on Tunnel Closing

11.1.1 Histogram analysis and preliminary wall extraction

The first, important step in our segmentation method is airway wall extraction (see Fig. 11.1). After histogram analysis of CT scans we can distinguish three intensity ranges with are of great importance from the bronchial tree segmentation point of view (see Fig. 11.2). The first one represents air voxels, the second one corresponds to voxels of internal border of bronchial walls and the last one represents soft tissue and blood voxels. Bronchial walls belongs to soft tissue range. However, differences in wall pixel intensities and wall thickness, at low and high level of an airway tree and common occurrence of other soft tissues, make very difficult to extract walls directly from borders of the range

as threshold parameters. Fortunately, in our application, we do not need to segment walls at this stage, instead the algorithm extracts only internal border of bronchial walls which is enough to perform a 3D tunnel closing procedure. The lower threshold value for this purpose is approximately situated on the border between air voxels range and internal border of bronchial walls. The higher threshold value is approximately situated between the range which represents internal border of bronchial walls and soft tissue range. Using this values, which can be easily and automatically selected, the algorithm can extract internal border of the walls on different levels of an airway tree. Moreover, small number of points in this range leads to a "clear" output image (without unnecessary soft tissues).

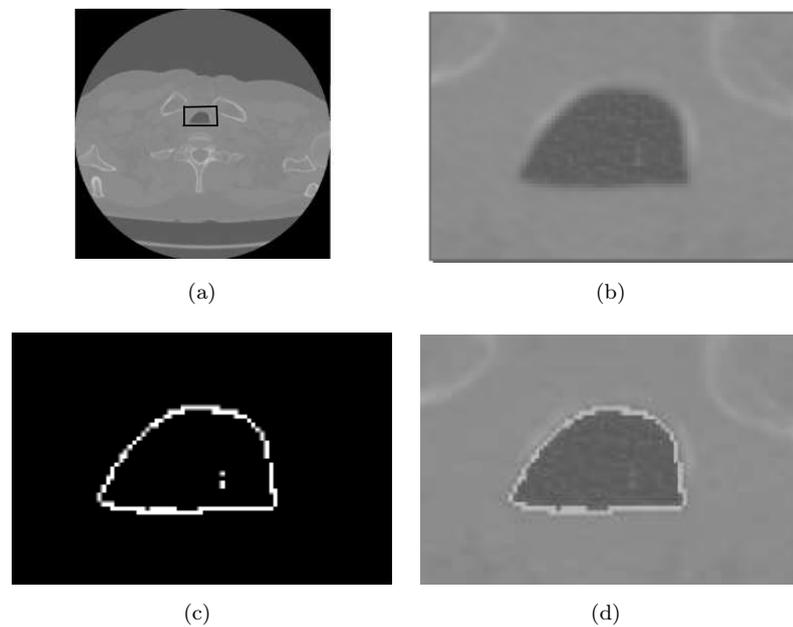


FIGURE 11.1: Airway tree wall extraction. (a): One 2D slice obtained from a 3D CT dataset; (b): Zoomed fragment of the trachea; (c): Extracted airway walls; (d): An input image (b) merged with the image (c).

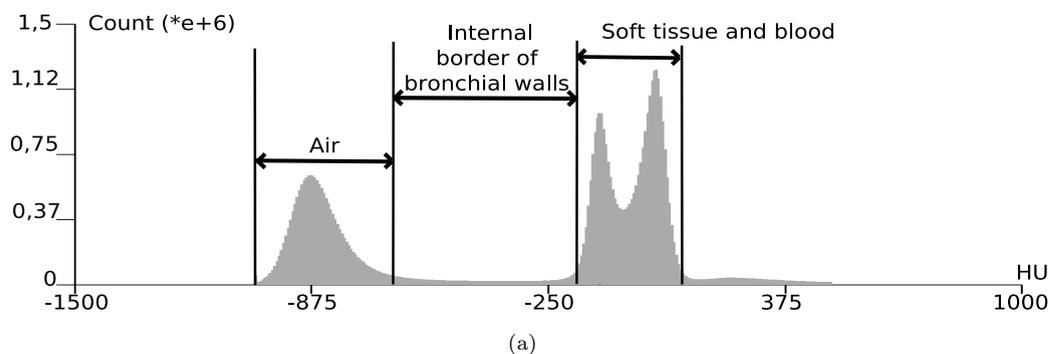


FIGURE 11.2: A typical histogram generated from a 3D chest image. Marked ranges correspond to air, internal part of bronchial walls, soft tissue and blood.

11.1.2 3D tunnel closing in airway walls

The main part of the presented segmentation algorithm consists of removing discontinuity in the bronchial walls by locally increase contrast of the walls, to eliminate problem of leakage. On the 3D binary image generated in the previous step (see Fig. 11.3(a)), one can notice that extracted area corresponding to the walls has many "holes" (see Fig. 11.3(b)) or more formally speaking it consist tunnels (see 2.1.5 for the formal definition). Such tunnels are results of the local lack of contrast in the CT dataset or presence of noise.

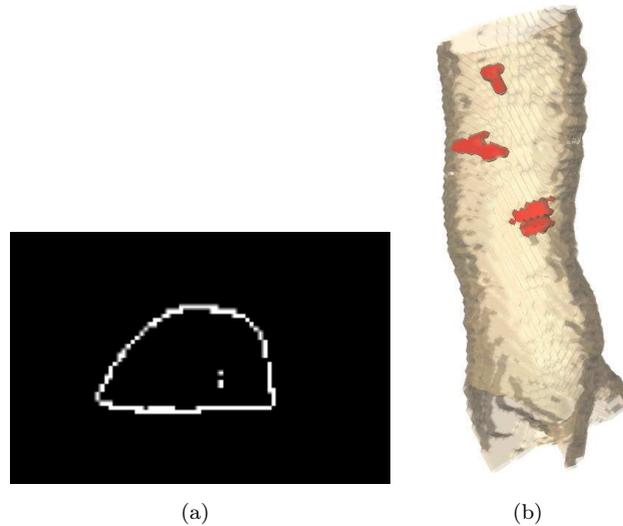


FIGURE 11.3: Tunnel closing in the extracted bronchial walls. (a): The 2D binary image which represent extracted bronchial walls; (b): The 3D reconstruction of (a) with well visible tunnels. After application of the tunnel closing algorithm all tunnels were closed (red patches).

In this step of segmentation algorithm our goal is to remove tunnels. To achieve this goal we use binary image generated in the previous step, which represents extracted walls, as an input for the tree-dimensional tunnel closing procedure. We use the tunnel closing algorithm described in the chapter 5. As result we obtain the 3D binary image where all tunnels were closed (see Fig. 11.3(b)).

11.1.3 Merging and final segmentation

On the last step, the algorithm combines the binary image with closed tunnels B , produced in previous step, with an input CT dataset A . The algorithm sets intensity of a voxel from image A to the maximal possible HU value H_{max} , only if the corresponding voxel from the image B has value 1. Then the standard region growing algorithm (RGA) is applied to produce the final segmentation result. The RGA needs two parameters: the first represents a threshold value which constrains the growth process, the second one

corresponds to a starting point which is called the seed. The first parameter is set to the $H_{max} - 1$ value and the seed can be selected manually or automatically using, for example, a simple method proposed in [Tschirren et al. \(2005a\)](#).

11.2 Segmentation Algorithms Comparison

The proposed method has been applied to test the segmentation of the chest CT images acquired using GE LightSpeed VCT multidetector CT scanner. The set of stack images is of size 512x512 and voxel dimensions are: $x = y = 0.527$ mm, $z = 0.625$ mm. All tests were performed on standard PC platform computer (CPU: INTEL 2Ghz). Fig. 11.4 shows the comparison between the proposed method and standard region growing approach on the same set of images. For the experiments we use 30 different CT datasets with standard quality (low dose scans) and the same number of datasets with improved quality (high dose scans). The CT datasets were provided by Department of Radiology and Diagnostic Imaging, Medical University of Lodz.

The experimental results showed that presented method works well in all tested cases. The leakage problem is fully eliminated and results are of much better quality than for standard region growing approach. The numbers of extracted branches by the proposed method and region growing method with optimal parameters selected manually are presented in table 11.1. It occurs that for all tested cases our algorithm gives better results than region growing approach. Moreover, for the cases which correspond to the CT data with higher quality where the tree is well defined (see for instance two first rows on Fig.11.4) almost 100% of the branches are extracted up to 5th order of a bronchi using the proposed method. The superiority of our approach is also clearly visible for the cases (see for instance two last rows on Fig. 11.4) which correspond to standard quality scans. The presented method extracted 100% branches up to 4th level in all cases, while region growing algorithm extracted only 37% in average or finished with a severe leak (see Fig. 11.4(h)).

The superiority of our approach is not only related to the number of levels extracted during segmentation process. It is also important from the practical point of view to segment the bronchial tree without unnecessary deformation like for example cavities, tunnels, or noise like distortions. Our approach generates bronchial trees without such deformations. On Fig. 11.6 we can see skeletonization results using bronchial trees generated by our method (Fig. 11.6(a)) and region growing approach (Fig. 11.6(b)). One can notice that skeletons are much more complicated and noisy, when region growing algorithm is used for the segmentation which will be source of errors in the further steps of the quantitative analysis process. From the other hand, skeletons generated using our segmentation results as an input are very good quality and can be used for reliable measurements.

The computation time for these two algorithms has been also evaluated. The proposed algorithm is much slower than region growing algorithm but its runtime does not exceed two minutes per volume, and it is faster than previously presented approaches e.g. [Tschirren et al. \(2005a\)](#)[Graham et al. \(2008\)](#). It is worth mentioning that the presented segmentation method is optimal (linear complexity) because it is only based on linear complexity algorithms. Compared to other methods like the one proposed in [Tschirren et al. \(2005a\)](#)[Graham et al. \(2008\)](#), our method can resolves less orders of the bronchi, especially for low dose scans. However, we propose algorithm which is simple, based on well defined notions and reliable in the part of the bronchial tree which is interesting from the medical point of view. The method is fully automatic, there is no need to tune any parameters in general case, However, thanks to histogram analysis it can be simply adapted to the CT data acquired by the different scanner or reconstructed using different reconstruction kernel.

TABLE 11.1: Fraction (in %) of average number of extracted branches at different levels of the airway tree. Results obtained using the proposed method (HC) and region growing method (RGA) with manually selected optimal parameters on the 30 cases with standard quality (low dose) and 30 cases with extended quality (high dose).

	Extended quality		Standard quality	
	HC	RGA	HC	RGA
2nd level[%]	100	100	100	50
3rd level[%]	100	80.6	100	50
4th level[%]	100	50.3	100	37.5
5th level[%]	96.2	32.2	66.3	5.0
6th level[%]	68.7	12.5	40.3	0
7th level[%]	49.5	6.2	23.3	0
Working time[sec]	100	4.5	118	4.2

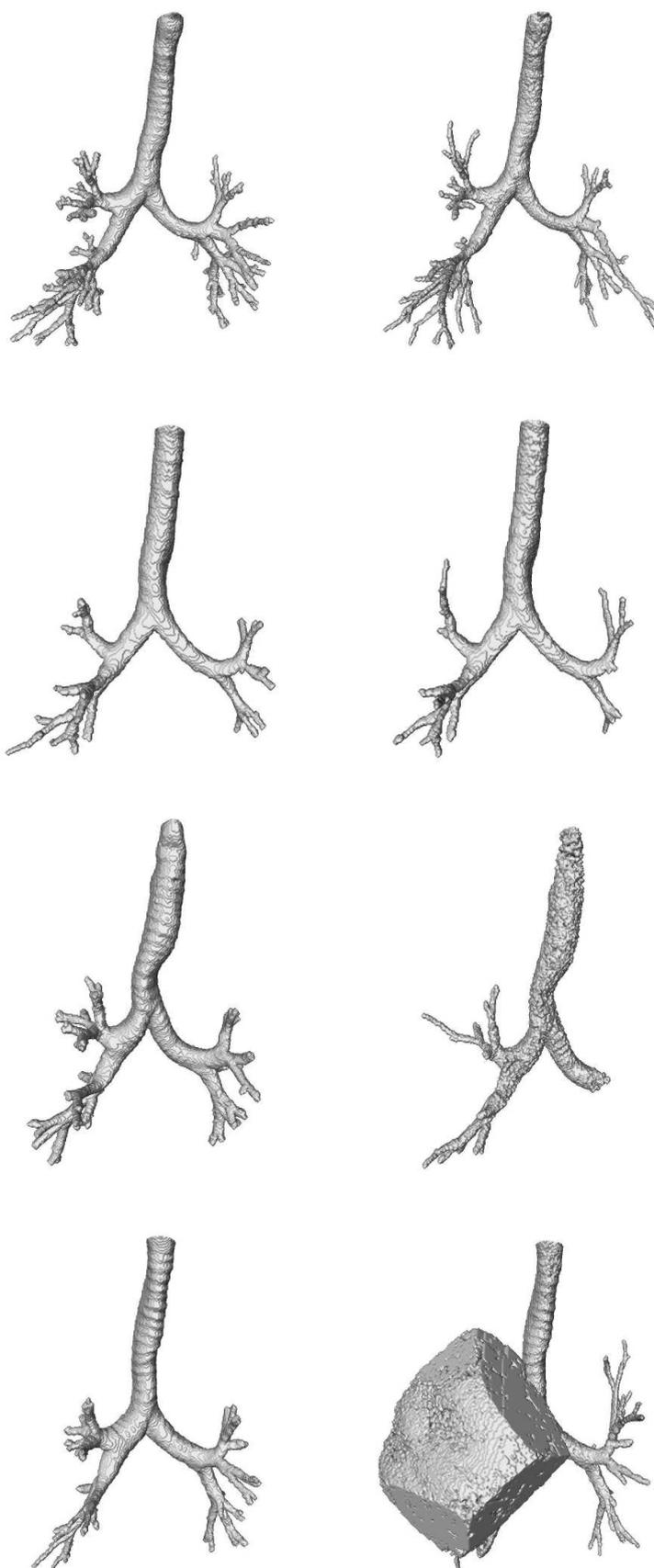


FIGURE 11.4: 3D visualization of segmentation results. Left column - proposed method, right column - the region growing procedure (Mori and et al., 1995). In each row the same CT dataset were used.

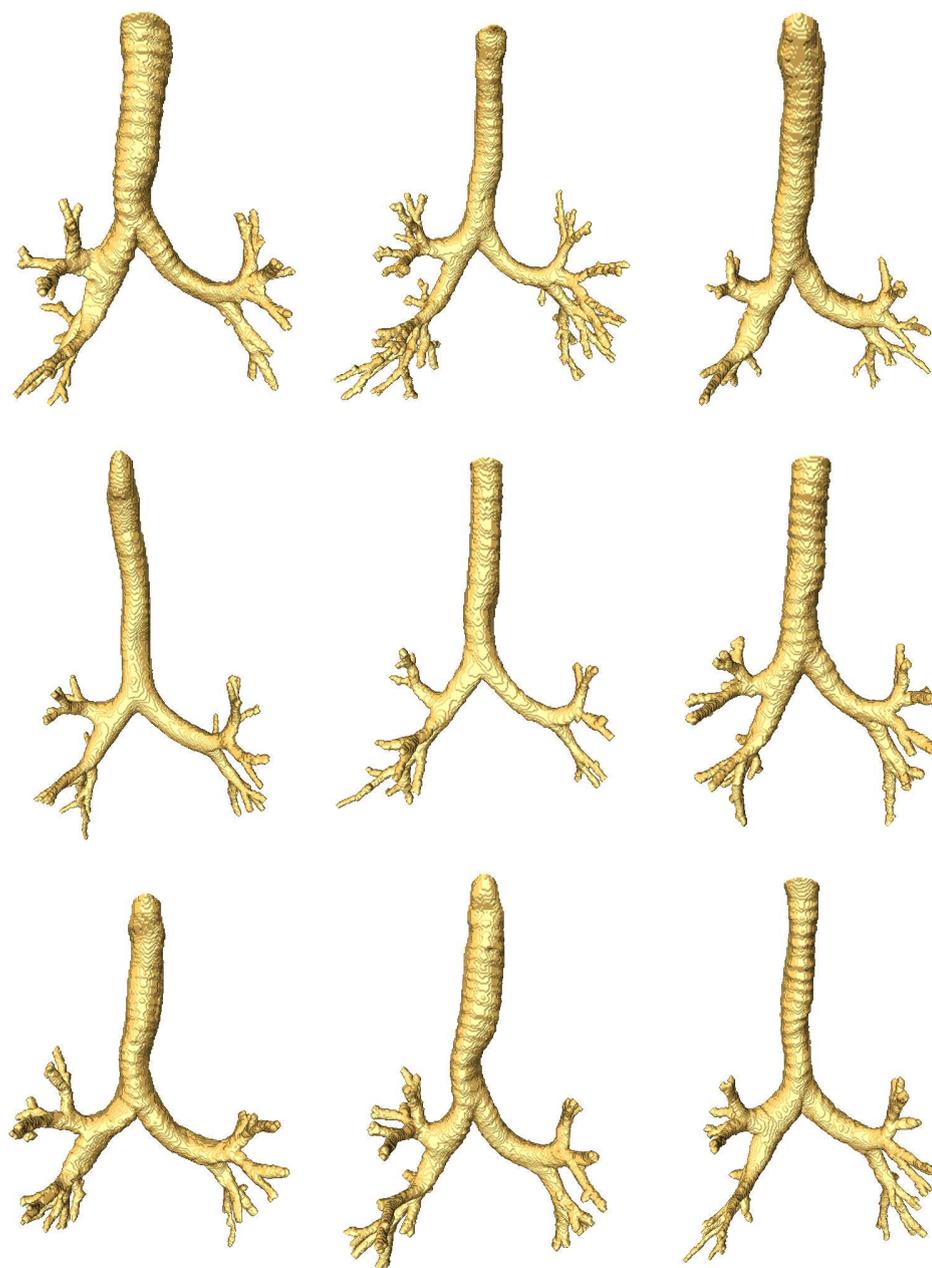


FIGURE 11.5: 3D visualization of segmentation results by proposed method.

