CENTRE FOR **STOCHASTIC GEOMETRY**
AND ADVANCED **BIOIMAGING**

RECONSTRUCTION OF TOPOLOGY AND GEOMETRY FROM DIGITISATIONS

PhD Thesis by
Sabrina Tang Christensen

Reconstruction of Topology and Geometry from Digitisations

PhD thesis by Sabrina Tang Christensen
Department of Mathematics, Aarhus University,
Centre for Stochastic Geometry and Advanced Bioimaging,
Ny Munkegade 118, 8000 Aarhus C, Denmark

Supervised by Andrew A. du Plessis

Submitted 9 August 2016

Cover art: DesignZone, Daniela Mayer, daniela@designzone.info

ENGLISH RÉSUMÉ

Elucidation of the relation between real world objects and their digitisations is an inherent challenge of image analysis. This thesis addresses the reconstruction of topological as well as geometrical features of sets in Euclidean 3-space respectively the Euclidean plane from binary images.

The first problem, the reconstruction of topology in dimension three, is approached using combinatorics of voxel reconstructions in combination with differential topology results. An improved digital reconstruction based on binary images of objects with sufficiently smooth boundary is proposed. It is shown that this reconstruction is ambient isotopic to the underlying object provided the resolution of the digitisation be sufficiently high and under certain assumptions on the classical voxel reconstruction. The exact lower bound on the resolution, related to the curvature of the boundary of the object, is given, and it is argued that this is the best possible bound for which topological equivalence is guaranteed. Making no restrictions on the classical reconstruction, similar results can be proved for slightly stronger assumptions on the resolution. It is conjectured that the lower of the bounds on the resolution also suffices in this setting.

For the reconstruction of geometry in dimension two, a practical approach employing digital algorithms is selected. The geometry of real world objects can be described by Minkowski tensors. It has been shown that these tensors can be approximated by digital algorithms, provided that the underlying object has positive reach and that a binary image of the object is available. Two such algorithms are implemented. Their suggested convergence properties are confirmed in practice via simulations on test sets, and recommendations for input arguments of the algorithms are given. Simulations imply that the accuracy of the estimators is directly related to the resolution of the digitisation, and, further, that an increase in complexity of the object necessitates higher resolutions. Finally, a third algorithm, combining and thus exploiting the most advantageous qualities of both aforementioned algorithms, is proposed.

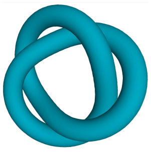
DANSK RESUMÉ

Belysningen af forholdet mellem objekter i den virkelige verden og deres digitaliseringer er en iboende udfordring i billedanalyse. Denne afhandling angår rekonstruktion af såvel topologiske som geometriske egenskaber ved mængder i euklidisk 3-rum henholdsvis det euklidiske plan ud fra binære billeder.

Det første problem, rekonstruktion af topologi i dimension tre, gribes an ved hjælp af kombinatorik af vokselrekonstruktioner kombineret med resultater inden for differentialtopologi. En forbedret digital rekonstruktion baseret på binære billeder af objekter med tilpas glat rand fremsættes. Det bevises, at denne rekonstruktion er ambient isotop til det underliggende objekt, forudsat opløsningen af digitaliseringen er tilpas høj og under visse antagelser om den klassiske vokselrekonstruktion. Den præcise nedre grænse for opløsningen, relateret til krumningen af objektets rand, gives, og der argumenteres for, at dette er den bedst mulige grænse, hvis topologisk ækvivalens skal kunne garanteres. Lignende resultater uden begrænsninger på den klassiske rekonstruktion kan bevises for stærkere antagelser om opløsningen. Det formodes, at den mindste af grænserne for opløsningen også er tilstrækkelig i dette tilfælde.

Til rekonstruktion af geometri i dimension to vælges en praktisk tilgang, som anvender digitale algoritmer. Geometrien af objekter i den virkelige verden kan beskrives ved Minkowski-tensorer. Det er blevet påvist, at disse tensorer kan approksimeres med digitale algoritmer, forudsat det underliggende objekt har positiv rækkevidde, og at et binært billede af objektet er tilgængeligt. To sådanne algoritmer implementeres. Deres teoretiske konvergensegenskaber bekræftes i praksis via simuleringer på testmængder, og der gives anbefalinger til input-argumenter for algoritmerne. Simuleringer indikerer, at nøjagtigheden af estimatorerne er direkte forbundet med opløsningen af digitaliseringen, og endvidere at forøgelse af objektets kompleksitet fordrer højere opløsninger. Endelig fremsættes en tredje algoritme, som kombinerer og dermed udnytter de mest fordelagtige kvaliteter ved begge førnævnte algoritmer.

PREFACE



The present PhD thesis consists of three chapters. In Chapter 1, an introduction to digital reconstruction is given and some of the key definitions of the thesis are presented. Chapter 2 is joint work between the author and main supervisor, Andrew du Plessis[*,†]. Here, a digital reconstruction is proposed for the class of r-regular objects in dimension three, and we show that the reconstruction is topologically equivalent to the reconstructed object in the sense that there exists an ambient isotopy between the two. Chapter 3 concerns the reconstruction of geometry in dimension two via Minkowski tensors of sets with positive reach. Estimators of the Minkowski tensors can be computed via two algorithms which have been implemented in MATLAB. This work is a collaboration between the author and co-supervisor Markus Kiderlen[*,‡].

The three chapters, which constitute the main body of the thesis, are followed by four appendices. These appendices contain additional theory and technical details not included in the main text, in order to further readability.

The research of the thesis has been presented in the following contexts:

- 13th International Workshop on Real and Complex Singularities, São Carlos, Brazil, July–August 2014;
- Visit to Gerd Schröder-Turk at Universität Erlangen-Nürnberg, Erlangen, Germany, October 2014;
- Ninth Internal CSGB Workshop, Middelfart, Denmark, November 2014;
- Visit to Herbert Edelsbrunner at IST Austria, Klosterneuburg, Austria, January 2015;
- Workshop on Integral Geometry and Valuation Theory, Zürich, Switzerland, June 2015;
- Eleventh Internal CSGB Workshop, Højbjerg, Denmark, November 2015;
- Workshop on Stochastic Geometry, Stereology and their Applications, Sønderborg, Denmark, June 2016.

The author recommends that this thesis be read as a PDF file and not in a printed version due to the large number of highly detailed figures and the intricate reference system: All 2D figures and plots are made in vector graphics (by use of Inkscape respectively PGFPlots), and the reader may prefer to zoom in to discover further details. 3D images are created using Sketchup Make and exported as PNG files and therefore do not contain the same level of detail, but the reader may still find it useful to have a closer look than the one available in the printed version. References, highlighted in maroon, are hyperlinks, which will direct the reader to: entries in the bibliography, illustrations of certain concepts introduced in the thesis, definitions, theorems, and so forth.

[*] Centre for Stochastic Geometry and Advanced Bioimaging (CSGB), Aarhus University
Web: www.CSGB.dk

[†] Associate professor Andrew du Plessis, matadp@math.au.dk

[‡] Associate professor Markus Kiderlen, kiderlen@math.au.dk

ACKNOWLEDGEMENTS

The author's PhD studies were financed in part by Centre for Stochastic Geometry and Advanced Bioimaging (CSGB), funded by the Villum Foundation; in part by Aarhus University.

I should like to thank my main supervisor, Andrew du Plessis, for being patient with me through the years and offering guidance in my work, and Markus Kiderlen for nudging me in the direction of the more applied topic of implementation of algorithms and for co-supervising the project.

Thanks to Anne Marie Svane for sharing her great knowledge of Minkowski tensors, and to both her and Astrid Kousholt for sharing office space and lunch breaks with me the last couple of years.

I owe a great thanks to Lars 'daleif' Madsen for helping me to achieve the highly customised layout of this thesis, and I do apologise for being so very demanding in this regard.

My family has been exceedingly supportive and helpful, especially during the final stages of my PhD studies. I could not have managed the combination of writing my thesis, renovating our newly purchased home, and applying for a job without their unconditional support.

A very special thanks goes out to Sean Geggie for supporting me through all my endeavours, not just the ones related to my PhD studies; for encouraging and taking care of me.

'Transversal' is a noun; the adjective is 'transverse'.

– J. H. C. Whitehead, 1959

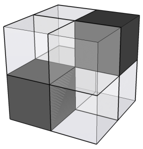
Transversal *[adjective]: (Of a line) cutting systems of lines.*

– Oxford Dictionaries

I will leave the definitions to the Mathematicians.

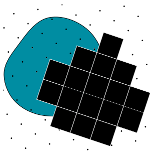
– Sabrina Tang Christensen, 2016

CONTENTS



One

INTRODUCTION



In image analysis, one seeks to extract information about a real world object given only a discrete representation of it – a *digitisation*. Its wide range of applications can be found in fields such as Biology [4, 9], Physics [21], and Materials Sciences [34, 35]. In his book [30] from 1982, Pavlidis proves that pixel and cellular reconstructions of regular objects in dimension two fully capture the topology, provided that the resolution of the digitisation be sufficiently high. In 2007, Stelldinger et al. sought to prove a similar result on topological as well as geometric properties of voxel reconstructions in the three-dimensional case for a class of objects with smooth boundary [38].

Another way of approaching the reconstruction of geometry is through Minkowski tensors, which are tensor valued generalisations of the Minkowski functionals; see for instance the excellent book [32] by Schneider for a review. The Minkowski tensors capture the entire geometry of an object, and in [19], algorithms for the computation of estimators of the Minkowski tensors from finite point samples, e.g. digitisations, of sets with positive reach are presented. Moreover, these estimators converge to the Minkowski tensors of the digitised object as the resolution tends to infinity.

In order to give an introduction to the work presented in this thesis, we give now the key definitions that we will be using throughout. We equip Euclidean n-space $\mathbb{R}^n$ with the standard inner product $\langle\,\cdot\,,\,\cdot\,\rangle$ and induced norm $|\cdot|$. Letting $\mathbb{Z}$ denote the integers, a **(cubic) lattice** $dL \subset \mathbb{R}^n$ is any rotated, translated version of $d\mathbb{Z}^n$, where $d > 0$ is called the **lattice distance**. An element $l \in dL$ is called a **sampling point**. We adopt a basic digitisation model of a bounded set $A \subset \mathbb{R}^n$ by considering the set

$$A_0 = A \cap dL$$

of all lattice points in A. Another way to describe the process in which A_0 is obtained is to say that we *sample A with dL* or, equivalently, that $A_0 = A \cap dL$ is a point sample of A. In signal processing, A_0 is interpreted as the result of an ideal sampler. The discrete set A_0 is called the **digitisation** of A by dL, and d^{-1} is called the **resolution** of the digitisation. Figures 1.1(a) and (b) illustrate the digitisation process.

The **Voronoi cell** of $x \in A_0$ with respect to the digitisation A_0 is defined as

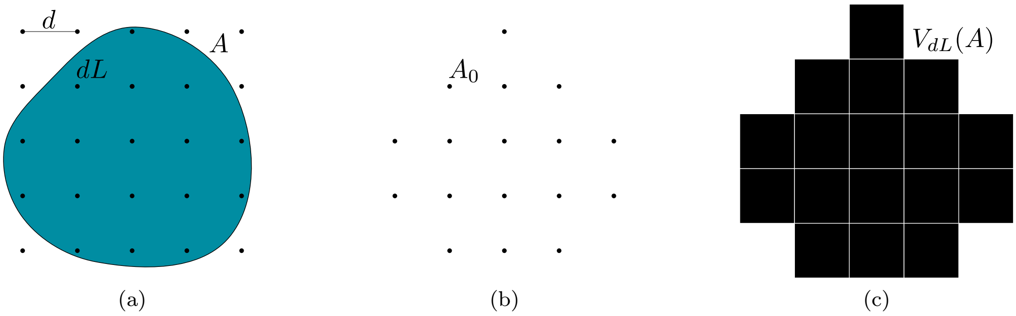
$$\mathcal{V}_{A_0}(x) = \{y \in \mathbb{R}^n \mid |y - x| \leq |y - z| \text{ for all } z \in L\}.$$

A Voronoi cell of a sampling point $x \in A_0$ is called a Voronoi cell of A_0. The Voronoi cells of A_0 have pairwise disjoint interiors, and their (finite) union coincides with $\mathbb{R}^n$. Similarly, we may choose to consider Voronoi cells with respect to a lattice dL (a locally finite set),

$$\mathcal{V}_{dL}(l) = \{y \in \mathbb{R}^n \mid |y - l| \leq |y - z| \text{ for all } z \in dL\}, \tag{1.1}$$

where $l \in dL$. When the lattice dL is understood, we shall write $\mathcal{V}(l) = \mathcal{V}_{dL}(l)$ for brevity.

For $l \in dL$, the set $\mathcal{V}(l)$ is an n-dimensional hypercube in $\mathbb{R}^n$ with side length d and centre l. For lattices dL in $\mathbb{R}^2$, $\mathcal{V}(l)$ can be interpreted as the **pixel** of $l \in dL$ and in dimension 3 the **voxel**.



(a) (b) (c)

Figure 1.1 We sample a set A in $\mathbb{R}^2$ with a lattice dL (a) in order to obtain the digitisation A_0 (b) and the black (pixel) reconstruction $V_{dL}(A)$ with resolution d^{-1} (c).

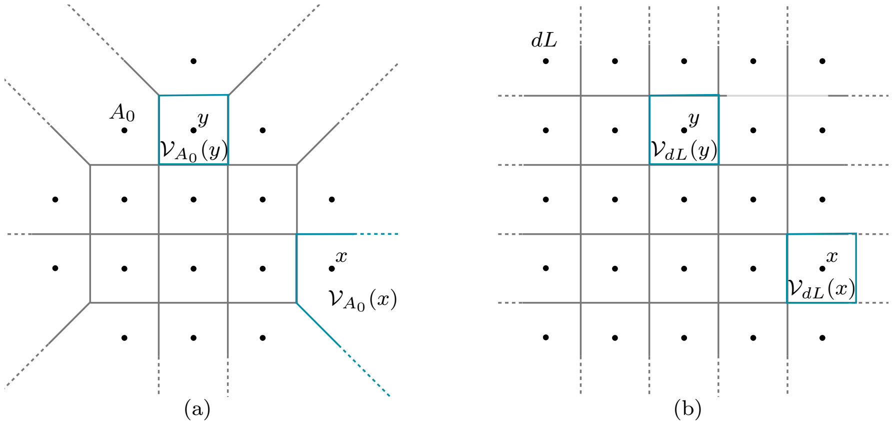


Figure 1.2 Comparison in dimension two of Voronoi cells with respect to a digitisation $A_0 \subset dL$ (a) and with respect to a lattice dL (b). Here, y is an inner point of A_0 whereas x is not.

Clearly, $\mathcal{V}_{A_0}(x)$ is a hypercube of side length d if and only if the $2n$ neighbours of x in the axis directions of dL are all elements of A_0. In this case, $\mathcal{V}_{A_0}(x) = \mathcal{V}_{dL}(x)$, and we say that x is an **inner point** of A_0. In the Euclidean plane $\mathbb{R}^2$, x is an inner point if its usual four-neighbours are in A_0; see Figure 1.2 for an illustration.

Definition 1.1 **Definition.** *The **(digital Voronoi) reconstruction** of A by dL is the union*

$$V_{dL}(A) = \bigcup_{l \in A_0} \mathcal{V}_{dL}(l)$$

of Voronoi cells with respect to dL of all sampling points in the digitisation $A_0 = A \cap dL$.

It is thus the collection of sampling points intersecting A that constitutes the digitisation and their corresponding Voronoi cells with respect to dL that constitute the reconstruction. We shall simply write $V_{dL}(A) = V$ when A and dL are understood. In dimension two (respectively three), V is called the **pixel (voxel) reconstruction** of A. The pixel reconstruction is also called the Gauss-digitisation [22]. An example of the reconstruction process of Definition 1.1 in dimension two is illustrated in Figure 1.1.

Definition 1.2 **Definition.** *Given a digital reconstruction V, a voxel $\mathcal{V}(l) \subset V$ is called a **black voxel**. A voxel $\mathcal{V}(l) \not\subset V$ is called a **white voxel**.*

The reconstruction V is thus the union of black voxels, as shown in Figure 1.1(c), and we refer to $V = V_{dL}(A)$ also as the **black-and-white reconstruction** of $A \subset \mathbb{R}^n$. Another possibility is to consider *greyscale images*, where one assigns an intensity ranging from 0 to 1 to each sampling point of the lattice, depending on the overlap of the object A with the Voronoi cell of the sampling point: Black voxels satisfy $\mathcal{V}(l) \subset A$; white voxels $\mathcal{V}(l) \cap A = \emptyset$. Such images are considered for instance in [39]. Working with greyscale images introduces a new level of challenges, and we shall not consider them here.

The question now is what can be deduced about the topology and geometry of A when we only have access to its digitisation $A \cap dL$. The answer depends not only on the resolution of the digitisation but also on the properties of the set A. Sets with *positive reach*, first introduced by Federer in [13], are shown in [19] to be of special interest with regard to the derivation of geometric properties from digitisations.

For a set $A \subset \mathbb{R}^n$, let $\delta_A : \mathbb{R}^n \to \mathbb{R}$ be the **distance function** on $\mathbb{R}^n$ defined by

$$\delta_A(x) = \inf\{|x - a| \mid a \in A\}, \tag{1.2}$$

i.e. $\delta_A(x)$ is the distance from $x \in \mathbb{R}^n$ to A. A **nearest point** to x in A is a point $y \in A$ such that $|x - y| = \delta_A(x)$. Let

$$\operatorname{Unp}(A) = \{x \in \mathbb{R}^n \mid x \text{ has a unique nearest point in } A\}. \tag{1.3}$$

For $r \geq 0$, write

$$A^r = \{x \in \mathbb{R}^n \mid \delta_A(x) \leq r\} \tag{1.4}$$

for the ***r*-parallel set** of A. Let $p_A : \operatorname{Unp}(A) \to A$ be the **metric projection**, which maps a point $x \in \operatorname{Unp}(A)$ to its unique nearest point $p_A(x)$ in A. Clearly p_A is well-defined.

Definition 1.3 **Definition.** *The **reach** of a set $A \subset \mathbb{R}^n$, denoted by* $\operatorname{Reach}(A)$*, is the supremum over all r such that $A^r \subset \operatorname{Unp}(A)$. If $\operatorname{Reach}(A) > 0$, A is a **set with positive reach**.*

In other words, the reach of A is the supremum over all $r \geq 0$ such that for all $x \in \mathbb{R}^n$ satisfying $\delta_A(x) < r$, there exists a unique point $a \in A$ nearest x. The class of sets with positive reach contains all convex sets, and certain non-convex sets are included as well as we shall see in Example 1.6. A set has infinite reach if and only if it is convex; the non-trivial implication is given by Motzkin's theorem; see for instance [32, Theorem 1.2.4]. Our work in Chapter 3 revolves around sets with positive reach.

Closely related to the notion of reach are those of *medial axis* and *exoskeleton*. Denote by ∂A the boundary of $A \subset \mathbb{R}^n$. For a set $A \subset \mathbb{R}^n$, the **medial axis**, $\operatorname{Ma}(A)$, of A is the set of points in $\mathbb{R}^n$ which do not have a unique nearest point on ∂A. The medial axis is sometimes also referred to as the *topological skeleton* of the set. We remark that $\operatorname{Ma}(A)$ is *not* the complement of $\operatorname{Unp}(A)$ defined in (1.3), since every point $a \in A$ has itself as unique nearest point in A and thus belongs to $\operatorname{Unp}(A)$, but a point in A may belong to $\operatorname{Ma}(A)$, e.g. the ellipse in $\mathbb{R}^2$ has medial axis the line segment connecting its foci. The complement of $\operatorname{Unp}(A) \subset \mathbb{R}^n$, on the other hand, is called the **exoskeleton** of A and is denoted by $\operatorname{Exo}(A)$. The exoskeleton coincides with the intersection of the medial axis and the complement A^C of A. Let $B^n_r(x) = \{y \in \mathbb{R}^n \mid |x - y| \leq r\}$ denote the closed ball in $\mathbb{R}^n$ with radius $r > 0$ and centre $x \in \mathbb{R}^n$. The boundary of the unit ball $B^n_1(0)$ centred at $0 \in \mathbb{R}^n$ is the unit sphere $S^{n-1} = \{y \in \mathbb{R}^n \mid |y| = 1\} \subset \mathbb{R}^n$. We write B_r for a closed ball of radius r when we do not need to specify the centre, and the dimension is understood. Note that $x \in \mathbb{R}^n$ belongs to $\operatorname{Exo}(A)$ if and only if $B^n_s(x)$ with radius $s = \delta_A(x)$ has more than one point of intersection with $\operatorname{cl}(A)$.

It would be natural to require that the resolution of a digitisation guarantee that no (connected) components of the digitised object go undetected by sampling. This is not in general ensured for sets with positive reach due to the possible existence of lower dimensional parts. We could circumvent this problem by requiring that our objects be *topologically regular*, which means that they are equal to the closure of their interior. In this case, letting $\operatorname{int}(A) = A \setminus \partial A$ denote the interior of A and $\operatorname{cl}(A) = A \cup \partial A$ the closure, $\operatorname{Reach}(A) = \operatorname{Reach}(\operatorname{int}(A)) = \operatorname{Reach}(\operatorname{cl}(A))$. Topological regularity will be preconditioned in Chapter 3. Another possibility, and the one we shall make use of in Chapter 2, is to consider *r-regular sets*. We will give two equivalent definitions of r-regularity.

Definition 1.4 **Definition.** *A closed set $A \subset \mathbb{R}^n$ is **r-regular** for $r \in (0, \infty)$ if one of the following equivalent statements applies:*

(i) *At any point $x \in \partial A$, there exist two closed balls $B_r \subset A$ and $B'_r \subset \operatorname{cl}(A^C)$ with radius r such that $B_r \cap B'_r = \{x\}$;*

(ii) *The sets A and $\operatorname{cl}(A^C)$ are equal to unions of closed balls with radius r.*

Notice that given any r' such that $0 < r' \leq r$, A is also r'-regular. That the two statements of Definition 1.4 are equivalent is formulated as Proposition A.1, and the proof, due to du Plessis, is presented in Section A.1 of Appendix A.

Example 1.5 **Example.** If $A \subset \mathbb{R}^n$ is r-regular, A has reach at least r and, in fact, so does $\operatorname{cl}(A^C)$. For consider any point $y \in \mathbb{R}^n$ with $s = \delta_{\partial A}(y) < r$, i.e. y is an element of the interior of at least one of the r-parallel sets A^r or $(A^C)^r$; see (1.2) for the definition of $\delta_{\partial A}$ and (1.4) for the definition of r-parallel sets. Let $x \in \partial A$ be a nearest point on ∂A to y. Then there exist $B_r \subset A$ and $B'_r \subset \operatorname{cl}(A^C)$ such that $B_r \cap B'_r = \{x\}$, and y is an element of one of these balls. Then the ball $B_s(y)$ is contained in either B_r or B'_r, and $B_s(y) \cap \partial A = \{x\}$, so x is the unique nearest point on ∂A to y. Taking supremum over $s < r$ yields the result. ■

Example 1.6 **Example.** From Definition 1.4(i), it is clear that the ball B_r is itself r-regular. Further examples and counter-examples of sets with positive reach and regularity are given in Figure 1.3 for four objects in $\mathbb{R}^3$.

A torus $T^3 \subset \mathbb{R}^3$ is constructed by identifying opposite sides of a rectangle as shown in Figure 1.4. Let one pair of sides have length a and the other b, see Figure 1.4(a), and assume that $a \geq b$. Identify first the sides of length a. We then obtain an open cylinder of length a; see Figure 1.4(b). Finally, we identify the sides with length b, which are now rings; see Figure 1.4(c). This yields a torus T^3. We can fit a ball of circumference b inside T^3.

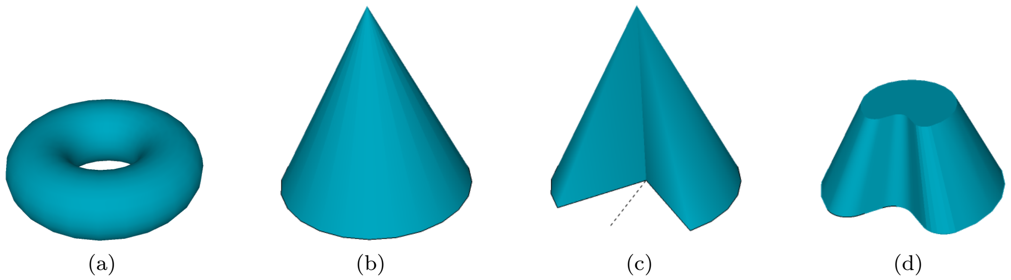


Figure 1.3
The non-convex solid torus in (a) has positive reach $r > 0$ and is r'-regular for some $r' \leq r$; the convex solid cone in (b) has infinite reach but is not r-regular; the non-convex solid partial cone in (c) has reach 0, and the smoothed non-convex solid partial cone in (d) has positive reach but is not r-regular.

Such a ball has radius $r = b/(2\pi)$. The radius of a circle with circumference a is $a/(2\pi)$. The radius of the hole in the torus is $r' = a/(2\pi) - b/(2\pi) = (a-b)/(2\pi)$. Consider now the solid version $\boldsymbol{T}^3$ of T^3. By Definition 1.3, $\text{Reach}(\boldsymbol{T}^3) = r'$, and Definition 1.4 states that for $r'' = \min\{r, r'\}$, $\boldsymbol{T}^3$ is r''-regular.

A solid cone is convex, and all convex sets have infinite, and thus positive, reach. It is not r-regular for any $r > 0$ since we cannot approximate the top of the cone with an r-ball, or, equivalently, the complement of the solid cone does not have positive reach since the medial axis, coinciding with the central axis of the cone, intersects the top of the cone. If we intersect the solid cone with a cube which has one edge coinciding with the central axis of the cone as shown in Figure 1.3(c), we obtain a non-convex set, a *partial cone*, which does not have positive reach. The dashed line in the figure indicates the medial axis or, equivalently, the exoskeleton. If we round off the edges of the partial cone and cut off the top, we obtain a solid object with finite reach; see Figure 1.3(d). ∎

The notion of r-regularity was introduced independently by Pavlidis [30, Definition 7.4] and Serra [36, Definition p. 144]. In our definition, r-regular sets are compact. This convention is line with the definition of Serra, and it means that if $A \subset \mathbb{R}^n$ is an r-regular set, the same applies to the closure of its complement, $\text{cl}(A^C)$. In [38, Definition 1], Stelldinger et al. define r-regularity as a property related to the boundary of a set; they only make assumptions on the interiors of the sets which it bounds. Incidentally, Stelldinger and Köthe also assume compactness in [37, Definition 3].

Initially inspired by [38], the author and main supervisor, Andrew du Plessis, began work on an improved voxel reconstruction of r-regular sets in dimension three with the aim of reconstructing geometric properties. Reconstruction of the geometry proved extremely challenging, so focus was moved instead to the reconstruction of topology. This work is presented in Chapter 2 and ultimately resulted in Theorem 2.14 which states that, given a sufficiently high resolution, the improved voxel reconstruction introduced in Definition 2.13 is homeomorphic as well as ambient isotopic to the underlying object. This is a stronger version of [38, Theorem 16] with an improved restriction on the resolution equal to just half of that required in [38]. The research presented in Chapter 2 has not been published prior to the submission of this thesis, but a paper is in preparation [7].

Concurrently with the above-mentioned work in dimension three, Hug, Kiderlen, and Svane were preparing the paper [19] which concerns the reconstruction of geometry of sets with positive reach in $\mathbb{R}^n$, for n any positive integer, via computation of Minkowski tensors. Estimators of the Minkowski tensors of finite point samples, e.g. digitisations, are obtained by solving systems of linear equations. The estimators converge as the resolution tends to infinity to the Minkowski tensors of the original object; the algorithms are said to be *multigrid convergent*. Due to obvious similarities to our original goal of reconstructing geometry, Markus Kiderlen invited the author to partake in their project by implementing

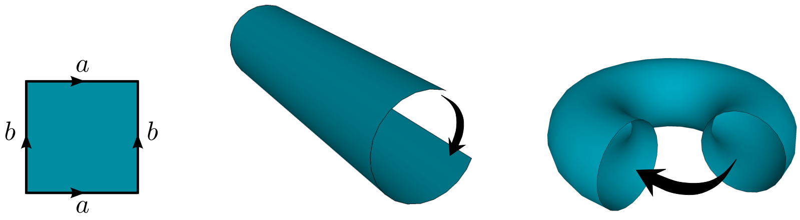


Figure 1.4
The torus is obtained by identifying opposite sides of a rectangle as indicated by the arrows in (a): First, we identify two sides to obtain an open cylinder as illustrated in (b); then we identify the two remaining sides to obtain a torus as illustrated in (c).

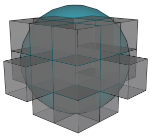
algorithms in MATLAB in dimension two. Chapter 3 contains a revised version of our electronic preprint [8] on this research including the theory of the paper [19], the implementations of the suggested algorithms in dimension two, and results from simulations using these programmes on data sets.

Chapters 2 and 3 can be read independently and both make use of the concepts related to digitisation introduced above. Appendix A is a copy of the notes [11] by du Plessis, containing results utilised by Chapter 2, followed by a proof of one of the key results on which the research in Chapter 2 is based. Appendix B serves to expand on certain concepts that, for the sake of readability, are introduced less rigorously in Chapter 2. Similarly, Appendix C contains computations of tensors not enclosed in Chapter 3. Finally, Appendix D presents documentation of the MATLAB-code for the programme Minktensor2D derived from Chapter 3.

The notation of the appendices builds on that introduced in the three main chapters. The intention is for the reader to first savour the general picture, which is corroborated mainly by figures and examples, from the three main chapters of the thesis and thereupon consult the appendices for the full mathematical details.

Two

Digital Reconstruction of r-Regular Surfaces



The structure of this chapter is as follows: In Section 2.1, we motivate the choice of r-regular sets as our focus by stating a theorem on 1:1 correspondence between the components of r-regular sets and their reconstructions given a sufficiently fine resolution. The possible configurations of black and white voxels of reconstructions are then illustrated. Section 2.2 contains the proof of the theorem on 1:1 correspondence by a series of results concerning the reconstruction. In Section 2.3, the notion of quasi-manifolds is defined, and we show that for slightly finer resolutions than that required in Section 2.1, reconstructions of r-regular sets are quasi-manifolds. This reduces the number of possible voxel configurations, which leads to the definition of the wedged reconstruction. The main theorem of this chapter now states that the wedged reconstruction is homeomorphic, indeed ambient isotopic, to the r-regular set. The proof is outlined and is shown to depend on the construction of a smoothed reconstruction and a vector field transverse to its boundary. Section 2.4 extends the 1:1 correspondence of components of object and reconstruction to the wedged reconstruction. Finally, in Section 2.5, the smoothed reconstruction and the vector field needed to prove the main theorem are constructed.

2.1 The Voxel Reconstruction

Previous work on the two-dimensional problem of reconstructing topology, e.g. [30], [36], suggests that the particular class of r-regular sets is of special interest. The r-regular sets in dimension three are the focus of [38] which first inspired the work presented in this chapter. Definition 1.4 of r-regular sets is stated in full generality, but in this chapter, we shall consider r-regular sets in $\mathbb{R}^3$ exclusively.

In [38], Stelldinger et al. consider r-regular sets and their voxel reconstructions introduced in Definition 1.1. Two sets in $\mathbb{R}^3$ are called ***r*-homeomorphic** if there exists a homeomorphism f between them such that $|f(x)-x| \leq r$ for all $x \in \mathbb{R}^3$. [38, Theorem 16] then states that, under certain assumptions on a voxel reconstruction V of an r-regular set A, V and A are r-homeomorphic. Their results are discussed in Section A.2. In this chapter, we continue the work in [38] with a focus on r-regular sets for the reconstruction of topology in dimension three, as we wish to show a stronger version of [38, Theorem 16], namely Theorem 2.14, to be stated in Subsection 2.3.2.

The choice of r-regular sets as our focus should be motivated by the following theorem, which is a rephrasing of [37, Theorem 1].

Theorem 2.1

Theorem. *If $d\sqrt{3}/2 < r$, there is a 1:1 correspondence between components of an r-regular set $A \subset \mathbb{R}^3$ and components of its voxel reconstruction $V = V_{dL}(A)$ by a lattice dL.*

The diameter of the voxels $\mathcal{V}(l)$ defined in (1.1) of the lattice dL is $d\sqrt{3}$, so Theorem 2.1 states that as long as half the diameter of the lattice is strictly smaller than r, the components of the r-regular set and those of the voxel reconstruction are in 1:1 correspondence.

We require that the resolution of the digitisation guarantee that no components of the digitised object go undetected by sampling. For $r < d\sqrt{3}/2$, the r-regular ball B_r^3 has diameter smaller than that of a d-lattice, which means that the sampling points of the lattice may not intersect B_r^3. This would result in an empty reconstruction; see Figure 2.1. Hence for d even slightly bigger than the bound given in Theorem 2.1, i.e. for a lattice even slightly coarser, the result will not hold, so $d\sqrt{3}/2$ is the best possible lower bound on r if sampling is to detect all components. We shall see in the next section, however, that this 1:1 correspondence between components does not necessarily imply equivalence of topology between a set and its reconstruction. We will prove Theorem 2.1 in Section 2.2.

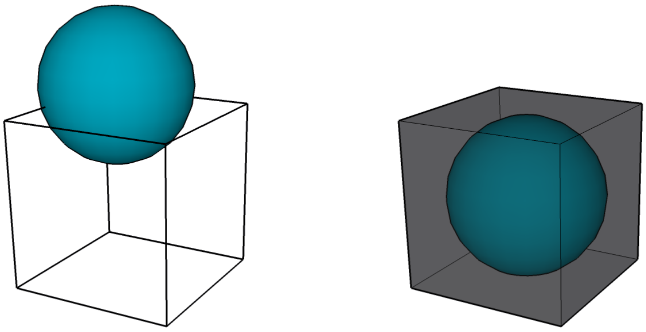


Figure 2.1
For small resolutions, sampling may not detect an object. The ball in this figure can have either no digitisation (left) avoiding all sampling points or one black voxel as its digitisation (right) when it intersects the corresponding sampling point.

Convention

Convention. For the remainder of this chapter, we let $A \subset \mathbb{R}^3$ be an r-regular set. We denote by V the voxel reconstruction of A with respect to a lattice dL, and r is assumed to be greater than $d\sqrt{3}/2$. ■

We will now take a closer look at the composition of V. Assume for now that no sampling points $l \in dL$ lie on ∂A. We shall see later that this is in fact not a restriction.

An example of a voxel reconstruction of the r-regular ball B_r is shown in Figure 2.2. A voxel reconstruction V consists of a number of vertices, edges, and faces, referred to here also as the k-cells of V for $k = 0, 1, 2$ respectively, namely those of the black voxels $\mathcal{V}(l) \subset V$.

Definition 2.2

Definition. *The k-cells of V which are contained in ∂V are called the **boundary cells** of V: the **boundary vertices**, **edges**, and **faces** respectively for $k = 0, 1, 2$.*

A black and a white voxel as defined in Definition 1.2 may have a number of common k-cells; the boundary cells of V. In other words, the boundary of V is the common boundary of the union of black voxels and the union of white voxels, so boundary cells are both black and white. The reconstruction V is thus the union of black voxels, and the union of white voxels is equal to $V^C \cup \partial V$.

Definition 2.3

Definition. *Two voxels are called **k-cell adjacent** if they share a k-cell for $k = 0, 1, 2$ and **adjacent** if they share any k-cells. If two voxels are adjacent but not face-adjacent, their common cells are called **critical cells**.*

The boundary ∂V should be a first approximation to ∂A. Consider a union K in $\mathbb{R}^3$ of 2-by-2-by-2 voxels of a lattice dL. All eight voxels have a common 0-cell, the centre of K. There are now $2^8 = 256$ distinct ways to assign a black or a white colour to each voxel in K. These can be reduced to just 14 unique choices [24], up to rotational and reflectional symmetry and complementarity (replacing white voxels with black voxels and vice versa), which we will refer to as the possible **voxel configurations**. The 14 voxel configurations are depicted as diagrams in Figure 2.3. In the figure, each configuration is assigned a number (n), $1 \leq n \leq 14$, and is henceforth referred to simply as (n) if specified individually. We have chosen to illustrate those configurations which contain the smallest number of black voxels over their complementary configurations, since we will later present them as three-dimensional voxel images built from black and white voxels. These are more easily interpreted when fewer black voxels are included.

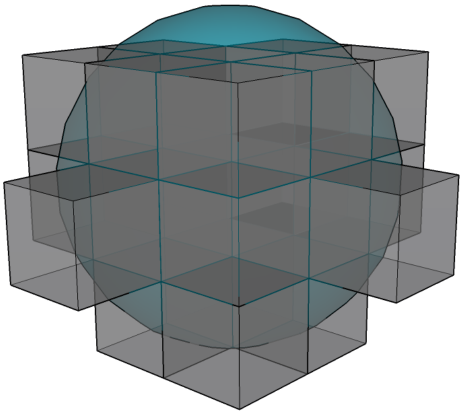
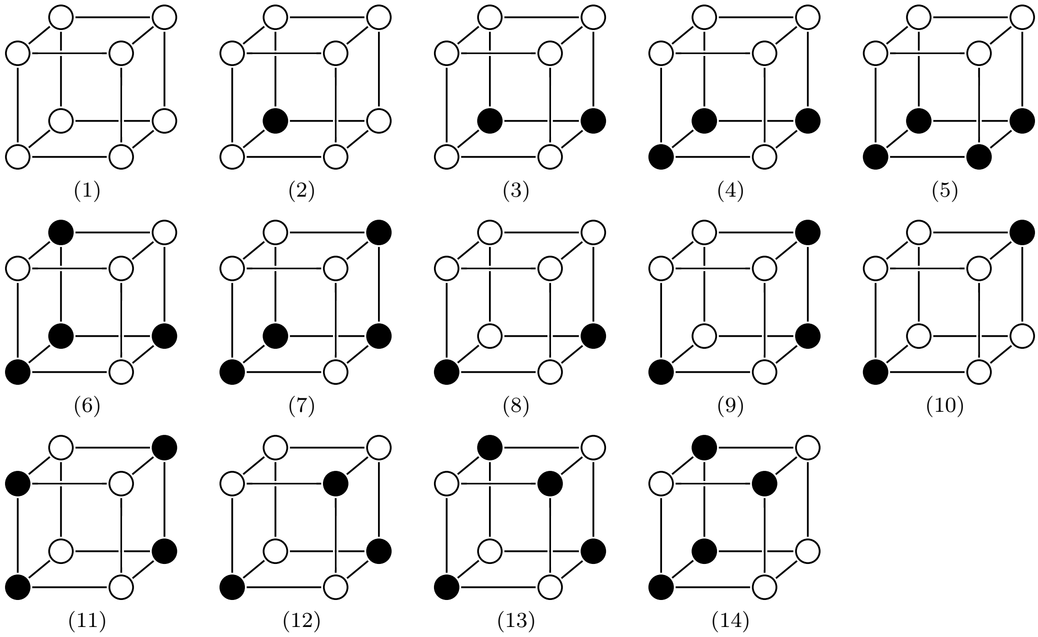


Figure 2.2
The black voxels constitute a voxel reconstruction of the blue ball.

Figure 2.3
Diagrams of the 14 distinct configurations of black and white voxels up to rotational and reflectional symmetry and complementarity. The balls represent the sampling points of the eight voxels and their colour that of the corresponding voxel.



2.2 COMPONENTS OF THE VOXEL RECONSTRUCTION

We return now to the proof of Theorem 2.1. Our proof follows that of [37, Theorem 1] to a great extent but is simplified since we treat only the three-dimensional case.

For a topological space X, we introduce the set $\mathcal{C}(X)$ of non-empty components of X, and we use the convention that $\mathcal{C} = \mathcal{C}(A)$. Let $d\sqrt{3}/2 < s < r$ and set

(2.1) $$X_\ominus(s) = X_\ominus = \{x \in X \mid \delta_{\partial X}(x) > s\};$$

the set of points $x \in X$ with distance greater than s to the boundary of X, where $\delta_{\partial X}$ is the function defined by (1.2). In Mathematical Morphology, $X_\ominus$ is called the *s-erosion of X*. Erosion of X by s corresponds to removing all points from X which are contained in a closed ball B_s with centre on ∂X;

(2.2) $$X_\ominus = X \setminus \bigcup_{x \in \partial X} B_s(x).$$

Now let $A' \in \mathcal{C}$ be a non-empty component of A and let B denote the set of sampling points in dL whose voxels intersect $A'_\ominus$,

(2.3) $$B = \bigl\{ l \in dL \bigm| \mathcal{V}(l) \cap A'_\ominus \neq \emptyset \bigr\}.$$

Lemma 2.4 **Lemma.** *Let $A' \in \mathcal{C}$ and let B be defined as in (2.3). Then the intersections $A' \cap dL$ and $A' \cap B$ are non-empty.*

Proof. Since A is r-regular, so is the component A', so A' contains a closed ball $B_r(x)$ for some $x \in \mathbb{R}^3$. Since $s < r$, x is an element of $A'_\ominus$ defined in (2.2), implying that $A'_\ominus$ is non-empty. We let $l \in dL$ denote the sampling point satisfying $x \in \mathcal{V}(l)$. If there is more than one such sampling point, l is chosen arbitrarily. By (2.3), l is then an element of B, and it satisfies $|x - l| \leq d\sqrt{3}/2 < s$. A sampling point l which is closer than s to a point in the s-erosion of A' is also an element of A'. Hence $l \in A' \cap dL$ and $l \in A' \cap B$. □

As a consequence of Lemma 2.4, sampling of A with dL detects all components of A.

Lemma 2.5 **Lemma.** *Any pair of voxels $\mathcal{V}(l)$ and $\mathcal{V}(l')$ with l, $l' \in B$ is connected by a chain of face-adjacent voxels whose sampling points are also in B.*

Proof. Since A' is r-regular and $s < r$, $A'_\ominus$ is a non-empty open connected set. This means the sets $\operatorname{int}(\mathcal{V}(l)) \cap A'_\ominus$ and $\operatorname{int}(\mathcal{V}(l')) \cap A'_\ominus$ are non-empty, so we choose points b in the former and b' in the latter. The set $A'_\ominus$ is path-connected since it is an open connected set in Euclidean 3-space. We can thus find a path $j \colon [0,1] \to A'_\ominus$ such that $j(0) = b$ and $j(1) = b'$. Without loss of generality, we may assume that j is smooth and, by transversality, that it intersects voxel boundaries only in the interior of the 2-cells of the voxels. The path j enters and leaves any voxel via the compact 2-cells a finite number of times. We define an ordering on the voxels as follows: Let $\mathcal{V}(l)$ or $\mathcal{V}(l')$ be the first voxel $\mathcal{V}_1$ which j enters, and let the other one be the last voxel $\mathcal{V}_n$ for some $n \in \mathbb{N}$. The second voxel, $\mathcal{V}_2$, is the one which j enters after leaving $\mathcal{V}_1$ the last time. Iteratively, $\mathcal{V}_k$ is the voxel which j enters after leaving $\mathcal{V}_{k+1}$ the last time. The iteration terminates once j enters $\mathcal{V}_n$ the first time. This order defines a chain of face-adjacent voxels, and since j is contained in $A'_\ominus$, the corresponding sampling points are elements of B. □

Lemma 2.6 **Lemma.** *For $A' \in \mathcal{C}$, each sampling point in $A' \cap dL'$ is either an element of B or connected to one such by a chain of face-adjacent voxels with sampling points in A'.*

Proof. Let $l \in dL$ be an element of A'. If $\delta_{\partial A'}(l)$ is greater than r, l is an element of $A'_\ominus$ because $s < r$, and hence $l \in B$. If not, consider any ball $B_r(o) \subset A'$ tangent to $\partial A'$ and containing l, where $o \in \mathbb{R}^3$. Such a ball exists due to r-regularity of A' and because $\delta_{\partial A'}(l) \leq r$. Clearly $\delta_{\partial A'}(o) = r$, and since $s < r$, o is an element of $A'_\ominus$. We introduce a coordinate system with o as the origin and axes parallel to the axis directions of dL. Let l have coordinates (x, y, z). Consider the six voxels with samplings points at $(x \pm d, y, z)$, $(x, y \pm d, z)$, and $(x, y, z \pm d)$. These are all the face-adjacent voxels to $\mathcal{V}(l)$. They are at least as far from o as l if and only if $|x|, |y|, |z| \leq d/2$. Thus, either o is contained in $\mathcal{V}(l)$, i.e. $l \in B$, or one of the adjacent voxels has sampling point (x', y', z') closer to o. In the latter case, (x', y', z') is clearly also contained in $B_r(o)$, and we repeat the above procedure for (x', y', z'). Iterating this process, at some step of the iteration, we will arrive at a sampling point l' which is at least as close to o as any other sampling point, i.e. o is contained in $\mathcal{V}(l')$. This means that l' is an element of B. We thus obtain a finite chain of face-adjacent voxels from $\mathcal{V}(l)$ to $\mathcal{V}(l')$, $l' \in B$, all with sampling points in the compact set $\bar{B}_r(o) \subset A'$. □

Equipped with these three lemmas, we can prove the following proposition.

Proposition 2.7 **Proposition.** *The voxel reconstruction $V_{dL}(A') = V(A')$ of $A' \in \mathcal{C}$ is connected.*

Proof. Lemma 2.4 ensures that the voxel reconstruction process maps any component $A' \in \mathcal{C}$ to V. We wish to show that A' is mapped to just one component $V' \in \mathcal{C}(V)$. Pick any sampling point l in A'. By Lemma 2.6, l is either itself in B or connected to a sampling point that is by a chain of face-adjacent voxels with sampling points in A'. All sampling points in B are connected by a chain of face-adjacent voxels with sampling points in $B \subset A'$ by Lemma 2.5, so all sampling points in A' are connected by face-adjacent voxels with sampling points in A'. Hence $V(A')$ is connected. □

Proposition 2.8 **Proposition.** *The voxel reconstruction V of A satisfies*

$$V = \bigcup_{A' \in \mathcal{C}} V(A'), \tag{2.4}$$

and this union is disjoint

Proof. It should be clear that (2.4) holds when the union is not assumed to be disjoint: by Proposition 2.8, the sets $V(A')$ such that $A' \in \mathcal{C}$ are the components of V, and V is exactly the union of its components. We wish to prove that, given $A', A'' \in \mathcal{C}$, their respective voxel reconstructions $V(A') = V'$ and $V(A'') = V''$ do not intersect. The boundaries $\partial A'$ and $\partial A''$ are compact, so there exist points $x \in \partial A'$ and $y \in \partial A''$ which minimise $|x - y|$ (these points are not necessarily unique). If we fix $x \in \partial A'$, this corresponds to minimising the distance function $\partial A'' \to \mathbb{R}$, $y \mapsto |x - y|$, which means y and x lie on a normal of $\partial A''$. Then the ball $B_{|x-y|}(m)$ of radius $|x - y|$ and with centre the midpoint m of the line segment from x to y is tangent to $\partial A''$ at y. By an analogous argument for $y \in \partial A''$ and the distance function $\partial A' \to \mathbb{R}$, $x \mapsto |x - y|$, $B_{|x-y|}(m)$ is tangent to $\partial A'$ at x. Due to r-regularity of A' and A'', $B_{|x-y|}(m)$ has radius at least r. Thus $|x - y| \geq 2r$.

Consider now two voxels $\mathcal{V}(l)$ and $\mathcal{V}(l')$ with $l \in A'$ and $l' \in A''$. By the above argument, $|l - l'| \geq 2r$. The maximal distance between sampling points of adjacent voxels is equal to the diameter of a voxel, $d\sqrt{3}$. Since $d\sqrt{3} < 2r$, $\mathcal{V}(l)$ and $\mathcal{V}(l')$ cannot be adjacent. Hence V' and V'' do not intersect, so the union in (2.4) is disjoint. □

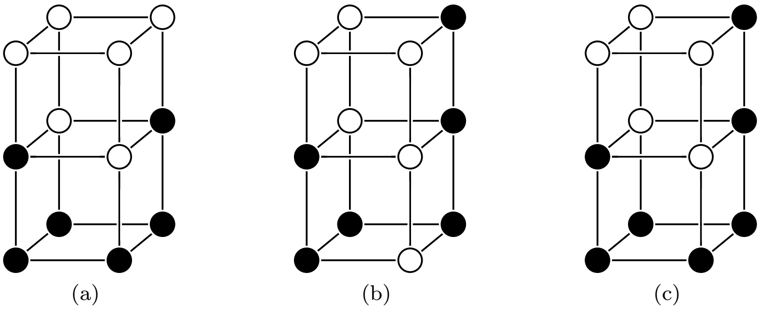


Figure 2.4
Allowed combinations up to rotational and reflectional symmetry and complementarity of configurations (8) and (9) of a voxel reconstruction. (a) is (8) paired with its own complement, (b) is (9) paired with its own complement, and (c) is (9) paired with the complement of (8).

This result now finally yields Theorem 2.1 which we recall here for convenience.

Theorem 2.1. *There is a 1:1 correspondence between components of A and components of V.*

Proof. From Proposition 2.8, we see that each $A' \in \mathcal{C}$ corresponds exactly to its unique reconstruction $V(A') \in \mathcal{C}(V)$, and all elements of $\mathcal{C}(V)$ are realised as the reconstructions of elements of $\mathcal{C}(A')$, thus yielding a 1:1 correspondence between components of A and those of V. □

It would now be desirable to obtain a result similar to Theorem 2.1 for the components of ∂A and ∂V. However, such a result cannot be established unless the boundaries are manifolds; a fact which, along with considerations of the topology of the voxel reconstruction, will motivate the introduction of an alternative reconstruction.

2.3 TOPOLOGY OF RECONSTRUCTIONS

Analogously to the terminology for the corresponding sets V and A, we will refer to ∂V as the **reconstruction** of ∂A. Since we are interested in the preservation of topology in reconstructions, the reconstruction process must preserve the components of the underlying set. By Theorem 2.1, this is ensured for our choice of voxel reconstruction since we require that $d\sqrt{3}/2 < r$. However, this choice of resolution does not imply the existence of a homeomorphism between ∂A and ∂V. As A is r-regular, its boundary is a 2-dimensional C^1 manifold by Proposition A.7(iv). Manifolds in $\mathbb{R}^3$ of dimension 2 are also called **surfaces**, and we will refer to boundaries of r-regular sets as ***r*-regular surfaces**. Contrarily, ∂V is not generally a manifold: configurations (8)–(14) do not have manifold boundary since they contain critical cells; see Definition 2.3. The voxel reconstruction is therefore too coarse for the reconstruction of topology. In Definition 2.13, we suggest a new digital reconstruction, which is ambient isotopic to the original object.

2.3.1 QUASI-MANIFOLD PROPERTIES

Restricting our attention to voxel reconstructions of r-regular sets with $d \cdot 0.95571 < r$ turns out to reduce the number of possible configurations in the voxel reconstruction.

Definition 2.9

Definition. *A voxel reconstruction is a **quasi-manifold (with boundary)** if it satisfies:*

(i) *The only configurations of black voxels (up to rotational and reflectional symmetry and complementarity) are (1)–(10) of Figure 2.3;*

(ii) *Configurations (8) and (9) are only allowed when paired with their own or each other's complement as illustrated in Figure 2.4;*

(iii) *Any pair of black voxels in the same component of the voxel reconstruction is connected by a chain of face-adjacent black voxels.*

To give a better picture of the topology and three-dimensional structure of a quasi-manifold, voxel configurations (1)–(10) are depicted in Figure 2.5 with black and white (transparent) voxels.

Theorem 2.10

Theorem. *Whenever A is an r-regular set with $d \cdot 0.95571 \cdots < r$, $V_{dL}(A)$ is a quasi-manifold.*

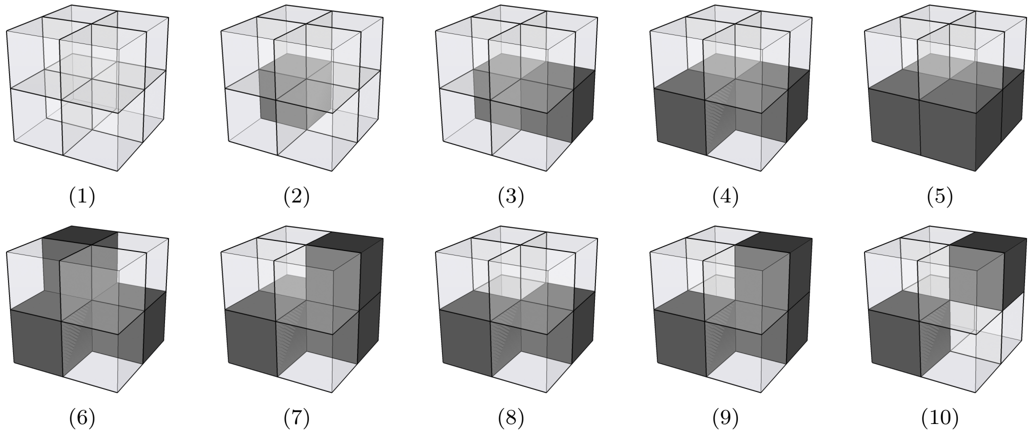


Figure 2.5
The ten possible configurations (up to rotational and reflectional symmetry and complementarity) of black and white voxels in a quasi-manifold.

That V satisfies Definition 2.9(i) and (ii) is proved in Section A.4 by use of results from [11]. The proof of Proposition 2.7 yields a stronger version of Definition 2.9(iii), namely with the relaxed condition $d\sqrt{3}/2 < r$.

Since we are working with $d\sqrt{3}/2 < r$, it would be desirable that Theorem 2.10 hold for this relaxed assumption on the radius of regularity. It seems that this is the case, but we have not so far been able to prove it.

Conjecture 2.11

Conjecture. *Whenever A is an r-regular set with $d\sqrt{3}/2 < r$, the voxel reconstruction $V_{dL}(A)$ is a quasi-manifold.*

On the other hand, what we can show is that all fourteen configurations of Figure 2.3 can occur for any smaller choice of radius of regularity.

Proposition 2.12

Proposition. *If A is an r-regular set and V the voxel reconstruction of A with respect to a lattice dL such that $d\sqrt{3}/2 < r$, all voxel configurations illustrated in Figure 2.3 are possible.*

Proof. Let K be a union of 2-by-2-by-2 voxels in $\mathbb{R}^3$ and denote by $x \in \mathbb{R}^3$ the centre of K. Set $s = d\sqrt{3}/2$ and let A be the ball $B_s(x)$. The boundary of A intersects all centres of the eight voxels in K, and A is s-regular. At any voxel centre l, we may use a smooth bump function to move the boundary of A slightly away from l to render the corresponding voxel $\mathcal{V}(l)$ either black or white. Such a small manipulation of A changes the curvature radius only a little, so the resulting object is s'-regular for some $s' < s$; see also Proposition A.8. In this way, we can determine the colour of any of the eight voxels in K, and hence any configuration of black and white voxels is possible for a radius of regularity just slightly smaller than $s = d\sqrt{3}/2$. □

The quasi-manifold property of V is of great interest since we can construct a manifold from the quasi-manifold in a canonical way and such that this manifold is ambient isotopic to the underlying r-regular set.

Convention

Convention. For the remainder of this chapter, the voxel reconstruction V is assumed to be a quasi-manifold. ■

The next subsection introduces a manifold, *the wedged reconstruction*, based on V.

2.3.2 WEDGED RECONSTRUCTION OF r-REGULAR SURFACES

As should be clear from Figure 2.5, if V contains any of (8)–(10) where critical cells occur, ∂V is not a manifold. The figure also illustrates the fact that critical cells are, in particular, boundary cells. We aim to modify V such that the boundary becomes a surface. To this end, we introduce a *wedging* of the critical edges and vertices through which we obtain the wedged reconstruction of A.

Consider the two configurations containing critical edges, (8) and (9). By Definition 2.9(ii), these are only allowed to occur in certain pairs, namely the ones illustrated in Figure 2.4. When defining a wedging of the voxel reconstruction, we thus consider these three 2-by-2-by-3 configurations. Black wedges are inserted into the critical edges as

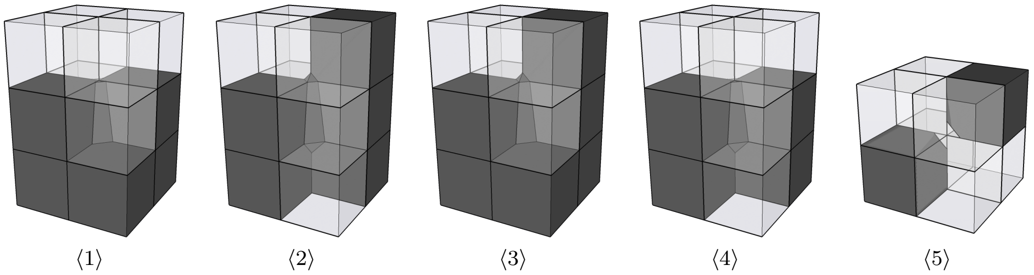


Figure 2.6 Wedging of the critical cells in a quasi-manifold. Each configuration ⟨1⟩–⟨4⟩ hides a black voxel in the far lower corner; compare with the diagram configurations of Figure 2.4.

illustrated in Figure 2.6, where we have also included the complement, 2.6⟨4⟩, of the configuration in Figure 2.4(c) for the sake of completeness. We denote the wedged configurations individually by ⟨1⟩–⟨4⟩ as indicated in the figure. The wedging of (10), which contains a critical vertex, is defined by insertion of a white wedge into the corners of the two black voxels. This new configuration is denoted by ⟨5⟩ and illustrated in Figure 2.6 as well.

