A new algorithmic approach to the computation of Minkowski functionals of germ-grain models

Simone Klenk[1], **Volker Schmidt**[1], **Evgueni Spodarev**[1]

March 17, 2004

Abstract. A new fast algorithm is proposed for simultaneous computation of all Minkowski functionals (or, equivalently, intrinsic volumes) of sets from the convex ring in $\mathbb{R}^d$ discretized with respect to a given rectangular lattice. For this purpose, a certain kind of polyhedral approximation is used to reconstruct their boundary structure. Furthermore, two efficient edge–corrected algorithms are given in order to estimate the specific intrinsic volumes of discretized stationary random closed sets in $\mathbb{R}^d$ from a single realization. For the planar case $d = 2$, the performance and precision of these algorithms is studied on various examples ranging from particular polyconvex sets to samples from Boolean models. The algorithms are implemented in Java for two different adjacency systems. Numerical experiments show that they are not sensitive to anisotropies. Comparisons to other related methods known in the literature are also provided.

Keywords. Stochastic geometry, random closed set, stationarity, Boolean model, discretization, adjacency system, volume fraction, specific boundary area, Euler–Poincaré characteristic, nonparametric estimation.

1 Introduction

Morphological characteristics of binary images such as volume, boundary area, curvature and connectivity number known as *Minkowski functionals* or *intrinsic volumes* are of great importance in statistical image analysis. They characterize the geometric structure of images and provide the basis for image modelling and classification. Mathematically, binary images can be thought of as continuous sets discretized with respect to a certain rectangular lattice. Furthermore, continuous geometric objects must be represented as ensembles of pixels on discrete grids in order to be processed by computers. Thus, the problem of fast, precise and robust computation of morphological characteristics of discretized sets has been lively discussed in the mathematical literature of the last decade; see e.g. [8], [9], [10], [17].

In the present paper, a new approach to the computation of Minkowski functionals for unions of convex sets (or, equivalently, *polyconvex sets*) is described. It leads to a fast and efficient algorithm that has the

[1] Universität Ulm, Abteilung Stochastik, D-89069 Ulm, Germany

following advantages. First, unlike other related methods, it computes all $d+1$ Minkowski functionals of a polyconvex set in $\mathbb{R}^d$ simultaneously. Thus, separate algorithms for the computation of each Minkowski functional are superfluous. Second, our algorithm shows high computational precision. Its edge–corrected counterparts for stationary random closed sets are on average more precise than other related methods described e.g. in [8]. Third, it is fast requiring only one single scan of the image. Finally, our algorithms are quite flexible since they depend on d free parameters, which have the meaning of dilation radii.

Notice that the theoretical background for our algorithms is provided by the recent paper [11] where tools of convex and stochastic geometry have been used to introduce new nonparametric estimators for specific intrinsic volumes of random closed sets (RACS) and to study their (asymptotic) statistical properties. This estimation method for deterministic and random polyconvex sets is briefly sketched in Section 2.1. The discretization of such sets with respect to a given rectangular lattice as well as their polyhedral approximation based on adjacency systems is discussed in Section 2.2. In connection with this, we suppose that the input image is given by a finite number of points and no other extra information about the image structure is available.

In Section 3, the corresponding computational algorithms are introduced for discretized deterministic and random polyconvex sets. In the deterministic case, a polyhedral approximation of the underlying continuous set is constructed and the intrinsic volumes of this approximated set are computed; see Section 3.1. If the input image is a discretized realization of a stationary RACS from the extended convex ring, then the edge–corrected algorithms presented in Section 3.2 are used to compute the estimators of specific intrinsic volumes. Some ideas for an appropriate choice of the dilation radii are discussed in Section 3.3.

The general framework of Section 3, which does not depend on the dimension d, is specified in Section 4 for the planar case $d = 2$ where weight functions are explicitly determined for each possible configuration of neighborhood pixels of a given boundary pixel of the input set. The three-dimensional case $d = 3$ will be discussed separately in a forthcoming paper.

For the planar case $d = 2$, the algorithmic approach explained in Section 4 has been implemented in Java and integrated into the GeoStoch library; see [2]. The code has been tested on various synthetic (i.e., simulated) images. In Section 5, the results of numerical experiments are discussed and compared to those of conventional computation methods such as the marching cube algorithm for deterministic polyconvex sets as well as the algorithms based on Crofton's formula, which are given in [8] for samples from stationary RACS. For test purposes, we used polyconvex sets with known Minkowski functionals as well as stationary germ–grain models with realizations from the extended convex ring and with known specific intrinsic volumes. Special attention is paid to anisotropic Boolean models with simple convex primary grains such as segments and rectangles.

2 Preliminaries

In this section, the theoretical background as introduced in [11] is briefly outlined, on which the practical computation of (specific) intrinsic volumes for deterministic and stationary random closed sets is based. Its algorithmic counterpart is discussed in Section 3. In the second part of the present section, a short introduction to discretization of continuous sets and their polyhedral approximation is given. A more detailed account of polyhedral approximation by means of adjacency systems can be found e.g. in [9].

2.1 Intrinsic volumes

2.1.1 Intrinsic volumes of deterministic sets

Let $\mathcal{K}$ be the family of all convex bodies in $\mathbb{R}^d$. Let $\mathcal{R}$ be the convex ring, i.e. the set of all finite unions of convex bodies in $\mathbb{R}^d$. By $V_d(K)$ we denote the volume of $K \in \mathcal{K}$. Let $o \in \mathbb{R}^d$ be the origin in $\mathbb{R}^d$ and $B_r(x)$ the closed ball in $\mathbb{R}^d$ with radius $r > 0$ and center at $x \in \mathbb{R}^d$. For each $j = 0, \dots, d$, nonnegative functionals $V_j : \mathcal{K} \to [0, \infty)$ exist such that for each $r > 0$ the volume $V_d(K \oplus B_r(o))$ of the so–called *parallel set* $K \oplus B_r(o)$ of any $K \in \mathcal{K}$ is given by *Steiner's formula*

$$V_d(K \oplus B_r(o)) = \sum_{j=0}^{d} r^{d-j} k_{d-j} V_j(K)\,, \tag{2.1}$$

where k_j is the volume of the unit ball in $\mathbb{R}^j$, $j = 0, \dots, d$. The functionals $V_j : \mathcal{K} \to [0, \infty)$ are called *intrinsic volumes.* Numbered in reverse order and properly normed, these functionals are also known as *Minkowski functionals* $W_j : \mathcal{K} \to [0, \infty)$, where $W_j(K) = k_j \, V_{d-j}(K) \,/\, \binom{d}{j}$ for any $K \in \mathcal{K}$. Later on, we shall use only intrinsic volumes because of convenience of notation.

Notice that for each $j = 0, \dots, d$, there exists a unique additive extension of the functional V_j to the convex ring $\mathcal{R}$, where this extension is given by the usual inclusion–exclusion technique; see e.g. Formula (2.2) in [11]. Our algorithm for the computation of intrinsic volumes is based on the following explicit extension of Steiner's formula to $\mathcal{R}$ proposed by Schneider (see e.g. [12]). For any $r > 0$ and $K \in \mathcal{R}$, we have

$$\rho_r(K) = \sum_{j=0}^{d-1} r^{d-j} k_{d-j} V_j(K)\,, \tag{2.2}$$

where the functional $\rho_r : \mathcal{R} \to \mathbb{R}$ is given by

$$\rho_r(K) = \int_{\mathbb{R}^d} I_r(K, x)\, dx \qquad \text{with } \; I_r(K, x) = \textstyle\sum_{q \neq x} J\left(K \cap B_r(x), q, x\right) \tag{2.3}$$

for each $K \in \mathcal{R}$ and the so-called *index function* $J\left(K \cap B_r(x), q, x\right)$ is defined by

$$J(K, q, x) = \begin{cases} 1 - \lim\limits_{\delta \to +0} \lim\limits_{\varepsilon \to +0} V_0\left(K \cap B_{|x-q|-\varepsilon}(x) \cap B_\delta(q)\right), & \text{if } q \in K, \\ 0, & \text{otherwise,} \end{cases} \tag{2.4}$$

for any $K \in \mathcal{R}$ and $q, x \in \mathbb{R}^d$. Notice that the sum in (2.3) extends only over a finite set of boundary points q of K; see [12]. Thus, for pairwise different radii $r_0, \dots, r_{d-1} > 0$, we get d equations of the type (2.2) which form the following system of linear equations

$$\rho_{r_0}(K) = \sum_{j=0}^{d-1} r_0^{d-j} k_{d-j} V_j(K)\,, \quad \dots\,, \quad \rho_{r_{d-1}}(K) = \sum_{j=0}^{d-1} r_{d-1}^{d-j} k_{d-j} V_j(K)\,, \tag{2.5}$$

or, equivalently, in matrix form we have $A_{r_0\dots r_{d-1}}V(K)=C(K)$, where

$$A_{r_0\dots r_{d-1}}=\begin{pmatrix} r_0^d k_d & r_0^{d-1}k_{d-1} & \dots & r_0^2 k_2 & r_0 k_1 & 0\\ r_1^d k_d & r_1^{d-1}k_{d-1} & \dots & r_1^2 k_2 & r_1 k_1 & 0\\ \dots & \dots & \dots & \dots & \dots & \dots\\ r_{d-1}^d k_d & r_{d-1}^{d-1}k_{d-1} & \dots & r_{d-1}^2 k_2 & r_{d-1} k_1 & 0\\ 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix}, \tag{2.6}$$

$V(K)=(V_0(K),\dots,V_{d-1}(K),V_d(K))^\top$ and $C(K)=(\rho_{r_0}(K),\dots,\rho_{r_{d-1}}(K),V_d(K))^\top$. Since the matrix $A_{r_0\dots r_{d-1}}$ in (2.6) is regular, a unique solution $V(K)$ of the above system of linear equations exists. Hence, using (2.5) one can compute the vector $V(K)$ of the intrinsic volumes of $K\in\mathcal{K}$, provided that the vector $C(K)$ is known.

2.1.2 Specific intrinsic volumes of stationary RACS

Let Ξ be a stationary RACS in $\mathbb{R}^d$ with realizations ξ from the extended convex ring $\mathcal{S}$, i.e., $\xi\cap K$ belongs to the usual convex ring $\mathcal{R}$ for any (compact and convex) $K\in\mathcal{K}$ and almost every realization ξ of Ξ. See e.g. [5], [13], [16] for definition and properties of stationary RACS.

For any set $K\subset\mathbb{R}^d$, the set of inner points of K in $\mathbb{R}^d$ is denoted by $\mathrm{int}(K)=K\setminus\partial K$, where ∂K is the boundary of K. Furthermore, for any nonempty $K\in\mathcal{R}$, let $N(K)=\min\{m\in\mathbb{N}: K=\bigcup_{i=1}^m K_i, K_i\in\mathcal{K}\}$ be the minimal number of convex components of the polyconvex set K, where we put $N(K)=0$ if $K=\emptyset$. If $E\,2^{N(\Xi\cap[0,1]^d)}<\infty$, then for any (monotonously increasing) sequence $\{W_n\}$ of compact and convex observation windows with $W_n=nK_0$ for some $K_0\in\mathcal{K}$ such that $V_d(K_0)>0$ and $o\in\mathrm{int}(K_0)$, the expectations $E\,V_j(\Xi\cap W_n)$ are well defined. Moreover, for each $j=0,\dots,d$, the limit

$$\overline{V}_j(\Xi)=\lim_{n\to\infty}\frac{E\,V_j(\Xi\cap W_n)}{V_d(W_n)} \tag{2.7}$$

exists and is called the j-th *specific intrinsic volume* of Ξ. Thus, (2.5) implies that the following system

$$A_{r_0,\dots,r_{d-1}}v=c \tag{2.8}$$

of $d+1$ linear equations holds, where $v=\big(\overline{V}_0(\Xi),\dots,\overline{V}_d(\Xi)\big)^\top$ is the vector of specific intrinsic volumes and

$$c=\left(\lim_{n\to\infty}\frac{E\,\rho_{r_0}(\Xi\cap W_n)}{V_d(W_n)},\dots,\lim_{n\to\infty}\frac{E\,\rho_{r_{d-1}}(\Xi\cap W_n)}{V_d(W_n)},\lim_{n\to\infty}\frac{E\,V_d(\Xi\cap W_n)}{V_d(W_n)}\right)^\top. \tag{2.9}$$

2.1.3 Unbiased and consistent estimators

The following approach to simultaneous estimation of all components of v has been introduced in [11]. For any $n\geqslant 1$ and $i=0,\dots,d-1$, let $\widehat{c}_{ni}=\rho_{r_i}(\Xi\cap W_n)\,\big/\,V_d(W_n)$, where

$$\rho_{r_i}(\Xi\cap W_n)=\int\limits_{\partial(\Xi\cap W_n)\oplus B_{r_i}(o)}\Bigg(\sum_{q\in\partial(\Xi\cap W_n)\cap B_{r_i}(x)\setminus\{x\}} J\big(\Xi\cap W_n\cap B_{r_i}(x),q,x\big)\Bigg)\,dx\,. \tag{2.10}$$

For $i = d$, we put $\widehat{c}_{nd} = V_d(\Xi \cap W_n) \big/ V_d(W_n)$. Hence, in view of (2.8) and (2.9), a natural estimator $\widehat{v}_n$ for v is given by $\widehat{v}_n = A^{-1}_{r_0 \ldots r_{d-1}} \widehat{c}_n$. Notice that $\widehat{c}_n = \big(\widehat{c}_{n0}, \ldots, \widehat{c}_{nd}\big)^\top$ and $\widehat{v}_n = \big(\widehat{v}_{n0}, \ldots, \widehat{v}_{nd}\big)^\top$ are asymptotically unbiased for c and v, respectively, as $n \to \infty$. Furthermore, a slightly modified estimator $\widetilde{c}_n$ for the vector c (and, in consequence, for v) has been considered in [11], which is based on the reduced observation window $W_n \ominus B_{r_i}(o)$. Namely, for each $i = 0, \ldots, d-1$ and any $n \geq 1$ such that $V_d(W_n \ominus B_{r_i}(o)) > 0$, let

$$\widetilde{c}_{ni} = \frac{\widetilde{\rho}_{n,r_i}(\Xi)}{V_d(W_n \ominus B_{r_i}(o))}\,, \tag{2.11}$$

where

$$\widetilde{\rho}_{n,r_i}(\Xi) = \int\limits_{W_n \ominus B_{r_i}(o)} \Big(\sum_{q \in \partial \Xi \cap B_{r_i}(x) \setminus \{x\}} J\big(\Xi \cap B_{r_i}(x), q, x\big)\Big)\, dx\,, \tag{2.12}$$

and put $\widetilde{c}_{nd} = \widehat{c}_{nd}$. Then, the edge–corrected estimators $\widetilde{c}_n = (\widetilde{c}_{n0}, \ldots, \widetilde{c}_{nd})^\top$ and $\widetilde{v}_n = \big(\widetilde{v}_{n0}, \ldots, \widetilde{v}_{nd}\big)^\top = A^{-1}_{r_0 \ldots r_{d-1}} \widetilde{c}_n$ are unbiased for c and v, respectively, where it is assumed that $V_d(W_n \ominus B_r(o)) > 0$ for $r = \max\{r_0, \ldots, r_{d-1}\}$. Moreover, it has been shown in [11] that the estimators $\widehat{c}_n$ and $\widetilde{c}_n$ as well as $\widehat{v}_n$ $\widetilde{v}_n$ possess nice asymptotic second–order properties, as $n \to \infty$. In particular, under some additional conditions, $\widetilde{c}_n, \widehat{c}_n$ and $\widetilde{v}_n, \widehat{v}_n$ are mean–square consistent for c and v, respectively. Furthermore, the asymptotic covariance matrix of these estimators can be determined, and a mean–square consistent estimator for the asymptotic covariance matrix can be constructed.

