A new algorithmic approach to the computation of Minkowski functionals of polyconvex sets

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

February 1, 2006

Abstract. An algorithm is proposed for the simultaneous computation of all Minkowski functionals (except for the volume) of sets from the convex ring in $\mathbb{R}^d$ discretized with respect to a given regular lattice. For this purpose, a polyhedral approximation is used to reconstruct their boundary structure. In the planar case $d = 2$, the performance and precision of the algorithm is studied on various examples of particular polyconvex sets. The algorithm is implemented in Java for two different approximation systems. The results of numerical experiments are compared with those obtained by other methods known in the literature.

Keywords. Quermaßintegral, intrinsic volume, discretization, volume, area, boundary length, Euler–Poincaré characteristic, porosity.

1 Introduction

Morphological characteristics of binary images such as volume, boundary area, curvature and connectivity number known as *Minkowski functionals*, *quermaßintegrals*, or *intrinsic volumes* are of great importance in geometry and image analysis; see e.g. [2], [8], [10]. 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 regular lattice. Likewise, in many cases, 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. [4], [5], [6], [13].

In the present paper, a new approach to the computation of Minkowski functionals for finite unions of convex sets (or, equivalently, *polyconvex sets*) is described. It leads to an algorithm that has the following advantages. First, unlike other related methods, it computes all Minkowski functionals (except for the volume) of a polyconvex set in $\mathbb{R}^d$ simultaneously. Thus, separate algorithms for the computation of each Minkowski functional are superfluous. Second, in numerical experiments for the planar case $d = 2$, our

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

algorithm showed high computational precision. Third, it is fast requiring only one single scan of the image. Finally, it is quite flexible since it possesses free parameters, which have the meaning of dilation radii.

The paper is organized as follows. In Section 2, some necessary preliminaries are given. In particular, the theoretical background for the algorithm is provided by an explicit extension of Steiner's formula to the convex ring; see Section 2.1. In Section 2.2, a short introduction into the discretization of continuous sets and their polyhedral approximation is given, because we suppose that the input image is given by a finite number of points and that no extra information about the image structure is available. In Section 3, the general idea of the algorithm is described. It is shown how a polyhedral approximation of the underlying continuous set is constructed and how the Minkowski functionals of this approximated set are computed. An upper bound on the computational error is given in Section 3.4. Some thoughts for an appropriate choice of the dilation radii are discussed in Section 3.6. The general framework of Section 3, which does not depend on dimension d, is then 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 $d = 3$ is briefly touched upon in Section 5. Notice that for the cases $d = 2, 3$, the algorithm has been implemented in Java and integrated into the GeoStoch library (see [1]), where the code has been tested on various examples of particular polyconvex sets. In Section 6, the results of numerical experiments are discussed and compared to those of conventional computation methods. For test purposes, polyconvex sets with known Minkowski functionals are used.

2 Preliminaries

2.1 Intrinsic volumes of polyconvex sets

Let $\mathcal{K}$ be the family of all compact, convex sets (or convex bodies) in $\mathbb{R}^d$, where $d \geqslant 2$ is an arbitrary fixed integer. Let $\mathcal{R}$ be the convex ring in $\mathbb{R}^d$, i.e. the family of all finite unions of convex bodies in $\mathbb{R}^d$. The elements of $\mathcal{R}$ will be referred to as *polyconvex sets*. By $V_d(K)$ we denote the d–dimensional 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 $x \in \mathbb{R}^d$. It is well known that nonnegative functionals $V_j : \mathcal{K} \to [0, \infty)$, $j = 0, \ldots, d$, 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 (j–dimensional) volume of the unit ball in $\mathbb{R}^j$, $j = 0, \ldots, d$. Notice that the functionals $V_j : \mathcal{K} \to [0, \infty)$ in (2.1) are called *intrinsic volumes*. Numbered in reverse order and properly normed, these functionals are also known as *Minkowski functionals* or *quermaßintegrals* $W_j : \mathcal{K} \to [0, \infty)$, where $W_j(K) = k_j\, V_{d-j}(K) \,/\, \binom{d}{j}$ for any $K \in \mathcal{K}$. Later on, we shall use intrinsic volumes because of convenience of notation. Intrinsic volumes have a nice geometric interpretation. In the 2D case, that is $d = 2$, $V_2(K)$ is the area and $2V_1(K)$ is the boundary length of K. In the 3D case, $V_3(K)$ is the usual volume, $2V_2(K)$ is the surface area and $V_1(K)/2$ is the mean breadth of $K \in \mathcal{K}$. For any $d \geqslant 2$, we have $V_0(K) = 1$ for any convex body $K \subset \mathbb{R}^d$.