The wedges introduced here are defined formally in Definition B.1 of Appendix B. In our definition of the wedging process, we have made the choice of adding black wedges to the configurations with critical edges, but we may as well remove black wedges or, equivalently, add white wedges. With ⟨5⟩, however, we do not have a similar freedom of choice. Notice also that subsequent to wedging, it is no longer meaningful to talk about ⟨1⟩ and ⟨3⟩ respectively ⟨2⟩ and ⟨4⟩ as complementary configurations. We can still talk about the complement of ⟨5⟩, which is obtained in the usual way, i.e. by reversing the black and white colours (also of the wedges).

Definition 2.13

Definition. *The **wedged (voxel) reconstruction** $W = W_V(A)$ of A is obtained from V by wedging critical edges and vertices of configurations (8)–(10) of V using the wedges given by Definition B.1.*

Since the wedged reconstruction is obtained from a voxel reconstruction which in turn depends on the choice of lattice, we say that W is the wedged reconstruction of A with respect to dL.

Convention

Convention. For the remainder of this chapter, W denotes the wedged reconstruction of A with respect to dL. ■

The motivation for introducing the wedged reconstruction is the main theorem of this chapter.

Theorem 2.14

Theorem. *W is homeomorphic to A. Moreover, there exists an ambient isotopy between W and A.*

The notion of *ambient isotopy* will be described in the following subsection, where we also outline the proof of the theorem. As mentioned in Section 2.1, Theorem 2.14 is a stronger version of [38, Theorem 16] which assumes $d\sqrt{3} < r$; we state the theorem for just half of their required radius of regularity. For the increased bound $d \cdot 0.95571 < r$, where we remark that $\sqrt{3}/2 < 0.95571 < \sqrt{3}$, the quasi-manifold property of V is ensured by Theorem 2.10, but if one can dismiss the existence of (11)–(14) in Figure 2.3, it suffices that $d\sqrt{3}/2 < r$.

We shall refer to ∂W as the **wedged reconstruction** of ∂A. Now, Theorem 2.1 was stated for A and its voxel reconstruction V. What we in fact need is the same result for the wedged voxel reconstruction and for the boundaries ∂A and ∂W.

Theorem 2.15

Theorem. *There is a 1:1 correspondence between components of A and components of W. The underlying voxel reconstruction V need not be a quasi-manifold.*

We prove this theorem in Section 2.2 and use it to argue that a similar result holds when A and W are replaced by their boundaries ∂A and ∂W:

Corollary 2.16

Corollary. *There is a 1:1 correspondence between components of ∂A and components of ∂W.*

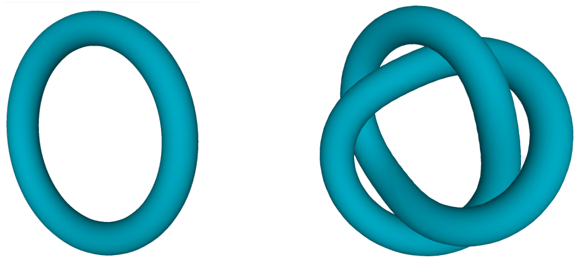


Figure 2.7 The unknot (left) is not ambient isotopic to the trefoil knot (right), since one cannot be twisted into the other without breaking the loop.

As opposed to Theorem 2.15, for the corollary, we need the underlying voxel reconstruction to be a quasi-manifold, since the proof relies on ∂W being a manifold as we shall see in Section 2.4 where we prove these results. We continue now with the examination of the topological properties of W.

2.3.3 Topological equivalence

For the results in this subsection, we need some preparatory topology. For a more thorough review, the reader may consult for instance [23]. For M and N smooth manifolds, a **diffeomorphism** is a smooth invertible map $M \to N$ with smooth inverse. A map $f\colon N \to M$ is a **smooth immersion** if its differential $\mathrm{d}f_p\colon T_pM \to T_{f(p)}N$ is injective at each point $p \in M$. A smooth immersion is called a **smooth embedding** if it is a homeomorphism onto its image. The manifold N is said to be a **smooth submanifold** of M if $N \subset M$ and the inclusion $N \hookrightarrow M$ is a smooth embedding. Smooth submanifolds are sometimes also called embedded or regular submanifolds. Smooth submanifolds are precisely the images of smooth embeddings: if $f\colon N \to M$ is an embedding, the image of f has the structure of a smooth submanifold. For any topological space X, we let Id_X denote the identity mapping on X, i.e. $\mathrm{Id}_X(x) = x$ for all $x \in X$.

Definition 2.17

Definition. *A **smooth isotopy***

$$F\colon N \times [0,1] \to M, \qquad F(x,t) = F_t(x),$$

*is a smooth map such that for each $t \in [0,1]$, the map $F_t\colon N \to M$ is a smooth embedding. Smooth submanifolds A, $B \subset M$ are called **(smoothly) isotopic in M** if there is a smooth isotopy $F\colon A \times [0,1] \to M$ satisfying $F_0 = \mathrm{Id}_A$ and $F_1(A) = B$.*

*A **smooth ambient isotopy** of M is a smooth map $F\colon M \times [0,1] \to M$ such that $F_0 = \mathrm{Id}_M$ and for each $t \in [0,1]$, the map $F_t\colon M \to M$ is a diffeomorphism. The map F has **compact support** if there exists a compact subset $K \subset M$ such that $F_t(x) = 0$ for all $(x,t) \in (M \setminus K) \times [0,1]$. Smooth submanifolds A, $B \subset M$ are **(smoothly) ambient isotopic in M** if there exists a smooth ambient isotopy $F\colon M \times [0,1] \to M$ satisfying $F_1(A) = B$.*

Whereas an isotopy is a deformation from one embedded object to another which distorts the embedding itself, an ambient isotopy distorts the entire ambient space. An intuitive picture of this can be given using knots: an ambient isotopy is one that deforms one knot into another without breaking it. Thus, the knots depicted in Figure 2.7 are examples of non-ambient isotopic knots.

Say we obtain an ambient isotopy F between ∂A and ∂W in $\partial A \times [-1,1]$, where $\partial W \subset \partial A \times [-1,1]$. We wish to generalise F to an ambient isotopy between A and W. This is achieved by observing that F_t is the identity map $\mathrm{Id}_{\mathbb{R}^3}$ near the boundary of $\partial A \times [-1,1]$, which follows from the considerations in [16, Chapter 8.1]. We can thus define a continuous piecewise map $G_t\colon \mathbb{R}^3 \to \mathbb{R}^3$ by

$$G_t(x) = \begin{cases} x & \text{if } x \in \mathbb{R}^3 \setminus (\partial A \times [-1,1]); \\ F_t(x) & \text{if } x \in \partial A \times [-1,1]. \end{cases}$$

Note that $\mathrm{Id}_{\mathbb{R}^3}(A \setminus (\partial A \times (-1,1))) = W \setminus (\partial A \times (-1,1))$. Furthermore, for $t \in [0,1]$, the diffeomorphisms F_t map ∂A to ∂W, $A^C \cap (\partial A \times [-1,1])$ to $W^C \cap (\partial A \times [-1,1])$, and $A \cap (\partial A \times [-1,1])$ to $W \cap (\partial A \times [-1,1])$. These properties related to G_t are illustrated in Figure 2.8. Thus G_t yields an ambient isotopy between A and W, which proves Theorem 2.14.

In Definition 2.27, surfaces T_{-1} and T_1, diffeomorphic to $\partial A \times \{-1\}$ and $\partial A \times \{1\}$ respectively, and a set $T_{[-1,1]}$, diffeomorphic to $\partial A \times [-1,1]$, are introduced. Our goal

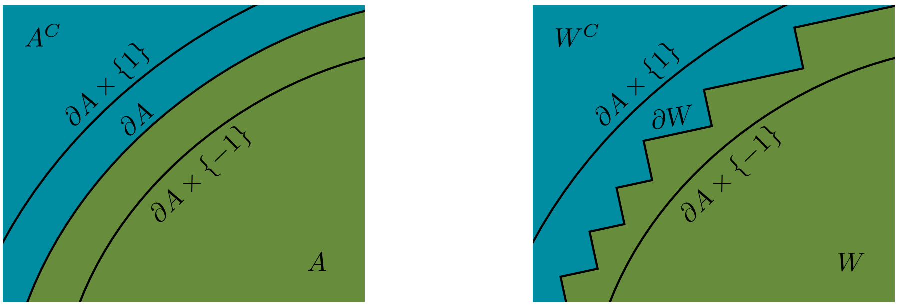


Figure 2.8
The map G_t maps the blue (respectively green) parts of the left-hand figure to the blue (green) parts of the right-hand figure. The blue (green) domain inside the tube $\partial A \times [-1, 1]$ on the left is mapped to the blue (green) domain inside $\partial A \times [-1, 1]$ on the right.

now is to prove that there exists an ambient isotopy between ∂A and ∂W in $T_{[-1,1]}$, thus proving Theorem 2.14.

Proposition 2.18 **Proposition.** *Suppose that ∂A and ∂W are connected. Then ∂W is homeomorphic to ∂A. Moreover, there exists an ambient isotopy between ∂A and ∂W in $T_{[-1,1]}$.*

This proposition is established by a series of results which we state without proof in the following in order to provide a clearer overview. The proofs will follow in subsequent sections. Notice also the assumption on connectivity of the boundaries. In fact, we need the statement for general compact surfaces; a matter which will be addressed in the proof of Proposition 2.18.

For connected closed surfaces S embedded in $\mathbb{R}^3$, the *Euler characteristic* completely determines the topological properties: S is homeomorphic to a sphere with g handles for some unique integer $g \geq 0$ called the genus of S; see Figure 2.9. The genus thus classifies S; this is commonly referred to as **surface classification**. The Euler characteristic, denoted by χ, determines g by the formula $\chi(S) = 2 - 2g$ [16].

Definition 2.19 **Definition.** *Let $X \subset \mathbb{R}^3$ and let S be a surface in $\mathbb{R}^3$. We say that S **separates** X if $X \setminus S$ has exactly two components and these have common boundary $S \cap X$ in X.*

We have the following theorem which is presumably well-known and which we prove in Subsection 2.5.1.

Theorem 2.20 **Theorem.** *Let M be a compact smooth n-manifold without boundary, and let $N \subset M \times (-1, 1)$ be a compact smooth n-manifold without boundary such that N separates $M \times [-1, 1]$, i.e. $M \times [-1, 1] \setminus N$ is a disjoint union of two open sets U, V with $M \times \{-1\} \subset U$ and $M \times \{1\} \subset V$.*

Let v be a continuous nowhere zero vector field on $M \times [-1, 1]$ which points inwards on $M \times \{-1\}$ and outwards on $M \times \{1\}$ and is transverse to N. Then

$$\chi(N) = \chi(M).$$

We remark that we do not need the superset of N to be a product of M and $(-1, 1)$ as long as we can find a set diffeomorphic to such a product.

Assume for now that ∂A is connected. We then wish to apply Theorem 2.20 to ∂A and, ideally, the boundary of W, which we will prove is also connected. However, ∂W is not a smooth manifold. In Definition 2.29, we introduce the *smoothed wedged voxel reconstruction* $Z_W(A) = Z$ of A. The smoothed reconstruction Z is obtained by *smoothing* W. This results in a reconstruction which has smooth manifold boundary homeomorphic to ∂W. We prove in Proposition 2.30 that ∂Z separates the set $T_{[-1,1]}$ introduced in Definition 2.27. Finally, in Definition 2.32, we introduce a vector field ξ satisfying the

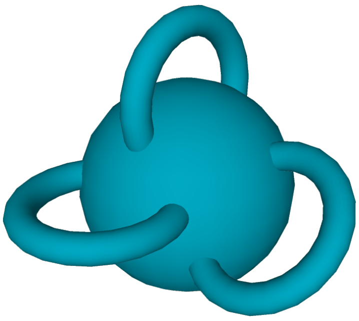


Figure 2.9
A sphere with g handles in $\mathbb{R}^3$ is the unit sphere S^2 for $g = 0$ and the g-fold torus T_g for $g \geq 1$. The picture shows a sphere with three handles.

remaining prerequisites of Theorem 2.20. It now follows that $\chi(\partial A) = \chi(\partial Z)$, so due to surface classification of compact connected orientable surfaces, ∂A is homeomorphic to ∂Z, which was again homeomorphic to ∂W. This proves the first part of Proposition 2.18.

To obtain an ambient isotopy between ∂A and ∂W, we apply the following proposition.

Proposition 2.21

Proposition (Chazal and Cohen-Steiner, 2005 [5]). *Let M and N be compact orientable surfaces such that:*

- *the surface M is homeomorphic to N;*
- *the surface N is a subset of $M \times [-1, 1]$;*
- *the intersections $N \cap (M \times \{-1\})$ and $N \cap (M \times \{1\})$ are empty;*
- *the set $(M \times [-1, 1]) \setminus N$ has two connected components; one containing $M \times \{-1\}$ and the other one containing $M \times \{1\}$.*

Then N is isotopic to M in $M \times [-1, 1]$.

The proof can be found in [5]. We thence obtain an isotopy between ∂A and ∂W in $\partial A \times [-1, 1]$ when these boundaries are connected. The isotopy extends to an ambient isotopy.

Theorem 2.22

Theorem (Isotopy extension theorem). *Let M be a smooth manifold, possibly with boundary; let N be a compact smooth submanifold of M, and let $H\colon N \times [-1, 1] \to M \setminus \partial M$ be an isotopy of N. Then H extends to an ambient isotopy $F\colon M \times [-1, 1] \to M$ with compact support.*

We refer to [16, p. 180] for a proof of the isotopy extension theorem. As a consequence of Theorem 2.22, when ∂A and ∂W are connected, we obtain an ambient isotopy between them which yields the second part of Proposition 2.18. We are now ready to prove Theorem 2.14.

Proof of Theorem 2.14. There exists an ambient isotopy in $\partial A \times [-1, 1]$ between the two boundaries ∂A and ∂W when these are both connected. Suppose that ∂A and ∂W are not connected. By Corollary 2.16, there is a 1:1 correspondence between their components. Denote the components of ∂A by $C_1, \dots, C_n$ and those of ∂W by $D_1, \dots, D_n$. Then for each component C_k of ∂A, $k \in \{1, \dots, n\}$, we have a compactly supported ambient isotopy $F_k\colon C_k \times [0, 1] \to D_k$ where, for each $t \in [0, 1]$, $F_k(x, t) = F_{k,t}(x)$. Consider the map

$$F_t(x) = \begin{cases} F_{k,t}(x) & \text{if } x \in C_k \times [-1, 1]; \\ x & \text{if } x \in \mathbb{R}^3 \setminus (\partial A \times [-1, 1]). \end{cases}$$

The map $F_{k,t}$ is the identity near the boundary of $C_k \times [-1, 1]$, so F_t is a diffeomorphism. Moreover, $F_0 = \mathrm{Id}_{\mathbb{R}^3}$ and $F_1(\partial A) = \partial W$, so it follows that the map $F\colon \mathbb{R}^3 \times [0, 1] \to \mathbb{R}^3$ given by $F(x, t) = F_t(x)$ is an ambient isotopy between ∂A and ∂W with compact support. In particular, ∂A and ∂W are homeomorphic. As argued in the paragraphs following the theorem, this now yields Theorem 2.14. □

In the remaining sections of this chapter, we will prove the results used for Theorem 2.14 stated above and construct the necessary objects on which to apply them.

2.4 COMPONENTS OF THE WEDGED RECONSTRUCTION

Let us return first to the proof of Theorem 2.15. We recall:

Theorem 2.15. *There is a 1:1 correspondence between components of A and components of W. The underlying voxel reconstruction V need not be a quasi-manifold.*

Proof. We will argue that Theorem 2.1 in fact yields this similar result with V replaced by W. Consider two arbitrary components V' and V'' of V, and let W' respectively W'' denote their wedged versions in accordance with Definition 2.13. We need to argue that W' and W'' are disjoint components of W. By Definition B.1, W is equal to the union of V with a number of black and white wedges. Each wedge is a compact convex set $C \subset \mathbb{R}^3$. For the black wedges, the set C is a polyhedron which has at least three faces intersecting the boundary of V, so insertion of the four different types of black wedges, $\langle 1\rangle$–$\langle 4\rangle$, preserves connectivity. Consider now (10). The two black voxels of the configuration are in the same component of V by Proposition 2.8. As argued in the proof of Proposition 2.7, two black voxels belonging to the same component of V are connected by a chain of face-adjacent black voxels (tautologically in the same component). The two black voxels in (10) are not face-adjacent, so disconnecting them by removal of the black wedges as illustrated by Figure 2.6$\langle 5\rangle$ does not affect connectivity of the component in which they lie. Hence W' and W'' are connected.

Proposition 2.8 states that V' and V'' are disjoint, so they are separated by at least one white voxel, i.e. for all $v \in V''$, $\delta_{V'}(v) \geq d$, where δ is the distance function defined in (1.2). For the convex hull C corresponding to a wedge of type $\langle 1\rangle$–$\langle 4\rangle$ inserted into a configuration K, any point $c \in C$ satisfies $\delta_K(c) < \varepsilon$, where $\varepsilon > 0$ is given by Definition B.1. We may assume that $\varepsilon/2 < d$, so for all $w \in W''$, $\delta_{W'}(w) \geq d - 2\varepsilon > 0$. Hence W' and W'' are disjoint. In conclusion, the 1:1 correspondence of A and V applies also when V is replaced with W, and this result is independent of whether V is a quasi-manifold. □

For the proof of Corollary 2.16, we need to establish the property of two-sidedness of surfaces embedded in $\mathbb{R}^3$. This follows from the Jordan-Brouwer separation theorem, which is a generalisation of the celebrated Jordan curve theorem to dimension three. First, we have the local version.

Lemma 2.23 **Lemma.** *Let S be a surface in $\mathbb{R}^3$ and let $p \in S$. Then there exists a connected open neighbourhood N of p in $\mathbb{R}^3$ such that $S \cap N$ is connected and S separates N.*

This lemma is proved in [28]. As a consequence, locally, a surface separates Euclidean space into two components. This is also true for the global case.

Theorem 2.24 **Theorem (Jordan-Brouwer separation theorem).** *Let S be a closed connected surface in $\mathbb{R}^3$. Then S separates $\mathbb{R}^3$ into two non-empty open sets.*

The Jordan-Brouwer separation theorem ensures that each component of ∂A and ∂W is two-sided; there is an inside and an outside domain separated by the component, e.g. the domain inside the unit sphere $S^2 \subset \mathbb{R}^3$ is the open unit ball $\mathring{B}^3_1$, and the domain outside is $\mathbb{R}^3 \setminus B^3_1$. The two-sided nature of the boundaries is presumably the natural intuition of such surfaces in dimension three. A proof of the theorem may be found in [28]. However, we are interested in a result for surfaces which are not necessarily connected.

Proposition 2.25 **Proposition.** *Let S be a closed surface in $\mathbb{R}^3$ with k components, $k \in \mathbb{N}$. Then S separates $\mathbb{R}^3$ into $k+1$ non-empty open sets.*

Proof. We prove this by induction over k. The case of $k = 1$ components is exactly the Jordan-Brouwer separation theorem: $\mathbb{R}^3$ is separated into two components. Assume now that k components of S separate $\mathbb{R}^3$ into $k+1$ components.

Let S be a closed surface with $k+1$ components, and let S' be any component of S. Then $S \setminus S'$ has k components, so by our induction hypothesis, $\mathbb{R}^3 \setminus (S \setminus S')$ has $k+1$ components. The component S' must be contained in one of these $k+1$ components since it is not connected to any other component of S. Denote by N the component of $\mathbb{R}^3 \setminus (S \setminus S')$ containing S'. We need to show that $N \setminus S'$ consists of exactly two components. By the Jordan-Brouwer separation theorem, $\mathbb{R}^3 \setminus S'$ is the disjoint union of two open sets A and B in $\mathbb{R}^3$ with common boundary S'; $\mathbb{R}^3 \setminus S' = A \sqcup B$, and $\partial A = \partial B = S'$. Now

$$N \setminus S' = N \cap (\mathbb{R}^3 \setminus S') = N \cap (A \sqcup B) = (N \cap A) \sqcup (N \cap B).$$

We need to show that $N \cap A$ and $N \cap B$ are non-empty and connected. There must be elements in N on either side of S' as N contains S'. Assume without loss of generality that $N \cap A$ is not connected; we wish to show that this leads to a contradiction. There exists a tubular neighbourhood $T_{S'}$ of S' which contains the trivial normal bundle of S'. The boundary of $T_{S'}$ has two components which are both diffeomorphic to S', and one of them is contained in A. Under the assumption that $N \cap A$ is not connected, we can take points x and y in different components of $N \cap A$ and outside of the normal bundle. By definition, N is connected, so there exists a path from x to y in N. Since x and y are not in the same component of $N \cap A$, this path must intersect $N \cap B$ and thus have (at least) two points of intersection with the boundary of $T_{S'}$. But these points of intersection can be joined instead by a path along the connected boundary of $T_{S'}$. This yields a path from x to y which does not intersect S', so x and y must be in the same component of $N \cap A$, and we have reached a contradiction. Hence $N \cap A$ must be connected. By an analogous argument, $N \cap B$ is connected, so S' separates N into two non-empty components. This means that S separates $\mathbb{R}^3$ into $k + 1 + 1 = k + 2$ non-empty open sets. □

With this generalised Jordan-Brouwer separation theorem, we are ready to prove Corollary 2.16 which we re-state here.

Corollary 2.16. *There is a 1:1 correspondence between components of ∂A and components of ∂W.*

Proof. We have established a 1:1 correspondence between the components of A and those of W in Theorem 2.15. Every component $\partial A'$ of ∂A is two-sided and separates $\mathbb{R}^3$ into $\operatorname{int}(A')$ and $\operatorname{int}((A')^C) = (A')^C$. By Proposition 2.25, ∂A separates $\mathbb{R}^3$ into $k+1$ open sets for $k \in \mathbb{N}$ the number of components of ∂A. Due to the 1:1 correspondence of components of A and W, this means that there is a 1:1 correspondence between the components of the boundaries of the sets: boundary components arise exactly as the surfaces separating the inside and outside of components of A respectively W. □

2.5 THE VECTOR FIELD ξ

In this section, we first prove Theorem 2.20, which is in fact a consequence of the Poincaré-Hopf theorem stated below. Subsequently, we commence the construction of a vector field ξ satisfying the prerequisites of Theorem 2.20. Definition 2.28 gives a first, naive suggestion for a candidate ξ_A for the required vector field. Later, we will define a vector field inspired by the smoothing of the voxel reconstruction introduced in Section B.2. This vector field is then merged with ξ_A to finally obtain ξ in Definition 2.32, after which we prove that it has the necessary properties in order for Theorem 2.20 to apply to A and its smoothed reconstruction.

2.5.1 A FIRST SUGGESTION FOR A VECTOR FIELD

Theorem 2.26 **Theorem (Poincaré-Hopf theorem).** *Suppose that M is a compact differentiable n-manifold and that v is a continuous vector field on M which has only isolated zeros $\{x_1, \dots, x_k\}$. Suppose further that v points in the outward normal direction of ∂M. Then*

$$\chi(M) = \sum_{i=1}^{k} \operatorname{ind}_{x_i}(v),$$

where $\operatorname{ind}_{x_i}(v)$ is the index of x_i.

The Poincaré-Hopf theorem was originally proved for $n = 2$ by Henri Poincaré. The proof by Heinz Hopf of the general theorem can be found in [17]. We will not define the concept of index here since we will only make use of its properties. The interested reader is referred to [27].

Theorem 2.20. *Let M be a compact smooth n-manifold without boundary, and let $N \subset M \times (-1, 1)$ be a compact smooth n-manifold without boundary such that N separates $M \times [-1, 1]$, i.e. $M \times [-1, 1] \backslash N$ is a disjoint union of two open sets U, V with $M \times \{-1\} \subset U$ and $M \times \{1\} \subset V$.*

Let v be a continuous nowhere zero vector field on $M \times [-1, 1]$ which points inwards on $M \times \{-1\}$ and outwards on $M \times \{1\}$ and is transverse to N. Then

$$\chi(N) = \chi(M).$$

Proof. Let T be a tubular neighbourhood of N in $M \times (-1, 1)$ such that T is of the form $\phi(N \times [-1, 1])$, where $\phi\colon N \times [-1, 1] \hookrightarrow M \times [-1, 1]$ is an embedding such that $\phi(x, 0) = x$ for all $x \in N$. After shrinking and re-parametrising as necessary, we can suppose that v is transverse to $\phi(N \times \{t\})$ for all $t \in [-1, 1]$.

Let W be a continuous vector field on N with only isolated zeros. We extend to a continuous vector field ω on T given by $\omega(\phi(x, t)) = \mathrm{d}\phi_{(x,t)}(W(x), 0)$ for all pairs of elements $(x,t) \in N \times [-1,1]$, where we use the natural isomorphism $T_{(x,t)}(N \times [-1,1]) = T_xN \oplus T_t[-1,1]$. Define a vector field u on $M \times [-1,1]$ by

$$u(y) = \begin{cases} -v(y) & \text{if } y \in U \backslash T; \\ v(y) & \text{if } y \in V \backslash T; \\ (1-t^2)\omega(y) + tv(y) & \text{if } y = \phi(x,t) \in T. \end{cases}$$

Clearly u is continuous, and the zeros of u are exactly the zeros of W. Denote these zeros by $\{x_1, \ldots, x_k\}$. Notice that u points in the outward direction of the boundary $\phi(N \times \{-1,1\})$ of T and also of the boundary $M \times \{-1,1\}$ of $M \times [-1,1]$. We conclude from the Poincaré-Hopf theorem that

$$\chi(M \times [-1,1]) = \sum_{i=1}^{k} \operatorname{ind}_{x_i}(u) = \chi(T).$$

Since ϕ gives a homeomorphism $N \times [-1,1] \simeq T$,

$$\chi(T) = \chi(N \times [-1,1]).$$

As N and $N \times [-1,1]$ are homotopy equivalent, $\chi(N \times [-1,1]) = \chi(N)$, because the Euler characteristic is homotopy invariant [15]. Similarly, $\chi(M \times [-1,1]) = \chi(M)$, and thus we conclude that

$$\chi(N) = \chi(M).$$ □

The vector field ξ_A should be defined on some domain diffeomorphic to $\partial A \times [-1,1]$.

Definition 2.27 **Definition.** *For f the function defined in (A.3), let $T_{[-1,1]} = f^{-1}[-d\sqrt{3}/2, d\sqrt{3}/2]$, and for A_f defined by (A.4), set $T_{-1} = \partial A_f(-d\sqrt{3}/2)$ and $T_1 = \partial A_f(d\sqrt{3}/2)$.*

It follows from Proposition A.10 that $T_{[-1,1]}$ is diffeomorphic to $\partial A \times [-1,1]$ and has boundary $T_{-1} \sqcup T_1$. In fact, the sets T_{-1} and T_1 are $(r - d\sqrt{3}/2)$-regular by Corollary A.9.

We are now ready to define a vector field on $T_{[-1,1]}$.

Definition 2.28 **Definition.** *Let ξ_A be the vector field on $T_{[-1,1]}$ consisting of unit vectors pointing inwards on T_{-1} and outwards on T_1 and lying along the normals to ∂A.*

This vector field has a behaviour on T_{-1} and T_1 consistent with the requirements of Theorem 2.20. Furthermore, ξ_A has no zeros, since for every point $a \in \partial A$, there exist unique points $a' \in T_{-1}$ and $a'' \in T_1$, which are given by $a' = a + (d\sqrt{3}/2)N(a)$ and $a'' = a - (d\sqrt{3}/2)N(a)$, with $N\colon \partial A \to \mathbb{R}^3$ as in Definition A.5, lying on the same normal of ∂A as a. This follows from Corollary A.9. However, without any knowledge of the boundary of the smoothed reconstruction, it is probably naive to assume that ξ_A is the right candidate for ξ. We need to define a smoothed voxel reconstruction and a vector field on a neighbourhood of its boundary which satisfies the transversality condition in Theorem 2.20. We can then merge this vector field with ξ_A to a vector field ξ on $T_{[-1,1]}$ meeting all of the conditions of the theorem.

2.5.2 Smoothing and transversality

Besides the vector field ξ, we need a smoothed reconstruction Z such that ξ is transverse to ∂Z in order to apply Theorem 2.20. The idea now is to have a particular vector field in mind as we define a smoothing of W so we can construct Z in a way that it becomes transverse to the desired vector field. A first suggestion could be to require that the vector field consist of vectors normal to the boundary faces of W. This is illustrated in the plane in Figure 2.10(a). In the figure, we have not yet taken boundary edges or vertices into consideration; this could for instance be achieved by the modification proposed in Figure 2.10(b), where the blue line segments representing a vector field are transverse to the boundary near the vertex or edge. We can then use a smooth interpolation from the lines normal to the boundary faces to the ones near the edge. A similar idea can be used also to introduce a smoothing of edges: we make a smooth transition from one boundary face to the other near the edge thus maintaining the transversality property of the vector field as illustrated in Figure 2.10(c). The vector field should be non-zero. If we stay close

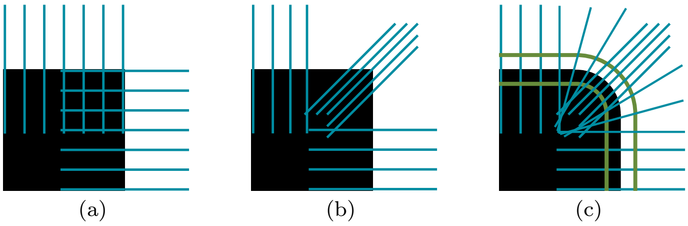


Figure 2.10 Illustration in a plane, parallel to some boundary face of W and intersecting a boundary edge, of a first suggestion for a vector field transverse to a smoothed reconstruction.

enough to the boundary, as indicated by the green lines in Figure 2.10(c), we can avoid any self-intersections of the normal lines. Using these ideas and extrapolating them to vertices, we make a more precise definition of the smoothing of W in Section B.2.

Definition 2.29

Definition. *The* ***smoothed (wedged voxel) reconstruction*** *Z of A is obtained from W via the smoothing procedure defined in Section B.2.*

The smoothed reconstruction Z is now based on the wedged reconstruction W, which in turn depends on the voxel reconstruction V and thus the lattice dL. We say that Z is the smoothed reconstruction of A with respect to dL.

Convention

Convention. For the remainder of this chapter, $Z \subset \mathbb{R}^3$ denotes the smoothed reconstruction of A with respect to dL. ■

In Subsection 2.3.3, we suggested that certain relations between the boundary ∂Z and $T_{[-1,1]}$ and its boundary, defined in Definition 2.27, hold. We now state and prove the proposition of these relations.

Proposition 2.30

Proposition. *We have the following relations between (the boundary of) Z and the set $T_{[-1,1]}$ with boundary $T_1 \sqcup T_{-1}$:*

(i) *The intersections $\partial Z \cap T_1$ and $\partial Z \cap T_{-1}$ are empty;*

(ii) *The boundary ∂Z is a subset of the set $T_{(-1,1)}$;*

(iii) *The boundary ∂Z separates the set $T_{[-1,1]}$.*

Proof. Consider first the reconstruction V of the r-regular set A. A point x on the boundary ∂V is by definition of V contained in the boundary of two adjacent voxels: a black voxel $\mathcal{V}(m) \subset V$ with centre $m \in A$ and a white voxel $\mathcal{V}(m') \not\subset V$ with centre $m' \in A^C$. There may be more choices for m and m' in which case one is chosen arbitrarily. The boundary ∂A must intersect the line segment from m to m' in some point n. We are assuming that no lattice points of dL are contained in ∂A, so n is equal to neither m nor m'. Half the diameter of a voxel equals $d\sqrt{3}/2$, so $|n - x| < d\sqrt{3}/2$, hence $\delta_{\partial A}(x) < d\sqrt{3}/2$ for all $x \in \partial V$, where δ is the distance function from (1.2).

On the other hand, the parallel surfaces T_1 and T_{-1} to ∂A are defined such that any point $y \in T_1 \sqcup T_{-1}$ satisfies $\delta_{\partial A}(y) = d\sqrt{3}/2$. Then clearly $\partial V \cap T_1$ and $\partial V \cap T_{-1}$ are empty. Furthermore, $\partial A \subset T_{[-1,1]}$, and ∂A separates $T_{[-1,1]}$, so if we can show that Proposition 2.30(i) holds, all three properties of the proposition will be satisfied due to the way that Z was obtained from V.

From Definitions B.1 and B.4, we see that the wedges used for defining W and Z are given in terms of coordinates depending on ε for some $\varepsilon > 0$. We can choose this ε such that the resulting boundary stays inside $T_{(-1,1)}$. Smoothing of the k-cells of the wedged object, i.e. W with the additional wedges of (6), is also kept inside an infinitesimal tubular neighbourhood of the boundary according to the smoothing process illustrated by Figure B.6, so again this guarantees that ∂Z is contained in $T_{(-1,1)}$. This concludes the proof. □

The vector field ξ_A from Definition 2.28 is defined on $T_{(-1,1)}$ and is transverse to ∂A since it consists of vectors that are normal to this surface. Having proved that ∂Z is also contained in $T_{(-1,1)}$, we wish to construct another vector field ξ_Z on (a subset of) $T_{(-1,1)}$, such that ξ_Z is transverse to ∂Z. Let $\varepsilon > 0$ and let Z_ε be the image of an embedding of $\partial Z \times [-\varepsilon, \varepsilon]$ in $\mathbb{R}^3$; a tubular neighbourhood of ∂Z. Then $Z_\varepsilon \subset T_{(-1,1)}$ for ε sufficiently small by Proposition 2.30.

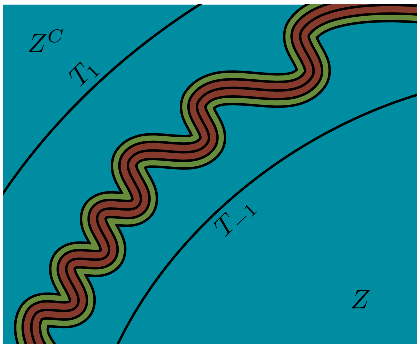


Figure 2.11
The bump function b is 1 on the neighbourhood $Z_{\varepsilon/2}$ (red) of ∂Z (∂Z is the boundary inside the red tube); between 0 and 1 on Z_ε (green), and 0 everywhere else (blue).

Definition 2.31 **Definition.** *For $\varepsilon > 0$, let ξ_Z be the vector field on $Z_\varepsilon \subset T_{(-1,1)}$ consisting of unit vectors pointing from black to white, i.e. from Z to Z^C, and lying along the line segments defined in Definition B.8.*

By definition of the vector field ξ_Z via the direction field ϕ_Z, ξ_Z is transverse to ∂Z; see Proposition B.9.

2.5.3 DEFINITION OF THE VECTOR FIELD

We now wish to merge the vector fields ξ_A from Definition 2.28 and ξ_Z from Definition 2.31. Just as we defined Z_ε, let $Z_{\varepsilon/2}$ be a tubular neighbourhood of ∂Z with half the thickness of Z_ε, and note that $\partial Z \subset Z_{\varepsilon/2} \subset Z_\varepsilon \subset T_{(-1,1)}$. We now define a function $b\colon \mathbb{R}^3 \to \mathbb{R}$ by:

(2.5)
$$\begin{cases} b(x) = 1 & \text{if } x \in Z_{\varepsilon/2}; \\ 0 < b(x) < 1 & \text{if } x \in Z_\varepsilon \setminus Z_{\varepsilon/2}; \\ b(x) = 0 & \text{if } x \in Z_\varepsilon^C; \end{cases}$$

see Figure 2.11 for an illustration of the domain of b. This function is an example of a **(smooth) bump function**, that is, a real-valued function on Euclidean space which is smooth and has compact support.

Definition 2.32 **Definition.** *Letting $b\colon \mathbb{R}^3 \to \mathbb{R}$ be the bump function given by (2.5), we define a vector field ξ on $T_{[-1,1]}$ by*

(2.6)
$$\xi = b\xi_Z + (1-b)\xi_A.$$

Proposition 2.33 **Proposition.** *The vector field ξ of Definition 2.32 is a continuous nowhere zero vector field on $T_{[-1,1]} \cong \partial A \times [-1,1]$, which points inwards on T_{-1} and outwards on T_1 and is transverse to ∂Z.*

Proof. Continuity of ξ is ensured by construction, since ξ_A and ξ_Z are continuous vector fields which are glued together using a continuous (smooth) bump function. On T_{-1} and T_1, ξ equals ξ_A; compare (2.5) and (2.6), so by Definition 2.28, ξ has the wanted direction on these surfaces. Similarly, ξ inherits the property of transversality to ∂Z from ξ_Z, since these vector fields are equal on $Z_{\varepsilon/2} \supset \partial Z$, and ξ_Z is transverse to ∂Z. It remains to be shown that ξ is nowhere zero.

The vector fields ξ_A and ξ_Z are both nowhere zero, so the only circumstance under which a zero of ξ can arise is if two vectors, one from each of the two vector fields, of opposite directions were to coincide inside $Z_\varepsilon \setminus Z_{\varepsilon/2}$ where the smooth transition is made. Let us assume that we have vectors $v \in \xi_Z$ and $w \in \xi_A$ such that $v = -w$ in Z_ε and show that this leads to a contradiction. Let $\lceil n \rfloor$ denote the configuration of Z containing v and w.

Being a vector from ξ_Z, v lies on some line segment L_v in $\lceil n \rfloor$ with endpoints on the boundary of $\lceil n \rfloor$. By definition of the direction field of ξ_Z, L_v has endpoints on opposing faces, edges, or vertices of $\lceil n \rfloor$. This means that the length of L_v is at least $2d$, i.e. two times the side length of a voxel. If $n \neq 10$, L_v intersects ∂Z exactly once, so one end lies in Z and the other in Z^C. This case is illustrated in Figure 2.12(a). If $n = 10$, there is the additional possibility that ∂Z intersects L_v twice so that both ends are inside Z and one part in between lies in Z^C. In this last case, the vector field is defined such that L_v intersects the midpoint m of the black voxel containing the point of origin of L_v in $\lceil 10 \rfloor$; see Figure 2.12(b). Equipping L_v with the direction of v, L_v originates in a black point

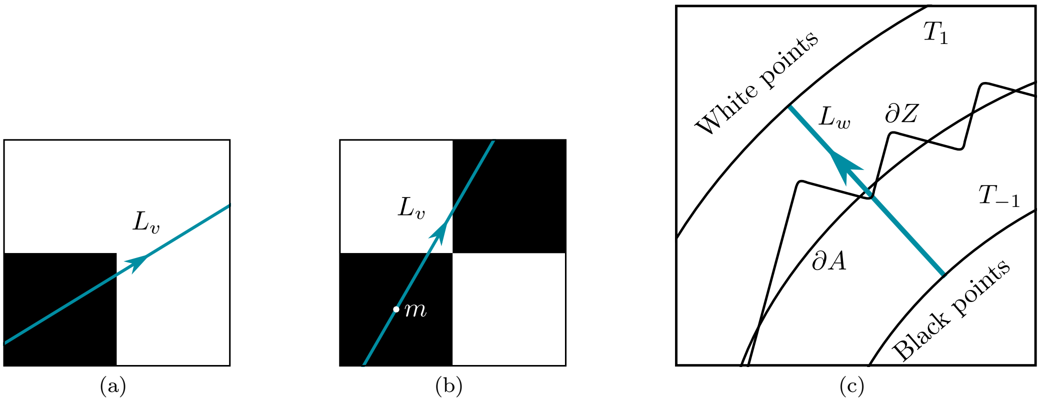


Figure 2.12 (a) shows an example of the first type of line segment L_v along which vectors in ξ_Z lie; (b) the second, relevant only for $\lceil 10 \rfloor$. (c) shows an example of a line segment L_w along which vectors in ξ_A lie.

and terminates in either a white point or, as remarked, for some cases of $\lceil 10 \rfloor$, in a black point. If the latter is the case, the black point of termination belongs to the black voxel in $\lceil 10 \rfloor$ which does not have m as midpoint.

We can make similar observations for the vector $w \in \xi_A$. This vector lies on a line segment L_w from T_{-1} to T_1 of length $d\sqrt{3}$; see Definition 2.27. Given the direction of w, L_w originates in a black point and terminates in a white point, since $T_{-1} \subset Z$ and $T_1 \subset Z^C$ as a consequence of Proposition 2.30. This is illustrated in Figure 2.12(c).

By assumption, $v = -w$ lies on $L_v \cap L_w$. Consider first the case where L_v intersects ∂Z exactly once. Now L_v is directed from black to white, so L_w must have the opposite direction. Since L_w has to originate in a black point, this point must lie outside of $\lceil n \rfloor$. Analogously, since L_w terminates in a white point, this point must be outside $\lceil n \rfloor$ as well. This means L_w must traverse the entire length of L_v through $\lceil n \rfloor$, but this is a contradiction as $d\sqrt{3} < 2d$.

In the case where L_v intersects ∂Z twice, we know that $n = 10$. The line segment L_w must terminate in a white point. The black voxel midpoint m that L_v intersects is contained in A, so the distance from m to T_1 is $\delta_{T_1}(m) > d\sqrt{3}/2$ because $m \notin \partial A$. Hence T_1 lies outside of this black voxel, and so L_w must terminate beyond m outside of $\lceil 10 \rfloor$. We need to check whether L_w might originate in a black point contained in the other black voxel of $\lceil 10 \rfloor$. Points in this black voxel all have distance greater than $d\sqrt{3}/2$ to m, which in turn has distance greater than $d\sqrt{3}/2$ to T_1, where both distances are along L_w. However, L_w has length $d\sqrt{3}$ which does not suffice, and hence we reach another contradiction.

We should also consider the complement of $\lceil 10 \rfloor$. Here, we can use a completely analogous argument with 'black' and 'white' interchanged and T_{-1} in place of T_1.

In conclusion, vectors of ξ_A and ξ_Z with opposite directions cannot coincide in $Z_\varepsilon \setminus Z_{\varepsilon/2}$, so ξ has no zeros. □

By Proposition 2.33, Definition 2.32 thus finally yields the vector field we need in order to apply Theorem 2.20 to ∂A and ∂Z. As argued in Subsection 2.3.3, this means that there exists a homeomorphism and an ambient isotopy between A and W as well as their boundaries; the main result of this chapter.

For convenience, we assumed early on that no sampling points of the lattice L lie on ∂A and claimed that this would have no influence on our results. We argue now that in the case where a sampling point intersects the boundary, we can move the boundary of A just slightly, say $\delta > 0$, in one of the normal directions. The resulting set A' is then $(r - \delta)$-regular. As long as we choose δ such that $\partial A'$ still avoids the boundary of the wedged reconstruction, the results of this chapter also apply to A'. Moving the boundary to either side of the sampling point naturally affects the reconstruction W, since the corresponding voxel will be either black or white, and the two different choices represent different reconstructions. They are, however, homeomorphic.

2.6 DISCUSSION

In this chapter, we have shown that for a sufficiently high resolution, $d^{-1} > \sqrt{3}/(2r)$, r-regular sets are ambient isotopic to their wedged reconstructions with respect to a lattice dL. For the wedged reconstruction to be well-defined, we need the reconstruction of the r-regular set to be a quasi-manifold which is only guaranteed for a slightly higher resolution, $d^{-1} > 0.95571/r$. The next step would thus be to examine if Conjecture 2.11

holds: whether a resolution $d^{-1} > \sqrt{3}/(2r)$ in fact implies that the voxel reconstruction of an r-regular set is a quasi-manifold.

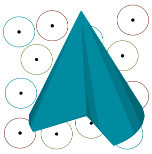
In addition, we are interested in reconstructing geometric features of r-regular sets such as surface area, volume, and curvature. This task is partly commenced in the following chapter, where we use an entirely different approach to the reconstruction of geometry, namely via estimators of the Minkowski tensors of sets with positive reach through digital algorithms. As noted in the introduction, r-regular sets have positive reach. The work presented in the next chapter only concerns planar sets, however, but the theory on which the algorithms are based is equally valid in higher dimensions.

We remark that the results of this chapter are only valid for black-and-white voxel reconstructions, whereas in applications, data may be given as greyscale images. Here, one assigns an intensity ranging from 0 to 1 to each sampling point of the lattice depending on the overlap of the object with the voxel corresponding to the sampling point: If the voxel is completely contained within the object, it is assigned a 1, corresponding to a black voxel, and if the voxel does not overlap with the object at all, a 0 is assigned, corresponding to a white voxel. Denote by O the union of black voxels and by N the union of white voxels. The boundaries ∂O and ∂N do not coincide but are both approximations to the boundary of the underlying object A. The question is whether O and N have the same topology as A in this setting. We are then back to considering the 14 voxel configurations of Figure 2.3, where white sampling points should be replaced by grey ones for configurations of O and black sampling points should be replaced by grey ones for configurations of N. It is likely possible to thin the number of configurations for r-regular sets depending on the size of the resolution as is done in Appendix A in the three-dimensional case. Higher resolutions are presumably needed here.

Finally, the work presented in this chapter only applies in dimension three. In [10], du Plessis has shown similar results for r-regular sets in dimension two and in fact also results concerning the reconstruction of geometry. However, generalising the theory from dimension two to dimension three poses considerable mathematical difficulties due to the combinatorics of the relation of the boundary to the lattice, represented by the voxel configurations, as well as the differential geometry of the boundary. It is not readily clear how results similar to those obtained in dimensions two and three could be achieved for even higher dimensions.

Three

COMPARISON OF TWO GLOBAL ALGORITHMS FOR MINKOWSKI TENSOR ESTIMATION



Traditional digital geometry [22] uses binary information to assess volume, surface area, or, more generally, intrinsic volumes of (sufficiently regular) sets $A \subset \mathbb{R}^n$. More recently, the estimation of Minkowski tensors has been suggested, as tensor-valued valuations allow for quantification of location- and orientation-related properties of A.

In [19], algorithms for the estimation of Minkowski tensors of sets with positive reach from digitisations are introduced. The derivation of the algorithms is based on a generalised Steiner formula, and the estimators can be shown to converge to the true tensors as the resolution d of the digitisation tends to infinity. This property is called multigrid convergence, and the results on multigrid convergence of the estimators are presented as Theorems 3.1 and 3.2 of this thesis.

In this chapter, we implement two of the algorithms described in [19] in MATLAB. In dimension n and for a set $A \subset \mathbb{R}^n$ with positive reach, $\text{Reach}(A) > 0$, one algorithm depends on the choice of $n+1$ radii $0 < R_0 < \cdots < R_n < \text{Reach}(A)$; the other on n radii $0 < R_0 < \cdots < R_{n-1} < \text{Reach}(A)$. We have implemented the algorithms in MATLAB in dimension $n = 2$. Based on simulations for both convex and non-convex test sets with positive reach via our digital algorithms, we will give recommendations for the choices of the radii on which the algorithms are based. Further, we explore the consequences of making erroneous choices of the radii, i.e. choosing them larger than $\text{Reach}(A)$. We also discuss a possible lower limit $d/\sqrt{2}$ for the radii. Finally, both algorithms are multigrid convergent, and we wish to explore how well the theory carries over when the algorithms are applied in the realistic setting of finite resolution.

In Section 3.1, we give the definition of Minkowski tensors. Section 3.2 presents the first algorithm and its implementation as well as a review of the method for obtaining the estimators via a generalised Steiner formula. In Section 3.3 follows the second algorithm and corresponding implementation. Having established the algorithms, we proceed to perform simulations on test sets in Section 3.4 and examine the theoretical properties of the algorithms in applications. In Section 3.5 we give recommendations for the choices of input arguments for the algorithms based on our findings in Section 3.4, and we suggest a third algorithm, which combines the strengths of the former two algorithms to produce the best possible estimators in this framework.

This chapter is a revised version of the electronic preprint [8], which has been modified slightly to fit into this thesis. Moreover, further details, such as additional computations and documentation of the code, are added in Appendices C and D.

3.1 MINKOWSKI TENSORS

One purpose of image processing is the extraction of geometric characteristics of an object from a digital image. We give here a short introduction to a rather general class of such characteristics; the so-called Minkowski tensors. For a more thorough review, the reader may consult the excellent book [32] by Schneider.

For $p \in \mathbb{N}_0$, let $\mathbb{T}^p$ denote the space of symmetric tensors of rank p on $\mathbb{R}^n$. Using the scalar product to identify $\mathbb{R}^n$ with its dual, an element T of $\mathbb{T}^p$ defines a symmetric p-linear functional $T\colon (\mathbb{R}^n)^p \to \mathbb{R}$. Any $T \in \mathbb{T}^p$ is determined by the numbers

$$T_{(i_1 \dots i_p)} = T(e_{i_1}, \dots, e_{i_p}), \tag{3.1}$$

$i_1, \dots, i_p \in \{1, \dots, n\}$, where $e_1, \dots, e_n$ is the standard basis in $\mathbb{R}^n$. In particular, a tensor of rank zero, one, and two can be identified with a scalar, a vector in $\mathbb{R}^n$, and a symmetric matrix of size $n \times n$, respectively. More generally, a tensor of rank p can be identified with a symmetric array of size n^p. When calculating tensors later on, it will be the entries of these arrays that we calculate.

Denote by x^r the r-fold tensor product of $x \in \mathbb{R}^n$, which is given by

$$x^r(v_1, \dots, v_r) = \prod_{i=1}^{r} \langle x, v_i \rangle, \qquad v_1, \dots, v_r \in \mathbb{R}^n,$$

and by ab the symmetric tensor product of symmetric tensors a and b. We use the convention that $x^0 = 1$.

For a compact subset A of $\mathbb{R}^n$ and $r \in \mathbb{N}_0$, we define an element of $\mathbb{T}^r$, called the ***r*th volume tensor** of A, by

$$\Phi_n^{r,0}(A) = \frac{1}{r!} \int_A x^r \, \mathrm{d}x. \tag{3.2}$$

Notice in particular that $\Phi_n^{0,0}(A)$ is simply the volume $V_n(A)$ of A. For $s \in \mathbb{N}$, we set $\Phi_n^{r,s}(A) = 0$. More general tensors related to A can be defined by integrating over boundary points and outer normal vectors when A is a set with positive reach.

Indeed, let $A \subset \mathbb{R}^n$ be a set with positive reach. For $k \in \{0, \dots, n-1\}$ and $r, s \in \mathbb{N}_0$, we define elements of $\mathbb{T}^{r+s}$ by

$$\Phi_k^{r,s}(A) = \frac{1}{r!s!} \frac{\omega_{n-k}}{\omega_{n-k+s}} \int_{\mathbb{R}^n \times S^{n-1}} x^r u^s \Lambda_k(A; \mathrm{d}(x,u)), \tag{3.3}$$

where ω_j is the surface area of S^{j-1}, and $\Lambda_k(A; \cdot)$ is the generalised curvature measure of A introduced in [40]. The tensors in (3.3) are the **surface tensors** of A.

For all $k \in \{0, \dots, n\}$ and $r, s \in \mathbb{N}_0$, $\Phi_k^{r,s}(A)$ are the **Minkowski tensors** of A. For $r = s = 0$ and $k \in \{0, \dots, n\}$, the Minkowski tensor $\Phi_k^{0,0}(A)$ coincides with the kth intrinsic volume $V_k(A)$ of A; see for instance [32], as was already noted in the case of the volume, $V_n(A)$. In particular, $2V_{n-1}(A)$ is the surface area of A when A does not have lower dimensional parts, and $V_0(A)$ is the Euler-Poincaré characteristic of A. By restricting the integrations in (3.2) and (3.3) to $B \cap A$ respectively $B \times S^{n-1}$ with $B \subset \mathbb{R}^n$ a Borel set, local Minkowski tensors can be defined as suggested in [18]. Although the digital algorithms for estimating Minkowski tensors [19] are formulated using the local versions, we restrict our considerations to the estimation of Minkowski tensors of the form (3.2) and (3.3).

3.2 MINKOWSKI TENSOR ESTIMATION FROM $n+1$ PARALLEL SETS

Steiner's formula, in its version for sets $A \subset \mathbb{R}^n$ with positive reach, shows that the parallel volume $V_n(A^R)$, where A^R is the R-parallel set of A defined in (1.4), is a polynomial in R of degree at most n as long as $0 \le R < \mathrm{Reach}(A)$. Up to constants, its coefficients coincide with the intrinsic volumes $V_0(A), \dots, V_n(A)$. This allows for the computation (estimation) of the intrinsic volumes when $V_n(A^R)$ is (approximately) known for different radii R. Replacing the parallel volume with the Voronoi tensor measures, to be defined below, this idea can be extended to an estimation procedure for Minkowski tensors.

3.2.1 THE VORONOI TENSOR MEASURES

For a compact set $A \subset \mathbb{R}^n$, for $r, s \in \mathbb{N}_0$, and for $R \ge 0$, the formula

$$\mathcal{V}_R^{r,s}(A) = \int_{A^R} p_A(x)^r (x - p_A(x))^s \, \mathrm{d}x \tag{3.4}$$

defines $\mathbb{T}^{r+s}$-valued measures $\mathcal{V}_R^{r,s}(A)$. These are called the (total) **Voronoi tensor measures** and were first introduced by Hug, Kiderlen, and Svane in [19]. In the case where A has positive reach, $0 \le R < \mathrm{Reach}(A)$, a Steiner-type formula implies that

$$\mathcal{V}_R^{r,s}(A) = r!s! \sum_{k=0}^{n} \kappa_{s+k} R^{s+k} \Phi_{n-k}^{r,s}(A), \tag{3.5}$$

where κ_j is the volume of the unit ball $B_1^j(0)$ in $\mathbb{R}^j$.

For $n+1$ different choices of R, $0 < R_0 < \cdots < R_n < \mathrm{Reach}(A)$, equation (3.5) gives rise to a system of $n+1$ linear equations for each pair $(r,s) \in \mathbb{N}_0^2$;

(3.6)
$$\begin{pmatrix} \mathcal{V}_{R_0}^{r,s}(A) \\ \vdots \\ \mathcal{V}_{R_n}^{r,s}(A) \end{pmatrix} = M_{R_0,\dots,R_n}^{r,s} \begin{pmatrix} \Phi_n^{r,s}(A) \\ \vdots \\ \Phi_0^{r,s}(A) \end{pmatrix},$$

with the matrix
$$M_{R_0,\dots,R_n}^{r,s} = r!s! \begin{pmatrix} \kappa_s R_0^s & \dots & \kappa_{s+n} R_0^{s+n} \\ \vdots & \ddots & \vdots \\ \kappa_s R_n^s & \dots & \kappa_{s+n} R_n^{s+n} \end{pmatrix}.$$

Since $R_i > 0$ for $i \in \{0,\dots,n\}$, the matrix $M_{R_0,\dots,R_n}^{r,s}$ is a Vandermonde-type matrix, and so in particular invertible, hence (3.6) can be solved for the Minkowski tensors;

(3.7)
$$\begin{pmatrix} \Phi_n^{r,s}(A) \\ \vdots \\ \Phi_0^{r,s}(A) \end{pmatrix} = \left(M_{R_0,\dots,R_n}^{r,s}\right)^{-1} \begin{pmatrix} \mathcal{V}_{R_0}^{r,s}(A) \\ \vdots \\ \mathcal{V}_{R_n}^{r,s}(A) \end{pmatrix}.$$

We wish to utilise (3.7) for the estimation of Minkowski tensors from digitisations. To this end, suitable estimators of the Voronoi tensor measures on the right-hand side of (3.7) must be introduced.

3.2.2 Estimators of Minkowski tensors from digitisations

Let dL be a lattice. Given a digitisation $A_0 = A \cap dL$ of a set $A \subset \mathbb{R}^n$ with positive reach and $\mathrm{Reach}(A) > R > 0$, we approximate the Voronoi tensor measure $\mathcal{V}_R^{r,s}(A)$, $r,s \in \mathbb{N}_0$, by the corresponding quantity $\mathcal{V}_R^{r,s}(A_0)$ for A_0. For $0 < R_0 < \cdots < R_n < \mathrm{Reach}(A)$, this yields estimators

(3.8)
$$\begin{pmatrix} \hat{\Phi}_n^{r,s}(A_0) \\ \vdots \\ \hat{\Phi}_0^{r,s}(A_0) \end{pmatrix} = \left(M_{R_0,\dots,R_n}^{r,s}\right)^{-1} \begin{pmatrix} \mathcal{V}_{R_0}^{r,s}(A_0) \\ \vdots \\ \mathcal{V}_{R_n}^{r,s}(A_0) \end{pmatrix}$$

for the Minkowski tensors; compare (3.7). The justification for replacing $\mathcal{V}_R^{r,s}(A)$ with its discrete counterpart $\mathcal{V}_R^{r,s}(A_0)$ is given in [19, Theorem 4.3], which implies that $\mathcal{V}_R^{r,s}(A_0)$ converges to $\mathcal{V}_R^{r,s}(A)$ as $d \to 0^+$ under weak assumptions on A. This yields the following convergence theorem result for $\hat{\Phi}_R^{r,s}(A_0)$ when A is topologically regular, that is, A is the closure of its interior.