We also remark that besides the *minus sampling* considered in (2.11)–(2.12), there is still another type of edge correction, which leads to unbiased estimators if the observation windows W_n are parallelepipeds. For example, suppose that $W_n = [-na, na]^d$ for some $a > 0$ and consider the "right upper boundary" $\partial^+ W_n$ of W_n, where $\partial^+ W_n = \big\{x = (x_1, \ldots, x_d) \in W_n : \max_{1 \leqslant i \leqslant d} x_i = na\big\}$. For any $n \geqslant 1$ and $i = 0, \ldots, d-1$, let

$$\widehat{c}^{\,+}_{ni} = \frac{\rho_{r_i}(\Xi \cap W_n) - \rho_{r_i}(\Xi \cap \partial^+ W_n)}{V_d(W_n)}\,, \tag{2.13}$$

where $\rho_{r_i}(\Xi \cap W_n)$ and $\rho_{r_i}(\Xi \cap \partial^+ W_n)$ are given as described in (2.3) and (2.10). Then, putting $\widehat{c}^{\,+}_{nd} = \widehat{c}_{nd}$, the estimator $\widehat{c}^{\,+}_n = \big(\widehat{c}^{\,+}_{n0}, \ldots, \widehat{c}^{\,+}_{nd}\big)^\top$ is unbiased for c; see e.g. [13], [19]. Furthermore, this implies that $\widehat{v}^{\,+}_n = A^{-1}_{r_0 \ldots r_{d-1}} \widehat{c}^{\,+}_n$ is unbiased for v and, under some additional conditions, it can be shown that $\widehat{c}^{\,+}_n$ and $\widehat{v}^{\,+}_n$ possess similar asymptotic properties as the other two estimators $\widehat{c}_n, \widetilde{c}_n$ and $\widehat{v}_n, \widetilde{v}_n$ for c and v, respectively, considered above.

2.2 Discretization and polyhedral approximation

2.2.1 Rectangular lattices and adjacency systems

Consider the subset $\mathbb{L}^d = \{x \in \mathbb{R}^d : x = \sum_{i=1}^d \lambda_i u_i,\ \lambda_i \in \mathbb{Z}\}$ of the d-dimensional Euclidean space $\mathbb{R}^d$, where the vectors $u_1 = (\Delta_1, 0, \ldots, 0)^\top, \ldots, u_d = (0, \ldots, 0, \Delta_d)^\top$ form an orthogonal basis of $\mathbb{R}^d$ and $\Delta_1, \ldots, \Delta_d > 0$ are some constants. Notice that $\mathbb{L}^d$ is called a *rectangular lattice* in $\mathbb{R}^d$, where $\Delta_1, \ldots, \Delta_d$ are the lattice *spacings*. The *unit cell* of the lattice, i.e., the Minkowski sum of the half-open segments $[o, u_1), \ldots, [o, u_d)$ will be denoted by L. In the following, we only consider cubic lattices, i.e., we assume that $\Delta_1 = \Delta_2 = \ldots = \Delta_d = \Delta > 0$, i.e., $\mathbb{L}^d = \Delta\mathbb{Z}^d$. However, the results given below can easily be extended to the case of a general rectangular lattice.

For a polytope $P \subset \mathbb{R}^d$ and for each $k = 0, \ldots, d$, the set of *k-facets* of P is denoted by $\mathcal{F}^k(P)$. For instance, $\mathcal{F}^0(P)$ is the set of vertices, $\mathcal{F}^1(P)$ is the set of edges of P, and $\mathcal{F}^d(P)$ is the polytope P itself. A family of convex polytopes $G = \{P_1, P_2, \ldots\}$ with inner points is called a *tessellation* of $\mathbb{R}^d$ if $\text{int}(P_i) \cap \text{int}(P_j) = \emptyset$ for any $i \neq j$, if G is locally finite, i.e., the number of its *cells* $P_1, P_2, \ldots$ that intersect an arbitrary bounded subset of $\mathbb{R}^d$ is finite, and if G covers the whole $\mathbb{R}^d$, i.e., $\cup_{i=1}^{\infty} P_i = \mathbb{R}^d$. Notice that the intersection $P_i \cap P_j$ of two different cells P_i, P_j is either empty or a k-facet of these cells for some $k < d$. A tessellation $G = \{P_1, P_2, \ldots\}$ is called *admissible* with respect to $\mathbb{L}^d$ if $\mathcal{F}^0(P_i) \subset \mathbb{L}^d$ for each $i = 1, 2, \ldots$, i.e., the vertices of each cell P_i are lattice points, and if G is invariant with respect to lattice translations, i.e., $G + x = G$ for all $x \in \mathbb{L}^d$. In particular, this implies that the polytopes from G are subsets of the lattice cells. A *superposition* G of admissible tessellations $G_1, \ldots, G_m$, $m \in \mathbb{N}$ is defined as $G = \{P : P = Q_1 \cap \ldots \cap Q_m,\ Q_1 \in G_1, \ldots, Q_m \in G_m\}$. Notice that the superposition of admissible tessellations must not be admissible, due to possible existence of vertices that are not lattice points.

The *adjacency system* $\mathbb{F}(G)$ with respect to the tessellation G is defined as $\mathbb{F}(G) = \bigcup_{k=0}^{d} \mathcal{F}^k(G)$, where $\mathcal{F}^k(G) = \bigcup_{i=1}^{\infty} \mathcal{F}^k(P_i)$ for each $k = 0, \ldots, d$. Furthermore, the *neighborhood relation* γ for $\mathbb{F}(G)$ is given by the set γ of non–ordered pairs $\langle x, y\rangle$ of vertices $x, y \in \mathbb{L}^d$ such that

$$\gamma = \left\{ \langle x, y\rangle : x, y \in \mathcal{F}^0(\overline{L}), (x, y) \subset L, [x, y] \subset \mathcal{F}^1(G) \right\} , \tag{2.14}$$

where $\overline{L}$ denotes the topological closure of the lattice cell L, and (x, y), $[x, y]$ are the open and closed segments, respectively, connecting the vertices x and y. The *neighborhood graph* Γ with respect to $\mathbb{F}(G)$ is then defined as $\Gamma = \left(\mathbb{L}^d, \bigcup_{x\in\mathbb{L}^d}(x + \gamma)\right)$, where $\mathbb{L}^d$ is the set of nodes and $\bigcup_{x\in\mathbb{L}^d}(x + \gamma)$ the set of non–oriented edges. For any point $x \in \mathbb{L}^d$, its *neighborhood* $N_\Gamma(x)$ with respect to the graph Γ is introduced as $N_\Gamma(x) = \{y \in \mathbb{L}^d : \langle x, y\rangle \in \bigcup_{z\in\mathbb{L}^d}(z + \gamma)\} \cup \{x\}$. Finally, we mention two special adjacency systems that will be used later on. Let $G_{\min}$ be the tessellation that consists only of the lattice cells. Then the adjacency system $\mathbb{F}(G_{\min})$ is called the *minimum adjacency.* On the other hand, $\mathbb{F}(G_{\max})$ is called the *maximum adjacency* if $G_{\max}$ is the superposition of all admissible tessellations with respect to the lattice $\mathbb{L}^d$.

2.2.2 Polyhedral approximation of continuous sets

Consider a set $K \in \mathcal{R}$ where $K \subset W$ for some observation window W. In computer applications, one often deals with binary images that are represented by finite sets of black pixels on the white background. To comprehend this situation, we assume that instead of the set K, its *discretization* $K \cap \mathbb{L}^d$ with respect to the lattice $\mathbb{L}^d$ is given and any other extra information about K is not available. It is convenient to represent $K \cap \mathbb{L}^d$ as a binary image, i.e., as a finite set of "black" or foreground pixels $x \in K \cap \mathbb{L}^d$ on the "white" grid $\mathbb{L}^d$ (the so–called background). This means that we identify the set $K \cap \mathbb{L}^d$ with its indicator function $\mathbf{1}_{K\cap\mathbb{L}^d} : \mathbb{L}^d \to \{0, 1\}$, i.e., $\mathbf{1}_{K\cap\mathbb{L}^d}(x) = 1$ if $x \in K \cap \mathbb{L}^d$, and $\mathbf{1}_{K\cap\mathbb{L}^d}(x) = 0$ otherwise.

In order to compute the left–hand side of the linear equations considered in (2.5) for a set $K \in \mathcal{R}$ from its discretization $K \cap \mathbb{L}^d$, one should be able to determine the sum of index functions in (2.3). Due to the geometric nature of the index function that implicitly involves the boundary of K, one has to define the "boundary" of $K \cap \mathbb{L}^d$. In other words, the boundary of K has to be "reconstructed" or, better to say, approximated from its discretized version $K \cap \mathbb{L}^d$. One possible way to do that is the following polyhedral approximation of K by means of an adjacency system $\mathbb{F}(G)$. For any $K \in \mathcal{R}$, the set of those elements of the adjacency system $\mathbb{F}(G)$ whose vertices belong to $K \cap \mathbb{L}^d$ is denoted by $K \sqcap \mathbb{F}(G) = \bigcup_{k=0}^{d}(K \sqcap \mathcal{F}^k(G))$, where $K \sqcap \mathcal{F}^k(G) = \{P \in \mathcal{F}^k(G) : \mathcal{F}^0(P) \subset K \cap \mathbb{L}^d\}$. The *polyhedral approximation* $K_{\mathbb{F}(G)}$ of K with respect to the adjacency system $\mathbb{F}(G)$ is then defined as $K_{\mathbb{F}(G)} = \bigcup_{P \in K \sqcap \mathbb{F}(G)} P$.

Notice that in contrast to the family of "construction bricks" $K \sqcap \mathbb{F}(G)$, the set $K_{\mathbb{F}(G)}$ is the (finite) union of all polytopes from $K \sqcap \mathbb{F}(G)$. A point $x \in K \cap \mathbb{L}^d$ is called a *boundary point* of the discretized set $K \cap \mathbb{L}^d$ with respect to the adjacency system $\mathbb{F}(G)$ if $x \in \partial K_{\mathbb{F}(G)}$. The set of all boundary points of $K \cap \mathbb{L}^d$ will be denoted by $\partial(K \cap \mathbb{L}^d)$.

In the following, saying that we compute the intrinsic volumes of the discretized set $K \cap \mathbb{L}^d$, we mean the computation of the intrinsic volumes of the polyhedral approximation $K_{\mathbb{F}(G)}$ of K.

3 Algorithm

In this section, we describe a new fast algorithm for the computation of intrinsic volumes of deterministic polyconvex sets and their random counterparts, respectively.

3.1 Deterministic polyconvex sets

Consider a set $K \in \mathcal{R}$ which entirely lies in a sampling window W, i.e. $K \subset W \subset \mathbb{R}^d$. In the following, an algorithm is described that approximates the vector of intrinsic volumes $V(K)$ on the basis of the polyhedral approximation $K_{\mathbb{F}(G)}$ of K defined in Section 2.2. Thus, instead of $\rho_r(K)$, $C(K)$, and $V(K)$, we will compute the corresponding approximations $\rho_r(K_{\mathbb{F}(G)})$, $C(K_{\mathbb{F}(G)})$, and $V(K_{\mathbb{F}(G)})$, respectively.

3.1.1 Basic idea and computational efficiency

Recall that formulae (2.2)–(2.5) provide the theoretical background for the practical computation of $V(K_{\mathbb{F}(G)})$. This means that first an algorithm should be constructed in order to compute the vector $C(K_{\mathbb{F}(G)})$ whose components are given by (2.3). Then, the vector $V(K_{\mathbb{F}(G)}) = A^{-1}_{r_0,\ldots,r_{d-1}} C(K_{\mathbb{F}(G)})$ can be easily determined, where $A^{-1}_{r_0,\ldots,r_{d-1}}$ is the inverse of the matrix $A_{r_0,\ldots,r_{d-1}}$ given in (2.6).

Thus, the main task is to compute the quantity $\rho_r(K_{\mathbb{F}(G)})$ for a fixed $r > 0$. This will be done in three steps. First, we discretize the integral in (2.3) with respect to the lattice $\mathbb{L}^d$, which gives

$$\rho_r(K_{\mathbb{F}(G)}) \approx \Delta^d \sum_{x \in (\partial K_{\mathbb{F}(G)} \oplus B_r(o)) \cap \mathbb{L}^d} \quad \sum_{q \in \partial K_{\mathbb{F}(G)} \setminus \{x\}} J(K_{\mathbb{F}(G)} \cap B_r(x), q, x) \tag{3.1}$$

or, equivalently,

$$\rho_r(K_{\mathbb{F}(G)}) \approx \Delta^d \sum_{x \in (\partial K_{\mathbb{F}(G)} \oplus B_r(o)) \cap \mathbb{L}^d} \Bigg(\sum_{q \in \partial(K \cap \mathbb{L}^d) \setminus \{x\}} + \sum_{q \in \partial K_{\mathbb{F}(G)} \setminus \big(\partial(K \cap \mathbb{L}^d) \cup \{x\}\big)} \Bigg) J(K_{\mathbb{F}(G)} \cap B_r(x), q, x)\,,$$

where the inner sum in (3.1) has been decomposed into two sums considering those boundary points $q \neq x$ of $K_{\mathbb{F}(G)}$ separately which belong to the lattice $\mathbb{L}^d$ and those which do not possess this property, respectively. Recall that the first as well as the second inner sum extend over finitely many $q \in \partial K_{\mathbb{F}(G)}$ only. For reasons of computational efficiency, we interchange the resulting sums getting

$$\rho_r(K_{\mathbb{F}(G)}) \approx \Delta^d \sum_{q\in\partial(K\cap\mathbb{L}^d)} S_r(q) + \Delta^d \sum_{k=1}^{d-1} \sum_{P\in\mathcal{F}^k(G),\, P\subset\partial K_{\mathbb{F}(G)}} M_r(P)\,, \tag{3.2}$$

where

$$S_r(q) = \sum_{x\in\mathbb{L}^d,\, 0<|x-q|\leqslant r} J(K_{\mathbb{F}(G)} \cap B_r(x), q, x)\,, \qquad M_r(P) = \sum_{x\in D_r(P)} J(K_{\mathbb{F}(G)} \cap B_r(x), \tau_P(x), x)\,, \tag{3.3}$$

and $D_r(P) = \{x \in \mathbb{L}^d : \tau_P(x) \in \mathrm{int}(P), 0 < |x - \tau_P(x)| \leqslant r\}$ is the set of those lattice points x for which their orthogonal projection $\tau_P(x)$ on the k–dimensional "plane" induced by the k–facet P belongs to the (k–dimensional) inner part $\mathrm{int}(P)$ of P and the Euclidean distance $|x - \tau_P(x)|$ is positive, but not larger than r. Finally, in the third step of the algorithm, we have to compute the inner sums $S_r(q)$ and $M_r(P)$ given in (3.3).

Notice that $\tau_P(x) \notin \mathbb{L}^d$ if $x \in D_r(P)$. Furthermore, the "weight" $M_r(P)$ of the k–facet $P \subset \partial K_{\mathbb{F}(G)}$ is equal to the cardinality $\mathrm{card}(D_r(P))$ of the set $D_r(P)$, since $J(K_{\mathbb{F}(G)} \cap B_r(x), \tau_P(x), x) = 1$ for each $x \in D_r(P)$. We also remark that the direct computation of the sum (3.1) would be not efficient. Indeed, let l be the number of (boundary) pixels in $\partial(K\cap\mathbb{L}^d)$ and m be the total number of pixels in the (discretized) window $W \cap \mathbb{L}^d$. Then, the direct computation of the sum (3.1) would require $O(m + l^2r^d)$ operations for each radius r, whereas the fast algorithm based on (3.2) has complexity $O(m)$; see Section 3.1.3.

3.1.2 Computation of the sums $S_r(q)$ and $M_r(P)$

In what follows, we show how the computational complexity, which is necessary to determine the sums in (3.2)–(3.3), can be reduced to $O(m)$ arithmetic operations by means of linear binary filtering. Then, the computations can be arranged in such a way that only one single scan of the image is required.