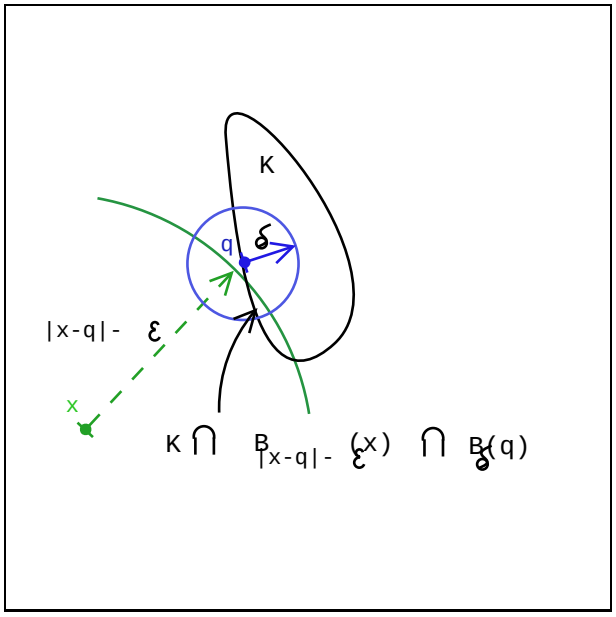

Figure 2.1: Computation of $J(K, q, x)$

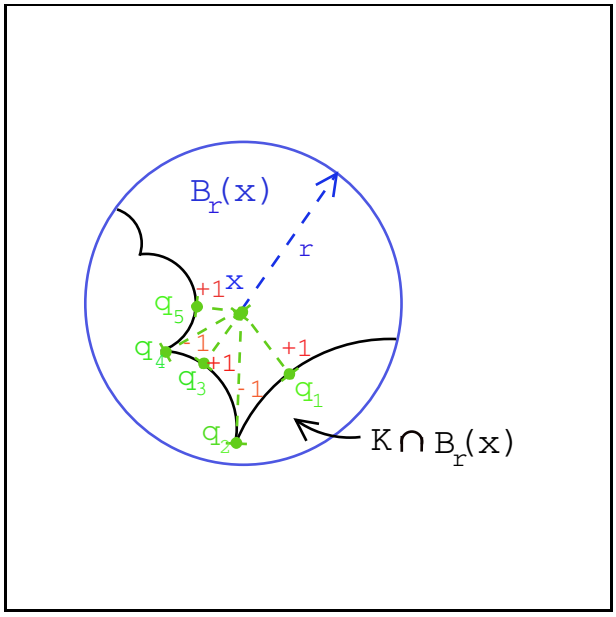

Figure 2.2: Computation of $I_r(K, x)$

For each $j = 0, \ldots, d$, there exists a unique additive extension of the functional V_j to the convex ring $\mathcal{R}$, given by the usual inclusion–exclusion technique; see e.g. formula (2) in [7]. Some of the intrinsic volumes (in particular, V_d and V_{d-1}) preserve their geometric interpretation, while others do not. For instance, $V_0(K)$ is equal to the *Euler–Poincaré characteristic* $\chi(K)$ of $K \in \mathcal{R}$, which describes the connectivity of the set K. In the 2D case, $\chi(K)$ is equal to the number of "clumps" minus the number of "holes" in K. In general, $\chi(K)$ can be represented as a linear combination of Betti numbers; see e.g. [6].

Our algorithm for the computation of intrinsic volumes is based on an explicit extension of Steiner's formula (2.1) to $\mathcal{R}$. This extension method makes use of the *index function* $J\left(K \cap B_r(x), q, x\right)$ defined by

$$J(K, q, x) = \begin{cases} 1 - \lim\limits_{\delta \to +0} \lim\limits_{\varepsilon \to +0} V_0\left(K \cap B_{|x-q|-\varepsilon}(x) \cap B_\delta(q)\right), & \text{if } q \in K, \\ 0, & \text{otherwise,} \end{cases} \tag{2.2}$$

for any $K \in \mathcal{R}$ and $q, x \in \mathbb{R}^d$. Figure 2.1 illustrates the computation of $J(K, q, x)$ for a simple convex body $K \subset \mathbb{R}^2$. Furthermore, for any $r > 0$, the functional $\rho_r : \mathcal{R} \to \mathbb{R}$ is defined by

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

for each $K \in \mathcal{R}$. Notice that the sum in (2.3) runs only over a finite set of boundary points q of K; see [8]. In Figure 2.2, the computation of $I_r(K, x)$ is illustrated. For the points q_1, q_3 and q_5 in Figure 2.2 that are projections of x on the boundary $\partial K \cap B_r(x)$ the index function is equal to 1, whereas it