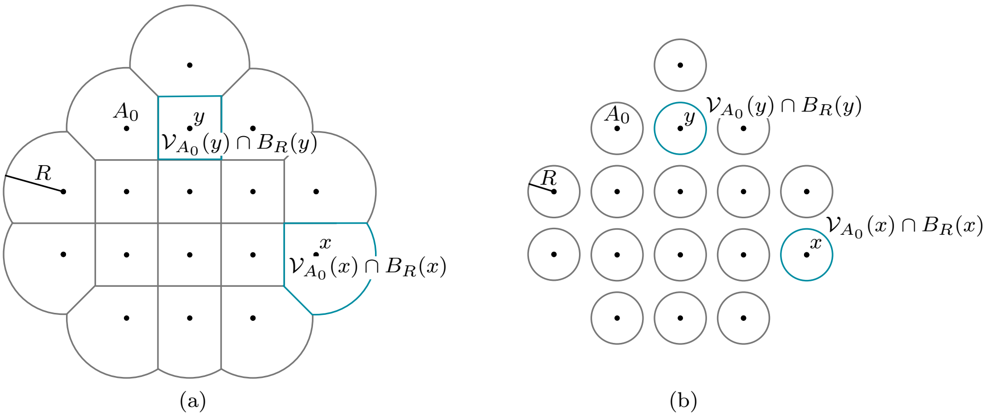
Theorem 3.1 **Theorem.** *Let $A \subset \mathbb{R}^n$ be compact and topologically regular. If $\hat{\Phi}_k^{r,s}(A_0)$ is defined by equation (3.8) with $A_0 = A \cap dL$ and $0 < R_0 < \cdots < R_n < \mathrm{Reach}(A)$, then $\hat{\Phi}_k^{r,s}(A_0)$ is **multigrid convergent** to $\Phi_k^{r,s}(A)$, which means*

(3.9)
$$\lim_{d \to 0^+} \hat{\Phi}_k^{r,s}(A_0) = \Phi_k^{r,s}(A).$$

That A must be topologically regular seems a reasonable restriction when working with digitisations, since the lower dimensional parts of an object are generally not visible in the digital image A_0. Note that topological regularity is automatically guaranteed for the ρ-regular (r is reserved for other purposes in this chapter) sets in Chapter 2.

A proof of Theorem 3.1 can be found in [19, Corollary 5.2]. If, in addition, A is convex or ρ-regular, the convergence in (3.9) is of order $O(d)$ when $r = s = 0$, that is, for estimators of the intrinsic volumes. For $r, s \neq 0$, convergence is known to be of order $O(\sqrt{d})$.

Equation (3.8) gives a set of estimators of all the Minkowski tensors, but alternative estimators exist in the case of the volume tensors given by (3.2). We will return to these considerations in Section 3.3, where we introduce an additional algorithm which exploits this fact. For now, we can use (3.8) to compute estimators of the Minkowski tensors of a set $A \subset \mathbb{R}^n$ with positive reach from a digitisation A_0. In the next subsection, we will explain how the Voronoi tensor measures can be computed for discrete sets.


A_0
y
$\mathcal{V}_{A_0}(y) \cap B_R(y)$
R
x
$\mathcal{V}_{A_0}(x) \cap B_R(x)$
A_0
y
$\mathcal{V}_{A_0}(y) \cap B_R(y)$
R
$\mathcal{V}_{A_0}(x) \cap B_R(x)$
x

(a) (b)

Figure 3.1 Comparison in dimension two of R-bounded Voronoi cells with respect to a digitisation $A_0 \subset dL$. In (a), $R > d/\sqrt{2}$, and in (b), $R < d/\sqrt{2}$.

3.2.3 Implementation in dimension two

For $x \in A_0$ and $R > 0$, we have the relation $\mathcal{V}_{A_0}(x) \cap A_0^R = \mathcal{V}_{A_0}(x) \cap B_R(x)$, and we will refer to this intersection as the ***R*-bounded Voronoi cell** of $x \in A_0$ with respect to A_0. The space $\mathbb{R}^n$ coincides with the finite union of the Voronoi cells of A_0, so A_0^R coincides with the union of R-bounded Voronoi cells of A_0. This yields a simplification of (3.4) for the discrete set A_0,

$$\mathcal{V}_R^{r,s}(A_0) = \sum_{x \in A_0} x^r \int_{\mathcal{V}_{A_0}(x) \cap B_R(x)} (y-x)^s \, \mathrm{d}y. \tag{3.10}$$

Thus, the Voronoi tensor measures for discrete digitisations can be reduced to a sum of contributions from each element of A_0, and these contributions depend only on the corresponding R-bounded Voronoi cells of A_0. The right-hand side of equation (3.8) is now easily computed using (3.10). This will be our first algorithm.

We choose to focus on dimension two for the implementation of the algorithm given by (3.8). In this case, (3.8) becomes

$$\begin{pmatrix} \hat{\Phi}_2^{r,s}(A_0) \\ \hat{\Phi}_1^{r,s}(A_0) \\ \hat{\Phi}_0^{r,s}(A_0) \end{pmatrix} = \left(M_{R_0,R_1,R_2}^{r,s} \right)^{-1} \begin{pmatrix} \mathcal{V}_{R_0}^{r,s}(A_0) \\ \mathcal{V}_{R_1}^{r,s}(A_0) \\ \mathcal{V}_{R_2}^{r,s}(A_0) \end{pmatrix} \tag{3.11}$$

with $r, s \in \mathbb{N}_0$, $0 < R_0 < R_1 < R_2 < \mathrm{Reach}(A)$, and

$$M_{R_0,R_1,R_2}^{r,s} = r!s! \begin{pmatrix} \kappa_s R_0^s & \kappa_{s+1} R_0^{s+1} & \kappa_{s+2} R_0^{s+2} \\ \kappa_s R_1^s & \kappa_{s+1} R_1^{s+1} & \kappa_{s+2} R_1^{s+2} \\ \kappa_s R_2^s & \kappa_{s+1} R_2^{s+1} & \kappa_{s+2} R_2^{s+2} \end{pmatrix}.$$

In practice, one is usually interested in the tensors of rank at most two, i.e. tensors for which $r + s \leq 2$. The relevant volumes κ_j of the unit ball in $\mathbb{R}^j$ are thus the ones with $j \in \{0, \ldots, 4\}$:

$$\kappa_0 = 1, \qquad \kappa_1 = 2, \qquad \kappa_2 = \pi, \qquad \kappa_3 = 4\pi/3, \qquad \kappa_4 = \pi^2/2. \tag{3.12}$$

From [32, (4.27),(5.18), and (5.30)], it follows that for $k \in \{0, \ldots, n-1\}$, the tensor $\Phi_k^{0,1}(A) = 0$ is trivial; see Lemma C.1. It is therefore not necessary to use approximations $\hat{\Phi}_k^{r,s}(A_0)$ in these cases. Nonetheless, our implementation calculates and reports these estimators, since values deviating considerably from the origin indicate that the resolution is not sufficiently high.

For a given digitisation A_0 of a set $A \subset \mathbb{R}^2$, the algorithm determines $\hat{\Phi}_k^{r,s}(A_0)$ by (3.10) and (3.11) based on three fixed radii $0 < R_0 < R_1 < R_2 < \mathrm{Reach}(A)$. The assumption that all radii must be smaller than $\mathrm{Reach}(A)$ requires the knowledge of (a positive lower bound of) the reach of A. Since the reach may not be accessible in applications, in the simulation section, Section 3.4, we also analyse the behaviour of the estimators when one or more radii are larger than $\mathrm{Reach}(A)$.

Consider Figure 3.1. The asymptotic result in Theorem 3.1 is based on increasing resolution $d \to 0^+$. Hence any given radius $R > 0$ is eventually larger than half the diameter $d\sqrt{2}$ of a pixel, implying that the union of all R-bounded Voronoi cells of points

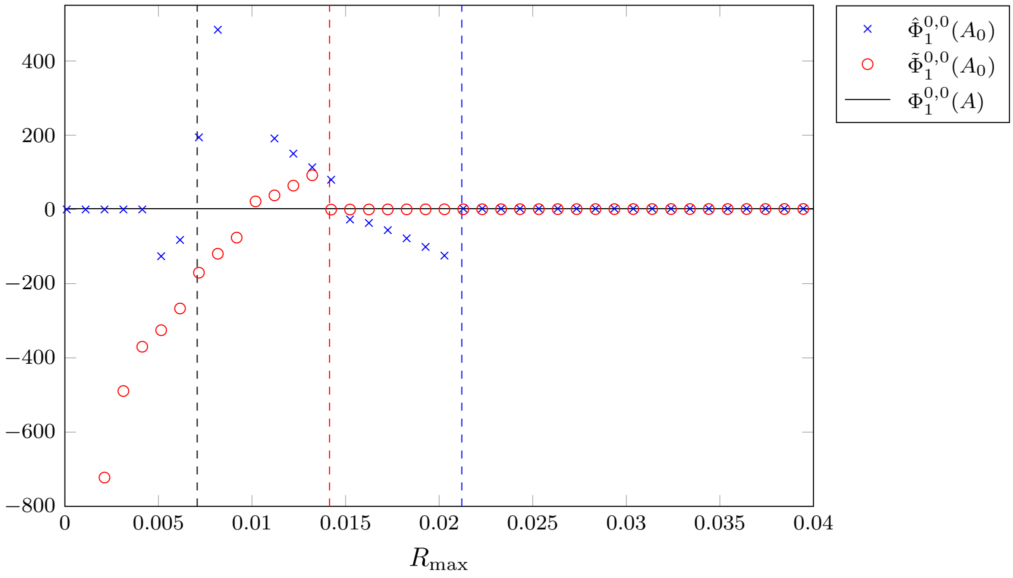


Figure 3.2 The estimators $\hat{\Phi}_1^{0,0}(A_0)$ and $\tilde{\Phi}_1^{0,0}(A_0)$ as functions of $R_{\max}$ for fixed resolution $d^{-1} = 100$ and different radii for the square $A = [0,1]^2$ with $R_0 = R_{\max}/n$ for $\mathcal{R}_n$, $n = 2, 3$. Here $\mathcal{R}_3$ denotes the algorithm presented in Section 3.2.3, $\mathcal{R}_2$ denotes the second algorithm to be introduced in Section 3.3, and $R_{\max}$ denotes the maximal radius R_{n-1} for $\mathcal{R}_n$. The third radius, R_1, of $\mathcal{R}_3$ is given by $R_1 = (R_0 + R_2)/2$. The second algorithm, $\mathcal{R}_2$, only depends on two radii. The dashed black line indicates the point when $R_{\max}$ is greater than $d/\sqrt{2}$; the coloured dashed lines when all radii of the corresponding algorithm are above $d/\sqrt{2}$.

$x \in A_0$ covers the Gauss-digitisation of A completely. On the other hand, if $R < d/\sqrt{2}$ for some $d > 0$, the R-bounded Voronoi cell of an inner point does not cover its pixel. In this case the replacement of $\mathcal{V}_R^{r,s}(A)$ by $\mathcal{V}_R^{r,s}(A_0)$ in the transition from (3.7) to (3.8) may cause unwanted errors. In fact, Figure 3.2 shows the behaviour of half the perimeter estimator $\hat{V}_1(A_0) = \hat{\Phi}_1^{0,0}(A_0)$ of the unit square $A = [0,1]^2$ when the radii vary in the vicinity of $d/\sqrt{2}$. The figure clearly illustrates the described effect. This effect can be avoided by choosing $R_0 > d/\sqrt{2}$, and we shall do so for the remainder of this chapter.

When $R > d/\sqrt{2}$, one can simplify the computations in the algorithm in the following way. Under this assumption, the R-bounded Voronoi cell $\mathcal{V}_{A_0}(x) \cap B_R(x)$ of an inner point $x \in A_0$ coincides with $\mathcal{V}_{dL}(x)$. Hence the contributions of all inner points of the digitisation to the integral $\int_{\mathcal{V}_{A_0}(x)\cap B_R(x)} (y - x)^s \, dy$ are the same and vanish when s is odd. This observation is exploited in the implementation of the algorithm in order to reduce computation times, which is discussed further in Subsection D.2.3.

With the aforementioned choices, we obtain the following algorithm, where we restrict attention to estimators of Minkowski tensors of rank at most two. These tensors have been shown to be most relevant in practical applications such as those suggested at the beginning of Chapter 1. An extension to higher rank tensors is straightforward and only requires minor modifications in the code.

Algorithm Minktensor2D3R ($\mathcal{R}_3$)

Input:

- Digitisation $A_0 = A \cap dL$ of a compact topologically regular set $A \subset \mathbb{R}^2$ with positive reach;
- Lattice distance $d > 0$ of dL;
- Two radii R_0 and R_2 satisfying $0 < R_0 < R_2 < \mathrm{Reach}(A)$.

Action:

- Calculate $\mathcal{V}_{R_i}^{r,s}(A_0)$ using (3.10) for $i \in \{0,1,2\}$, with $R_1 = (R_0 + R_2)/2$, and for all $r, s \in \{0,1,2\}$ with $r + s \le 2$;
- Determine the corresponding Minkowski tensors by (3.11).

Output:

- Estimators $\hat{\Phi}_k^{r,s}(A_0)$ for all $k, r, s \in \{0,1,2\}$ with $r+s \le 2$ (including the trivial ones of the form $\hat{\Phi}_k^{0,1}(A_0)$ for model control) of the corresponding Minkowski tensors.

The programme will give a warning if radii are chosen below the recommended lower limit $d/\sqrt{2}$ as discussed above. If any result of the algorithm is numerically below 10^{-6}, the output is rounded off to zero.

3.3 MINKOWSKI TENSOR ESTIMATION FROM n PARALLEL SETS

As proved in [19], it is possible to refine the algorithm introduced in Subsection 3.2.2 by using simpler estimators of the volume tensors defined in (3.2). Since we require that the set $A \subset \mathbb{R}^n$ be compact and topologically regular, for $r \in \mathbb{N}_0$, one can use the estimators

(3.13)
$$\tilde{\Phi}_n^{r,0}(A_0) = \frac{d^n}{r!} \sum_{x \in A_0} x^r$$

for the volume tensors. This is the natural generalisation of the usual volume estimator in the case $r = 0$, which counts foreground pixels; see for instance [29]. We have the following result on convergence from [19, Section 5.1].

Theorem 3.2

Theorem. *Let $A \subset \mathbb{R}^n$ be compact and topologically regular, and suppose that ∂A is a Lebesgue null set. Then $\tilde{\Phi}_n^{r,0}(A_0)$, defined by equation (3.13) with $A_0 = A \cap dL$, is multigrid convergent to $\Phi_n^{r,0}(A)$, i.e.*

(3.14)
$$\lim_{d \to 0^+} \tilde{\Phi}_n^{r,0}(A_0) = \Phi_n^{r,0}(A).$$

The assumption that ∂A is a Lebesgue null set is weaker than that of positive reach, so these estimators converge towards the true volume tensors for a larger class of sets than $\hat{\Phi}_n^{r,0}(A_0)$ do. Moreover, they do not require any knowledge of the reach of A. If, in addition, ∂A is an $(n-1)$-rectifiable set, i.e. if ∂A is the image of a bounded subset of $\mathbb{R}^{n-1}$ under a Lipschitz map, the convergence in (3.14) is of order $O(d)$. This condition is for instance satisfied by ρ-regular sets.

In addition to their advantageous convergence properties, using the estimators in (3.13) of the volume tensors, we can derive an alternative algorithm for computation of estimators of the Minkowski tensors of sets A with positive reach which is based on only n fixed radii $0 < R_0 < \cdots < R_{n-1} < \text{Reach}(A)$.

Indeed, let $0 < R < \text{Reach}(A)$ and consider the Voronoi tensor measures $\mathcal{V}_R^{r,s}(A)$ defined by (3.5). Subtract $r!s!\kappa_s R^s \Phi_n^{r,s}(A)$ from both sides of the equation to obtain

$$\mathcal{V}_R^{r,s}(A) - r!s!\kappa_s R^s \Phi_n^{r,s}(A) = r!s! \sum_{k=1}^{n} \kappa_{s+k} R^{s+k} \Phi_{n-k}^{r,s}(A).$$

In the case $s \in \mathbb{N}$, $\Phi_n^{r,s}(A) = 0$, so this equation takes the simple form

(3.15)
$$\mathcal{V}_R^{r,s}(A) = r!s! \sum_{k=1}^{n} \kappa_{s+k} R^{s+k} \Phi_{n-k}^{r,s}(A), \qquad s \in \mathbb{N},$$

and when $s = 0$,

$$\mathcal{V}_R^{r,0}(A) - r!\Phi_n^{r,0}(A) = r! \sum_{k=1}^{n} \kappa_k R^k \Phi_{n-k}^{r,0}(A).$$

Similarly to the approach used in Subsection 3.2.1, estimators of the Minkowski tensors are now obtained from this formula combined with (3.10) and (3.15):

(3.16)
$$\begin{pmatrix} \tilde{\Phi}_{n-1}^{r,s}(A_0) \\ \vdots \\ \tilde{\Phi}_0^{r,s}(A_0) \end{pmatrix} = \left(N_{R_0,\ldots,R_{n-1}}^{r,s}\right)^{-1} \begin{pmatrix} \mathcal{V}_{R_0}^{r,s}(A_0) - r!\tilde{\Phi}_n^{r,s}(A_0) \\ \vdots \\ \mathcal{V}_{R_{n-1}}^{r,s}(A_0) - r!\tilde{\Phi}_n^{r,s}(A_0) \end{pmatrix}$$

for all $r, s \in \mathbb{N}_0$, where a new Vandermonde-type matrix

$$N_{R_0,\ldots,R_{n-1}}^{r,s} = r!s! \begin{pmatrix} \kappa_{s+1} R_0^{s+1} & \ldots & \kappa_{s+n} R_0^{s+n} \\ \vdots & \ddots & \vdots \\ \kappa_{s+1} R_{n-1}^{s+1} & \ldots & \kappa_{s+n} R_{n-1}^{s+n} \end{pmatrix}$$

is introduced. The number of equations in the linear system described by (3.16) is reduced by one compared to (3.8). Once more, we note that for $s \in \mathbb{N}$, $\Phi_n^{r,s}(A)$ equals 0 and can thus be estimated by $\tilde{\Phi}_n^{r,s}(A_0) = 0$, so the terms $r!\tilde{\Phi}_n^{r,s}(A_0)$ are set equal to zero.

It follows from Theorems 3.1 and 3.2 that $\tilde{\Phi}_k^{r,s}(A_0)$ is multigrid convergent for all $k \in \{0, \ldots, n\}$ and $r, s \in \mathbb{N}_0$ if A is compact and topologically regular and $0 < R_0 < \cdots < R_{n-1} < \mathrm{Reach}(A)$. A combination of [19, Theorem 4.2 and Lemma 5.1] and the remarks following Theorem 3.2 implies that this convergence is of order $O(d)$ when A is convex or ρ-regular and $r = s = 0$. For $r, s \neq 0$, the speed of convergence is $O(\sqrt{d})$.

Again, this algorithm has been implemented in the planar case. For $n = 2$, the algorithm in (3.16) reduces to

$$\begin{pmatrix} \tilde{\Phi}_1^{r,s}(A_0) \\ \tilde{\Phi}_0^{r,s}(A_0) \end{pmatrix} = \left(N_{R_0,R_1}^{r,s}\right)^{-1} \begin{pmatrix} \mathcal{V}_{R_0}^{r,s}(A_0) - r!\tilde{\Phi}_2^{r,s}(A_0) \\ \mathcal{V}_{R_1}^{r,s}(A_0) - r!\tilde{\Phi}_2^{r,s}(A_0) \end{pmatrix} \tag{3.17}$$

for $r, s \in \mathbb{N}_0$, where $\tilde{\Phi}_2^{r,0}(A_0) = (d^2/r!) \sum_{x \in A_0} x^r$ and $N_{R_0,R_1}^{r,s}$ is defined as

$$N_{R_0,R_1}^{r,s} = r!s! \begin{pmatrix} \kappa_{s+1} R_0^{s+1} & \kappa_{s+2} R_0^{s+2} \\ \kappa_{s+1} R_1^{s+1} & \kappa_{s+2} R_1^{s+2} \end{pmatrix}$$

with κ_j given by (3.12) for $j \in \{1, \ldots, 4\}$.

Algorithm Minktensor2D2R ($\mathcal{R}_2$)

Input:

- Digitisation $A_0 = A \cap dL$ of a compact topologically regular set $A \subset \mathbb{R}^2$ with positive reach;
- Lattice distance $d > 0$ of dL;
- Two radii R_0 and R_1 satisfying $d/\sqrt{2} < R_0 < R_1 < \mathrm{Reach}(A)$.

Action:

- Calculate $\tilde{\Phi}_2^{r,0}(A_0)$ using (3.13) for $r \in \{0, 1, 2\}$;
- Calculate $\mathcal{V}_{R_i}^{r,s}(A_0)$ using (3.10) for $i \in \{0, 1\}$ and for all $r, s \in \{0, 1, 2\}$ with $r + s \leq 2$;
- Determine the corresponding Minkowski tensors by (3.17).

Output:

- Estimators $\tilde{\Phi}_k^{r,s}(A_0)$ for all $k, r, s \in \{0, 1, 2\}$ with $r + s \leq 2$ (including the trivial ones of the form $\hat{\Phi}_k^{0,1}(A_0)$ for model control) of the corresponding Minkowski tensors.

The programme will give a warning if radii are chosen below the recommended lower limit $d/\sqrt{2}$ as discussed above. If any result of the algorithm is numerically below 10^{-6}, the output is rounded off to zero.

3.4 Simulations

In the preceding sections, we have introduced two different algorithms for the computation of estimators of the Minkowski tensors of sets $A \subset \mathbb{R}^2$ with positive reach from digitisations $A_0 = A \cap dL$. We denote by $\mathcal{R}_3$ the first algorithm, introduced in Section 3.2, which is based on three radii $0 < R_0 < R_1 < R_2 < \mathrm{Reach}(A)$, and by $\mathcal{R}_2$ the second algorithm, introduced in Section 3.3, based on two radii $0 < R_0 < R_1 < \mathrm{Reach}(A)$. Theorems 3.1 and 3.2 on multigrid convergence of the algorithms do not depend on the specific choice of the radii, but we expect the choice to affect the accuracy at least to some extent: We have already argued that choosing radii below $d/\sqrt{2}$ is problematic, so we will impose the restriction $d/\sqrt{2} < R_0$ henceforth. Finally, we wish to explore the consequences of choosing one or more radii larger than $\mathrm{Reach}(A)$.

For $\mathcal{R}_3$, three Voronoi tensor measures $\mathcal{V}_{R_i}(A_0)$, $i \in \{0, 1, 2\}$, must be calculated and a system of three linear equations must be solved. For $\mathcal{R}_2$, only two Voronoi tensor measures $\mathcal{V}_{R_i}(A_0)$, $i \in \{0, 1\}$, are calculated and a system of two linear equations solved. As a trade-off, we must calculate estimators of the volume tensors separately, but these are computed quite effectively using the algorithm in (3.13). In particular, note that for the volume tensors, we need only compute the sum $\sum_{x \in A_0} x^r$ once for each r, the relevant values being $r \in \{0, 1, 2\}$, resulting in three simple computations. For the Voronoi tensor measures, computed via (3.10), we need computations for each choice of $r \in \{0, 1, 2\}$,

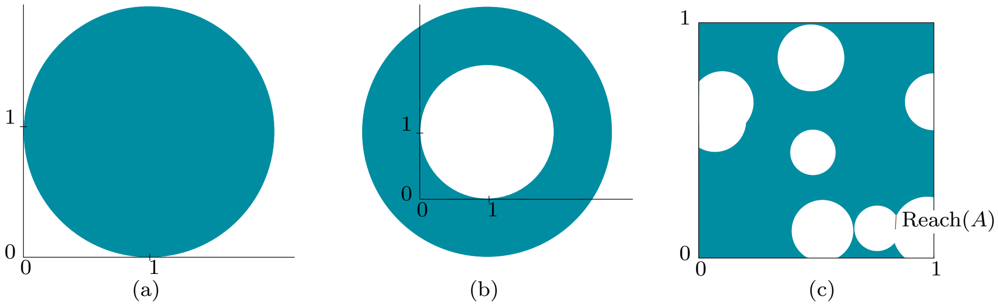


Figure 3.3 The three test sets used in our simulations. (a) shows the disc $B_1 + (1,1)^T$, (b) the annulus $\mathrm{cl}(B_2 \setminus B_1) + (1,1)^T$, and (c) the complement of one realisation of a Boolean model with intensity $\gamma = 10$ and uniform radius distribution $U(0.08, 0.16)$ for the typical particle.

for each choice of $s \in \{0,1,2\}$ such that $r + s \leq 2$, and for each choice of R_i. For $\mathcal{R}_3$ with three radii R_i, $i \in \{0,1,2\}$, this gives twelve computations; for $\mathcal{R}_2$ with two radii R_i, $i \in \{0,1\}$, only nine. Hence the second algorithm requires fewer computations, and moreover, the number of computations of the more complicated integral on the right-hand side of (3.10), which depends on the Voronoi diagram, is reduced by three. We would thus expect $\mathcal{R}_2$ to be faster than $\mathcal{R}_3$ as the size of the data set A_0 increases or, equivalently, as the resolution grows.

3.4.1 Choices of radii

The purpose of this section is to give recommendations for proper choices of radii based on simulations with test sets. Let $R_{\max}$ be the maximal radius equal to R_2 in the case of $\mathcal{R}_3$ and to R_1 in the case of $\mathcal{R}_2$. To evaluate the behaviour of the estimators for varying choices of radii, the following procedure is used.

Procedure $\mathcal{P}$ for selecting radii

(1) For some fixed resolution, consider the interval $(d/\sqrt{2}, m)$ where $3d/\sqrt{2} < m < \mathrm{Reach}(A)$. Set $R_{\max} = m$ and $R_0 = R_{\max}/n$ for algorithm $\mathcal{R}_n$, $n = 2,3$. This way, the radii are evenly spaced over the interval $(0, m)$ and all greater than $d/\sqrt{2}$. We now let m vary and plot the estimators of the Minkowski tensors as functions of m.

(2) Based on our findings in (1) and applying the same resolution, we choose a maximal radius in the interval $(3d/\sqrt{2}, \mathrm{Reach}(A))$ for which we get satisfactory estimators in the previous simulations. We then choose R_0 in the interval $(d/\sqrt{2}, R_{\max})$. We plot the estimators as functions of R_0.

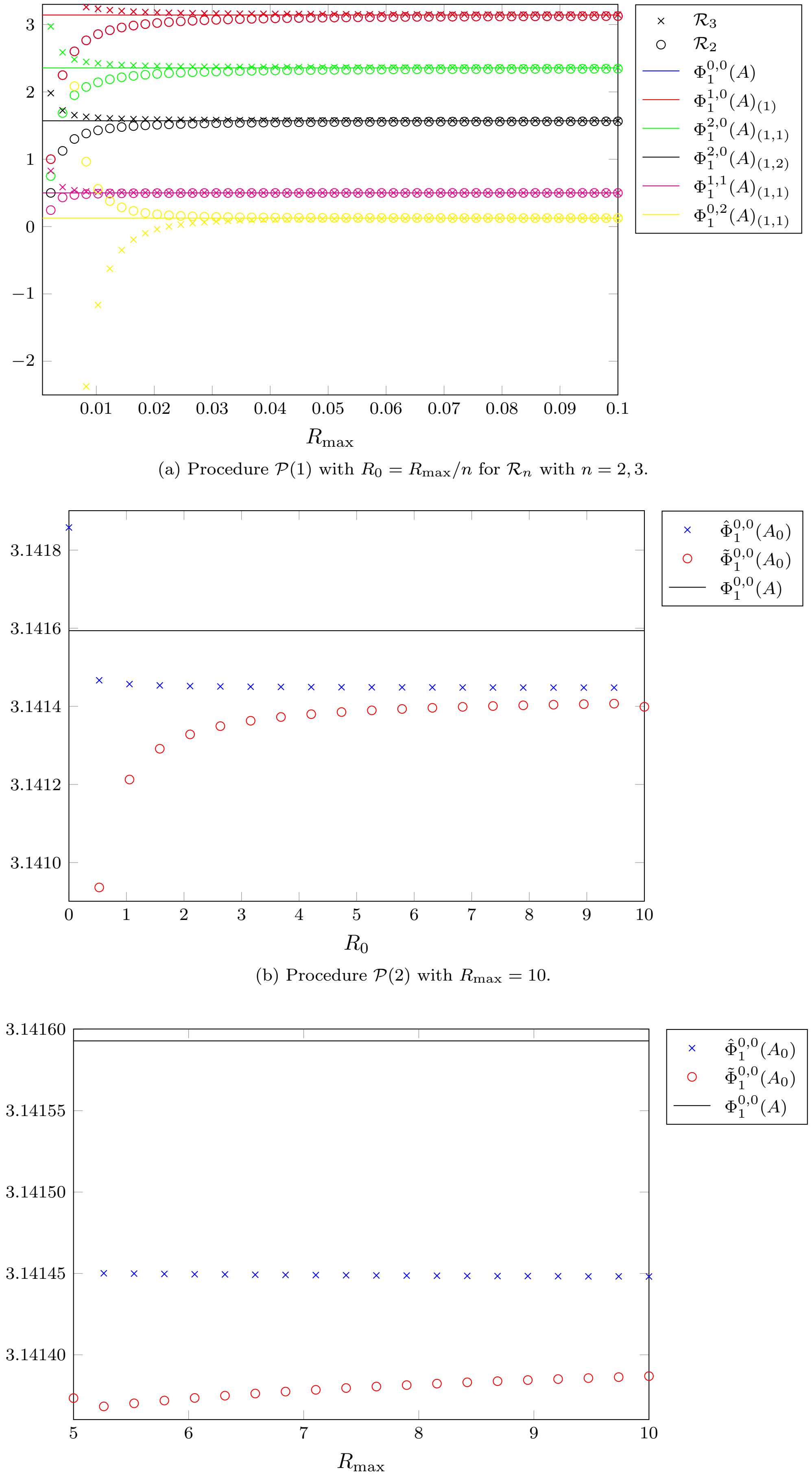
(3) Based on our findings in (2) and applying the same resolution, we choose a minimal radius $R_0 > d/\sqrt{2}$ for which we get satisfactory estimators in the previous simulations. We then choose $R_{\max}$ in the interval (R_0, m) where $R_0 < m < \mathrm{Reach}(A)$. We plot the estimators as functions of $R_{\max}$.

If A is convex, there is no upper constraint on m or $R_{\max}$ in the above procedure. The procedure requires some choice of resolution. For the simulations below, we choose resolution $d^{-1} = 1000$.

We now apply procedure $\mathcal{P}$ to different test sets in $\mathbb{R}^2$. As test sets we have chosen, with increasing geometric complexity, a convex disc, a non-convex annulus, and a realisation of the complement of a planar Boolean model; see Figure 3.3. In Appendix C, the Minkowski tensors of rank at most two of these test sets are calculated, and we provide documentation of the digitisation of the test sets.

Our starting point is the simple case of a translated unit disc $A = B_1 + (1,1)^T$, where we set $B_1 = B_1(0)$. Subsection C.1.1 provides detailed computations of the Minkowski tensors of A. We apply procedure $\mathcal{P}$ to A to find suitable radii for each of the algorithms; see Figure 3.4. In this figure, we only report results for the tensors $\Phi_k^{r,s}(A)$, $r, s \in \{0,1,2\}$, $r + s \leq 2$, in the case $k = 1$. The corresponding results for $k = 0$ are qualitatively the same. The case $k = 2$ stands out in that $\mathcal{R}_2$ makes use of (3.13) for the estimation of the volume tensors, and this formula does not depend on the radii. The results for the tensors with $k = 2$ are, however, very similar to those with $k \in \{0,1\}$ for the algorithm $\mathcal{R}_3$, but to illustrate the difference in behaviour of $\mathcal{R}_2$, Figure 3.5 shows the results of step (1) of procedure $\mathcal{P}$ for the volume tensor $\Phi_2^{0,0}(A)$.

Figure 3.4 The quality of estimators of the tensors $\Phi_1^{r,s}(A)$, $r, s \in \{0,1,2\}$, $r+s \leq 2$, for fixed resolution $d^{-1} = 1000$ and different radii for the disc $A = B_1 + (1,1)^T$. The estimators of $\Phi_1^{0,0}(A)$ and $\Phi_1^{1,0}(A)_{(1)}$ overlap completely, thus explaining why $\Phi_1^{0,0}(A)$ does not show in (a).

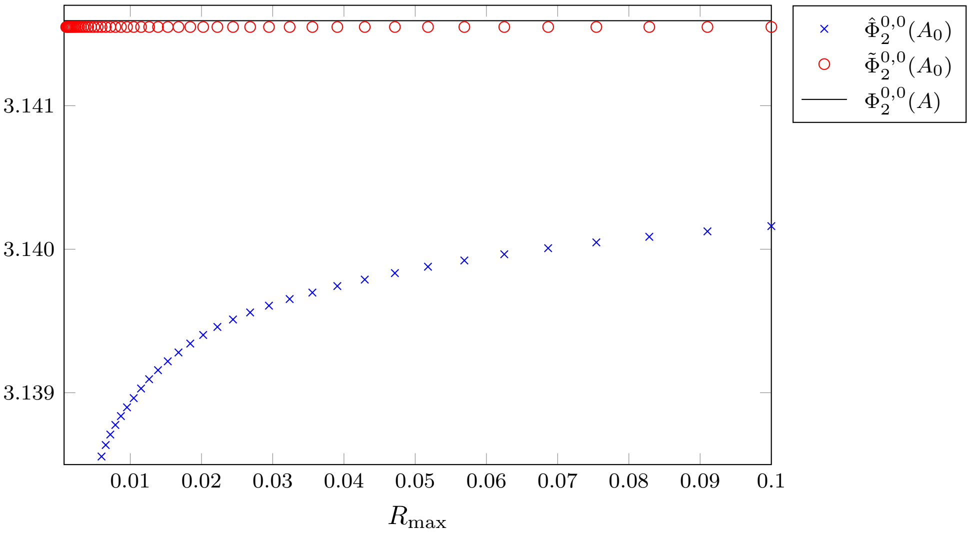


(a) Procedure $\mathcal{P}(1)$ with $R_0 = R_{\max}/n$ for $\mathcal{R}_n$ with $n = 2, 3$.

(b) Procedure $\mathcal{P}(2)$ with $R_{\max} = 10$.

(c) Procedure $\mathcal{P}(3)$ with $R_0 = 5$.

Figure 3.5
Procedure $\mathcal{P}(1)$ for the disc $A = B_1 + (1,1)^T$ with $d^{-1} = 1000$ and $R_0 = R_{\max}/n$ for $\mathcal{R}_n$ with $n = 2, 3$.



To report results, we use the identification of rank p tensors with the entries of the array of size n^p as explained in (3.1). As there are 12 non-trivial components of tensors $\Phi_1^{r,s}(A)$ with $r, s \in \{0,1,2\}$, $r+s \leq 2$, we only report a selection of such entries. To concretise, $\Phi_1^{0,0}(A) = \pi$ is half the perimeter of A, and $\Phi_1^{1,0}(A)$ is, up to normalisation, the centre of gravity of a uniform mass distribution on ∂A. Hence $\Phi_1^{1,0}(A)_{(1)}$ is the projection of this centre of gravity onto the x-axis. Another example is

$$\Phi_1^{2,0}(A) = \frac{1}{2}\int_{\partial A} x^2 \mathcal{H}^1(\,\mathrm{d}x),$$

where $\mathcal{H}^1$ denotes the one-dimensional Hausdorff measure. Hence

$$\Phi_1^{2,0}(A)_{(i,j)} = \frac{1}{2}\int_{\partial A} x_i x_j \mathcal{H}^1(\,\mathrm{d}x), \qquad i,j \in \{1,2\}.$$

For Figures 3.4(b) and (c), we have chosen to plot just a single non-trivial array entry. The reason for this is that the difference between algorithms $\mathcal{R}_3$ and $\mathcal{R}_2$ is only visible on a very small scale; much smaller than the difference in values of the tensors. Analogously, for the volume tensors $\Phi_2^{r,0}(A)$ with $r \in \{0,1,2\}$, we plot only the entry corresponding to the area $\Phi_2^{0,0}(A) = \pi$ in Figure 3.5. The reader should note the precision on the y-axis before concluding that $\mathcal{R}_3$ is significantly poorer than $\mathcal{R}_2$ for volume tensor estimation; from $R_{\max} > 0.0023$, the error of any volume tensor estimator computed by $\mathcal{R}_3$ in this step is below 1%.

The behaviour shown for these selected tensors is representative of the tensors of A. No estimators of tensors that are equal to zero are included in our plots because the estimators likewise equal zero independently of the choices of radii in any step of procedure $\mathcal{P}$ for both algorithms.

From Figure 3.4(a), representing step (1) of $\mathcal{P}$, we observe that unless the radii are all chosen below 0.02, the choice of radii does not seem to have much effect on the computed estimators. In the figure, we have zoomed in on the small interval $(d/\sqrt{2}, 0.1]$ to illustrate this point. When $R_{\max}$ is chosen larger than 0.4, the error of the estimators is below 1% for both algorithms. For $\mathcal{R}_3$ in particular, $R_{\max}$ need only be greater than 0.2 to obtain this accuracy. This means that even though we have no upper limit on $R_{\max}$, we need not choose an extremely large value but could select $R_{\max}$ equal to e.g. a few times the diameter of the object. In Figure 3.4(b), which corresponds to step (2) of procedure $\mathcal{P}$, we examine the algorithms for $R_{\max} = 10$. We observe that problems arise only when R_0 is chosen in the lower half of the interval $(0, 10)$. Indeed, if we choose $R_0 = 0.01$ or 0.1 in step (3), we get large errors for $\mathcal{R}_2$. In comparison, $\mathcal{R}_3$ is less affected by our choice of R_0 and is generally more accurate than $\mathcal{R}_2$. In Figure 3.4(c), we set $R_0 = 5$ and observe that varying $R_{\max}$ has close to no effect as long as we do not choose it too close to R_0.

There is an apparent tendency in Figures 3.4(b) and (c) of the estimators to converge to some wrong value of the tensor. This, however, is not in conflict with our theoretical expectations. The estimators are expected to converge to the true tensors as we increase the resolution, and in these simulations, we have fixed a resolution and are simply examining the importance of the choices of radii. The examination of the importance of the resolution follows in the next subsection.

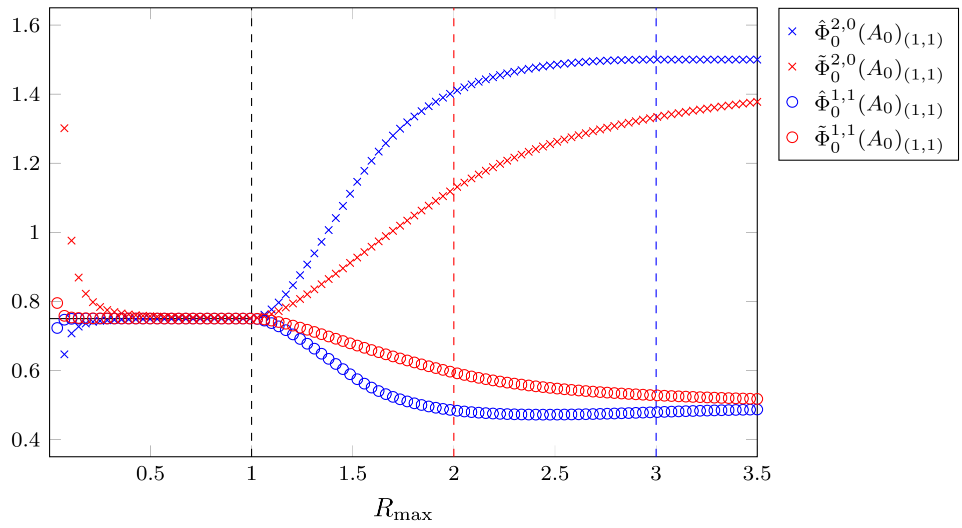


Figure 3.6 Procedure $\mathcal{P}(1)$ for the annulus $A = \mathrm{cl}(B_2 \setminus B_1) + (1,1)^T$ with $d^{-1} = 1000$ and $R_0 = R_{\max}/n$ for $\mathcal{R}_n$ with $n = 2, 3$. The dashed black line indicates the point when $R_{\max}$ is greater than $\mathrm{Reach}(A)$; the coloured dashed lines when all radii of the corresponding algorithm are above $\mathrm{Reach}(A)$.

From Figure 3.4(a), we see a tendency of the algorithms to approximate the tensors either from above or below. One could ask whether relations

$$\hat{\Phi}_k^{r,s}(A_0) \le \Phi_k^{r,s}(A) \le \tilde{\Phi}_k^{r,s}(A_0)$$

exist, possibly with the inequalities reversed depending on the choice of $k, r, s \in \{0, 1, 2\}$, $r + s \le 2$. However, we see from Figures (b) and (c) that this is not the case, as both algorithms underestimate the tensor.

Analogously, we now follow procedure $\mathcal{P}$ in the case of a non-convex set. We choose to consider the annulus $A = \mathrm{cl}(B_2 \setminus B_1) + (1,1)^T$, the Minkowski tensors of which are calculated in Subsection C.1.2. The reach of A is 1, so we must choose $d/\sqrt{2} < R_0 < R_1 < R_2 < 1$. However, we now repeat procedure $\mathcal{P}(1)$ for this non-convex set and allow for radii above the theoretical upper limit $\mathrm{Reach}(A)$ in order to investigate how this affects the estimators. The result is illustrated in Figure 3.6, where we have chosen to plot the non-zero tensor entries $\Phi_0^{2,0}(A)_{(1,1)}$ and $\Phi_0^{1,1}(A)_{(1,1)}$. Clearly, as soon as $R_{\max} > \mathrm{Reach}(A)$, the estimators become highly unreliable. There is no visible effect when the lower radii also surpass the upper bound.

For radii within the permitted range, $(d/\sqrt{2}, \mathrm{Reach}(A))$, we get a picture similar to the one for the disc, but since the radii are now limited to a smaller interval, we see the effect of choosing the radii too close to one another to a greater extent. The picture is similar for the remaining tensors: for $R_{\max}$ greater than 0.25, the error of $\mathcal{R}_3$ is below 1%, whereas we need $R_{\max} > 0.5$ for the same precision in $\mathcal{R}_2$. In general, results improve for higher choices of $R_{\max}$, and with $R_{\max} = 0.95$, we obtain an error below 0.3%. For step (2) of procedure $\mathcal{P}$, we thus pick $R_{\max} = 0.95$. The picture here resembles Figure 3.6, so we do not include the plot. Again, we need to choose R_0 near the middle of the interval $(d/\sqrt{2}, R_{\max})$ in order to get satisfactory results, suggesting a strategy for choosing R_0 when $R_{\max}$ is fixed. This becomes even more apparent when we simulate for fixed R_0 in step $\mathcal{P}(3)$. Indeed for R_0 equal to for instance 0.1, the estimators are rather poor, most prominently for $\mathcal{R}_2$, no matter how we choose $R_{\max}$. Choosing $R_0 = 0.5$ gives much more accurate estimators, in particular if $R_{\max}$ is chosen close to $\mathrm{Reach}(A)$.

It thus appears that a recommendation for non-convex sets is to choose $R_{\max}$ close to the upper limit $\mathrm{Reach}(A)$ and set $R_0 = R_{\max}/2$. These recommendations are at least valid for the non-convex annulus, and it is now natural to ask whether they apply also in the case of other, possibly more complex, non-convex sets.

One candidate for a more complex non-convex set is the complement of a (stationary) Boolean model. We will not go into details with the definition of Boolean models, since we will only consider one concrete example; the interested reader is instead referred to [33, Section 4.3]. Consider a stationary Poisson point process in $\mathbb{R}^2$ with intensity $\gamma > 0$. A random compact set is attached to each point of the process in such a way that the random sets are independent of each other and of the Poisson process. The union Z of these sets forms a stationary Boolean model. We consider here a Poisson process with intensity $\gamma = 10$ and discs as random compact sets with a radius distribution given by the uniform distribution $U(a, b)$ in the interval between $a = 0.08$ and $b = 0.16$.

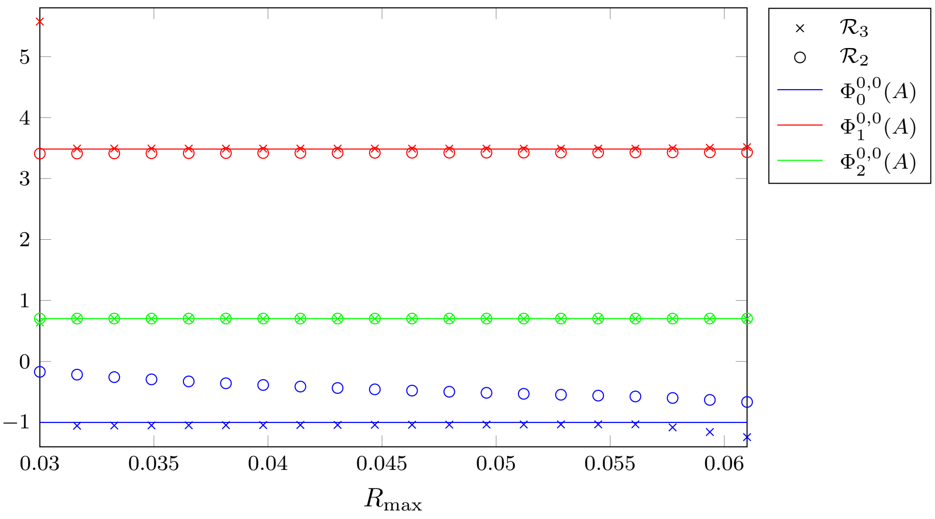


Figure 3.7
Procedure $\mathcal{P}(3)$ for the complement A of the Boolean model in Figure Figure 3.3(c) with $d^{-1} = 1000$ and $R_0 = 0.03$.

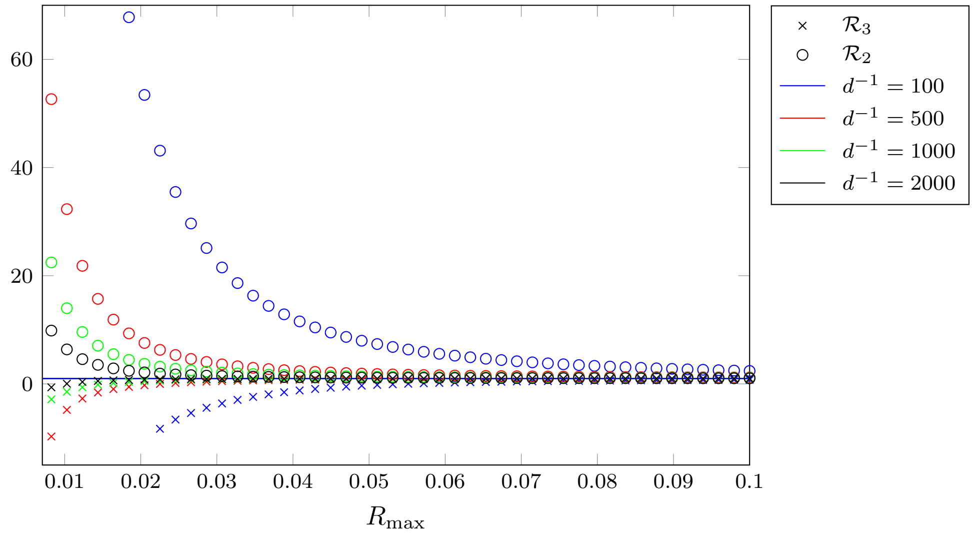
We will analyse Minkowski tensors of the Boolean model in the unit square $W = [0,1]^2$; see Figure 3.3(c) for the realisation that was used in the analysis. It is clear that $Z \cap W$ does not in general have positive reach (positive reach is only possible if none of the discs overlap). Therefore, we consider instead its complement $A = \mathrm{cl}(W \setminus Z)$ in W. This model is described further in Subsection C.1.3, and its intrinsic volumes are calculated. We have indicated the distance determining the reach of A with a red line in Figure 3.3(c). As this is measured manually, we determine a lower bound $0.0614 < \mathrm{Reach}(A)$ slightly smaller than the reach. We then perform procedure $\mathcal{P}$ but restrict attention to the tensors of rank zero: the Euler characteristic $\Phi_0^{0,0}(A)$, half the perimeter $\Phi_1^{0,0}(A)$, and the area $\Phi_2^{0,0}(A)$ as illustrated by Figure 3.7. Here, we show the result of simulations for step (3) of procedure $\mathcal{P}$ with $R_0 = 0.03$, which was found by steps (1) and (2) to be a good choice for the minimal radius. For this particular model, it seems we have a challenge in finding truly good radii for obtaining satisfactory estimators. Especially $\mathcal{R}_2$ is prone to error.

One explanation for the less precise estimators of the complement of the Boolean model might be that we need to consider the model for higher resolutions. It is then relevant to ask whether the same choices of radii are equally good for different choices of resolution. We will explore this question shortly. The procedure $\mathcal{P}$ can be applied to a test set for each of the algorithms $\mathcal{R}_3$ and $\mathcal{R}_2$ in turn in order to find good choices of radii for each of them, but one may also choose to find those radii which yield the best results for both algorithms simultaneously. The latter allows for what could be considered a better basis for comparison of the algorithms later on when we fix the radii and let the resolution vary. Incidentally, for the three test sets above, we have chosen the same radii for $\mathcal{R}_3$ and $\mathcal{R}_2$ in the different steps of procedure $\mathcal{P}$ because the best choices for $\mathcal{R}_2$ work equally well for $\mathcal{R}_3$ (although not necessarily the other way around).

We check now whether our choice of resolution affects the way in which we should choose the radii for a given test set. Consider again the disc $B_1 + (1,1)^T$. The simulations performed for this set, illustrated by Figure 3.4, make use of a resolution $d^{-1} = 1000$. We now carry out the same simulation but for different resolutions. The results are shown in Figure 3.8, where the effect is illustrated for the Euler characteristic, $\Phi_0^{0,0}(A)$. The graphs, each representing one of four chosen resolutions, exhibit a similar behaviour. As could be expected, higher resolution yields better estimators, but from the figure it would seem that the resolution only determines how large radii are needed for good estimators, i.e. for low resolutions, one needs to be able to choose rather high radii whereas high resolutions allow for choices of smaller radii. This tendency is reproducible for the two other test sets as well for all the estimators. This means that if we have a model for which the choice of radii is restricted by the reach, we can compensate by choosing a higher resolution. Moreover, if one choice of radii works well for a given resolution, that same choice will also work for higher resolutions.

As stated in Subsection 3.2.3, $\mathcal{R}_3$ sets $R_1 = (R_0 + R_{\max})/2$ by default. We have examined the effect of varying R_1 in the interval $(R_0, R_{\max})$, but except for when R_1 is very close to the limits of the interval, the choice of R_1 seems to have no effect at all. Therefore, minimising the number of input arguments for the algorithm by pre-assigning R_1 seems the best solution.

Figure 3.8
Comparison of effect of choices of radii on estimators of the Euler characteristic, $\Phi_0^{0,0}(A)$, for different choices of resolution in procedure $\mathcal{P}(1)$ for $A = B_1 + (1,1)^T$.



Concluding this section, we can give the following general recommendations for the choice of radii. It is advantageous to choose $R_{\max}$ large; we suggest 95% of Reach(A) in the non-convex case and several diameters of A in the convex case. The smallest radius, R_0, is recommended to be chosen in the middle of the interval $(d/\sqrt{2}, R_{\max})$, which turned out to work well for $\mathcal{R}_2$. Concerning $\mathcal{R}_3$, the choice of R_0 seems less critical as long as it is not chosen too close to either end of the interval $(d/\sqrt{2}, R_{\max})$. For $\mathcal{R}_3$, setting $R_1 = (R_0 + R_{\max})/2$ is recommended, and this choice has already been made in our implementation of the algorithm.

3.4.2 Influence of resolution on the quality of estimators

Having chosen radii for the three test sets based on procedure $\mathcal{P}$ for a given resolution, we examine the influence of the resolution on the computed estimators; a task which was partly commenced by the comparison in Figure 3.8. We simulate for varying choices of resolution and plot the estimators as functions hereof.

Figure 3.9 shows results for the three test sets of the previous subsection. As we have fixed the radii, we have automatically set a lower limit for the resolution, since we need to make sure that $d/\sqrt{2} < R_0$. As the three sets are plotted together, the set with the smallest R_0, the complement of the Boolean model, determines the minimal resolution. Thus for $R_0 = 0.03$, the resolution should be at least 24.

The functions plotted in Figure 3.9 are not, as previously, the estimators. Rather, we plot the absolute deviation of the estimator from the true tensor; the **absolute residual**. The residual $\hat{\Phi}_k^{r,s}(A_0) - \Phi_k^{r,s}(A)$ of the estimator $\hat{\Phi}_k^{r,s}(A_0)$ is denoted by $\mathrm{Res}(\hat{\Phi}_k^{r,s}(A_0))$ for $k, r, s \in \{0,1,2\}$ with $r + s \leq 2$. Figure 3.9(a) shows the result for the Euler characteristic, $\Phi_0^{0,0}(A)$, and Figure 3.9(b) that for the area, $\Phi_2^{0,0}(A)$. The functions are plotted on a log-log scale. This way, we are able to check the rate of convergence. Indeed, summing up on previous remarks in Sections 3.2 and 3.3, for estimators of the intrinsic volumes, $\Phi_k^{0,0}(A)$ with $k \in \{0,1,2\}$, calculated by either algorithm, the rate of convergence is of order $O(d)$ if A is convex or ρ-regular. In fact, for the volume tensors, $\mathcal{R}_2$ yields estimators that converge to $\Phi_2^{r,0}(A)$, $r \in \{0,1,2\}$, with speed $O(d)$ if A is a compact topologically regular set with boundary a 1-rectifiable Lebesgue null set, which is less restrictive than the requirements of convexity or ρ-regularity.

That the speed of convergence of the estimator $\hat{\Phi}_k^{r,s}(A_0)$ towards $\Phi_k^{r,s}(A)$ is $O(d^m)$ for $m \in \mathbb{R}$ means that

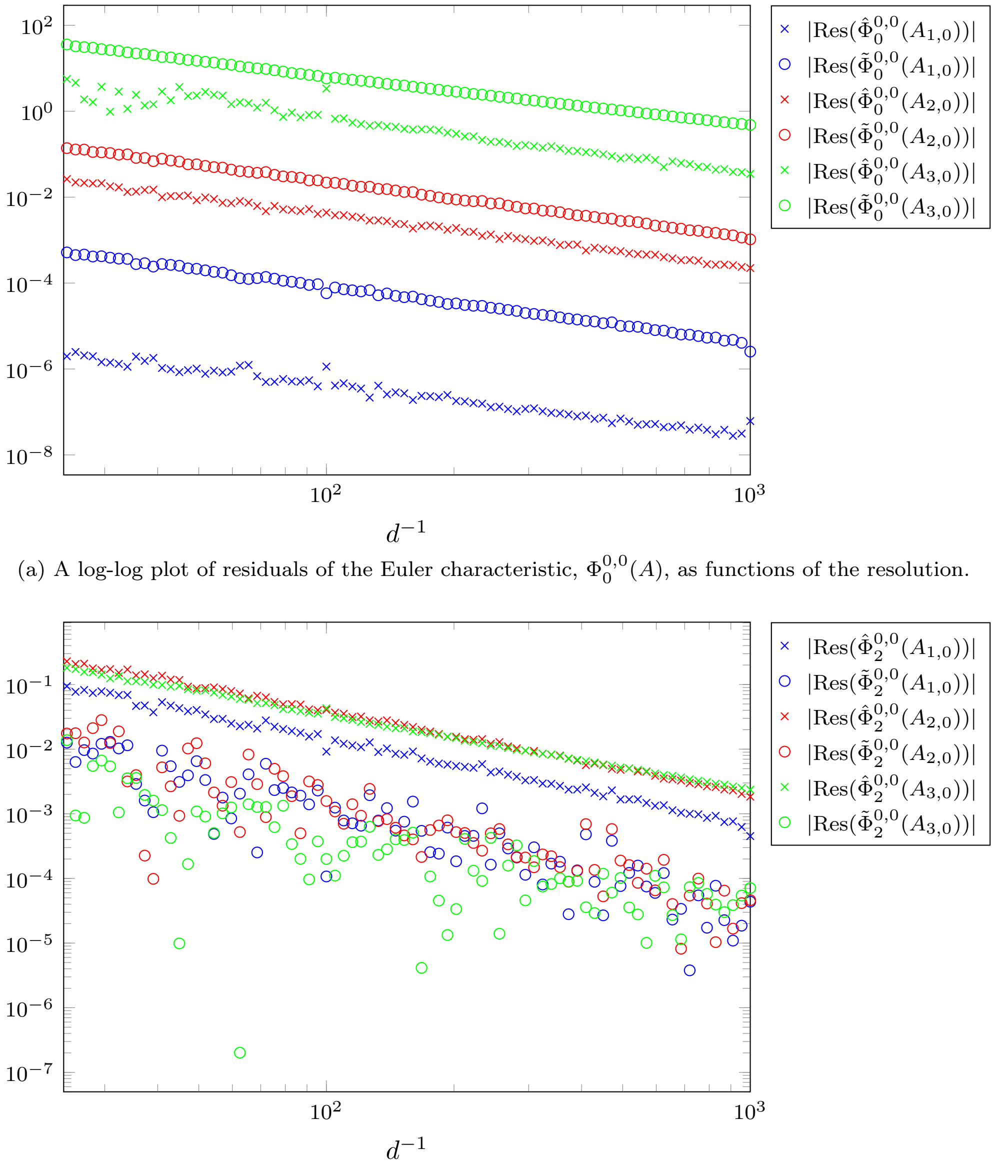
$$\mathrm{Res}(\hat{\Phi}_k^{r,s}(A_0)) \leq c \cdot d^m$$

for some $c \in \mathbb{R}$, where $\hat{\Phi}_k^{r,s}(A_0)$ is inherently a function of d. This implies that

$$\log|\mathrm{Res}(\hat{\Phi}_k^{r,s}(A_0))| \leq \log(c) + m\log(d). \tag{3.18}$$

The disc is both convex and ρ-regular for $\rho = 1$, and the annulus is ρ-regular for $\rho = 1$. The complement of the Boolean model is neither, but its boundary is 1-rectifiable. Hence, by (3.18), we expect to see linear graphs in Figures 3.9(a) and (b) with slope m equal to -1 (we are plotting as functions of d^{-1} and not d) for the disc and the annulus.