To see this, we first notice that the index function $J(K_{\mathbb{F}(G)} \cap B_r(x), q, x)$ can be interpreted as one minus the "local" Euler–Poincaré characteristic of $K_{\mathbb{F}(G)} \cap B_r(x)$ at $q \in \partial K_{\mathbb{F}(G)}$ "in direction" $q - x$. Hence, the sum $S_r(q)$ in (3.3) does not depend on the location of $q \in \partial(K \cap \mathbb{L}^d)$ in $W \cap \mathbb{L}^d$ but only on the behavior of the boundary $\partial K_{\mathbb{F}(G)}$ in a small neighborhood of q, i.e., on the configurations of foreground pixels in the lattice neighborhood $N_\Gamma(q)$ introduced in Section 2.2. To handle this situation, a standard tool of image analysis, the so–called *linear binary filter* can be used to code all possible configurations of foreground and background pixels in the lattice neighborhoods of the image; see e.g. [8], [15]. Let $N_\Gamma(q) = \{q_0, \ldots, q_\nu\}$ be the lattice neighborhood of a pixel $q = q_0 \in \mathbb{L}^d$ that contains the lattice points $q_1, \ldots, q_\nu \in \mathbb{L}^d$. For each $q \in \partial(K \cap \mathbb{L}^d)$, the binary image $\{\mathbf{1}_{K\cap\mathbb{L}^d}(x),\, x \in N_\Gamma(q)\}$ is coded by a sum of exponents of two, considering the bijective mapping

$$\{\mathbf{1}_{K\cap\mathbb{L}^d}(x),\, x \in N_\Gamma(q)\} \mapsto b(K \cap \mathbb{L}^d, N_\Gamma(q)) = \sum_{j=0}^{\nu} \mathbf{1}_{K\cap\mathbb{L}^d}(q_j)2^j\,. \tag{3.4}$$

Then, instead of computing the sum $S_r(q)$ for each point $q \in \partial(K \cap \mathbb{L}^d)$, it can be computed for each neighborhood configuration $i = b(K \cap \mathbb{L}^d, N_\Gamma(q))$, $i = 0, \ldots, 2^\nu - 1$, and weighted by its frequency $h_{S,i}$ among all coded neighborhood configurations of the image $\mathbf{1}_{K \cap \mathbb{L}^d}$. In other words, the first sum in (3.2) rewrites $\Delta^d \sum_{i=0}^{2^\nu-1} h_{S,i} S_{r,i}$, where $S_{r,i}$ denotes the sum $S_r(q)$ for a boundary point $q \in \partial(K \cap \mathbb{L}^d)$ with the neighborhood $N_\Gamma(q)$ such that $b(K \cap \mathbb{L}^d, N_\Gamma(q)) = i$. This approach is efficient since the number 2^ν of possible neighborhood configurations is, as a rule, much smaller than the number m of pixels in $W \cap \mathbb{L}^d$.

Anyhow, the algorithm for the computation of $S_{r,i}$ for each $i = 0, \ldots, 2^\nu - 1$ heavily depends on the dimension d. In Section 4, the case $d = 2$ is considered in detail, whereas the 3–dimensional case will be discussed separately in a forthcoming paper. Notice however that for a boundary pixel $q \in \partial(K \cap \mathbb{L}^d)$ with neighborhood configuration $\{\mathbf{1}_{K \cap \mathbb{L}^d}(x),\, x \in N_\Gamma(q)\}$ of a given code i, it is not necessary to compute the index $J(K_{\mathbb{F}(G)} \cap B_r(x), q, x)$ in the sum

$$S_{r,i} = \sum_{x \in \mathbb{L}^d,\, 0 < |x-q| \leqslant r} J(K_{\mathbb{F}(G)} \cap B_r(x), q, x) \tag{3.5}$$

separately for each $x \in \mathbb{L}^d$ with $0 < |x - q| \leqslant r$. Instead, the sum $S_{r,i}$ can be computed as a whole; see Section 4 for details in the planar case $d = 2$.

Likewise, the sum $M_r(P)$ in (3.3) does not depend on the location of the k–dimensional polytope P but on its orientation with respect to the lattice $\mathbb{L}^d$. Hence, there exist at most $\mu = \sum_{k=1}^{d-1} \operatorname{card}(\mathcal{F}^k(\bar{L}))$ possible types of partial sums $M_r(P)$ that we denote by $M_{r,i}$, $i = 0, \ldots, \mu - 1$. Each of them can be computed just by computing the cardinality of the set $D_r(P)$. For any polytope $P \in \bigcup_{k=1}^{d-1} \mathcal{F}^k(G)$ of a given type i, let $h_{M,i}$ be the number of such polytopes in $\partial K_{\mathbb{F}(G)}$. Notice that these numbers can be computed on the basis of neighborhood configurations simultaneously with the frequencies $h_{s,i}$ during the first scan of the image. Finally, an approximation of $\rho_r(K_{\mathbb{F}(G)})$ is obtained by

$$\rho_r(K_{\mathbb{F}(G)}) \approx \Delta^d \Big(\sum_{i=0}^{2^\nu-1} h_{S,i} S_{r,i} + \sum_{i=0}^{\mu-1} h_{M,i} M_{r,i} \Big) . \tag{3.6}$$

3.1.3 Overview of the individual steps

The algorithm described above can be summarized as follows.

1. Scan the image and code all its neighborhood configurations according to (3.4)
2. For each neighborhood configuration of type $i = 0, \ldots, 2^\nu - 1$, compute its frequency $h_{S,i} \geqslant 0$ among all coded neighborhoods $\{\mathbf{1}_{K \cap \mathbb{L}^d}(x),\, x \in N_\Gamma(q)\}$ of type i with $q \in \partial(K \cap \mathbb{L}^d)$, i.e., compute the neighborhood histogram of the boundary $\partial(K \cap \mathbb{L}^d)$.
3. For each polytope P of type $i = 0, \ldots, \mu - 1$, compute the frequency $h_{M,i} \geqslant 0$ of its occurrence in $\partial K_{\mathbb{F}(G)}$.
4. For any i with $h_{S,i} > 0$, compute $S_{r,i}$ for $r = r_0, \ldots, r_{d-1}$ as given in (3.5).

5. For any i with $h_{M,i} > 0$, compute $M_{r,i}$ for $r = r_0, \ldots, r_{d-1}$.

6. For $r = r_0, \ldots, r_{d-1}$, compute an approximation of $\rho_r(K_{\mathbb{F}(G)})$ using (3.6). Form the corresponding approximation of the vector $C(K_{\mathbb{F}(G)}) = (\rho_{r_0}(K_{\mathbb{F}(G)}), \ldots, \rho_{r_{d-1}}(K_{\mathbb{F}(G)}), V_d(K_{\mathbb{F}(G)}))^\top$, where the value $V_d(K_{\mathbb{F}(G)})$ is approximated just by counting of all foreground pixels in $W \cap \mathbb{L}^d$.

7. Compute an approximation of $V(K_{\mathbb{F}(G)}) = A^{-1}_{r_0 \ldots r_{d-1}} C(K_{\mathbb{F}(G)})$.

Notice that for an arbitrary number n of d–tuples of dilation radii $(r_{0j}, \ldots, r_{d-1,j})$, $j = 1, \ldots, n$, only one scan of the image is required to perform the above algorithm n times. Furthermore, it is possible to compute the values $S_{r,i}$, $M_{r,i}$ for all plausible radii r in advance and to store them in an array in order to use these values in each program run. Doing so, the complexity of the algorithm is $O(m)$.

3.2 Samples from stationary RACS

Let $K = \xi \cap W$ be that part of a realization ξ of a stationary RACS Ξ sampled in a (compact and convex) observation window $W \subset \mathbb{R}^d$. Notice that this leads to edge effects, because the (unbounded) stationary RACS Ξ is sampled only in the bounded window W. Thus, two edge–corrected modifications of the algorithm described in Section 3.1 are proposed in order to compute the estimators $\widetilde{v} = \left(\widetilde{v}_0, \ldots, \widetilde{v}_d\right)^\top = A^{-1}_{r_0 \ldots r_{d-1}} \widetilde{c}$ and $\widehat{v}^{\,+} = \left(\widehat{v}_0^{\,+}, \ldots, \widehat{v}_d^{\,+}\right)^\top = A^{-1}_{r_0 \ldots r_{d-1}} \widehat{c}^{\,+}$ introduced in Section 2.1 for the vector v of specific intrinsic volumes of the stationary RACS Ξ.

3.2.1 Two edge–corrected modifications of the algorithm

As in Section 3.1, we replace K by its polyhedral approximation $K_{\mathbb{F}(G)}$. In particular, in the definition of the estimator $\widetilde{c} = (\widetilde{c}_0, \ldots, \widetilde{c}_d)$ given in (2.11)–(2.12), we replace Ξ by $K_{\mathbb{F}(G)}$ and $V_d(W \ominus B_{r_i}(o))$ by $\Delta^d \text{card}(W \ominus B_{r_i}(o)) \cap \mathbb{L}^d)$, respectively. Thus, we consider the vector

$$\widetilde{c}(K_{\mathbb{F}(G)}) = \left(\frac{\widetilde{\rho}_{r_0}(K_{\mathbb{F}(G)})}{\Delta^d \,\text{card}((W \ominus B_{r_0}(o)) \cap \mathbb{L}^d)}, \ldots, \frac{\widetilde{\rho}_{r_{d-1}}(K_{\mathbb{F}(G)})}{\Delta^d \,\text{card}((W \ominus B_{r_{d-1}}(o)) \cap \mathbb{L}^d)}, \frac{\text{card}(K \cap \mathbb{L}^d)}{\text{card}(W \cap \mathbb{L}^d)} \right),$$

where we discretize the integral in (2.12) with respect to the lattice $\mathbb{L}^d$ getting

$$\widetilde{\rho}_r(K_{\mathbb{F}(G)}) \approx \Delta^d \sum_{x \in (\partial K_{\mathbb{F}(G)} \oplus B_r(o)) \cap (W \ominus B_r(o)) \cap \mathbb{L}^d} \;\; \sum_{q \in \partial K_{\mathbb{F}(G)} \setminus \{x\}} J(K_{\mathbb{F}(G)} \cap B_r(x), q, x) \tag{3.7}$$

for each $r = r_0, \ldots, r_{d-1}$, and, after changing the order of summation,

$$\widetilde{\rho}_r(K_{\mathbb{F}(G)}) \approx \Delta^d \sum_{q \in \partial(K \cap \mathbb{L}^d)} \widetilde{S}_r(q) + \Delta^d \sum_{k=1}^{d-1} \; \sum_{P \in \mathcal{F}^k(G),\, P \subset \partial K_{\mathbb{F}(G)}} \widetilde{M}_r(P)\,,$$

where

$$\widetilde{S}_r(q) = \sum_{x \in (W \ominus B_r(o)) \cap \mathbb{L}^d,\, 0 < |x-q| \leqslant r} J(K_{\mathbb{F}(G)} \cap B_r(x), q, x)\,, \quad \widetilde{M}_r(P) = \sum_{x \in \widetilde{D}_r(P)} J(K_{\mathbb{F}(G)} \cap B_r(x), \tau_P(x), x)\,,$$

and $\widetilde{D}_r(P) = \{x \in (W \ominus B_r(o)) \cap \mathbb{L}^d : \tau_P(x) \in \operatorname{int}(P), 0 < |x - \tau_P(x)| \leqslant r\}$.

In the present case, the quantities $\widetilde{S}_r(q)$ and $\widetilde{M}_r(P)$ depend not only on the type i of neighborhood configuration but also on the location of q and P, respectively. Indeed, only those lattice points x are counted in the above sums which have distance from q and P, respectively, not larger than r and lie in the reduced sampling window $W \ominus B_r(o)$. However, if the sampling window is large relative to the radius r, i.e., if $V_d(W \ominus B_{2r}(o)) > 0$, then $\widetilde{S}_r(q) = S_r(q)$ and $\widetilde{M}_r(P) = M_r(P)$ for lattice points $q \in \partial(K \cap \mathbb{L}^d) \cap (W \ominus B_{2r}(o))$ and $P \subset (W \ominus B_{2r}(o))$, respectively. In fact, for such q and P, the circle $B_r(q)$ and the set $\{x \in \mathbb{R}^d : 0 < |x - \tau_P(x)| \leqslant r\}$ belong to the reduced sampling window $W \ominus B_r(o)$. Thus, we have

$$\begin{aligned} \widetilde{\rho}_r(K_{\mathbb{F}(G)}) \quad \approx \quad & \Delta^d \sum_{q \in \partial(K \cap \mathbb{L}^d) \cap W \ominus B_{2r}(o)} S_r(q) + \Delta^d \sum_{k=1}^{d-1} \sum_{P \in \mathcal{P}_1} M_r(P) \\ & + \Delta^d \sum_{q \in \partial(K \cap \mathbb{L}^d) \setminus W \ominus B_{2r}(o)} \widetilde{S}_r(q) + \Delta^d \sum_{k=1}^{d-1} \sum_{P \in \mathcal{P}_2} \widetilde{M}_r(P)\,, \end{aligned} \tag{3.8}$$

where $\mathcal{P}_1 = \{P \in \mathcal{F}^k(G) : P \subset \partial K_{\mathbb{F}(G)} \cap (W \ominus B_{2r}(o))\}$ and $\mathcal{P}_2 = \{P \in \mathcal{F}^k(G) : P \subset \partial K_{\mathbb{F}(G)}\} \setminus \mathcal{P}_1$. The first two sums of the latter expression can be computed as in Section 3.1 using (3.6). However, for those q and D being close to the boundary of the window W, the quantities $\widetilde{S}_r(q)$ and $\widetilde{M}_r(P)$ have to be computed directly.

Suppose now that $W = [-a, a]^d$ is a cubic observation window for some $a > 0$. Then, in order to compute the estimators $\widehat{c}^{\,+}$ and $\widehat{v}^{\,+} = A^{-1}_{r_0 \ldots r_{d-1}} \widehat{c}^{\,+}$ introduced in Section 2.1, we consider the vector

$$\widehat{c}^{\,+}(K_{\mathbb{F}(G)}) = \left(\frac{\rho^+_{r_0}(K_{\mathbb{F}(G)}, W)}{\Delta^d \operatorname{card}((W \ominus B_{r_0}(o)) \cap \mathbb{L}^d)}, \ldots, \frac{\rho^+_{r_{d-1}}(K_{\mathbb{F}(G)}, W)}{\Delta^d \operatorname{card}((W \ominus B_{r_{d-1}}(o)) \cap \mathbb{L}^d)}, \frac{\operatorname{card}(K \cap \mathbb{L}^d)}{\operatorname{card}(W \cap \mathbb{L}^d)} \right),$$

where $\rho^+_{r_i}(K_{\mathbb{F}(G)}, W) = \rho_{r_i}(K_{\mathbb{F}(G)} \cap W) - \rho_{r_i}(K_{\mathbb{F}(G)} \cap \partial^+ W)$ for $i = 0, \ldots, d-1$. Notice that $\rho_{r_i}(K_{\mathbb{F}(G)} \cap W)$ and $\rho_{r_i}(K_{\mathbb{F}(G)} \cap \partial^+ W)$ can be computed in the same way as it has been discussed in Section 3.1 for the deterministic case.

3.2.2 Overview of the individual steps

For each radius $r = r_0, \ldots, r_{d-1}$, the first edge–corrected algorithm described in Section 3.2.1 can be split into the following steps:

1. Scan the reduced image $\{\mathbf{1}_{\xi \cap \mathbb{L}^d}(x),\ x \in W \ominus B_{2r}(o)\}$ and code all its neighborhood configurations according to (3.4).
2. For each neighborhood configuration of type $i = 0, \ldots, 2^\nu - 1$, compute its frequency $h_{S,i} \geqslant 0$ among all coded neighborhoods $\{\mathbf{1}_{(K \cap \mathbb{L}^d}(x),\ x \in N_\Gamma(q)\}$ of type i with $q \in \partial(K \cap \mathbb{L}^d) \cap (W \ominus B_{2r}(o))$, i.e., compute the neighborhood histogram of the boundary $\partial(K \cap \mathbb{L}^d) \cap (W \ominus B_{2r}(o))$.
3. For each polytope $P \in \mathcal{P}_1$ of type $i = 0, \ldots, \mu - 1$, compute the frequency $h_{M,i} \geqslant 0$ of its occurrence in $\mathcal{P}_1$.