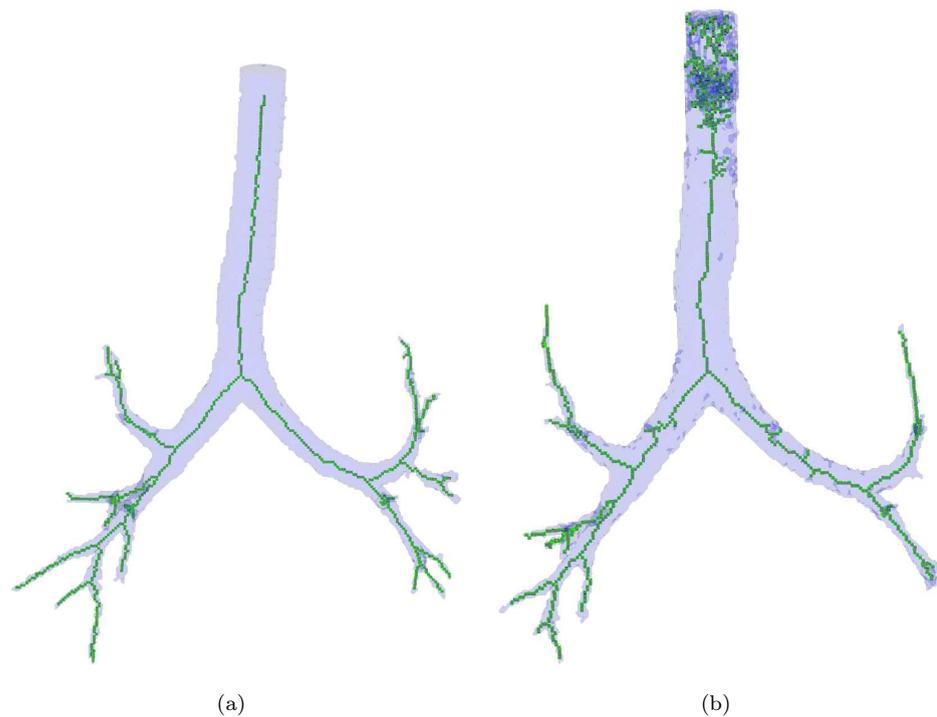


FIGURE 11.6: An example of skeletonisation results (the same algorithm) on the tree segmented from the same CT dataset. (a): The skeleton of the tree segmented based on hole closing; (b): The skeleton of the tree segmented using simple region growing approach. Note that in this case the skeleton is much more complicated (especially in top of the trachea) even if the tree is segmented reasonably well.

Chapter 12

Robust 3D Skeletonisation of Pulmonary Airway Tree Structures

In the chapter 3.1 we present a set of properties which are used to evaluate quality of the skeleton in particular applications. In this thesis we are interested in producing skeletons which hold all of such properties. That is, the skeleton of bronchial tree should be one voxel thick (singularity), it should have the same topology as original object (homotopic), it should be generated fully automatically (automation) and fast (cost effectiveness). The skeleton should also be perfectly centered (centricity) and it should be robust to noise (robustness) to allow reliable measurement.

The first four properties we can guarantee by focus only on skeletonisation algorithms which are constructed using well defined mathematical notions and have formal proofs of these properties. Based on literature of the subject, we select skeletonisation algorithms which belongs to the homotopic thinning group either in the voxel: PAL98 (Palagyi and Kuba, 1998), PAL06 (Palagyi et al., 2006), RAY (Raynal, 2010) or cubical complex framework: ACK3 (Bertrand and Couprie), ACK3A (Bertrand and Couprie), TCC (Chaussard, 2010). We select older methods and recently proposed ones.

To evaluate quality of the results based on the last two properties (centricity, robustness) we prepared a set of experiments. As an input we use special set of three-dimensional bronchial tree phantoms. In this set we include models which can be categorized into fourteen different classes: basic (straight) or extended (blend) models without any additional distortions, the same models with added noise (3%, 5% and 10% of object's volume) and finally models with noise (3%, 5% and 10% of object's volume) and smoothed (see Fig. 12.1). To prepare such phantoms, we use strategy presented in chapter 10.

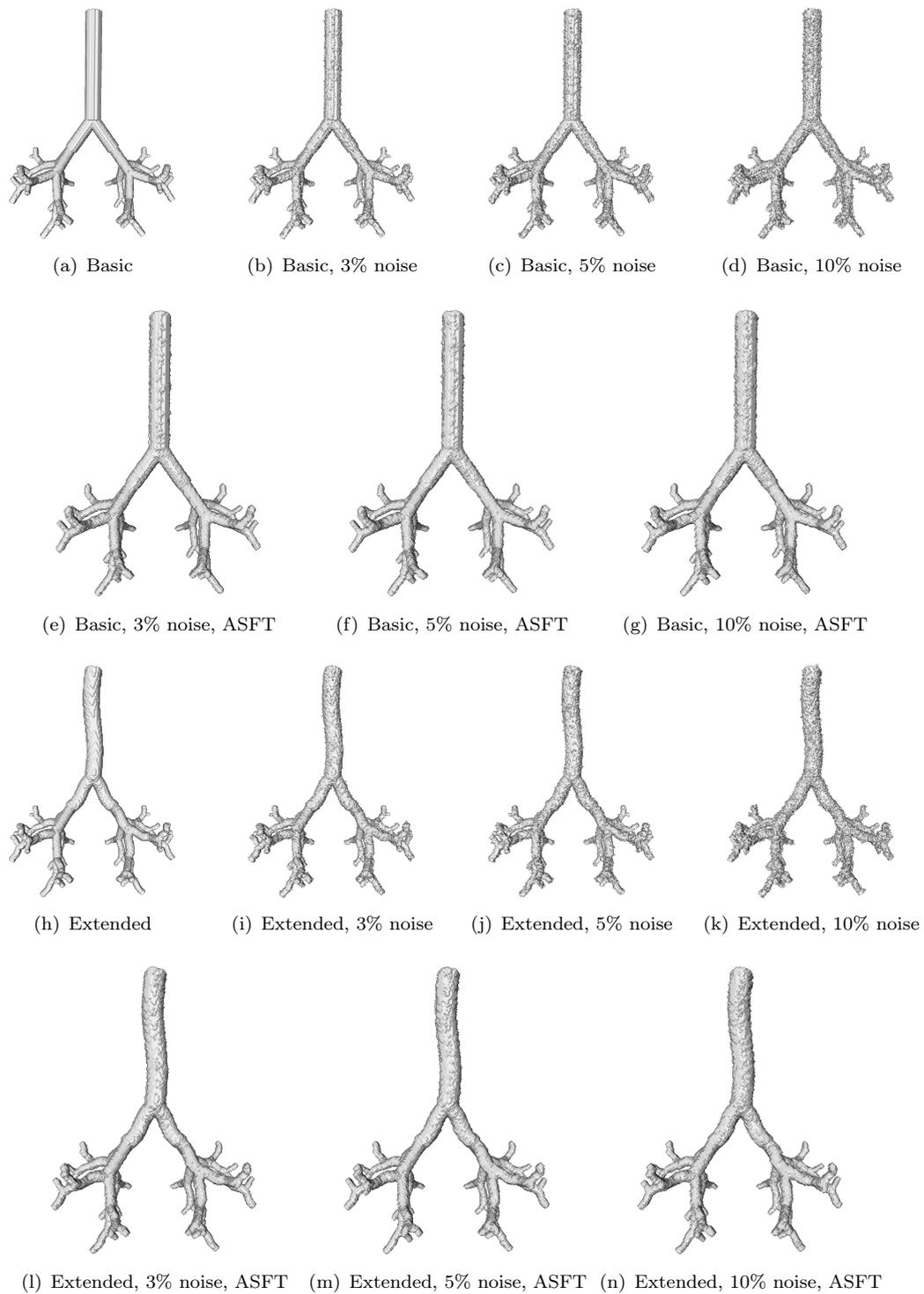


FIGURE 12.1: The set of bronchial tree phantoms used for the experiments

12.1 Skeleton Centricity

In this section, we compare the centricity property between different skeletonisation algorithms.

Globally, centricity property will be tested by measuring the "difference" between the medial axis of a shape X , which hold from the definition perfect centricity property, and the skeleton of the same shape X . The "difference" will be measured using the dissimilarity measure proposed by [Dubuisson-Jolly and Jain \(1994\)](#) (see also chapter 7.4.3).

We conducted our experiments using the discrete adaptive medial axis introduced in chapter 7.1, since, it has ability to preserve features at different scale. In addition the discrete adaptive medial axis is based on the discrete λ -medial axis which cope reasonably well with shape deformation like rotation or noise ([Chaussard, 2010](#); [Chaussard et al., 2009](#)). Therefore, it perfectly fit in our application where we need generate medial axis of the "deformed" airway trees consists of the large and thin branches at the same time.

To make an experiment more robust, we have also performed simple pruning strategy to remove spurious branches occurred using some skeletonisation algorithms, and to be sure that length of the skeleton is not larger than the medial axis. In the perfect case the skeleton should be subset of the corresponding medial axis. The drawback of the pruning is the problem of removing the terminal segments of the tree. However, since they are very thin they do not have important impact on the experiment results. As a pruning strategy we perform several steps of the basic homotopic thinning algorithms (see chapter 3.3 algorithm 3.3.1).

In table 12.1 and 12.2 we give the average dissimilarity between $DAMA(X, \gamma)$ and the skeleton $S(X)$ generated by the all tested algorithms. The parameter γ was evaluated experimentally and separately for each phantom to produce well filtered medial axis. Results are given for all fourteen classes of phantoms. Values shown are averages of results obtained for the 10 different models in each class. The last row indicates all the results averaged together. The lower dissimilarity value is better.

TABLE 12.1: Average dissimilarity between DAMA of X and skeleton of X calculated for basic phantoms (10 phantoms in each class).

	Basic	Noise			Noise+ASFT			Avr
		3%	5%	10%	3%	5%	10%	
ACK3	0.081	0.149	0.191	0.242	0.137	0.143	0.148	0.155
ACK3a	0.149	0.211	0.240	0.311	0.219	0.225	0.241	0.228
PAL06	0.097	0.138	0.169	0.230	0.090	0.100	0.137	0.140
PAL98	0.129	0.244	0.329	0.406	0.173	0.177	0.210	0.240
RAY	0.117	0.199	0.271	0.413	0.207	0.210	0.227	0.230
TCC	0.097	0.142	0.166	0.264	0.124	0.168	0.178	0.160

TABLE 12.2: Average dissimilarity between DAMA of X and skeleton of X calculated for extended phantoms (10 phantoms in each class).

	Extended	Noise			Noise+ASFT			Avr
		3%	5%	10%	3%	5%	10%	
ACK3	0.078	0.181	0.183	0.186	0.151	0.154	0.157	0.160
ACK3a	0.165	0.244	0.256	0.258	0.201	0.226	0.239	0.230
PAL06	0.100	0.141	0.145	0.198	0.097	0.101	0.123	0.130
PAL98	0.182	0.246	0.317	0.440	0.179	0.211	0.235	0.260
RAY	0.165	0.182	0.227	0.355	0.207	0.258	0.258	0.240
TCC	0.126	0.160	0.176	0.190	0.159	0.172	0.175	0.170

12.2 Robustness to Noise

It is well known that many skeletonisation strategies are sensitive to border noise and geometrical deformations. Since, the segmented airway trees can be corrupted as results of the not sufficient segmentation, or they have irregular border due to anatomical characteristic, it is useful to test how skeletonisation strategies work in practice.

The quality of the skeleton in respect to robustness to noise can be simply evaluated by counting the number of spurious branches of the skeleton appeared due to presence of noise. In practice, we only need to count the number of the end points of the skeleton, which directly correspond to the number of terminal branches and then compare this value to the expected one. The reference value is know for each phantom and it is constant for the class.

In table 12.3 and 12.4 we give the average number of the spurious skeleton branches. Results are given for all fourteen classes of phantoms. Values shown are averages of results obtained for the 10 different models in each class. The lowest value is better, and 0 means that no extra branch appear.

TABLE 12.3: Average number of the spurious skeleton branches calculated for basic phantoms (10 phantoms in each class).

	Basic	Noise			Noise+ASFT		
		3%	5%	10%	3%	5%	10%
ACK3	0	150	243	515	9	17	13
ACK3a	0	35	80	232	0	0	0
PAL06	0	52	112	325	0	0	0
PAL98	3	140	210	425	28	38	49
RAY	0	39	90	280	0	0	0
TCC	0	0	0	0	0	0	0

Figure 12.2, 12.3, 12.4 and 12.5 show skeletonisation results, using each of compared method, on exemplary phantoms.

TABLE 12.4: Average number of the spurious skeleton branches calculated for extended phantoms (10 phantoms in each class).

	Extended	Noise			Noise+ASFT		
		3%	5%	10%	3%	5%	10%
ACK3	0	160	242	553	7	15	13
ACK3a	0	30	79	218	0	2	0
PAL06	0	50	122	341	0	1	0
PAL98	5	159	227	426	45	42	55
RAY	0	52	80	290	0	0	0
TCC	0	0	0	0	0	0	0

12.3 Discussion

Algorithm PAL06 yield the best results in the centricity experiment, however algorithms TCC and ACK3 performs nearly as well as PAL06. On the other hand, the PAL06 algorithm shows that is not robust to noise which limits its practical application. This drawback of the PAL06 algorithm was expected, since, in [Palagyi et al. \(2006\)](#) authors try to use this skeletonisation strategy in the problem of generating skeleton of the bronchial trees without success. Therefore, they propose complex pruning strategy to remove spurious branches. The reported results shows that proposed pruning method works well in general case, however it needs several additional steps to perform and it not guarantee that we achieve well centered skeleton with detected all branches of the tree.

Taking above into consideration experiments shows that, the recently proposed TCC algorithm fares better on the average than other algorithms, with regards to both criteria. As addition, we performed some tests using TCC algorithm on the real airways segmented from the real CT datasets to see how it work in practice (see Fig. 12.6). However, since, the reference values were missing we can only evaluate skeleton properties qualitatively. The results showed that the generated skeletons were centered and spurious branches do not appeared. However, we notice that in some cases, very small branches are missing in the skeleton. This, property can be considered as a drawback, however such small branches, in general, are a segmentation errors and do not provide useful informations. Even if they correspond to the real segments of the tree, making further measurements on such segments can be source of errors. Note that, other tested algorithms missed the segments very rarely, which is expected behavior, since they are not robust to noise as much as the TCC algorithm. However, as long as it is very important to remove spurious branches appeared in the other tested algorithms, one has to use some additional pruning step. However, common pruning strategies can cause also missing branches to disappear.

The results achieved by the TCC algorithm are very promising and by using it we can solve many problems reported in the literature of the subject. TCC algorithm allow us to use the skeleton without any additional refinement steps, like for instance pruning, in general case. Since, the TCC algorithm perform well according to all desirable properties we consider it as a robust tool which help us to perform experiments in the next chapter.