Figure 3.9
Plots of the absolute Minkowski tensor residuals as functions of the resolution for the three test sets $A_1 = B_1 + (1,1)^T$ with $R_0 = 5$, $R_{\max} = 10$; $A_2 = \mathrm{cl}(B_2 \backslash B_1) + (1,1)^T$ with $R_0 = 0.5$, $R_{\max} = 0.95$, and A_3 the complement of the Boolean model with $R_0 = 0.03$, $R_{\max} = 0.05$. In the legends, $A_{i,0}$ denotes the digitisation $A_i \cap dL$ for $i \in \{1,2,3\}$.



(a) A log-log plot of residuals of the Euler characteristic, $\Phi_0^{0,0}(A)$, as functions of the resolution.

(b) A log-log plot of residuals for the area, $\Phi_2^{0,0}(A)$, as functions of the resolution.

For the complement of the Boolean model we expect the same in Figure 3.9(b), but we have no theoretical results on its convergence rate for the Euler characteristic.

In Figure 3.9(a), linear fitting approximations yield:

$$\begin{aligned}
|\mathrm{Res}(\hat\Phi_0^{0,0}(A_{1,0}))| &\approx 1.19 \cdot \log(d) - 9.15;\\
|\mathrm{Res}(\tilde\Phi_0^{0,0}(A_{1,0}))| &\approx 1.28 \cdot \log(d) - 3.52;\\
|\mathrm{Res}(\hat\Phi_0^{0,0}(A_{2,0}))| &\approx 1.27 \cdot \log(d) + 0.36;\\
|\mathrm{Res}(\tilde\Phi_0^{0,0}(A_{2,0}))| &\approx 1.30 \cdot \log(d) + 2.18;\\
|\mathrm{Res}(\hat\Phi_0^{0,0}(A_{3,0}))| &\approx 1.69 \cdot \log(d) + 7.86;\\
|\mathrm{Res}(\tilde\Phi_0^{0,0}(A_{3,0}))| &\approx 1.30 \cdot \log(d) + 8.01,
\end{aligned}$$

where $A_{i,0}$ denotes the digitisation $A_i \cap dL$. These simulations lead to the following conclusions: Firstly, for the given example sets, the asymptotic speed of convergence already holds for finite resolution for a range of d that is realistic in practical applications. Even for the Boolean model, for which we do not have an asymptotic speed of convergence bound from theory, the speed is at least linear. Secondly, and maybe even more interestingly, the speed of convergence in the examples is clearly better than d^m with $m = 1$ as suggested by the theory. This indicates that an improved asymptotic convergence rate holds and could possibly be proven, at least for certain subclasses of sets with positive reach.

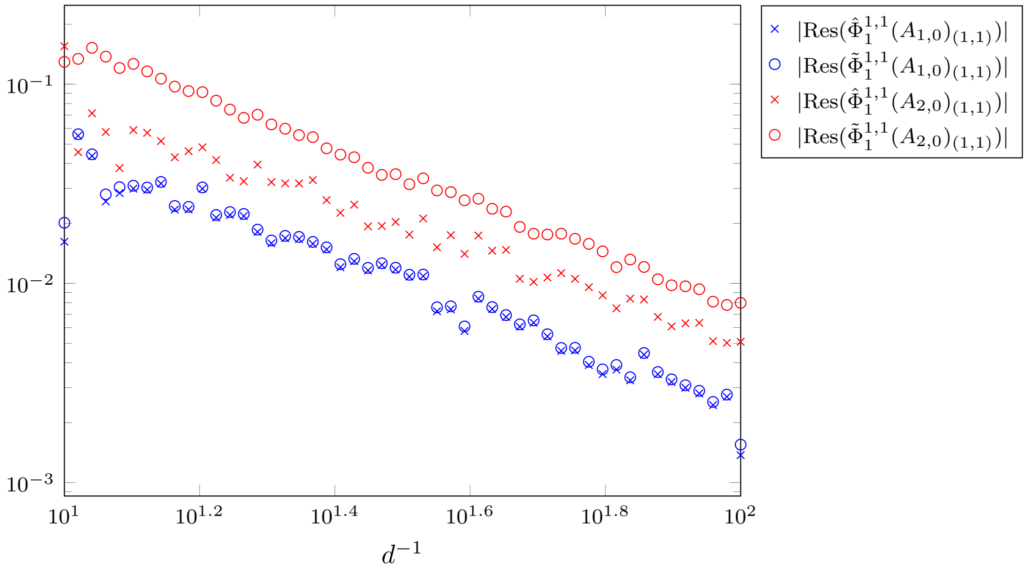


Figure 3.10
A log-log plot of the absolute Minkowski tensor residuals for a rank two tensor as a function of the resolution for $A_1 = B_1 + (1,1)^T$ with $R_0 = 5$, $R_{\max} = 10$ and $A_2 = \mathrm{cl}(B_2 \setminus B_1) + (1,1)^T$ with $R_0 = 0.5$, $R_{\max} = 0.95$.

Similar conclusions can be made for Figure 3.9(b), where the linear equations are given by:

$$
\begin{aligned}
|\mathrm{Res}(\hat{\Phi}_2^{0,0}(A_{1,0}))| &\approx 1.30 \cdot \log(d) + 1.72;\\
|\mathrm{Res}(\tilde{\Phi}_2^{0,0}(A_{1,0}))| &\approx 1.56 \cdot \log(d) + 0.30;\\
|\mathrm{Res}(\hat{\Phi}_2^{0,0}(A_{2,0}))| &\approx 1.28 \cdot \log(d) + 2.66;\\
|\mathrm{Res}(\tilde{\Phi}_2^{0,0}(A_{2,0}))| &\approx 1.41 \cdot \log(d) - 0.31;\\
|\mathrm{Res}(\hat{\Phi}_2^{0,0}(A_{3,0}))| &\approx 1.42 \cdot \log(d) + 3.41;\\
|\mathrm{Res}(\tilde{\Phi}_2^{0,0}(A_{3,0}))| &\approx 1.24 \cdot \log(d) - 2.27.
\end{aligned}
$$

Again, the speed of convergence is super-linear in the considered range of d. It appears that the speed of convergence of $\tilde{\Phi}_2^{0,0}(A_0)$ is better than the one of $\hat{\Phi}_2^{0,0}(A_0)$ except for the case $A = A_3$, but the slope in this case is supposedly less meaningful due to the huge fluctuation of the residuals (green circles). In fact, the coefficient of determination equals 0.59 compared to 0.87 for $A = A_1$ and 0.78 for $A = A_2$, indicating that the regression line is a rather poor approximation in the case $A = A_3$.

That $\tilde{\Phi}_2^{0,0}(A_0)$ performs particularly well for A_1 and A_2 is explained by the fact that this estimator is based on the basic counting scheme (3.13). For the circle, $A = A_1$, the speed of convergence problem corresponds essentially to the Gauss circle problem, where the best known exponent [20] is $m = 285/208 \approx 1.37$, but it is conjectured [14] to be $m = 1.5 - \varepsilon$ for arbitrary $\varepsilon > 0$.

The graphs for $\mathcal{R}_2$ exhibit strong fluctuations of the residuals. However, as the estimators for the volume tensors, $\tilde{\Phi}_2^{r,0}(A_0)$ with $r \in \{0,1,2\}$, are essentially based on counting sampling points in A as explained above, a so-called Zitterbewegung effect [1, p. 307], known from systematic sampling, occurs here, thus explaining the fluctuations.

The rate of convergence for $\hat{\Phi}_k^{r,s}(A_0)$ and $\tilde{\Phi}_k^{r,s}(A_0)$ with $k, r, s \in \{0,1,2\}$ and $r + s \in \{1,2\}$ for convex and ρ-regular sets is $O(\sqrt{d})$. This asymptotic result is also tested in practice for the disc and the annulus in Figure 3.10. Here, the absolute residual for the tensor $\Phi_1^{1,1}(A)_{(1,1)}$ is plotted. As before, we obtain linear equations:

$$
\begin{aligned}
|\mathrm{Res}(\hat{\Phi}_1^{1,1}(A_{1,0})_{(1,1)})| &\approx 1.30 \cdot \log(d) - 0.13;\\
|\mathrm{Res}(\tilde{\Phi}_1^{1,1}(A_{1,0})_{(1,1)})| &\approx 1.30 \cdot \log(d) - 0.10;\\
|\mathrm{Res}(\hat{\Phi}_1^{1,1}(A_{2,0})_{(1,1)})| &\approx 1.23 \cdot \log(d) + 0.32;\\
|\mathrm{Res}(\tilde{\Phi}_1^{1,1}(A_{2,0})_{(1,1)})| &\approx 1.31 \cdot \log(d) + 1.17,
\end{aligned}
$$

which do not corroborate the theoretically expected rate of convergence corresponding to a slope $m = 0.5$ but rather point to an even higher speed of convergence comparable to that for the intrinsic volumes.

From Figures 3.9 and 3.10, it is evident that the more complicated the test set, the higher resolution is required in order for the algorithms to produce good estimators. For the disc, $\mathcal{R}_3$ has en error below 1% for resolutions above $d^{-1} = 46$ compared to $d^{-1} = 80$

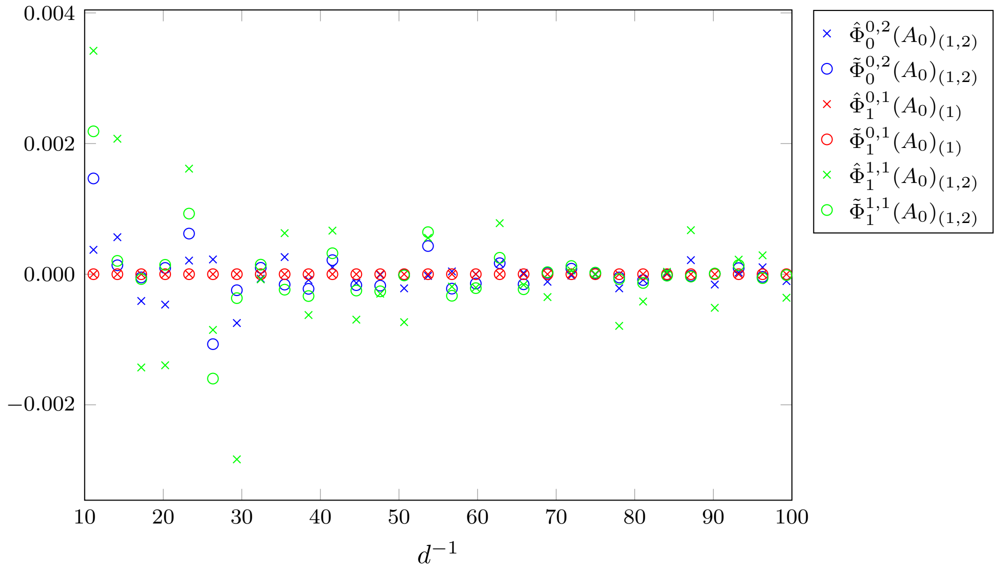


Figure 3.11 Plot of the Minkowski tensor estimators of tensors equal to zero as functions of the resolution for $A = \text{cl}(B_2 \setminus B_1) + (1,1)^T$ with $R_0 = 0.5$, $R_{\max} = 0.95$.

for the annulus. For the complement of the Boolean model, we need resolutions well above 2000 to obtain such accurate estimators. In comparison, $\mathcal{R}_2$ has an error below 1% for resolutions above $d^{-1} = 45$ for the disc but $d^{-1} = 285$ for the annulus. We cannot obtain this accuracy for the Boolean model with resolutions below 10 000, but above 9000, we can get below 5%.

The majority of the previous plots have neglected to report results for tensors that are (trivially) equal to zero with the exception of the Euler characteristic of the annulus in Figure 3.9(a). For completeness, we plot such estimators in Figure 3.11, again for the annulus. We see that not all estimators are identically zero, and here, too, results depend on the resolution. All zero-tensor estimators are reported by the algorithms $\mathcal{R}_3$ and $\mathcal{R}_2$, so these can be used as indicators of the overall quality of the estimators for a given resolution. The zero-tensors for the disc are in fact identically zero regardless of resolution, confirming that more complicated sets demand higher resolutions.

From the simulations, we make the following observations. Firstly, it has proven particularly difficult to compute good estimators of the Euler characteristic, $\Phi_0^{0,0}(A)$, for both algorithms $\mathcal{R}_3$ and $\mathcal{R}_2$, unless we consider very simple models such as the disc. Secondly, $\mathcal{R}_2$ produces better estimators of the volume tensors than $\mathcal{R}_3$. Lastly, smaller resolutions suffice for simpler models. This means that for the simplest model, the disc, all tensors are computed fairly accurately by both algorithms already for low resolutions, and residuals of both algorithms are low. In this case, the error of $\mathcal{R}_2$ is smaller than that of $\mathcal{R}_3$ because the volume tensors are the ones more difficult to estimate. However, as complexity of the model increases, difficulty of estimating the Euler characteristic surpasses the volume tensors, and so $\mathcal{R}_3$ becomes the better choice of algorithm since it estimates all but the volume tensors better than $\mathcal{R}_2$ does.

Regarding the speeds on convergence, the theoretical results for both algorithms in fact appear to hold under a resolution regime that is realistic in practice. In fact, the speeds of convergence deduced from our simulations are higher than the theoretically expected and also apply to sets that do not meet the necessary requirements for the theorems on multigrid convergence to apply.

Finally, running our simulations in MATLAB, we have observed that $\mathcal{R}_2$ is noticeably faster than $\mathcal{R}_3$. This is likely due to the considerations made at the beginning of Section 3.4.

3.5 Discussion

Based on our findings from the various plots in Section 3.4, we conclude that the algorithms $\mathcal{R}_3$ and $\mathcal{R}_2$ do in fact yield good estimators of the Minkowski tensors in the setting of finite resolution. Algorithm $\mathcal{R}_3$ has proven more accurate than $\mathcal{R}_2$ except in the case of the volume tensors.

On the other hand, $\mathcal{R}_2$ is generally faster than $\mathcal{R}_3$ in computing the estimators, and for sufficiently high resolutions, $\mathcal{R}_2$ does produce satisfactory estimators. Computation times are, however, fairly small in general. Thus, to obtain the best results, our recommendation is to use (3.13) to obtain reliable estimators of the volume tensors and subsequently use $\mathcal{R}_3$ to estimate the remaining Minkowski tensors. Hence we suggest not to use $\mathcal{R}_2$ at all apart from its volume tensor estimators in (3.13). First applying (3.13) and thereupon $\mathcal{R}_3$ increases computation times but yields far better estimators. The algorithm defined in this way is named Minktensor2D:

Algorithm Minktensor2D

Input:

- Digitisation $A_0 = A \cap dL$ of a compact topologically regular set $A \subset \mathbb{R}^2$ with positive reach;
- Lattice distance $d > 0$ of dL;
- Two radii R_0 and R_2 satisfying $d/\sqrt{2} < R_0 < R_2 < \text{Reach}(A)$.

Action:

- Calculate $\hat{\Phi}_2^{r,0}(A_0)$ using (3.13) for $r \in \{0, 1, 2\}$;
- Calculate $\mathcal{V}_{R_i}^{r;s}(A_0)$ using (3.10) for $i = 0, 1, 2$, with $R_1 = (R_0 + R_2)/2$, and for all $r, s \in \{0, 1, 2\}$ with $r + s \leq 2$;
- Determine the corresponding Minkowski tensors by (3.11).

Output:

- Estimators $\hat{\Phi}_k^{r,s}(A_0)$ for all $k, r, s \in \{0, 1, 2\}$ with $r + s \leq 2$ and $s = 0$ whenever $k = 2$ (including the trivial ones of the form $\hat{\Phi}_k^{0,1}(A_0)$ for model control) of the corresponding Minkowski tensors.

If any result of the algorithm is numerically below 10^{-6}, the output is rounded off to zero.

All MATLAB-files for Minktensor2D are available for download [6]. We have documented the code for Minktensor2D in Appendix D, where a printout of the .m-files of the programme is also provided.

With regard to the practical use of Minktensor2D, we suggest that in the setting of finite positive reach, one chooses $R_{\max}$ close to but strictly smaller than the reach; for instance $R_{\max}$ equal to 95% of Reach(A). The lower radius R_0 should be chosen near $R_{\max}/2$. When there is no upper bound on $R_{\max}$, i.e. the reach is infinite, $R_{\max}$ can be chosen arbitrarily large, but a choice equal to several diameters of the object is advised. Accuracy of the algorithms is then determined by the resolution. The more complicated the object, the larger resolution is required.

A disadvantage of the algorithm is that one needs to know (a lower bound for) the reach of the object. This is a hindrance if one wishes to extract geometric knowledge of some unknown set. Simply guessing the reach can cause major errors, since it is vital that the radii chosen for the algorithm are strictly smaller than the reach, as illustrated in Figure 3.6. Further, in applications, data is not always given as black-and-white digitisations but rather as greyscale images. Local algorithms for Minkowski tensor estimation based on greyscale input can for instance be found in [39].

We conclude by mentioning that an implementation of the above algorithms in dimension three or higher is possible. However, it appears to be difficult to determine the integrals on the right-hand side of (3.10) numerically. One possibility would be to use a sufficiently fine approximation of the R-bounded Voronoi cells by polytopes for which an exact evaluation of the corresponding integral is possible. This has already been used in [26] to solve a related problem.

A

SUPPORTING RESULTS OF CHAPTER TWO

This appendix is a copy of the notes [11] by du Plessis which have not yet been published but contain the main results on which our work in Chapter 2 is based. We therefore choose to include them here for completeness. The notation has been adapted to fit this thesis, and small comments to link the notes to the remaining parts of the thesis have been made. Otherwise, the contribution of the author is limited to proof reading, minor elaborations, and the addition of figures.

In the first section, a proof of the equivalence of the statements in Definition 1.4 as well as additional properties related to the boundary of r-regular sets is presented. The second section concerns the geometry of r-regular sets, and in the third section, Theorem A.21, with which we will be able to prove the remaining parts of Theorem 2.10, is proved. The final section is the author's elaboration on how Theorem A.21 can be used to prove Theorem 2.10.

A.1 DEFINITIONS AND BASIC RESULTS

In Definition 1.4, we present two definitions of r-regularity, which we assert are equivalent. This section commences with the proof of this claim. Subsequently, we will state, and in fact revise, some results from [12] concerning smoothness properties of the boundaries of r-regular sets, which are used in Chapter 2 for instance to define the set $T_{[-1,1]}$ diffeomorphic to $\partial A \times [-1, 1]$.

Proposition A.1

Proposition. *Let $A \subset \mathbb{R}^n$ be a closed set, and let $r \in (0, \infty)$. Then the following are equivalent:*

(i) *At any point $x \in \partial A$, there exist two closed balls $B_r \subset A$ and $B'_r \subset \mathrm{cl}(A^C)$ with radius r such that $B_r \cap B'_r = \{x\}$;*

(ii) *The sets A and $\mathrm{cl}(A^C)$ are equal to unions of closed balls with radius r.*

Proof. (i) ⇒ (ii): Consider a point $a \in A$. If $\delta_{\partial A}(a) \geq r$, a is contained in the closed ball $B_r(a) \subset A$. Otherwise, denote by s the distance $\delta_{\partial A}(a) < r$ from a to ∂A. Since ∂A is closed, there exists $x \in \partial A$ with $|a - x| = s$. Then $x \in B_s(a) \subset A$. Let $B_r \subset A$ and $B'_r \subset \mathrm{cl}(A^C)$ be closed balls with $B_r \cap B'_r = \{x\}$. Now $x \in B_s(a) \cap B_r$. If $B_s(a)$ and B'_r meet in more than one point, $\mathrm{int}(B_s(a)) \cap \mathrm{int}(B'_r)$ is non-empty. But

$$\mathrm{int}(B_s(a)) \cap \mathrm{int}(B'_r) \subset B_s(a) \cap B'_r \subset A \cap \mathrm{cl}(A^C) = \partial A,$$

and thus ∂A contains a non-empty open set. This is a contradiction since A is closed, so $B_s(a) \cap B'_r = \{x\}$. It follows that $B_s(a)$, B_r, and B'_r share a common tangent line at x, and that $B_s(a) \subset B_r$. In particular, a is contained in a closed ball of radius r contained in A. Thus A is equal to a union of closed balls with radius r. Interchanging A and $\mathrm{cl}(A^C)$ in the above leads to the same conclusion for $\mathrm{cl}(A^C)$.

(ii) ⇒ (i): Let $x \in \partial A$. Since A is closed, $\partial A = A \cap \mathrm{cl}(A^C)$, so there exist closed balls B_r and B'_r such that $x \in B_r \cap B'_r$. As argued in the first half of the proof, this implies that $B_r \cap B'_r = \{x\}$. □

In the following, we shall examine more closely the boundaries of r-regular sets.

Convention

Convention. For the remainder of this appendix, we let $A \subset \mathbb{R}^n$ be an r-regular set for $r \in (0, \infty)$. ■

Proposition A.2

Proposition. *Let $x \in \partial A$ and $s \in (0, r]$. Then there exist unique balls $B_s \subset A$ with $x \in B_s$ and $B'_s \subset \mathrm{cl}(A^C)$ with $x \in B'_s$.*

Proof. Let $x \in \partial A$, and let $B_r \subset A$ and $B'_r \subset \mathrm{cl}(A^C)$ be closed balls with radius r such that $B_r \cap B'_r = \{x\}$. Such balls exist by r-regularity of A. For $s \in (0, r]$, let $B_s \subset A$ be a closed s-ball containing x. By the same argument as in the first half of the proof of Proposition A.1, $B_s \cap B'_r = \{x\}$. It follows that B_r, B'_r, and B_s share a common tangent line at x and $B_s \subset B_r$. If $s = r$, B_s is equal to B_r, so B_s is the unique closed s-ball in A containing x. If $s < r$, B_s is the unique closed s-ball contained in B_r which is tangent to B_r at x.

Conversely, let $B'_s \subset \mathrm{cl}(A^C)$ be a closed s-ball containing x. Arguing as above, if $s = r$, B'_s is equal to B'_r, so B'_s is the unique closed s-ball in $\mathrm{cl}(A^C)$ containing x. If $s < r$, B'_s is the unique closed s-ball contained in B'_r which is tangent to B'_r at x. □

For x and s as in Proposition A.2, we will denote by $\mathcal{B}^i_s(x) \subset A$ the **unique inner ball of A** and by $\mathcal{B}^o_s(x) \subset \mathrm{cl}(A^C)$ the **unique outer ball of A**. Proposition A.2 then implies that the balls $B_r \subset A$ and $B'_r \subset \mathrm{cl}(A^C)$ with $B_r \cap B'_r = \{x\}$ of Definition 1.4(i) are unique, i.e. they are equal to $\mathcal{B}^i_r(x)$ and $\mathcal{B}^o_r(x)$ respectively.

For any closed set $X \subset \mathbb{R}^n$ and $s \geq 0$, let

(A.1)
$$\begin{aligned} U_s(X) &= \{x \in \mathbb{R}^n \mid \delta_{\partial X}(x) < s\}; \\ U^i_s(X) &= U_s(X) \cap X = \{x \in X \mid \delta_{\partial X}(x) < s\}; \\ U^o_s(X) &= U_s(X) \cap \mathrm{cl}(X^C) = \{x \in \mathrm{cl}(X^C) \mid \delta_{\partial X}(x) < s\}. \end{aligned}$$

Then $U_s(X) = U^i_s(X) \cup U^o_s(X)$, and $U^i_s(X) \cap U^o_s(X) = \partial X$. We shall use the convention that $U_s = U_s(A)$, $U^i_s = U^i_s(A)$, and $U^o_s = U^o_s(A)$. We now have the following corollary of Proposition A.2, where we recall the definition of $\mathrm{Unp}(X)$ from (1.3).

Corollary A.3 **Corollary.** *The set U_r is a subset of* $\mathrm{Unp}(\partial A)$.

Proof. We first consider $a \in U^i_r$. Let $s = \delta_{\partial A}(a) < r$. Since ∂A is closed, there exists $x \in \partial A$ such that $|x - a| = s$. The ball $B_s(a)$ is contained in A, and it contains x. Thus $B_s(a) = \mathcal{B}^i_s(x)$ by Proposition A.2. Since $s < r$, $\mathcal{B}^i_s(x) \backslash \{x\} \subset \mathrm{int}(\mathcal{B}^i_r(x)) \subset A$. This implies that $\mathcal{B}^i_s(x) \cap \partial A = \{x\}$, i.e. x is the unique point in ∂A satisfying $|x - a| = s = \delta_{\partial A}(a)$. In particular, $a \in \mathrm{Unp}(\partial A)$.

Now let $a \in U^o_r$ and $s = \delta_{\partial A}(a) < r$. Let $x \in \partial A$ be such that $|x - a| = s$. As above, we see that $B_s(a) = \mathcal{B}^o_s(x)$, and x is the unique point in ∂A satisfying $|x - a| = s = \delta_{\partial A}(a)$, so $a \in \mathrm{Unp}(\partial A)$.

Since $U_r = U^i_r \cup U^o_r$, we conclude that $U_r \subset \mathrm{Unp}(\partial A)$. □

For $a \in U_r$, the metric projection $p_{\partial A}(a)$ of a is thus well-defined. We now wish to use a series of results from [12] in order to prove certain properties of the boundary of r-regular sets. In this paper, Duarte and Torres define an *open* subset $X \subset \mathbb{R}^n$ to be r-regular if X and $\mathrm{int}(X^C)$ are both connected unions of open r-balls. The results we need rely on [12, Proposition 5], which essentially states that for such X, Proposition A.1(i) holds for $\mathrm{cl}(X)$. However, it is implicitly assumed that $\partial X = \partial(\mathrm{int}(X^C))$, but this is not necessarily true. Indeed, let B_r be a closed r-ball and L be either a point or a closed bounded line segment such that $\delta_L(B_r) \geq 2r$. Then $X = (B_r \cup L)^C$ is a connected union of open r-balls, and $\mathrm{int}(X^C)$ is itself an open r-ball, but $\partial X = \partial B_r \cup L$ whilst $\partial(\mathrm{int}(X^C)) = \partial B_r$. This difficulty is resolved by the following proposition.

Proposition A.4 **Proposition.** *Let $X \subset \mathbb{R}^n$ be an open set with $\partial X = \partial(\mathrm{int}(X^C))$. Then the following statements are equivalent:*

(i) *The closure* $\mathrm{cl}(X)$ *of X is r-regular;*

(ii) *The sets X and* $\mathrm{int}(X^C)$ *are equal to unions of open balls with radius r.*

Proof. (i) ⇒ (ii): Observe that $\partial(\mathrm{cl}(X)) = \partial(\mathrm{int}(X^C)) = \partial X$, where the last equality follows by assumption. For any $x \in X$, we claim that there exists an open r-ball containing x and which does not intersect ∂X. Then, since the ball is connected, it will be contained in X.

Suppose first that $\delta_{\partial X}(x) \geq r$. Then the open r-ball with centre x avoids ∂X. Otherwise, suppose $\delta_{\partial X}(x) \in (0, r]$. Then x is contained in $\mathrm{int}(\mathcal{B}^i_r(p_{\partial X}(x)))$, which is an open r-ball that avoids ∂X. Thus X is a union of open r-balls. A similar argument, replacing $\mathrm{cl}(X)$ with X^C, shows that $\mathrm{int}(A^C)$ is also a union of open r-balls.

(ii) ⇒ (i): Let $x \in \partial X$, and let $\{x_i\} \subset X$ be a sequence converging to x. For $i = 1, 2, \dots,$ let A_i be an open r-ball containing x_i such that $A_i \subset X$, and let a_i be the centre of A_i. We have

(A.2)
$$|a_i - x| \leq |a_i - x_i| + |x_i - x| < r + |x_i - x|.$$

Since $|x_i - x| \to 0$ as $i \to \infty$, $\{a_i\}$ is bounded and thus has a convergent subsequence. Denote by a the limit of the bounded subsequence. Replacing $\{a_i\}$ with this subsequence and $\{x_i\}$ with its corresponding subsequence, we can suppose $x_i \to x$ and $a_i \to a$. Letting $i \to \infty$ in (A.2) yields $|a - x| < r$. Since $A_i \cap \partial X$ is empty, we have $\delta_{\partial X}(a_i) \geq r$, and letting $i \to \infty$ admits $\delta_{\partial X}(a) \geq r$. Combining this with $|a - x| < r$, we obtain

$$|a - x| = r = \delta_{\partial X}(a),$$

so $B_r(a)$ contains x and is contained in $\mathrm{cl}(X)$.

We have $x \in \partial X = \partial(\mathrm{int}(X^C)) = \partial(\mathrm{cl}(X))$. Arguing as above with X replaced by $\mathrm{int}(X^C)$ yields a closed r-ball containing x and contained in X^C.

Now we can argue as in the proof of Proposition A.1 and conclude that $\mathrm{cl}(X)$ is r-regular. □

Definition A.5

Definition. *We define a map $N\colon \partial A \to \mathbb{R}^n$ by letting $N(x)$ be the outward-pointing unit normal at x to the boundary of the ball $\mathcal{B}_r^i(x) \subset A$.*

Notice that $-N(x)$ is then the outward pointing unit normal to the boundary of $\mathcal{B}_r^o(x) \subset \mathrm{cl}(A^C)$. For $s \in (-r, r)$,

$$\begin{cases} x + sN(x) \in A & \text{if } s \leq 0; \\ x + sN(x) \in \mathrm{cl}(A^C) & \text{if } s \geq 0. \end{cases}$$

It follows from Corollary A.3 that if $x \in \partial A$ and $s \in (-r, r)$, then $p_{\partial A}(x + sN(x)) = x$. Further, if $y \in U_r^i$ and $s = \delta_{\partial A}(y) < r$, then $y = p_{\partial A}(y) - sN(p_{\partial A}(y))$, and similarly, if $y \in U_r^o$ and $s = \delta_{\partial A}(y) < r$, then $y = p_{\partial A}(y) + sN(p_{\partial A}(y))$.

Definition A.6

Definition. *For $X, Y \subset \mathbb{R}^n$, a function $f\colon X \to Y$ is* ***Lipschitz continuous*** *if there exists a constant $k \geq 0$ such that*

$$|f(x) - f(y)| \leq k|x - y|$$

for all $x, y \in X$. The constant k is called a ***Lipschitz constant*** *for f.*

We now state the results from [12] without the additional assumption of connectedness for r-regular sets since it is not used in the proof.

Proposition A.7

Proposition (Duarte and Torres, 2014 [12]).

(i) *The map $N\colon \partial X \to \mathbb{R}^n$ from Definition A.5 is Lipschitz continuous with Lipschitz constant $1/r$.*

(ii) *Let $s \in (0, 2r/3)$. Then, with U_s^i defined as in (A.1), the map $p_{\partial U_s^i}$ is Lipschitz continuous with Lipschitz constant $\sqrt{2r/(2r - 3s)}$.*

(iii) *Let $f\colon U_r^i \to \mathbb{R}$ be the function given by*

(A.3)
$$f(x) = \langle (x - p_{\partial A}(x)), N(p_{\partial A}(x)) \rangle.$$

Then for $s \in (0, 2r/3)$, $f_{|U_s^i}$ is continuously differentiable with gradient function given by $\mathrm{grad} f(x) = N(p_{\partial A}(x))$ for all $x \in U_s^i$.

(iv) *The boundary $\partial A = f^{-1}(0)$ is a C^1 submanifold of $\mathbb{R}^n$ of dimension $n - 1$.*

Proof. (i)–(iii) are proved as [12, Propositions 6, 7, and 8]; (iv) is [12, Theorem 2] and follows from (iii) because $\mathrm{grad} f$ is nowhere zero on ∂A. As mentioned, none of these results employ connectedness of A. □

Consider for $s \in (0, r)$ the s-parallel set A^s. By some abuse of notation, we denote by A^{-s} the s-parallel set of $\mathrm{cl}(A^C)$. Intuitively, one could get the idea that the boundaries ∂A^s and ∂A^{-s} are $(r-s)$-regular sets; that moving ∂A to either of these boundaries corresponds to shrinking the ball $\mathcal{B}^i_r(x)$ and enlarging the ball $\mathcal{B}^o_r(x)$, or the other way around, for any $x \in \partial A$. In fact, it turns out that this intuition is correct. Let $A_\ominus$ be defined by (2.1).

Proposition A.8

Proposition. *Let $s \in (0, r)$. Then $\mathrm{cl}(A_\ominus(s)) = \bar{A}_\ominus$ is $(r-s)$-regular.*

Proof. Let $\{N(x) \mid x \in \partial A\}$ be the outward-pointing unit normal bundle for ∂A. Then

$$\partial \bar{A}_\ominus = \{a \in A \mid \delta_{\partial A}(a) = s\} = \bigcup_{x \in \partial A} \{x - sN(x)\}.$$

For $x \in \partial A$, let $x' = x - sN(x) \in \partial \bar{A}_\ominus$, and let $x'' = x - rN(x) = x' - (r-s)N(x)$. Let $y' \in \partial \bar{A}_\ominus$ such that $y' \neq x'$, where y' is defined similarly to x', so $y' = y - sN(y)$ for some $y \in \partial A$ with $y \neq x$.

We now have

$$|x'' - y| \leq |x'' - y'| + |y' - y| = |x'' - y'| + s.$$

Now x'' is the centre of $\mathcal{B}^i_r(x) \subset A$, and $\mathcal{B}^i_r(x) \cap \partial A = \{x\}$, so $r < |x'' - y|$. Thus $r < |x'' - y'| + s$, whence $r - s < |x'' - y'|$. This means that $B_{(r-s)}(x'') \subset \bar{A}_\ominus$ and $B_{(r-s)}(x'') \cap \partial \bar{A}_\ominus = \{x'\}$.

Now let $x^* = x' - (r-s)N(x) = x - (2s-r)N(x)$, so $x^* \notin \bar{A}_\ominus$. Consider first the case where $2s < r$ so that $x^* \in A^C$. Let y' and y be defined as above. Let z be a point of intersection of ∂A and the line segment joining y' to x^*, and notice that z cannot be equal to x. We have

$$|x^* - y'| = |x^* - z| + |z - y'|.$$

Furthermore, $|z - y'| \geq s$ since $\delta_{\partial A}(y') = |y' - y| = s$, and $|x^* - z| > r - 2s$ since $\delta_{\partial A}(x^*) = |x^* - x| = r - 2s$. Hence $|x^* - y'| > s + (r - 2s) = r - s$, and thus $B_{(r-s)}(x^*) \subset A^C_\ominus$ and $B_{(r-s)}(x^*) \cap \partial \bar{A}_\ominus = \{x'\}$.

Now consider the case $2s \geq r$. Here, $x^* \in A$. Let y' and y be defined as above. Since $\delta_{\partial A}(y') = s$ and y is the unique point on ∂A satisfying $|y - y'| = s$,

$$s < |y' - x| \leq |y' - x^*| + |x^* - x| = |y' - x^*| + 2s - r,$$

so that $|y' - x^*| > s - (2s - r) = r - s$. Hence we again conclude that $B_{(r-s)}(x^*) \subset A^C_\ominus$ and $B_{(r-s)}(x^*) \cap \partial \bar{A}_\ominus = \{x'\}$.

By the same argument as in Proposition A.1, it now follows that $\bar{A}_\ominus$ is $(r-s)$-regular. □

As a consequence of Proposition A.8, we have:

Corollary A.9

Corollary. *Let $s \in (-r, r)$ and define*

(A.4)

$$A_f(s) = A_f = \{a \in U^i_r \mid f(a) \leq s\}.$$

for $f \colon U^i_r \to \mathbb{R}$ the function defined in (A.3). Then A_f is $(r-a)$-regular.

Proof. For $a \in U^i_r$, we have $f(a) = \langle (a - p_{\partial A}(a)), N(p_{\partial A}(a)) \rangle$, and so $|f(a)| = \delta_{\partial A}(a)$. Thus

$$A_f = \begin{cases} \bar{A}_\ominus(s) = \{a \in A \mid \delta_\partial(a) \geq s\} & \text{if } s \leq 0; \\ \bar{U}^i_s(\mathrm{cl}(A^C)) = \{a \in A \mid \delta_{\partial A} \leq s\} & \text{if } s \geq 0. \end{cases}$$

Since A and $\mathrm{cl}(A^C)$ are r-regular, the result now follows from Proposition A.8. □

The boundaries $A_f(s)$, $s \in (-r, r)$, are C^1 submanifolds of $\mathbb{R}^n$ by Proposition A.7(iv), and they share a common normal bundle in the sense that if a' is an element of $\partial A_f(s)$, then $a' = a - sN(a)$ for some unique $a \in \partial A$, and the normal for $\partial A_f(s)$ at a' is $N(a)$. It follows also that f is in fact continuously differentiable on all of U^i_r.

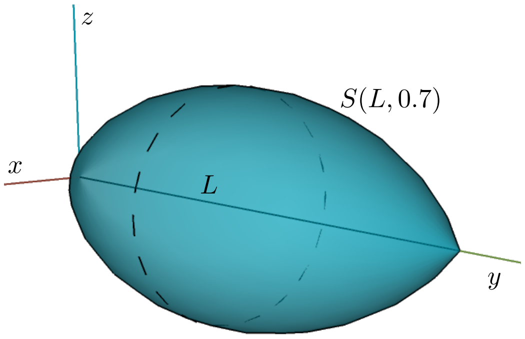


Figure A.1
The figure shows the 0.7-spindle $S(L, 0.7)$ in $\mathbb{R}^3$ of a line L with endpoints $(0, 0, 0)$ and $(0, 1, 0)$.

Proposition A.10

Proposition. *Let $s_1, s_2 \in (-r, r)$ with $s_1 < s_2$ and $f \colon U_r^i \to \mathbb{R}$ the function defined in (A.3). Then:*

(i) *the preimage $f^{-1}[s_1, s_2]$ has C^1 boundary $\partial A_f(s_1) \cup \partial A_f(s_2)$;*

(ii) *the preimage $f^{-1}[s_1, s_2]$ is C^1-diffeomorphic to $A \times [0, 1]$.*

Proof. (i): Clearly $f^{-1}[s_1, s_2]$ has boundary $f^{-1}(\{s_1, s_2\})$, which is C^1 as remarked above. Further, $f^{-1}[s_1, s_2] = \partial A_f(s_1) \cup \partial A_f(s_2)$ by Corollary A.9. This proves (i).

(ii): We now observe that $f^{-1}[s_1, s_2] = \{x + sN(x) \mid x \in \partial A,\ s \in [s_1, s_2]\}$, so that the map $\phi \colon A \times [0, 1] \to f^{-1}[s_1, s_2]$ given by $\phi(x, t) = x + ((1 - t)s_1 + ts_2)N(x)$ is a homeomorphism. It is not, however, a C^1-diffeomorphism when N is not continuously differentiable. Instead, we consider a close C^1 approximation $\eta \colon \bar{U}_r^i \to \mathbb{R}^n$ to $N \colon \bar{U}_r^i \to \mathbb{R}^n$. Let γ be an integral curve of η. We note that

$$\frac{\mathrm{d}}{\mathrm{d}t}(f \circ \gamma)(t) = \operatorname{grad} f_{\gamma(t)} \gamma'(t) = \langle N(\gamma(t)), \eta(\gamma(t)) \rangle,$$

which we can suppose is uniformly close to 1. It now follows from general theory that there is an open neighbourhood W of $f^{-1}(s_1) \times \{0\}$ in $f^{-1}(s_1) \times \mathbb{R}$ and a C^1-diffeomorphism $\Gamma \colon W \to U_r^i$ such that $t \mapsto \Gamma(x, t)$ is the complete integral curve through x for each $x \in f^{-1}(s_1)$. In particular, $\Gamma^{-1}(f^{-1}(s_2))$ is a C^1 submanifold of W,

$$\Gamma^{-1}(f^{-1}(s_2)) = \{(x, T(x)) \mid x \in f^{-1}(s_1)\},$$

where $T \colon f^{-1}(s_1) \to (0, \infty)$ is a C^1-function. Define now a map $\psi \colon f^{-1}(s_1) \times [0, 1] \to f^{-1}[s_1, s_2]$ by

$$\psi(x, t) = \Gamma(x, tT(x)).$$

Then ψ is the required C^1-diffeomorphism, which proves (ii). □

A.2 Some geometry of r-regular sets

We proceed in this section to examine the geometry of r-regular sets in order to establish the notation and results necessary for proving Theorem 2.10. This entails a review of certain results from [38], which are generalised from $\mathbb{R}^3$ to $\mathbb{R}^n$, as well as the introduction of *spindles*.

Definition A.11

Definition. *Let L be a closed line segment in $\mathbb{R}^n$ of length $|L| < 2r$. The **r-spindle** $S(L, r)$ generated by L is the intersection of all r-balls whose boundaries contain both endpoints of L.*

Example A.12

Example. In the Euclidean plane, $\mathbb{R}^2$, there are just two closed r-balls, or, since $n = 2$, r-discs, B_r and B_r' whose boundaries contain both endpoints of L. The discs B_r and B_r' are each the mirror image of the other in the affine line containing L.

In $\mathbb{R}^3$, each closed r-ball with boundary containing the endpoints of L is uniquely determined by its centre, and these centres trace out a circle in the affine plane orthogonal to L and passing through its midpoint. The intersection of all these balls is the solid of revolution obtained by rotating a two-dimensional r-spindle generated by L around the affine line containing L. An example of a spindle in $\mathbb{R}^3$ is shown in Figure A.1.

Lemma A.13 **Lemma.**

(i) *Let $0 < s \leq r$, and let $M \subset N \subset \mathbb{R}^n$ be closed line segments of length $|M| \leq |N| < 2s$. Then $S(M,r) \subset S(N,s)$.*

(ii) *Let $M \subset \mathbb{R}^n$ be a closed line segment of length $|M| < 2r$. Then $S(M,r)$ is the intersection of all closed balls of radius at most r that contain M.*

(iii) *Let M be a closed line segment of length $|M| < 2r$ in $\mathbb{R}^n$. Then $S(M,r)$ is the intersection of all closed balls of radius less than r that contain M.*

Proof. (i): Intersecting with affine planes containing M allows us to reduce the problem of (i) to the case $n = 2$. So let $M \subset N \subset \mathbb{R}^2$. Denote by a and b the endpoints of M and by $\alpha(x)$ the angle $\angle axb$ for $x \in \mathbb{R}^2$. Then

$$S(M,r) = \left\{ x \in \mathbb{R}^2 \;\middle|\; \alpha(x) \text{ is obtuse with } \sin(\alpha(x)) \leq \tfrac{|M|}{2r} \right\}.$$

This equality follows from the fact, already known to Euclid, that the angle subtended by a chord at points of a circle is constant in the corresponding major arc and minor arc together with a calculation of the sine of this angle: If the circle has radius r and the chord length l, then this sine is $l/2r$ for both the major and the minor arc. Analogously, with c and d the endpoints of N and $\beta(x)$ the angle $\angle cxd$ for $x \in \mathbb{R}^2$,

$$S(N,s) = \left\{ x \in \mathbb{R}^2 \;\middle|\; \beta(x) \text{ is obtuse with } \sin(\beta(x)) \leq \tfrac{|N|}{2s} \right\}.$$

Now let $x \in S(M,r)$. Then $\alpha(x)$ is obtuse, so the larger angle $\beta(x)$ is, too. Moreover, $|M| \leq |N|$ and $r \geq s$ imply $|M|/(2r) \leq |N|/(2s)$, and so $x \in S(N,s)$ as required, thus proving (i).

(ii): Now write S for the intersection of all closed balls of radius at most r which contain M. By definition, $S \subset S(M,r)$. Let B_s be a closed s-ball containing M, where $s \leq r$. The intersection of B_s with the affine line containing M is a closed line segment N containing M. By (i), $S(M,r) \subset S(N,s)$, and $S(N,s) \subset B_s$. Thus $S(M,r) \subset S$, which concludes the proof of (ii).

(iii): Let S be the intersection of all closed balls of radius less than r which contain M. Then clearly $S(M,r) \subset S$. Let $x \in S$. Then $x \in S(M,s)$ for all $s \in (|M|/2, r)$. We need to show that $x \in S(M,r)$. Intersecting with the affine plane containing M and x allows us to reduce to the case $n = 2$. Denote by a and b the endpoints of M and by $\alpha(x)$ the angle $\angle axb$. The characterisation of spindles in the case $n = 2$ in (i) shows that $\alpha(x)$ is obtuse with $\sin(\alpha(x)) \leq |M|/(2s)$ for all $s \in (|M|/2, r)$. Taking the limit as $s \to r$ gives $\sin(\alpha(x)) \leq |M|/(2r)$, and the spindle characterisation of (i) yields $x \in S(M,r)$. □

Definition A.14 **Definition.** *For X and Y topological spaces with $Y \subset X$, a* ***retraction*** *(of X onto Y) is a continuous map $f\colon X \to Y$ such that $f_{|Y}$ is the identity mapping Id_Y on Y.*

We define a retraction $\rho\colon A \cup U_r \to A$ by

$$\rho(a) = \begin{cases} p_{\partial A}(a) & \text{for } a \in U_r \setminus A; \\ a & \text{for } a \in A. \end{cases} \tag{A.5}$$

Continuity of ρ follows from the observation $p_{\partial A}(a) = a$ for $a \in \partial A$.

The following proposition is shown in [38] in $\mathbb{R}^3$. For the approach of Stelldinger et al. to the proposition and its subsequent corollary, see [38, Theorem 6 and Lemma 8]. Recall the definition of $U_r(A) = U_r$ from (A.1).

Proposition A.15 **Proposition (Stelldinger et al., 2007 [38]).** *Let $x, y \in A$ with $|x - y| < 2r$, let $L \subset \mathbb{R}^n$ be the line segment from x to y, and let ρ be the retraction defined in (A.5). Then:*

(i) *the line segment L is an element of $A \cup U_r$ and $\rho_{|L}$ is injective;*

(ii) *for $s \leq r$ and B_s any s-ball containing x and y, $\rho(L)$ is a subset of B_s.*

Proof. (i): Let $z \in L \setminus A$. Since z is on the line segment joining x and y, $|x - z| + |z - y| = |x - y|$, and $|x - y| < 2r$, so at least one of $|x - z|$ and $|z - y|$ is strictly smaller than r. Hence $\delta_{\partial A}(z) = \delta_A(z) < r$, so $z \in U_r$. It follows that $L \subset A \cup U_r$.

Suppose now that ρ is not injective: We assume $z_1, z_2 \in L$ with $z_1 \neq z_2$ and $\rho(z_1) = \rho(z_2) =: w$. Thus $z_1, z_2 \in \rho^{-1}(w) = \{w + sN(w) \mid s \in [0, 2r)\}$. In particular, at least one of z_1 and z_2, let us say z_1 for definiteness, is contained in the open line segment $\lambda = \{x + sN(w) \mid s \in (0, 2r)\}$, which is contained in the interior of $\mathcal{B}^o_r(w)$, i.e. in A^C.

The unique affine line l containing x and y also contains z_1 and z_2, which lie between x and y on l. The open line segment λ is a connected open subset of l that intersects L at z_1 but does not meet the endpoints x and y, which are contained in A. It follows that $\lambda \subset L$. This gives a contradiction, since λ has length $2r$, while L has length strictly less than $2r$. Thus our supposition was false, and ρ is injective.

(ii): Let $z \in L \setminus A$ and write $w = \rho(z)$ so that $z = w + tN(w)$ for some $t \in (0, r)$. Let P be the affine plane containing L and w. Letting B_s be an s-ball containing x and y, $C = P \cap B_s$ is an s'-disc containing L for some $s' \in (0, s]$, whilst $B = P \cap \mathcal{B}^o_r(w)$ is an r-disc, because P contains the centre $w + rN(w)$ of $\mathcal{B}^o_r(w)$. The set $M = B \cap L$ is the line segment whose interior is contained in A^C and hence avoids x and y. Since $z \in M$, we conclude that $M \subset L$.

Let $S(B)$ be the intersection of B with its mirror image across the affine line l containing L. Similarly, let $S(C)$ be the intersection of C with its mirror image across l. Clearly $M = l \cap S(B)$, and $M \subset L \subset S(C)$. By Lemma A.13(i), $S(B) \subset S(C)$. Thus $z \in S(B) \subset S(C) \subset C$, and we can conclude that $\rho(L) \subset C \subset B_s$. □

From Proposition A.15(i), it follows that $\rho(L)$ is a simple curve in A joining x and y.

Corollary A.16 **Corollary.** *Let $x, y \in A$ with $|x - y| < 2r$, let $L \subset \mathbb{R}^n$ be the line segment from x to y, and let ρ be the retraction defined in (A.5). Then $\rho(L) \subset S(L, r)$.*

Proof. This is immediate from Proposition A.15(ii) and the definition of spindles. □

Lemma A.17 **Lemma.** *Let $x, y \in A$ with $|x - y| < 2r$ and let $L \subset \mathbb{R}^n$ be the line segment from x to y. Then orthogonal projection P of $\mathbb{R}^n$ onto the affine line l containing L gives a homeomorphism $P\colon \rho(L) \to L$.*

Proof. The orthogonal projection P maps $S(L, r)$ onto L. Thus, since $\rho(L)$ is a connected path from x to y, so is $P(\rho(L))$, hence P maps $\rho(L)$ onto L.

Suppose $v, w \in \rho(L)$ with $P(v) = P(w) =: z$ and, without loss of generality, that $|v - z| \geq |w - z|$. Since the centre of the r-ball $\mathcal{B}^o_r(w)$ lies on the opposite side of z to v in the affine line $v + N(v)\mathbb{R}$, $\mathcal{B}^o_r(w) \cap P^{-1}(z)$ is a closed $(n-1)$-dimensional ball C whose boundary has v as unique nearest point to z. Thus $w \in \operatorname{int}(C) \subset \operatorname{int}(\operatorname{cl}(A^C)) = A^C$. This contradicts the fact that $w \in \rho(L) \subset A$, so no such v and w can exist. Thus $P|_{\rho(L)}$ is injective.

Since L is compact, so is $\rho(L)$. Then, since $P|_{\rho(L)}\colon \rho(L) \to L$ is continuous and bijective, it is a homeomorphism. □

Proposition A.18 **Proposition.** *Let $x, y, z \in A$ be three non-collinear points, and suppose x, y, z are all elements of a closed s-ball $B_s \subset \mathbb{R}^n$ for some $s \in (0, r)$. Let $T \subset \mathbb{R}^n$ be the plane triangle with vertices x, y, and z, and write L_x, L_y, and L_z for the three sides of T respectively opposite x, y and z. Then there exists a continuous map $\sigma\colon T \to A$ such that:*

(i) *the image $\sigma(T)$ of T under σ is a subset of B_s;*

(ii) *the function σ is identical to ρ on $L_x \cup L_y \cup L_z$.*

Proof. (i): Let Δ be the plane triangle in $\mathbb{R}^3$ with vertices $(1, 0, 0), (0, 1, 0), (0, 0, 1)$ so that

$$\Delta = \{(s, t, u) \in \mathbb{R}^3 \mid s, t, u \in [0, 1] \text{ and } s + t + u = 1\}.$$

The map $\phi\colon \Delta \to T$ defined by $\phi(s, t, u) = sx + ty + uz$ for $(s, t, u) \in \Delta$ is continuous and bijective because of the non-collinearity of x, y and z, so it is a homeomorphism, and the points of T are uniquely and continuously parametrised by ϕ.

We will define $\sigma\colon T \to A$ via the following parametrisation:

$$\sigma(sx + ty + uz) = \begin{cases} \rho\big((1-u)\rho\big(\tfrac{1}{s+t}(sx + ty) + uz\big)\big) & \text{for } (s, t, u) \in \Delta \setminus \{(0, 0, 1)\}; \\ z & \text{for } (s, t, u) = (0, 0, 1). \end{cases}$$

We need to show that σ is well-defined.

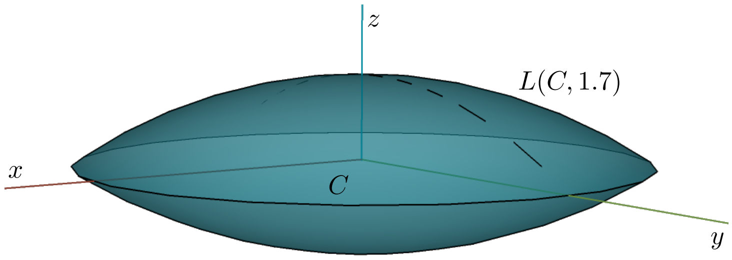


Figure A.2 The figure shows the 1.7-lens $L(C, 1.7)$ of a circle C with radius 1 and centre $(0, 0, 0)$.

This is clear when $(s, t, u) = (0, 0, 1)$, where $\sigma(z) = z \in A \cap B_s$. Now suppose $(s, t, u) \in \Delta \setminus \{(0, 0, 1)\}$. Observe first that $w_{s,t} = (sx + ty)/(s + t) \in L_z$. Since B_s is convex, L_z is a subset of B_s and so has length at most $2s$; the diameter of B_s. Since $s < r$, it follows from Proposition A.15 that $\rho_{|L_z}$ exists and that $\rho(L_z) \subset B_s$. Thus $\rho(w_{s,t})$ is defined and contained in B_s.

Next, observe that $v_{s,t,u} = (1 - u)\rho(w_{s,t}) + uz$ is contained in the line segment $K_{s,t}$ from $\rho(w_{s,t})$ to z. These endpoints are both in B_s, so applying Proposition A.15 again, $\rho_{|K_{s,t}}$ is defined and $\rho(K_{s,t}) \subset B_s$. Thus $\sigma(sx+ty+uz) = \rho(v_{s,t,u})$ is defined and contained in B_s. Hence σ is well-defined and $\sigma(T) \subset B_s$.

We must also see that σ is continuous. This is clear from its definition on $T \setminus \{z\}$, since ρ is continuous. To study continuity at z, let $\{(s_m, t_m, u_m) \mid m \in \mathbb{N}\}$ be a sequence in $\Delta \setminus \{(0, 0, 1)\}$ converging to $(0, 0, 1)$. We must show that $\sigma(s_m x + t_m y + u_m z) \to z$ as $m \to \infty$.

For all $m \in \mathbb{N}$, $w_{s_m,t_m} = (s_m x + t_m y)/(s_m + t_m) \in L_z$, so $\rho(w_{s_m,t_m}) \in \rho(L_z)$. Since L_z is compact, so it $\rho(L_z)$, so $|\rho(w_{s_m,t_m}) - z|$ is bounded.

Now consider $v_{s_m,t_m,u_m} = (1 - u_m)\rho(w_{s_m,t_m}) + u_m z$. We have

$$\begin{aligned} |v_{s_m,t_m,u_m} - z| &= |(1 - u_m)\rho(w_{s_m,t_m}) + u_m z - z| \\ &= |(1 - u_m)(\rho(w_{s_m,t_m}) - z)| \\ &= (1 - u_m)|\rho(w_{s_m,t_m}) - z|, \end{aligned}$$

which converges to 0 as $m \to \infty$. Thus $v_{s_m,t_m,u_m} \to z$ as $m \to \infty$, whence, since ρ is continuous, $\sigma(s_m x + t_m y + u_m z) = \rho(v_{s_m,t_m,u_m}) \to \rho(z) = z$ as $m \to \infty$, as required.

(ii): We now compute σ on the edges of T. We note first that $\sigma(z) = z = \rho(z)$. The line segment $L_x \setminus \{z\}$ is parametrised by $q \mapsto (1 - q)y + qz$ with $q \in [0, 1)$, and $\sigma((1 - q)y + qz) = \rho((1 - q)\rho(y) + qz) = \rho((1 - q)y + qz)$, so $\sigma_{|L_x} = \rho_{|L_x}$.

Similarly, the line segment $L_y \setminus \{z\}$ is parametrised by $q \mapsto (1 - q)x + qz$ for $q \in [0, 1)$, and $\sigma((1 - q)x + qz) = \rho((1 - q)\rho(x) + qz) = \rho((1 - q)x + qz)$, so $\sigma_{|L_y} = \rho_{|L_y}$.