4. For any i with $h_{S,i} > 0$, compute $S_{r,i}$ as given in (3.5).

5. For any i with $h_{M,i} > 0$, compute $M_{r,i}$.

6. Scan the rest of the image $\{\mathbf{1}_{\xi \cap \mathbb{L}^d}(x),\ x \in W \setminus (W \ominus B_{2r}(o))\}$. For any $q \in \partial(K \cap \mathbb{L}^d) \setminus (W \ominus B_{2r}(o))$ and $P \in \mathcal{P}_2$, compute the sums $\widetilde{M}_r(P)$ and $\widetilde{S}_r(q)$.

7. Compute an approximation of $\widetilde{\rho}_r(K_{\mathbb{F}(G)})$ using (3.8). Form the corresponding approximation of the vector $\widetilde{c}(K_{\mathbb{F}(G)})$.

8. Compute the approximation of $\widetilde{v}(K_{\mathbb{F}(G)}) = A^{-1}_{r_0 \dots r_{d-1}} \widetilde{c}(K_{\mathbb{F}(G)})$.

As in the deterministic case of Section 3.1, the first edge–corrected algorithm requires only one single scan of the image. However, the complexity of computation is $O(m + lr^d)$ and, therefore, higher than in the deterministic case. On the other hand, if W is a parallelepiped, then the edge–corrected algorithm considered in (2.13) can be used, which is based on the computation of $\widehat{c}^{\,+}(K_{\mathbb{F}(G)})$. This algorithm can be entirely reduced to the deterministic case and, therefore, has complexity $O(m)$.

3.3 Appropriate choice of dilation radii

3.3.1 Selection of d dilation radii

Computer experiments showed that the accuracy of our algorithm for the computation of $V(K_{\mathbb{F}(G)})$, $\widetilde{v}(K_{\mathbb{F}(G)})$, and $\widehat{v}^{\,+}(K_{\mathbb{F}(G)})$, respectively, heavily depends on the choice of the d-tuple of radii $(r_0, \dots, r_{d-1})$. For instance, the error that appears in discretizing the integral in (2.3) is substantial for small radii, e.g. $r_0 = 1, \dots, r_{d-1} = d$. Therefore, the problem of an appropriate choice of the radii $r_0, \dots, r_{d-1}$ arises.

For deterministic polyconvex sets K, there are no restrictions on the r_i from above. Thus, the largest values of r_i can be chosen in such a way that the run times of the algorithm are still acceptable. In practice, this could be $r_i \approx 10000$. If $K = \xi \cap W$ for some realization ξ of a stationary RACS Ξ and the first edge–corrected algorithm based on minus–sampling is considered, then the maximum radius $r = \max\{r_0, \dots, r_{d-1}\}$ should not exceed the half–diameter of W, which ensures that $V_d(W \ominus B_r(o)) > 0$. However, if the reduced observation window $W \ominus B_r(o)$ is too small, the computations are unstable because the number of lattice points within $W \ominus B_r(o)$ is not large enough to provide a good approximation (3.7) of the integral in (2.12). Hence, the radii $r_0, \dots, r_{d-1}$ should be chosen not too small and not too large compared to the diameter of W. Furthermore, enlarging the window W, i.e., increasing the resolution of the image allows larger values of the radii $r_0, \dots, r_{d-1}$. On the other hand, if the observation window W is a parallelepiped and the second edge–corrected algorithm is used, then the radii r_i should be chosen possibly large. The above recommendations have been proved by numerical experiments; see Section 5.

3.3.2 Methods for more than d dilation radii

The computational results can be significantly improved if, instead of taking d dilation radii $r_0, \dots, r_{d-1}$, the image is analyzed for more than d radii. Then, our approach can be combined with various standard methods of statistics in order to further improve the estimation of Minkowski functionals. In the following, we just mention two of such possibilities.

Suppose that the image is a polyconvex set $K \subset W$ which is analyzed for n different d–tuples of dilation radii $r^{(i)} = (r_{0i}, \ldots, r_{d-1,i})$. Let $V^{(i)}(K_{\mathbb{F}(G)})$ denote the output of our algorithm for the i-th d–tuple $r^{(i)}$ of radii; $i = 1, \ldots, n$. Then, the sample $\bigl(V^{(1)}(K_{\mathbb{F}(G)}), \ldots, V^{(n)}(K_{\mathbb{F}(G)})\bigr)$ of size n is formed, where the numerical experiments showed that the sample mean

$$\overline{V}(K_{\mathbb{F}(G)}) = \frac{1}{n}\sum_{i=1}^{n} V^{(i)}(K_{\mathbb{F}(G)})$$

is more precise and much less sensitive to outliers resulting from the discretization error. Notice that instead of the sample mean, other sample functions like e.g. the median can be used in order to compute approximations for $V(K_{\mathbb{F}(G)})$.

On the other hand, even better results can be obtained by the least squares method, where a single n–tuple of radii $(r_0, \ldots, r_{n-1})$ with $n > d$ is considered. In the case of a deterministic polyconvex set $K \subset W$, this leads to the following overdetermined system of linear equations, which corresponds to (2.5):

$$\begin{pmatrix} \rho_{r_0}(K_{\mathbb{F}(G)}) \\ \vdots \\ \rho_{r_{n-1}}(K_{\mathbb{F}(G)}) \end{pmatrix} = \begin{pmatrix} r_0^d k_d & r_0^{d-1} k_{d-1} & \ldots & r_0^2 k_2 & r_0 k_1 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ r_{n-1}^d k_d & r_{n-1}^{d-1} k_{d-1} & \ldots & r_{n-1}^2 k_2 & r_{n-1} k_1 \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_{d-1} \end{pmatrix}, \tag{3.9}$$

or, in matrix form, $C'(K_{\mathbb{F}(G)}) = A'\,x$, where $C'(K_{\mathbb{F}(G)}) = \bigl(\rho_{r_0}(K_{\mathbb{F}(G)}), \ldots, \rho_{r_{n-1}}(K_{\mathbb{F}(G)})\bigr)^\top$, A' denotes the matrix at the right-hand side of (3.9), and $x = (x_0, \ldots, x_{d-1})^\top \in \mathbb{R}^d$ is some d–dimensional vector. Notice that, typically, there exists no $x \in \mathbb{R}^d$ which solves (3.9) exactly. However, it is well known that the vector $V^*(K_{\mathbb{F}(G)}) = \bigl((A')^\top A'\bigr)^{-1}(A')^\top C'(K_{\mathbb{F}(G)})$ is the unique solution of the minimization problem

$$\bigl|\, C'(K_{\mathbb{F}(G)}) - A'\,V^*(K_{\mathbb{F}(G)})\,\bigr| = \min_{x \in \mathbb{R}^d} \bigl|\, C'(K_{\mathbb{F}(G)}) - A'\,x)\,\bigr|$$

and, therefore, can be regarded as approximation of $V'(K_{\mathbb{F}(G)}) = (V_0(K_{\mathbb{F}(G)}), \ldots, V_{d-1}(K_{\mathbb{F}(G)}))^\top$. If $K = \xi \cap W$ for some realization ξ of a stationary RACS Ξ and the first edge–corrected algorithm is used, then a system of linear equations can be considered which is similar to (3.9) and which leads to a quite good approximation of the vector $v\,'(K_{\mathbb{F}(G)}) = (v_0(K_{\mathbb{F}(G)}), \ldots, v_{d-1}(K_{\mathbb{F}(G)}))^\top$ of specific intrinsic volumes. In the case of the second edge–corrected algorithm, we can proceed in a completely analogous way; see Section 5 for further details.

4 The planar case

The general description of our algorithm, which has been given in Section 3 for the d–dimensional case, can be specified for $d = 2$ and $d = 3$, respectively. In the present section, the planar case $d = 2$ is considered in detail, where we focus on the case of deterministic polyconvex sets, since its stochastic counterparts for realizations of stationary RACS differ only in performing the edge correction. A thorough study of the algorithmic specifications for 3D binary images will be the subject of a forthcoming paper.

4.1 Neighborhood configurations of boundary points

Two particular adjacency systems for the polyhedral approximation of discretized sets are standard in image analysis: the minimum adjacency $\mathbb{F}(G_{\min})$ and the maximum adjacency $\mathbb{F}(G_{\max})$ mentioned in Section 2.2. Anyhow, recall that the general algorithmic approach of Section 3 can be applied to any adjacency system.

4.1.1 4–neighborhood and 8–neighborhood

In this section, we briefly describe how $\mathbb{F}(G_{\min})$ and $\mathbb{F}(G_{\max})$ are specified in the planar case $d = 2$.

Let $l_0 = (0,0)$, $l_1 = (\Delta, 0)$, $l_2 = (\Delta, \Delta)$, $l_3 = (0, \Delta)$ be the vertices of the unit cell L of the square lattice $\mathbb{L}^2 = \Delta\mathbb{Z}^2$. Furthermore, by $\mathrm{conv}\{x_1, \ldots, x_k\}$, we denote the convex hull of the points $x_1, \ldots, x_k \in \mathbb{R}^2$. The maximum adjacency $\mathbb{F}(G_{\max})$ in $\mathbb{R}^2$ can be generated by all lattice rotations, translations, and intersections of the following family

$$F = \{\mathrm{conv}\{l_0, \ldots, l_i\},\ i = 0, \ldots, 3\} = \big\{\{l_0\},\ [l_0, l_1],\ \mathrm{conv}\{l_0, l_1, l_2\},\ \overline{L}\big\}$$

of polygons in $\mathbb{R}$ and $\mathbb{R}^2$, i.e., $\mathbb{F}(G_{\max}) = \{T_1(P_1) \cap \ldots \cap T_k(P_k) : P_i \in F,\ T_i \in \mathbb{T}^2,\ i = 1, \ldots, k,\ k \in \mathbb{N}\}$, where $\mathbb{T}^2$ is the invariance group of $\mathbb{L}^2$, i.e., the group of all rigid motions in $\mathbb{R}^2$ that map $\mathbb{L}^2$ into itself. It can be easily seen that $\mathbb{F}(G_{\max})$ implies the neighborhood relation γ that is well–known as the 8–*neighborhood* in image analysis; see [15]. Namely, each point $x_0 \in \mathbb{L}^2$ has exactly $\nu = 8$ neighbors $x_1 = x_0 + l_1$, $x_2 = x_0 + l_2$, $x_3 = x_0 + l_3$, $x_4 = x_0 - l_1 + l_3$, $x_5 = x_0 - l_1$, $x_6 = x_0 - l_2$, $x_7 = x_0 - l_3$, $x_8 = x_0 + l_1 - l_3$, where we briefly write $N_\Gamma(x_0) = \{x_0, \ldots, x_8\}$. Notice that the pixels of $N_\Gamma(x_0)$ are counterclockwise ordered in a "spiral" way beginning with the central pixel x_0; see Figure 4.1.

The minimum adjacency $\mathbb{F}(G_{\min})$ in $\mathbb{R}^2$ can be generated by all lattice translations and intersections of the closed lattice cell $\overline{L}$ of $\mathbb{L}^2$:

$$\mathbb{F}(G_{\min}) = \{T_1(\overline{L}) \cap \ldots \cap T_k(\overline{L}) : T_i \in \mathbb{T}^2,\ i = 1, \ldots, k,\ k \in \mathbb{N}\}\,.$$

It can be easily seen that $\mathbb{F}(G_{\min})$ yields the so–called 4–*neighborhood*; see Figure 4.1. That is, each point $x_0 \in \mathbb{L}^2$ has exactly $\nu = 4$ neighbors $x_1 = x_0 + l_1$, $x_3 = x_0 + l_3$, $x_5 = x_0 - l_1$, $x_7 = x_0 - l_3$.

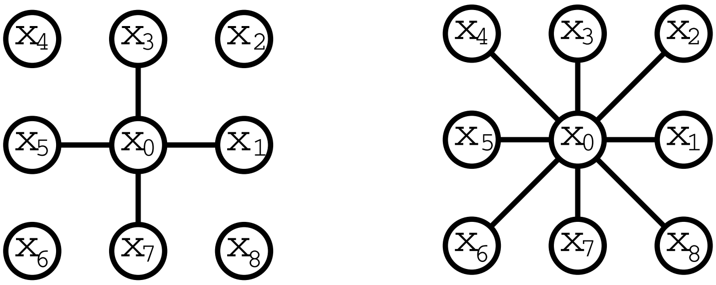

Figure 4.1: 4–neigborhood and 8–neighborhood

4.1.2 Boundary points

Suppose that the discretization $K \cap \mathbb{L}^2$ of a deterministic polyconvex set $K \subset W \subset \mathbb{R}^2$ is given. Using the polygons of $\mathbb{F}(G_{\max})$ as construction stones, the approximation $K_{\mathbb{F}(G_{\max})}$ can be built, which itself is a polygon with the following boundary structure. A point $q_0 \in K \cap \mathbb{L}^2$ is a boundary point of $K_{\mathbb{F}(G_{\max})}$, i.e., $q_0 \in \partial K_{\mathbb{F}(G_{\max})}$, if at least one pixel q_i of its 4–neighborhood $\{q_1, q_3, q_5, q_7\}$ does not belong to $K \cap \mathbb{L}^2$, where $q_1 = q_0 + l_1$, $q_3 = q_0 + l_3$, $q_5 = q_0 - l_1$, $q_7 = q_0 - l_3$. In terms of binary images, a foreground pixel q_0 belongs to $\partial K_{\mathbb{F}(G_{\max})}$ if there is at least one background pixel in its 4–neighborhood.

Analogously, on the basis of $\mathbb{F}(G_{\min})$, the polygonal approximation $K_{\mathbb{F}(G_{\min})}$ of a polyconvex set K can be built from its digitized version $K \cap \mathbb{L}^2$. Then, a point $q_0 \in K \cap \mathbb{L}^2$ is a boundary point of $K_{\mathbb{F}(G_{\min})}$, i.e., $q_0 \in \partial K_{\mathbb{F}(G_{\min})}$, if at least one lattice point q_i of its 8–neighborhood $\{q_1, \ldots, q_8\}$ does not belong to $K \cap \mathbb{L}^2$. Hence, although the 4–neighborhood relation is used for the polygonal approximation of the set K, we have to consider the 8–neighborhood of a given pixel to decide whether it belongs to the boundary $\partial K_{\mathbb{F}(G_{\min})}$.

4.1.3 Neighborhood configurations

Altogether, there are $2^9 = 512$ possible configurations of 8–neighborhoods. After coding them as described in (3.4), we only need to consider neighborhoods $N_\Gamma(x_0) = \{x_0, \ldots, x_8\}$ of foreground lattice points $x_0 \in K \cap \mathbb{L}^2$. They can be easily recognized by their code $b(K \cap \mathbb{L}^2, N_\Gamma(x_0)) > 0$ which is an odd number because $\mathbf{1}_{K \cap \mathbb{L}^2}(x_0) = 1$ if $x_0 \in K \cap \mathbb{L}^2$. Thus, the number of different neighborhood configurations of foreground pixels is reduced to 256.

We first consider the maximum adjacency $\mathbb{F}(G_{\max})$ in $\mathbb{R}^2$ and analyze all possible types of neighborhood configurations for pixels on the boundary of the polygon $K_{\mathbb{F}(G_{\max})}$, for which the sums $S_{r,i}$ appearing in (3.5) coincide. Notice that by rigid motions from $\mathbb{T}^2$ and reflections, one can reduce the above number of 256 different configurations to 51. Furthermore, we omit those configurations $\{\mathbf{1}_{K \cap \mathbb{L}^2}(x),\ x \in N_\Gamma(x_0)\}$ with $x_0 \notin \partial(K \cap \mathbb{L}^2)$. Then, the number of remaining different neighborhood configurations of boundary pixels is 45; see Figures 4.2–4.3. For the neighborhood $N_\Gamma(q_0) = \{q_0, \ldots, q_8\}$ of each boundary point $q_0 \in \partial(K \cap \mathbb{L}^2)$, we consider the pixel values $b_0, \ldots, b_8$, where $b_i = \mathbf{1}(q_i \in K \cap \mathbb{L}^2)$ for $i = 0, \ldots, 8$. These pixel values are given in Table 4.1. Notice that the image frequencies of different (up to rotations or reflections) neighborhood configurations of the same type i are summed up to $h_{S,i}$. For instance, the neighborhood configurations 100001001, 101001000, 110000100, 100010010, 100100001, 101000010, 110010000, and 100100100 are of type $i = 7$. They differ from each other only by rotations on 90°, 180°, 270° and reflections with respect to the axes (x_1, x_5), (x_3, x_7) and diagonals (x_2, x_6), (x_4, x_8); see Figure 4.1. After scanning the image, the frequencies of occurrence of these neighborhood configurations are summed up to $h_{S,7}$.