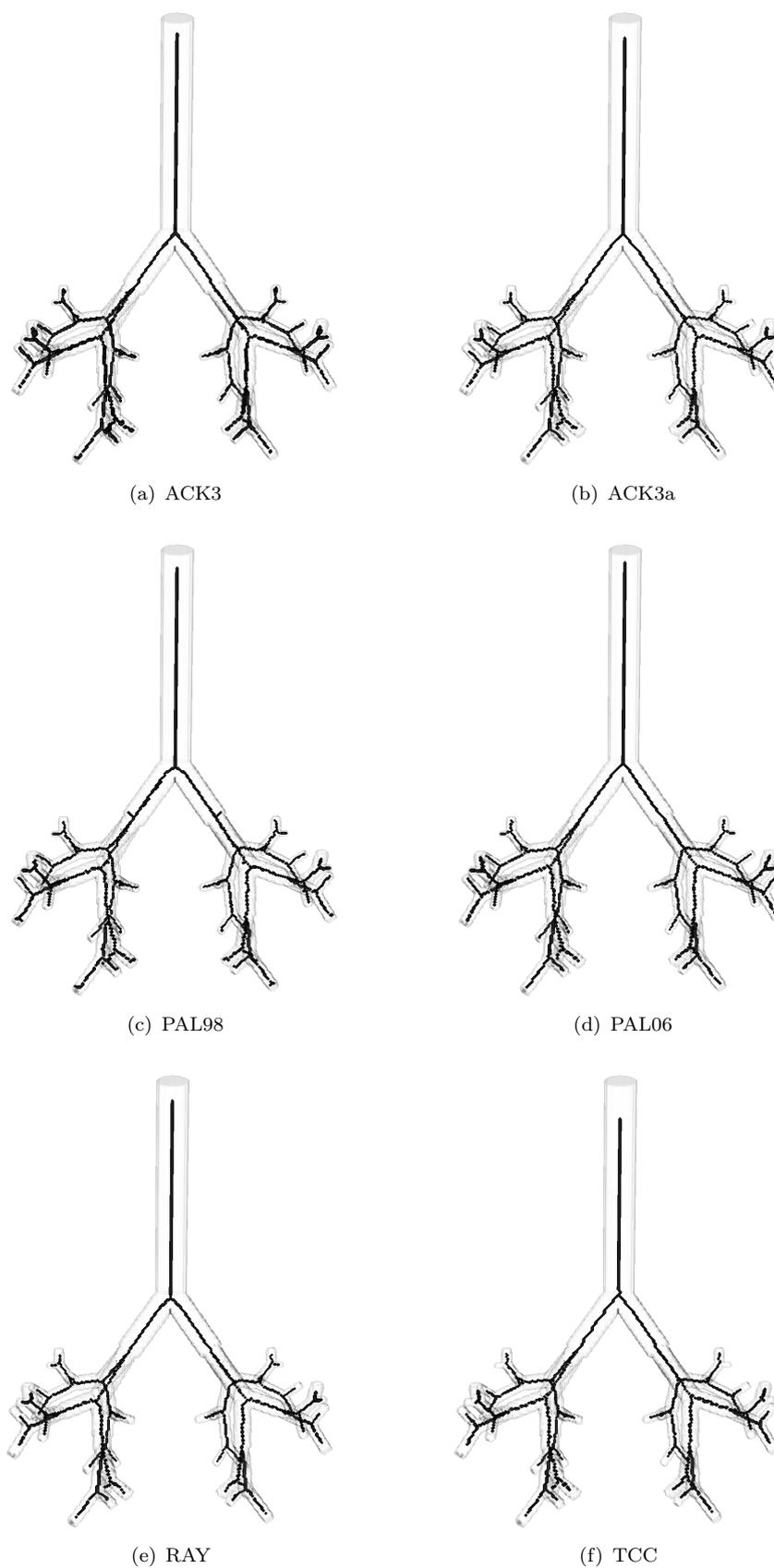


FIGURE 12.2: The skeletonisation results of the basic phantom.

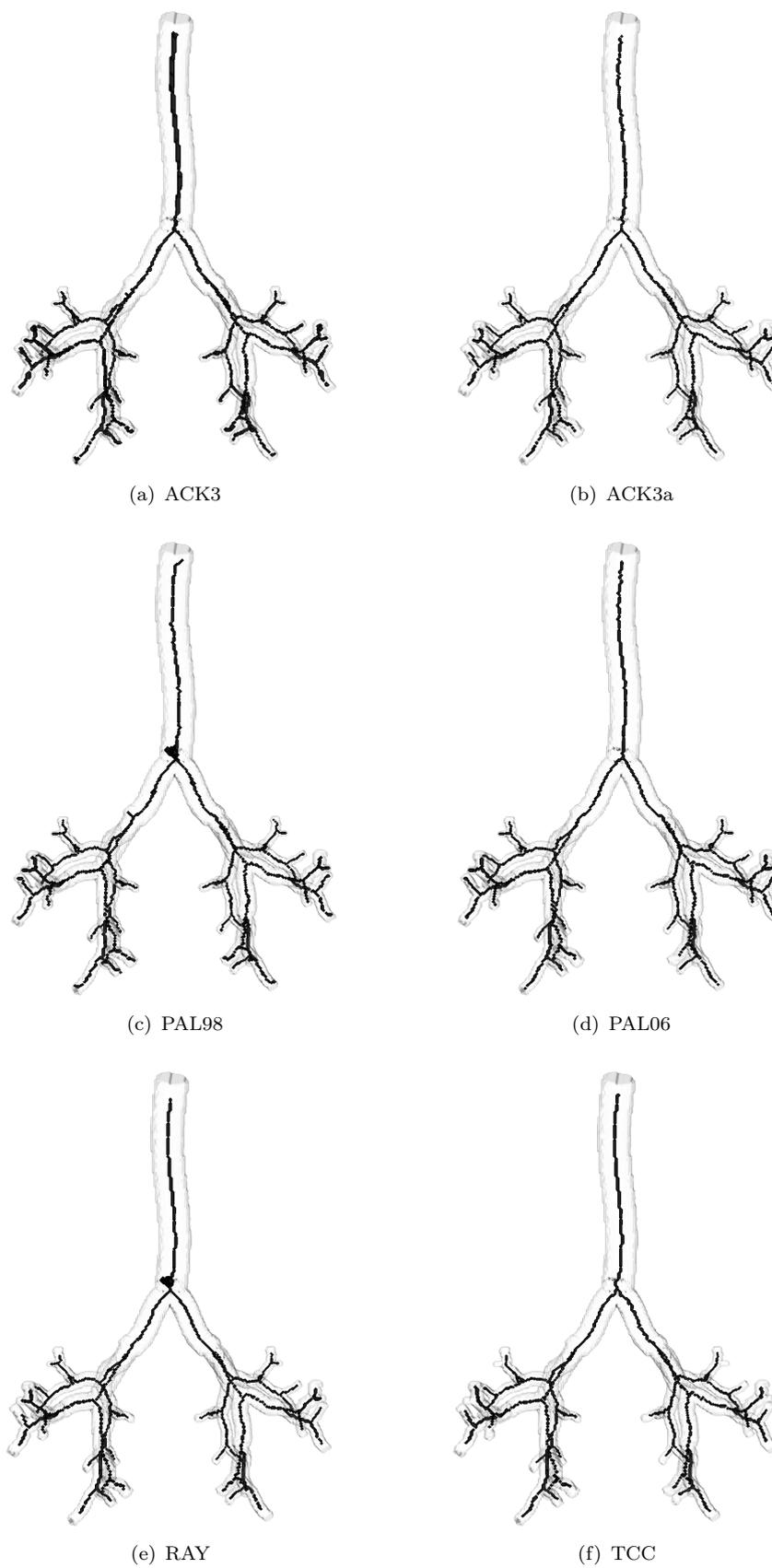


FIGURE 12.3: The skeletonisation results of the extended phantom.

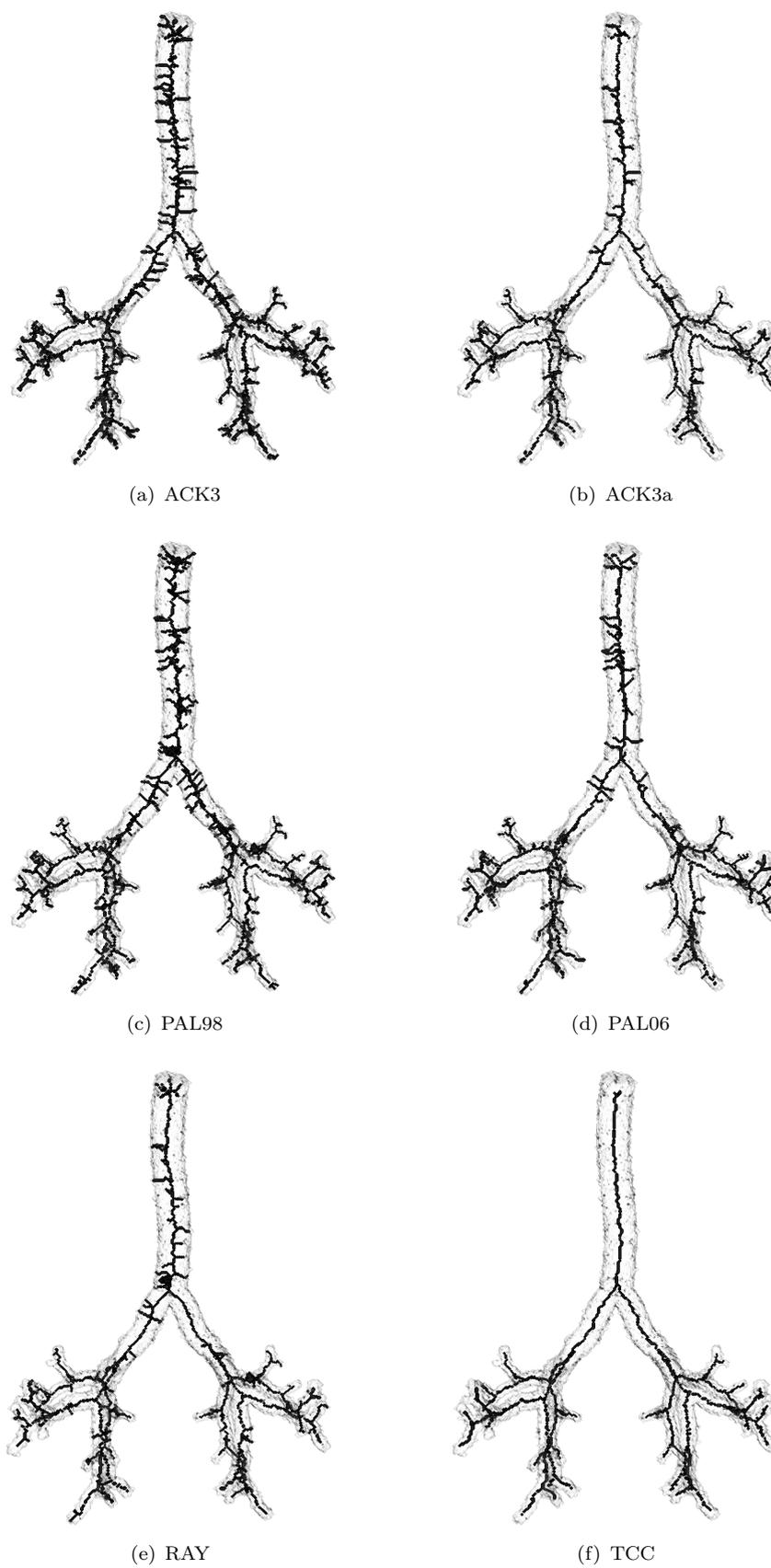


FIGURE 12.4: The skeletonisation results of the extended phantom with 5% of noise.

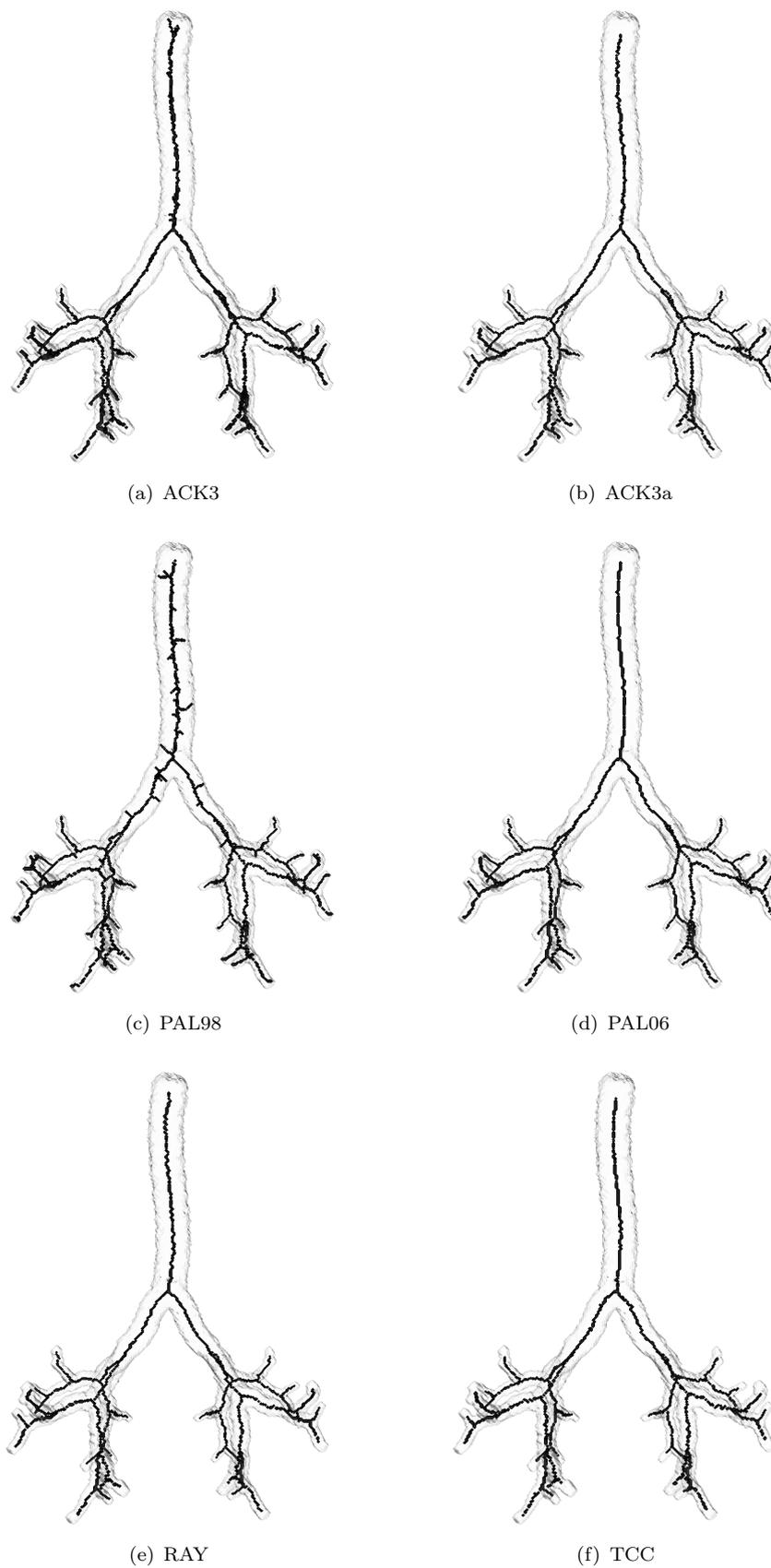


FIGURE 12.5: The skeletonisation results of the extended phantom with 10% of noise followed by ASFT filtering.

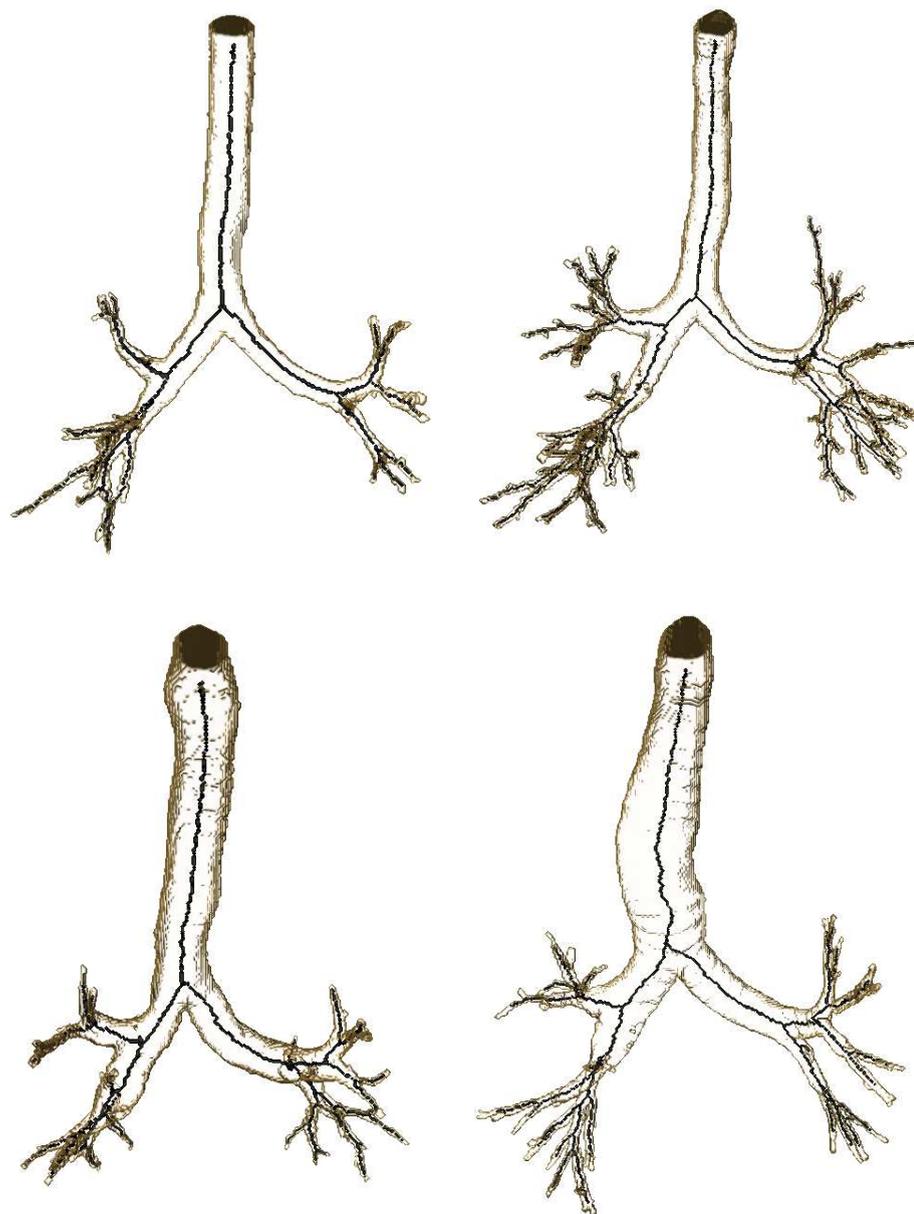


FIGURE 12.6: The TCC skeletonisation results of the four segmented airways tree from the real CT dataset.