Finally, L_z is parametrised by $q \mapsto (1 - q)x + qy$, $q \in [0, 1]$, and $\sigma((1 - q)x + qy) = \rho(\rho((1 - q)x + qy)) = \rho((1 - q)x + qy)$, so $\sigma_{|L_z} = \rho_{|L_z}$.

In conclusion, $\sigma = \rho$ on $L_x \cup L_y \cup L_z$. □

With the notation of Proposition A.18, it is immediate that $\sigma(T)$ is contained in the intersection $P_r(T)$ of all closed balls of radius less than r which contain T. This set is a little difficult to describe in general but can often be usefully approximated:

Example A.19

Example. In $\mathbb{R}^3$, consider the plane triangle T with vertices x, y, and z, and let C be the **circumcircle** for T, i.e. the unique circle containing x, y, and z. Suppose C has radius $s < r$. Then $P_r(T)$ is contained in the intersection of all closed balls of radius less than r that contain C in their boundaries, which is in fact the intersection of the two closed r-balls containing C in their boundaries; the ***r*-lens** $L(C, r)$ associated to C. An example of a lens is shown in Figure A.2. Intersecting with the three closed r-balls containing T with centre in the plane spanned by T and whose boundaries pass through two of the vertices of T yields the **reduced *r*-lens** $R(C, r)$. ■

In [38, Lemma 11], Stelldinger et al. claim a stronger result in $\mathbb{R}^3$ than Proposition A.18. However, du Plessis found problems in the proof. We state the lemma here and give counterexamples to each of the three statements:

Lemma (Stelldinger et al., 2007 [38, Lemma 11]). *Let $A \subset \mathbb{R}^3$ be an r-regular set for $r \in (0, \infty)$, let $x, y, x \in A$ be such that $|x - y|, |y - z|, |z - x| < 2r$, and let T be the plane triangle with vertices x, y, and z. Then:*

(i) *the restriction* $\rho_{|T}\colon T \to A$ *of* ρ *to* T *is well-defined;*

(ii) *the image* $\rho(T)$ *of* T *under* ρ *is a subset of* $P_r(T)$*;*

(iii) *the image* $\rho(T)$ *of* T *under* ρ *is homeomorphic to a closed disc.*

Example A.20

Example. All three statements are false in their current generality. Indeed, consider $A = \{x \in \mathbb{R}^3 \mid |x| \geq r\}$. Then $A \subset \mathbb{R}^3$ is clearly r-regular.

(i): Let $d \in (r, 2r/\sqrt{3})$ and let T_1 be the plane triangle with vertices $x_1 = d(1,0,0)$, $y_1 = d(-1/2, \sqrt{3}/2, 0)$, and $z_1 = d(-1/2, -\sqrt{3}/2, 0)$. Then T_1 is a equilateral triangle with side length $d\sqrt{3} \in (r\sqrt{3}, 2r)$. We observe that $(0,0,0) \in T_1$ and $(0,0,0) \notin A$, but ρ cannot be defined on T_1, because there is no unique nearest point on ∂A to $(0,0,0)$, since ∂A is the sphere of radius r and centre $(0,0,0)$. Thus $\rho_{|T_1}\colon T_1 \to A$ is not well-defined.

(ii): For d as above, consider the plane triangle T_2 with vertices $x_2 = d(-3d/2, 0, 0)$, $y_2 = y_1$, and $z_2 = z_1$. All three vertices are contained in A. Then T_2 is also an equilateral triangle with side length $d\sqrt{3} \in (r\sqrt{3}, 2r)$, and its circumcircle C has centre $(-d, 0, 0)$ and radius d. Note that $(0,0,0) \notin T_2$, so $\rho_{|T_2}$ is well-defined with $\rho(T_2) \subset A$. Indeed, it is readily verified that $\rho(T_2) = T_2 \setminus \{x \in \mathbb{R}^3 \mid |x| < r\}$.

We claim that T_2 is not contained in any closed ball of radius less than d. For let $s \in (d\sqrt{3}/2, d)$ and suppose T_2 is a subset of some closed s-ball B_s. Let L_1, L_2, and L_3 be the sides of T_2 opposite x_2, y_2, and z_2 respectively. Denote by D the plane disc with boundary C, and observe that D is equal to the union of T and the three d-spindles generated by L_1, L_2, and L_3. By Lemma A.13(i), $S(L_i, d) \subset S(L_i, s)$, and $S(L_i, s) \subset B_s$, so $D \subset B_s$. But this is impossible since D has diameter $2d$ while B_s has diameter $2s < 2d$, so no such B_s exists, proving the claim.

It follows that $P_r(T_2)$ is empty, so $\rho(T_2) \not\subset P_r(T_2)$.

(iii): Let x_3 and y_3 be the endpoints of a closed minor arc B of a circle C with radius r and centre $(0,0,0)$, and let y_3 be an element of the interior of B. Since $C \subset \partial A$, $x_3, y_3, z_3 \in \partial A \subset A$. Let T_3 be the triangle with vertices x_3, y_3, and z_3; it is a genuine triangle since x_3, y_3, z_3 are not collinear. Notice that $T \setminus \{x_3, y_3, z_3\} \not\subset A$. Since its vertices lie in a minor arc of C, T_3 does not contain the centre $(0,0,0)$ of C, so $\rho_{|T_3}$ is well-defined. Since ρ is given by radial projection to ∂A away from $(0,0,0)$ on $\mathbb{R}^n \setminus (A \cup \{(0,0,0)\})$, we have $\rho(T_3) = A$. Certainly, then, T_3 is not homeomorphic to $\rho(T_3)$.

It is also possible to give examples with the triangle vertices in $A \setminus \partial A$. Let a and b be the midpoints of the sides of T_3 opposite x_3 and y_3 respectively. Then $|a|, |b| < r$. Let

$$\varepsilon = \frac{\min\left\{\frac{r}{|a|} - 1, \frac{r}{|b|} - 1\right\}}{2},$$

let $x_3' = (1+\varepsilon)x_3$, $y_3' = (1+\varepsilon)y_3$, and $z_3' = (1+\varepsilon)z_3$, and let T_3' be the triangle with vertices x_3', y_3', and z_3'. Then $\rho_{|T_3'}$ is well-defined, and $\rho(T_3') = T_3' \setminus \{x \in \mathbb{R}^3 \mid |x| < r\}$, which is the union of the arc B with three disjoint sets D_1, D_2, and D_3 each homeomorphic to a closed disc. Hence $\rho(T_3')$ is not homeomorphic to T_3'. ■

It is plausible that (i) and (ii) of [38, Lemma 11] are correct if it is also assumed that the three points $x, y, z \in A$ lie in a closed ball of radius strictly less than r as in Proposition A.18, but we will not consider this question here since the map σ of Proposition A.18 will suffice for our purposes.

However, even with this extra condition, statement (iii) in [38, Lemma 11] does not hold in general, as the examples T_3 and T_3' show, since the vertices involved there can be chosen arbitrarily close to one another.

A.3 Configurations of points and regularity in $\mathbb{R}^3$

We consider the following collections of points in $\mathbb{R}^3$, where $d > 0$:

$$R_1 = \{d(0,0,1), d(1,1,1)\}, \qquad G_1 = \{d(0,1,i), d(1,0,i) \mid i \in \{0,1,2\}\},$$
$$R_2 = \{d(0,0,i), d(1,1,i) \mid i \in \{0,1\}\}, \qquad G_2 = \{d(0,1,i), d(1,0,i) \mid i \in \{0,1\}\},$$
$$R_3 = \{d(0,1,0), d(0,0,1), d(1,1,1)\}, \qquad G_3 = \{d(0,0,0), d(0,1,1), d(1,0,1)\}.$$

The aim of this section is to prove the following theorem.

Theorem A.21 **Theorem (du Plessis, 2016 [11]).** *Let $t_1 = t_2 = \sqrt{3}/2$ and $t_3 = 0.95571$, and let $\Phi\colon \mathbb{R}^3 \to \mathbb{R}^3$ be any isometry. For each $i \in \{1, 2, 3\}$, if $r > t_i d$ and A is an r-regular set, then neither*

$$\Phi(R_i) \subset A \quad \textit{and} \quad \Phi(G_i) \subset A^C$$

nor

$$\Phi(R_i) \subset A^C \quad \textit{and} \quad \Phi(G_i) \subset A$$

can occur.

The proof requires a series of results which we state below.

Lemma A.22 **Lemma.** *It suffices to prove the statements of Theorem A.21 for $\Phi = \mathrm{Id}_{\mathbb{R}^3}$.*

Proof. If $A \subset \mathbb{R}^n$ is r-regular, then so is $\Phi(A)$ for any isometry Φ of $\mathbb{R}^n$ since $B_r \subset \mathbb{R}^n$ is a closed r-ball if and only if $\Phi(B_r)$ is a closed r-ball. □

Lemma A.23 **Lemma.** *Suppose $i \in \{1, 2, 3\}$, and suppose $R_i \subset A$ and $G_i \subset A^C$ cannot occur for any r-regular set A with $r > t_i d$ for $t_1 = t_2 = \sqrt{3}/2$ and $t_3 = 0.95571$. Then $R_i \subset A^C$ and $G_i \subset A$ cannot happen either.*

Proof. Suppose $R_i \subset A^C$ and $G_i \subset A$ hold for some r-regular set A with $r > t_i d$. Let $m = \min\{\delta_A(x) \mid x \in R_i\}$, choose $\varepsilon \in (0, \min\{r - t_i d, m\})$, and consider

(A.6)
$$X_\varepsilon = \{x \in A^C \mid \delta_A(x) \geq \varepsilon\}$$

Then X_ε is $(r - \varepsilon)$-regular by Proposition A.8, and we have $R_i \subset X_\varepsilon$, since $\varepsilon < m$, and $G_i \subset X_\varepsilon^C$, since $A \subset X_\varepsilon^C$. But $r - \varepsilon > t_i d$, so by our hypothesis, this cannot occur, and we have reached a contradiction. □

Lemma A.24 **Lemma.** *Let $c = d(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ and $i \in \{2, 3\}$, and suppose $R_i \cup \{c\} \subset A$ and $G_i \subset A^C$ cannot occur for any r-regular set A with $r > t_i d$ for $t_1 = t_2 = \sqrt{3}/2$ and $t_3 = 0.95571$. Then $R_i \subset A$ and $G_i \subset A^C$ cannot happen either.*

Proof. Suppose $R_i \subset A$ and $G_i \subset A^C$ hold for some r-regular set A with $r > t_i d$. Then $c \in A$ would contradict the hypothesis, so $c \in A^C$. Let $m = \min\{\delta_A(x) \mid x \in G_i \cup \{u\}\}$ and $\varepsilon \in (0, \min\{r - t_i d, m\})$, and let X_ε be defined as in (A.6). Then X_ε is $(r - \varepsilon)$-regular by Proposition A.8, and we have $R_i \subset X_\varepsilon^C$, since $A \subset X_\varepsilon^C$, and $G_i \cup \{c\} \subset X_\varepsilon$, since $\varepsilon < m$.

In the case $i = 2$, let Φ be the rotation through $\pi/2$ radians about an axis through c and parallel to the z-axis. For $i = 3$, let Φ be the reflection in the plane through c and parallel to the xz-coordinate plane. Then for $i \in \{2, 3\}$, Φ is an isometry of $\mathbb{R}^3$ fixed on c such that $\Phi(R_i) = G_i$ and $\Phi(G_i) = R_i$. Thus

$$R_i \cup \{c\} = \Phi(G_i \cup \{c\}) \subset \Phi(X_\varepsilon) \quad \text{and} \quad G_i = \Phi(R_i) \subset \Phi\big(X_\varepsilon^C\big) = \Phi(X_\varepsilon)^C.$$

But $r - \varepsilon > t_i$ and $\Phi(X_\varepsilon)$ is $(r - \varepsilon)$-regular, so this contradicts our hypothesis. □

For $c = d(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$, we now define

$$S_1 = R_1, \quad S_2 = R_2 \cup \{c\}, \quad \text{and} \quad S_3 = R_3 \cup \{c\}.$$

It remains to be proven:

Proposition A.25 **Proposition.** *Let $t_1 = t_2 = \sqrt{3}/2$ and $t_3 = 0.95571$, and let $A \subset \mathbb{R}^3$ be any r-regular set. For each $i \in \{1, 2, 3\}$, if $r > t_i d$, then*

$$S_i \subset A \quad \textit{and} \quad G_i \subset A^C$$

cannot occur.

It should be clear that Theorem A.21 follows from this proposition and the three lemmas above. We need an additional four lemmas to prove Proposition A.25.

Lemma A.26 **Lemma.** *Let L be a closed line segment in $\mathbb{R}^3$ of length $2l < 2r$.*

(i) *The maximal distance of points in the r-spindle $S(L,r)$ from L is $r - \sqrt{r^2 - l^2}$, this being attained at all points in the circle of intersection of $\partial S(L,r)$ with the plane orthogonal to L through the midpoint of L.*

(ii) *The angle at the endpoints of L between L and tangents at the endpoints to the minor arcs rotated to give $S(L,r)$ has sine l/r.*

Proof. (i): Let C be a circle of radius r containing the endpoints of L. Then the distance from the centre c of C to L, attained at the midpoint p of L, is $\sqrt{r^2 - l^2}$, which can be seen by applying Pythagoras' theorem to a triangle with vertices c, p, and an endpoint of L. Thus the maximal distance of L from the minor arc that L cuts out from C, which is clearly attained along the radius through p, is $r - \sqrt{r^2 - l^2}$. Since $\partial S(L,r)$ is the surface of revolution obtained by rotating this minor arc about L, the result follows.

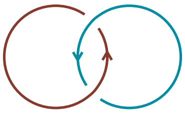
(ii): Let p be the midpoint of L and let C be as above. The tangent to C at an endpoint b of L is orthogonal to the radial line from the centre c of C to b, so the angle between L and this tangent is $\pi/2 - \angle pbc = \angle pcb$, and this angle has sine l/r. □

We observe that for $0 \le a \le r$, the function $r \mapsto r - \sqrt{r^2 - a^2}$ is decreasing on the interval (a, ∞), since its derivative with respect to r at $r \in (a, \infty)$ is $1 - r/(r^2 - a^2)^{3/2}$, which is negative.

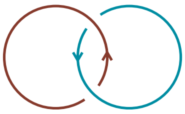
Lemma A.27 **Lemma.** *Let $D \subset \mathbb{R}^3$ be a disc of radius $s < r$. Then the maximal distance of points in the r-lens $L(\partial D, r)$ from D is $r - \sqrt{r^2 - s^2}$, this being attained at the two points of intersection of $L(\partial D, r)$ with the line orthogonal to D through the midpoint of D.*

Proof. Intersecting $L(\partial D, r)$ with a plane orthogonal to D through the midpoint of D yields a two-dimensional r-spindle $S(L,r)$, where L is a line segment with endpoints on ∂D and midpoint p coinciding with the centre of D, and the length of L equals the diameter of D. Arguing as in the proof of Lemma A.26(i), the maximal distance of points of this spindle from L is $r - \sqrt{r^2 - s^2}$, attained on the intersection of $\partial S(L,r)$ with the plane orthogonal to L through p. Rotating about the line orthogonal to D through the midpoint of D yields $L(\partial D, r)$, and the result follows. □

For the proof of the following lemma, we will need the notion of **linking number** of oriented simple closed curves, i.e. oriented closed curves that do not intersect themselves. The linking number, first introduced by Gauss, equals the number of times closed curves wind around each other. For instance, the curves



wind around each other exactly once resulting in a linking number 1, and changing the crossings, but not the orientations, to



changes the sign of the linking number, which is now -1.

Lemma A.28 **Lemma.** *Let $B = B_1(0) \subset \mathbb{R}^3$ with boundary S^2, and let $D \subset B$ be the unit disc in $\mathbb{R}^2 \times \{0\}$ with centre 0.*

Let Γ be the image of a simple path $\gamma\colon [0,1] \to B \setminus \partial D$ from a point in one component of $S^2 \setminus \partial D$ to a point in the other component.

Let $g\colon S^1 \to B \setminus \Gamma$ be homotopic to a homeomorphism $h\colon S^1 \to \partial D$ through maps to $B \setminus \Gamma$. Then g is not homotopic to a constant map through maps to $B \setminus \Gamma$.

Proof. Let $a > 1$, let α be a simple parametrisation of the line segment from $a\gamma(0)$ to $a\gamma(1)$, and let β be a simple parametrisation of a great circle in aS^2 from $a\gamma(0)$ to $a\gamma(1)$. Then the join of α and β yields an oriented simple closed curve in $\mathbb{R}^3$ which has linking number ± 1 with the oriented closed curve given by any homeomorphism $h\colon S^1 \to \partial D$.

The parametrised line segment from $\gamma(0)$ to $\gamma(1)$ is clearly homotopic relative to its endpoints to γ through maps to $B \setminus \partial D$. Thus replacing this parametrised line segment with γ gives another simple closed curve μ in $\mathbb{R}^3$ with linking number ± 1 with the oriented closed curve given by h.

For $\mathrm{Im}(\mu)$ the image of μ, the theory of linking numbers yields

$$H_1(\mathrm{Im}(\mu)^C) \cong h_*(H_1(S^1)) \cong H_1(S^1) \cong \mathbb{Z}.$$

It follows that h is not homotopic to a constant map through maps to $\mathrm{Im}(\mu)^C$. Evidently h is not homotopic to a constant map through maps to $B \setminus (B \cap \mu) = B \setminus \Gamma$. Since g is homotopic to h through such maps, g is not homotopic to a constant map through such maps either. □

For this next lemma, consider a continuous path $\gamma\colon [0,1] \to X$, where X is any metric space. Then γ is ***m*-piecewise injective** if there exists a sequence $\{t_0, \dots, t_m\} \subset [0,1]$ with $0 = t_0 < t_1 < \dots < t_{m-1} < t_m = 1$ such that $\gamma_{|[t_{i-1},t_i]}$ is injective for $i \in \{1, \dots, m\}$.

Lemma A.29

Lemma. *Let X be a metric space. Let $n \in \mathbb{N}$, let $x_0, x_1 \in X$ with $x_0 \neq x_1$, and let $\gamma\colon [0,1] \to X$ be a continuous path from $x_0 = \gamma(0)$ to $x_1 = \gamma(1)$ which is m-piecewise injective. Then there exists an injective continuous path $\gamma'\colon [0,1] \to X$ from x_0 to x_1 such that $\mathrm{Im}(\gamma') \subset \mathrm{Im}(\gamma)$.*

Proof. Suppose that $T = \{t_0, \dots, t_m\} \subset [0,1]$ with $0 = t_0 < t_1 < \dots < t_{m-1} < t_m = 1$ such that $\gamma_{|[t_{i-1},t_i]}$ is injective for $i \in \{1, \dots, m\}$. If $m = 1$, the result follows since a 1-piecewise injective path is injective. If $m > 1$, we will show that there exists an $(m-j)$-injective continuous path $\gamma'\colon [0,1] \to X$ from x_0 to x_1 with $\mathrm{Im}(\gamma') \subset \mathrm{Im}(\gamma)$, where $1 \le j \le m-1$. The result will then follow by downward induction.

Consider the double-point set for γ,

$$P = \{t \in [0,1] \mid \exists s \in [0,1] \setminus \{t\} \text{ such that } \gamma(s) = \gamma(t)\}.$$

If P is empty, γ is injective and we are done, so suppose that P is non-empty, and set $u = \inf(P)$. Then $u \neq 1$, since if $\gamma(s) = \gamma(1)$ with $s \neq 1$, then $s < 1$ and $s \in P$, so $\inf(P) \neq 1$. Hence $u \in [t_{i-1}, t_i)$ for some $k \in \{1, \dots, m\}$.

Since $u = \inf(P)$, there is a sequence $\{u_m\} \subset P$ with $u_m \to u$ as $m \to \infty$. By definition of P, there exists a sequence $\{v_m\}$ with $v_m \in P$, $v_m \neq u_m$, and $\gamma(v_m) = \gamma(u_m)$. Passing to subsequences, we can suppose $\{v_m\}$ is convergent with limit $v \in [0,1]$. We have

$$\gamma(v) = \lim_{m\to\infty} \gamma(v_m) = \lim_{m\to\infty} \gamma(u_m) = \gamma(u)$$

by continuity of γ. We claim that $u \neq v$. Fur suppose $u = v$. Then there exists $l \in \mathbb{N}$ such that $u_l, v_l \in [t_{k-1}, t_k)$. Now $\gamma_{|[t_{k-1},t_k)}$ is injective and $u_l \neq v_l$, so $\gamma(u_l) \neq \gamma(v_l)$, and we have reached a contradiction. Thus $u \neq v$ as claimed, and $u \in P$.

Let $w = \gamma^{-1}(\gamma(u)) \setminus \{u\}$. Then $w \in P$. We claim that $w > t_k$. For suppose otherwise. Then $w \in (u, t_k]$ since $w \geq \inf(P)$ and $w \neq u$. Now $\gamma_{|[u,t_k]}$ is injective, since $[u, t_k] \subset [t_{k-1}, t_k]$, so $\gamma(w) \neq \gamma(u)$, which is a contradiction. Hence $w > t_k$.

Now let $x = \max(\gamma^{-1}(u))$. This maximum exists because $\gamma^{-1}(u)$ is a closed subset of $[0,1]$. Then $x \in (t_j, t_{j+1}]$ for some $k \le j \le m-1$. Consider the join of $\gamma_{|[0,u]}$ and $\gamma_{|[x,t_{j+1}]}$ obtained by identifying u and x. This join is continuous since $\gamma(u) = \gamma(x)$. It is also injective: Firstly, $[0,u) \notin P$, so $[0,u)$ is not in the double-point set P' of the join. Secondly, $u \notin P'$ since u and x are identified and $\gamma^{-1}(\gamma(u)) \subset [u,x]$. Finally, $\gamma_{|[x,t_{j+1}]}$ is injective, so its points cannot be in P' either. It now follows that the join γ^* of $\gamma_{|[0,u]}$ and $\gamma_{|[x,1]}$ is $(m-j)$-piecewise injective with associated sequence $\{0, t_{j+1}, \dots, t_m\}$. Here, the interval $[0, t_{j+1}]$ is understood as the join of the intervals $[0,u]$ and $[x, t_{j+1}]$ with u and x identified. Clearly, $\gamma^*(0) = x_0$ and $\gamma^*(1) = x_1$, and $\mathrm{Im}(\gamma^*) \subset \mathrm{Im}(\gamma)$.

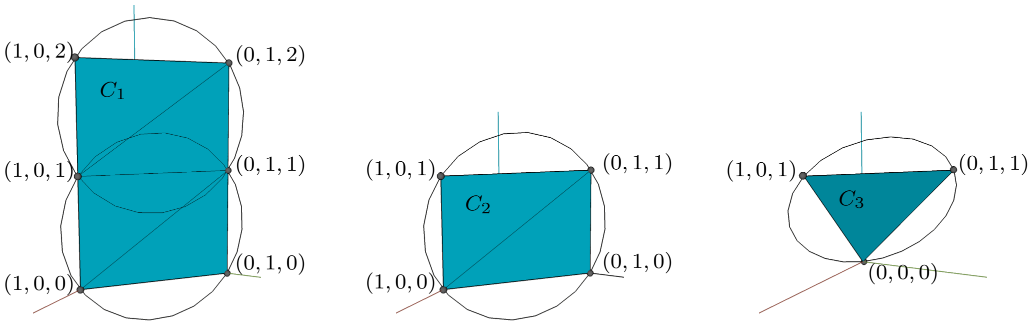
Since joins of closed bounded intervals are homeomorphic to closed bounded intervals in a rather obvious way, we can re-parametrise γ^* to find an $(m-j)$-piecewise injective path γ' with domain $[0,1]$ and the same image as γ^*. □

Proof of Proposition A.25. Let $i \in \{1,2,3\}$. Suppose $A \subset \mathbb{R}^3$ is r'-regular with $r' > t_i d$ and that S_i and G_i satisfy $S_i \subset A$ and $G_i \subset A^C$. Let $m = \min\{\delta_A(x) \mid x \in G_i\}$, let

$\varepsilon \in (0, \min(r' - t_i d, m))$, and let $X_\varepsilon = \{x \in \mathbb{R}^3 \mid \delta_A(x) \geq \varepsilon\}$. Then $r > t_i d$, A and X_ε are r-regular, $A \cap X_\varepsilon = \emptyset$, $S_i \subset A$, and $G_i \subset X_\varepsilon$. We will derive a contradiction, showing that the supposition above is impossible.

Let $i \in \{1, 2, 3\}$. We observe that G_i spans a plane convex set C_i with G_i in its boundary. The set C_i is a union of triangles with vertices in G_i and disjoint interiors, each triangle having circumcircle with radius at most $d\sqrt{3}/2$. The maps σ to X_ε, given by Proposition A.18, associated to these triangles satisfy $\sigma = \rho_{X_\varepsilon}$ on the sides of the triangles, where $\rho_{X_\varepsilon} \colon X_\varepsilon \cup U_r(X_\varepsilon) \to X_\varepsilon$ is a retraction of X_ε defined as in (A.5) and $U_r(X_\varepsilon)$ is given by (A.1). Thus, the maps σ fit together to yield a map $\sigma \colon C_i \to X_\varepsilon$ with $\sigma(C_i)$ contained in the union Q_i of the reduced r-lenses of the circumcircles of the triangle. In more detail:

- the set C_1 is a $2d$-by-$d\sqrt{2}$-rectangle, made up of two d-by-$d\sqrt{2}$-rectangles, and there are just two distinct circumcircles involved: the two circles of radius $d\sqrt{3}/2$ at the centres of the two d-by-$d\sqrt{2}$-rectangles;
- the set C_2 is a d-by-$d\sqrt{2}$-rectangle, and the circumcircles of the two triangles involved coincide as the circle of radius $d\sqrt{3}/2$ centred at the centre of the rectangle;
- the set C_3 is an equilateral triangle with side-length $d\sqrt{2}$. Its circumcircle has radius $d\sqrt{2/3}$.



Let P_i be the plane containing C_i. We note that there is a homeomorphism of pairs $\psi_i \colon (Q_i, Q_i \cap P_i) \to (B, D)$, where $B = B_1^3(0)$ is the closed unit ball and D the closed unit disc centred at 0 in $0 \in \mathbb{R}^2 \times \mathbb{R}$.

Let $H \colon D \to C_i$ be any homeomorphism. Identifying S^1 with ∂D, H restricts to a homeomorphism $h \colon S^1 \to \partial C_i$. Thus S^1 is subdivided into closed arcs with disjoint interiors each mapped homeomorphically by h to a closed line segment in ∂C_i whose intersection with G_i consists of its endpoints. Denote these line segments by $L_1, \ldots, L_m$. Let $j \in \{1, \ldots, m\}$. Then $\rho_{X_\varepsilon}|_{L_j} \colon L_j \to \rho_{X_\varepsilon}(L_j)$ is homotopic to the orthogonal projection of L_j to the boundary arc of $S(L_j, r) \cap P_i$ contained in $\partial Q_i \cap P_i$ through maps to $S(L_j, r)$. Since $\sigma = \rho_{X_\varepsilon}$ on ∂C_i, we conclude that $g' = \sigma \circ h$ is homotopic to a homeomorphism $k \colon S^1 \to \partial Q_i \cap P_i$ through maps to $\bigcup_{j=1}^m S(L_j, r)$.

We now turn our attention to constructing a path in A, from a point in one component of $Q_i \setminus (Q_i \cap P_i)$ to a point in the other, which does not meet $\bigcup_{j=1}^r S(L, r)$. We will find a sequence of points $x_0, \ldots, x_n \in A$ for $n \geq 1$ such that:

(a) the points x_0 and x_n are not contained in Q_i and lie in different components of P_i^C;

(b) the distance $|x_k - x_{k+1}|$ is less than $2r$ for $k \in \{0, \ldots, n-1\}$;

(c) the polygonal path p with vertices $x_1, \ldots, x_n$ in that order, which is the join of the line segments M_k from x_{k-1} to x_k for $k \in \{1, \ldots, n\}$, meets P_i only at points of C_i;

(d) for $k \in \{1, \ldots, n\}$, $S(M_k, r) \cap P_i = \emptyset$ unless $M_k \cap P_i$ is non-empty;

(e) for $k \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$, $S(M_k, r) \cap S(L_j, r) = \emptyset$.

Suppose the above is satisfied. The path $\gamma_1 = \rho \circ p \colon [0,1] \to A$, with $\rho \colon A \cup U_r \to A$ given by (A.5), is a path from x_0 to x_n, and its image is contained in the union $\bigcup_{k=1}^n S(M_k, r)$ so does not meet $\bigcup_{j=1}^m S(L_j, r)$ and meets P_i only in $C_i \setminus \bigcup_{j=1}^m S(L_j, r)$. By Lemma A.29, we can then find an injective path γ_2 from x_0 to x_n with $\Gamma_2 = \operatorname{Im}(\gamma_2) \subset \operatorname{Im}(\gamma_1)$, so $\Gamma_2 \subset A$, $\Gamma_2 \cap \bigcup_{j=1}^m S(L_j, r) = \emptyset$, and $\Gamma_2 \cap P_i \subset C_i \setminus \bigcup_{j=1}^m S(L_j, r)$.

Write W_0 and W_n for the components of P_i^C containing x_0 respectively x_n. Let

$$t_0 = \max\{t \mid \gamma_2(t) \in \partial Q_i \cap W_0\} \quad \text{and} \quad t_n = \min\{t \mid \gamma_2(t) \in \partial Q_i \cap W_n\}.$$

Since $\partial Q_i \cap P_i \subset \bigcup_{j=1}^m S(L_j, r)$, we have $\Gamma_2 \cap \partial Q_i \cap P_i = \emptyset$, so $\gamma_2((t_0, t_n)) \subset \operatorname{int}(Q_i)$.

Re-parametrising $\gamma_2|_{[t_0,t_n]}$ yields a simple path $\gamma_3 \colon [0,1] \to A \cap (Q_i \setminus (\partial Q_i \cap P_i))$ from a point in $W_0 \cap \partial Q_i$ to a point in $W_n \cap \partial Q_i$. Write Γ_3 for the image of γ_3.

Since $\Gamma_3 \cap \bigcup_{j=1}^m S(L_j, r) = \emptyset$, the homotopy from g' to k, described above, is through maps in $G_i \setminus \Gamma_3$. Now apply the homeomorphism ψ_i also described above. We find that the simple path $\gamma = \psi_i \cdot \gamma_3$ with image $\Gamma = \psi_i(\Gamma_3)$ and the map g' satisfy the hypothesis of Lemma A.28 so that g' is not homotopic to a constant map through maps to $B \setminus \Gamma$. Applying the homeomorphism ψ_i^{-1}, it follows that $g = \psi_i \circ g'$ is not homotopic to a constant map through maps to $G_i \setminus \Gamma_3$.

On the other hand, $g = \sigma \cdot h = \sigma \circ H_{|\partial D}$, and the homotopy $F_t \colon \partial D \times [0,1] \to \sigma(D)$ given by $F(x,t) = \sigma \cdot H(tx)$ for $x \in \partial D$ and $t \in [0,1]$ is a homotopy from g to the constant map to $H(0)$. Since $\sigma(D) \subset Q_i \cap X_\varepsilon$, $A \cap X_\varepsilon = \emptyset$, and $\Gamma_3 \subset Q_i \cap A$, we have $\sigma(D) \subset Q_i \setminus \Gamma_3$, so g is in fact homotopic to a constant map through maps to $Q_i \setminus \Gamma_3$. This contradiction will complete the proof. We have yet to show that there exist sequences $\{x_0, \ldots, x_n\} \subset A$ with the properties (a)–(e). The details of the proof are rather different in the three cases corresponding to $i \in \{1,2,3\}$ of the proposition, so we treat them separately.

Case 1: In this case,

$$S_1 = \{d(0,0,1), d(1,1,1)\} \quad \text{and} \quad G_1 = \{d(0,1,i), d(1,0,i) \mid i \in \{0,1,2\}\}.$$

Let $n = 1$ and take $x_0 = d(0,0,1)$ and $x_1 = d(1,1,1)$.

(a): The points x_0 and x_1 clearly lie in different components of the complement to the set $P_1 = \{(x,y,z) \in \mathbb{R}^3 \mid x + y = 1\}$. Further, the maximum distance of points in Q_1 from P_1 is $r - \sqrt{r^2 - d^2/2}$, and since $r > d\sqrt{3}/2$, this distance is less than

$$\frac{d\sqrt{3}}{2} - \sqrt{\left(\frac{d\sqrt{3}}{2}\right)^2 - \frac{d^2}{2}} = \frac{d(\sqrt{3}-1)}{2} < \frac{d\sqrt{2}}{2} = \delta_{P_1}(x_0) = \delta_{P_1}(x_1),$$

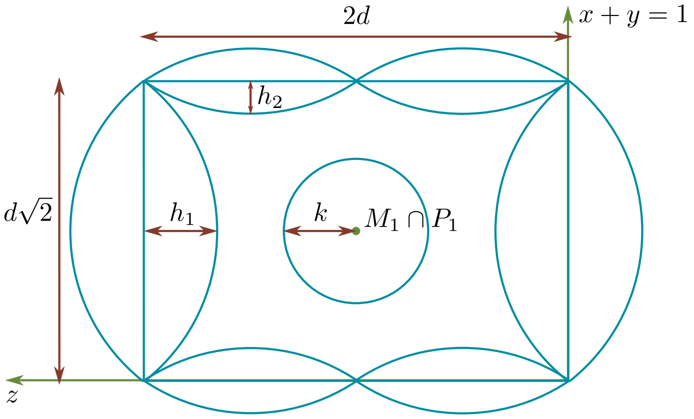
so $x_0, x_1 \notin Q_i$.

(b): This property holds since $|x_0 - x_1| = d\sqrt{2} < 2r$.

(c): The set P_1 meets the line segment M_1 joining x_0 and x_1 at exactly one point; the centre $d(\frac{1}{2}, \frac{1}{2}, 1)$ of C_1.

(d): This property holds trivially.

(e): It suffices to show for $i \in \{1, \ldots, 6\}$ that the orthogonal projections of $S(M_1, r)$ and $S(L_i, r)$ onto P_1 do not intersect. These projections are shown in the following diagram, which in fact also describes the intersections of the spindles with P_1:



Here, the two-dimensional spindles in the diagram are the projections of $S(L_i, r)$ to P_1 coinciding with $S(L_i, r) \cap P_1$, and the circle in the middle is the projection of $S(M_1, r)$ to P_1 coinciding with $S(M_1, r) \cap P_1$.

In the diagram, $k = h_1 = r - \sqrt{r^2 - d^2/2}$ and $h_2 = r - \sqrt{r^2 - d^2/4}$. Since $r > d\sqrt{3}/2$, we have

$$k = h_1 < d\sqrt{3}/2 - \sqrt{\left(\frac{d\sqrt{3}}{2}\right)^2 - \frac{d^2}{2}} = \frac{d(\sqrt{3}-1)}{2}$$

and

$$h_2 < d\sqrt{3}/2 - \sqrt{\left(\frac{d\sqrt{3}}{2}\right)^2 - \frac{d^2}{4}} = \frac{d(\sqrt{3}-\sqrt{2})}{2}.$$

From this, we see that

$$2k + 2h_1 < 2d(\sqrt{3}-1) < 2d$$

and

$$2k + 2h_2 < d(2\sqrt{3} - 1 - \sqrt{2}) < d\sqrt{2}$$

so that the disc $S(M_1, r) \cap P_1$ avoids the two-dimensional spindles $S(L_i, r) \cap P_1$ for all $i \in \{1, \ldots, 6\}$. Thus (e) holds, completing the argument.

Case 2: In this case,

$$S_2 = \{d(0,0,i), d(1,1,i) \mid i \in \{0,1\}\} \cup \{d(\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2})\}$$

and

$$G_2 = \{d(0,1,i), d(1,0,i) \mid i \in \{0,1\}\}.$$

We take $n = 4$ and let $x_0 = (0,0,0)$, $x_2 = d(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$, and $x_4 = d(1,1,1)$.

To define x_1, consider the line segment N_1 from x_0 to $y_0 = d(0,0,1)$, which has length $d < 2r$. Then $\rho_{|N_1}$ is a path in A from x_0 to y_0 with image in $S(N_1, r)$ which meets the plane $W = \{(x,y,z) \in \mathbb{R}^3 \mid z = \frac{1}{2}\}$ in a unique point, which we take as x_1.

Similarly, taking N_4 to be the line segment from x_4 to $y_4 = d(1,1,0)$, $\rho_{|N_4}$ is a path in A from x_4 to y_4 with image in $S(N_4, r)$ meeting W in a unique point, which we take as x_3.

(a): Since $P_2 = \{(x,y,z) \in \mathbb{R}^3 \mid x + y = 1\}$, clearly x_0 and x_4 lie in different components of P_2^C. Further, x_0 and x_4 lie in the sphere of radius $d\sqrt{3}/2$ with centre x_2. Since Q_2 is contained in this sphere and meets its boundary only at the points of G_2, we have $x_0, x_4 \notin Q_2$.

(b): The spindle $S(N_1, r)$ meets W in the disc D_1 with radius

$$r - \sqrt{r^2 - \left(\frac{d}{2}\right)^2} < \frac{d\sqrt{3}}{2} - \sqrt{\left(\frac{d\sqrt{3}}{2}\right)^2 - \frac{d^2}{4}} = \frac{d(\sqrt{3}-\sqrt{2})}{2}$$

and centre $d(0,0,\frac{1}{2})$, so

$$|x_0 - x_1| < \sqrt{\frac{d^2}{4} + \left(\frac{d(\sqrt{3}-\sqrt{2})}{2}\right)^2} = \frac{d\sqrt{6 - 2\sqrt{6}}}{2} < \frac{d\sqrt{3}}{2} < r.$$

Essentially the same calculations shows that $|x_4 - x_3| < r$.

Notice that D_1 is contained in the interior of the disc with radius $d\sqrt{3}/2$ and centre x_2, so $|x_1 - x_2| < d\sqrt{3}/2 < r$. Similarly, $|x_2 - x_3| < r$.

(c): The line segments N_1 and N_4 have constant distance $d\sqrt{2}/2$ from P_2, so their r-spindles $S(N_1, r)$ and $S(N_4, r)$ do not meet P_2 and lie in different components of P_2^C. Since $M_1 \subset S(N_1, r)$ and $M_4 \subset S(N_4, r)$, this implies that x_1 and x_3 lie in different components of P_2^C. It follows that the polygonal path p with vertices $x_0, \ldots, x_4$ meets P_1 only at x_2, the centre of C_2.

(d): Since $M_1 \subset S(N_1, r)$ and $M_4 \subset S(N_4, r)$, we have $S(M_1, r) \subset S(N_1, r)$ and likewise $S(M_4, r) \subset S(N_4, r)$, so $S(M_1, r) \cap P_2 = \emptyset$ and $S(M_4, r) \cap P_2 = \emptyset$.

(e): The minimum distance is $d\sqrt{2}/2 - (r - \sqrt{r^2 - d^2/4})$ from points in $S(N_1, r)$ and $S(N_4, r)$ to P_1, whilst it is $r - \sqrt{r^2 - d^2/2}$ for points in $\bigcup_{j=1}^{4} S(L_j, r)$. Since $r > d\sqrt{3}/2$,

$$\left(r - \sqrt{r^2 - \frac{d^2}{4}}\right) + \left(r - \sqrt{r^2 - \frac{d^2}{2}}\right) < \frac{d(2\sqrt{3} - \sqrt{2} - 1)}{2} < \frac{d\sqrt{2}}{2},$$

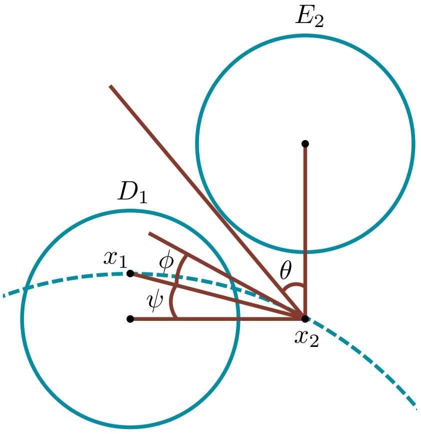
so $S(N_1, r)$ and $S(N_4, r)$ do not intersect $S(L_j, r)$ for any $j \in \{1, \ldots, 4\}$. Hence neither do the r-spindles $S(M_1, r)$ and $S(M_4, r)$ contained in them.

Consider now the r-spindle $S(M_2, r)$. The line segment M_2 has length less than $d\sqrt{3}/2$, so points of this spindle have distance from M_2 less than

$$r - \sqrt{r^2 - \left(\frac{d\sqrt{3}}{4}\right)^2} < \frac{d\sqrt{3}}{2} - \sqrt{\left(\frac{d\sqrt{3}}{2}\right)^2 - \left(\frac{d\sqrt{3}}{4}\right)^2} = \frac{d(2\sqrt{3}-3)}{4}.$$

The r-spindles with axes L_1 and L_3 parallel to W have nearest points to W at a distance $d/2 - (r - \sqrt{r^2 - d^2/2})$ which is larger than $d/2 - d(\sqrt{3}-1)/2 = d(1 - \sqrt{3}/2)$. Since $(2\sqrt{3}-3)/4 < 1-\sqrt{3}/2$, $S(M_2, r)$ does not intersect $S(L_1, r) \cup S(L_3, r)$. Similarly, $S(M_3, r)$ does not intersect $S(L_1, r) \cup S(L_3, r)$.

Finally, we consider the possible intersections of $S(M_2, r)$ and $S(M_3, r)$ with $S(L_2, r)$ and $S(L_4, r)$. It suffices to show that there are no intersections of their projections to W which in fact are their intersections with W. We write $E_2 = W \cap S(L_2, r)$ and similarly $E_4 = W \cap S(L_4, r)$. These are discs of radius $r - \sqrt{r^2 - d^2/2} < d(\sqrt{3} - \sqrt{2})/2$. The symmetry of the situation means that it is enough to consider the possible intersection of $W \cap S(M_2, r)$ with E_2, where we suppose that $\min\{\delta_{E_2}(m) \mid m \in M_2\}$ is at most $\min\{\delta_{E_4}(m) \mid m \in M_2\}$.



Denoting by R_2 the radius of E_2, the angle θ between the line from the centre of E_2 to x_2 to a tangent line to ∂E_2 through x_2 satisfies

$$\sin(\theta) = \frac{2R_2}{d\sqrt{2}} < \frac{2d(\sqrt{3}-\sqrt{2})}{2d\sqrt{2}} = \frac{\sqrt{3}-\sqrt{2}}{\sqrt{2}} = \sqrt{\frac{3}{2}} - 1,$$

so $\theta < 0.23$. The angle ψ between M_2 and the line from x_2 to the centre of D_1 is at most θ, since $x_1 \in D_1$, so $\psi < 0.23$. Lastly, the angle ϕ between M_2 and the tangent at x_2 to the boundary of $W \cap S(M_2, r)$ satisfies

$$\sin(\phi) = \frac{|M_2|}{2r} < \frac{2d\sqrt{3}}{4d\sqrt{3}} = \frac{1}{2},$$

so $\phi < \pi/6$. Thus $\theta + \phi + \psi < 0.98 < \pi/2$, and it follows that $S(M_2, r) \cap S(L_2, r) = \emptyset$.

Case 3: In this case,

$$S_3 = \{d(0,1,0), d(0,0,1), d(1,1,1), d(\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2})\}$$

and

$$G_3 = \{d(0,0,0), d(0,1,1), d(1,0,1)\}.$$

We take $n = 2$ and $x_0 = d(0,0,1)$, $x_1 = d(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$, and $x_2 = d(1,1,1)$.

(a): Now C_3 is the equilateral triangle with side length $d\sqrt{2}$ and vertices the points of G_3. It is contained in the plane $P_3 = \{(x,y,z) \in \mathbb{R}^3 \mid x+y-z = 0\}$. The circumcircle of C_3 has centre $c = d(\frac{1}{3}, \frac{1}{3}, \frac{2}{3})$ ans radius $d\sqrt{2/3}$. It follows that the maximum distance of points of the reduced r-lens Q_3 of C_3 from P_3 is

$$r - \sqrt{r^2 - \frac{2d^2}{3}} < \frac{d\sqrt{3}}{2} - \sqrt{\frac{3d^2}{4} - \frac{2d^2}{3}} = \frac{d}{\sqrt{3}}$$

since $r > t_3 > d\sqrt{3}/2$.

The line segment M_1 from x_0 to x_1 is orthogonal to P_3 and intersects P_3 at c, so $\delta_{P_3}(x_0) = |x_0 - c| = d/\sqrt{3}$, and $x_0 \notin Q_3$.

The line segment from x_2 to $c' = d(\frac{2}{3}, \frac{2}{3}, \frac{4}{3}) \in P_3$ is orthogonal to P_3, so $\delta_{P_3}(x_2) = |x_2 - c'| = d/\sqrt{3}$, and $x_2 \notin Q_3$.

Finally, x_0 and x_2 lie in different components of P_3^C, since the normals from x_0 and x_2 to P_3 points in opposite directions.

(b): We notice that $|x_0 - x_1| = d\sqrt{3}/2 = |x_1 - x_2|$, and $r > d\sqrt{3}/2$.

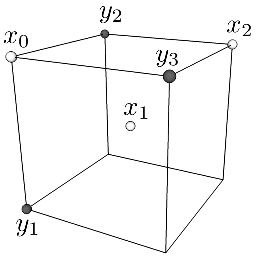
(c): Clearly $x_2 = p \cap P_3 \subset C_3$.

(d): This property holds trivially.

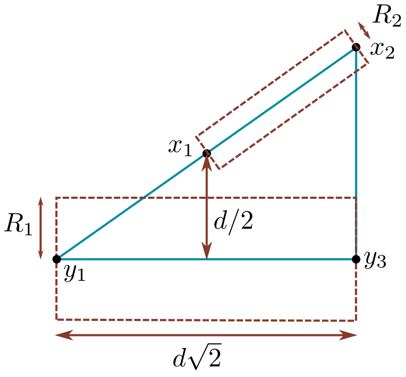
(e): Up to orthogonal symmetries, there are two cases of r-spindle intersection that we must consider:

(1) The case $S(M_2, r) \cap S(L_2, r)$, where L_2 is the line segment from $y_1 = d(0,0,0)$ to $y_3 = d(1,0,1)$;

(2) The case $S(M_1, r) \cap S(L_1, r)$, where L_1 is the line segment from $y_2 = d(0,1,1)$ to y_3.



For case (1), consider the triangle T with vertices y_1, y_3, and x_2. The midpoint of the line segment from y_1 to x_2 is x_1. The r-spindle $S(L_2, r)$ is contained in the orthogonal disc bundle for L_2 of radius $R_1 = r - \sqrt{r^2 - d^2/2}$, whilst $S(M_2, r)$ is contained in the orthogonal disc bundle for M_2 of radius $R_2 = r - \sqrt{r^2 - 3d^2/16}$. We will show that these disc bundles do not intersect. It will suffice to consider the orthogonal projections to the plane containing T of these disc bundles.

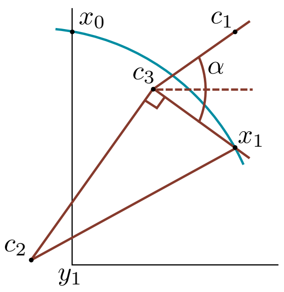


It is clear that it suffices to show that $R_1 + R_2 < d/2$. Since $r > d\sqrt{3}/2$, we have

$$R_1 + R_2 < \left(\frac{d\sqrt{3}}{2} - \sqrt{\frac{3d^2}{4} - \frac{d^2}{2}}\right) + \left(\frac{d\sqrt{3}}{2} - \sqrt{\frac{3d^2}{4} - \frac{3d^2}{16}}\right) = d\left(\sqrt{3} - \frac{5}{4}\right) < \frac{d}{2}.$$

Thus the two disc bundles do not intersect, and this implies $S(M_2, r) \cap S(L_2, r) = \emptyset$.

For the case (2) and Z the plane $\{(x, y, z) \in \mathbb{R}^3 \mid x = y\}$, we consider orthogonal projection to Z. The orthogonal projections of the r-spindles $S(M_1, r)$ and $S(L_1, r)$ are in fact their intersections with this plane, which are the two-dimensional r-spindle for M_1 respectively the closed disc E_1 of radius $r - \sqrt{r^2 - d^2/2}$ centred at $c_1 = d(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. We will show that these do not intersect when $r = t_3 d$, which will imply $S(M_1, t_3 d) \cap S(L_1, t_3 d) = \emptyset$.



Let c_2 be the centre of the closed r-disc E_2 through x_0 and x_1 which has as minor arc the boundary of $Z \cap S(M_1, r)$ above, in the sense of larger z-coordinate, M_1. It will be enough to show that $E_1 \cap E_2 = \emptyset$ or, equivalently, to show that the distance between the centres of the discs, $|c_1 - c_2|$, is greater than the sum of their radii, $r + (r - \sqrt{r^2 - d^2/2}) = 2r - \sqrt{r^2 - d^2/2}$.

The distance l from c_2 to the midpoint c_3 of M_1 is $\sqrt{r^2 - 3d^2/16}$, whilst the distance m from c_3 to c_1 is $|d(\frac{1}{4}, \frac{1}{4}, \frac{3}{4}) - d(\frac{1}{2}, \frac{1}{2}, 1)| = d\sqrt{3}/4$. The triangle with vertices c_3, c_1, and x_1 is isosceles and its angle $\alpha = \angle c_1 c_3 x_1$ satisfies $\sin(\alpha/2) = (d/4)/(d\sqrt{3}/4) = 1/\sqrt{3}$, whence $\sin(\alpha) = 2\sin(\alpha/2)\cos(\alpha/2) = 2\sqrt{2}/3$. Using the cosine formula in the triangle with vertices c_2, c_3, and c_1 yields

$$|c_2 - c_1|^2 = l^2 + m^2 - 2lm\cos(\pi/2 + \alpha) = l^2 + m^2 + 2lm\sin(\alpha),$$

since $\cos(\pi/2 + \alpha) = \cos(\alpha)\sin(\pi/2) - \sin(\alpha)\cos(\pi/2)$, so, substituting, we obtain

$$\begin{aligned}|c_2 - c_1|^2 &= \left(r^2 - \frac{3d^2}{16}\right) + \frac{3d^2}{16} + d\sqrt{r^2 - \frac{3d^2}{16}}\,\frac{d\sqrt{3}}{4}\,\frac{2\sqrt{2}}{3} \\ &= r^2 + d\sqrt{r^2 - \frac{3d^2}{16}}\,\frac{\sqrt{2}}{\sqrt{3}} \\ &= r^2 d\sqrt{\frac{2r^2}{3} - \frac{d^2}{8}}.\end{aligned}$$

It follows that r must satisfy

$$r^2 + d\sqrt{\frac{2r^2}{3} - \frac{d^2}{8}} > \left(2r - \sqrt{r^2 - \frac{d^2}{2}}\right)^2,$$

or, equivalently, that $t = r/d$ must satisfy

$$t^2 + \sqrt{\frac{2t^2}{3} - \frac{1}{8}} - \left(2t - \sqrt{t^2 - \frac{1}{2}}\right) > 0.$$

A calculation shows that $t_3 = 0.95571$ satisfies this inequality.

Thus $S(M_1, t_3 d) \cap S(L_1, t_3 d) = \emptyset$. Since $r > t_3 d$, we have $S(M_1, r) \subset S(M_1, t_3 d)$ and $S(L_1, r) \subset S(L_1, t_3 d)$, so $S(M_1, r) \cap S(L_1, r) = \emptyset$. This concludes the proof. □

A.4 QUASI-MANIFOLD PROPERTIES

Proposition A.25 was the last piece of the proof of Theorem A.21. We will show now how this theorem can be used to prove Theorem 2.10.

Theorem 2.10. *Whenever A is an r-regular set with $d \cdot 0.95571\cdots < r$, $V_{dL}(A)$ is a quasi-manifold.*

Proof. To show that $V = V_{dL}(A)$ is a quasimanifold, according to Definition 2.9, we need to check that configurations (11)–(14) cannot occur, and that configurations (8) and (9) only occur paired with their own or the other's complementary configuration. The third property of quasi-manifolds, that any pair of black voxels in the same component of V is connected by a chain of face-adjacent black voxels, is already satisfied when $r > d\sqrt{3}/2$ by the proof of Proposition 2.7.

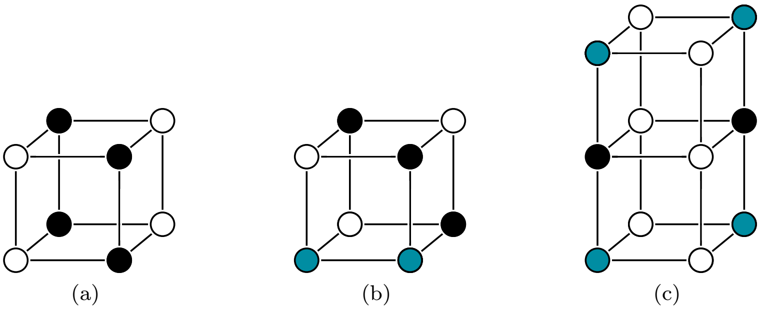


Figure A.3
These configurations of black and white lattice points in a lattice dL cannot occur in the digitisation of an r-regular set satisfying $r > d \cdot 0.95571$. Blue points represent lattice points which can be either black or white.

For $t_1 = t_2 = \sqrt{3}/2$ and $t_3 = 0.95571$, the hypothesis $r > 0.95571$ ensures that $r > t_i$ for $i \in \{1, 2, 3\}$, and so Theorem A.21 applies for each choice of $i \in \{1, 2, 3\}$. Let us depict the situation of the theorem; see Figure A.3. In the figure, the elements of R_i are represented as black points and elements of G_i as white points corresponding to $R_i \subset A$ and $G_i \subset A^C$. The blue points can be elements of either A or A^C. Figure A.3(a) shows R_2 and G_2, (b) shows R_3 and G_3, and (c) shows R_1 and G_1. By Theorem A.21, neither of the configurations in the three subfigures are allowed to occur.

Consider first Figure A.3(a). This is (11) (up to rotation), hence we can rule out this configuration of V.

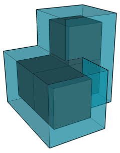
Consider now Figure A.3(b). Replacing both blue points with white respectively black points yields (a rotation of) (12) respectively its complementary configuration; replacing the leftmost blue point with a black and the other with a white point yields (13), and, finally, replacing the leftmost blue point with a white and the other with a black point gives (a rotation of) (14).

It remains to be shown that (8) and (9) can only occur in the pairs specified by Figure 2.4. Consider Figure A.3(c). Replacing the two upper blue points with white points, we obtain a 2-by-2-by-3 configuration with (8) as the upper 2-by-2-by-2 configuration. If we instead replace just one upper blue point with a black and one with a white point, we get (a rotation of) (9) as the upper 2-by-2-by-2 configuration. The figure then implies that we cannot have elements of A^C at the two white points at once. One can now take all other possible configurations of black and white points in the lower 2-by-2-by-1 configuration with either (8) or (9) above, remembering that occurrences of (11)–(14) are not allowed, as proved above, and the resulting possible configurations, up to rotation, reflection, and complementarity, are the three shown in Figure 2.4, hence (8) and (9) can only occur in the pairs claimed.

Reversing the colours in Figure A.3 so points of R_i are white and those of G_i are black for $i \in \{1, 2, 3\}$ leads to similar conclusions as above only for the complementary configurations. In conclusion, V is a quasi-manifold. □

B

TECHNICAL ARGUMENTS OF CHAPTER 2



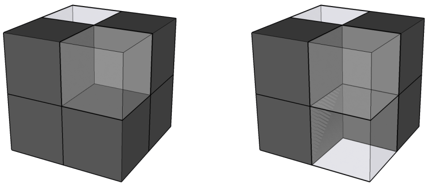
This appendix covers the more technical details left out in Chapter 2. Firstly, in Section B.1, we will formally introduce the wedges inserted into the voxel reconstruction of an r-regular set to obtain the wedged reconstruction whenever the underlying reconstruction is a quasi-manifold. Secondly, in Section B.2, a smoothing of the wedged reconstruction, which is used to obtain a smoothed reconstruction, is defined, and finally, in Section B.3, we are ready to give the definition of a direction field near the boundary of the smoothed reconstruction, which is utilised for the attainment of the necessary vector field on which the main theorem of Chapter 2 relies.

B.1 DEFINITION OF THE WEDGED RECONSTRUCTION

We give now a formal definition of the wedges introduced in Subsection 2.3.2. Recall that $V_{dL}(A) = V$ denotes the voxel reconstruction of an r-regular set A by a lattice dL. For $1 \leq n \leq 10$, denote by $(n)^*$ the complement of a configuration of V, i.e. $(n)^*$ is the configuration obtained by inverting the colours of (n). Then the allowed occurrences of (8) and (9), according to Definition 2.9(ii), are the combinations of these configurations with (8)* and (9)*. These two complementary configurations are illustrated in Figure B.1. The allowed combinations of the four configurations, illustrated in Figure 2.4, are: the 2-by-2-by-3 configuration that consists of (8)* and (8) and corresponds to Figure 2.4(a); the 2-by-2-by-3 configuration that consists of (9)* and (9) and corresponds to Figure 2.4(b); the 2-by-2-by-3 configuration that consists of (8)* and (9) and corresponds to Figure 2.4(c), and the 2-by-2-by-3 configuration that consists of (9)* and (8) and corresponds to the complement of Figure 2.4(c). It is these 2-by-2-by-3 configurations and (10) for which we need to define wedges.