Considering the minimum adjacency $\mathbb{F}(G_{\min})$, the family of neighborhood configurations for pixels on the boundary of the polygon $K_{\mathbb{F}(G_{\min})}$ can be analyzed in a similar way. As before, the number of different 8–neighborhood configurations of foreground pixels is equal to 256. Then, by rigid motions from $\mathbb{T}^2$ and reflections, the number of different neighborhood configurations of boundary points $q_0 \in \partial(K \cap \mathbb{L}^2)$ is reduced to 50. The first 45 neighborhood configurations coincide with those given in Table 4.1. The five new configurations given in the left part of Table 4.5 result from the changed definition of the boundary pixels of $K_{\mathbb{F}(G_{\min})}$.

i	$b_0,\ldots,b_8$	i	$b_0,\ldots,b_8$	i	$b_0,\ldots,b_8$	i	$b_0,\ldots,b_8$	i	$b_0,\ldots,b_8$
1	100000000	10	100001110	19	100010101	28	101011010	37	111011001
2	100001000	11	110011000	20	100011110	29	101011100	38	111011010
3	100000100	12	101011000	21	110111000	30	100011011	39	101011011
4	110001000	13	100011001	22	100111001	31	101010101	40	101110101
5	100001010	14	100011010	23	111011000	32	110001111	41	101111110
6	100001100	15	100011100	24	110011001	33	101111001	42	101111011
7	100001001	16	110101000	25	110011010	34	101111010	43	101110111
8	100000101	17	100101001	26	110011100	35	111000111	44	111110101
9	101000100	18	101001001	27	101011001	36	110111001	45	101111111

Table 4.1: Pixel values for the neighborhood configurations of boundary points

4.2 Computation of $S_{r,i}$ and $M_{r,i}$ for the maximum adjacency $\mathbb{F}(G_{\max})$

4.2.1 Polygonal approximation

Consider the maximum adjacency $\mathbb{F}(G_{\max})$ in $\mathbb{R}^2$. Then, for each type of the 45 neighborhood configurations given in Table 4.1, the sums $S_{r,i}$ have to be computed using (3.5). In connection with this, the polygonal approximations of these neighborhood configurations given in Figures 4.2–4.3 must be analyzed. It turns out that they contain 11 different types of boundary elements of which the whole boundary $\partial K_{\mathbb{F}(G_{\max})}$ is made; see Figure 4.4. These boundary elements are given by the neighborhood configurations enumerated in Table 4.1 by 1, 2, 3, 6, 15, 10, 20, 32, 35, 41, and 45, respectively. Furthermore, each neighborhood configuration in Table 4.1 can contain up to 4 different boundary elements; see Figures 4.2–4.3. Thus, in accordance with (3.5), we can write

$$S_{r,i} = \sum_{j=1}^{11} \omega_{ij} J_j, \qquad i = 1,\ldots,45\,, \tag{4.1}$$

where J_j denotes the partial sum in (3.5) which corresponds to a boundary element of type $j = 1,\ldots,11$ and $\omega_{ij} \in \{0,1,2,3,4\}$ is the number of such boundary elements in the neighborhood configuration of type i. The complete list of weights ω_{ij} is given in Table 4.2.

4.2.2 Partial sums J_j corresponding to given boundary elements

In this section, we describe how the partial sums J_j introduced in (4.1) can be computed. Let $q \in \partial(K \cap \mathbb{L}^2)$ be the central pixel of a neighborhood configuration which contains a boundary element of type j as shown in Figure 4.4. Introduce the set

$$H(q) = H(N_\Gamma(q), K_{\mathbb{F}(G_{\max})}) = \{x \in \mathbb{R}^2 : J(K_{\mathbb{F}(G_{\max})}, q, x) \neq 0\}\,.$$

i	$\omega_{i1},\dots,\omega_{i11}$	i	$\omega_{i1},\dots,\omega_{i11}$	i	$\omega_{i1},\dots,\omega_{i11}$	i	$\omega_{i1},\dots,\omega_{i11}$	i	$\omega_{i1},\dots,\omega_{i11}$
1	10000000000	10	00000001000	19	00000000020	28	00000000011	37	00000000011
2	01000000000	11	00010000001	20	00000000100	29	00010000010	38	00000000010
3	00100000000	12	00000000110	21	00010000000	30	00001000000	39	00000000020
4	00002000000	13	00001000001	22	00000000002	31	00000000040	40	00000000030
5	00000001000	14	00000000100	23	00010000010	32	00010000000	41	00000000001
6	00000100000	15	00000010000	24	00000000002	33	00000000011	42	00000000010
7	00000000101	16	00010000000	25	00000000001	34	00000000001	43	00000000020
8	00000010010	17	00000000002	26	00000000002	35	00001000000	44	00000000020
9	00002000000	18	00000000012	27	00000000021	36	00000000001	45	00000000010

Table 4.2: Number of boundary elements ω_{ij} for $\mathbb{F}(G_{\max})$

Depending on the type j of the boundary element, the set $H(q)$ can be a half–line ($j = 4, 5$), a sector between two half–lines ($j = 2, 3, 6, \dots, 11$) or the whole plane ($j = 1$). By the definition of the index function given in (2.4), we have $J(K_{\mathbb{F}(G_{\max})}, q, x) = j_0 \in \{1, -1\}$ for each $x \in H(q)$, where the values j_0 are given in Table 4.3. Notice that the rule in computing j_0 is simple. If q is a point of convexity of $K_{\mathbb{F}(G_{\max})}$, then $j_0 = 1$. Otherwise, we have $j_0 = -1$.

j	1	2	3	4	5	6	7	8	9	10	11
j_0	1	1	1	1	1	1	1	1	1	-1	-1

Table 4.3: Computation of partial sums J_j for $\mathbb{F}(G_{\max})$

Introduce the *index sector* $IS_j = H(q) \cap \mathbb{L}^2 \cap B_r(q) \setminus \{q\}$. For the 11 possible types of boundary elements, their index sectors are marked red in Figure 4.4. The dashed parts of the boundary do not belong to the index sectors whereas the solid parts do. Then, J_j rewrites

$$J_j = j_0 \cdot \operatorname{card}(IS_j)\,. \tag{4.2}$$

The number of lattice points $\operatorname{card}(IS_j)$ as a function of the radius r is given in Table 4.4, where the following notation is used: $a_0(r) = \operatorname{card}(B_r(o) \cap \mathbb{L}^2) - 1$, $a_1(r) = \lfloor r/\Delta \rfloor$, $a_2(r) = \lfloor r/(\sqrt{2}\Delta) \rfloor$, and $\lfloor a \rfloor = \max\{n \in \mathbb{N} \cup \{0\} : n \leqslant a\}$ is the integer part of $a > 0$.

4.2.3 Computation of $M_{r,i}$

In the planar case, there are only two different types of segments on the boundary of the polygon $K_{\mathbb{F}(G_{\max})}$. Modulo lattice translations and rotations, these segments are $P_0 = [l_0, l_1]$ and $P_1 = [l_0, l_2]$. Thus, for the number μ of different types of segments introduced in Section 3.1.2, we have $\mu = 2$. It is evident that $D(P_0) = \emptyset$ and, therefore, $M_{r,0} = 0$. On the other hand, for the diagonal P_1, we have $M_{r,1} = \lfloor r/(\sqrt{2}\Delta) - 1/2 \rfloor + 1$. The number $h_{M,1}$ of diagonals of type P_1 can be computed during the first scan of the image. Indeed, any diagonal $[x, y] \subset \partial K_{\mathbb{F}(G_{\max})}$ belongs to the neighborhoods $N_\Gamma(x)$ and

j	$\mathrm{card}(IS_j)$	j	$\mathrm{card}(IS_j)$
1	$a_0(r)$	7	$a_0(r)/4 + a_2(r)$
2	$a_0(r)/2 + a_1(r)$	8	$a_0(r)/4 + a_1(r)$
3	$a_0(r)/2 + a_2(r)$	9	$a_0(r)/8 + (a_1(r) + a_2(r))/2$
4	$a_1(r)$	10	$a_0(r)/4 - a_2(r)$
5	$a_2(r)$	11	$a_0(r)/8 - (a_1(r) + a_2(r))/2$
6	$3a_0(r)/8 + (a_1(r) + a_2(r))/2$		

Table 4.4: Cardinality of index sectors IS_j for $\mathbb{F}(G_{\max})$

$N_\Gamma(y)$. Thus, the frequency $h_{M,1}$ is equal to the total number of such diagonals in the neighborhood configurations of all boundary points divided by two.

4.3 Computation of $S_{r,i}$ and $M_{r,i}$ for the minimum adjacency $\mathbb{F}(G_{\min})$

The polygonal approximation of 50 neighborhood configurations on the basis of $\mathbb{F}(G_{\min})$ yields 5 different types of boundary elements; see Figure 4.5. These boundary elements are given by the neighborhood configurations enumerated in Tables 4.1 and 4.5 by 1, 2, 32, 10, and 50, respectively. Any neighborhood configuration from Tables 4.1 and 4.5 can contain up to 4 different boundary elements. Thus, as in the case of adjacency $\mathbb{F}(G_{\max})$, we can write $S_{r,i} = \sum_{j=1}^{5} \omega_{ij} J_j$ for $i = 1, \dots, 50$, where J_j denotes the partial sum in (3.5) which corresponds to a boundary element of type $j = 1, \dots, 5$ and $\omega_{ij} \in \{0, 1, 2, 3, 4\}$ is the number of such boundary elements in the neighborhood configuration of type i. The values of $J_j = j_0 \, \mathrm{card}(IS_j)$ are given in Table 4.5, where all possible index sectors IS_j, $j = 1, \dots, 5$ are marked red in Figure 4.5. The weights ω_{ij} are given in Table 4.6. Notice that the boundary of $\mathbb{F}(G_{\min})$ consists of one type of segments only, i.e., $\mu = 1$. Modulo lattice translations and rotations, this is $P_0 = [l_0, l_1]$ with $M_{r,0} = 0$. Hence, the second sum $\sum_{i=0}^{\mu-1} h_{M,i} M_{r,i}$ in (3.6) vanishes.

i	$b_0, \dots, b_8$
46	110101010
47	110111010
48	111111010
49	110111011
50	110111111

j	1	2	3	4	5
j_0	1	1	1	1	-1

j	$\mathrm{card}(IS_j)$
1	$a_0(r)$
2	$a_0(r)/2 + a_1(r)$
3	$a_1(r)$
4	$a_0(r)/4 + a_1(r)$
5	$a_0(r)/4 - a_2(r)$

Table 4.5: New neighborhood configurations (left); computation of the partial sums J_j (center); cardinality of index sectors IS_j (right) for $\mathbb{F}(G_{\min})$

i	$\omega_{i1},\ldots,\omega_{i5}$	i	$\omega_{i1},\ldots,\omega_{i5}$	i	$\omega_{i1},\ldots,\omega_{i5}$	i	$\omega_{i1},\ldots,\omega_{i5}$	i	$\omega_{i1},\ldots,\omega_{i5}$
1	10000	11	00200	21	00101	31	10000	41	00100
2	01000	12	01000	22	00010	32	00100	42	00101
3	10000	13	01000	23	00200	33	00010	43	00200
4	00200	14	00011	24	00200	34	00101	44	00010
5	00011	15	01000	25	00102	35	00010	45	00100
6	01000	16	00102	26	00200	36	00101	46	00004
7	01000	17	00011	27	01000	37	00200	47	00003
8	10000	18	01000	28	00011	38	00102	48	00002
9	10000	19	10000	29	01000	39	00011	49	00002
10	00010	20	00010	30	00011	40	01000	50	00001

Table 4.6: Number of boundary elements ω_{ij} for $\mathbb{F}(G_{\min})$

4.4 Edge correction

For $d = 2$, the edge correction considered in (2.13) can be efficiently implemented as follows. Instead of correcting the functionals ρ_r, apply the edge correction directly to V_0 and V_1, i.e., consider

$$\widehat{v}_0^{\,+} = \frac{V_0(\xi \cap W) - V_0(\xi \cap \partial^+ W)}{V_2(W)}\,, \qquad \widehat{v}_1^{\,+} = \frac{V_1(\xi \cap W) - V_1(\xi \cap \partial^+ W)}{V_2(W)}\,,$$

where ξ is a realization of a stationary RACS Ξ and W is a parallelepiped. Notice that the values of $V_0(\xi \cap W)$ and $V_1(\xi \cap W)$ are computed as described in Sections 4.1–4.3 above. However, the term $V_0(\xi \cap \partial^+ W)$ can be computed by counting the number of connected components (line segments) of $\xi \cap \partial^+ W$. Likewise, $2V_1(\xi \cap \partial^+ W)$ is equal to the number of foreground pixels in the right upper boundary $\partial^+ W$ of W. We also remark that the precision of this method can be improved if the same procedure is simultaneously applied to the left upper, right lower, and left lower boundaries of W, respectively, and if then the arithmetic mean of the four results is taken, cf. [14].

The edge correction described in (2.11)–(2.12) posesses different properties. In particular, this method based on minus sampling is less restrictive with respect to the geometry of the observation window. On the other hand, it is slower (with complexity $O(m + lr^2)$, see Section 3.2) whereas the edge correction considered in (2.13) runs faster and is easier to implement. However, numerical experiments show that the estimators with the edge correction (2.13) mostly have lower accuracy of computation for $G = G_{\max}$ than those with the edge correction described in (2.11)–(2.12), cf. Tables 5.3–5.4.

4.5 Practical choice of dilation radii

Numerical experiments show that the discretization error of the algorithms is smaller for rational values of the radii r_i than for integer–valued radii. The reason lies in the fluctuations of the function $a_0(r)$ for integer–valued and rational r, respectively. Furthermore, the precision of the above algorithm without edge correction can be significantly improved by using a larger number $n \geqslant 1000$ of dilation radii in the least squares method mentioned in Section 3.3.2. For example, the following recursive rule for the choice of r_i yields good computational results together with nice run times of a couple of seconds: $r_0 = 5000$, $r_{i+1} = r_i + 20.3$, $i = 0, \ldots, 999$. The same rule is applicable to the edge–corrected algorithm based on (2.13). For the edge correction based on minus sampling, the radii $r_0 = 5$, $r_{i+1} = r_i + 1.3$, $i = 0, \ldots, 49$ appear to perform pretty well in the case $W = [0, 999]^2$. For $n > 50$ and larger radii, the run times increase drastically. Then, a Penthium IV processor (1.9 GHz) requires up to two hours to process a 1000×1000 binary image with complicated boundary structure.

5 Numerical examples

In this section, the results of numerical experiments are discussed and compared to those of conventional computation methods such as the marching cube algorithm for deterministic polyconvex sets as well as the algorithms based on Crofton's formula, which are given in [8] for samples from stationary RACS. For test purposes, we use polyconvex sets with known Minkowski functionals as well as stationary Boolean models with realizations from the extended convex ring and with known specific intrinsic volumes. Besides the comparison of various algorithmic approaches, this allows the comparison of numerical results with the theoretical values of intrinsic volumes, which are determined by means of analytical formulae.