Chapter 13

Accurate Cross Sections Generation of The Human Airways Trees

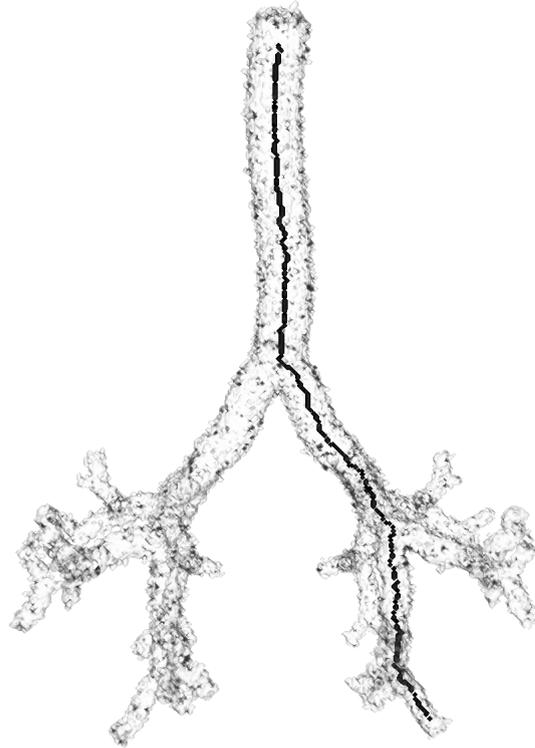
To generate accurately orthogonal cross section of the airway tree we need to evaluate a local direction of the airway segment in each point of interest. This can be done by using one of the tangent estimators along three-dimensional discrete curve (see chapter 6), defined by the skeleton of the tree. However, such curve can be corrupted, due to presence of noise or some arbitrary choices made by used skeletonization strategy. On the other hand, the proper tangent direction is crucial to allow reliable measurements. Thus, the strategy used to solve such problem has to be chosen carefully. The tangent estimator should be accurate and robust and should allow to work with 3D noisy curves (not smoothed) at fixed resolution.

To evaluate which tangent estimator can give us the most accurately results, in our particular application, we experiment several tangent estimators on the 3D bronchial models with known diameter. We use, for the experiment, exactly the same set of airway phantoms as presented in previous chapter, that is, 140 phantoms classified into 14 groups (simple and extended phantoms with distortion introduced in chapter 10).

General idea of the experiment is presented in the next subsection. Note that we use two separate strategy for the simple and extended phantoms respectively.

13.1 Methodology of Tangent Estimators Comparison

For the models without noise, we generate a set of the cross sections of the tree along arbitrary selected path. Selected path is the longest one according to the model (see Fig. 13.1) and is always the same for all estimators. The path used in our experiment was defined using the most reliable skeletonisation algorithm (TCC), according to results introduced in chapter 12.



(a) Extended, 10% noise

FIGURE 13.1: The 3D reconstruction of one of bronchial tree models with selected path (black curve) used for experiments

To avoid errors we focus only on points where the branch and cross sections are well defined, that is, where two branches do not have any intersection. On each cross section, we calculate the lumen area and diameter of the tree. For this purpose we use very simple method based on counting the number of segmented points which corresponds to the tree. Then, we compare calculated value to the reference value extracted from the phantom.

For the models with added noise, since, the noise locally change diameter of the tree we lost reliable information about the reference diameter of the phantom. Therefore, in practice we need to define two separate strategy, one for the models with and without noise respectively. We are interested in algorithms, both skeletonization and tangent estimation, which are noise insensitive, as much as possible or used together they have ability to compensate error introduced by presence of noise. Therefore, we can expect that results produced on the noisy model should be the same as on the corresponding model without noise. Therefore, to evaluate measurement error we evaluate tangent direction based on the path generated on noisy model. However, the lumen diameter of the each cross section is evaluated on the corresponding model without noise. In this case if the results will be accurate we can say that the pair of skeletonisation and tangent estimation algorithm produce results which are not affected by the noise.

13.2 Results and Discussion

In table 13.1 we give the average root mean square error (RMSE) between the reference lumen diameter and estimated one, along whole selected path. For estimation we use several different tangent estimators. The simple discrete methods NAvr1, NAvr3 and NAvr5 based on fixed computational window set to 1, 3 and 5 respectively, and following strategy proposed by (Matas et al., 1995) to evaluate tangent direction. We also use three adaptive discrete methods: The CSTE which based on curve segmentation into set of DSSs (Coeurjolly et al., 2000) and our new 3D λ -MSTD estimator. We test two version of λ - *MSTD* we call it L-MSTD and L-MSTDt2 where we set thickness used for DSS recognition to 1 and 2, respectively. The last method in comparison belong to the continuous estimator group and use uniform cubical splines to locally approximate the curve (de Vieilleville and Lachaud, 2009). All select, for the comparison, tangent estimators have different properties and abilities to deal with noise and curves with low and high curvature variations.

TABLE 13.1: Average RMSE between reference lumen diameter and evaluated one calculated for basic and extended phantoms (10 phantoms in each class).

	L-MSTD	L-MSTDt2	NAvr1	NAvr3	NAvr5	CSTE	STE
Basic	1.03	1.11	2.62	0.97	0.99	1.11	1.11
3% noise	1.27	1.12	10.53	4.14	2.02	1.07	2.16
5% noise	1.21	1.09	10.37	4.62	5.97	2.58	0.96
10% noise	1.23	1.03	12.08	3.27	1.17	2.09	1.10
3% noise, ASFT	1.23	1.06	8.83	1.70	0.98	1.16	1.01
5% noise, ASFT	1.31	1.07	11.70	1.83	1.00	1.18	1.05
10% noise, ASFT	0.99	0.85	7.99	1.73	0.94	1.19	1.52
Extended	1.09	1.02	10.47	1.70	1.05	2.42	0.97
3% noise	1.16	1.05	4.45	1.47	1.03	1.42	1.27
5% noise	1.32	1.10	14.83	2.68	1.15	1.97	1.16
10% noise	1.21	1.12	16.59	2.43	1.18	2.01	1.30
3% noise, ASFT	1.38	1.06	5.12	1.47	1.00	1.16	1.08
5% noise, ASFT	1.35	0.99	4.08	1.50	0.99	3.00	1.02
10% noise, ASFT	1.00	1.07	14.05	3.45	1.00	3.01	1.12
Average	1.202	1.057	9.557	2.360	1.460	1.860	1.196

Figure 13.2 and 13.3 shows the evolution of the lumen for the L-MSTD, L-MSTDt2, NAvr5 and STE along whole path, extracted from the basic and extended phantoms, respectively. To make differences better visible we removed from the plot the worst algorithms (according to results in table 13.1), that is, NAvr1, NAvr3 and CSTE.

As we expect the simplest method, algorithm NAvr1 and NAvr3, with fixed computational window, gives very poor results in general. Only in simplest case (basic algorithm without any distortions) the NAvr3 gives reasonably results. Better results can

be achieved by simply increasing the size of the computational window as in NAvr5. It means that, this size of the computational window is more adopted to the local curve geometry than previous ones. The algorithm NAvr5 is classified in the middle of our comparison. However, its practical application is limited, since, the computational window is fixed and in some part of the curve it is not well adopted anymore, which cause large errors in results, and makes NAvr5 not reliable in general case.

One can be surprised, that the CSTE estimator, which is an adaptive method produce result worse than NAvr5. However, the CSTE method is based on very simple curve segmentation into set of maximal segments. The method does not use the notions of tangential cover nor any combinations of DSSs. Thus, CSTE is very sensitive to small distortions introduced to the curve, even more than NAvr3 in which we averaging results. Locally CSTE can produce very small segments and in consequences not accurate estimation. However, this method is multigrid convergent, in opposite to any method with fixed computational window. Therefore we can expect better results if we increase resolution.

The best and comparable results were achieved by the L-MSTD, L-MSTDt2 and STE, respectively. These estimators produce results close in average to 1 and with very small standard deviation, which makes them very accurate and reliable. Thus, the experiments also verify the quality of the new tangent estimator (L-MSTD), proposed in this thesis. In practice, its behavior in average, is the same as the continuous method (STE). If we consider the fact that our method is much simpler (parameterless), faster (linear complexity) and has good multigrid convergence behavior, the results are very promising. Note that, in the simplest case, where the segments are straight lines (see fig. 13.2(a)) the STE estimator can give not accurate results. Moreover, according to results achieved by the L-MSTDt2 algorithm, the biggest drawback of the proposed method, that is, its sensitivity to the small distortion introduced to the curve, can be simply overcome by changing the vertical thickness used for the maximal segment recognition. Such modification can cause algorithm to lose accuracy in general case. However, in our particular application, where we work with curves with relatively low curvature variations, we achieve the best results in the comparison. Therefore, all of these three methods can be used in the problem of quantitative analysis of the bronchial trees. However, only L-MSTD estimator (or its modification L-MSTDt2) can guarantee speed and higher accuracy with increasing quality of the input data and the user does not need to set any parameters.

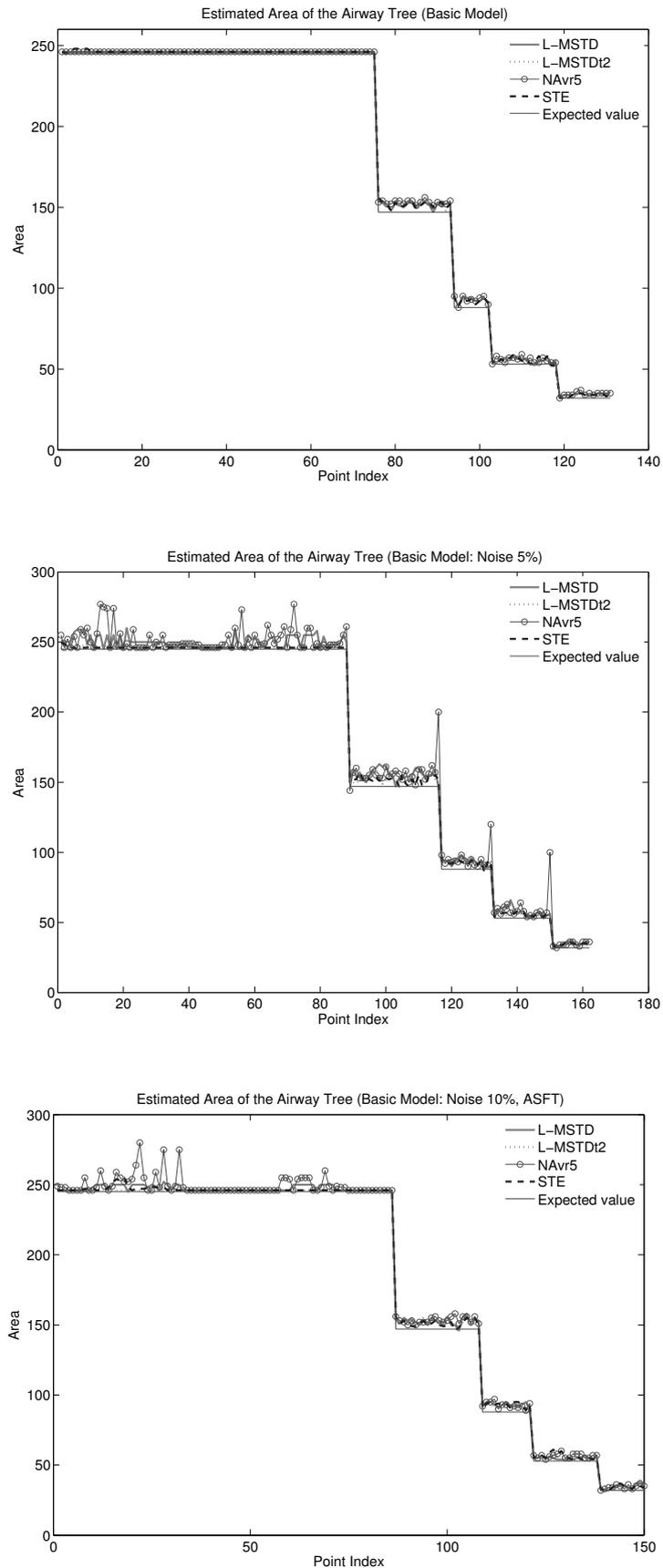


FIGURE 13.2: A comparison between reference value from the phantom and estimated area using different tangent estimators.

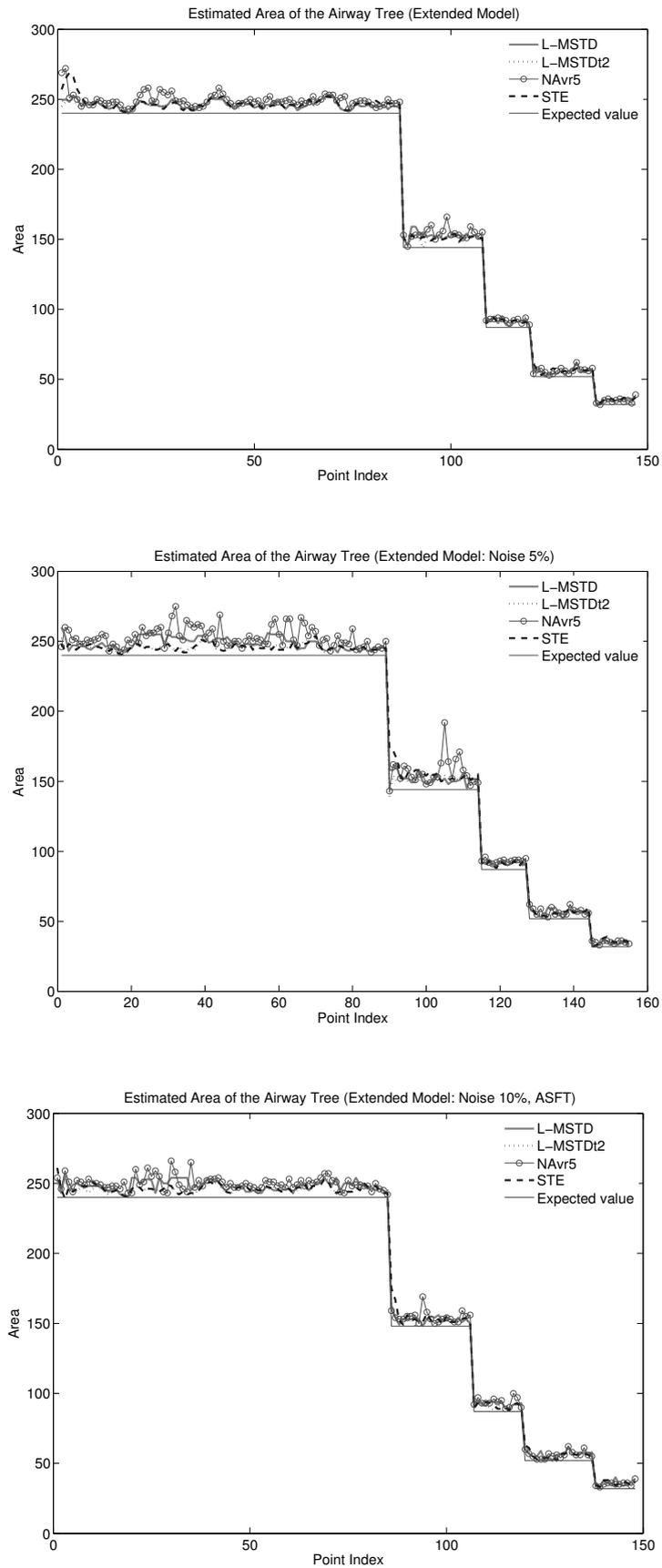


FIGURE 13.3: A comparison between reference value from the phantom and estimated area using different tangent estimators.

Chapter 14

Conclusions

In this thesis, we studied different methods used for image processing and analysis which are build on top of the discrete topology and geometry. First, we define basic notions used in discrete topology and geometry then we showed that several class of discrete methods like skeletonisation algorithms, medial axes, tunnels closing algorithms and tangent estimators, are widely used in several different practical application. We also make an analysis of properties and limitations of such methods.

In the next part we introduce two new methods for medial axis filtering of objects with features at different scales. Then we compared our new filtering criterions with other filtering methods. For the sake of fair comparison, we introduced new methodology to conduct experiments in which we evaluate several different properties of filtering methods. The experiments shows the superiority of our approaches and makes new proposed methods useful in many application. In this part we also propose a new discrete adaptive tangent estimator along three-dimensional discrete curves. Then we evaluate it multigird convergence property.

Finally, in the last part we showed that discrete geometry and topology algorithms can be useful in the problem of quantitative analysis of the human airway trees based on computed tomography images. According to proposed in the literature design of such system we applied discrete topology and geometry algorithms to solve particular problems at each step of the quantitative analysis process. We compare the quality of different algorithms using set of experiments conducted on computer phantoms and real CT dataset. We show that recently proposed methods which works in cubical complex framework, together with methods introduced in this work can overcome problems reported in the literature and can be a good basis for the further implementation of the system for automatic quantification of bronchial tree properties.

14.1 Contribution of this thesis

Important contributions of this work have been done in different fields of research:

Digital Topology - In the digital topology our contributions are two new medial axis filtering methods which are able to effectively prune medial axes points in object with features at different scales:

- The hierarchical scale medial axis which is a new method for medial axis filtering. The method is based on previously proposed scale axis transform. Our method is free of drawbacks introduced in the previously proposed method.
- The discrete adaptive medial axis which is based on discrete λ -medial axis where the filtering parameter is dynamically adapted to the local size of the object.