We introduce for each of (8)*, (9)*, and (10) a coordinate system in order to precisely define the wedges. Since we will need this later on, we introduce coordinate systems for configurations (1)–(7) of Figure 2.5 as well, thus covering all configurations of W. Let any such coordinate system have origin in the centre of the configuration and axes parallel to the axis directions of dL. Each coordinate system divides $\mathbb{R}^3$ into eight regions called **octants**. The octants are determined by the signs of the coordinates of the points they contain, e.g. $(+, +, -)$ is the octant where the first and second coordinates of all points are positive and the third coordinate is negative. We specify the axes by imposing the coordinate system in Figure B.2 on each of the ten configurations with the same 'orientation' as they are given in Figure 2.5. More precisely, these coordinate systems are determined by which octants contain black voxels, so these are specified in Table B.1. Configuration (1) is not included in the table since it does not contain any black voxels, and so all possible choices of coordinate system are symmetric. For configurations (2), (5), (6), (7), (8)*, and (10), there are more possible choices of coordinate systems, which are identical up to rotation of the configuration, so one can be chosen arbitrarily.

We are now ready to define the wedges of W. The **convex hull** conv(X) of a set of points $X \subset \mathbb{R}^3$ is the smallest convex set in $\mathbb{R}^3$ containing X.



(8)* (9)*

Figure B.1
The complementary configurations of (8) and (9).

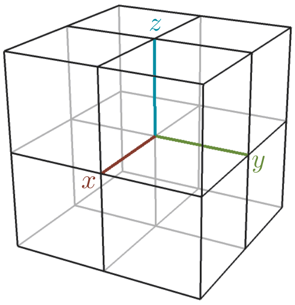


Figure B.2 Coordinate system imposed on configurations (1)–(7) and ⟨1⟩–⟨3⟩ of W.

Table B.1 Specification of coordinate systems for each configuration of W containing black voxels.

Configuration	Octants containing black voxels
(2)	$(-,-,-)$
(3)	$(-,-,-)$, $(-,+,-)$
(4)	$(-,-,-)$, $(-,+,-)$, $(+,-,-)$
(5)	$(-,-,-)$, $(-,+,-)$, $(+,-,-)$, $(+,+,-)$
(6)	$(-,-,-)$, $(-,+,-)$, $(+,-,-)$, $(-,-,+)$
(7)	$(-,-,-)$, $(-,+,-)$, $(+,-,-)$, $(-,+,+)$
(8)*	$(-,-,-)$, $(-,+,-)$, $(+,-,-)$, $(+,+,-)$, $(+,-,+)$, $(-,+,+)$
(9)*	$(-,-,-)$, $(-,+,-)$, $(+,-,-)$, $(+,-,+)$, $(-,+,+)$
(10)	$(+,-,-)$, $(-,+,+)$

Definition B.1

Definition. *Let $\varepsilon > 0$, and set*

$$\begin{aligned}
W_1 &= \operatorname{conv}\{(0,0,0),(\varepsilon,0,0),(0,\varepsilon,0),(0,0,d),(\varepsilon/2,0,d),(0,\varepsilon/2,d)\};\\
W_2 &= \operatorname{conv}\{(\varepsilon/2,0,d),(0,\varepsilon/2,d),(-\varepsilon/2,0,d),(0,-\varepsilon/2,d),(0,0,d+\varepsilon/2)\};\\
W_3 &= \operatorname{conv}\{(0,0,0),(\varepsilon,0,0),(0,\varepsilon,0),(0,0,-\varepsilon)\};\\
W_4 &= \operatorname{conv}\{(0,0,0),(-\varepsilon,0,0),(0,-\varepsilon,0),(0,0,d),(-\varepsilon/2,0,d),(0,-\varepsilon/2,d)\};\\
W_5 &= \operatorname{conv}\{(0,0,0),(0,\varepsilon,0),(0,0,\varepsilon),(-\varepsilon,0,0)\}.
\end{aligned}$$

We define:

⟨1⟩ *as the 2-by-2-by-3 configuration corresponding to (8) paired with (8)*, where we colour the wedge W_1 black in the coordinate system of (8)*;*

⟨2⟩ *as the 2-by-2-by-3 configuration corresponding to (9) paired with (9)*, where we colour the wedges W_1, W_2, W_3, and W_4 black in the coordinate system of (9)*;*

⟨3⟩ *as the 2-by-2-by-3 configuration corresponding to (9) paired with (8)*, where we colour the wedges W_1, W_2, and W_4 black in the coordinate system of (8)*;*

⟨4⟩ *as the 2-by-2-by-3 configuration corresponding to (8) paired with (9)*, where we colour the wedges W_1, W_3, and W_4 black in the coordinate system of (9)*;*

⟨5⟩ *as the configuration obtained by colouring the wedge W_5 in (10) white.*

The wedging defined in Definition B.1 can now be applied to all rotations and reflections of the indicated pairs of configurations, which yields a wedging of all critical cells of the voxel reconstruction V. Notice that the wedge W_2 will intersect the upper black voxel of (9) in the definitions of ⟨2⟩ and ⟨3⟩. This only means that the definition applies to both reflection symmetric configurations composed of (9) and either (8)* or (9)*.

The wedged configurations ⟨1⟩–⟨5⟩ are illustrated in Figure 2.6. For ⟨1⟩ and ⟨5⟩, we in fact need to insert or remove two wedges, but due to the symmetry of the coordinate systems mentioned above, we need only define one and then apply the wedging procedure to all possible choices of coordinate systems for these configurations. Note also that for the complement of (10), instead of colouring the wedge white, we colour it black.

Analogously to adding black wedges to configurations (8)* and (9)*, one could add white wedges inserted into the black voxels and then use the same line of arguments as presented in Chapter 2 to obtain a homeomorphism and an ambient isotopy between A and W. However, this is not the case for (10), where the wedges need to have the opposite colour of the two odd voxels of the configuration. Notice also that for the complement of this configuration, the analogue of inserting white wedges into the black voxels of (10) is

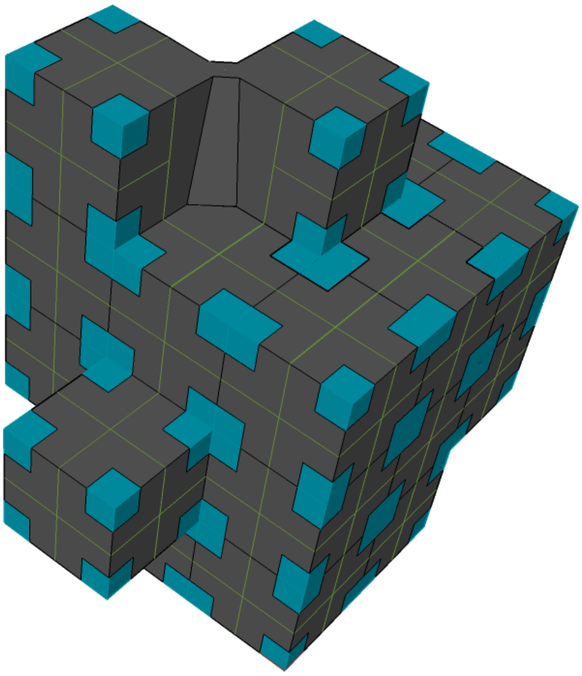


Figure B.3
Example of the loci (green) and centre discs (blue) of a wedged reconstruction. The discs are the black and blue regions bounded by the loci, and the annuli are the black regions which are bounded both by a locus and the boundary of a centre disc.

to insert black wedges into the two white wedges in order to disconnect them and thereby remove the critical vertex.

By construction, W is homotopic to V, but as opposed to V, W has manifold boundary ∂W. Thus, the results of Subsection 2.3.3 will apply to W once we have constructed a smoothed reconstruction Z homeomorphic to W and shown that the results apply to Z.

B.2 The smoothed reconstruction

In order to obtain a smoothing of W, we will define a smoothing for each configuration (1)–(7) of the voxel reconstruction V and for the wedged configurations $\langle 1\rangle$–$\langle 5\rangle$; see Figures 2.5 and 2.6 respectively. These cover all possible configurations of W, so we will simply refer to (1)–(7) and $\langle 1\rangle$–$\langle 5\rangle$ as the configurations of W. The smoothing should be defined in a way such that the smoothed configurations are compatible: when combined, they should give rise to a smoothed voxel reconstruction Z with smooth boundary. At the beginning of Subsection 2.5.2, we argued that the smoothing must take into consideration the subsequent construction of a vector field which should be transverse to the thus smoothed boundary ∂Z. We will keep this in mind as we define smoothings locally of the *loci* and *annuli* of each configuration.

B.2.1 Loci and annuli

Let us consider the lattice dL, where $d\sqrt{3}/2 < r$, on which W is based. In the following, let $K_s(k, dL) = K_s(k) \subset \mathbb{R}^3$ denote a regular cube with side length $s \in \mathbb{R}$, centre $k \in \mathbb{R}^3$, and edges parallel to the axis directions of dL. We use the convention that the **non-critical boundary vertices of W** are the non-critical boundary vertices of V. The wedged reconstruction, W, of course has more non-critical vertices at the boundary due to its wedges, but we treat these separately. The non-critical boundary vertices of V are indeed preserved under the wedging process, since the wedges do not affect other vertices of V than the critical ones. We now define certain subsets of the boundary ∂W which will be used to specify a smoothing of W and, thereafter, the vector field ξ.

Definition B.2

Definition. *Let v be a non-critical boundary vertex of W, and let $\varepsilon > 0$. The* ***disc****, the* ***centre disc****, and the* ***annulus*** *of v are the sets*

$$\mathcal{D}_v = K_d(v) \cap \partial W, \qquad \mathcal{C}_v = K_{2\varepsilon}(v) \cap \partial W, \qquad \mathcal{A}_v = \mathrm{cl}(\mathcal{D}_v \setminus \mathcal{C}_v)$$

respectively. The ***locus*** *of v is $\mathcal{L}_v = \partial\mathcal{D}_v$, and the* ***centre locus*** *of v is $\partial\mathcal{C}_v$.*

Examples of the concepts introduced in this definition are shown in Figure B.3.

The critical cells of V are ‘repaired’ under the wedging process, and new boundary cells arise. Thus an alternative definition of discs and loci is needed in the case of vertices

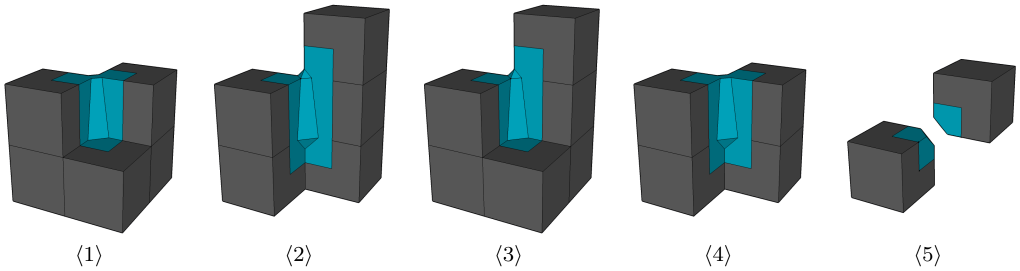


Figure B.4 The disc of each type of wedge in the wedged reconstruction is illustrated in blue; its boundary is the locus.

belonging to wedges. Consider for a moment the five configurations ⟨1⟩–⟨5⟩. The non-critical boundary vertices of these, when viewed as isolated reconstructions, have well-defined loci, and these loci enclose those subsets of the boundary that intersect faces of the wedges.

Definition B.3

Definition. *The* ***disc*** $\mathcal{D}$ *of a wedge of type* $\langle n \rangle$ *in* W *for* $1 \leq n \leq 5$ *is the compact subset of* ∂W *intersecting a face of the wedge and enclosed by the loci of the non-critical boundary vertices of* $\langle n \rangle$*. The* ***locus*** $\mathcal{L}$ *of the wedge is the boundary of its disc,* $\mathcal{L} = \partial\mathcal{D}$*.*

The subsets defined in Definition B.3 are illustrated in Figure B.4. Together, the loci in Definitions B.2 and B.3 are called the **loci of ∂W**. The locus $\mathcal{L}$ (of a vertex or wedge) is an embedding $S^1 \hookrightarrow \mathbb{R}^3$ of the unit circle $S^1 \subset \mathbb{R}^2$ associated to each non-critical boundary vertex and to each wedge of W. The disc, centre disc, and annulus consist of a number of k-cells for all $k = 0, 1, 2$, whereas the locus and the centre locus consist of only 0- and 1-cells. The annulus of a non-critical vertex will be used in the next section to define the smoothing of the wedged reconstruction, but the smoothing of ⟨1⟩–⟨5⟩ will be defined differently, so we do not need the notion of 'annuli of wedges'.

B.2.2 SMOOTHING OF THE LOCI AND ANNULI

Each locus of ∂W consists of a number of 0- and 1-cells. Some 1-cells have length d, and we wish to subdivide these into smaller segments of length $d/2$. Each such 1-cell of a locus of ∂W with endpoints k_1 and k_2 and midpoint m is divided into two line segments: one with endpoints k_1 and m; one with endpoints k_2 and m. These line segments, along with the remaining length $d/2$ 1-cells of the loci, are called the **locus segments** of ∂W. Every locus segment shares an endpoint with up to four other locus segments: two or three on the same boundary face as itself and exactly one on an edge-adjacent boundary face. We will call these locus segments **adjacent** and, in particular, the latter is called the **edge-adjacent locus segment**. We now divide the locus segments into two types. A **type I** locus segment is contained in a boundary face which lies in the same plane as the boundary face containing the corresponding edge-adjacent locus segment; a **type II** locus segment is contained in a boundary face which lies in a plane perpendicular to the boundary face containing the corresponding edge-adjacent locus segment. The two types of locus segments are illustrated in Figure B.5.

In order to define a smoothing of W, we first define a smoothing of the loci of ∂W. For each locus $\mathcal{L}$ of ∂W, we wish to smooth the vertices of $\mathcal{L}$. For any vertex w of $\mathcal{L}$, there are two orthogonal locus segments s and s' which have w as a common endpoint. For $\varepsilon > 0$, define lines ϕ and ψ by the parametric equations

$$\begin{aligned} x_\phi(t) &= \varepsilon, & y_\phi(t) &= \varepsilon t; \\ x_\psi(t) &= \varepsilon - \varepsilon t, & y_\phi(t) &= \varepsilon, \end{aligned} \tag{B.1}$$

for $t \in \mathbb{R}$. The lines ϕ and ψ are illustrated in Figure B.6(a). The locus segments s and s'

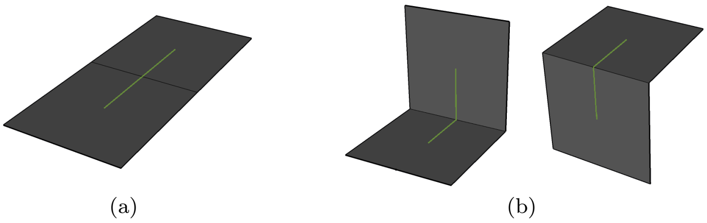


Figure B.5 (a) shows locus segments of type I, and (b) shows the two cases of type II locus segments.

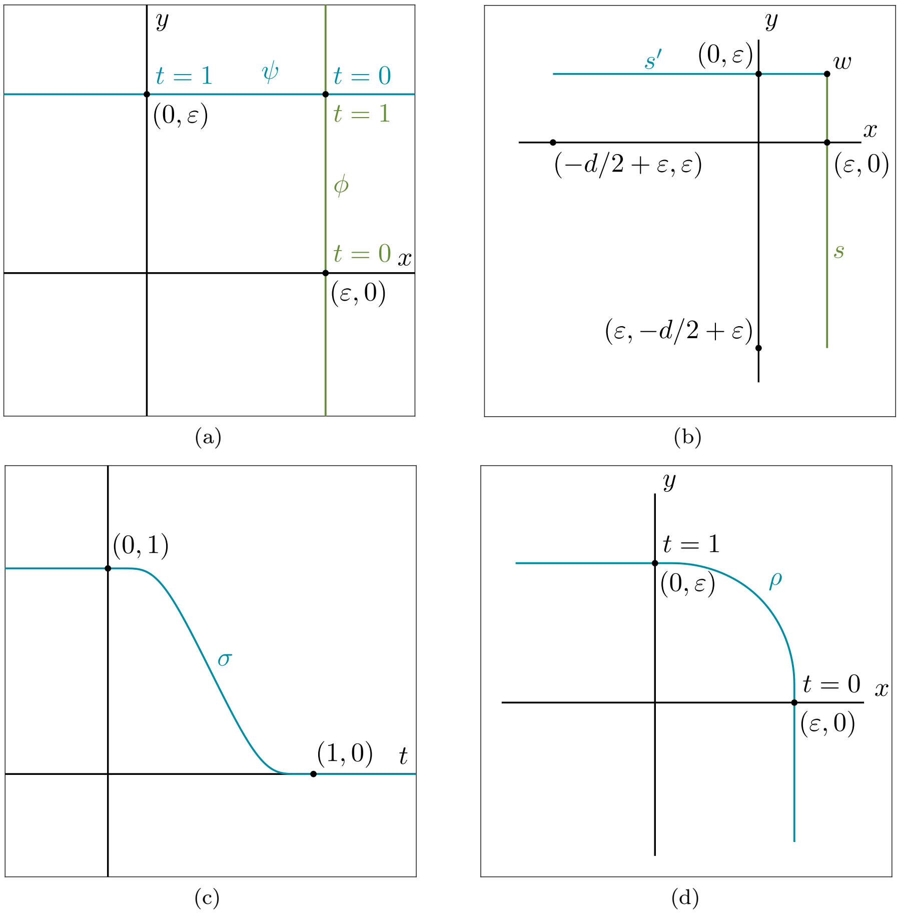


Figure B.6
Plots of the lines and functions used to define the smoothing process. (a) shows the lines ϕ and ψ defined by the parametric equations in (B.1); (b) shows the locus segments s and s' represented as segments of the lines ϕ and ψ respectively; (c) shows the function $\sigma\colon \mathbb{R} \to \mathbb{R}$ defined by (B.2), and (d) shows the curve ρ defined by the parametric equations in (B.3). All plots are shown in the plane $z = 0$.

can be represented by $s = \phi(t)$ for $-d/(2\varepsilon) \le t \le 1$ and $s' = \psi(t)$ for $0 \le t \le (2\varepsilon+d)/(2\varepsilon)$ when $w = (\varepsilon, \varepsilon, 0)$ and $z = 0$ is the plane spanned by s and s'; see Figure B.6(b). We now define a cutoff function $\sigma\colon \mathbb{R} \to \mathbb{R}$ given by

$$\sigma(t) = \frac{f(1-t)}{f(1-t) + f(t)}, \tag{B.2}$$

which is illustrated in Figure B.6(c), and where $f\colon \mathbb{R} \to \mathbb{R}$ is the function

$$f(t) = \begin{cases} e^{-1/t} & \text{for } t > 0; \\ 0 & \text{for } t \le 0. \end{cases}$$

Finally, let ρ be the smooth curve defined by the parametric equations

$$\begin{aligned} x_\rho(t) &= x_\phi(t)\sigma(t) + x_\psi(t)(1-\sigma(t)) = \varepsilon\sigma(t) + (\varepsilon - \varepsilon t)(1-\sigma(t)), \\ y_\rho(t) &= y_\phi(t)\sigma(t) + y_\psi(t)(1-\sigma(t)) = \varepsilon t\sigma(t) + \varepsilon(1-\sigma(t)), \end{aligned} \tag{B.3}$$

for $t \in \mathbb{R}$, see Figure B.6(d). The curve ρ is a smooth transition from s to s' for $-d/(2\varepsilon) \le t \le (2\varepsilon + d)/(2\varepsilon)$. Using such smoothings, we obtain a smoothing of all loci of W which is consistent with our initial idea for a transverse vector field as discussed in Subsection 2.5.2. An example of the smoothed loci is given in Figure B.7.

We now extend the smoothing of the loci to a smoothing of the annuli of ∂W. For each non-critical boundary vertex v of W, we take the smoothing defined for the loci and use it on the centre locus $\partial\mathcal{C}_v$ as well. We then use linear interpolation from the smoothed locus to the smoothed centre disc in order to obtain a smoothing of $\mathcal{A}_v$ which is exactly bounded by $\mathcal{L}_v \cup \partial\mathcal{C}_v$. This yields a smoothing of all annuli $\mathcal{A}$ which were defined for (n), $2 \le n \le 7$. We wish to smooth all discs of W as well, so we still need a smoothing procedure for the centre discs for (n) as well as for the discs of $\langle m\rangle$, $1 \le m \le 5$. Let us attend to the former problem first.

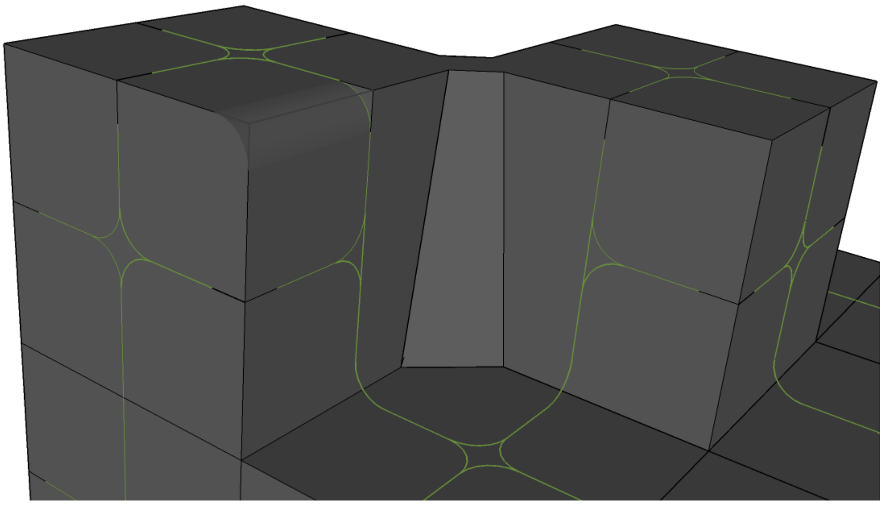
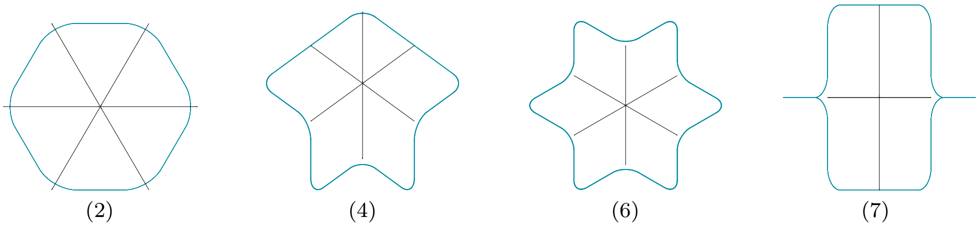


Figure B.7 Example of the smoothing of loci (green) of a wedged voxel reconstruction. The frontmost vertex and corresponding edge have been rendered slightly transparent in order to show how the smoothed locus of the edge is moved inside the reconstruction. This is the case for all Type II locus segments which have an edge-adjacent locus segment on the same voxel. The smoothed locus of Type II locus segments with edge-adjacent loci in adjacent voxels are moved outside of the wedged reconstruction.

Table B.2 Coordinate pairs specifying for configurations (2), (5), (6), and (7) a line used to define a smoothing of the configuration.

Configuration	Coordinate pairs
(2)	$(-d/2, -d/2, -d/2)$, $(d/2, d/2, d/2)$
(4)	$(0, 0, -d/2)$, $(0, 0, d/2)$
(6)	$(-d/2, -d/2, -d/2)$, $(d/2, d/2, d/2)$
(7)	$(-d/2, 0, -d/2)$, $(d/2, 0, d/2)$



(2) (4) (6) (7)

Figure B.8 Projections of the boundary $\partial\mathcal{C}_{v_n}$ of the centre disc of a configuration (n) onto the plane N_n. The black lines represent the edges of the wedged voxel reconstruction.

B.2.3 Smoothing of the centre discs

For configurations (n), $2 \leq n \leq 7$, the previous section describes how to smooth the annuli. Configuration (1) can be omitted from all smoothing considerations since it does not contain any black voxels. Smoothing the centre discs requires a little more work since we are still interested in a smoothing which allows for the construction of a vector field transverse to the smooth boundary. It should be relatively clear how the smoothing of the annuli supports the construction of the transverse vector field illustrated by Figure 2.10(c). We would like to continue to use interpolation to define our smoothing and, eventually, to define the vector field ξ.

Denote by v_n the centre of any configuration (n). In the case of (3), consider the smoothing of the type II locus segments. This moves the smoothed locus inside W. We can parallel shift this procedure along the boundary edge of W from one side of the locus to the other to smooth the entire edge of (3) contained within $\mathcal{D}_{v_3}$, thus yielding a smoothing of $\mathcal{D}_{v_3}$. We denote by $\lceil 3 \rfloor$ the smoothed version of (3).

Naturally, $\mathcal{D}_{v_1}$ and $\mathcal{D}_{v_5}$, and thereby $\mathcal{C}_{v_1}$ and $\mathcal{C}_{v_5}$, are already smooth. To be consistent with the notation, we let $\lceil 1 \rfloor$ and $\lceil 5 \rfloor$ denote the smoothed versions of (1) and (5) respectively, which are thus identical to the non-smoothed configurations.

Consider now the remaining configurations (2), (4), (6), and (7). If the vector field is to be symmetrical for a given configuration, there is a natural choice of line segment through v_n for each n along which the vector field should lie. Each such line segment can be determined by specification of its two endpoints which are given in Table B.2, where the coordinates are given in the coordinate system of the configuration, see Table B.1. We now proceed to smooth the relevant centre discs.

Denote by N_n the plane through v_n with normal the line specified by Table B.2 for any choice of orientation of the normal. We now project the smoothed centre locus $\partial\mathcal{C}_{v_n}$ onto N_n. The projected boundaries thus obtained are shown in Figure B.8. We can think of the smoothed centre locus $\partial\mathcal{C}_{v_n}$ as a graph over its projection to N_n. In order for us to use

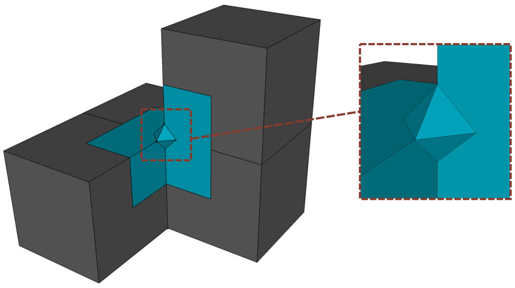


Figure B.9 Wedging of (7), where the disc, which is now a disc of a wedge, is indicated (blue).

any argument on preservation of transversality when smoothly interpolating lines through $\partial\mathcal{C}_{v_n}$ to $\mathcal{D}_{v_n}$ as we will do in the next section, the projected image cannot have edges of the configuration with coinciding projections. This, however, is the case for (7): we should be able to see the projection of six edges meeting at v_7, but we see only four. Therefore, we will need another approach to smooth this configuration. For (2), (4), and (6), we now smoothly interpolate from the smoothed boundary $\partial\mathcal{C}_{v_n}$ to N_n inside $\mathcal{C}_{v_n}$. This results in a smoothing of the vertex of the three configurations and hence a smoothing of the entire disc $\mathcal{D}_n$. Denote by $\lceil n\rfloor$, $n = 2, 4, 6$, the smoothed configurations thus obtained. We have still to define a smoothing of (7) and ⟨1⟩–⟨5⟩.

B.2.4 Smoothing of wedges

Due to the situation illustrated by Figure B.8(7), if we want to use the approach of interpolating with N_7 in $\mathcal{C}_{v_7}$ for (7), we need to introduce a wedge at v_7 much as was done for ⟨1⟩–⟨5⟩ in Definition B.1 to move those planes that collapse under projection onto N_7.

Definition B.4

Definition. *For $\varepsilon > 0$ given by Definition B.1, set*

$$W_6 = \operatorname{conv}\{(\varepsilon/2, 0, 0), (-\varepsilon/2, 0, 0), (0, \varepsilon/2, 0), (0, -\varepsilon/2, 0), (0, 0, \varepsilon/2), (0, 0, -\varepsilon/2)\}.$$

We define ⟨6⟩ as the configuration obtained by colouring the wedge W_4 in (7) black.

The wedged version ⟨6⟩ of (7) is illustrated in Figure B.9.

We now wish to define a common procedure for smoothing the six wedged configurations ⟨1⟩–⟨6⟩. This is done by using the same procedure as the one used for loci in Subsection B.2.2: whenever we have a pair of 2-cells of the boundary of a wedged reconstruction intersecting in a 0- or 1-cell belonging to a wedge, we smoothly interpolate from one to the other in an 'ε neighbourhood' of the edge as was done in the aforementioned procedure illustrated by Figure B.6. We thus obtain smoothed versions of the wedged configurations. The designations $\lceil 1\rfloor$–$\lceil 6\rfloor$ are already in use for the smoothed version of (1)–(6), but since ⟨6⟩ corresponds to a wedged version of (7) and ⟨5⟩ to (10), we shall denote their smoothed versions by respectively $\lceil 7\rfloor$ and $\lceil 10\rfloor$. The wedged configurations ⟨1⟩ and ⟨3⟩ are based on (8)*, so we name the smoothed version of ⟨1⟩ $\lceil 8a\rfloor$ and that of ⟨3⟩ $\lceil 8b\rfloor$. Similarly, $\lceil 9a\rfloor$ denotes the smoothed version of ⟨2⟩ and $\lceil 9b\rfloor$ that of ⟨4⟩. When referring to smoothings $\lceil 1\rfloor$–$\lceil 10\rfloor$, it is understood that $\lceil 8\rfloor$ means $\lceil 8a\rfloor$ and $\lceil 8b\rfloor$; similarly for $\lceil 9\rfloor$ and $\lceil 9a\rfloor$, $\lceil 9b\rfloor$.

Definition B.5

Definition. *We define smoothings $\lceil 1\rfloor$–$\lceil 10\rfloor$ of (1)–(6) and ⟨1⟩–⟨6⟩ by the smoothing procedure described in Section B.2.*

Together, the smoothed configurations constitute the smoothed wedged voxel reconstruction Z or, for short, the smoothed reconstruction, with respect to the lattice dL and based on the wedged reconstruction W. It is easily verified that the configurations $\lceil n\rfloor$ of Z, $1 \le n \le 10$, are compatible: the configurations of W are compatible, and the smoothed loci are compatible since the loci are compatible. For each configuration of W, the smoothing inside the locus of the corresponding vertex or wedge does not affect adjacent configurations.

The smoothing procedure renders Z homeomorphic to W, and the boundary of Z is smooth. It should be immediate that the smoothing procedure does not affect the number of connected components of the reconstruction: smoothing is carried out in the small

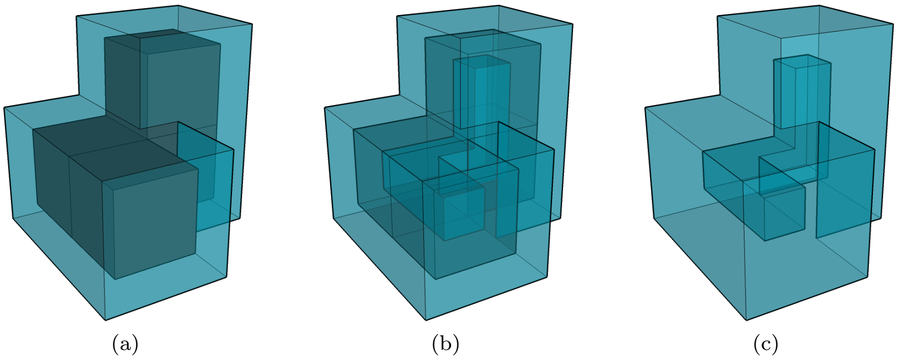


Figure B.10 (a) Cubic $d/2$-dilation (blue) of the boundary of configuration (6) of V. (b) has added transparency of the black voxels of (6) in order to better visualise the boundary of the cubic dilation inside the configuration; in (c), the black voxels have been removed altogether.

ε neighbourhoods of the edges of W and does not cause components of W to become connected; nor does it disconnect components.

B.3 Construction of the vector field

The construction of a vector field ξ satisfying our requirements as per Theorem 2.20 was commenced in Subsection 2.5.1. We have defined the vector field ξ_A which will be merged with ξ_Z (to be defined below) in order to obtain a vector field ξ with the desired properties. We present now the neighbourhood of ∂Z on which to define a direction field ϕ_Z for ξ_Z and subsequently define the direction field itself.

B.3.1 Cubic dilation

Recall that, at the beginning of Subsection B.2.1, for $s \in \mathbb{R}$ and $k \in \mathbb{R}^3$, we introduced the cube $K_s(k) \subset \mathbb{R}^3$ with side length s, centre k, and edges parallel to the axis directions of dL.

Definition B.6

Definition. *For $s \in \mathbb{R}$ and X any subset of $\mathbb{R}^3$, the* ***cubic s-dilation*** *of ∂X with respect to dL is the set*

$$K_s(X, dL) = K_s(X) := \bigcup_{x \in \partial X} K_s(x).$$

Figure B.10 shows an example of a cubic dilation. The idea now is to define a direction field ϕ_Z on the cubic $d/2$-dilation of ∂V which we claim is also a neighbourhood of ∂Z.

Lemma B.7

Lemma. *The boundary ∂Z is a subset of the interior of $K_d(V)$.*

Proof. Let $\varepsilon > 0$ be the one given by Definition B.1. From the definition of the convex hulls that constitute the wedges of Definitions B.1 and B.4, we see that the distance from any point contained in a wedge of type $\langle 1\rangle$–$\langle 4\rangle$ or $\langle 6\rangle$ to the boundary of the underlying set V is strictly smaller than ε. This means that all points inside these wedges are contained in $K_d(V)$ as long as we choose $\varepsilon < d/2$. Similarly, the points removed from V when inserting the white wedges of $\langle 5\rangle$ are closer to ∂V than ε. This means that the manipulation of the boundary of ∂V, which is only done locally for configurations (7)–(10), does not cause the resulting boundary to intersect $K_d(V)$. Similarly, we see that the smoothing procedure used to define ∂Z is a manipulation of the wedged object inside an ε neighbourhood of the boundary; see Figure B.6. Hence ∂Z is contained inside the interior of $K_d(V)$. □

B.3.2 A direction field on the cubic dilation

We will now define a direction field ϕ_Z inside $K_d(V)$ such that the line segments constituting ϕ_Z are transverse to ∂Z. We base our construction on the idea initially used for defining the smoothing process described in Section B.2. We need coordinate systems of the configurations of ∂Z, and each smoothed configuration $\lceil n \rfloor$ naturally inherits its coordinate system from the underlying configuration (n) for $2 \leq n \leq 10$ if we let the coordinate system of the 3-by-2-by-2 configurations $\lceil 8a \rfloor$ and $\lceil 8b \rfloor$ be determined by (8)* and $\lceil 9a \rfloor$ and $\lceil 9b \rfloor$ be determined by (9)*; see Table B.1. Furthermore, $\lceil 7 \rfloor$ presents a special case which must be treated separately. The direction field here is not as straightforward, and we need the wedge introduced in Definition B.4 to avoid problems with transversality later on.

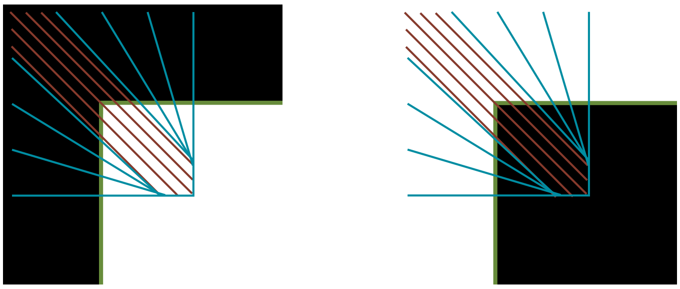


Figure B.11
The line segments (blue and red) which we interpolate to obtain Λ_{II} for the two cases of type II locus segments (green) of ∂V. The picture is shown in the plane containing the line segments.

We will define line segments for the direction field for each configuration of ∂Z in turn, but first, we define ϕ_Z on the smoothed loci of ∂Z.

Recall that the smoothing process does not affect locus segments of ∂V of type I, so these same locus segments constitute part of the smoothed loci of ∂Z. Type II locus segments are smoothed, and in this smoothed form, they make up the remaining parts of the smoothed loci of ∂Z. For any locus segment s of type I, consider all line segments of length d with midpoint on s that are normal to the (smoothed) boundary face of ∂Z containing s. Denote the collection of such line segments by Λ_{I}. Any locus segment s of ∂V of type II meets another type II locus segment s' of ∂V in a boundary edge of ∂V in some point $v \in \partial V$. Consider now a coordinate system with axes parallel to s, s', and the edge in which they meet and such that v has coordinates $(d/2, d/2, 0)$, the other endpoint of s has coordinates $(0, d/2, 0)$, and that of s' coordinates $(d/2, 0, 0)$. The locus segments s and s' are smoothed in an ε neighbourhood of p, so inside this region, we take Λ_{II} to be the collection of line segments parallel to the line segment from $(0, 0, 0)$ to $(d, d, 0)$ with endpoints on $\partial K_d(V)$ and midpoint on the ε-region of s and s' near v. On the endpoints of s and s', we take line segments of length d with midpoint at the endpoint, such that the line segments are normal to the (smoothed) boundary face, and add these to Λ_{II}. To obtain the final line segments of Λ_{II}, we now smoothly interpolate from the line segments near v to those at the endpoints of s and s' and take segments with endpoints on $\partial K_d(V)$ and midpoints on s or s' parallel to those obtained by interpolation. The line segments of Λ_{II} are illustrated in Figure B.11.

In a similar fashion, we now define collections $\Lambda_{\lceil n \rfloor}$ of line segments for each configuration $\lceil n \rfloor$ of Z, $1 \leq n \leq 10$. In the coordinate system of each configuration, consider the origin. Except for configuration $\lceil 1 \rfloor$, the origin coincides with a boundary vertex v_n of the underlying voxel reconstruction V. We will refer to the smoothed version of the associated locus $\mathcal{L}_{v_n}$ as **the (smooth) locus of $\lceil n \rfloor$**. The loci of all $\lceil n \rfloor$ are called the loci of Z. The collections Λ_{I} and Λ_{II} thus define line segments through all points of the loci of Z. We now wish to define line segments of the direction field ϕ_Z through the region bounded by the locus of each $\lceil n \rfloor$, the **(smooth) disc of $\lceil n \rfloor$**. Line segments for each configuration have been specified by coordinate pairs determining their endpoints in Table B.3. The complement of $\lceil 10 \rfloor$ is not treated as a special case here since, analogously to $\lceil 2 \rfloor$–$\lceil 7 \rfloor$, we will use the same line segments when the black and white colours of the configuration (and the wedges) are interchanged to obtain the complementary configuration.

For configuration $\lceil 5 \rfloor$, we have defined line segments through all points of the smoothed disc, and for $\lceil 3 \rfloor$, we have specified line segments through the edge from $(-d/2, 0, 0)$ to $(d/2, 0, 0)$. For $\lceil 8a \rfloor$, $\lceil 8b \rfloor$, $\lceil 9a \rfloor$, and $\lceil 9b \rfloor$, we have specified endpoints of three line segments, P_1, P_2, and P_3, and for the remaining configurations of Z, endpoints of a single line segment are defined. The definition of $\mathcal{L}_{\lceil 5 \rfloor}$ is thus complete. For $\lceil 3 \rfloor$, for $\varepsilon > 0$ given by the smoothing process in Subsection B.2.2 and for $-d/2 \leq \alpha \leq d/2$, we take all line segments parallel to the one defined in Table B.3 for this fixed α with midpoint on a line segment with endpoints $(0, 0, 0)$ and $(\beta, 0, 0)$ or $(0, 0, -\beta)$ for $0 < \beta \leq 2\varepsilon$ and with endpoints on $\partial K_d(V)$. We now use linear interpolation from these line segments to the ones defined on the locus of $\lceil 3 \rfloor$ in the plane $x = \alpha$ and take the line segments parallel to these and with endpoints on $\partial K_d(V)$. We define $\mathcal{L}_{\lceil 3 \rfloor}$ as the collection of all such line segments for all choices of α.

For configuration $\lceil 10 \rfloor$, we define the line segments of $\mathcal{L}_{\lceil 10 \rfloor}$ as all radial lines of the ball $B_{d/2}(x)$ with $x = (d/2, -d/2, -d/2)$ inside the cube with side length d centred at $(0, 0, 0)$.

For the remaining configurations, inside a 2ε neighbourhood of the origin containing the smoothed part of the vertex, we take as line segments for $\Lambda_{\lceil n \rfloor}$ the line segments parallel to those defined through the origin in Table B.3 with endpoints on $\partial K_d(V)$ and midpoint on the smoothed boundary. We then interpolate from these line segments to any other

Table B.3 Coordinate pairs specifying for configurations $\lceil 1\rfloor$–$\lceil 10\rfloor$ line segments used to define the direction field ϕ_Z.

Configuration	Coordinate pairs
$\lceil 2\rfloor$	$(-d/2,-d/2,-d/2)$, $(d/2,d/2,d/2)$
$\lceil 3\rfloor$	$(d/2,\alpha,d/2)$, $(-d/2,\alpha,-d/2)$ for $-d/2\leq\alpha\leq d/2$
$\lceil 4\rfloor$	$(0,0,-d/2)$, $(0,0,d/2)$
$\lceil 5\rfloor$	$(\alpha,\beta,d/2)$, $(\alpha,\beta,-d/2)$ for $-d/2\leq\alpha\leq d/2$, $-d/2\leq\beta\leq d/2$
$\lceil 6\rfloor$	$(-d/2,-d/2,-d/2)$, $(d/2,d/2,d/2)$
$\lceil 7\rfloor$	$(-d/2,0,-d/2)$, $(d/2,0,d/2)$
$\lceil 8a\rfloor$	P_1: $(d/2,d/2,d/2)$, $(-d/2,-d/2,-d/2)$
	P_2: $(0,0,3d/2)$, $(0,0,d/2)$
	P_3: $(-d/2,-d/2,d/2)$, $(d/2,d/2,-d/2)$
$\lceil 8b\rfloor$	P_1: $(d/2,-d/2,3d/2)$, $(-d/2,d/2,d/2)$
	P_2: $(d/2,d/2,0)$, $(-d/2,-d/2,0)$
	P_3: $(-d/2,-d/2,d/2)$, $(d/2,d/2,-d/2)$
$\lceil 9a\rfloor$	P_1: $(d/2,-d/2,3d/2)$, $(-d/2,d/2,d/2)$
	P_2: $(d/2,d/2,d/2)$, $(-d/2,-d/2,-d/2)$
	P_3: $(-d/2,-d/2,d/2)$, $(d/2,d/2,-d/2)$
$\lceil 9b\rfloor$	P_1: $(0,0,3d/2)$, $(0,0,d/2)$
	P_2: $(d/2,d/2,0)$, $(-d/2,-d/2,0)$
	P_3: $(-d/2,-d/2,d/2)$, $(d/2,d/2,-d/2)$
$\lceil 10\rfloor$	$(0,0,0)$, $(d/2,-d/2,-d/2)$

line segments defined by Table B.3 and to the line segments on the corresponding locus. In case of multiple line segments, we need to make sure that interpolation creates line segments that intersect the boundary ∂Z only once. We let the length of the line segments be determined by the cubic dilation $K_d(V)$: the line segments should have endpoints on the boundary of $K_d(V)$, which is already satisfied for the endpoints defined in Table B.3. This defines the line segments of $\Lambda_{\lceil n\rfloor}$.

As indicated in Subsection B.2.3, $\lceil 7\rfloor$ represents a special case where more caution must be exercised. We have introduced the additional wedges in Definition B.4 to make sure that the projection of the smoothed centre disc is 1:1 for sufficiently small ε and hence to avoid the situation in Figure B.8(7).

The above yields a collection $\Lambda_{\lceil n\rfloor}$ of line segments which, together with the line segments Λ_{I} and Λ_{II} through the loci, can be used to define a direction field on the cubic dilation $K_d(V)$. For the wedged configurations $\langle 1\rangle$–$\langle 5\rangle$, the line segments are defined such that if they enter a black wedge from a white voxel, they continue through the wedge and into a black voxel.

Definition B.8

Definition. *We define the direction field ϕ_Z on $K_d(V)$ as the collection of line segments from Λ_{I}, Λ_{II}, and $\Lambda_{\lceil n\rfloor}$ defined in Subsection B.3.2.*

This rather technical introduction of ϕ_Z now allows for the definition of a vector field ξ_Z on a neighbourhood $Z_\varepsilon\subset K_d(V)$; see Definition 2.31. Transversality of ξ_Z to ∂Z then follows from the same property of the direction field.

Proposition B.9

Proposition. *The line segments of ϕ_Z are transverse to ∂Z.*

Proof. The locus segments of type I of ∂V are preserved during smoothing and can thus be thought of as type I locus segments of ∂Z as well. Any point on such a locus segment intersects a line segment from Λ_{I}, and these line segments are normal, and hence transverse, to ∂Z.

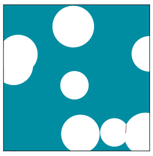
Locus segments of type II of ∂V are modified in one end as illustrated by Figure B.6(d) during smoothing, and these smoothed line segments are now the type II locus segments of ∂Z. Before smoothing, points on the locus segment intersect the line segments defined for Λ_{II} transversely; see Figure B.11. The smoothing process is now defined in an ε-region of the point p in which the two type II locus segments meet. Using pictures similar to those in Figure B.8, we can project such a neighbourhood onto one of the two planes with normal the line parallel to the line segment l of Λ_{II} through p. The boundary of the ε neighbourhood can be interpreted as the graph over its projection, so transversality is guaranteed since it is obtained in the projected picture: all line segments of Λ_{II} are parallel to l inside a 2ε neighbourhood, and these are transverse to the plane and thus the projection of the boundary of the ε neighbourhood.

With these conventions on locus segments of the boundary of the smoothed reconstruction, we notice that every point on a locus segment of ∂Z is intersected by a line segment from either Λ_{I} or Λ_{II} which is transverse to ∂Z. The line segments defined in Table B.3 are defined such that they are transverse to ∂Z as well and intersect ∂V exactly at the vertices where the normals of the boundary change direction. Using linear interpolation of the line segments to obtain ϕ_Z, and making similar observations on the projected ε neighbourhoods near vertices to planes with normals lines parallel to the lines segments of $\Lambda_{\lceil n \rfloor}$, ensures that we do not get line segments which are parallel to the boundary faces of Z at the points where they intersect ∂Z; this is assured by the wedging of those configurations containing critical cells and the special case (7). □

In this thesis, we have chosen to first introduce a smoothing Z of the wedged reconstruction and then define a vector field in a neighbourhood of the smooth manifold ∂Z, but the order of the two procedures can be changed. Indeed, the smoothing process makes such small corrections to the boundary that it has no influence on how we define the vector field, and transversality properties are not altered by the bump functions used to obtain the smoothing.

C

TEST SETS: TENSORS AND DIGITISATIONS



The simulations performed with Minktensor2D3R and Minktensor2D2R in Section 3.4 are carried out for three test sets: a disc, an annulus, and the complement of a Boolean model. The estimators computed by the algorithms are compared to the actual Minkowski tensors of the three sets. In this appendix, we show how these true tensors are calculated. Furthermore, we document how the test sets are digitised for use in the algorithms.

C.1 CALCULATION OF TENSORS

Recall from Section 3.1 that for a compact set $A \subset \mathbb{R}^n$, the rth volume tensor of A is

$$\Phi_n^{r,0}(A) = \frac{1}{r!} \int_A x^r \, \mathrm{d}x \tag{3.2}$$

with $r \in \mathbb{N}_0$, and if in addition A has positive reach, the surface tensors of A are

$$\Phi_k^{r,s}(A) = \frac{1}{r!s!} \frac{\omega_{n-k}}{\omega_{n-k+s}} \int_{\mathbb{R}^n \times S^{n-1}} x^r u^s \Lambda_k(A; \mathrm{d}(x,u)) \tag{3.3}$$

for $k \in \{0, \dots, n-1\}$ and $r, s \in \mathbb{N}_0$.

For the calculations below, we will refer to a number of equations from the book [32] by Schneider. We shall use the notation (SX.Y), with a prefixed S in the equation number, to refer to formula (X.Y) from this source.

As a first observation, a combination of formulae in [32] yields:

Lemma C.1

Lemma. *For $A \subset \mathbb{R}^n$ a compact set with positive reach, the Minkowski surface tensors $\Phi_k^{0,1}(A) = 0$ are trivial for $k \in \{0, \dots, n-1\}$.*

Proof. For $k \in \{0, \dots, n-1\}$ and η a Borel subset of S^{n-1}, the formula

$$S_k(A;\eta) = \Theta_k(A; \mathbb{R}^n \times \eta) \tag{S4.11}$$

defines a measure $S_k(A; \cdot\,)$ on sets of normal vectors from the support measure $\Theta_k(A; \cdot\,)$ of order k on A [32, Section 4.2]. We will not go into details with these measures here as we shall simply exploit their properties. With $\rho > 0$, we have

$$S_{n-1}(A^\rho; \cdot\,) = \sum_{k=0}^{n-1} \rho^{n-1-k} \binom{n-1}{k} S_k(A; \cdot\,), \tag{S4.27}$$

where A^ρ is the ρ-parallel set of A, and where S is the mixed area measure defined in [32, Theorem 5.1.7], which, by (S5.18) with $m = 2$, $\lambda_1 = 1$, $\lambda_2 = \rho$, $K_1 = A$, and $K_2 = B_1^n$, satisfies

$$\begin{aligned} S_{n-1}(A^\rho; \cdot\,) = S_{n-1}(A + \rho B_1^n; \cdot\,) &= \sum_{i_1,\dots,i_{n-1}=1}^{2} \lambda_{i_1} \dots \lambda_{i_{n-1}} S(K_{i_1}, \dots, K_{i_{n-1}}; \cdot\,) \\ &= \sum_{k=0}^{n-1} \rho^{n-1-k} \binom{n-1}{k} S(\underbrace{A, \dots, A}_{k}; \underbrace{B_1^n, \dots, B_1^n}_{n-1-k}; \cdot\,). \end{aligned}$$

We now compare this equation to (S4.27) above. Both expressions are polynomials in ρ. Since they are both equal to $S_{n-1}(A^\rho; \cdot\,)$, they must have they same coefficients, and we conclude that

$$S(\underbrace{A, \dots, A}_{k}; \underbrace{B_1^n, \dots, B_1^n}_{n-1-k}; \cdot\,) = S_k(A; \cdot\,).$$

For $k \in \{0, \dots, n-1\}$, (S5.30) implies that

$$\int_{S^{n-1}} uS(\underbrace{A, \dots, A}_{k}; \underbrace{B_1^n, \dots, B_1^n}_{n-1-k}; \mathrm{d}u) = 0,$$

so by the formula for the Minkowski surface tensors,

$$\begin{aligned}\Phi_k^{0,1}(A) &= \frac{\omega_{n-k}}{\omega_{n-k+1}} \int_{\mathbb{R}\times S^{n-1}} x^r u^s \Lambda_k(A; \mathrm{d}(x,u)) \\ &= \frac{\binom{n}{k}}{n\kappa_{n-k}} \frac{\omega_{n-k}}{\omega_{n-k+1}} \int_{\mathbb{R}\times S^{n-1}} x^r u^s S_k(A; \mathrm{d}(x,u)) = 0,\end{aligned}$$

where the last equality follows from (S4.11) and (S4.18), which give the relations of $\Lambda_k(A; \cdot)$ and $S_k(A; \cdot\,)$ to $\Theta_k(A; \cdot\,)$. □

C.1.1 The disc

Consider the unit disc $B_1 = B_1(0)$ in $\mathbb{R}^2$. Due to symmetry, many of the non-trivial Minkowski tensors of this set equal zero. Thus, to obtain a slightly more complicated model with more non-zero tensors, we consider a translation of B_1 and use as test set $A = B_1 + (1,1)^T$.

To compute the surface tensors of A, we make some simplifications of the integral in (3.3). First, for $k \in \{0, \dots, n-1\}$, the generalised curvature measure $\Lambda_k(K; \cdot\,)$ for a compact set $K \subset \mathbb{R}^n$ with positive reach can be re-written in terms of the support measure $\Theta_k(K; \cdot\,)$ mentioned above:

(S4.18)
$$\Lambda_k(K; \cdot\,) = \frac{\binom{n}{k}}{n\kappa_{n-k}} \Theta_k(K; \cdot\,).$$

In addition, we can use a Steiner formula to obtain

$$\int_{\mathbb{R}^n \times S^{n-1}} x^r u^s \Lambda_k(K+z; \mathrm{d}(x,u)) = \int_{\mathbb{R}^n \times S^{n-1}} (x+z)^r u^s \Lambda_k(K; \mathrm{d}(x,u))$$

for $z \in \mathbb{R}^n$ any translation and $r, s \in \mathbb{N}_0$.

Using the above, we now compute the Minkowski tensors of A in $\mathbb{R}^2$. For $k = 0$, we have

$$\begin{aligned}\Phi_0^{r,s}(A) &= \frac{1}{r!s!} \frac{\omega_2}{\omega_{2+s}} \int_{\mathbb{R}^2 \times S^1} \big(x + (1,1)^T\big)^r u^s \Lambda_0(B_1; \mathrm{d}(x,u)) \\ &= \frac{1}{r!s!} \frac{2\pi}{\omega_{2+s}} \frac{\binom{2}{0}}{2\kappa_2} \int_{\mathbb{R}^2 \times S^1} \big(x + (1,1)^T\big)^r u^s \Theta_0(B_1; \mathrm{d}(x,u)) \\ &= \frac{1}{r!s!} \frac{\pi}{\omega_{2+s}} \frac{1}{\pi} \int_{\mathbb{R}^2} \big(u + (1,1)^T\big)^r u^s \mathcal{H}^1(\mathrm{d}u),\end{aligned}$$

where the last equality follows from [32, Lemma 4.2.2] and (S4.29). This implies that we can compute the surface tensors for $k = 0$ by

$$\Phi_0^{r,s}(A) = \frac{1}{r!s!} \frac{1}{\omega_{2+s}} \int_{S^1} \big(u + (1,1)^T\big)^r u^s \,\mathrm{d}u.$$

This expression can be computed explicitly using

(3.12)
$$\kappa_0 = 1, \qquad \kappa_1 = 2, \qquad \kappa_2 = \pi, \qquad \kappa_3 = 4\pi/3, \qquad \kappa_4 = \pi^2/2$$

together with $\omega_0 = \kappa_0$ and $\kappa_k = k\omega_k$ for $k \in \mathbb{N}$.

We can now calculate the Minkowski surface tensors of A of rank at most two, i.e. for $r, s \in \mathbb{N}_0$ with $r + s \leq 2$. For $k = 0$,

$$\begin{aligned}\Phi_0^{0,0}(A) &= \frac{1}{2\pi} \int_{S^1} 1 \,\mathrm{d}u = 1; \\ \Phi_0^{1,0}(A) &= \frac{1}{2\pi} \int_{S^1} \begin{pmatrix} u_1 + 1 \\ u_2 + 1 \end{pmatrix} \mathrm{d}u = \begin{pmatrix} 1 \\ 1 \end{pmatrix};\end{aligned}$$

$$\begin{aligned}
\Phi_0^{2,0}(A) &= \frac{1}{4\pi}\int_{S^1}\begin{pmatrix}(u_1+1)^2 & (u_1+1)(u_2+1)\\ (u_1+1)(u_2+1) & (u_2+1)^2\end{pmatrix}\mathrm{d}u\\
&= \frac{1}{4\pi}\int_0^{2\pi}\begin{pmatrix}(\cos(\theta)+1)^2 & (\cos(\theta)+1)(\sin(\theta)+1)\\ (\cos(\theta)+1)(\sin(\theta)+1) & (\cos(\theta)+1)^2\end{pmatrix}\mathrm{d}\theta = \begin{pmatrix}3/4 & 1/2\\ 1/2 & 3/4\end{pmatrix};\\
\Phi_0^{1,1}(A) &= \frac{1}{4\pi}\int_{S^1}\begin{pmatrix}(u_1+1)u_1 & (u_1+1)u_2\\ (u_2+1)u_1 & (u_2+1)u_2\end{pmatrix}\mathrm{d}u\\
&= \frac{1}{4\pi}\int_0^{2\pi}\begin{pmatrix}(\cos(\theta)+1)\cos(\theta) & (\cos(\theta)+1)\sin(\theta)\\ (\sin(\theta)+1)\cos(\theta) & (\sin(\theta)+1)\sin(\theta)\end{pmatrix}\mathrm{d}\theta = \begin{pmatrix}1/4 & 0\\ 0 & 1/4\end{pmatrix};\\
\Phi_0^{0,2}(A) &= \frac{1}{4\pi^2}\int_{S^1}\begin{pmatrix}u_1^2 & u_1u_2\\ u_1u_2 & u_2^2\end{pmatrix}\mathrm{d}u\\
&= \frac{1}{4\pi^2}\int_0^{2\pi}\begin{pmatrix}\cos^2(\theta) & \cos(\theta)\sin(\theta)\\ \cos(\theta)\sin(\theta) & \sin^2(\theta)\end{pmatrix}\mathrm{d}\theta = \begin{pmatrix}1/(4\pi) & 0\\ 0 & 1/(4\pi)\end{pmatrix},
\end{aligned}$$

and $\Phi_0^{0,1}(A) = (0,0)^T$ by Lemma C.1.