5.1 Deterministic polyconvex sets

We first consider several examples of deterministic polyconvex sets K such that the Euler–Poincaré characteristic $V_0(K_{\mathbb{F}(G)})$, the boundary length $2\,V_1(K_{\mathbb{F}(G)})$ and the area $V_2(K_{\mathbb{F}(G)})$ of the polygonal approximation $K_{\mathbb{F}(G)}$ of K can be computed directly. Then, using the algorithm described in Section 4 for the planar case $d = 2$, we compute $V_0^*(K_{\mathbb{F}(G)})$ and $2\,V_1^*(K_{\mathbb{F}(G)})$ (see Section 3.3.2 for notation), where we compare these values with $V_0(K_{\mathbb{F}(G)})$ and $2\;V_1(K_{\mathbb{F}(G)})$, respectively. The computations are performed for both the minimum and the maximum adjacency, i.e., for $G = G_{\min}$ and $G = G_{\max}$, respectively.

In particular, in order to evaluate the accuracy of our algorithm, we first compute the exact length $2\,V_1(K_{\mathbb{F}(G)})$ of the boundary $\partial K_{\mathbb{F}(G)}$ consisting of a sequence of line segments. Notice that in the case of the minimum adjacency $\mathbb{F}(G_{\min})$, there is only one type of such segments, namely those that connect a point $x_0 \in \mathbb{L}^2$ to another one from its 4-neighborhood $\{x_1, x_3, x_5, x_7\}$. Each line segment of this type has length Δ. In case of the maximum adjacency $\mathbb{F}(G_{\max})$, diagonal line segments must be considered as well. These are the segments that link a point $x_0 \in \mathbb{L}^2$ with one of its neighbors x_2, x_4, x_6 or x_8. Their length is $\sqrt{2}\Delta$.

The number of the above described line segments (and hence the boundary length) is computed using a method similar to the marching squares algorithm; see [4]. In Figures 5.1 and 5.2, five basic types of such squares are presented for the minimum and maximum adjacencies $\mathbb{F}(G_{\min})$ and $\mathbb{F}(G_{\max})$, respectively. The remaining squares can be generated by rotation. The boundary length and the area attributed to each square type are given in Table 5.1. They clearly differ from the canonical weights of the marching squares algorithm since our computations have to be conform with the algorithmic approach stated in Sections 3 and 4.

G	square type	boundary length	area	G	square type	boundary length	area
$G_{\min}$	1	0	0	$G_{\max}$	1	0	0
$G_{\min}$	2	Δ	0	$G_{\max}$	2	Δ	0
$G_{\min}$	3	0	0	$G_{\max}$	3	$2\sqrt{2}\Delta$	0
$G_{\min}$	4	2Δ	0	$G_{\max}$	4	$\sqrt{2}\Delta$	$\Delta^2/2$
$G_{\min}$	5	0	Δ^2	$G_{\max}$	5	0	Δ^2

Table 5.1: Weights for the basic types of squares

Hence, the exact values of the boundary length $2\,V_1(K_{\mathbb{F}(G)})$ and, similarly, the area $V_2(K_{\mathbb{F}(G)})$ of $K_{\mathbb{F}(G)}$ can be computed easily. Furthermore, for the examples considered in the present section, the Euler-Poincaré characteristic $V_0(K_{\mathbb{F}(G)})$ is determined by counting the number of "clumps" minus the number of "holes".

For the images given in Figures 5.3–5.7, the values obtained for $2\ V_1(K_{\mathbb{F}(G)})$, $V_0(K_{\mathbb{F}(G)})$, $2\ V_1^*(K_{\mathbb{F}(G)})$, and $V_0^*(K_{\mathbb{F}(G)})$ are presented in Table 5.2, where these values are rounded up to the 6th digit after comma. For the algorithm described in Sections 3 and 4, we used the dilation radii $r_0 = 5000$, $r_{i+1} = r_i + 20.3$, $i = 0, \dots, 999$ combined with the least squares method.

Figure	G	$2\,V_1(K_{\mathbb{F}(G)})$	$V_0(K_{\mathbb{F}(G)})$	$2\,V_1^*(K_{\mathbb{F}(G)})$	$V_0^*(K_{\mathbb{F}(G)})$
5.3	$G_{\min}$	756.0	3	755.998293	3.0
5.3	$G_{\max}$	758.828427	2	758.827287	2.0
5.4	$G_{\min}$	1084.0	1	1083.99943	1.0
5.4	$G_{\max}$	1040.651804	1	1040.651169	1.0
5.5	$G_{\min}$	632.0	2	631.998862	2.0
5.5	$G_{\max}$	528.901587	2	528.900294	2.0
5.6	$G_{\min}$	1346.0	1	1345.99943	1.0
5.6	$G_{\max}$	1151.217388	-2	1151.218213	-2.0
5.7	$G_{\min}$	3428.0	0	3427.999997	4.009811E-11
5.7	$G_{\max}$	2970.584053	-5	2970.586176	-5.0

Table 5.2: Exact and approximated values of intrinsic volumes

Notice that the Euler-Poincaré characteristic $V_0(K_{\mathbb{F}(G)})$ of the polyconvex set K in Figure 5.3 depends on its polygonal approximations $K_{\mathbb{F}(G_{\min})}$ and $K_{\mathbb{F}(G_{\max})}$. Evidently, the two upper rectangles are not connected in $K_{\mathbb{F}(G_{\min})}$, whereas they form one "clump" in $K_{\mathbb{F}(G_{\max})}$. However, in both cases the boundary lengths are similar (but not equal!). Furthermore, the polygonal approximation $K_{\mathbb{F}(G_{\max})}$ of the union K of overlapping balls in Figure 5.6 produces three little holes of side length Δ at the intersection points of their bounding circles. For convenience, in Figure 5.6, the regions of their location are zoomed in. Notice that these holes do not exist in $K_{\mathbb{F}(G_{\min})}$, which leads to different values for the Euler-Poincaré characteristic.

In Figure 5.7, a discretized set is considered that contains the five possible boundary elements for $\mathbb{F}(G_{\min})$, which have been described in Figure 4.5, and the 11 possible boundary elements for $\mathbb{F}(G_{\max})$ given in

Figure 4.4. If the computations for the image in Figure 5.7 are based on the minimum adjacency $\mathbb{F}(G_{\min})$, then 7 clumps and 7 holes are obtained (see the marked regions). However, using $\mathbb{F}(G_{\max})$ for the polygonal approximation, 4 clumps and 9 holes occur.

5.2 Samples from stationary Boolean models

Let $X = \{X_1, X_2, \ldots\}$ be a stationary Poisson point process in $\mathbb{R}^2$ with intensity λ and let $\Xi_1, \Xi_2, ...$ be a sequence of independent and identically distributed random compact sets in $\mathbb{R}^2$ that are independent of X. Furthermore, assume that the realizations of Ξ_n belong to the convex ring $\mathcal{R}$ with probability one. Suppose that $E\, V_2(\Xi_0 \oplus K) < \infty$ for each $K \in \mathcal{K}$, where Ξ_0 is a compact RACS distributed as Ξ_n. Then, one can show that $\Xi = \bigcup_{n=1}^{\infty}(X_n + \Xi_n)$ is a stationary RACS in $\mathbb{R}^2$, whose realizations belong to the extended convex ring $\mathcal{S}$ with probability one, where Ξ is called a *Boolean model* with intensity λ and generic grain Ξ_0. Notice that Boolean models form a special case of so–called *germ–grain models*, where the $X_1, X_2, \ldots$ are interpreted as germs and the $\Xi_1, \Xi_2, ...$ as (primary) grains.

Boolean models possess nice analytical properties; see e.g. [7] and [16]. In particular, their specific intrinsic volumes can be determined explicitly. The area fraction $p_\Xi = \overline{V}_2(\Xi)$ and the specific boundary length $L_\Xi = 2\overline{V}_1(\Xi)$ are given by

$$p_\Xi = 1 - \exp\bigl(-\lambda E A(\Xi_0)\bigr)\,, \qquad L_\Xi = \lambda\,(1 - p_\Xi) E U(\Xi_0)\,, \tag{5.1}$$

where $A(\Xi_0)$ $\bigl(= V_2(\Xi_0)\bigr)$ denotes the area of the generic grain Ξ_0 and $U(\Xi_0)$ $\bigl(= 2V_1(\Xi_0)\bigr)$ is its perimeter. If the generic grain Ξ_0 is convex and isotropic, i.e., its distribution is invariant with respect to rotations, then the specific Euler–Poincaré characteristic $\chi_\Xi = \overline{V}_0(\Xi)$ is given by

$$\chi_\Xi = (1 - p_\Xi)\Bigl(\lambda - \frac{\lambda^2}{4\pi}(EU(\Xi_0))^2\Bigr)\,;$$

see [1] and [6]. However, for anisotropic primary grains, the Euler–Poincaré characteristic χ_Ξ is more involved. Then,

$$\chi_\Xi = (1 - p_\Xi)\,(\lambda - A(M(\Xi_0), -M(\Xi_0)))\,, \tag{5.2}$$

where $A(M(\Xi_0), -M(\Xi_0))$ is the *mixed area* of the so–called *mean body* $M(\Xi_0)$. A detailed description of the meaning of formula (5.2) can be found e.g. in [13] and [18]. If the mean body $M(\Xi_0)$ is a convex polygon, then the mixed area is given by

$$A(M(\Xi_0), -M(\Xi_0)) = \frac{1}{2}\sum_{i=1}^{n} l_i h(M(\Xi_0), -n_i)\,,$$

where $l_1, \ldots, l_n$ are the lengths and $n_1, \ldots, n_n$ the outer unit normals to the edges of the polygon. For a convex body $K \in \mathcal{K}$, the *support function* $h(K, .)$ is given by $h(K, u) = \max_{x \in K}\langle x, u\rangle$ for $u \in \mathbb{R}^2$, where $\langle x, u\rangle = x_1 u_1 + x_2 u_2$ is the scalar product in $\mathbb{R}^2$.

In our numerical experiments, we used formula (5.2) in order to determine χ_Ξ for the following two special cases. If the generic grain Ξ_0 is a square $\Xi_0 = R_\alpha[0, a]^2$ with random side length a rotated by a deterministic angle α, then

$$\chi_\Xi = (1 - p_\Xi)\,\bigl(\lambda - \lambda^2 (E\, a)^2\bigr)\,;$$

see [3]. For a line segment $\Xi_0 = R_\alpha[0,a]$ with random length a and rotation angle α taking values α_0 and α_1 with probabilities 1/2, respectively, it can be shown that the specific Euler–Poincaré characteristic χ_Ξ is given by

$$\chi_\Xi = \lambda - \lambda^2 (E\,a)^2 \sin|\alpha_0 - \alpha_1|/4\,.$$

In the following, the computational results are discussed, which have been obtained for different Boolean models using the edge corrected algorithms described in Sections 3 and 4. In each case, 200 realizations of the corresponding Boolean model have been simulated in the observation window $W = [0,999]^2$, where the intensity λ is chosen to match the volume fraction of 0.2, 0.5, and 0.8, respectively. The simulated images were analyzed using the dilation radii $r_0 = 5$, $r_{i+1} = r_i + 1.3$, $i = 0,\ldots,49$ combined with the least squares method. The results presented in Tables 5.3–5.4 are the arithmetic means of estimated specific intrinsic volumes taken over 200 realizations. The mean estimated specific boundary length will be denoted by $\widetilde{L}$ and the mean estimated specific Euler–Poincaré characteristic by $\widetilde{\chi}$ if the edge correction (2.11)–(2.12) is used. In the case of the edge correction (2.13), the corresponding notation is $\widehat{L}^+$ and $\widehat{\chi}^+$. These values are compared with the theoretical values $L_\Xi = 2\overline{V_1}(\Xi)$ and $\chi_\Xi = \overline{V_0}(\Xi)$ as well as with the specific intrinsic volumes $L_{\Xi_{\mathbb{F}(G)}} = 2\overline{V_1}(\Xi_{\mathbb{F}(G)})$ and $\chi_{\Xi_{\mathbb{F}(G)}} = \overline{V_0}(\Xi_{\mathbb{F}(G)})$, where the primary grains Ξ_n are replaced by their polygonal approximations $\Xi_{n,\mathbb{F}(G)}$. In order to determine the area and the perimeter of the correspondingly approximated generic grain $\Xi_{0,\mathbb{F}(G)}$, the "marching squares"-like method described in Section 5.1 is used. Furthermore, the specific intrinsic volumes are estimated using the integral-geometric algorithms given e.g. in [8]. The resulting mean estimated specific boundary length and mean estimated specific Euler–Poincaré characteristic will be denoted by $\widehat{L}$ and $\widehat{\chi}$, respectively.

In this connection, we consider the following relative errors, that is, the relative deviations of the algorithmic values from the theoretical ones expressed in percent:

$$\begin{aligned}
\delta_{\widetilde{L},L_{\Xi_{\mathbb{F}(G)}}} &= \left(\widetilde{L}/L_{\Xi_{\mathbb{F}(G)}} - 1\right)\cdot 100\,, & \delta_{\widetilde{\chi},\chi_{\Xi_{\mathbb{F}(G)}}} &= \left(\widetilde{\chi}/\chi_{\Xi_{\mathbb{F}(G)}} - 1\right)\cdot 100\,,\\
\delta_{\widetilde{L},L_\Xi} &= \left(\widetilde{L}/L_\Xi - 1\right)\cdot 100\,, & \delta_{\widetilde{\chi},\chi_\Xi} &= (\widetilde{\chi}/\chi_\Xi - 1)\cdot 100\,,\\
\delta_{\widehat{L},L_{\Xi_{\mathbb{F}(G)}}} &= \left(\widehat{L}/L_{\Xi_{\mathbb{F}(G)}} - 1\right)\cdot 100\,, & \delta_{\widehat{\chi},\chi_{\Xi_{\mathbb{F}(G)}}} &= \left(\widehat{\chi}/\chi_{\Xi_{\mathbb{F}(G)}} - 1\right)\cdot 100\,,\\
\delta_{\widehat{L},L_\Xi} &= \left(\widehat{L}/L_\Xi - 1\right)\cdot 100\,, & \delta_{\widehat{\chi},\chi_\Xi} &= (\widehat{\chi}/\chi_\Xi - 1)\cdot 100\,,\\
\delta_{\widehat{L}^+,L_{\Xi_{\mathbb{F}(G)}}} &= \left(\widehat{L}^+/L_{\Xi_{\mathbb{F}(G)}} - 1\right)\cdot 100\,, & \delta_{\widehat{\chi}^+,\chi_{\Xi_{\mathbb{F}(G)}}} &= \left(\widehat{\chi}^+/\chi_{\Xi_{\mathbb{F}(G)}} - 1\right)\cdot 100\,,\\
\delta_{\widehat{L}^+,L_\Xi} &= \left(\widehat{L}^+/L_\Xi - 1\right)\cdot 100\,, & \delta_{\widehat{\chi}^+,\chi_\Xi} &= (\widehat{\chi}^+/\chi_\Xi - 1)\cdot 100\,.
\end{aligned}$$

For the following three types of examples, computations have been performed for 200 realizations of Boolean models with particular generic grains Ξ_0.

1. Spherical generic grain $\Xi_0 = B_a(o)$, where the radius $a \sim U[20,40]$ is distributed uniformly in the interval $[20,40]$.

2. Generic grain $\Xi_0 = R_\alpha[0,a]^2$ being a square of side length $a \sim U[20,40]$ rotated by an angle α. In total, one isotropic ($\alpha \sim U[0,\pi/2]$) and three anisotropic cases ($\alpha = 0, \pi/6, \pi/4$) have been considered.

3. Generic grain $\Xi_0 = R_\alpha[0, a]$ being a line segment with side length $a \sim U[20, 40]$. Likewise, we considered both isotropic grains ($\alpha \sim U[0, \pi]$) and the anisotropic *Manhattan model* (the distribution of the rotation angle α has two atoms at $\pi/12$ and $\pi/3$ with weights 1/2, i.e., $\alpha \sim 1/2\left(\delta_{\pi/12}(\cdot) + \delta_{\pi/3}(\cdot)\right)$).