In this field we have also propose a new methodology for fair comparison between different filtering methods. We define some set of different experiments which allow to compare filtering criterions with different meaning of the filtering parameter.

Finally, we compare skeletonisation algorithms based on desirable set of properties in real practical application which verify that the recently proposed method TCC posses a set of properties which makes this method useful in many application, including problem of quantitative analysis of the human airway trees.

Digital Geometry - In this field we proposed a new tangent estimators along three-dimensional discrete curves, called 3D λ -MSTD. The method is efficient and parameterless. An we show experimentally that this method has multigrid convergent behavior. We also show that in practical application this method gives results as good as the continuous methods.

Medical Image Processing - In this field we propose a robust method for segmenting airway tree from CT datasets. The method is based on the tunnel closing algorithm and is used as a tool to repair, damaged by acquisition errors, CT images. We also proposed an algorithm for creation of an artificial model of the bronchial tree and we used such model to validate algorithms presented in this work.

The following publications resulted from the work presented here:

1. Postolski, M., M. Couprie and M. Janaszewski. Scale Filtered Euclidean Medial Axis. *DGCI 2013*: 360-371. 2013
2. Pluta, K., M. Janaszewski and M. Postolski. New Algorithm for Modeling of Bronchial Trees. *Image Processing and Communications*. 17(4): 179-190. 2013

3. Postolski, M., M. Janaszewski, Y. Kenmochi, and J-O. Lachaud. Tangent Estimation Along 3D Digital Curves. *21st International Conference on Pattern Recognition, Japan*: 2079-2082. 2012
4. Janaszewski, M., M. Postolski, and L. Babout. Robust Algorithm for Tunnel Closing in 3D Volumetric Objects Based on Topological Characteristics of Points. *Pattern Recognition Letters*. 32(16): 2231-2238. 2011
5. Postolski, M., M. Janaszewski, L. Jopek and L. Babout. Tangent Direction Estimation in Each Point of 3D Volumetric Curve in Application to Quantitative Human Airway Trees Analysis. *Journal of Academy of Science and Technology, Automatics*, 15:219-234. 2011
6. Janaszewski, L., M. Postolski, L. Babout and L. Jopek. Topological and Geometrical Hole Closing in 3D Objects. *Journal of Academy of Science and Technology, Automatics*, 15:131-141. 2011
7. Postolski, M., M. Janaszewski, L. Jopek and L. Babout. 3D Skeletonization of Pulmonary Airway Tree structure. *Journal of Academy of Science and Technology, Automatics*, 14:337-353. 2010
8. Fabijanska, A., M. Postolski, M. Janaszewski and L. Babout. Comparative Analysis of Bronchial Tree Segmentation Algorithms. *Journal of Academy of Science and Technology, Automatics*, 13:821-830. 2009
9. Janaszewski, L., L. Babout, M. Postolski and L. Jopek. Hole Segmentation in Volumetric Objects. *Journal of Academy of Science and Technology, Automatics*, 13:855-864. 2009
10. Janaszewski, L., L. Babout, M. Postolski and L. Jopek. Hole Closing in 3D Volumetric Objects. *Journal of Academy of Science and Technology, Automatics*, 13:865-878. 2009
11. Postolski, M., M. Janaszewski, A. Fabijanska, L. Babout, M. Couprie, M. Jedrzejczyk and L. Stefanczyk. Reliable Airway Tree Segmentation Based on Hole Closing in Bronchial Walls. *Advances in Soft Computing, Springer*, 57:389-396. 2009
12. Janaszewski, L., M. Postolski, L. Babout and E. Kacki. Comparison of Several Centerline Extraction Algorithms for Virtual Colonoscopy. *Advances in Soft Computing, Springer*, 65:241-254. 2009

Bibliography

- Gamal H. Abdel-Hamid and Yee-Hong Yang. Multiresolution skeletonization: An electrostatic field-based approach. In *ICIP (1)*, pages 949–953, 1994.
- R.O. Duda A.E. Brain and J.H. Munson. Graphical data processing research study and experimental investigation. Technical report, AI Center, SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025, Aug 1965.
- Narendra Ahuja and Jen-Hui Chuang. Shape representation using a generalized potential field model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(2):169–176, 1997. ISSN 0162-8828.
- Zouina Aktouf, Gilles Bertrand, and Laurent Perroton. A three-dimensional holes closing algorithm. In Serge Miguet, Annick Montanvert, and Stéphane Ubéda, editors, *DGCI*, volume 1176 of *Lecture Notes in Computer Science*, pages 36–47. Springer, 1996. ISBN 3-540-62005-2.
- Zouina Aktouf, Gilles Bertrand, and Laurent Perroton. A three-dimensional holes closing algorithm. *Pattern Recognition Letters*, 23(5):523–531, 2002.
- Brett Allen, Brian Curless, and Zoran Popović. **The space of human body shapes: reconstruction and parameterization from range scans.** *ACM Trans. Graph.*, 22(3):587–594, July 2003. ISSN 0730-0301.
- Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications*, 19:127–153, 2000.
- Ian M. Anderson and James C. Bezdek. Curvature and tangential deflection of discrete arcs: A theory based on the commutator of scatter matrix pairs and its application to vertex detection in planar shape data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:27–40, 1984. ISSN 0162-8828.
- D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. **SCAPE: Shape completion and animation of people.** *ToG*, 24(3):408–416, 2005.

- L. Antiga, B. Ene-Iordache, and A. Remuzzi. Computational geometry for patient-specific reconstruction and meshing of blood vessels from angiography. *IEEE Trans. Medical Imaging*, 22:674–684, 2003.
- C. Arcelli. Pattern thinning by contour tracing. *Computer Graphics and Image Processing*, 17(2):130–144, 1981.
- T. Asano, Y. Kawamura, R. Klette, and K. Obokata. Minimum-length polygons in approximation sausages. *Proceedings of the 4th International Workshop on Visual Form*, pages 103–112, 2001.
- D. Attali, J.-D. Boissonnat, and H. Edelsbrunner. Stability and computation of the medial axis — a state-of-the-art report. In *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, pages 109–125. Springer-Verlag, 2009.
- D. Attali and J.O. Lachaud. Delaunay conforming iso-surface, skeleton extraction and noise removal. *Computational Geometry: Theory and Applications*, 19:175–189, 2001.
- D. Attali and A. Montanvert. Modeling noise for a better simplification of skeletons. In *Proc. of International Conference on Image Processing*, pages III: 13–16, 1996.
- D. Attali, G. Sanniti di Baja, and E. Thiel. Pruning discrete and semicontinuous skeletons. In *Conference on Image Analysis and Processing*, volume 974 of *LNCS*, pages 488–493. Springer, 1995.
- Dominique Attali and Jacques-Olivier Lachaud. Constructing iso-surfaces satisfying the delaunay constraint: Application to the skeleton computation. In *ICIAP*, pages 382–387. IEEE Computer Society, 1999.
- Dominique Attali and Annick Montanvert. **Computing and simplifying 2d and 3d continuous skeletons**. *Computer Vision and Image Understanding*, 67(3):261 – 273, 1997. ISSN 1077-3142.
- Marco Attene and Bianca Falcidieno. **ReMESH: An interactive environment to edit and repair triangle meshes**. In *Shape Modeling International*, page 41. IEEE Computer Society, 2006. ISBN 0-7695-2591-1.
- S. Aylward and E. Bullitt. Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction. *IEEE Trans. Medical Imaging*, 21: 61–75, 2002.
- Stephen R. Aylward, Julien Jomier, Sue Weeks, and Elizabeth Bullitt. **Registration and analysis of vascular images**. *Int. J. Comput. Vision*, 55(2-3):123–138, November 2003. ISSN 0920-5691.

- L. Babout, T.J. Marrow, D. Engelberg, and P.J. Withers. X-ray microtomographic observation of intergranular stress corrosion cracking in sensitised austenitic stainless steel. *Mater. Sci. Technol.*, 22:1068–1075, 2006.
- A. Bac, N. V. Tran, and M. Daniel. **A multistep approach to restoration of locally undersampled meshes**. In *Geometric Modeling and Processing*, pages xx–yy, 2008.
- David C. Banks and Bart A. Singer. Vortex tubes in turbulent flows: Identification, representation, reconstruction. pages 132–139. Society Press, 1994.
- Gill Barequet and Micha Sharir. Filling gaps in the boundary of a polyhedron. *Computer Aided Geometric Design*, 12:207–229, 1995.
- Dirk Bartz, Wolfgang Strasser, Martin Skalej, and Dorothea Welte. Interactive exploration of extra- and intracranial blood vessels. In *In Proc. of IEEE Visualization*, pages 389–392, 1999.
- Gerhard H. Bendels, Ruwen Schnabel, and Reinhard Klein. Detail-preserving surface inpainting. In *The 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST)*, pages 41–48. Eurographics Association, Eurographics Association, November 2005.
- Thierry M. Bernard and Antoine Manzanera. Improved low complexity fully parallel thinning algorithm. In *ICIAP '99: Proceedings of the 10th International Conference on Image Analysis and Processing*, page 215, Washington, DC, USA, 1999. IEEE Computer Society. ISBN 0-7695-0040-4.
- G. Bertrand. On p-simple points. *Comptes rendus de l'Academie des Sciences, Serie Math*, 1(321):1077–1084, 1995.
- G. Bertrand. Sufficient conditions for 3D parallel thinning algorithms. In R. A. Melter, A. Y. Wu, F. L. Bookstein, & W. D. Green, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 2573 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 52–60, August 1995.
- G. Bertrand. Sufficient conditions for 3d parallel thinning algorithms. *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 2573 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 52–60, 1995.
- G. Bertrand and M. Couprie. Three-dimensional parallel thinning algorithms based on critical kernels.
- G. Bertrand and M. Couprie. A new 3d parallel thinning scheme based on critical kernels. *Discrete Geometry for Computer Imagery. Lecture Notes in Computer Science*, 4245: 580–591, 2006.

- Gilles Bertrand. Simple points, topological numbers and geodesic neighborhoods in cubic grids. *Pattern Recognition Letters*, 15(10):1003–1011, 1994.
- Gilles Bertrand. On critical kernels. *Comptes Rendus de l'Academie des Sciences, Serie Math*, 1(345):363–367, 2007.
- Gilles Bertrand and Michel Couprie. **Two-dimensional thinning algorithms based on critical kernels**. *Journal of Mathematical Imaging and Vision*, 31(1):35–56, May 2008.
- Gilles Bertrand and Grégoire Malandain. A new characterization of three-dimensional simple points. *Pattern Recogn. Lett.*, 15(2):169–175, 1994. ISSN 0167-8655.
- D. Bilgen. Segmentation and analysis of the human airway tree from 3d x-ray ct images. *Master's thesis*, 2000.
- Robert L. Blanding, George M. Turkiyyah, Duane W. Storti, and Mark A. Ganter. **Skeleton-based three-dimensional geometric morphing**. *Comput. Geom. Theory Appl.*, 15(1-3):129–148, February 2000. ISSN 0925-7721.
- V. Blanz, A. Mehl, T. Vetter, and H. p. Seidel. A statistical method for robust 3d surface reconstruction from sparse data. In *In Int. Symp. on 3D Data Processing, Visualization and Transmission*, pages 293–300, 2004.
- Volker Blanz and Thomas Vetter. **A morphable model for the synthesis of 3d faces**. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-48560-5.
- H. Blum. An associative machine for dealing with the visual field and some of its biological implications. *Biol. prototypes and synthetic systems*, 1:244–260, 1961.
- H. Blum. An associative machine for dealing with the visual field and some of its biological implications. In E. E. Bernard and M. R. Kare, editors, *Biological Prototypes and Synthetic Systems*, volume 1, pages 244–260, NY, 1962. Plenum Press. 2nd Annual Bionics Symposium, held at Cornell University, 1961.
- H Blum. A transformation for extracting new descriptors of shape. In *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, 1967.
- Jan Helge Bøhn and Michael J. Wozny. **A topology-based approach for shell-closure**. In *Selected and Expanded Papers from the IFIP TC5/WG5.2 Working Conference on Geometric Modeling for Product Realization*, pages 297–319, Amsterdam, The Netherlands, The Netherlands, 1992. North-Holland Publishing Co. ISBN 0-444-81662-3.
- G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27:321–345, 1984.

- G. Borgefors, I. Ragnemalm, and Sanniti di Baja. The euclidean distance transform: finding the local maxima and reconstructing the shape. In *Procs. of the 7th Scand. Conf. on image analysis*, volume 2, pages 974–981, 1991.
- Gunilla Borgefors, Ingela Nystrom, and Gabriella Sanniti Di Baja. Computing skeletons in three dimensions. *Pattern Recognition*, 32(7):1225 – 1236, 1999. ISSN 0031-3203.
- J. Bousquet, P.K. Jeffery, W.W. Busse, and et al. Asthma. from bronchoconstriction to airways inflammation and remodeling. *Am J Respir Crit Care Med*, 161:1720–1745, 2000.
- E.A. Boyden. Segmental anatomy of the lungs. *McGraw-Hill*, 1955.
- John Branch, flavio Prieto, and Pierre Boulanger. **Automatic hole-filling of triangular meshes using local radial basis function**. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, 3DPVT '06, pages 727–734, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2825-2.
- Jonathan W. Brandt and V. Ralph Algazi. Continuous skeleton computation by voronoi diagram. *CVGIP: Image Underst.*, 55(3):329–338, 1992. ISSN 1049-9660.
- Toby P. Breckon and Robert B. Fisher. Amodal volume completion: 3d visual completion. *Comput. Vis. Image Underst.*, 99(3):499–526, 2005. ISSN 1077-3142.
- Angela Brennecke and Tobias Isenberg. 3d shape matching using skeleton graphs. In *In Simulation and Visualization*, pages 299–310, 2004.
- L. Buzer. An elementary algorithm for digital line recognition in the general case. *Lecture Notes in Computer Science*, 3429:299–310, 2005.
- L. Calabi. A study of the skeleton of plane figures. Technical Report Technical Report 60429, Parke Mathematical Laboratories, 1965.
- F. Cazals and M. Pouget. Estimating differential quantities using polynomial fitting of osculating jets. *Comput. Aided Geom. Des.*, 22:121–146, 2005.
- F. Chabat, X.P. Hu, D.M. Hansell, and G.Z. Yang. Ers transform for the automated detection of bronchial abnormalities on ct of the lungs. *IEEE transactions on medical imaging*, 20, 2001.
- J.M. Chassery and A. Montanvert. *Géométrie discrète*. Hermès, 1991.
- John Chaussard. *Topological tools for discrete shape analysis*. PhD thesis, Université Paris-Est, December 2010.
- John Chaussard and Michel Couprie. Surface thinning in 3d cubical complexes. In *IWCIA*, pages 135–148, 2009.

- John Chaussard, Michel Couprie, and Hugues Talbot. **A discrete lambda-medial axis.** In Srećko Brlek, Christophe Reutenauer, and Xavier Provençal, editors, *Discrete Geometry for Computer Imagery*, volume 5810 of *Lecture Notes in Computer Science*, pages 421–433. Springer, 2009.
- John Chaussard, Michel Couprie, and Hugues Talbot. Robust skeletonization using the discrete lambda-medial axis. *Pattern Recognition Letters*, In Press,(9):1–10, 2010.
- F. Chazal and A. Lieutier. The λ -medial axis. *Graphical Models*, 67(4):304–331, 2005.
- Fu che Wu, Wan chun Ma, Ping chou Liou, Rung-Huei Laing, and Ming Ouhyoung. Skeleton extraction of 3d objects with visible repulsive force. In *Eurographics Symp. On Geometry Processing*, 2003.
- Chun-Yen Chen and Kuo-Young Cheng. **A sharpness-dependent filter for recovering sharp features in repaired 3d mesh models.** *IEEE Transactions on Visualization and Computer Graphics*, 14(1):200–212, January 2008. ISSN 1077-2626.
- W.J.R. Chen, D.S.P. Shiah, and C.S. Wang. A three-dimensional model of the upper tracheobronchial tree. *Bulletin of Mathematical Biology*, 42:847–859, 1980.
- Z. Chen and S. Molloi. Automatic 3d vascular tree construction in ct angiography. *Computerized Med. Imaging Graphics*, 27:469–479, 2003.
- R. Chiplunkar, J.M. Reinhardt, and E.A. Hoffman. Segmentation and quantitation of the primary human airway tree. *SPIE Medical Imaging*, 1997.
- Jen-Hi Chuang, Chi-Hao Tsai, and Min-Chi Ko. **Skeletonization of three-dimensional object using generalized potential field.** *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1241–1251, November 2000. ISSN 0162-8828.
- Wan chun Ma, Fu che Wu, and Ming Ouhyoung. Skeleton extraction of 3d objects with radial basis functions. In *Proceedings of Shape Modeling International 2003*, pages 207–215, 2003.
- U. Clarenz, U. Diewald, G. Dziuk, M. Rumpf, and R. Rusu. **A finite element method for surface restoration with smooth boundary conditions.** *Comput. Aided Geom. Des.*, 21(5):427–445, May 2004. ISSN 0167-8396.
- D. Coeurjolly. *Algorithmique et géométrie pour la caractérisation des courbes et des surfaces.* PhD thesis, Université Lyon 2, 2002.
- David Coeurjolly. Fast and accurate approximation of the euclidean opening function in arbitrary dimension. In *Proceedings of the 2010 20th International Conference on Pattern Recognition, ICPR '10*, pages 229–232, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-4109-9.