For the surface tensors corresponding to $k = 1$, we have

$$\begin{aligned}
\Phi_1^{r,s}(A) &= \frac{1}{r!s!}\frac{\omega_1}{\omega_{1+s}}\frac{\binom{2}{1}}{2\kappa_1}\int_{\mathbb{R}^2\times S^1}\left(x+(1,1)^T\right)^r u^s\Theta_1(B_1;\,\mathrm{d}(x,u))\\
&= \frac{1}{r!s!}\frac{1}{\omega_{1+s}}\int_{\mathbb{R}^2}\left(u+(1,1)^T\right)^r u^s\mathcal{H}^1(\,\mathrm{d}u),
\end{aligned}$$

where the last equality uses [32, Lemma 4.2.2] and (S4.31). Comparing with the formula for $\Phi_0^{r,s}(A)$, we see that

$$\Phi_1^{r,s}(A) = \frac{\omega_{2+s}}{\omega_{1+s}}\Phi_0^{r,s}(A),$$

and so we get

$$\begin{aligned}
\Phi_1^{0,0}(A) &= \pi\Phi_0^{0,0}(A) = \pi;\\
\Phi_1^{1,0}(A) &= \pi\Phi_0^{1,0}(A) = \begin{pmatrix}\pi\\ \pi\end{pmatrix};\\
\Phi_1^{2,0}(A) &= \pi\Phi_0^{2,0}(A) = \begin{pmatrix}3\pi/4 & \pi/2\\ \pi/2 & 3\pi/4\end{pmatrix};\\
\Phi_1^{1,1}(A) &= 2\Phi_0^{1,1}(A) = \begin{pmatrix}1/2 & 0\\ 0 & 1/2\end{pmatrix};\\
\Phi_1^{0,2}(A) &= \frac{\pi}{2}\Phi_0^{0,2}(A) = \begin{pmatrix}1/8 & 0\\ 0 & 1/8\end{pmatrix},
\end{aligned}$$

and $\Phi_1^{0,1}(A) = (0,0)^T$ trivially.

Finally, for $k = 2$, the volume tensors of A are given by

$$\Phi_2^{r,0}(A) = \frac{1}{r!}\int_A x^r\,\mathrm{d}x = \frac{1}{r!}\int_{B_1}\left(x+(1,1)^T\right)^r\mathrm{d}x$$

for $r \in \{0,1,2\}$, so

$$\begin{aligned}
\Phi_2^{0,0}(A) &= \int_{B_1}1\,\mathrm{d}x = \pi;\\
\Phi_2^{1,0}(A) &= \int_{B_1}\begin{pmatrix}x_1+1\\ x_2+1\end{pmatrix}\mathrm{d}x = \int_0^{2\pi}\int_0^1\begin{pmatrix}r^2\cos(\theta)+r\\ r^2\sin(\theta)+r\end{pmatrix}\mathrm{d}r\,\mathrm{d}\theta = \begin{pmatrix}\pi\\ \pi\end{pmatrix};\\
\Phi_2^{2,0}(A) &= \frac{1}{2}\int_{B_1}\begin{pmatrix}(x_1+1)^2 & (x_1+1)(x_2+1)\\ (x_1+1)(x_2+1) & (x_2+1)^2\end{pmatrix}\mathrm{d}x\\
&= \frac{1}{2}\int_0^{2\pi}\int_0^1\begin{pmatrix}r(r\cos(\theta)+1)^2 & r(r\cos(\theta)+1)(r\sin(\theta)+1)\\ r(r\cos(\theta)+1)(r\sin(\theta)+1) & r(r\sin(\theta)+1)^2\end{pmatrix}\mathrm{d}r\,\mathrm{d}\theta\\
&= \begin{pmatrix}5\pi/8 & \pi/2\\ \pi/2 & 5\pi/8\end{pmatrix}.
\end{aligned}$$

C.1.2 The annulus

We use the annulus as an example of a non-convex set with positive reach. Consider the balls (discs) $B_1 = B_1(0)$ and $B_2 = B_2(0)$ in $\mathbb{R}^2$. We obtain a compact set by taking the closure $\mathrm{cl}(B_2 \setminus B_1)$ of the intersection of the 2-disc with the 1-disc. As we observed for $B_1 + (1,1)^T$, more non-zero Minkowski tensors arise when we translate the annulus, so we take as test set the annulus $A = \mathrm{cl}(B_2 \setminus B_1) + (1,1)^T$. Notice that all $x \in B_1 \setminus \{0\}$ belong to $\mathrm{Unp}(A)$, and B_2 has infinite reach, so the reach of A equals the radius of B_1; $\mathrm{Reach}(A) = 1$.

To compute the Minkowski surface tensors of A, we need to simplify (3.3). To this end, let η be a Borel subset of $\mathbb{R}^n \times S^{n-1}$. Let $K \subset \mathbb{R}^n$ be a compact set with positive reach, let $x \in \mathrm{Unp}(K)$, and denote by $u_K(x)$ the unit vector from $p_K(x)$ to x. Then for $0 < \rho < \mathrm{Reach}(K)$, [32, Theorem 4.2.1] and (S4.18) yield

$$\lambda\big(\{x \in \mathbb{R}^n \mid 0 < \delta_K(x) \leq \rho \text{ and } (p_K(x), u_K(x)) \in \eta\}\big) = \sum_{k=0}^{n-1} \rho^{n-k} \kappa_{n-k} \Lambda_k(K; \eta), \tag{C.1}$$

where λ denotes the Lebesgue measure. We now consider the annulus $K = \mathrm{cl}(B_2 \setminus B_1) = A - (1,1)^T \subset \mathbb{R}^2$ with $\mathrm{Reach}(K) = 1$. For $0 < \rho < 1$, the set $\{x \in \mathbb{R}^2 \mid 0 < \delta_K(x) < \rho\}$ coincides with the union $\mathrm{int}(B_1 \setminus B_{1-\rho}) \cup \mathrm{int}(B_{2+\rho} \setminus B_2)$. Hence for $f\colon \mathbb{R}^2 \times S^1 \to [0,\infty)$ a measurable function, we have

$$\begin{aligned}
&\int_{\{x \in \mathbb{R}^2 | 0 < \delta_K(x) \leq \rho\}} f(p_K(x), u_K(x))\, \mathrm{d}x \\
&\quad = \int_{B_1 \setminus B_{1-\rho}} f(p_K(x), u_K(x))\, \mathrm{d}x + \int_{B_{2+\rho} \setminus B_2} f(p_K(x), u_K(x))\, \mathrm{d}x \\
&\quad = \int_{1-\rho}^{1} \int_{S^1} f(p_K(rv), u_K(rv)) r\, \mathrm{d}v\, \mathrm{d}r + \int_2^{2+\rho} \int_{S^1} f(p_K(rv), u_K(rv)) r\, \mathrm{d}v\, \mathrm{d}r.
\end{aligned}$$

For the left-hand integral in this last equation, as $v \in S^1$ and since $(1-\rho)v < tv < v$, it follows that $p_K(tv) = v$ and $u_K(tv) = -v$. For the right-hand integral, $2v < tv < (2+\rho)v$, so $p_K(tv) = 2v$ and $u_K(tv) = v$. We thus obtain

$$\begin{aligned}
&\int_{\{x \in \mathbb{R}^2 | 0 < \delta_K(x) \leq \rho\}} f(p_K(x), u_K(x))\, \mathrm{d}x \\
&\quad = \int_{1-\rho}^{1} \int_{S^1} f(v, -v) r\, \mathrm{d}v\, \mathrm{d}r + \int_2^{2+\rho} \int_{S^1} f(2v, v) r\, \mathrm{d}v\, \mathrm{d}r.
\end{aligned}$$

Now using

$$\int_{1-\rho}^{1} r\, \mathrm{d}r = \rho - \frac{\rho^2}{2} \quad \text{and} \quad \int_2^{2+\rho} r\, \mathrm{d}r = 2\rho + \frac{\rho^2}{2},$$

we get

$$\begin{aligned}
&\int_{\{x \in \mathbb{R}^2 | 0 < \delta_K(x) \leq \rho\}} f(p_K(x), u_K(x))\, \mathrm{d}x \\
&\quad = \left(\rho - \frac{\rho^2}{2}\right) \int_{S^1} f(v, -v)\, \mathrm{d}v + \left(2\rho + \frac{\rho^2}{2}\right) \int_{S^1} f(2v, v)\, \mathrm{d}v \\
&\quad = \rho \int_{S^1} f(v, -v) + 2f(2v, v)\, \mathrm{d}v + \frac{\rho^2}{2} \int_{S^1} f(2v, v) - f(v, -v)\, \mathrm{d}v.
\end{aligned}$$

Comparing this last equation to (C.1) with $n = 2$, we see that

$$\int_{\mathbb{R}^2 \times S^1} f(x, u) \Lambda_0(K; \mathrm{d}(x, u)) = \frac{1}{2\kappa_2} \int_{S^1} f(2v, v) - f(v, -v)\, \mathrm{d}v$$

and

$$\int_{\mathbb{R}^2 \times S^1} f(x, u) \Lambda_1(K; \mathrm{d}(x, u)) = \frac{1}{\kappa_1} \int_{S^1} f(v, -v) + 2f(2v, v)\, \mathrm{d}v.$$

We now substitute $f(x,u) = x^r u^s$ and translate by $(1,1)^T$ to obtain

$$\int_{\mathbb{R}^2 \times S^1} x^r u^s \Lambda_0(A;\, \mathrm{d}(x,u)) = \frac{1}{2\pi} \int_{S^1} \big(2v + (1,1)^T\big)^r v^s - \big(v + (1,1)^T\big)(-v)^s \,\mathrm{d}v$$

and

$$\int_{\mathbb{R}^2 \times S^1} f(x,u) \Lambda_1(A;\, \mathrm{d}(x,u)) = \frac{1}{2} \int_{S^1} \big(v + (1,1)^T\big)^r (-v)^s + 2\big(2v + (1,1)^T\big)^r v^s \,\mathrm{d}v,$$

so we conclude that

$$\Phi_0^{r,s}(A) = \frac{1}{r!s!} \frac{1}{\omega_{2+s}} \int_{S^1} \big(2v + (1,1)^T\big)^r v^s - \big(v + (1,1)^T\big)(-v)^s \,\mathrm{d}v$$

and

$$\Phi_1^{r,s}(A) = \frac{1}{r!s!} \frac{1}{\omega_{1+s}} \int_{S^1} \big(v + (1,1)^T\big)^r (-v)^s + 2\big(2v + (1,1)^T\big)^r v^s \,\mathrm{d}v.$$

We are now finally ready to calculate the Minkowski surface tensors of A.

For $k = 0$,

$$\Phi_0^{0,0}(A) = \frac{1}{2\pi} \int_{S^1} 0 \,\mathrm{d}v = 0;$$

$$\Phi_0^{1,0}(A) = \frac{1}{2\pi} \int_{S^1} \begin{pmatrix} 2v_1 + 1 - v_1 - 1 \\ 2v_2 + 1 - v_2 - 1 \end{pmatrix} \mathrm{d}v = \frac{1}{2} \int_0^{2\pi} \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} \mathrm{d}\theta = \begin{pmatrix} 0 \\ 0 \end{pmatrix};$$

$$\begin{aligned}
\Phi_0^{2,0}(A) &= \frac{1}{4\pi} \int_{S^1} \begin{pmatrix} (2v_1+1)^2 - (v_1+1)^2 & (2v_1+1)(2v_2+1) - (v_1+1)(v_2+1) \\ (2v_1+1)(2v_2+1) - (v_1+1)(v_2+1) & (2v_2+1)^2 - (v_2+1)^2 \end{pmatrix} \mathrm{d}v \\
&= \frac{1}{4\pi} \int_0^{2\pi} \begin{pmatrix} 3\cos^2(\theta) + 2\cos(\theta) & 3\cos(\theta)\sin(\theta) + 2\cos(\theta) \\ 3\cos(\theta)\sin(\theta) + 2\sin(\theta) & 3\sin^2(\theta) + 2\sin(\theta) \end{pmatrix} \mathrm{d}\theta \\
&= \begin{pmatrix} 3/4 & 0 \\ 0 & 3/4 \end{pmatrix};
\end{aligned}$$

$$\begin{aligned}
\Phi_0^{1,1}(A) &= \frac{1}{4\pi} \int_{S^1} \begin{pmatrix} (2v_1+1)v_1 + (v_1+1)v_1 & (2v_1+1)v_2 + (v_1+1)v_2 \\ (2v_v+1)v_1 + (v_2+1)v_1 & (2v_2+1)v_2 + (v_2+1)v_2 \end{pmatrix} \mathrm{d}v \\
&= \frac{1}{4\pi} \int_0^{2\pi} \begin{pmatrix} \cos^2(\theta) & \cos(\theta)\sin(\theta) \\ \cos(\theta)\sin(\theta) & \sin^2(\theta) \end{pmatrix} \mathrm{d}\theta = \begin{pmatrix} 3/4 & 0 \\ 0 & 3/4 \end{pmatrix};
\end{aligned}$$

$$\Phi_0^{0,2}(A) = \frac{1}{4\pi^2} \int_{S^1} \begin{pmatrix} v_1^2 - v_1^2 & v_1v_2 - v_1v_2 \\ v_1v_2 - v_1v_2 & v_2^2 - v_2^2 \end{pmatrix} \mathrm{d}v = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},$$

and Lemma C.1 implies that $\Phi_0^{0,1}(A) = \Phi_1^{0,1}(A) = (0,0)^T$. For the remaining surface tensors with $k = 1$,

$$\Phi_1^{0,0}(A) = \frac{1}{2} \int_{S^1} 3 \,\mathrm{d}v = 3\pi;$$

$$\Phi_1^{1,0}(A) = \frac{1}{2} \int_{S^1} \begin{pmatrix} v_1 + 1 + 4v_1 + 2 \\ v_2 + 1 + 4v_2 + 2 \end{pmatrix} \mathrm{d}v = \frac{1}{2} \int_0^{2\pi} \begin{pmatrix} 5\cos(\theta) + 3 \\ 5\sin(\theta) + 3 \end{pmatrix} \mathrm{d}\theta = \begin{pmatrix} 3\pi \\ 3\pi \end{pmatrix};$$

$$\begin{aligned}
\Phi_1^{2,0}(A) &= \frac{1}{4} \int_{S^1} \begin{pmatrix} (v_1+1)^2 + 2(2v_1+1)^2 & (v_1+1)(v_2+1) + 2(2v_1+1)(2v_2+1) \\ (v_1+1)(v_2+1) + 2(2v_1+1)(2v_2+1) & (v_2+1)^2 + 2(2v_2+1)^2 \end{pmatrix} \mathrm{d}v \\
&= \frac{1}{4} \int_0^{2\pi} \begin{pmatrix} 9\cos(\theta)^2 + 3 & 9\cos(\theta)\sin(\theta) + 3 \\ 9\cos(\theta)\sin(\theta) + 3 & 9\sin(\theta)^2 + 3 \end{pmatrix} \mathrm{d}\theta = \begin{pmatrix} 15\pi/4 & 3\pi/2 \\ 3\pi/2 & 15\pi/4 \end{pmatrix};
\end{aligned}$$

$$\begin{aligned}
\Phi_1^{1,1}(A) &= \frac{1}{2\pi} \int_{S^1} \begin{pmatrix} 2(2v_1+1)v_1 - (v_1+1)v_1 & 2(2v_1+1)v_2 - (v_1+1)v_2 \\ 2(2v_2+1)v_1 - (v_2+1)v_1 & 2(2v_2+1)v_2 - (v_2+1)v_2 \end{pmatrix} \mathrm{d}v \\
&= \frac{1}{2\pi} \int_0^{2\pi} \begin{pmatrix} 3\cos^2(\theta) & 3\cos(\theta)\sin(\theta) \\ 3\cos(\theta)\sin(\theta) & \sin^2(\theta) \end{pmatrix} \mathrm{d}\theta = \begin{pmatrix} 3/2 & 0 \\ 0 & 3/2 \end{pmatrix};
\end{aligned}$$

$$\begin{aligned}
\Phi_1^{0,2}(A) &= \frac{1}{8\pi} \int_{S^1} \begin{pmatrix} v_1^2 + 2v_1^2 & v_1v_2 + 2v_1v_2 \\ v_1v_2 + 2v_1v_2 & v_2^2 + 2v_2^2 \end{pmatrix} \mathrm{d}v \\
&= \frac{1}{8\pi} \int_0^{2\pi} \begin{pmatrix} 3\cos^2(\theta) & 3\cos(\theta)\sin(\theta) \\ 3\cos(\theta)\sin(\theta) & \sin^2(\theta) \end{pmatrix} \mathrm{d}\theta = \begin{pmatrix} 3/8 & 0 \\ 0 & 3/8 \end{pmatrix}.
\end{aligned}$$

The volume tensors of A, corresponding to $k = 2$, are given by the formula

$$\Phi_2^{r,0}(A) = \frac{1}{r!} \int_A x^r \,\mathrm{d}x = \frac{1}{r!} \int_{B_2 \setminus B_1} \big(x + (1,1)^T\big)^r \,\mathrm{d}x$$

for $r \in \{0,1,2\}$ from which we calculate

$$\begin{aligned}
\Phi_2^{0,0}(A) &= \int_{B_2\setminus B_1} 1\,\mathrm{d}x = 3\pi;\\
\Phi_2^{1,0}(A) &= \int_{B_2\setminus B_1} \begin{pmatrix} x_1+1\\ x_2+1\end{pmatrix} \mathrm{d}x = \int_0^{2\pi}\int_1^2 \begin{pmatrix} r^2\cos(\theta)+r\\ r^2\sin(\theta)+r\end{pmatrix} \mathrm{d}r\,\mathrm{d}\theta = \begin{pmatrix}3\pi\\3\pi\end{pmatrix};\\
\Phi_2^{2,0}(A) &= \frac{1}{2}\int_{B_2\setminus B_1} \begin{pmatrix} (x_1+1)^2 & (x_1+1)(x_2+1)\\ (x_1+1)(x_2+1) & (x_2+1)^2\end{pmatrix} \mathrm{d}x\\
&= \frac{1}{2}\int_0^{2\pi}\int_1^2 \begin{pmatrix} r(r\cos(\theta)+1)^2 & r(r\cos(\theta)+1)(r\sin(\theta)+1)\\ r(r\cos(\theta)+1)(r\sin(\theta)+1) & r(r\sin(\theta)+1)^2\end{pmatrix} \mathrm{d}r\,\mathrm{d}\theta\\
&= \begin{pmatrix} 27\pi/8 & 3\pi/2\\ 3\pi/2 & 27\pi/8\end{pmatrix}.
\end{aligned}$$

C.1.3 The complement of a Boolean model

The more complicated of the three test sets, the complement of a Boolean model, is realised using the `spatstat` [2] package in R. The code is provided below.

R-code for creating Boolean model data

```
library("nlme")
library("spatstat")
library("foreign")

gamma <- 10 # Intensity
w <- as.owin(c(0,1,0,1)) # Test window
X <- rpoispp(gamma, win=w) # Poisson point process

# Attach radii i.i.d. uniform in [Rmin,Rmax]
Rmin <- 0.08
Rmax <- 0.16
marks(X) <- runif(X$n,min=Rmin,max=Rmax)

write.table(X, "MarkedPoissonProcess.dat", row.names=FALSE, col.names=FALSE)
```

We choose a Boolean model with intensity $\gamma = 10$ and uniform radius distribution $U(0.08, 0.16)$, since this is likely to yield a model with several components in the test window $[0,1]^2 \subset \mathbb{R}^2$ and with some of the particles overlapping. By plotting, as described below, we check that the model is suited for use in the algorithm: if the overlap of any two discs is very small, so is the reach of the complement of the model, and for simulation purposes, we prefer a model with a reach which is not too close to zero. With the code above, we have obtained one such model, the .dat-file of which is

```
0.476299664704129  0.850764135597274 0.140963518023491
0.997265604557469  0.664132648380473 0.119821783825755
0.525292261270806  0.115457823267207 0.130210266839713
0.757840970531106  0.126047324389219 0.095637433771044
0.971295412397012  0.113157438812777 0.146272454746068
0.0676232073456049 0.58372062491253  0.131762228570878
0.484636666253209  0.448957734275609 0.0956309075094759
0.0988877536728978 0.661047839326784 0.131928788013756
```

The first two columns contain the first respectively second coordinates of the centres of the discs constituting the model; the third column their radius. These discs now form a Boolean model Z. We plot the discs in the window $[0,1]^2$ to obtain a picture of the model Z; see Figure 3.3(c).

The algorithms discussed in Chapter 3 are only valid for sets with positive reach, and the Boolean model Z in Figure 3.3(c) does not satisfy this requirement: When two discs overlap, sharp corners arise such as the one determining the size of the reach (marked with a small red line) of the complement in the figure. The complement $A = \mathrm{cl}([0,1]^2 \setminus Z)$, on the other hand, does have positive reach, and we have chosen this particular set as the third test set. To obtain (a value strictly smaller than) the reach of A, we measure the length of the red line, which is (at most) 0.0614.

As might be evident from the two preceding subsections, re-writing (3.3) into a form which can be explicitly computed requires many calculations, and for a set composed of not one or two but eight discs, this would prove a challenge. We therefore choose a different approach to determining the Minkowski tensors of A. In fact, we consider only the intrinsic volumes $\Phi_k^{0,0}(A)$ for $k \in \{0, 1, 2\}$.

The Euler characteristic $\Phi_0^{0,0}(A)$ is easily determined since it is simply equal to the number of connected components minus the number of holes. Notice that the black border of Figure 3.3(c) is not part of the model; it only represents the window $[0, 1]^2$. There is only one component of A since the white discs do not enclose any blue regions, and the number of holes equals two, since there are just two discs which are not connected to $\mathbb{R}^2 \setminus [0, 1]^2$. This yields $\Phi_0^{0,0}(A) = 1 - 2 = -1$.

The tensor $\Phi_1^{0,0}(A)$ equals half the perimeter of the set. We estimate this by manual measurement: we measure the angles of those arcs that contribute to the perimeter and use the true radii from the .dat-file. To the arc lengths, we add the perimeter of $[0, 1]^2$ minus the holes due to discs connected to $\mathbb{R}^2 \setminus [0, 1]^2$. We thus obtain an estimator $\check{\Phi}_1^{0,0}(A) = 6.9669/2 = 3.4835$ of half the perimeter.

To determine the area of A, we sample the area of the Boolean model via the following MATLAB-function, which loads the .dat-file created in R.

Sampling of area

```
function [Area] = SampleBooleanArea()
% SAMPLE_BOOLEAN_AREA estimates the area of a Boolean model by sampling
% with a lattice of resolution 50000

% Load the Boolea model data
[DataFileName,DataPathName] = uigetfile('*.dat','Select data file (.dat)
    containing the Boolean model');

Data = load([DataPathName DataFileName]);

Res = 50000;        % Resolution (reciprocal lattice distance) of sampling
    lattice
Area = 0;           % Counter for area of the Boolean model
N = size(Data,1);   % Total number of Poisson distributed points

for iii = 1:Res
    for jjj = 1:Res
        kkk = 1;                 % Data point number
        GoOn = 1;                % Indicator of loop termination
        Intersection = 0;        % Indicator of intersection between sampling
            point and disc
        xxx = (iii/Res);         % x-increment
        yyy = (jjj/Res);         % y-increment

        % We now sample all Poisson distributed points of the Boolean
        % model. If any of the discs intersect the sampling point (iii,jjj),
        % the loop terminates and we add the area of its pixel to the total
        % area:
        while (kkk <= N && GoOn)
            Intersection = ((Data(kkk,1)-xxx).^2 + (Data(kkk,2)-yyy).^2)
                           <= Data(kkk,3).^2;
            GoOn = ~Intersection;
            kkk = kkk+1;
        end
        Area = Area + Intersection; % The area of a pixel is set to 1
    end
end

Area = Area*(1/Res^2);      % Scaling of the area by the true area of a pixel
```

The function returns the value `0.2968` for our specific .dat-file, and so the estimated area of A is $\check{\Phi}_2^{0,0}(A) = 1 - 0.2968 = 0.7032$.

C.2 DIGITISATION OF THE TEST SETS

In order to be able to use the three test sets described above with Minktensor2D3R, Minktensor2D2R, or Minktensor2D, we must first digitise them. We do so using MATLAB.

A digitisation depends on the resolution d of the lattice dL as well as some choice of origin. Figures 3.8–3.10 represent simulations where the same set A is digitised with different resolutions. The digitisation is given as $A_0 = A \cap dL$ with L some choice of rotated, translated version of $\mathbb{Z}^2$, so while d is varying, we have implicitly made a choice of origin which must be held fixed as the resolution varies. This must be taken into consideration as we write the code to produce our digitised test sets. For the complement of a Boolean model, the .dat-file produced in R needs to be loaded. Furthermore, in our digitisation of Boolean models, we require that the lattice distance be an integer. The .m-files for the digitisations are given below.

Digitisation of the disc

```
function [Origin] = DATAdisc(Res)
% Origin = DATAdisc(Res) writes a .dat-file containing the digitisation of a unit
% disc translated to (1,1)^T. The input for the function is the resolution of the
% digitisation. The function returns the origin in terms of row and column number
% of the array in the .dat-file.

R = 2*ceil(Res)+1;
A = zeros(R,R);
Origin = [R,1];                     % [Row,Column]
DiscCentre = [R-Res,1+Res];         % [Row,Column]

for iii = 1:R
    for jjj = 1:R
        A(iii,jjj) = ((iii-DiscCentre(1))^2 + (jjj-DiscCentre(2))^2 <= Res^2);
    end
end

save Disc.dat A -ASCII
```

Digitisation of the annulus

```
function [Origin] = DATAannulus(Res)
% Origin = DATAannulus(Radius) writes a .dat-file containing the digitisation of
% an annulus equal to the (closure of the) 2-disc minus the unit disc translated
% to (1,1)^T. The input for the function is the resolution of the digitisation.
% The function returns the origin in terms of row and column number of the array
% in the .dat-file.

R = 2*ceil(Res);
A = zeros(2*R+1,2*R+1);
Origin = [2*R+1-ceil(Res),1+ceil(Res)];
DiscCentre = [Origin(1)-Res,Origin(2)+Res];

for iii = 1:2*R+1
    for jjj = 1:2*R+1
        Coordinate = (iii-DiscCentre(1))^2 + (jjj-DiscCentre(2))^2;
        A(iii,jjj) = (Coordinate >= Res^2) && (Coordinate <= (2*Res)^2);
    end
end

save Annulus.dat A -ASCII
```

Digitisation of the complement of a Boolean model

```
function [Origin] = DigitiseBooleanBackground(Res)
% Origin = DigitiseBooleanBackground(Res) writes a .dat-file containing the
% digitisation of the complement of a Boolean model with the resolution Res and
% returns the origin in terms of row and column number of the array in the
% .dat-file.

[DataFileName,DataPathName] = ...
uigetfile('*.dat','Select data file (.dat) containing the Boolean model');

Data = load([DataPathName DataFileName]);

DataSize = size(Data,1);

Res = round(Res);
Origin = [Res,1];
A = zeros(Res,Res);

% [Continues on next page]
```

```
for iii = 1:Res
    for jjj = 1:Res
        kkk = 1;
        GoOn = 1;
        Contained = 0;
        xxx = (iii/Res);
        yyy = (jjj/Res);
        while ((kkk <= DataSize) && GoOn)
            Contained = ((Data(kkk,1)-xxx)^2+(Data(kkk,2)-yyy)^2) ...
            <= Data(kkk,3)^2;
            A(iii,jjj) = ~Contained;
            GoOn = ~Contained;
            kkk = kkk+1;
        end
    end
end

B = rot90(A); % We rotate to get a coordinate system with origin at [Res,1]

save BooleanBackground.dat B -ASCII
```

D

DOCUMENTATION OF MINKTENSOR2D

This appendix presents a worked example of how to use the programme Minktensor2D, which was examined in Chapter 3, followed by some documentation for the code. The programme consists of a number of .m-files, its components, and we describe the structure of the code and the underlying theory of the respective components. All code in the .m-files is documented, but some steps of the algorithm need additional explanation thus included here.

The programme Minktensor2D is available for download [6]. In the last section of this appendix, we also include the code for completeness.

D.1 WORKED EXAMPLE: MINKTENSOR2D

We begin this appendix with a worked example of the use of Minktensor2D. The reader is encouraged to follow the procedure for his/her own digitisation of a set with positive reach after downloading the programme, or one of the digitisations of Section C.2 can be used.

The main file of the code is `Minktensor2D.m`. The file should be placed in a folder together with the remaining .m-files of the programme which are:

```
computeMinkowskiTensorEstimators.m
computeTensorMeasureComponents.m
computeVolumeTensorEstimators.m
computeVoronoiCells.m
computeVoronoiTensorMeasures.m
convertDigitisationToCoordinates.m
intersectLineCircle.m
intersectLinesegmentCircle.m
RBoundedVoronoiVertex.m
retrieveData.m
retrieveRadii.m
verticesToCounterclockwiseOrder.m
```

Use of the programme is now explained in a series of steps which should be followed using the desired digitisation.

Step 1. Preparing a digitisation: `Minktensor2D.m` takes as input a digitisation A_0 of some set $A \subset \mathbb{R}^2$ with positive reach. The digitisation should be given as a file containing a whitespace separated list of zeros and ones, e.g.

```
0 0 0 0 1 0 0 0 0
0 0 1 1 1 1 1 0 0
0 1 1 1 1 1 1 1 0
0 1 1 0 0 0 1 1 0
1 1 1 0 0 0 1 1 1
0 1 1 0 0 0 1 1 0
0 1 1 1 1 1 1 1 0
0 0 1 1 1 1 1 0 0
0 0 0 0 1 0 0 0 0
```

is a digitisation of an annulus. Let us denote this particular test set by T_0. A `1` indicates that the corresponding lattice point is contained in the digitisation; a `0` that it is not. Such a file must be available upon execution of the programme.

Step 2. Determining the lattice distance: Next, one must determine the lattice distance d of the digitisation. This equals the reciprocal of the resolution of the digitisation, and the user is expected to know d or d^{-1} of the digitisation chosen in step 1.

For the digitised annulus T_0, we know that the lattice distance d_T is 0.5.

Step 3. Determining the origin of the digitisation: The programme needs the user to input the origin of the digitisation. The origin must be given in terms of the row and column numbers of the digitisation. The origin may not coincide with a lattice point, so the specified row and column numbers can be any real number, e.g. a digitisation with n rows and m columns could be placed in a coordinate system such that the mth element of row n is at coordinate (s,t) for any $s,t \in \mathbb{R}$, so we specify a row number $n+t$ and a column number $m-s$.

The set T_0 has origin at the lattice point in the lower left corner of the digitisation, so the row number is $r_T = 9$ and the column number is $c_T = 1$.

Step 4. Choosing radii for the algorithm: Two radii $1/\sqrt{2} < R_0 < R_2 < \text{Reach}(A)$ must be chosen for the algorithm. Based on the results in Subsection 3.4.1, we recommend $R_2 = 0.95 \cdot \text{Reach}(A)$ in the case of finite reach or, in the case of convex sets with infinite reach, R_2 equal to several diameters of the digitised set. The second radius R_0 should be chosen close to $R_2/2$ and never below $d/\sqrt{2}$.

The annulus T digitised to T_0 equals $\text{cl}(B_2 \setminus B_1) + (1,1)^T$, so its reach is 1. We choose $R_1 = 0.95$ and $R_0 = 0.5$.

Step 5. Running the programme: Having completed the preparatory steps 1–4, we are now ready to execute the programme Minktensor2D. This is done by running `Minktensor2D.m` in MATLAB. Once the programme is running, the user is prompted to:

(1) specify a file containing a digitisation;

(2) enter the row number of the origin;

(3) enter the column number of the origin;

(4) enter a lower radius, R_0;

(5) enter an upper radius, R_2.

The programme then outputs estimators of all Minkowski tensors $\Phi_k^{r,s}(A)$ for $k,r,s \in \{0,1,2\}$ with $r+s \leq 2$ and $s=0$ when $k=2$.

Since we have chosen a digitisation T_0 with a very low resolution, $(d_T)^{-1} = 2$, in our example, the programme yields very poor estimators.

D.2 DOCUMENTATION OF THE PROGRAMME

The algorithm Minktensor2D for obtaining estimators of the Minkowski tensors of sets with positive reach is explained in depth in Chapter 3. However, the implementation itself requires many subtle calculations. These are explained in the following.

Convention

Convention. For the remainder of this appendix, A will denote a set with positive reach and $A_0 = A \cap dL$ a digitisation of A by a lattice dL with lattice distance $d > 0$. ■

D.2.1 STRUCTURE OF THE .m-FILES

Let us first consider the structure of the .m-files that constitute `Minktensor2D`. The main file is `Minktensor2D.m`, and this file makes use of a number of primary .m-files which in turn make use of a number of secondary .m-files and so on. The structure is as follows:

`Mintensor2D.m`:

(1) `retrieveData.m`

(2) `retrieveRadii.m`

(3) `computeVoronoiCells.m`

  ○ `convertDigitisationToCoordinates.m`

(4) `computeVolumeTensorEstimators.m`

(5) `computeMinkowskiTensorEstimators.m`

- ○ `computeVoronoiTensorMeasures.m`
  - – `verticesToCounterclockwiseOrder.m`
  - – `intersectLinesegmentCircle.m`
  - – `RBoundedVoronoiVertex.m`
    - ∗ `intersectLineCircle.m`
  - – `computeTensorMeasureComponents.m`

The secondary .m-files of `Mintensor2D.m` are executed in a particular order, 1–5, and these as well as their secondary files, the tertiary files, are executed only once. However, the files secondary and tertiary to `computeVoronoiTensorMeasures.m` are executed a number of times not necessarily following each other.

Given the structure of the code, one can read through the .m-files in Section D.3 to figure out how the algorithm is implemented. A full understanding of the code is facilitated by the explanations in the following subsections.

D.2.2 Lattice distance as unity

Reading through the .m-files which constitute the programme `Minktensor2D` as described above, in `retrieveRadii.m`, one will notice that the radii are all scaled by the lattice distance. All calculations are in fact made with d as unity. This is only corrected for at the very end of the programme, when the Voronoi tensors measures are calculated in `computeMinkowskiTensors.m`. The reason for the scaling is that it is much easier to convert lattice points to coordinates when the distance between neighbouring rows and columns of the digitisation file is 1, and the volume of the Voronoi cell of any inner point of the digitisation also becomes 1. We need to consider the effect of this choice on the algorithm.

We wish to solve the system

$$\begin{pmatrix} \mathcal{V}^{r,s}_{R_0}(A_0) \\ \mathcal{V}^{r,s}_{R_1}(A_0) \\ \mathcal{V}^{r,s}_{R_2}(A_0) \end{pmatrix} = M^{r,s}_{R_0,R_1,R_2} \begin{pmatrix} \Phi^{r,s}_2(A_0) \\ \Phi^{r,s}_1(A_0) \\ \Phi^{r,s}_0(A_0) \end{pmatrix}. \tag{D.1}$$

as explained in Subsection 3.2.3. The simplification $d = 1$ means that we are in fact computing tensors for the set $B_0 = A_0/d$ as well as using radii $1/\sqrt{2} < R_0/d < R_1/d < R_2/d < \mathrm{Reach}(A)/d$, so we obtain Voronoi tensor measures $\mathcal{V}^{r,s}_{R_i/d}(B_0)$ instead of $\mathcal{V}^{r,s}_{R_i}(A_0)$ for $i \in \{0, 1, 2\}$. We thus obtain a system

$$\begin{pmatrix} \mathcal{V}^{r,s}_{R_0/d}(B_0) \\ \mathcal{V}^{r,s}_{R_1/d}(B_0) \\ \mathcal{V}^{r,s}_{R_2/d}(B_0) \end{pmatrix} = M^{r,s}_{R_0,R_1,R_2} \begin{pmatrix} d^{-s} & 0 & 0 \\ 0 & d^{-s-1} & 0 \\ 0 & 0 & d^{-s-2} \end{pmatrix} \begin{pmatrix} \Phi^{r,s}_2(B_0) \\ \Phi^{r,s}_1(B_0) \\ \Phi^{r,s}_0(B_0) \end{pmatrix}. \tag{D.2}$$

Note that

$$\Phi^{r,s}_k(A) = \frac{1}{r!s!} \frac{\omega_{d-k}}{\omega_{d-k+s}} \int_{\Sigma} x^r u^s \Lambda(A; \mathrm{d}(x,u))$$

implies $\Phi^{r,s}_k(A/d) = d^{-r-k}\Phi^{r,s}_k(A)$, where the exponent of d stems from the scaling of x^r and the generalised curvature measures $\Lambda(A; \mathrm{d}(x,u))$. We can now re-write (D.2) to

$$\begin{aligned} \begin{pmatrix} \mathcal{V}^{r,s}_{R_0/d}(B_0) \\ \mathcal{V}^{r,s}_{R_1/d}(B_0) \\ \mathcal{V}^{r,s}_{R_2/d}(B_0) \end{pmatrix} &= M^{r,s}_{R_0,R_1,R_2} \begin{pmatrix} d^{-s} & 0 & 0 \\ 0 & d^{-s-1} & 0 \\ 0 & 0 & d^{-s-2} \end{pmatrix} \begin{pmatrix} d^{-2-r}\Phi^{r,s}_2(A_0) \\ d^{-1-r}\Phi^{r,s}_1(A_0) \\ d^{-r}\Phi^{r,s}_0(A_0) \end{pmatrix} \\ &= M^{r,s}_{R_0,R_1,R_2} \cdot d^{-s-r-2} \begin{pmatrix} \Phi^{r,s}_2(A_0) \\ \Phi^{r,s}_1(A_0) \\ \Phi^{r,s}_0(A_0) \end{pmatrix} \\ &= d^{-s-r-2} \begin{pmatrix} \mathcal{V}^{r,s}_{R_0}(A_0) \\ \mathcal{V}^{r,s}_{R_1}(A_0) \\ \mathcal{V}^{r,s}_{R_2}(A_0) \end{pmatrix}, \end{aligned}$$

and we conclude $\mathcal{V}^{r,s}_{R_i/d}(B_0) = d^{s+r+2}\mathcal{V}^{r,s}_{R_i}(A_0)$ for $i \in \{0,1,2\}$ and $r,s \in \{0,1,2\}$ with $r+s \leq 2$. We can thus use the Voronoi tensor measures $\mathcal{V}^{r,s}_{R_i/d}(B_0)$ to obtain $\mathcal{V}^{r,s}_{R_i}(A_0)$, $i \in \{0,1,2\}$ and then solve the original system (D.1) for the Minkowski tensors. This fact is exploited in the .m-file `computeMinkowskiTensors.m` when all Voronoi tensors measures $\mathcal{V}^{r,s}_{R_i/d}(B_0)$ have been computed. The method for computing the tensor measures in practice is described in Subsection D.2.4.

The scaling of the digitisation must also be taken into consideration when the volume tensor estimators are calculated in `computeVolumeTensorEstimators.m`. From (3.13) and by replacing A_0 with $B_0 = A_0/d$, we get

$$\tilde{\Phi}^{r,0}_2(A_0) = \frac{d^2}{r!}\sum_{x\in A_0} x^r = \frac{d^2}{r!}\sum_{x\in B_0}(x\cdot d)^r = \frac{d^{r+2}}{r!}\sum_{x\in B_0} x^r,$$

and this formula is used in `computeVolumeTensorEstimators.m` to obtain estimators for the volume tensors of A_0. The estimators are output as scalars, vectors, and 2-by-2 matrices as explained in (3.1), so for $x = (x_1,x_2)^T$,

$$\begin{aligned}
\tilde{\Phi}^{0,0}_2(A_0) &= d^2\sum_{x\in B_0} x^0 = d^2\sum_{x\in B_0} 1 = d^2|B_0|;\\
\tilde{\Phi}^{1,0}_2(A_0) &= d^3\sum_{x\in B_0} x^1 = d^3\sum_{x\in B_0}\begin{pmatrix} x_1\\ x_2\end{pmatrix};\\
\tilde{\Phi}^{2,0}_2(A_0) &= d^4\sum_{x\in B_0} x^2 = d^4\sum_{x\in B_0}\begin{pmatrix} x_1^2 & x_1x_2\\ x_1x_2 & x_2^2\end{pmatrix},
\end{aligned}$$

where $|B_0|$ denotes the cardinality of B_0, and where we use the identity $x_1x_2 = x_2x_2$ in the last line (the tensors are symmetric).

D.2.3 Optimising computations of Voronoi cells

The most time consuming step of the programme is the calculation of Voronoi cells for the sampling points in the digitisation. The vertices of the Voronoi cells are determined by use of the Math function `voronoin` in MATLAB, which is based on `Qhull` [3]. We give now a description of how to reduce the computation time, at least for large digitisations. This method has been implemented in `computeVoronoiCells.m`.

We need the Voronoi cells to compute the Voronoi tensor measures via (3.10),

$$\mathcal{V}^{r,s}_{R_i}(A_0) = \sum_{x\in A_0} x^r \int_{\mathcal{V}_{A_0}(x)\cap B_{R_i}(x)} (y-x)^s\,\mathrm{d}y. \tag{3.10}$$

One way to reduce computation times is the following: As we shall see in Subsection D.2.4, the contribution of an inner point $x \in A_0$ to the Voronoi tensors measures is determined solely by the coordinates of x as well as the volume of its R-bounded Voronoi cell, that is, the vertices of its Voronoi cell play no part in the calculations. Hence, there is no reason to compute the Voronoi vertices for inner points. We still need the Voronoi vertices for all **outer points** of the digitisation, i.e. points in A_0 that are not inner. The vertices of the Voronoi cell of such a point $x \in A_0$ are determined by the coordinates of other outer points as well as by any four-neighbours of x in A_0; there can be up to three four-neighbours of x in A_0, since x is an outer point. The four-neighbours of x may be inner points. We will refer to such inner points in A_0 with outer points as four-neighbours as **semi-outer points** of the digitisation.

Together, the inner points and the semi-outer points of A_0 determine the Voronoi cells, and hence their vertices, of all inner points of A_0. However, some Voronoi cells will be unbounded, and `voronoin` specifies only the vertices of the Voronoi cells, which means we lack information of those edges that have vertices at infinity. We need to be able to determine those edges later on when we calculate the R_i-bounded Voronoi cells for $i \in \{0,1,2\}$. To circumvent this problem, we introduce artificial **border points** of the digitisation; points that, in the coordinate system of the digitisation, are placed so far away from any point $x \in A_0$ that they do not alter the edges of $\mathcal{V}_{A_0}(x)$ inside $B_{R_i}(x)$ for $i \in \{0,1,2\}$, but so that every outer point will have its Voronoi cell bounded by these border points.

Having determined the inner and semi-outer points of the digitisation as well as border points, we use `voronoin` to compute the Voronoi cells of all these points in `computeVoronoiCells.m`. The function `voronoin` returns an array of vertices of the Voronoi cells as well as a cell array with indices for the Voronoi cell of each point. We then delete the indices corresponding to semi-outer and border points, and the border points are discarded. We thus end up with artificially bounded Voronoi cells for all outer points x of A_0 which coincide with $\mathcal{V}_{A_0}(x)$ inside $B_{R_i}(x)$ for all $i \in \{0,1,2\}$.

The above procedure of course adds slightly to the computation time in the case where there are only few or no inner points in the digitisation, since the programme runs through all the points of the digitisation to check whether they are inner or semi-inner. However, for large data sets, the loss of time on this additional check is by far outweighed by the increase of effectiveness when the computation of the Voronoi cells of all inner points that are not semi-outer is avoided.

D.2.4 Calculation of the Voronoi tensor measures

To calculate the Voronoi tensor measures in `computeVoronoiTensorMeasures.m` for the scaled digitisation $A_0/d = B_0$, we utilise (3.10). Computing the integrals

$$\mathbb{I}_R^s(x) = \int_{\mathcal{V}_{B_0}(x)\cap B_R(x)} (y-x)^s \,\mathrm{d}y$$

for $x \in B_0$, $R > 1/\sqrt{2}$, and $s \in \{0,1,2\}$ in practice is not straightforward, since we need to specify the R-bounded Voronoi cell $\mathcal{V}_{B_0}(x) \cap B_R(x)$ of x. We have the vertices of the Voronoi cells of all points of B_0 at our disposal; those of the outer points of the digitisation are computed in `computeVoronoiCells.m`, and the inner points x have trivial Voronoi cells equal to squares centred at x and with sides of length 1 parallel to the axis directions of the given coordinate system. The Voronoi tensor measures can be split into contributions from inner and outer points of the digitisation. Denote by $\mathrm{outer}(B_0)$ the outer points and by $\mathrm{inner}(B_0)$ the inner points of B_0. Then

$$\mathcal{V}_R^{r,s}(B_0) = \sum_{x\in B_0} x^r \mathbb{I}_R^s(x) = \sum_{x\in \mathrm{inner}(B_0)} x^r \mathbb{I}_R^s(x) + \sum_{x\in \mathrm{outer}(B_0)} x^r \mathbb{I}_R^s(x) \tag{D.3}$$

for $r, s \in \{0,1,2\}$ such that $r+s \le 2$ and $R > 1/\sqrt{2}$. We will now describe the procedure for computing the integral $\mathbb{I}_R^s(x)$ for inner respectively outer points of B_0. The considerations made below for the R-bounded Voronoi cells of inner respectively outer points with $R > 1/\sqrt{2}$ are illustrated in Figure 3.1(a).

Inner points: Inner points $x \in B_0$ are easily handled since we require that R be greater than $1/\sqrt{2}$. This means that the R-bounded Voronoi cell $\mathcal{V}_{B_0}(x) \cap B_R(x)$ is simply equal to $\mathcal{V}_{B_0}(x)$, which, as mentioned above, is the unit square $[0,1]^2$ translated to x. Thus,

$$\begin{aligned}
\mathbb{I}_R^0(x) &= \int_{[0,1]^2+x} (y-x)^0 \,\mathrm{d}y = \int_{[0,1]^2} 1 \,\mathrm{d}y = \mathrm{vol}([0,1]^2) = 1;\\
\mathbb{I}_R^1(x) &= \int_{[0,1]^2} y \,\mathrm{d}y = \int_{-1/2}^{1/2}\int_{-1/2}^{1/2} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \mathrm{d}y_1\,\mathrm{d}y_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix};\\
\mathbb{I}_R^2(x) &= \int_{[0,1]^2} y^2 \,\mathrm{d}y = \int_{-1/2}^{1/2}\int_{-1/2}^{1/2} \begin{pmatrix} y_1^2 & y_1y_2 \\ y_1y_2 & y_2^2 \end{pmatrix} \mathrm{d}y_1\,\mathrm{d}y_2 = \begin{pmatrix} 1/12 & 0 \\ 0 & 1/12 \end{pmatrix},
\end{aligned}$$

where $y = (y_1, y_2)^T$. We then compute the terms $x^r\mathbb{I}_R^s(x)$ in (D.3) for all inner points $x \in B_0$ using these three equations.

Outer points: For outer points $x \in \mathrm{outer}(B_0)$, we need a procedure for determining the vertices of the R-bounded Voronoi cell of x. In Subsection D.2.3, we described how `computeVoronoiCells.m` computes a finite polygon, which we denote now by $\tilde{\mathcal{V}}_{B_0}(x)$, for outer points x; the **bounded Voronoi cell** of x determined by the choice of border points. The bounded Voronoi cell of x satisfies $\tilde{\mathcal{V}}_{B_0}(x) \subset \mathcal{V}_{B_0}(x)$, and most importantly $\tilde{\mathcal{V}}_{B_0}(x) \cap B_R(x) = \mathcal{V}_{B_0}(x) \cap B_R(x)$.

Let $v_1, \dots, v_k$ be the vertices of $\tilde{\mathcal{V}}_{B_0}(x)$, where $k \ge 3$ since $\tilde{\mathcal{V}}_{B_0}(x)$ is a polygon, and suppose that $v_1, \dots, v_k$ are ordered counterclockwise relative to x. We wish to convert $v_1, \dots, v_k$ to vertices (if any) of the R-bounded Voronoi cell $\mathcal{V}_{B_0}(x) \cap B_R(x)$ (which might

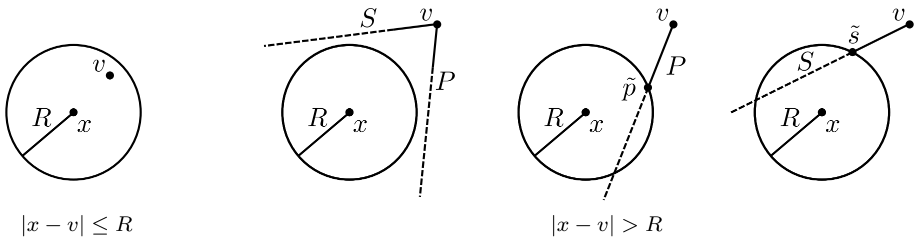


Figure D.1
The different cases to consider when determining whether a vertex v of the Voronoi cell of an outer point $x \in A_0$ is, or gives rise to, a vertex of the R-bounded Voronoi cell.

equal the disc $B_R(x)$). There turns out to be several different cases to treat. Let $v = v_i$ for any $i \in \{1, \dots, k\}$, let $p = p_i$ be the previous vertex, $p_i = v_{i-1}$ for $i \geq 2$ and $p_1 = v_k$, and let $s = s_i$ be the subsequent vertex, $s_i = v_{i+1}$ for $i < k$ and $s_k = v_1$. The cases to be treated below are illustrated in Figure D.1.

If $|x - v| \leq R$, v is already a vertex of the R-bounded Voronoi cell, and v is connected to the subsequent vertex of $\mathcal{V}_{B_0}(x) \cap B_R(x)$ by a line segment in the boundary of the R-bounded Voronoi cell.

Otherwise, if $|x - v| > R$, v is not a vertex of the R-bounded Voronoi cell. Denote by $P = P_i$ the line segment from x to p and by $S = S_i$ the line segment from x to s. We need to determine whether v 'gives rise to' any vertices of the R-bounded Voronoi cell: if S or P intersects the boundary of $B_R(x)$, new vertices arise in the points of intersection of the line segments and $\partial B_R(x)$ nearest to v. Hence, we need to examine whether S or P intersect the boundary of $B_R(x)$.

If neither P nor S intersects $\partial B_R(x)$, v does not give rise to vertices of the R-bounded Voronoi cell and is not itself a vertex of it. Thus v can be discarded.

If P intersects $\partial B_R(x)$, v gives rise to a vertex $\tilde p = \tilde p_i$ of the R-bounded Voronoi cell unless $\tilde p = p$, in which case no new vertex arises from this intersection. If $\tilde p \neq p$, $\tilde p$ is a vertex of the R-bounded Voronoi cell, and it is connected to the subsequent vertex of $\mathcal{V}_{B_0}(x) \cap B_R(x)$ by an arc in $\partial(\mathcal{V}_{B_0}(x) \cap B_R(x))$.

If S intersects $\partial B_R(x)$, v gives rise to a vertex $\tilde s = \tilde s_i$ of the R-bounded Voronoi cell unless $\tilde s = s$, in which case no new vertex arises from this intersection. If $\tilde s \neq s$, $\tilde s$ is a vertex of the R-bounded Voronoi cell, and it is connected to the subsequent vertex of $\mathcal{V}_{B_0}(x) \cap B_R(x)$ by a line segment in $\partial(\mathcal{V}_{B_0}(x) \cap B_R(x))$.

We follow the procedure described above for all outer points of B_0 in order to determine all vertices $w_1, \dots, w_m$, $m \geq 0$, of $\mathcal{V}_{B_0}(x) \cap B_R(x)$ and how these are connected to the subsequent vertex along $\partial(\mathcal{V}_{B_0}(x) \cap B_R(x))$, either by line segments or arcs. This information is now input into `computeTensorMeasureComponents.m`. This script computes the terms $x^r \mathbb{I}^s_R(x)$. To compute the integral $\mathbb{I}^s_R(x)$, we will split the region of integration, i.e. the R-bounded Voronoi cell, into subregions determined by the vertices $w_1, \dots, w_m$ (if any).

We write line(x) for vertices $w = w_i$ such that w is connected to the subsequent R-bounded Voronoi vertex $u = u_i$ by an arc in $\partial(\mathcal{V}_{B_0} \cap B_R(x))$. Consider for $w \in \text{line}(x)$ the unbounded region C with edges the half-open line segments with end point x and intersecting w respectively u. We now take the intersection of C with $B_R(x)$. This forms a region which we will denote by $\vartriangleright(w)$.

Similarly, we write arc(x) for vertices w such that w is connected to u by a line segment in $\partial(\mathcal{V}_{B_0} \cap B_R(x))$. We denote by $\triangleright(w)$ the polygon with vertices x, w, and u. The R-bounded Voronoi cell of x then coincides with the union over all w_i of such $\triangleright(w_i)$ and $\vartriangleright(w_i)$, and these subsets have pairwise disjoint interiors. We can thus compute the integral $\mathbb{I}^s_R(x)$ by

$$\mathbb{I}^s_R(x) = \sum_{w \in \text{arc}(x)} \iint_{\vartriangleright(w)} (y - x)^s \, \mathrm{dA} + \sum_{w \in \text{line}(x)} \iint_{\triangleright(w)} (y - x)^s \, \mathrm{dA}$$

for $x \in \text{outer}(B_0)$, $s \in \{0, 1, 2\}$, and $R > 1/\sqrt{2}$, and these integrals over regions $\vartriangleright(w)$ and $\triangleright(w)$ are now possible to compute in MATLAB.

Indeed, if $w \in \text{arc}(x)$, let u be the subsequent point to w as above. Denote by θ_w the angle of $w - x$ in the coordinate system of the digitisation and by θ_u that of $u - x$. Then

$$\iint_{\vartriangleright(w)} (y - x)^s \, \mathrm{dA} = \int_{\theta_w}^{\theta_u} \int_0^R \begin{pmatrix} r^2 \cos(\theta) \\ r^2 \sin(\theta) \end{pmatrix}^s \mathrm{d}r \, \mathrm{d}\theta.$$

The primitive functions of the integrals have been computed for each $s \in \{0, 1, 2\}$, and these are written as functions of θ_w, θ_u, and R in `computeTensorMeasureComponents.m`, for example

$$\iint_{\triangleright(w)} (y-x)^1 \,\mathrm{dA} = \frac{1}{3}\begin{pmatrix} R^3\big(\sin(\theta_u) - \sin(\theta_w)\big) \\ R^3\big(\cos(\theta_u) - \cos(\theta_w)\big) \end{pmatrix}.$$

If there are no vertices or just one of the R-bounded Voronoi cell,

$$\mathbb{I}_R^{r,s}(x) = \int_0^{2\pi}\int_0^R \begin{pmatrix} r^2\cos(\theta) \\ r^2\sin(\theta) \end{pmatrix}^s \mathrm{d}r\,\mathrm{d}\theta.$$

If, on the other hand, $w \in \operatorname{line}(x)$, integration in MATLAB is achieved as follows: We again denote by u the subsequent vertex to w. The idea now is to rotate and translate our vectors to a coordinate system with axes p, q and origin x such that w lies on the p-axis. Let $p_1 = |w - x| \neq 0$, $\tilde{p} = (\tilde{p}_1, \tilde{p}_2)^T = (w-x)/p_1$, $\tilde{q} = (-\tilde{p}_2, \tilde{p}_1)^T$, $p_0 = \langle u - x, \tilde{p}\rangle$, and $q_0 = \langle u - x, \tilde{q}\rangle$. Then the coordinates of u in the p,q-coordinate system are (p_0, q_0). Now $\tilde{p}$ is a unit vector on the p-axis and $\tilde{q}$ a unit vector on the q-axis. Let $e_1 = (1,0)^T$ and $e_2 = (0,1)^T$ be a basis for the coordinate system of the digitisation. Expressed in the p,q-coordinate system, e_1 and e_2 become

$$\begin{pmatrix} \langle e_1, \tilde{p}\rangle\tilde{p} + \langle e_1, \tilde{q}\rangle\tilde{q} \\ \langle e_2, \tilde{p}\rangle\tilde{p} + \langle e_2, \tilde{q}\rangle\tilde{q} \end{pmatrix} =: \begin{pmatrix} e_{1,p}\tilde{p} + e_{1,q}\tilde{q} \\ e_{2,p}\tilde{p} + e_{2,q}\tilde{q} \end{pmatrix}.$$

For the integration, we will also need to express the line through x and u as a function $g(p)$ and similarly the line through u and w as a function $h(p)$, since these functions will be needed to define the region of integration. If, however, $p_0 = 0$, we will not need the function $g(p)$, and if $p_0 = p_1$, we will not need $h(p)$.