In the first two examples, the intensity λ is chosen to match the volume fraction of 0.2, 0.5, and 0.8. In the last example, intensity values $\lambda = 0.008, 0.003490658, 0.0001$ ($\lambda = 0.008, 0.006285393, 0.0006285393$) have been used to cover the situations of negative, zero, and positive Euler-Poincaré numbers for the process of isotropic line segments and for the Manhattan model, respectively.

Due to the lack of space and the similarity of trends in estimation, the numerical results are given only for Boolean models with isotropic and anisotropic square grains ($\alpha = \pi/4$) of Example 2, see Tables 5.3 and 5.4, respectively. It can be seen from the tables that the use of G_{min} can not be recommended because of the substantial discretization error. Namely, the precision of the estimation methods proposed above depends on the goodness of the polygonal approximation of the boundary of the underlying unknown geometric object. Smooth boundary structure can be hardly approximated by G_{min} unless the grains of the germ–grain model are rectangles with sides parallel to the main directions of $\mathbb{L}^2$. In this case, our edge corrected algorithms perform pretty well both with G_{min} and G_{max}. This dependance on the quality of discretization can be seen clearly on the example of a Boolean model with square primary grains that have the same deterministic orientation angle α. The estimation is very precise for $\alpha = 0$ and then diminishes for $\alpha = \pi/6$ for both G_{min} and G_{max}. For $\alpha = \pi/4$, the accuracy of estimation is quite acceptable for $G_{\max}$ and miserable for $G_{\min}$.

On the opposite, the adjacency system $G_{\max}$ mostly provides the polygonal approximation of satisfactory quality. Numerical results show the significant growth of precision of the estimators $\widetilde{L}$, $\widetilde{\chi}$ and $\widehat{L}^+$, $\widehat{\chi}^+$ in comparison with $\widehat{L}$, $\widehat{\chi}$ (on average up to 10% of the relative error). Notice that the improvement of estimation is stronger for the specific Euler–Poincaré characteristic than for the boundary length. It is a general trend. It is also worth mentioning that the values $\widetilde{L}$, $\widetilde{\chi}$ are much closer to L_Ξ, χ_Ξ than to $L_{\Xi_{\mathbb{F}(G)}}$, $\chi_{\Xi_{\mathbb{F}(G)}}$. Vice versa, the values $\widehat{L}^+$, $\widehat{\chi}^+$ fit $L_{\Xi_{\mathbb{F}(G)}}$, $\chi_{\Xi_{\mathbb{F}(G)}}$ better.

The estimation methods remain practicable also for lower dimensional geometric structures such as the line–segment processes of Example 3 or the point processes in $\mathbb{R}^2$. Since the total boundary length increases in these cases, the run times of the algorithms increase as well, especially for the slow edge correction (2.11)–(2.12) based on the minus sampling. For point processes, the specific Euler number (i.e., the intensity of the point process) is reproduced by our algorithms quite well whereas the theoretical zero boundary length is not. Instead of it, the algorithms yield a quite random positive or negative output. The reason for that lies possibly in the high discretization error of the integral (2.10). Hence, straightforward counting methods seem to be more effective for point processes because of their simplicity.

Acknowledgements

The authors are grateful to Dietrich Stoyan for his valuable comments on the efficient choice of dilation radii. Furthermore, they would like to thank Ralph Guderlei for his help in implementing and testing the algorithms in Java.

G	$G_{\min}$	$G_{\max}$	$G_{\min}$	$G_{\max}$	$G_{\min}$	$G_{\max}$
p_Ξ	0.2	0.2	0.5	0.5	0.8	0.8
L_Ξ	0.022888	0.022888	0.044511	0.044511	0.041475	0.041475
$L_{\Xi_{\mathbb{F}(G)}}$	0.028811	0.023716	0.058168	0.047360	0.058311	0.046472
$\widetilde{L}$	0.018226	0.022690	0.036120	0.044440	0.033269	0.041903
$\delta_{\widetilde{L},L_\Xi}$, %	-20.37	-0.87	-18.85	-0.16	-19.79	1.03
$\delta_{\widetilde{L},L_{\Xi_{\mathbb{F}(G)}}}$, %	-36.74	-4.33	-37.90	-6.17	-42.95	-9.83
$\widehat{L}^+$	0.028674	0.023810	0.056379	0.046826	0.053679	0.044644
$\delta_{\widehat{L}^+,L_\Xi}$, %	25.28	4.03	26.66	5.20	29.42	7.64
$\delta_{\widehat{L}^+,L_{\Xi_{\mathbb{F}(G)}}}$, %	-0.47	0.40	-3.08	-1.13	-7.94	-3.93
$\widehat{L}$	0.022655	0.022655	0.043692	0.043692	0.039933	0.039933
$\delta_{\widehat{L},L_\Xi}$, %	-1.02	-1.02	-1.84	-1.84	-3.72	-3.72
$\delta_{\widehat{L},L_{\Xi_{\mathbb{F}(G)}}}$, %	-21.37	-4.47	-24.89	-7.74	-31.52	-14.07
$\chi_\Xi \times 10^4$	1.386668	1.386668	0.563821	0.563821	-3.349056	-3.349056
$\chi_{\Xi_{\mathbb{F}(G)}} \times 10^4$	1.131163	1.379504	-1.162705	0.432863	-7.901173	-4.017129
$\widetilde{\chi} \times 10^4$	2.692086	1.258333	2.977282	-0.286120	-0.998591	-4.968132
$\delta_{\widetilde{\chi},\chi_\Xi}$, %	94.14	-9.25	428.05	-150.75	-70.18	48.34
$\delta_{\widetilde{\chi},\chi_{\Xi_{\mathbb{F}(G)}}}$, %	137.99	-8.78	-356.07	-166.10	-87.36	23.67
$\widehat{\chi}^+ \times 10^4$	1.353650	1.225650	0.384400	-0.339450	-3.516950	-5.000850
$\delta_{\widehat{\chi}^+,\chi_\Xi}$, %	-2.38	-11.61	-31.82	-160.21	5.01	49.32
$\delta_{\widehat{\chi}^+,\chi_{\Xi_{\mathbb{F}(G)}}}$, %	19.67	-11.15	-133.06	-178.42	-55.49	24.49
$\widehat{\chi} \times 10^4$	1.379808	1.195991	0.433003	-0.403219	-3.485730	-5.070448
$\delta_{\widehat{\chi},\chi_\Xi}$, %	-0.49	-13.75	-23.20	-171.52	4.08	51.40
$\delta_{\widehat{\chi},\chi_{\Xi_{\mathbb{F}(G)}}}$, %	21.98	-13.30	-137.24	-193.15	-55.88	26.22

Table 5.3: Boolean model with isotropic square grains

G	$G_{\min}$	$G_{\max}$	$G_{\min}$	$G_{\max}$	$G_{\min}$	$G_{\max}$
p_Ξ	0.2	0.2	0.5	0.5	0.8	0.8
L_Ξ	0.022888	0.022888	0.044511	0.044511	0.041475	0.041475
$L_{\Xi_{\mathbb{F}(G)}}$	0.032148	0.022568	0.065477	0.045029	0.066769	0.044114
$\widetilde{L}$	0.013878	0.021603	0.027354	0.042629	0.025931	0.041438
$\delta_{\widetilde{L},L_\Xi}$, %	-39.37	-5.62	-38.54	-4.23	-37.48	-0.09
$\delta_{\widetilde{L},L_{\Xi_{\mathbb{F}(G)}}}$, %	-56.83	-4.27	-58.22	-5.33	-61.16	-6.07
$\widehat{L}^+$	0.031687	0.022710	0.062301	0.045258	0.059374	0.044245
$\delta_{\widehat{L}^+,L_\Xi}$, %	38.44	-0.78	39.97	1.68	43.16	6.68
$\delta_{\widehat{L}^+,L_{\Xi_{\mathbb{F}(G)}}}$, %	-1.43	0.63	-4.85	0.51	-11.08	0.30
$\widehat{L}$	0.021539	0.021539	0.041826	0.041826	0.038733	0.038733
$\delta_{\widehat{L},L_\Xi}$, %	-5.89	-5.89	-6.03	-6.03	-6.61	-6.61
$\delta_{\widehat{L},L_{\Xi_{\mathbb{F}(G)}}}$, %	-33.00	-4.56	-36.12	-7.11	-41.99	-12.20
$\chi_\Xi \times 10^4$	1.498411	1.498411	1.238828	1.238828	-1.888622	-1.888622
$\chi_{\Xi_{\mathbb{F}(G)}} \times 10^4$	1.160484	1.538032	-1.022177	1.418247	-7.773372	-1.763065
$\widetilde{\chi} \times 10^4$	3.881478	1.097168	5.847188	-1.165512	2.374416	-6.973096
$\delta_{\widetilde{\chi},\chi_\Xi}$, %	159.04	-26.78	371.99	-194.08	-225.72	269.22
$\delta_{\widetilde{\chi},\chi_{\Xi_{\mathbb{F}(G)}}}$, %	234.47	-28.66	-672.03	-182.18	-130.55	295.51
$\widehat{\chi}^+ \times 10^4$	1.481650	1.065450	1.168650	-1.253500	-1.986300	-7.014900
$\delta_{\widehat{\chi}^+,\chi_\Xi}$, %	-1.12	-28.89	-5.66	-201.18	5.17	271.43
$\delta_{\widehat{\chi}^+,\chi_{\Xi_{\mathbb{F}(G)}}}$, %	27.68	-30.73	-214.33	-188.38	-74.45	297.88
$\widehat{\chi} \times 10^4$	1.480372	1.062937	1.164503	-1.264403	-2.000587	-7.040211
$\delta_{\widehat{\chi},\chi_\Xi}$, %	-1.20	-29.06	-6.00	-202.06	5.93	272.77
$\delta_{\widehat{\chi},\chi_{\Xi_{\mathbb{F}(G)}}}$, %	27.57	-30.89	-213.92	-189.15	-74.26	299.32

Table 5.4: Boolean model with anisotropic square grains; $\alpha = \pi/4$

References

[1] Davy, P. J. (1976). Projected thick sections through multi-dimensional particle aggregates. *J. Appl. Probab.* **13,** 714–722. Correction: *J. Appl. Probab.* **15** (1978), 456.

[2] GeoStoch. Java library, Departments of Stochastics and Applied Information Processing, University of Ulm. http://www.geostoch.de 2003.

[3] Hall, P. (1988). *Introduction to the Theory of Coverage Processes.* J. Wiley & Sons, New York.

[4] Lorensen, W. and Cline, H. (1987). Marching cubes: a high resolution 3d surface reconstruction algorithm. *Computer Graphics* **21,** 163–169.

[5] Mecke, J., Schneider, R., Stoyan, D. and Weil, W. (1990). *Stochastische Geometrie.* Birkhäuser, Basel.

[6] Miles, R. (1976). Estimating aggregate and overall characterictics from thick sections by transmission microscopy. *J. Microscopy* **107,** 227–233.

[7] Molchanov, I. S. (1997). *Statistics of the Boolean Model for Practitioners and Mathematicians.* J. Wiley & Sons, Chichester.

[8] Ohser, J. and Mücklich, F. (2000). *Statistical Analysis of Microstructures in Materials Science.* J. Wiley & Sons, Chichester.

[9] Ohser, J., Nagel, W. and Schladitz, K. (2002). The Euler number of discretized sets – on the choice of adjacency in homogeneous lattices. In *Morphology of Condensed Matter.* ed. K. Mecke and D. Stoyan. vol. 600 of *Lect. Notes Physics.* Springer, Berlin pp. 275–298.

[10] Robins, V. (2002). Computational topology for point data: Betti numbers of α–shapes. In *Morphology of Condensed Matter.* ed. K. Mecke and D. Stoyan. vol. 600 of *Lect. Notes Physics.* Springer, Berlin pp. 261–274.

[11] Schmidt, V. and Spodarev, E. Joint estimators for the specific intrinsic volumes of stationary random sets. Preprint, University of Ulm 2003. (submitted).

[12] Schneider, R. (1993). *Convex Bodies. The Brunn–Minkowski Theory.* Cambridge University Press, Cambridge.

[13] Schneider, R. and Weil, W. (2000). *Stochastische Geometrie.* Teubner Skripten zur Mathematischen Stochastik. Teubner, Stuttgart.

[14] Schwandtke, A., Stoyan, D. and Schmidt, V. (1988). Improved estimation in planar sampling. *Acta Stereologica* **7,** 143–153.

[15] Soille, P. (2003). *Morphological Image Analysis.* 2nd ed. Springer, Heidelberg.

[16] Stoyan, D., Kendall, W. S. and Mecke, J. (1995). *Stochastic Geometry and its Applications* 2nd ed. J. Wiley & Sons, Chichester.

[17] Vogel, H. (1997). Digital unbiased estimation of the Euler–Poincaré characteristic in different dimensions. *Acta Stereologica* **16,** 97–104.

[18] Weil, W. (1995). The estimation of mean shape and mean particle number in overlapping particle systems in the plane. *Adv. Appl. Probab.* **27,** 102–119.

[19] Weil, W. and Wieacker, J. A. (1984). Densities for stationary random sets and point processes. *Adv. Appl. Probab.* **16,** 324–346.

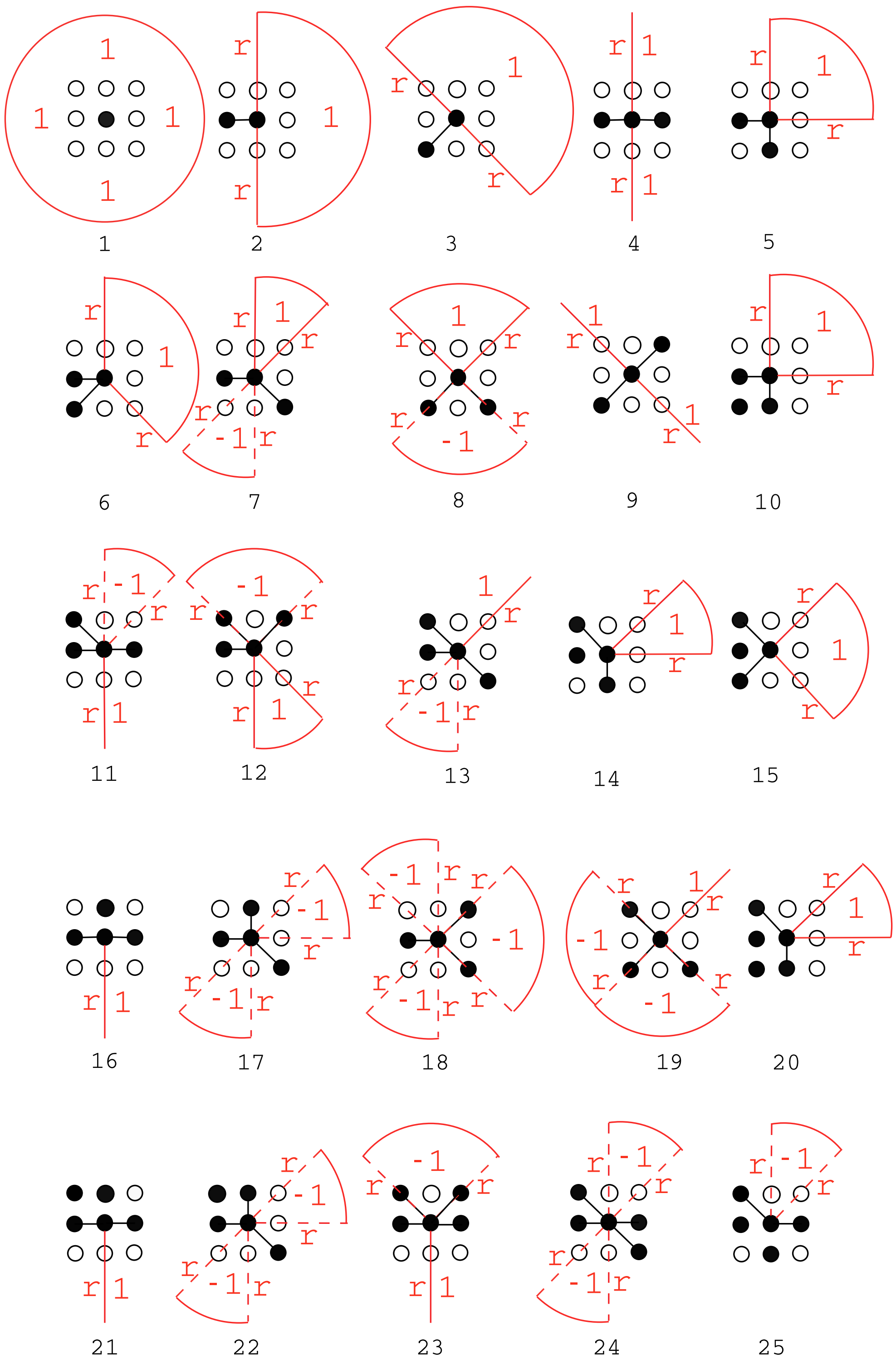

Figure 4.2: Polygonal approximation of neighborhood configurations for $\mathbb{F}(G_{\max})$