- David Coeurjolly, Isabelle Debled-Rennesson, and Olivier Teytaud. Segmentation and length estimation of 3d discrete curves. In Gilles Bertrand, Atsushi Imiya, and Reinhard Klette, editors, *Digital and Image Geometry*, volume 2243 of *Lecture Notes in Computer Science*, pages 299–317. Springer, 2000. ISBN 3-540-43079-2.
- David Coeurjolly, Isabelle Debled-Rennesson, and Olivier Teytaud. Segmentation and length estimation of 3d discrete curves. In *Digital and Image Geometry, Advanced Lectures [based on a winter school held at Dagstuhl Castle, Germany in December 2000]*, pages 299–317, 2001a.
- David Coeurjolly and Reinhard Klette. A comparative evaluation of length estimators of digital curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26:252–257, 2004.
- David Coeurjolly, Serge Miguet, and Laure Tougne. Discrete curvature based on osculating circle estimation. In *Proceedings of the 4th International Workshop on Visual Form, IWVF-4*, pages 303–312, 2001b.
- David Coeurjolly and Annick Montanvert. Optimal separable algorithms to compute the reverse euclidean distance transformation and discrete medial axis in arbitrary dimension. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):437–448, 2007. ISSN 0162-8828.
- David Coeurjolly and Stina Svensson. Estimation of curvature along curves with application to fibres in 3d images of paper. In *Proceedings of the 13th Scandinavian conference on Image analysis, SCIA'03*, pages 247–254, 2003.
- Nicu D. Cornea, M. Fatih Demirci, Deborah Silver, Ali Shokoufandeh, Sven J. Dickinson, and Paul B. Kantor. **3d object retrieval using many-to-many matching of curve skeletons**. In *Proceedings of the International Conference on Shape Modeling and Applications 2005, SMI '05*, pages 368–373, Washington, DC, USA, 2005a. IEEE Computer Society. ISBN 0-7695-2379-X.
- Nicu D. Cornea, Deborah Silver, and Patrick Min. Curve-skeleton properties, applications, and algorithms. *IEEE Trans. Vis. Comput. Graph.*, 13(3):530–548, 2007.
- Nicu D. Cornea, Deborah Silver, Xiaosong Yuan, and Raman Balasubramanian. Computing hierarchical curve-skeletons of 3d objects. *The Visual Computer*, 21(11):945–955, 2005b.
- M. Couprie. Note on fifteen 2d parallel thinning algorithms. In *Internal Report, Universit  de Marne-la-Valle*, 2006.
- M. Couprie and G. Bertrand. Topology preserving alternating sequential filter for smoothing 2d and 3d objects. *Journal of Electronic Imaging*, 13:720–730, 2004.
- M. Couprie and G. Bertrand. New characterizations of simple points, minimal non-simple sets and p-simple points in 2d, 3d and 4d discrete spaces. In *DGCI'08: Proceedings of*

- the 14th IAPR international conference on Discrete geometry for computer imagery*, pages 105–116, 2008.
- Michel Couprie and Gilles Bertrand. New characterizations of simple points in 2d, 3d, and 4d discrete spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):637–648, 2009.
- Michel Couprie and Gilles Bertrand. Discrete topological transformations for image processing. In Valentin Brimkov, editor, *Digital Geometry Algorithms, Theoretical Foundations and Applications to Computational Imaging*. Springer, 2012. to appear.
- Michel Couprie, David Coeurjolly, and Rita Zrour. Discrete bisector function and euclidean skeleton in 2d and 3d. *Image Vision Comput.*, 25(10):1543–1556, 2007.
- B. Curless and M. Levoy. **A volumetric method for building complex models from range images**. In *SigGraph-96*, pages 303–312, 1996.
- F. PrÃ©teux D. Perchet, C. Fetita. Advanced navigation tools for virtual bronchoscopy. In *Proceedings SPIE Conference on Image Processing: Algorithms and Systems III - IST, SPIE Symposium on Electronic Imaging, Science and Technology '04, San Jose, CA*, volume 5298, January 2004.
- Edward Roy Davies and A. P. N. Plummer. Thinning algorithms: A critique and a new methodology. *Pattern Recognition*, 14(1-6):53–63, 1981.
- James Davis, Steven R. Marschner, Matt Garr, and Marc Levoy. Filling holes in complex surfaces using volumetric diffusion. In *In First International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 428–438, 2001.
- F. de Vieilleville and J.-O. Lachaud. Convex shapes and convergence speed of discrete tangent estimators. *Lecture Notes in Computer Science*, 4292, 2006.
- François de Vieilleville. *Analyse des parties linéaires des objets discrets et estimateurs de caractéristiques géométriques*. PhD thesis, Université Bordeaux 1, 2007.
- François De Vieilleville and Jacques-Olivier Lachaud. Experimental comparison of continuous and discrete tangent estimators along digital curves. In *Proceedings of the 12th international conference on Combinatorial image analysis, IWCI'08*, pages 26–37, 2008.
- François de Vieilleville and Jacques-Olivier Lachaud. Comparison and improvement of tangent estimators on digital curves. *Pattern Recognition*, 42(8):1693–1707, 2009.
- François de Vieilleville, Jacques-Olivier Lachaud, and Fabien Feschet. Convex digital polygons, maximal digital straight segments and convergence of discrete geometric estimators. *Journal of Mathematical Imaging and Vision*, 27:139–156, 2007.

- I. Debled-Rennesson, F. Feschet, and J. Rouyer-Degli. Optimal blurred segments decomposition of noisy shapes in linear time. *Comput. Graph.*, 30:30–36, 2006.
- Isabelle Debled-Rennesson and Jean-Pierre Reveillès. A linear algorithm for segmentation of digital curves. *IJPRAI*, 9:635–662, 1995.
- Tamal K. Dey and Jian Sun. **Defining and computing curve-skeletons with medial geodesic function**. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, SGP '06, pages 143–152, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association. ISBN 3-905673-36-3.
- Gabriella Sanniti di Baja and Stina Svensson. A new shape descriptor for surfaces in 3d images. *Pattern Recognition Letters*, 23(6):703–711, 2002.
- Gabriella Sanniti di Baja and Edouard Thiel. Skeletonization algorithm running on path-based distance maps. *Image Vision Comput.*, 14(1):47–57, 1996.
- K. Do Yeon and P. Jong Won. Virtual angiography for diagnosis of carotid artery stenosis. *Journal of KISS: Software and Applications*, 30:821–828, 2003.
- L. Dorst and A.W.M. Smeulders. Length estimators for digitized contours. *Comput. Vision Graph. Image Process*, 40:311–333, 1987.
- M.-P. Dubuisson-Jolly and A.K. Jain. A modified hausdorff distance for object matching. *Proceedings of the 12th International Conference on Pattern Recognition*, 1:566–568, 1994.
- Ulrich Eckhardt. **A note on rutovitz' method for parallel thinning**. *Pattern Recognition Letters*, 8(1):35 – 38, 1988. ISSN 0167-8655.
- Ulrich Eckhardt and Gerd Maderlechner. Invariant thinning. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(5):1115–1144, 1993.
- M. Eden. A two-dimensional growth process. In *Proceedings of the 4th Berkeley Symposium on Mathematical Statistics and Probabilities*, 4:223–239, 1961.
- M. van Eede, D. Macrini, A. Telea, C. Sminchisescu, and S. S. Dickinson. Canonical skeletons for shape matching. In *Proceedings of the 18th International Conference on Pattern Recognition - Volume 02*, ICPR '06, pages 64–69, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2521-0.
- Henri-Alex Esbelin and Rémy Malgouyres. Convergence of binomial-based derivative estimation for c2noisy discretized curves. In *Proceedings of the 15th IAPR international conference on Discrete geometry for computer imagery*, DGCI'09, pages 57–66, 2009.
- Alexandre Faure, Lilian Buzer, and Fabien Feschet. Tangential cover for thick digital curves. *Pattern Recognition*, 42(10):2279–2287, 2009.

- Fabien Feschet. Multiscale analysis from 1d parametric geometric decomposition of shapes. In *ICPR*, pages 2102–2105, 2010.
- Fabien Feschet and Laure Tougne. Optimal time computation of the tangent of a discrete curve: Application to the curvature. In *DGCI*, pages 31–40, 1999.
- C.I. Fetita and F. Preteux. Quantitative 3d ct bronchography. *Proceedings IEEE International Symposium on Biomedical Imaging (ISBI'02)*, 2002.
- C. Fiorio, C. Mercat, and F. Rieux. Curvature estimation for discrete curves based on auto-adaptive masks of convolution. *Lectures Notes in Computer Science*, 6026:47–59, 2010.
- Oliver fleischmann, Lennart Wietzke, and Gerald Sommer. A novel curvature estimator for digital curves and images. In *Proceedings of the 32nd DAGM conference on Pattern recognition*, pages 442–451, 2010.
- Daniel S. Fritsch, Stephen M. Pizer, Bryan S. Morse, David H. Eberly, and Alan Liu. **The multiscale medial axis and its applications in image registration**. *Pattern Recogn. Lett.*, 15(5):445–452, May 1994. ISSN 0167-8655.
- R. Furukawa, T. Itano, A. Morisaka, and H. Kawasaki. **Improved space carving method for merging and interpolating multiple range images using information of light sources of active stereo**. In *ACCV*, pages II: 206–216, 2007.
- C. J. Gau and T.Y. Kong. Minimal non-simple sets in 4d binary images. *Graph. Models*, 65(3):112–130, 2003.
- Y Ge and J.M. Fitzpatrick. On the generation of skeletons from discrete euclidean distance maps. *Trans. on PAMI*, 18(11):1055–1066, 1996.
- G. Gerig, T. Kollera, G. Szekely, C. Brechbuhler, and O. Kubler. Symbolic descriptions of 3-d structures applied to cerebral vessel tree obtained from mr angiography volume data. *Lecture Notes in Computer Science*, 687:94–111, 1993.
- Peter Giblin and Benjamin B. Kimia. **A formal classification of 3d medial axis points and their local geometry**. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):238–251, January 2004. ISSN 0162-8828.
- Joachim Giesen, Balint Miklos, Mark Pauly, and Camille Wormser. The scale axis transform. In *Proceedings of the 25th annual symposium on Computational geometry, SCG '09*, pages 106–115, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-501-7.
- H.L. Gillis and K.R. Lutchen. How heterogeneous bronchoconstriction affects ventilation distribution in human lungs: a morphometric model. *Ann. Biomed. Eng.*, 27:1–13, 1999.

- Marcel J.E. Golay. Hexagonal parallel pattern transformations. *IEEE Trans. Comput.*, 18(8):733–740, 1969. ISSN 0018-9340.
- W. Gong and G. Bertrand. A simple parallel 3d thinning algorithm. In *10th International Conference on Pattern Recognition, 1990. Proceedings.*, pages 188–190, 1990.
- R. Gonzales and R. Woods. Digital image processing. *Reading, MA: Addison-Wesley*, 1992.
- M.W. Graham, J.D. Gibbs, and W.E. Higgins. Robust system for human airway-tree segmentation. *Medical Imaging 2008: Image Processing. Proceedings of the SPIE.*, 2008.
- H. Gray. Anatomy of the human body. *Philadelphia: Lea and Febiger*, 1918.
- S.B. Gray. Local properties of binary images in two dimensions. *IEEE Trans. Computers*, 20:551–561, 1971.
- T. Grigorishin, G. Abdel-Hamid, and Y. Yang. **Skeletonisation: An electrostatic field-based approach**. *Pattern Analysis and Applications*, 1:163–177, 1998. ISSN 1433-7541. 10.1007/BF01259366.
- Tong-Qiang Guo, Ji-Jun Li, Jian-Guang Weng, and Yue-Ting Zhuang. Filling holes in complex surfaces using oriented voxel diffusion. pages 4370–4375, aug. 2006.
- Zicheng Guo and Richard W. Hall. Fast fully parallel thinning algorithms. *CVGIP: Image Understanding*, 55(3):317–328, 1992. ISSN 1049-9660.
- R. W. Hall. Fast parallel thinning algorithms: parallel speed and connectivity preservation. *Communications of the ACM*, 32(1):124–131, 1989. ISSN 0001-0782.
- R.W. Hall. Tests for connectivity preservation for parallel reduction operators. *Topology and its Applications*, 46(3):199–217, 1992.
- Taosong He, Lichan Hong, Dongqing Chen, and Zhengrong Liang. **Reliable path for virtual endoscopy: Ensuring complete examination of human organs**. *IEEE Transactions on Visualization and Computer Graphics*, 7(4):333–342, October 2001. ISSN 1077-2626.
- Simon Hermann and Reinhard Klette. A comparative study on 2d curvature estimators. In *Proceedings of the International Conference on Computing: Theory and Applications*, ICCTA '07, pages 584–589, 2007.
- W.H. Hesselink and J.B.T.M. Roerdink. Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform. *IEEE Trans. on PAMI*, 30(12):2204–2217, 2008.

- Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, and Toshiyasu L. Kunii. **Topology matching for fully automatic similarity estimation of 3d shapes**. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques, SIGGRAPH '01*, pages 203–212, New York, NY, USA, 2001. ACM. ISBN 1-58113-374-X.
- T. Hildebrand and P. RÅ¼egsegger. A new method for the model-independent assessment of thickness in three-dimensional images. *Journal of Microscopy*, 185(1):67–75, 1997. ISSN 1365-2818.
- C.J. Hilditch. **Linear skeletons from square cupboards**. *Machine Intelligence*, 403, 1969.
- Tomio Hirata. A unified linear-time algorithm for computing distance maps. *Inf. Process. Lett.*, 58(3):129–133, 1996. ISSN 0020-0190.
- E.A. Hoffman, J.M. Reinhardt, M. Sonka, and et al. Characterization of the interstitial lung diseases via density based and texture-based analysis of computed tomography images of lung structure and function. *Acad. Radiology*, 10:1104–1118, 2003.
- K. Horsfield, G. Dart, D.E. Olson, G.F. Filley, and G. Cumming. Models of the human bronchial tree. *J. Appl. Physiol.*, 31:202–217, 1971.
- J. Hulin. *Axe Mdian Discret : Proprits Arithmtiques et Algorithmes*. PhD thesis, Universit Aix-Marseille II, Marseille, 2009.
- M.N. Huxley. Exponential sums and lattice points. *Proceedings of the London Mathematical Society*, 60:471–502, 1990.
- Marcin Janaszewski, Michel Couprie, and Laurent Babout. Geometric approach to hole segmentation and hole closing in 3d volumetric objects. In *CIARP '09: Proceedings of the 14th Iberoamerican Conference on Pattern Recognition*, pages 255–262, Berlin, Heidelberg, 2009a. Springer-Verlag. ISBN 978-3-642-10267-7.
- Marcin Janaszewski, Michel Couprie, and Laurent Babout. Hole filling in 3d volumetric objects. *Pattern Recognition*, 43(10):3548–3559, 2010.
- Marcin Janaszewski, Michal Postolski, and Laurent Babout. Robust algorithm for tunnel closing in 3d volumetric objects based on topological characteristics of points. *Pattern Recognition Letters*, 32(16):2231–2238, 2011.
- Marcin Janaszewski, Michal Postolski, Laurent Babout, and Edward Kacki. **Comparison of several centerline extraction algorithms for virtual colonoscopy**. In Edward Kacki, Marek Rudnicki, and Joanna Stempczynska, editors, *Computers in Medical Activity*, volume 65 of *Advances in Intelligent and Soft Computing*, pages 241–254. Springer Berlin Heidelberg, 2009b. ISBN 978-3-642-04461-8.
- Jiaya Jia and Chi-Keung Tang. **Inference of segmented color and texture description by tensor voting**. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):771–786, June 2004. ISSN 0162-8828.

- Guangxiang Jiang and Lixu Gu. **An automatic and fast centerline extraction algorithm for virtual colonoscopy**. *Conference Proceedings of the International Conference of IEEE Engineering in Medicine and Biology Society*, 5:5149–5152, 2005.
- A. Jonas and N. Kiryati. Digital representation schemes for 3d curves. *Pattern Recognition*, 30:1803–1816, 1997.
- Tao Ju. Fixing geometric errors on polygonal models: a survey. *J. Comput. Sci. Technol.*, 24(1):19–29, 2009. ISSN 1000-9000.
- Yongtae Jun. A piecewise hole filling algorithm in reverse engineering. *Computer-Aided Design*, 37(2):263–270, 2005.
- Kolja Kähler, Jörg Haber, Hitoshi Yamauchi, and Hans-Peter Seidel. **Head shop: generating animated head models with anatomical structure**. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '02, pages 55–63, New York, NY, USA, 2002. ACM. ISBN 1-58113-573-4.
- D.-G Kang and J.B. Ra. A new path planning algorithm for maximizing visibility in computed tomography colonography. *IEEE Transactions on Medical Imaging*, 24:957–968, 2005.
- P. Kardos, G. Nemeth, and K. Palagyi. Topology preserving 2-subfield 3d thinning algorithms. In *Proceedings of the IASTED International Conference on Signal Processing, Pattern Recognition and Applications*, pages 310–316, 2010.
- Sagi Katz and Ayellet Tal. **Hierarchical mesh decomposition using fuzzy clustering and cuts**. *ACM Trans. Graph.*, 22(3):954–961, July 2003. ISSN 0730-0301.
- N.D. Kehtarnavaz and R.J.P. de Figueiredo. A 3-d contour segmentation scheme based on curvature and torsion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10:707–713, 1988.
- B. Kerautret and J. O. Lachaud. Curvature estimation along noisy digital contours by approximate global optimization. *Pattern Recogn.*, 42(10):2265–2278, 2009.
- B. Kerautret, J.-O. Lachaud, and B. Naegel. Comparison of discrete curvature estimators and application to corner detection. In *Proceedings of the 4th International Symposium on Advances in Visual Computing*, ISVC '08, pages 710–719, 2008.
- Bertrand Kerautret and Jacques-Olivier Lachaud. Meaningful scales detection along digital contours for unsupervised local noise estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34:2379–2392, 2012.
- Ron Kimmel, Doron Shaked, Nahum Kiryati, and Alfred M. Bruckstein. Skeletonization via distance maps and level sets. *Computer Vision and Image Understanding*, 62:382–391, 1995.