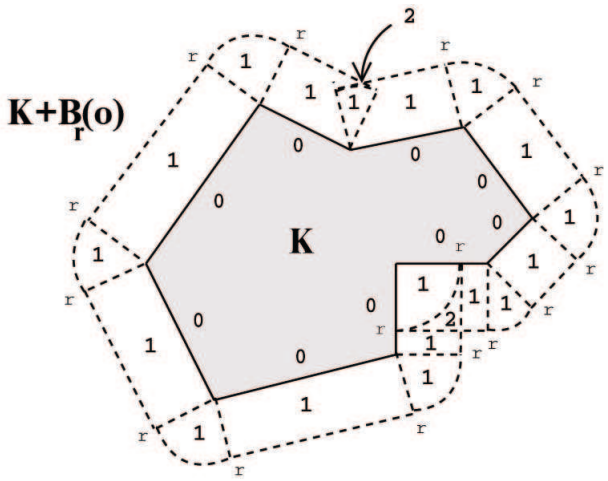

Figure 2.3: Volume of the parallel set counted with multiplicities

is equal to -1 for q_2 and q_4 that are concavity points of $\partial K \cap B_r(x)$. For all other $q \in K \cap B_r(x)$ the index function vanishes, i.e. $J(K \cap B_r(x), q, x) = 0$. Consequently, in this example, $I_r(K, x) = 1$ holds. For any $K \in \mathcal{K}$ we have $I_r(K, x) = \mathbf{1}(x \in (K \oplus B_r(o)) \setminus K)$ and therefore $\rho_r(K) = V_d((K \oplus B_r(o)) \setminus K)$. In general, $\rho_r(K)$ can be interpreted for $K \in \mathcal{R}$ as the volume of the parallel set $K \oplus B_r(o)$ counted with multiplicities; see Figure 2.3. Considering the weighted volume $\rho_r(K)$ instead of the usual volume $V_d(K \oplus B_r(o))$ of the parallel set $K \oplus B_r(o)$, the following extension of Steiner's formula (2.1) to the convex ring $\mathcal{R}$ holds (see e.g. [8], pp. 220–222):

$$\rho_r(K) = \sum_{j=0}^{d-1} r^{d-j} k_{d-j} V_j(K)\,, \qquad K \in \mathcal{R}\,. \tag{2.4}$$

For pairwise different radii $r_0, \ldots, r_{d-1} > 0$, we get d equations of the type (2.4) which form the following system of linear equations

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

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

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

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

2.2 Discretization and polyhedral approximation

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 $\mathbb{R}^d$, 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. The set $\mathbb{L}^d$ is called a *rectangular lattice* with lattice *spacings* $\Delta_1, \ldots, \Delta_d$. The *unit cell* L of the lattice is the Minkowski sum of the half-open segments $[o, u_1), \ldots, [o, u_d)$. In the following, we will consider cubic lattices $\mathbb{L}^d = \Delta\mathbb{Z}^d$, i.e., $\Delta_1 = \Delta_2 = \ldots = \Delta_d = \Delta > 0$. However, our results can be easily extended to the case of general rectangular lattices.

Consider an arbitrary set $K \in \mathcal{R}$. Since in many computer applications one deals with binary images that are represented by finite sets of black pixels on the white background, we assume that the *discretization* $K \cap \mathbb{L}^d$ of K with respect to the lattice $\mathbb{L}^d$ is given and any other extra information about K is not available. It is convenient to interpret $K \cap \mathbb{L}^d$ 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 sides 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 evaluate 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 to approximate K by a union of polytopes with vertices belonging to the grid $\mathbb{L}^d$.

To describe this approximation procedure formally, the following notation is useful. For any 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. Let $G = \{P_1, P_2, \ldots, P_n\}$ be a set polytopes with $P_i \subseteq \overline{L}$ and $\mathcal{F}^0(P_i) \subseteq \mathcal{F}^0(\overline{L})$, for $i = 1, \ldots, n$, where $\overline{L}$ denotes the topological closure of the lattice cell L. The set G is called a *generator* of the approximation. The invariance group of $\mathbb{L}^d$, i.e., the group of all rigid motions in $\mathbb{R}^d$ that map $\mathbb{L}^d$ onto itself is denoted by $\mathbb{T}^d$. Applying this group to the generator G results in the set

$$\mathcal{G} = \{T(P) :\ P \in G,\, T \in \mathbb{T}^d\}\,. \tag{2.7}$$

The *approximation system* $\mathbb{F}(G)$ with respect to the generator G is defined by $\mathbb{F}(G) = \bigcup_{k=0}^d \mathcal{F}^k(G)$, where $\mathcal{F}^k(G) = \bigcup_{Q\in\mathcal{G}} \mathcal{F}^k(Q)$ for each $k = 0, \ldots, d$.