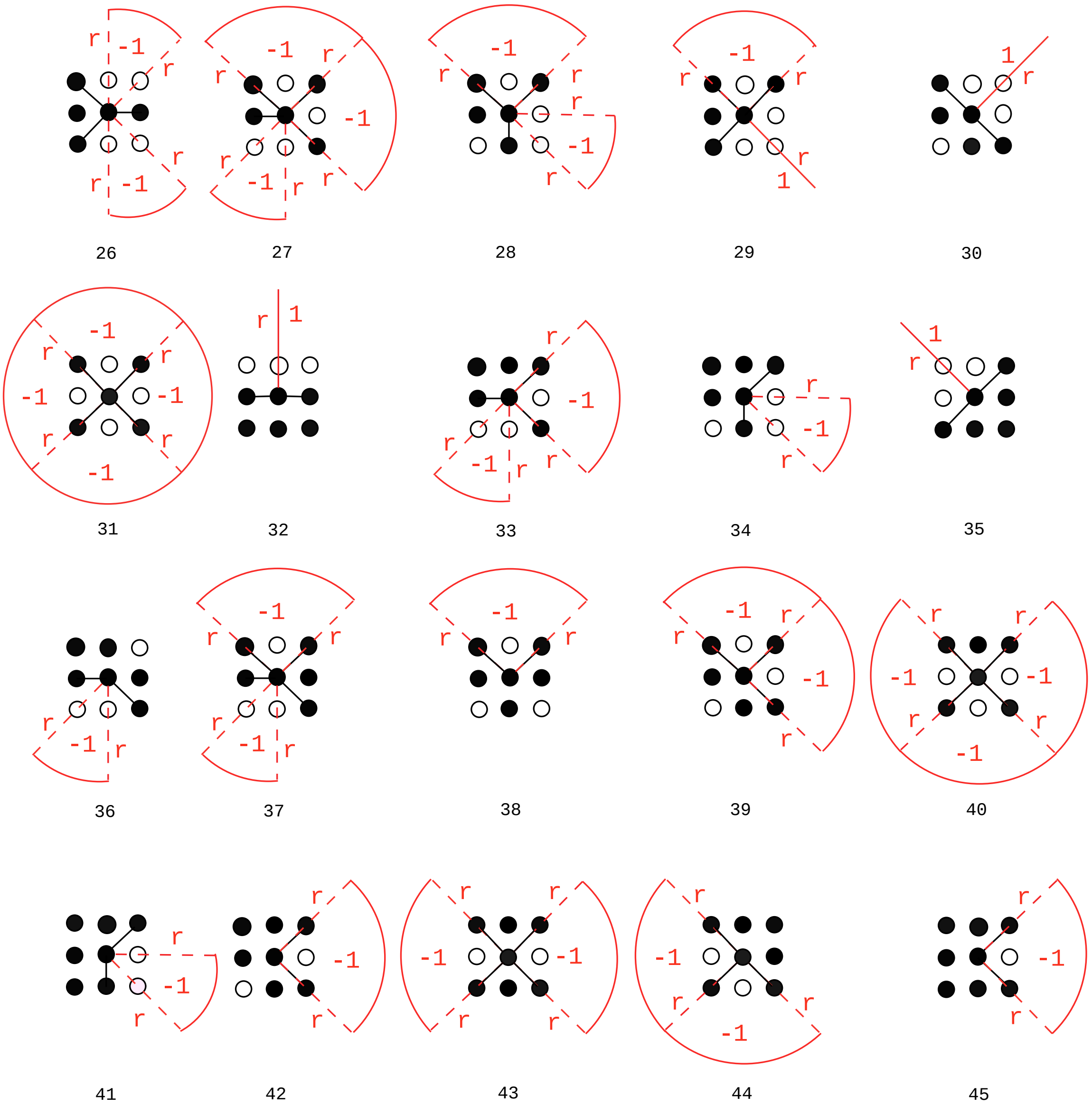

Figure 4.3: Polygonal approximation of neighborhood configurations for $\mathbb{F}(G_{\max})$ (continuation)

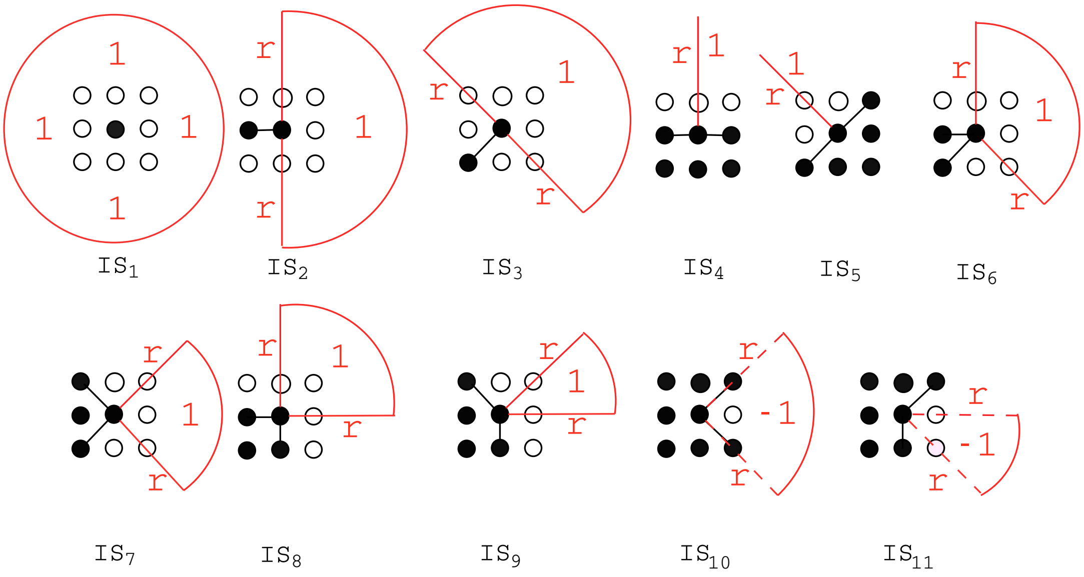

Figure 4.4: Boundary elements with their index sectors for $\mathbb{F}(G_{\max})$

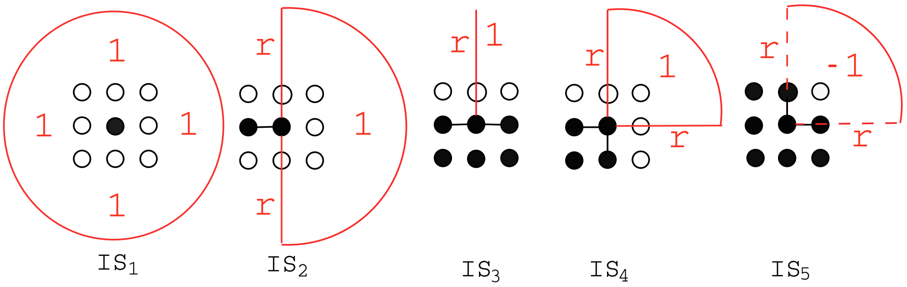

Figure 4.5: Boundary elements with their index sectors for $\mathbb{F}(G_{\min})$

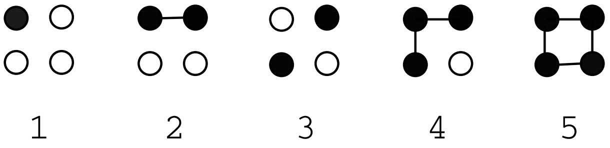

Figure 5.1: Basic squares for $\mathbb{F}(G_{\min})$

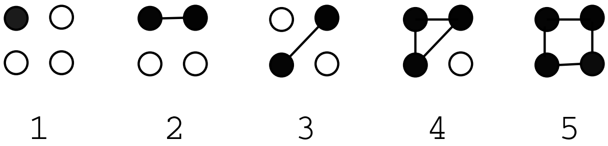

Figure 5.2: Basic squares for $\mathbb{F}(G_{\max})$

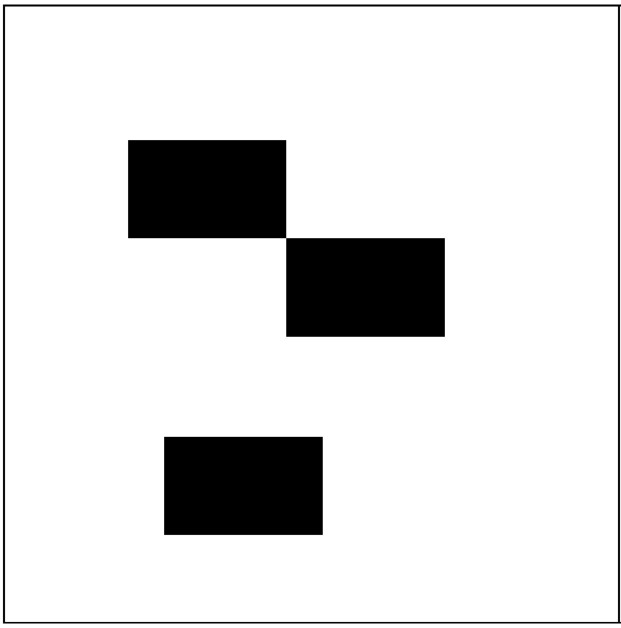

Figure 5.3: Union of non-overlapping rectangles

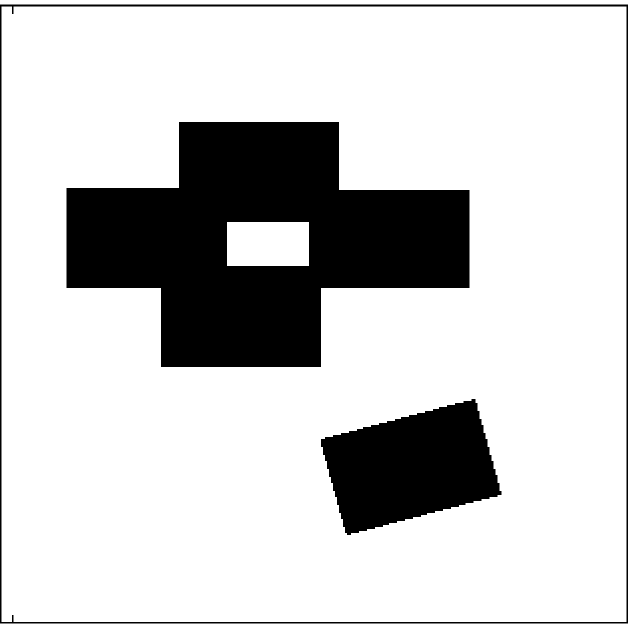

Figure 5.4: Rotated, overlapping rectangles

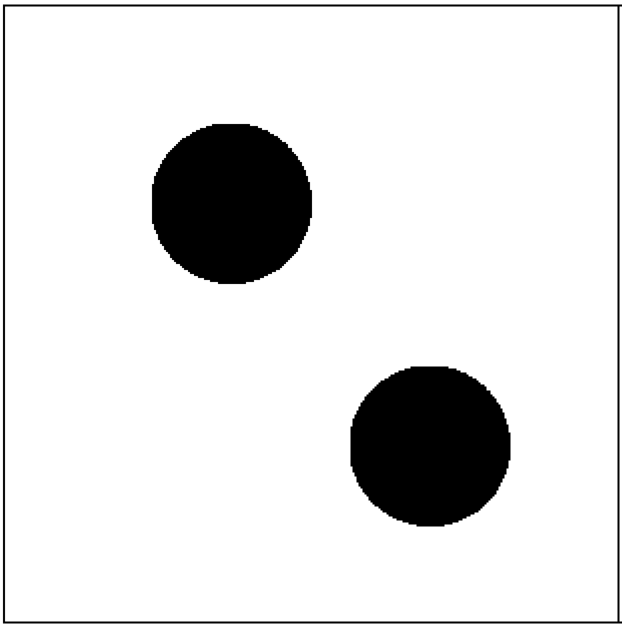

Figure 5.5: Unions of non-overlapping balls

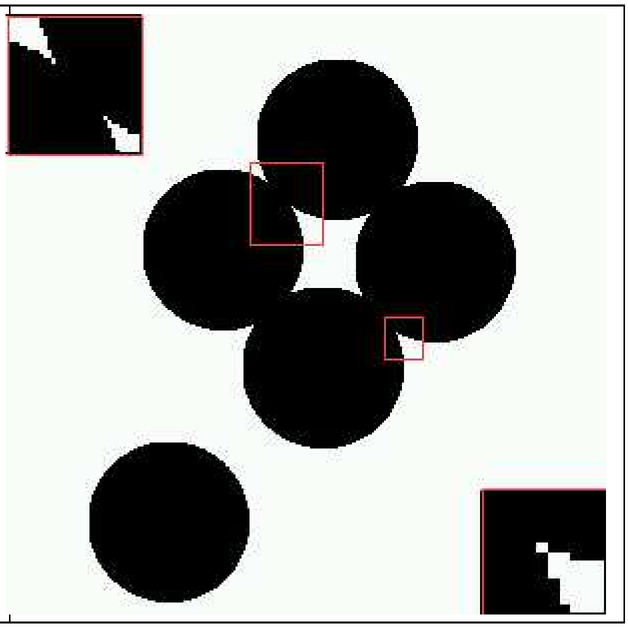

Figure 5.6: Overlapping balls

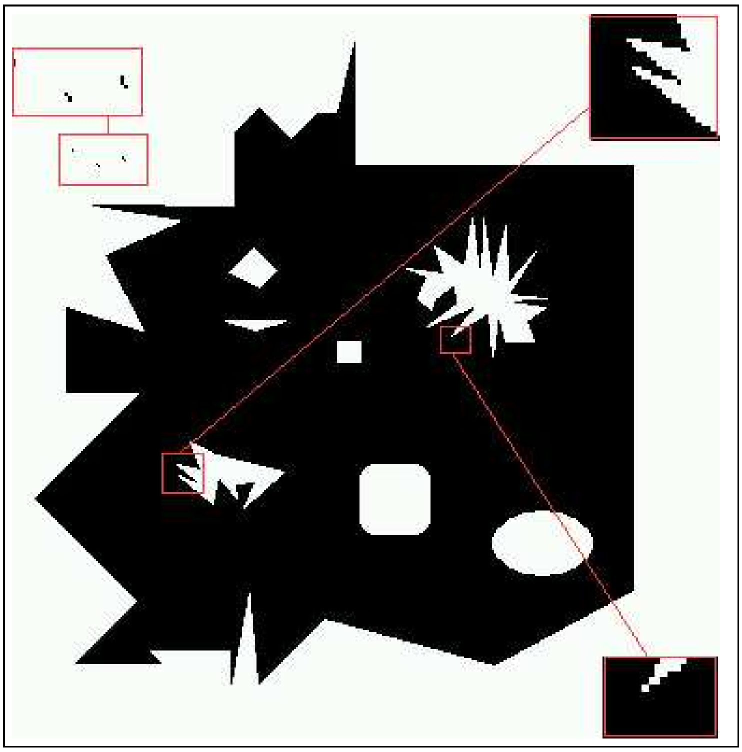

Figure 5.7: Image containing all possible boundary elements