- G.G. King, N.L. Muller, K.P. Whittall, Q.S. Xiang, and P.D. Pare. An analysis algorithm for measuring airway lumen and wall areas from high-resolution computed tomographic data. *Am. J. R. Cr. Care Med*, 161:574–580, 2000.
- N. Kiryati and O. Kubler. On chain code probabilities and length estimators for digitized three dimensional curves. *Pattern Recognition*, pages 259–262, 1992.
- H. Kitaoka, Y. Park, J. Tschirren, J.M. Reinhardt, M. Sonka, G. McLennan, and E.A. Hoffman. Automated nomenclature labeling of the bronchial tree in 3d-ct lung images. *MICCAI 2002, Tokyo, Japan*, pages 1–11, 2002.
- H. Kitaoka, R. Takaki, and B. Suki. A three-dimensional model of the human airway tree. *J. Appl. Physiol.*, 87:2207–2217, 1999.
- R. Klette and A. Rosenfeld. Digital geometry - geometric methods for digital picture analysis. *Morgan Kaufman, San Francisco*, 2004.
- Reinhard Klette and Joviša Žunić. Multigrid convergence of calculated features in image analysis. *J. Math. Imaging Vis.*, 13:173–191, 2000.
- T. Yung Kong and Azriel Rosenfeld. Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing*, 48(3):357–393, 1989.
- T.Y. Kong. Problem of determining whether a parallel reduction operator for n-dimensional binary images always preserves topology. *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, volume 2060 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 69–77, 1993.
- T.Y. Kong. On topology preservation in 2-d and 3-d thinning. *International journal of pattern recognition and artificial intelligence*, 9:813–844, 1995.
- T.Y. Kong and C.J. Gau. Minimal non-simple sets in 4-dimensional binary images with (8,80)-adjacency. *In Lecture notes in computer science*, pages 318–333, 2004.
- V. Kovalevsky and S. Fuchs. Theoretical and experimental analysis of the accuracy of perimeter estimates. *Proceedings of Robust Computer Vision*, pages 218–242, 1992.
- V. A. Kovalevsky. Finite topology as applied to image analysis. In *Computer Vision, Graphics, and Image Processing*, pages 141–161, 1989.
- K. Krissian, G. Malandain, N. Ayache, R. Vaillant, and Y. Troussset. Model based detection of tubular structures in 3d images. *Computer Vision Image Understanding*, 80:130–171, 2000.
- Z. Kulpa. Area and perimeter measurement of blobs in discrete binary pictures. *Comput. Graph. Image Process.*, 6:434–451, 1977.

- Amitesh Kumar, Alan M. Shih, Yasushi Ito, Douglas H. Ross, and Bharat K. Soni. A hole-filling algorithm using non-uniform rational b-splines. In Michael L. Brewer and David L. Marcum, editors, *IMR*, pages 169–182. Springer, 2007. ISBN 978-3-540-75102-1.
- J.-O. Lachaud and A. Vialard. Geometric measures on arbitrary dimensional digital surfaces. In *Lectures Notes in Computer Science*, DGCI'03, pages 434–443, 2003.
- Jacques-Olivier Lachaud, Anne Vialard, and François de Vieilleville. Analysis and comparative evaluation of discrete tangent estimators. In *DGCI*, pages 240–251, 2005a.
- Jacques-Olivier Lachaud, Anne Vialard, and François de Vieilleville. Analysis and comparative evaluation of discrete tangent estimators. In *Proceedings of the 12th international conference on Discrete Geometry for Computer Imagery*, DGCI'05, pages 240–251, 2005b.
- Jacques-Olivier Lachaud, Anne Vialard, and François de Vieilleville. Fast, accurate and convergent tangent estimation on digital contours. *Image Vision Comput.*, 25(10): 1572–1587, 2007.
- Louisa Lam, Seong-Whan Lee, and Ching Y. Suen. Thinning methodologies - a comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 869–885, 1992.
- TY. Law and P.A. Heng. Automated extraction of bronchus from 3d ct images of lung based on genetic algorithm and 3d region growing. *SPIE Proceedings on Medical Imaging*, pages 906–916, 2000.
- Francis Lazarus and Anne Verroust. Metamorphosis of cylinder-like objects. *Journal of Visualization and Computer Animation*, 8(3):131–146, 1997.
- T.C. Lee, R.L. Kashyap, and C.N. Chu. Building skeleton models via 3-d medial surface/axis thinning algorithms. *Graphical Models and Image Processing*, 56(6):462–478, 1994.
- Alexandre Lenoir, Raphaël Malgouyres, and Marinette Revenu. Fast computation of the normal vector field of the surface of a 3-d discrete object. In *Discrete Geometry for Computer Imagery*, volume 1176 of *Lecture Notes in Computer Science*, pages 101–112. 1996.
- Bruno Lévy. **Dual domain extrapolation**. *ACM Trans. Graph.*, 22(3):364–369, July 2003. ISSN 0730-0301.
- T. Lewiner, J.D. Gomes Jr., H. Lopes, and M. Craizer. Curvature and torsion estimators based on parametric curve fitting. *Computers and Graphics*, 29:641–655, 2005.
- Peter Liepa. Filling holes in meshes. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 200–205, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. ISBN 1-58113-687-0.

- André Lieutier. **Any open bounded subset of \mathbb{R}^n has the same homotopy type than its medial axis.** In *Proceedings of the eighth ACM symposium on Solid modeling and applications*, SM '03, pages 65–75, New York, NY, USA, 2003. ACM. ISBN 1-58113-706-0.
- C. Lohou and G. Bertrand. A 3d 12-subiteration thinning algorithm based on p-simple points. *Discrete Applied Mathematics*, 139:171–195, 2004a.
- C. Lohou and G. Bertrand. A 3d 12-subiteration thinning algorithm based on p-simple points. *Discrete Appl. Math.*, 139(1):171–195, 2004b.
- C. Lohou and G. Bertrand. A 3d 6-subiteration curve thinning algorithm based on p-simple points. *Discrete Applied Mathematics*, 151:198–228, 2005a.
- C. Lohou and G. Bertrand. A 3d 6-subiteration curve thinning algorithm based on p-simple points. *Discrete Appl. Math.*, 151(1):198–228, 2005b.
- Christophe Lohou and Bruno Miguel. Detection of the Aortic Intimal Tears by using 3D Digital Topology. In Beraldin, JA and Cheok, GS and McCarthy, MB and NeuschaeferRube, U and Baskurt, AM and McDowall, IE and Dolinsky, M, editor, *THREE-DIMENSIONAL IMAGING, INTERACTION, AND MEASUREMENT*, volume 7864 of *Proceedings of SPIE-The International Society for Optical Engineering*. Soc Imaging Sci & Technol; SPIE, SPIE-INT SOC OPTICAL ENGINEERING, 2011. Conference on the Three-Dimensional Imaging, Interaction, and Measurement, San Francisco, CA, JAN 24-27, 2011.
- William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. volume 21, pages 163–169, July 1987.
- E. Cohen L.S. Tekumalla. A hole-filling algorithm for triangular meshes. technical report uucs-04-019. Technical report, University of Utah, 2004.
- C.M. Ma. On topology preservation in 3d thinning. *CVGIP: Image Understanding*, 59(3):328–339, 1994.
- C.M. Ma and S.Y. Wan. Parallel thinning algorithms on 3d (18, 6) binary images. *Computer Vision and Image Understanding*, 80(3):364–378, 2000.
- C.M. Ma and S.Y. Wan. A medial-surface oriented 3-d two-subfield thinning algorithm. *Pattern Recognition Letters*, 22(13):1439–1446, 2001.
- C.M. Ma, S.Y. Wan, and H.K. Chang. Extracting medial curves on 3d images. *Pattern Recognition Letters*, 23(8):895–904, 2002a.
- C.M. Ma, S.Y. Wan, and et al. Three-dimensional topology preserving reduction on the 4-subfields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1594–1605, 2002b.

- M. Maddah, A. Afzali-Kusha, and H. Soltanian-Zadeh. Efficient center-line extraction for quantification of vessels in confocal microscopy images. *Med. Phys.*, 30:204–211, 2003.
- I. Mäkelä and A. Dolenc. *Some Efficient Procedures for Correcting Triangulated Models*. Helsinki University of Technology, 1993. ISBN 9789512217335.
- Grégoire Malandain and Sara Fernández-Vidal. Euclidean skeletons. *Image Vision Comput.*, 16(5):317–327, 1998.
- Rémy Malgouyres, florent Brunet, and Sébastien Fourey. Binomial convolutions and derivatives estimation from noisy discretizations. In *Proceedings of the 14th IAPR international conference on Discrete geometry for computer imagery*, DGCI'08, pages 370–379, 2008.
- A. Manzanera, T.M. Bernard, F. Preteux, and B. Longuet. Medial faces from a concise 3d thinning algorithm. *Computer Vision, IEEE International Conference on*, 1:337, 1999.
- M. Marji. *On the detection of dominant points on digital planar curves*. PhD thesis, Wayne State University, Detroit, Michigan, 2003.
- T.J. Marrow, L. Babout, A.P. Jivkov, P. Wood, D. Engelberg, N. Stevens, P.J. Withers, and R.C. Newman. **Three dimensional observations and modelling of intergranular stress corrosion cracking in austenitic stainless steel**. *Journal of Nuclear Materials*, 352(1-3):62 – 74, 2006. ISSN 0022-3115. Proceedings of the E-MRS 2005 Spring Meeting Symposium N on Nuclear Materials (including the 10th Inert Matrix Fuel Workshop).
- T.B. Martonen, Y. Yang, and M. Dolovich. Definition of airway composition within gamma camera images. *J. Thorac. Imaging*, 9:188–197, 1994.
- Jiri Matas, Z. Shao, and Josef Kittler. Estimation of curvature and tangent direction by median filtered differencing. In *ICIAP*, pages 83–88, 1995.
- G. Matheron. *Examples of topological properties of skeletons*, volume 2, pages 217–238. Academic Press, 1988.
- S. Matsuoka, Y. Kurihara, Y. Nakajima, H. Niimi, H. Ashida, and K. Kaneoya. Serial change in airway lumen and wall thickness at thin-section ct in asymptomatic subjects. *Radiology*, 234:595–603, 2005.
- SA McDonald, N Ravirala, PJ Withers, and A Alderson. In situ three-dimensional x-ray microtomography of an auxetic foam under tension. *Scripta Materialia*, 60:232–235, 2009.
- R. Medina, A. Wahle, M.E. Olszewski, and M. Sonka. Curvature and torsion estimation for coronary-artery motto analysis. *SPIE Medical Imaging*, 5369:504–515, 2004.

- A. Meijster, J.B.T.M. Roerdink, and W.H. Hesselink. A general algorithm for computing distance transforms in linear time. In *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 331–340, 2000.
- F. Meyer. *Cytologie quantitative et morphologie mathématique*. PhD thesis, École des mines de Paris, 1979.
- Balint Miklos, Joachim Giesen, and Mark Pauly. Discrete scale axis representations for 3d geometry. *ACM Trans. Graph.*, 29:101:1–101:10, July 2010. ISSN 0730-0301.
- F. Mokhtarian. A theory of multiscale, torsion-based shape representation for space curves. *Computer Vision and Image Understanding*, 68:1–17, 1997.
- F Mokhtarian and A Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8:34–43, 1986.
- Claudio Montani, Riccardo Scateni, and Roberto Scopigno. **A modified look-up table for implicit disambiguation of marching cubes**. *The Visual Computer*, 10(6):353–355, 1994.
- K. Mori and et al. Automated extraction and visualization of bronchus from 3d ct images of lung. *Proc. of 1st International Conferance on Computer Vision, Virtual Reality and Robotic in Medicine (CVRMed'95)*, pages 542–548, 1995.
- K. Mori, J. Hasegawa, Y. Suenaga, and J. Toriwaki. Automated anatomical labeling of the bronchial branch and its application to the virtual bronchoscopy system. *IEEE Trans. Medical Imaging*, 19:103–114, 2000.
- K. Mori, J. Hasegawa, J. Toriwaki, H. Anno, and K. Katada. A fast rendering method using the tree structure of objects in virtualized bronchus endoscope system. *Lecture Notes in Computer Science*, 1131:33–42, 1996.
- J. Mukherjee, P.P. Das, and B.N. Chatterji. On connectivity issues of espta. *Pattern Recognition Letters*, 80(3):364–378, 2000.
- CJ. Murray and AD. Lopez. Alternative projections of mortality and disability by cause 1990-2020: Global burden of disease study. *Lancet*, 349:1498–1504, 1997.
- M. Näf, G. Székely, R. Kikinis, M. E. Shenton, and O. Kübler. 3d voronoi skeletons and their usage for the characterization and recognition of 3d organ shape. *Comput. Vis. Image Underst.*, 66(2):147–161, 1997. ISSN 1077-3142.
- D. Nain, S. Haker, R. Kikinis, and W..E.L.. Grimson. An interactive virtual endoscopy tool. pages 4–, 10 2001.
- T.R. Nelson and D.K. Manchester. Modeling of lung morphogenesis using fractal geometries. *IEEE Trans. Med. Imag.*, 7:321–327, 1988.

- G. Nemeth, P. Kardos, and K. Palagyi. Topology preserving 3d thinning algorithms using four and eight subfields. *Image Analysis and Recognition*, pages 316–325, 2010.
- Gregory M. Nielson and Bernd Hamann. **The asymptotic decider: resolving the ambiguity in marching cubes**. In *Proceedings of the 2nd conference on Visualization '91, VIS '91*, pages 83–91, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press. ISBN 0-8186-2245-8.
- I. Nystrom. Skeletonization applied to magnetic resonance angiography images. *SPIE*, 3338:693–701, 2003.
- R. Ogniewicz. A multiscale MAT from Voronoi diagrams: The skeleton-space and its application to shape description and decomposition. In C. Arcelli, L.P. Cordella, and G. S. di Baja, editors, *Aspects of Visual Form Processing*, 2nd International Workshop on Visual Form, pages 430–439, Singapore, May-June 1994. IAPR, World Scientific. Workshop held in Capri, Italy.
- R. Ogniewicz and M. Ilg. Voronoi skeletons: Theory and applications. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 63–69, Champaign, Illinois, June 1992.
- R.L. Ogniewicz and O. Kübler. Hierarchic voronoi skeletons. *Pattern Recognition*, 28(33):343–359, 1995.
- K. Palagyi. A 3-subiteration 3d thinning algorithm for extracting medial surfaces. *Pattern Recognition Letters*, 23(6):663–675, 2002.
- K. Palagyi. A 3-subiteration surface-thinning algorithm. In *Computer Analysis of Images and Patterns*, pages 628–635, 2007a.
- K. Palagyi. A subiteration-based surface-thinning algorithm with a period of three. *Pattern Recognition*, pages 294–303, 2007b.
- K. Palagyi and A. Kuba. A 3d 6-subiteration thinning algorithm for extracting medial lines. *Pattern Recognition Letters*, 19(7):613–627, 1998.
- K. Palagyi and A. Kuba. Directional 3d thinning using 8 subiterations. In *Discrete Geometry for Computer Imagery*, pages 325–336, 1999a.
- K. Palagyi and A. Kuba. A parallel 3d 12-subiteration thinning algorithm. *Graphical Models and Image Processing*, 61(4):199–221, 1999b.
- K. Palagyi, J. Tschirren, E.A. Hoffman, and M. Sonka. Quantitative analysis of pulmonary airway tree structure. *Computers in Biology and Medicine*, 36:974–996, 2006.
- Kálmán Palágyi. A 3-subiteration 3d thinning algorithm for extracting medial surfaces. *Pattern Recognition Letters*, 23(6):663 – 675, 2002. ISSN 0167-8655.