Suppose $p_0 \neq 0$. Clearly, $g(p) = ap$ for some $a \in \mathbb{R}$. Since $g(p_0) = q_0$, $a = q_0/p_0$. For $p_0 \neq p_1$, write $h(p) = bp + c$ for $b, c \in \mathbb{R}$. Note that $h(p_0) = q_0 = bp_0 + c$, so $c = q_0 - bp_0$. Further, $h(p_1) = 0 = bp_1 + c = bp_1 + q_0 - bp_0 = b(p_1 - p_0) + q_0$, so $b = q_0/(p_0 - p_1)$ and $c = q_0(1 + p_0/(p_1 - p_0))$. To summarise,

$$g(p) = \frac{q_0}{p_0}p \qquad \text{for } p_0 \neq 0$$

and

$$h(p) = \frac{q_0}{p_0 - p_1}p + q_0\left(1 + \frac{p_0}{p_1 - p_0}\right) \qquad \text{for } p_0 \neq p_1.$$

We now have

$$\begin{aligned} &\iint_{\triangleright(w)} (y-x)^s \,\mathrm{dA} \\ &\quad = \int_0^{p_0}\int_0^{g(p)} \begin{pmatrix} e_{1,p}p + e_{1,q}q \\ e_{2,p}p + e_{2,q}q \end{pmatrix}^s \mathrm{d}q\,\mathrm{d}p + \int_{p_0}^{p_1}\int_0^{h(p)} \begin{pmatrix} e_{1,p}p + e_{1,q}q \\ e_{2,p}p + e_{2,q}q \end{pmatrix}^s \mathrm{d}q\,\mathrm{d}p \end{aligned} \tag{D.4}$$

for $p_0 \neq (0, p_1)$ and $s \in \{0, 1, 2\}$. As for $w \in \operatorname{arc}(x)$, the primitive functions of the integral have been computed and are given as functions of p_0, q_0, p_1, $e_{1,p}$, $e_{1,q}$, $e_{2,p}$, and $e_{2,q}$ in `computeTensorMeasureComponents.m`, e.g.

$$\iint_{\triangleright(w)} (y-x)^1 \,\mathrm{dA} = \frac{1}{6}\begin{pmatrix} p_0q_0(2p_0e_{1,p} + q_0e_{1,q}) - q_0(p_0 - p_1)(2p_0e_{1,p} + q_0e_{1,q}) \\ p_0q_0(2p_0e_{2,p} + q_0e_{2,q}) - q_0(p_0 - p_1)(2p_0e_{2,p} + q_0e_{2,q}) \end{pmatrix}.$$

The first term of each of the two entries of the vector on the right-hand side corresponds to the first double integral on the right-hand side of (D.4) and, analogously, the second term to the second double integral. For $p_0 = 0$, the first double integral of (D.4) vanishes, and so the first term of the vector entries should not be included, but notice that for $p = 0$, the term in fact automatically vanishes. The same applies to the second term for $p_0 = p_1$. Thus the primitive functions can be defined as above also for $p_0 = 0$ and $p_0 = p_1$ which is exploited in `computeTensorMeasureComponents.m`. Similar considerations can be made for $s \in \{0, 2\}$.

The above allows us to compute the remaining terms $x^r\mathbb{I}_R^s(x)$ in (D.3) for all outer points of B_0.

D.3 MATLAB-CODE FOR MINKTENSOR2D

This section contains the code for every .m-file used in Minktensor2D.

Minktensor2D.m

```
% Minktensor2D computes estimators of Minkowski tensors in dimension 2 for
% compact sets with positive reach given a digitisation of the set.

% User must input a digitisation in form of a datafile containing an array
% of 0's and 1's, where a 1 indicates that the corresponding lattice point
% is contained within the set with positive reach; a 0 that it is not. The
% user must also specify the lattice distance as well as the origin, where
% the latter must be expressed as a row and column number corresponding to
% the 0-1-array (where these numbers need not be integers nor positive
% since the origin may not coincide with a lattice point). Finally, a lower
% and an upper radius for use in the algorithm must be chosen. The radii
% must not be lower than 1/sqrt(2) times the lattice distance, and neither
% radius must exceed the reach of the digitised object.

% Primary (necessary) .m-files:
% computeMinkowskiTensorEstimators.m
% computeVolumeTensorEstimators.m
% computeVoronoiCells.m
% retrieveData.m
% retrieveRadii.m
%
% Secondary .m-files used by the primary .m-files
% computeTensorMeasureComponents.m
% computeVoronoiTensorMeasures.m
% convertDigitisationToCoordinates.m
% intersectLineCircle.m
% intersectLinesegmentCircle.m
% RBoundedVoronoiVertex.m
% verticesToCounterclockwiseOrder.m

% Retrieve digitisation data, lattice distance, and origin from user
[digitisation,a,origin] = retrieveData();

% Retrieve radii from user
radii = retrieveRadii(a);

% Find planar coordinates of the sampling points in digitisation as well as
% the vertices and indices of the Voronoi cells of the sampling points
[outerPoints,innerPoints,VoronoiIndices,VoronoiVertices] = ...
   computeVoronoiCells(digitisation,radii(3),origin);

% Compute estimators of the volume tensors
[Phi200,Phi210,Phi220] = ...
   computeVolumeTensorEstimators(outerPoints,innerPoints,a);

% Compute estimators of the Minkowski tensors of rank at most two
[Phi000,Phi010,Phi020,Phi001,Phi011,Phi002,Phi100,Phi110,Phi120,...
   Phi101,Phi111,Phi102] = computeMinkowskiTensorEstimators(radii,...
   innerPoints,outerPoints,VoronoiIndices,VoronoiVertices,a);

% Display all estimators

disp('The estimators of the Minkowski tensors are:')

disp('Phi000'),disp(Phi000);
disp('Phi010'),disp(Phi010);
disp('Phi020'),disp(Phi020);
disp('Phi001'),disp(Phi001);
disp('Phi011'),disp(Phi011);
disp('Phi002'),disp(Phi002);

disp('Phi100'),disp(Phi100);
disp('Phi110'),disp(Phi110);
disp('Phi120'),disp(Phi120);
disp('Phi101'),disp(Phi101);
disp('Phi111'),disp(Phi111);
disp('Phi102'),disp(Phi102);

disp('Phi200'),disp(Phi200);
disp('Phi210'),disp(Phi210);
disp('Phi220'),disp(Phi220);
```

RetrieveData.m

```
function [digitisation,latDist,origin] = retrieveData()
% digitisation = retrieveData() returns an array corresponding to a loaded
% data file specified by the user
% [dagitisation,latDist,origin] = retrieveData() also returns the lattice
% distance latDist and origin of the digitisation both specified by the
% user

% Ask user to choose a data file of type .dat
[DataFileName,DataPathName] = ...
   uigetfile('*.dat','Select data file (.dat) containing the digitisation');

% Load the specified data file
digitisation = load([DataPathName DataFileName]);

% Determine the size of the data file
DataColumns = size(digitisation,2);
DataRows = size(digitisation,1);

% Ask user to input the lattice distance of the digitisation
latDist = input('>> Please enter the lattice distance of your digitisation. ');

% If latDist is not a positive number, request a new lattice distance value
while (( ~isnumeric(latDist) ) || ( isempty(latDist) ) || (latDist <= 0))
    latDist = ...
       input('>> Your input is not a positive number. Please try again: ');
end

disp(...
  'Entries of the data file are given by coordinates [row number,column number]')
disp('read from top to bottom respectively left to right.')
disp(['Coordinates of the origin are required.',...
  'Since the origin may not be a data point, '])
disp(...
  'in the following, row and column numbers are allowed to be any real number.')

% Ask the user for the origin of the data file
originr = ...
   input('>> Please enter the row number of the origin in the data file. ');
% 'Row number' of the origin

while (( originr < 1 ) || (originr > DataRows))
    originr = ...
       input('>> Your input is not a valid row number. Please try again: ');
end

originc = ...
   input('>> Please enter the column number of the origin in the data file. ');
% 'Column number' of the origin

while (( originc < 1 ) || (originc > DataColumns))
    originc = ...
       input('>> Your input is not a valid column number. Please try again: ');
end

origin = [originr,originc];
```

retrieveRadii.m

```
function [radii] = retrieveRadii(a)
% radii = retrieveRadii(a) returns three radii [R0,R1,R2] (where R0<R2 are
% specified by the user and R1=(R0+R2)/2) for use in the algorithm
% Minktensor2D for a digitisation with lattice distance a.

% Ask user for lower radius for use in the algorithm
lowerRadius = input('>> Please enter a lower radius for use in the algorithm. ');

% Continue to prompt for lowerRadius if user does not input a positive
% number greater than a/sqrt(2)
while ( ~isnumeric(lowerRadius) ) || ( isempty(lowerRadius) ) || ...
   ( lowerRadius <= (a/sqrt(2)) )
    disp(['>> Your input is not a positive number greater than the lattice ',...
   'distance divided by the square root of two.'])
    lowerRadius = input('Please try again: ');
end
```

```
% Ask user for upper radius for use in the algorithm
disp('>> Please enter the upper radius for use in the algorithm.')
upperRadius = input('(Number must not exceed the reach of the object!) ');

% Continue to prompt for upperRadius if user does not input a positive
% number greater than lowerRadius
while ( ~isnumeric(upperRadius) ) || ( isempty(upperRadius) ) || ...
   ( upperRadius <= lowerRadius ) || ( isnan(upperRadius) ) || ...
   ( isinf(upperRadius) )
    disp(['>> Your input is not a positive number (greater than the lower '...
    'radius chosen). '])
    upperRadius = input('Please try again: ');
end

% Scale radii with lattice distance
upperRadius = upperRadius/a;
lowerRadius = lowerRadius/a;

radii = linspace(lowerRadius,upperRadius,3);
```

computeVoronoiCells.m

```
function [outerPoints,innerPoints,VoronoiIndices,VoronoiVertices] = ...
   computeVoronoiCells(digitisation,r,origin)
% [outerPoints,innerPoints,VoronoiIndices,VoronoiVertices] =
% computeVoronoiCells(digitisation,r,origin) returns an array outerPoints
% of coordinates of the sampling points in digitisation which have a
% non-trivial Voronoi cell; an array innerPoints of coordinates of inner
% points in digitisation, which have trivial Voronoi cell; an array
% VoronoiVertices of vertices of Voronoi cells of points in
% [outerPoints;innerPoints], and a cell array VoronoiIndices whose n'th row
% correspondonds to the indices of the Voronoi vertices in VoronoiVertices
% of the n'th point in [outerPoints;innerPoints].

% Necessary .m-files:
% convertDigitisationToCoordinates.m

% Convert digitisation to points in the plane
[innerPoints,outerPoints,VoronoiRelevant] = ...
   convertDigitisationToCoordinates(digitisation,origin);

if isnan(innerPoints)
    allPoints = outerPoints;
else
    allPoints = [innerPoints;outerPoints];
end

numberOfPoints = size(allPoints,1);

% Find the extremal coordinates of the digitisation
xMin = min(allPoints(1:numberOfPoints,1));
xMax = max(allPoints(1:numberOfPoints,1));
yMin = min(allPoints(1:numberOfPoints,2));
yMax = max(allPoints(1:numberOfPoints,2));

xMid = (xMax - xMin)/2 + xMin;
yMid = (yMax - yMin)/2 + yMin;

% We will insert 'border points' far away from the sampling points in order
% to obtain bounded Voronoi cells.
% The distance from digitisation to border points is set to three times the
% maximal radius. Thus the relevant parts of the Voronoi cells of the
% digitisation will not be affected.

borderPointDistance = 3*ceil(r);

% Create border points
borderPoints = zeros(8,2);
borderPoints(1,:) = [xMin-borderPointDistance,yMid];
borderPoints(2,:) = [xMid,yMax+borderPointDistance];
borderPoints(3,:) = [xMax+borderPointDistance,yMid];
borderPoints(4,:) = [xMid,yMin-borderPointDistance];
borderPoints(5,:) = [xMin-borderPointDistance,yMin-borderPointDistance];
borderPoints(6,:) = [xMin-borderPointDistance,yMax+borderPointDistance];
borderPoints(7,:) = [xMax+borderPointDistance,yMin-borderPointDistance];
borderPoints(8,:) = [xMax+borderPointDistance,yMax+borderPointDistance];
```

```
% Create the Voronoi vertices and indices for all relevant sampling points
% of the digitisation
if isnan(VoronoiRelevant)

    VoronoiRelevantPoints = [];

% If there are Voronoi relevant inner points, we use them when computing
% the Voronoi cells
else

   VoronoiRelevantPoints = VoronoiRelevant;

end

% The Voronoi diagram is now computed for all relevant points
[VoronoiVertices,excessVoronoiIndices] = ...
   voronoin([outerPoints;borderPoints;VoronoiRelevantPoints]);

% Clean up VoronoiIndices by removing indices correponding to artificial
% points
numberOfRelevantIndices = size(outerPoints,1);
VoronoiIndices = excessVoronoiIndices(1:numberOfRelevantIndices,:);
```

convertDigitisationToCoordinates.m

```
function [innerPoints,outerPoints,VoronoiRelevant] = ...
   convertDigitisationToCoordinates(digitisation,origin)
% [innerPoints,outerPoints,VoronoiRelevant] =
% convertDigitisationToCoordinates(digitisation,origin) returns an array
% outerPoints of coordinates of the sampling points in digitisation which
% have a non-trivial Voronoi cell; an array innerPoints of coordinates of
% inner points in digitisation, which have trivial Voronoi cell, and an
% array of those points in digitisation which define the non-trivial
% Voronoi cells of the outer points

digitisation = flipud(digitisation); % Matrix indices are read from the
% upper left corner whereas planar coordinates are read from the bottom
% left corner, so we flip digitisation horisontally

numberOfRows = size(digitisation,1);
numberOfColumns = size(digitisation,2);

originx = origin(2) - 1;
originy = numberOfRows - origin(1);

% Prepare arrays for the three types of sampling points and allocate space
% for the maximal number of rows

arrayInnerPoints = zeros(numberOfRows*numberOfColumns,2);
arrayOuterPoints = arrayInnerPoints;
arrayVoronoiRelevant = arrayInnerPoints;

% Counters for the number of sampling points of each type
countInner = 0;
countOuter = 0;
countVoronoi = 0;

for iii = 1:(numberOfRows)
    for jjj = 1:(numberOfColumns)

        x = digitisation(iii,jjj);

        % Only 1's are converted to coordinates
        if x

            notBorder = (iii-1)*(jjj-1)*(iii-numberOfRows)*(jjj-numberOfColumns);
            % false if x lies in first or last column and/or row

            if notBorder

                xr = digitisation(iii+1,jjj); % point to the right of x
                xl = digitisation(iii-1,jjj); % point to the left of x
                xu = digitisation(iii,jjj+1); % point above x
                xd = digitisation(iii,jjj-1); % point below x

                inner = xr*xl*xu*xd; % true if x is an inner point
                if inner

                    countInner = countInner + 1;
                    arrayInnerPoints(countInner,:) = [jjj-1,iii-1];

                    innerBorder = ~((iii-2)*(jjj-2)*(iii-numberOfRows+1)*...
                        (jjj-numberOfColumns+1)); % true if x is a four-neighbour
                        % of a non-inner point. If so, x determines the Voronoi
                        % vertices of its neighbour

                    if innerBorder

                        countVoronoi = countVoronoi + 1;
                        arrayVoronoiRelevant(countVoronoi,:) = [jjj-1,iii-1];

                    else
                        % Four-neighbours of the four-neighbours of x
                        xrr = digitisation(iii+2,jjj);
                        xru = digitisation(iii+1,jjj+1);
                        xrd = digitisation(iii+1,jjj-1);
                        xll = digitisation(iii-2,jjj);
                        xlu = digitisation(iii-1,jjj+1);
                        xld = digitisation(iii-1,jjj-1);
                        xuu = digitisation(iii,jjj+2);
                        xdd = digitisation(iii,jjj-2);

                        % Check which of the four-neighbours of x are inner
                        % points
                        rightInner = xrr*xru*xrd;
                        leftInner = xll*xlu*xld;
                        upInner = xuu*xru*xlu;
                        downInner = xdd*xrd*xld;

                        relevant = ~(rightInner*leftInner*upInner*downInner);

                        if relevant
                            countVoronoi = countVoronoi + 1;
                            arrayVoronoiRelevant(countVoronoi,:) = [jjj-1,iii-1];
                        end
                    end
                else
                    countOuter = countOuter + 1;
                    arrayOuterPoints(countOuter,:) = [jjj-1,iii-1];
                end
            else
                countOuter = countOuter + 1;
                arrayOuterPoints(countOuter,:) = [jjj-1,iii-1];
            end
        end
    end
end

if countInner

    % innerPoints trims arrayInnerPoints
    innerPoints = arrayInnerPoints(1:countInner,:);
    innerPoints(:,1) = innerPoints(:,1) - originx;
    innerPoints(:,2) = innerPoints(:,2) - originy;

else
    innerPoints = NaN;

end

if countVoronoi

    % VoronoiRelevant trims arrayVoronoiRelevant
    VoronoiRelevant = arrayVoronoiRelevant(1:countVoronoi,:);
    VoronoiRelevant(:,1) = VoronoiRelevant(:,1) - originx;
    VoronoiRelevant(:,2) = VoronoiRelevant(:,2) - originy;

else

    VoronoiRelevant = NaN;

end
```

```
% outerPoints trims arrayOuterPoints
outerPoints = arrayOuterPoints(1:countOuter,:);
outerPoints(:,1) = outerPoints(:,1) - originx;
outerPoints(:,2) = outerPoints(:,2) - originy;
```

computeVolumeTensorEstimators.m

```
function [Phi200,Phi210,Phi220] = computeVolumeTensorEstimators(outerPoints,
    innerPoints,a)
% [Phi200,Phi210,Phi220] =
% computeVolumeTensorEstimators(outerPoints,innerPoints,a) returns
% estimators of the non-trivial volume tensors of rank at most two in
% dimension two based on arrays outerPoints and innerPoints of coordinates
% of all outer and inner sampling points of a ditigisation with lattice
% distance a

numberOfPoints = size(outerPoints,1) + size(innerPoints,1);
allPoints = [outerPoints;innerPoints];

% (r=0)-tensor estimator
sumxr0 = numberOfPoints;
Phi20 = a^2*sumxr0;

% (r=1)-tensor estimator
sumxr1 = sum(allPoints,1);
Phi21 = a^3*sumxr1;

% (r=2)-tensor estimator
sumxr21 = sum(allPoints.^2,1);
sumxr22 = sum(allPoints(:,1).*allPoints(:,2),1);
Phi22a = (a^4/2)*sumxr21;
Phi22b = (a^4/2)*sumxr22;
Phi22 = [Phi22a,Phi22b];

% Tensors very close to zero are rounded off
Phi200 = Phi20*(Phi20>1e-6);
Phi210 = [Phi21(1)*(abs(Phi21(1))>1e-6) ; Phi21(2)*(abs(Phi21(2))>1e-6)];
Phi220 = [Phi22(1)*(abs(Phi22(1))>1e-6) Phi22(3)*(abs(Phi22(3))>1e-6);...
   Phi22(3)*(abs(Phi22(3))>1e-6) Phi22(2)*(abs(Phi22(2))>1e-6)];
```

computeMinkowskiTensorEstimators.m

```
function [Phi000,Phi010,Phi020,Phi001,Phi011,Phi002,Phi100,Phi110,Phi120,...
   Phi101,Phi111,Phi102] = computeMinkowskiTensorEstimators(radii,...
   innerPoints,outerPoints,VoronoiIndices,VoronoiVertices,a)
% [Phi000,Phi010,Phi020,Phi001,Phi011,Phi002,Phi100,Phi110,Phi120,Phi101,...
% Phi111,Phi102] = computeMinkowskiTensorsEstimators(radii,...
% innerPoints,outerPoints,VoronoiIndices,VoronoiVertices,a,Phi20,Phi21,Phi22)
% returns all computed estimators (not including the volume tensors) of
% the Minkowski tensors of a set with positive reach given an array
% containing three radii below the reach of the set, arrays with the inner
% and outer points of the digitisation of the set, an array VoronoiVertices
% with the Voronoi vertices of the outer points as well as a cell array
% containing the indices of the vertices for each sampling point. Finally,
% the lattice distance a must be specified.

% Necessary .m-files:
% computeVoronoiTensorMeasures.m

% Compute the Voronoi tensor measures
[VR00,VR20,VR02,VR01,VR10,VR11] = ...
   computeVoronoiTensorMeasures(radii,innerPoints,outerPoints,...
   VoronoiIndices,VoronoiVertices);

Kappa = [1 2 pi 4*pi/3 pi^2/2]; % Volumes of the unit ball in Euclidean n-space
   % where n equals the column number of Kappa minus 1

% Radii and Voronoi tensor measures for the non-scaled digitisation
radii = a*radii;

VR00 = a^(2)*VR00;
VR20 = a^(4)*VR20;
VR02 = a^(4)*VR02;
VR01 = a^(3)*VR01;
VR10 = a^(3)*VR10;
VR11 = a^(4)*VR11;
```

```
% Inverses of the constant matrices for s=0,1,2
M0 = [Kappa(1) Kappa(2)*radii(1) Kappa(3)*radii(1)^2;...
      Kappa(1) Kappa(2)*radii(2) Kappa(3)*radii(2)^2;...
      Kappa(1) Kappa(2)*radii(3) Kappa(3)*radii(3)^2];

M1 = [Kappa(2)*radii(1) Kappa(3)*radii(1)^2 Kappa(4)*radii(1)^3;...
      Kappa(2)*radii(2) Kappa(3)*radii(2)^2 Kappa(4)*radii(2)^3;...
      Kappa(2)*radii(3) Kappa(3)*radii(3)^2 Kappa(4)*radii(3)^3];

M2 = [Kappa(3)*radii(1)^2 Kappa(4)*radii(1)^3 Kappa(5)*radii(1)^4;...
      Kappa(3)*radii(2)^2 Kappa(4)*radii(2)^3 Kappa(5)*radii(2)^4;...
      Kappa(3)*radii(3)^2 Kappa(4)*radii(3)^3 Kappa(5)*radii(3)^4];

% Solving the systems of linear equations for the Minkowski tensors
% r = s = 0
b00 = [VR00(1) ; VR00(2) ; VR00(3)];
Phi00 = M0\b00;

Phi100 = Phi00(2)*(abs(Phi00(2))>1e-6);
Phi000 = Phi00(3)*(abs(Phi00(3))>1e-6);

% r = 2, s = 0
b201 = [VR20(1,1) ; VR20(1,2) ; VR20(1,3)];
EtPhi20 = M0\b201;
b202 = [VR20(2,1) ; VR20(2,2) ; VR20(2,3)];
ToPhi20 = M0\b202;
b203 = [VR20(3,1) ; VR20(3,2) ; VR20(3,3)];
TrePhi20 = M0\b203;

Phi120 = ...
   [EtPhi20(2)*(abs(EtPhi20(2))>1e-6) TrePhi20(2)*(abs(TrePhi20(2))>1e-6);...
    TrePhi20(2)*(abs(TrePhi20(2))>1e-6) ToPhi20(2)*(abs(ToPhi20(2))>1e-6)]/2;
Phi020 = ...
   [EtPhi20(3)*(abs(EtPhi20(3))>1e-6) TrePhi20(3)*(abs(TrePhi20(3))>1e-6);...
    TrePhi20(3)*(abs(TrePhi20(3))>1e-6) ToPhi20(3)*(abs(ToPhi20(3))>1e-6)]/2;

% r = 0, s = 2
b021 = [VR02(1,1) ; VR02(1,2) ; VR02(1,3)];
EtPhi02 = M2\b021;
b022 = [VR02(2,1) ; VR02(2,2) ; VR02(2,3)];
ToPhi02 = M2\b022;
b023 = [VR02(3,1) ; VR02(3,2) ; VR02(3,3)];
TrePhi02 = M2\b023;

Phi102 = ...
   [EtPhi02(2)*(abs(EtPhi02(2))>1e-6) TrePhi02(2)*(abs(TrePhi02(2))>1e-6);...
    TrePhi02(2)*(abs(TrePhi02(2))>1e-6) ToPhi02(2)*(abs(ToPhi02(2))>1e-6)]/2;
Phi002 = ...
   [EtPhi02(3)*(abs(EtPhi02(3))>1e-6) TrePhi02(3)*(abs(TrePhi02(3))>1e-6);...
    TrePhi02(3)*(abs(TrePhi02(3))>1e-6) ToPhi02(3)*(abs(ToPhi02(3))>1e-6)]/2;

% r = 0, s = 1
b011 = [VR01(1,1) ; VR01(1,2) ; VR01(1,3)];
EtPhi01 = M1\b011;
b012 = [VR01(2,1) ; VR01(2,2) ; VR01(2,3)];
ToPhi01 = M1\b011;

Phi101 = [EtPhi01(2)*(abs(EtPhi01(2))>1e-6) ; ToPhi01(2)*(abs(ToPhi01(2))>1e-6)];
Phi001 = [EtPhi01(3)*(abs(EtPhi01(3))>1e-6) ; ToPhi01(3)*(abs(ToPhi01(3))>1e-6)];

% r = 1, s = 0
b101 = [VR10(1,1) ; VR10(1,2) ; VR10(1,3)];
EtPhi10 = M0\b101;
b102 = [VR10(2,1) ; VR10(2,2) ; VR10(2,3)];
ToPhi10 = M0\b102;

Phi110 = [EtPhi10(2)*(abs(EtPhi10(2))>1e-6) ; ToPhi10(2)*(abs(ToPhi10(2))>1e-6)];
Phi010 = [EtPhi10(3)*(abs(EtPhi10(3))>1e-6) ; ToPhi10(3)*(abs(ToPhi10(3))>1e-6)];

% r = 1, s = 1
b111 = [VR11(1,1) ; VR11(1,2) ; VR11(1,3)];
EtPhi11 = M1\b111;
b112 = [VR11(2,1) ; VR11(2,2) ; VR11(2,3)];
ToPhi11 = M1\b112;
b113 = [VR11(3,1) ; VR11(3,2) ; VR11(3,3)];
TrePhi11 = M1\b113;
```

```
Phi111 = ...
   [EtPhi11(2)*(abs(EtPhi11(2))>1e-6) TrePhi11(2)*(abs(TrePhi11(2))>1e-6);...
    TrePhi11(2)*(abs(TrePhi11(2))>1e-6) ToPhi11(2)*(abs(ToPhi11(2))>1e-6)];
Phi011 = ...
   [EtPhi11(3)*(abs(EtPhi11(3))>1e-6) TrePhi11(3)*(abs(TrePhi11(3))>1e-6);...
    TrePhi11(3)*(abs(TrePhi11(3))>1e-6) ToPhi11(3)*(abs(ToPhi11(3))>1e-6)];
```

computeVoronoiTensorMeasures.m

```
function [VR00,VR20,VR02,VR01,VR10,VR11] = computeVoronoiTensorMeasures(radii,...
   innerPoints,outerPoints,VoronoiIndices,VoronoiVertices)
% [VR00,VR20,VR02,VR01,VR10,VR11] = computeVoronoiTensorMeasures(radii,...
% innerPoints,outerPoints,VoronoiIndices,VoronoiVertices)
% returns the Voronoi tensor measures of a set with positive reach given an
% array containing three radii below the reach of the set and greater than
% 1/sqrt(2), arrays with the inner and outer points of the digitisation of
% the set, a cell array VoronoiVertices with the Voronoi vertices of the
% outer points as well as an array containing the indices of the vertices
% for each sampling point.

% Necessary .m-files:
% verticesToCounterclockwiseOrder.m
% intersectLinesegmentCircle.m
% RBoundedVoronoiVertex.m
% computeTensorMeasureComponents.m

numberOfInnerPoints = size(innerPoints,1)*(~isnan(innerPoints(1)));
numberOfOuterPoints = size(outerPoints,1);

% Arrays containing the entries for the Voronoi tensor measures are
% prepared. The three columns correspond to the three radii

VR00 = [0 0 0];     % r=0, s=0 (0-tensor)
VR20 = zeros(3,3);  % r=2, s=0 (2-tensor)
VR02 = VR20;        % r=0, s=2 (2-tensor)
VR01 = zeros(2,3);  % r=0, s=1 (1-tensor)
VR10 = VR01;        % r=1, s=0 (1-tensor)
VR11 = VR20;        % r=0, s=1 (1-tensor)

% First, contributions from the inner points to all Voronoi tensor measures
% are calculated
if numberOfInnerPoints
    for iii = 1:numberOfInnerPoints

        x = innerPoints(iii,:);
        x1 = x(1);
        x2 = x(2);

        for kkk = 1:3
            VR00(kkk) = VR00(kkk) + 1;
            VR20(:,kkk) = VR20(:,kkk) + [x1^2 ; x2^2 ; x1*x2];
            VR02(:,kkk) = VR02(:,kkk) + [1 ; 1 ; 0]*(1/12);
            VR10(:,kkk) = VR10(:,kkk) + [x1 ; x2];
        end
    end
end

% Next, contributions from the outer points to all Voronoi tensor measures
% are calculated
for iii = 1:numberOfOuterPoints
    x = outerPoints(iii,:);

    indices = VoronoiIndices{iii}; % Indices of the Voronoi vertices of x
    vertices = VoronoiVertices(indices,:); % Voronoi vertices of x
    vertices = verticesToCounterclockwiseOrder(vertices,x); % Make sure the
    % Voronoi vertices are in counterclockwise order with respect to x
    nVertices = size(vertices,1);

    % Arrays containing the contributions to the entries of the Voronoi
    % tensor measures from x are prepared
    componentsVR00 = [0 0 0];
    componentsVR20 = zeros(3,3);
    componentsVR02 = componentsVR20;
    componentsVR01 = zeros(2,3);
    componentsVR10 = zeros(2,3);
    componentsVR11 = componentsVR20;
```

```
for kkk = 1:3
    R = radii(kkk);

    column = ones(nVertices,1);
    coordinateMatrix = [x(1)*column x(2)*column];

    vTypes = (R < sqrt(sum((vertices-coordinateMatrix).^2,2)));  % Assign a 1
    % to vertices that are further away from the current coordinate than R; 0
    % otherwise

    ConcatenatedVertices = [vertices(nVertices,:); vertices; vertices(1,:)];
    % Concatenated array with Vertices where the last row is copied to the
    % first row and the first row is copied to the last row

    % We now adjust all Voronoi vertices to vertices lying on the
    % R-bounded Voronoi cell. Additionally, we determine whether a
    % vertex is connected to its subsequent neighbour by an arc
    % (type 1) or a line (type 0).

    adjustedVertices = zeros(2*nVertices,3); % For saving the adjusted
    % vertices and their integration type. Each vertex is adjusted to at most
    % 2 new vertices
    vertexCounter = 0;

    for jjj = 1:nVertices
        % The types of the vertex and its two neighbours decide how the
        % vertex should be adjusted
        v = vertices(jjj,:); % v is the jjj'th Voronoi vertex of x
        vType = vTypes(jjj); % vType is the type of v

        % If v is inside the R-ball with centre x, it is a vertex of the
        % R-bounded Voronoi cell of type 0
        if ~vType
            vertexCounter = vertexCounter + 1;
            adjustedVertices(vertexCounter,1:2) = v;
            adjustedVertices(vertexCounter,3) = 0;

        else
            p = ConcatenatedVertices(jjj,:);
            s = ConcatenatedVertices(jjj+2,:);

            % Check whether sv and vp intersect the R-ball
            intersectionsPV = intersectLinesegmentCircle(v,p,x,R);
            intersectionsVS = intersectLinesegmentCircle(v,s,x,R);

            if (~intersectionsPV && ~intersectionsVS)   % If none of the line
            % segments intersect, the vertex is irrelevant
                continue

            else

                if intersectionsPV
                    v1 = RBoundedVoronoiVertex(p,v,x,R); % Adjust v to the
                    % point where pv intersects the R-ball

                    if isequal(v1,p)
                        continue
                    else
                        vertexCounter = vertexCounter + 1;
                        adjustedVertices(vertexCounter,1:2) = v1;
                        adjustedVertices(vertexCounter,3) = 1;
                    end
                end

                if intersectionsVS
                    v2 = RBoundedVoronoiVertex(s,v,x,R); % Adjust v to the
                    % point where vs intersects the R-ball

                    if isequal(v2,s)
                        continue
                    else

                        vertexCounter = vertexCounter + 1;
                        adjustedVertices(vertexCounter,1:2) = v2;
                        adjustedVertices(vertexCounter,3) = 0;
                    end
                    end

                end

            end

        end

        % Remove all zero-rows from adjustedVertices to obtain the vertices
        % of the R-bounded Voronoi cell of x
        RboundedVertices = adjustedVertices(1:vertexCounter,:);

        % Add the terms for this coordinate and radius to the Voronoi
        % tensor measures
        [componentsVR00(kkk),componentsVR20(:,kkk),componentsVR02(:,kkk),...
          componentsVR01(:,kkk),componentsVR10(:,kkk),componentsVR11(:,kkk)] =...
          computeTensorMeasureComponents(R,x,RboundedVertices);

    end

    VR00(1) = VR00(1) + componentsVR00(1);
    VR00(2) = VR00(2) + componentsVR00(2);
    VR00(3) = VR00(3) + componentsVR00(3);

    VR20(:,1) = VR20(:,1) + componentsVR20(:,1);
    VR20(:,2) = VR20(:,2) + componentsVR20(:,2);
    VR20(:,3) = VR20(:,3) + componentsVR20(:,3);

    VR02(:,1) = VR02(:,1) + componentsVR02(:,1);
    VR02(:,2) = VR02(:,2) + componentsVR02(:,2);
    VR02(:,3) = VR02(:,3) + componentsVR02(:,3);

    VR01(:,1) = VR01(:,1) + componentsVR01(:,1);
    VR01(:,2) = VR01(:,2) + componentsVR01(:,2);
    VR01(:,3) = VR01(:,3) + componentsVR01(:,3);

    VR10(:,1) = VR10(:,1) + componentsVR10(:,1);
    VR10(:,2) = VR10(:,2) + componentsVR10(:,2);
    VR10(:,3) = VR10(:,3) + componentsVR10(:,3);

    VR11(:,1) = VR11(:,1) + componentsVR11(:,1);
    VR11(:,2) = VR11(:,2) + componentsVR11(:,2);
    VR11(:,3) = VR11(:,3) + componentsVR11(:,3);

end
```

verticesToCounterclockwiseOrder.m

```
function [counterclockwiseVertices] = ...
   verticesToCounterclockwiseOrder(vertices,referencePoint)
% [counterClockwiseVertices] = ...
% verticesToCounterclockwiseOrder(vertices,referencePoint) returns an array
% of vertices of a polygon in counterclockwise order with respect to a
% reference point, referencePoint, given an array of vertices of the
% polygon. It is assumed that the polygon is closed, that the polygon is
% not self-intersecting or has holes, and that the vertices are not all
% collinear.

% Return an error if less than three vertices are provided
if all(size(vertices) < 3)
    error('Three points are needed to define polygon.')
end

% We check whether the vertices are ordered in rows. If not, we transpose

if size(vertices,1) < 3
    flip = 1;
    adjustedVertices = vertices';
    adjustedVertices(:,1) = adjustedVertices(:,1) - referencePoint(1);
    adjustedVertices(:,2) = adjustedVertices(:,2) - referencePoint(2);
else
    flip = 0;
    adjustedVertices = vertices;
    adjustedVertices(:,1) = adjustedVertices(:,1) - referencePoint(1);
    adjustedVertices(:,2) = adjustedVertices(:,2) - referencePoint(2);
end
```

```
count = 0;
n = size(adjustedVertices,1);

% Sum over the edges between all vertices
for iii = 1:(n-1)

    count = count + (adjustedVertices(iii+1,1)-adjustedVertices(iii,1))*...
       (adjustedVertices(iii+1,2)+adjustedVertices(iii,2));

end
% Also sum the edge from the last to the first point
count = count + (adjustedVertices(1,1)-adjustedVertices(n,1))*...
   (adjustedVertices(1,2)+adjustedVertices(n,2));

% Count is plus/minus twice the enclosed area, with a plus if the vertices
% are in clockwise order relative to the reference point. If so, we flip
% the sequence of the vertices
if count > 0
    adjustedVertices = flipud(adjustedVertices);
end

% Translate back to the original coordinate system
adjustedVertices(:,1) = adjustedVertices(:,1) + referencePoint(1);
adjustedVertices(:,2) = adjustedVertices(:,2) + referencePoint(2);

% If the vertices were transposed, transpose them back
if flip
    counterclockwiseVertices = adjustedVertices';
else
    counterclockwiseVertices = adjustedVertices;
end
```

intersectLinesegmentCircle.m

```
function [intersection] = intersectLinesegmentCircle(p1,p2,c,r)
% [intersections] = intersectLinesegmentCircle(p1,p2,c,r) determines
% whether the line segment from p1 to p2 intersects the circle with radius
% r and centre c. intersection equals the number of intersection points
% between the line segment and the circle which can be 0, 1, or 2

if isequal(p1,p2)

    error('The two points are equal. No line segment can be determined.')

end

p = p2-p1; % Vector from p1 to p2
np = norm(p);
punit = p/np;
l = c - p1; % Vector from p1 to the centre of the circle
nProj = dot(l,punit); % Length of the projection of l onto p

% Find the closest point on the line segment to the centre c

if nProj <= 0
    closest = p1;
elseif nProj > np
    closest = p2;
else
    Proj = punit*nProj;
    closest = Proj + p1;
end

distance = c - closest; % Vector from the closest point to c

nDistance = norm(distance);

% Find out whether the line segment and the circle intersect

if nDistance > r
    intersection = 0;  % The line segment does not intersect the circle
elseif nDistance < r
    intersection = 2;  % The line segment intersects the circle in two points
else
    intersection = 1;  % The line segment is tangent to the circle
end
```

RboundedVoronoiVertex.m

```
function [correctedVertex] = RBoundedVoronoiVertex(neighbourVertex,vertex,c,R)
% [correctedVertex] = RBoundedVoronoiVertex(neighbourVertex,vertex,c,r)
% returns the R-bounded Voronoi vertex corresponding to vertex by
% translating it to the nearest point of intersection between the circle
% with radius R and centre c and the line segment from neighbourVertex to vertex.

% Necessary .m-files:
% intersectLineCircle.m

[xIntersect,yIntersect] = intersectLineCircle(neighbourVertex(1),...
   neighbourVertex(2),vertex(1),vertex(2),c(1),c(2),R);
   % Coordinates of intersection of the line and the circle with given radius and
   % centre

[minimalDistance,index] = min([norm(vertex - [xIntersect(1),yIntersect(1)])...
   norm(vertex - [xIntersect(2),yIntersect(2)])]); % Find the index of the
   % intersection point closest to vertex

correctedVertex = [xIntersect(index) yIntersect(index)];
```

intersectLineCircle.m

```
function [xout,yout] = intersectLineCircle(point1x,point1y,point2x,point2y,...
   centrex,centrey,radius)
% [xout,yout] = intersectLineCircle(point1x,point1y,point2x,point2y,...
% centrex,centrey,radius)
% returns arrays xout and yout with the x- and y-coordinates of the
% intersection points of a line with a circle in the plane. If the line is
% tangent to the circle, the point of intersection is stored as two
% identical points; if there are no points of intersection, the arrays
% contain NaN as entries.
% Input arguments are coordinates (point1x,point1y) and (point2x,point2y)
% of two points on the line, the coordinates (centrex,centrey) of the
% centre of the circle, and the radius of the circle.

% Translate the coordinates to a system with the centre of the circle at
% the origin
x1 = point1x-centrex;
y1 = point1y-centrey;
x2 = point2x-centrex;
y2 = point2y-centrey;

% Compute the parameters of the system
dx = x2-x1;
dy = y2-y1;
dr = sqrt(dx^2+dy^2);
determinant = x1*y2-x2*y1;
discriminant = radius^2*dr^2-determinant^2;
epsilon = 1e-14; % Precision

% Compute the points of intersection (i.e. the solutions to the second
% order equation)
if discriminant > epsilon % Two points of intersection

    x = [(determinant*dy+sgn(dy)*dx*sqrt(discriminant))/dr^2 ; ...
        (determinant*dy-sgn(dy)*dx*sqrt(discriminant))/dr^2];
    y = [(-determinant*dx+abs(dy)*sqrt(discriminant))/dr^2 ; ...
        (-determinant*dx-abs(dy)*sqrt(discriminant))/dr^2];

elseif abs(discriminant) < epsilon % One point of intersection (the line
% is tangent to the circle)

    x = determinant*dy/dr^2*ones(2,1);
    y = -determinant*dx/dr^2*ones(2,1);

else % No points of intersection
    x = NaN*ones(2,1);
    y = NaN*ones(2,1);
end

% Translate the points of intersection back to a system with the centre of
% the circle at (centrex,centrey)
xout = x+centrex;
yout = y+centrey;
```

```
function s = sgn(x)
% sgn is a signum function which returns -1 for all negative numbers and 1
% otherwise
s = sign(sign(x)+0.5);
```

computeTensorMeasureComponents.m

```
function [componentVR00,componentVR20,componentVR02,componentVR01,...
   componentVR10,componentVR11] = ...
   computeTensorMeasureComponents(R,x,VoronoiVertices)
% [ComponentVR00,ComponentVR20,ComponentVR02,ComponentVR01,...
% ComponentVR10,ComponentVR11] = ...
% computeTensorMeasureComponents(r,x,vertices) returns the contributions to
% the Voronoi tensors measures of x when x has R-bounded Voronoi vertices
% VoronoiVertices

vertexCounter = size(VoronoiVertices,1);

x1 = x(1);
x2 = x(2);

% Functions for integrals involved in the Minkowski tensor measures in
% polar coordinates

ints211Ice = @(tMin,tMax,rMax) (1/16)*(rMax.^4).*(2*(tMax-tMin)+...
   sin(2*tMax)-sin(2*tMin)); % Primitive function for the coordinate (1,1) for
   % tensors with s=2 ('ice' region)
ints222Ice = @(tMin,tMax,rMax) (1/16)*(rMax.^4).*(2*(tMax-tMin)-...
   sin(2*tMax)+sin(2*tMin)); % Primitive function for the coordinate (2,2) for
   % tensors with s=2 ('ice' region)
ints212Ice = @(tMin,tMax,rMax) ((sin(tMax)^2 - sin(tMin)^2)*(rMax^4))/8;
% Primitive function for the coordinate (1,2)=(2,1) for tensors with s=2 ('ice'
% region)

ints11Ice = @(tMin,tMax,rMax) (rMax^3*(sin(tMax) - sin(tMin)))/3; % Primitive
% function for the coordinate (1,1) for tensors with s=1 ('ice' region)
ints12Ice = @(tMin,tMax,rMax) -(rMax^3*(cos(tMax) - cos(tMin)))/3; % Primitive
% function for the coordinate (2,1) for tensors with s=1 ('ice' region)

% If the R-bounded Voronoi cell is a disc, we simply integrate over this
if (~vertexCounter || vertexCounter == 1)

    volume = pi*R^2;

    % r = s = 0
    componentVR00 = volume;

    % r = 2, s = 0
    componentVR20 = [x1^2*volume ; x2^2*volume ; x1*x2*volume];

    % r = 0, s = 2
    componentVR02 = [ints211Ice(0,2*pi,R) ; ints222Ice(0,2*pi,R) ; ...
       ints212Ice(0,2*pi,R)];

    % r = 0, s = 1
    componentVR01 = [ints11Ice(0,2*pi,R) ; ints12Ice(0,2*pi,R)];

    % r = 1, s = 0
    componentVR10 = [x1*volume ; x2*volume];

    % r = 1, s = 1
    componentVR11 = [x1*componentVR01(1) ; x2*componentVR01(2) ; ...
       0.5*(x1*componentVR01(2) + x2*componentVR01(1))];

else

    concatenatedVertices = [VoronoiVertices; VoronoiVertices(1,:)];

    % We prepare the components for the Voronoi tensor measures

    componentVR00 = 0;                  % r = s = 0 tensor
    componentVR20 = [0 0 0]';           % r = 2, s = 0 tensor
    componentVR02 = componentVR20;      % r = 0, s = 2 tensor
    componentVR01 = [0 0]';             % r = 0, s = 1 tensor
    componentVR10 = componentVR01;      % r = 1, s = 0 tensor
    componentVR11 = componentVR20;      % r = 1, s = 1 tensor
```

```
ints0Ice = @(tMin,tMax,rMax) (1/2)*(rMax.^2).*(tMax-tMin); % Primitive
% function for tensors with s=0 ('ice' region)

% Functions for integrals involved in the Minkowski tensor measures in
% Cartesian coordinates of a rotated coordinate system

ints0Triangle = @(s0,t0,s1) (s0*t0)/2 - (t0*(s0 - s1))/2; % Primitive
% function for tensors with s=0 ('triangle' region)

ints2XXTriangle = @(s0,t0,s1,xs,xt) (s0*t0*(3*s0^2*xs^2 + ...
   3*s0*t0*xs*xt + t0^2*xt^2))/12 - (t0*(s0 - s1)*...
   (3*s0^2*xs^2 + 2*s0*s1*xs^2 + 3*s0*t0*xs*xt + s1^2*xs^2 + ...
   s1*t0*xs*xt + t0^2*xt^2))/12; % Primitive function for the coordinates
% (1,1) and (2,2) for tensors with s=2 ('triangle' region)

ints2XYTriangle = @(s0,t0,s1,xs,xt,ys,yt) (s0*t0*(6*s0^2*xs*ys + ...
   2*t0^2*xt*yt + 3*s0*t0*xs*yt + 3*s0*t0*xt*ys))/24 - (t0*(s0 - s1)*...
   (6*s0^2*xs*ys + 2*s1^2*xs*ys + 2*t0^2*xt*yt + 4*s0*s1*xs*ys + ...
   3*s0*t0*xs*yt + 3*s0*t0*xt*ys + s1*t0*xs*yt + s1*t0*xt*ys))/24;
% Primitive function for the coordinates (1,2)=(2,1) for tensors with s=2
% ('triangle' region)

ints1Triangle = @(s0,t0,s1,xs,xt) (s0*t0*(2*s0*xs + t0*xt))/6 - ...
   (t0*(s0 - s1)*(2*s0*xs + s1*xs + t0*xt))/6;  % Primitive function for the
% coordinates (1,1) and (2,1) for tensors with s=1 ('triangle' region)

% Coordinates are translated to a coordinate system with x as origin
s = concatenatedVertices(1,1:2)-x;

for jjj = 1:vertexCounter

    v = s;                                  % Current vertex
    s = concatenatedVertices(jjj+1,1:2)-x;  % Subsequent vertex

    vType = concatenatedVertices(jjj,3);    % Connection type of the current
    % vertex

    % The vertex connection type now determines which type of integral we
    % need to compute

    % Connection type 1 (arc)
    if vType

        % Find the angles of the two points
        if v(1) == 0
            vAngle = norm(sign(v(2)))*(pi/2 + (mod(sign(v(2))+2,3))*pi);
        else
            vAngle = atan(v(2)/v(1)) + (v(1)<0)*pi + (v(1)>0 && v(2)<0)*2*pi;
        end

        if s(1) == 0
            sAngle = norm(sign(s(2)))*(pi/2 + (mod(sign(s(2))+2,3))*pi);
        else
            sAngle = atan(s(2)/s(1)) + (s(1)<0)*pi + (s(1)>0 && s(2)<0)*2*pi;
        end

        % Adjustment of angles for the integration
        if vAngle > sAngle
            sAngle = sAngle + 2*pi;
        end

        % Computation of the Voronoi tensor measure contributions from
        % the current R-bounded Voronoi vertex

        integrals0Ice = ints0Ice(vAngle,sAngle,R); % Integral calculated for
        % all component terms where s = 0 for inner Voronoi vertices
        integrals11Ice = ints11Ice(vAngle,sAngle,R);
        integrals12Ice = ints12Ice(vAngle,sAngle,R);

        % r = s = 0
        componentVR00 = componentVR00 + integrals0Ice;

        % r = 2, s = 0
        componentVR20(1) = componentVR20(1) + x1^2*integrals0Ice;
        componentVR20(2) = componentVR20(2) + x2^2*integrals0Ice;
        componentVR20(3) = componentVR20(3) + x1*x2*integrals0Ice;
```

```
            % r = 0, s = 2
            componentVR02(1) = componentVR02(1) + ints211Ice(vAngle,sAngle,R);
            componentVR02(2) = componentVR02(2) + ints222Ice(vAngle,sAngle,R);
            componentVR02(3) = componentVR02(3) + ints212Ice(vAngle,sAngle,R);

            % r = 0, s = 1
            componentVR01(1) = componentVR01(1) + integrals11Ice;
            componentVR01(2) = componentVR01(2) + integrals12Ice;

            % r = 1, s = 0
            componentVR10(1) = componentVR10(1) + x1*integrals0Ice;
            componentVR10(2) = componentVR10(2) + x2*integrals0Ice;

            % r = 1, s = 1
            componentVR11(1) = componentVR11(1) + x1*integrals11Ice;
            componentVR11(2) = componentVR11(2) + x2*integrals12Ice;
            componentVR11(3) = componentVR11(3) + 0.5*(x1*integrals12Ice + ...
               x2*integrals11Ice);

        % Connection type 0 (line segment)

        else
            % Constants for computing the Voronoi tensor measure contribution
            % from the current R-bounded Voronoi vertex

            normV = norm(v);
            u = v/normV;
            v = [-u(2) u(1)];
            s0 = dot(s,u);
            t0 = dot(s,v);
            xs = dot([1 0],u);
            xt = dot([1 0],v);
            ys = dot([0 1],u);
            yt = dot([0 1],v);

            integrals0Triangle = ints0Triangle(s0,t0,normV);
            integrals1xTriangle = ints1Triangle(s0,t0,normV,xs,xt);
            integrals2yTriangle = ints1Triangle(s0,t0,normV,ys,yt);

            % r = s = 0
            componentVR00 = componentVR00 + integrals0Triangle;

            % r = 2, s = 0
            componentVR20(1) = componentVR20(1) + x1^2*integrals0Triangle;
            componentVR20(2) = componentVR20(2) + x2^2*integrals0Triangle;
            componentVR20(3) = componentVR20(3) + x1*x2*integrals0Triangle;

            % r = 0, s = 2
            componentVR02(1) = componentVR02(1) + ...
               ints2XXTriangle(s0,t0,normV,xs,xt);
            componentVR02(2) = componentVR02(2) + ...
               ints2XXTriangle(s0,t0,normV,ys,yt);
            componentVR02(3) = componentVR02(3) + ...
               ints2XYTriangle(s0,t0,normV,xs,xt,ys,yt);

            % r = 0, s = 1
            componentVR01(1) = componentVR01(1) + integrals1xTriangle;
            componentVR01(2) = componentVR01(2) + integrals2yTriangle;

            % r = 1, s = 0
            componentVR10(1) = componentVR10(1) + x1*integrals0Triangle;
            componentVR10(2) = componentVR10(2) + x2*integrals0Triangle;

            % r = 1, s = 1
            componentVR11(1) = componentVR11(1) + x1*integrals1xTriangle;
            componentVR11(2) = componentVR11(2) + x2*integrals2yTriangle;
            componentVR11(3) = componentVR11(3) + ...
               0.5*(x1*integrals2yTriangle + x2*integrals1xTriangle);

        end
    end
end
```

Bibliography

[1] Adrian Baddeley and Eva B.V. Jensen. *Stereology for statisticians.* CRC Press, 2004.

[2] Adrian Baddeley, Ege Rubak, and Rolf Turner. Spatial point patterns: Methodology and applications with R. URL http://www.crcpress.com/Spatial-Point-Patterns-Methodology-and-Applications-with-R/Baddeley-Rubak-Turner/9781482210200/.

[3] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22 (4):469–83, 1996.

[4] Claus Beisbart, Marconi S. Barbosa, Herbert Wagner, and Luciano da F. Costa. Extended morphometric analysis of neuronal cells with Minkowski valuations. *The European Physical Journal B-Condensed Matter and Complex Systems*, 52(4):531–46, 2006.

[5] Frédéric Chazal and David Cohen-Steiner. A condition for isotopic approximation. *Graphical Models*, 67(5):390–404, 2005.

[6] Sabrina Tang Christensen. Minktensor2D for MATLAB. URL https://gist.github.com/SabrinaGeggie/08634a26664ed7e17344422d42293a6b.

[7] Sabrina Tang Christensen and Andrew Allan du Plessis. Reconstruction of topology from digitisations of r-regular sets in $\mathbb{R}^3$. In preparation, 2016.

[8] Sabrina Tang Christensen and Markus Kiderlen. Comparison of two global digital algorithms for Minkowski tensor estimation. *CSGB Research Reports*, (10), July 2016.

[9] Ulrich Clarenz, Martin Rumpf, and Alexandru Telea. Robust feature detection and local classification for surfaces based on moment analysis. *IEEE Transactions on Visualization and Computer Graphics*, 10(5):516–24, 2004.

[10] Andrew Allan du Plessis. On the reconstruction of r-regular curves in 2D. In preparation, 2016.

[11] Andrew Allan du Plessis. Notes on r-regularity. Notes handed out from supervisor to the author, 2016.

[12] Pedro Duarte and Maria Joana Torres. Smoothness of boundaries of regular sets. *Journal of mathematical imaging and vision*, 48:106–13, 2014.

[13] Herbert Federer. Curvature measures. *Transactions of the American Mathematical Society*, 93(3):418–91, December 1959.

[14] Richard Guy. *Unsolved problems in number theory*, volume 1. Springer Science & Business Media, 2013.

[15] Allen Hatcher. *Algebraic topology.* Cambridge University Press, 2001.

[16] Morris W. Hirsch. *Differential Topology.* Springer-Verlag, 1976.

[17] Heinz Hopf. Vektorfelder in n-dimensionalen Mannigfaltigkeiten. *Matematische Annalen*, 96:225–50, 1927.

[18] Daniel Hug and Rolf Schneider. Local tensor valuations. *Geometric and Functional Analysis*, 24(5):1516–64, 2014.

[19] Daniel Hug, Markus Kiderlen, and Anne Marie Svane. Voronoi-based estimation of Minkowski tensors from finite point samples. *CSGB Research Reports*, (06), April 2016.

[20] Martin N. Huxley. Integer points, exponential sums and the Riemann zeta function. *Number theory for the millennium, II (Urbana, IL, 2000)*, pages 275–90, 2002.

[21] Sebastian C. Kapfer, Walter Mickel, Fabian M. Schaller, Markus Spanner, Christian Goll, Tomoaki Nogawa, Nobuyasu Ito, Klaus Mecke, and Gerd E. Schröder-Turk. Local anisotropy of fluids using Minkowski tensors. *Journal of Statistical Mechanics: Theory and Experiment*, 2010(11):P11010, 2010.

[22] Reinhard Klette and Azriel Rosenfeld. *Digital geometry: Geometric methods for digital picture analysis*. Elsevier, 2004.

[23] John M. Lee. *Introduction to Smooth Manifolds*, volume 218. Springer, second edition, 2013.

[24] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. 21(4):163–9, 1987.

[25] MATLAB. *Version 9.0.0.341360, Release 2016a*. The MathWorks, Inc., Natick, Massachusetts, United States.

[26] Quentin Merigot, Maks Ovsjanikov, and Leonidas Guibas. Voronoi-based curvature and feature estimation from point clouds. *IEEE Transactions on Visualization and Computer Graphics*, 17(6):743–756, 2011.

[27] John W. Milnor. *Topology from the differentiable viewpoint*. The University Press of Virginia, Charlottesville, 1965.

[28] Sebastián Montiel, Antonio Ros, and Donald G Babbitt. *Curves and surfaces*, volume 69. American mathematical society Providence, 2005.

[29] Joachim Ohser and Frank Mücklich. *Statistical Analysis of Microstructures in Materials Sciences*. Wiley, New York, 2000.

[30] Theo Pavlidis. *Algorithms for graphics and image processing*. Springer-Verlag Berlin – Heidelberg, 1982.

[31] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL https://www.R-project.org/.

[32] Rolf Schneider. *Convex bodies: The Brunn-Minkowski Theory*. Cambridge University Press, second expanded edition edition, 2014.

[33] Rolf Schneider and Wolfgang Weil. *Stochastic and Integral Geometry*. Springer-Verlag Berlin Heidelberg, 2008.

[34] G.E. Schröder-Turk, W. Mickel, S.C. Kapfer, M.A. Klatt, F.M. Schaller, M.J.F. Hoffmann, N. Kleppmann, P. Armstrong, A. Inayat, D. Hug, et al. Minkowski tensor shape analysis of cellular, granular and porous structures. *Advanced Materials*, 23 (22-23):2535–53, 2011.

[35] Gerd E. Schröder-Turk, Walter Mickel, Matthias Schröter, Gary W. Delaney, Mohammad Saadatfar, Tim J. Senden, Klaus Mecke, and Tomaso Aste. Disordered spherical bead packs are anisotropic. *EPL (Europhysics Letters)*, 90(3):34001, 2010.

[36] Jean Serra. *Image analysis and mathematical morphology*, volume 1. Academic Press, Inc., 1982.

[37] Peer Stelldinger and Ullrich Köthe. Towards a general sampling theory for shape preservation. *Image and Vision Computing*, 23(2):237–48, 2005.

[38] Peer Stelldinger, Longin Jan Latecki, and Marcelo Siqueira. Topological equivalence between a 3D object and the reconstruction of its digital image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):126–40, 2007.

[39] Anne Marie Svane. Estimation of Minkowski tensors from digital grey-scale images. *Image Anal. Stereol.*, 34:51–61, 2015.

[40] Martina Zähle. Integral and current representation of Federer's curvature measures. *Archiv der Mathematik*, 46(6):557–567, 1986.

INDEX