- Kálmán Palágyi. A 3d fully parallel surface-thinning algorithm. *Theor. Comput. Sci.*, 406(1-2):119–135, 2008. ISSN 0304-3975.
- Seyoun Park, Xiaohu Guo, Hayong Shin, and Hong Qin. **Surface completion for shape and appearance**. *Vis. Comput.*, 22(3):168–180, March 2006. ISSN 0178-2789.
- W. Park, E.A. Hoffman, and M. Sonka. Segmentation of intrathoracic airway trees: a fuzzy logic approach. *Medical Imaging, IEEE Transactions*, 17:489–497, 1998.
- Paresh S. Patel, David L. Marcum, and Michael G. Remotigue. Stitching and filling: Creating conformal faceted geometry. In Byron W. Hanks, editor, *Proceedings of the 14th International Meshing Roundtable, IMR 2005, September 11-14, 2005, San Diego, CA, USA*, pages 239–256. Springer, 2005. ISBN 978-3-540-25137-8.
- Mark Pauly, Niloy J. Mitra, Joachim Giesen, Markus Gross, and Leonidas J. Guibas. Example-based 3d scan completion. In *SGP '05: Proceedings of the third Eurographics symposium on Geometry processing*, page 23, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association. ISBN 3-905673-24-X.
- Mark Pauly, Niloy J. Mitra, Johannes Wallner, Helmut Pottmann, and Leonidas J. Guibas. **Discovering structural regularity in 3d geometry**. *ACM Trans. Graph.*, 27(3):43:1–43:11, August 2008. ISSN 0730-0301.
- RA. Pauwels and KF. Rabe. Burden and clinical features of chronic obstructive pulmonary disease (copd). *Lancet*, 364:613–620, 2004.
- T. Pavlidis. A thinning algorithm for discrete binary images. *Computer Graphics and Image Processing*, 13(2):142–157, 1980.
- T. Pavlidis. A flexible parallel thinning algorithm. In *Pattern Recognition and Image Processing*, pages 162–167, 1981.
- Jean-Philippe Pernot, George florin Moraru, and Philippe Véron. **Filling holes in meshes using a mechanical model to simulate the curvature variation minimization**. *Computers & Graphics*, 30(6):892–902, 2006.
- C. Pisupati, L. Wolf, W. Mitzner, and E. Zerhouni. Segmentation of 3d pulmonary trees using mathematical morphology. *Mathematical morphology and its applications to image and signal processing*, pages 406–416, 1996.
- C. Pisupati, L. Wolff, W. Mitzner, and E. Zerhouni. A central axis algorithm for 3d bronchial tree structures. *Proceedings of International Symposium on Computer Vision - ISCV*, pages 259–264, 1995.
- Stephen M. Pizer, Daniel S. Fritsch, Paul A. Yushkevich, Valen E. Johnson, and Edward L. Chaney. Segmentation, registration and measurement of shape variation via image object shape. *IEEE Trans. Med. Imaging*, pages 851–865, 1999.

- William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. flannery. *Numerical recipes in C (2nd ed.): the art of scientific computing*. Cambridge University Press, New York, NY, USA, 1992.
- F. Preteux, C.I. Fetita, P. Grenier, and A. Capderou. Modeling, segmentation, and caliber estimation of bronchi in high-resolution computerized tomography. *Journal of Electronic Imaging*, 8:36–45, 1998.
- C. Pudney. Distance-ordered homotopic thinning: a skeletonization algorithm for 3D digital images. *Computer Vision and Image Understanding*, 72(3):404–413, 1998.
- O.G. Raabe, H.C. Yeh, H.M. Schum, and R.F. Phalen. Tracheobronchial geometry-human, dog, rat, hamster. albuquerque. *NM: Inhalation Toxicology Research Institute, Lovelace Foundation for Medical Evaluation and Research. Report LF-53*, 1976.
- B.K. Ray and R. Pandyan. An adaptive corner detector for planar curves. *Pattern recognition*, 36:703–708, 2003.
- Benjamin Raynal. *Applications of Digital Topology For Real-Time Markerless Motion Capture*. PhD thesis, Université Paris-Est, December 2010.
- E. Remy and E. Thiel. Look-Up Tables for Medial Axis on Squared Euclidean Distance Transform. In *11th DGCI, Discrete Geometry for Computer Image*, volume 2886 of *Lectures Notes in Computer Science*, pages 224–235, Naples, Italy, Nov 2003.
- Eric Remy and Edouard Thiel. Exact medial axis with euclidean distance. *Image Vision Comput.*, 23(2):167–175, 2005.
- J.-P. Réveilles. Géométrie discrete, calcul en nombres entiers et algorithmique. thèse d'état. *Université Louis Pasteur, Strasbourg*, 1991.
- Christian Ronse. Minimal test patterns for connectivity preservation in parallel thinning algorithms for binary digital images. *Discrete Appl. Math.*, 21(1):67–79, 1988. ISSN 0166-218X.
- A. Rosenfeld. *Digital image processing*. Academic Press, 1982.
- Azriel Rosenfeld. Connectivity in digital pictures. *J. ACM*, 17(1):146–160, 1970.
- Azriel Rosenfeld. Digital topology. *The American Mathematical Monthly*, 86(8):621–630, 1979.
- Tristan Roussillon and Jacques-Olivier Lachaud. Accurate curvature estimation along digital contours with maximal digital circular arcs. In *Proceedings of the 14th international conference on Combinatorial image analysis, IWCIA'11*, pages 43–55, 2011.
- Martin Rumpf and Alexandru Telea. A continuous skeletonization method based on level sets. In *VISSYM '02: Proceedings of the symposium on Data Visualisation 2002*, pages

- 151–ff, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association. ISBN 1-58113-536-X.
- D. Rutovitz. Pattern recognition. *Journal of the Royal Statistical Society. Series A (General)*, 129(4):504–530, 1966.
- R. Sagawa and K. Ikeuchi. **Hole filling of a 3D model by flipping signs of a signed distance field in adaptive resolution.** *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(4):686–699, April 2008.
- Ryusuke Sagawa and Katsushi Ikeuchi. **Taking consensus of signed distance field for complementing unobservable surface.** In *3DIM*, pages 410–417. IEEE Computer Society, 2003. ISBN 0-7695-1991-1.
- P. K. Saha and B. B. Chaudhuri. 3d digital topology under binary transformation with applications. *Comput. Vis. Image Underst.*, 63(3):418–429, May 1996. ISSN 1077-3142.
- T. Saito and J.-I. Toriwaki. New algorithms for euclidean distance transformation of an n-dimensional digitized picture with applications. *Pattern Recognition*, 27(11):1551–1565, 1994.
- K. Santilli, K. Bemis, D. Silver, J. Dastur, and P. Rona. **Generating realistic images from hydrothermal plume data.** In *Proceedings of the conference on Visualization '04, VIS '04*, pages 91–98, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7803-8788-0.
- A. Saragaglia, C.I. Fetita, and F. Preteux. Assessment of airway remodeling in asthma: volumetric versus surface quantification approaches. *Lectures Notes in Computer Science*, 4191:413–420, 2006.
- André Vital Saúde. New reduced discrete euclidean nd medial axis with optimal algorithm. *Pattern Recognition Letters*, 32(1):91–99, 2011.
- André Vital Saúde, Michel Couprie, and Roberto de Alencar Lotufo. Exact euclidean medial axis in higher resolution. In *DGCI*, pages 605–616, 2006.
- Vladimir Savchenko and Nikita Kojekine. An approach to blend surfaces. In *Advances in Modeling, Animation and Rendering*, pages 139–150. Springer, 2002.
- P.M. Scott. Stress corrosion cracking in pressurized water reactors-interpretation, modeling and remedies. *Corrosion*, 56:771–782, 2000.
- J. Serra. *Image analysis and mathematical morphology*. Academic Press, 1982.
- D. Shaked and A. Bruckstein. Pruning medial axes. *Comput. Vision Image Understanding*, 69:156–169, 1998.

- Andrei Sharf, Marc Alexa, and Daniel Cohen-Or. Context-based surface completion. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 878–887, New York, NY, USA, 2004. ACM.
- D. Sharvit, J. Chan, H. Tek, and B.B. Kimia. Symmetry-based indexing of image databases. *Journal of Visual Communication and Image Representation*, 9(4):366–380, 1998.
- K. Siddiqi, S. Bouix, A. Tannenbaum, and S. Zucker. The hamilton-jacobi skeleton. In *International Conference on Computer Vision (ICCV)*, pages 828–834, 1999.
- F. Sloboda, B. Zacko, and J. Stoer. On approximation of planar one-dimensional continua. *Proceedings of Advances in Digital and Computational Geometry*, pages 113–160, 1998.
- P. Soille. *Morphological image analysis*. Springer-Verlag, 1999.
- M. Sonka, W. Park, and E.A. Hoffman. Rule-based detection of intrathoracic airway trees. *Computer. Med. Imag. Graph.*, 15:314–326, 1996.
- R. Stefanelli and A. Rosenfeld. Some parallel thinning algorithms for digital pictures. *J. ACM*, 18(2):255–264, 1971.
- I. Stojmenovic and R. Tomic. Digitization schemes and the recognition of digital straight lines, hyperplanes and flats in arbitrary dimensions. *Vision Geometry, contemporary Mathematics Series*, 119:197–212, 1991.
- H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. **Skeleton based shape matching and retrieval**. In *Proceedings of the Shape Modeling International 2003*, SMI '03, pages 130–, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1909-1.
- S. Svensson and GS. di Baja. Simplifying curve skeletons in volume images. *Comput. Vision Image Understanding*, 90:242–257, 2003.
- Stina Svensson, Ingela Nyström, and Gabriella Sanniti di Baja. Curve skeletonization of surface-like objects in 3d images guided by voxel classification. *Pattern Recognition Letters*, 23(12):1419–1426, 2002.
- M. Tajine and A. Daurat. On local definitions of length of digital curves. *Lecture Notes in Computer Science*, 2886, 2003.
- H. Talbot and L. Vincent. Euclidean skeletons and conditional bisectors. In *Proceedings of VCIP'92, SPIE*, volume 1818, pages 862–876, 1992.
- M. Howatson Tawhai, A.J. Pullan, and P.J. Hunter. Generation of an anatomically based three-dimensional model of the conducting airways. *Annals of Biomedical Engineering*, 28:793–802, 2000.

- Alexandru Telea and Anna Vilanova. A robust level-set algorithm for centerline extraction. In *Proceedings of the symposium on Data visualisation 2003*, VISSYM '03, pages 185–194, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. ISBN 1-58113-698-6.
- J. Toriwaki and K. Mori. Distance transformation and skeletonization of 3d pictures and their applications to medical images. *Lecture Notes in Computer Science*, 2243: 412–429, 2001a.
- Jun-ichiro Toriwaki and Kensaku Mori. Distance transformation and skeletonization of 3d pictures and their applications to medical images. In *Digital and Image Geometry, Advanced Lectures [based on a winter school held at Dagstuhl Castle, Germany in December 2000]*, pages 412–428, London, UK, 2001b. Springer-Verlag. ISBN 3-540-43079-2.
- T. Tozaki, Y. Kawata, N. Niki, and et al. Pulmonary organs analysis for differential diagnosis based on thoracic thin-section ct images. *Nuclear Science, IEEE Transaction*, 45:3075–3082, 1998.
- J. Tschirren, E.A Hoffman, G. McLennan, and M. Sonka. Intrathoracic airway trees: Segmentation and airway morphology analysis from low-dose ct scans. *Medical Imaging, IEEE Transactions*, 24(12):1529–1539, 2005a.
- J. Tschirren, G. McLennan, K. Palagyi, E.A. Hoffman, and M. Sonka. Matching and anatomical labeling of human airway tree. *IEEE Trans Med Imaging*, 24:1540–1547, 2005b.
- C. van Ertbruggen, C. Hirsch, and M. Paiva. Anatomically based three dimensional model of airways to simulate flow and particle transport using computational fluid dynamics. *J. Appl. Physiol.*, 98:970–980, 2005.
- Allen van Gelder and Jane Wilhelms. **Topological considerations in isosurface generation.** *ACM Trans. Graph.*, 13(4):337–375, October 1994. ISSN 0730-0301.
- Michal Varnuska, J. Parus, and Ivana Kolingerova. Simple holes triangulation in surface reconstruction. In *Proceedings of Algorithmy*, pages 280–289, 2005.
- J. Verdera, V. Caselles, M. Bertalmio, and G. Sapiro. Inpainting surface holes. In *In Int. Conference on Image Processing*, pages 903–906, 2003.
- Anne Vialard. Geometrical parameters extraction from discrete paths. In *Proceedings of the 6th International Workshop on Discrete Geometry for Computer Imagery*, DCGA '96, pages 24–35, 1996a.
- Anne Vialard. Geometrical parameters extraction from discrete paths. In *Proceedings of the 6th International Workshop on Discrete Geometry for Computer Imagery*, DCGA '96, pages 24–35, 1996b.

- L. Vincent. Efficient computation of various types of skeletons. In *Proceedings of Medical Imaging V, SPIE*, volume 1445, pages 297–311, 1991.
- Benjamin Vrolijk, Freek Reinders, and Frits H. Post. Feature tracking with skeleton graphs. In *Data Visualization: The State of the Art*, pages 37–52, 2003.
- Marc. Wagner, Ulf. Labsik, and Gunther Greiner. Repairing non-manifold triangle meshes using simulated annealing. *International Journal of Shape Modeling*, 9(2): 137–153, 2003.
- Ming Wan, Frank Dachille, and Arie Kaufman. **Distance-field based skeletons for virtual navigation**. In *Proceedings of the conference on Visualization '01, VIS '01*, pages 239–246, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7803-7200-X.
- Ming Wan Ming Wan, Zhengrong Liang Zhengrong Liang, Qi Ke Qi Ke, Lichan Hong Lichan Hong, I Bitter, and A Kaufman. **Automatic centerline extraction for virtual colonoscopy**. *IEEE Transactions on Medical Imaging*, 21(12):1450–1460, 2002.
- S. Wan, A. Kiraly, E. Ritman, and W. Higgins. Extraction of the hepatic vasculature in rats using 3-d micro-ct images. *IEEE Trans. Medical Imaging*, 19:964–971, 2000.
- Gianning Wang and Manuel M. Oliveira. Filling holes on locally smooth surfaces reconstructed from point clouds. *Image Vision Comput.*, 25(1):103–113, 2007.
- Rainer Wegenkittl, Anna Vilanova i Bartrolí, Balint Hegedüs, Daniel Wagner, Martin C. Freund, and Meister Eduard Gröller. **Mastering interactive virtual bronchoscopy on a low-end pc**. (TR-186-2-00-13), March 2000.
- E.R. Weibel. Morphometry of the human lung. *New York: Academic*, 1963.
- J.H.C. Whitehead. Simplicial spaces, nuclei and m-groups. *Proceedings of the London Mathematical Society*, 45:243–327, 1939.
- R. Wiemker, T. Blaffert, T. Blow, S. Renisch, and C. Lorenz. Automated assessment of bronchial lumen, wall thickness and bronchial tree using high-resolution ct. *International Congress Series*, 1268:967–972, 2004.
- A.P. Witkin. Scale-space filtering. *8th International Joint Conference on Artificial Intelligence*, 2, 1983.
- S. Wood, A. Zerhouni, J. Hoford, E. Hoffman, and W. Mitzner. Measurement of three-dimensional lung tree structures using computed tomography. *J. Appl. Physiol.*, 79: 1687–1697, 1995.
- M. Worring and A.W.M. Smeulders. Digital curvature estimation. *CVGIP-Image Understanding*, 58:366–382, 1993.

- Chunxia Xiao, Wenting Zheng, Yongwei Miao, Yong Zhao, and Qunsheng Peng. **A unified method for appearance and geometry completion of point set surfaces**. 2007. ISSN 1432-2315.
- W. Xie, R.P. Thompson, and R. Perucchio. A topology-preserving parallel 3d thinning algorithm for extracting the curve skeleton. *Pattern Recognition*, 36(7):1529–1544, 2003.
- Songhua Xu, Athinodoros Georghiades, Holly Rushmeier, Julie Dorsey, and Leonard McMillan. **Image guided geometry inference**. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, 3DPVT '06, pages 310–317, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2825-2.
- L. Yang and F. Albrechtsen. Discrete green's theorem and its application in moment computation. In *Proc. 1st Int. Conf. Electronics and Information Technology*, page 2731, 1994a.
- L. Yang and F. Albrechtsen. Fast computation of invariant geometric moments: a new method giving correct results. In *Proc. 12th Int. Conf. Pattern Recognition*, 1994b.
- L. Yang and F. Albrechtsen. Fast and exact computation of cartesian geometric moments using discrete green's theorem. *Pattern Recognition*, 29:1061–1073, 1996.
- L. Yang, F. Albrechtsen, T. Lonnestad, and P. Grottum. Methods to estimate areas and perimeters of blob-like objects: A comparison. In *In Proc. IAPR Workshop on Machine Vision Applications*, pages 272–276, 1994.
- K-C. Yu, E.L. Ritman, and W.E. Higgins. System for the analysis and visualization of large 3d anatomical trees. *Computer. Biol. Med.*, 37:1802–1820, 2007.
- Huang Yunbao, Wang Qifu, Huang Zhengdong, and Wu Jianjie. **Repairing incomplete measured data with a deformable model under constraints of feature shapes**. *The International Journal of Advanced Manufacturing Technology*, 28(1):92–100, 2006.
- Hong-Kai Zhao, Stanley Osher, and Ronald Fedkiw. Fast surface reconstruction using the level set method. In *VLSM '01: Proceedings of the IEEE Workshop on Variational and Level Set Methods (VLSM'01)*, page 194, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-1278-X.
- Wei Zhao, Shuming Gao, and Hongwei Lin. A robust hole-filling algorithm for triangular mesh. *The Visual Computer*, 23(12):987–997, 2007.
- Yong Zhou and Arthur W. Toga. Efficient skeletonization of volumetric objects. *IEEE Trans. Vis. Comput. Graph.*, 5(3):196–209, 1999.