For any $K \in \mathcal{R}$, the family of those elements of $\mathbb{F}(G)$ 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 $\mathbb{F}(G)$ 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 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 $\partial(K \cap \mathbb{L}^d)$.

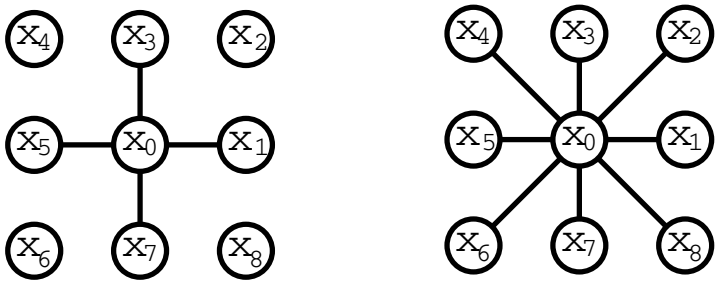

Figure 2.4: 4–neigborhood and 8–neighborhood

The above approximation method is bound to the connectivity or neighborhood relation on graphs. A *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

$$\gamma = \left\{\langle x, y\rangle : x, y \in \mathcal{F}^0(\overline{L}), (x, y) \subset L, [x, y] \subset \mathcal{F}^1(G)\right\} , \tag{2.8}$$

where (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 $N_\Gamma(x) = \{y \in \mathbb{L}^d : \langle x, y\rangle \in \bigcup_{z\in\mathbb{L}^d}(z+\gamma)\} \cup \{x\}$.

2.3 Examples for the planar case

In the following, two special approximation systems for the 2D case are described. Let $d = 2$ and 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 points $x_1, \ldots, x_k \in \mathbb{R}^2$.

The generator $G_{\max}$ of the so–called *maximal* approximation system $\mathbb{F}(G_{\max})$ consists only of one triangle, namely $G_{\max} = \{\mathrm{conv}\{l_0, l_1, l_2\}\}$. 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 [11]. 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 2.4. Roughly speaking, the approximation system $\mathbb{F}(G_{\max})$ contains all lattice points, all edges between neighboring lattice points and all triangles whose vertices are neighboring lattice points in the sense of the 8–neighborhood.

The minimal approximation system $\mathbb{F}(G_{\min})$ in $\mathbb{R}^2$ is generated by the closed lattice cell $\overline{L}$ of $\mathbb{L}^2$, i.e., $G_{\min} = \left\{\overline{L}\right\}$. It can be easily seen that $\mathbb{F}(G_{\min})$ yields the so–called 4–*neighborhood*; see Figure 2.4. That is, each point $x_0 \in \mathbb{L}^2$ has exactly $\nu = 4$ neighbors $x_1 = x_0 + l_1$, $x_3 = x_0 + l_3$, $x_5 = x_0 - l_1$, $x_7 = x_0 - l_3$.

Suppose that the discretization $K \cap \mathbb{L}^2$ of a 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 $K_{\mathbb{F}(G_{\max})}$, that is

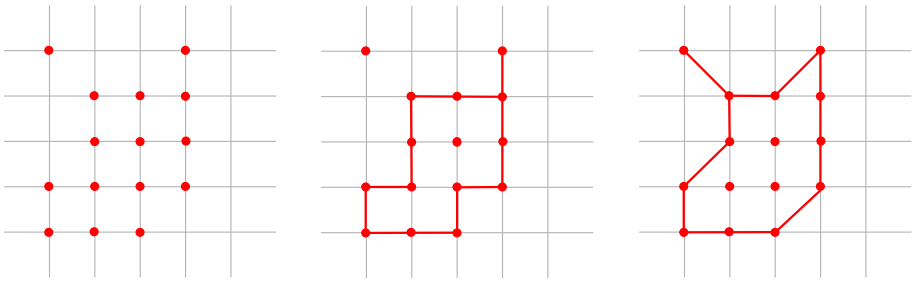

Figure 2.5: Approximation of a discretized set by $\mathbb{F}(G_{\min})$ and $\mathbb{F}(G_{\max})$ (left to right)

$q_0 \in \partial K_{\mathbb{F}(G_{\max})}$, 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 $K_{\mathbb{F}(G_{\min})}$, that is $q_0 \in \partial K_{\mathbb{F}(G_{\min})}$, 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 $\partial K_{\mathbb{F}(G_{\min})}$ or not.

Figure 2.5 shows a discretized set $K \cap \mathbb{L}^2$ as well as its approximations using $\mathbb{F}(G_{\min})$ and $\mathbb{F}(G_{\max})$, respectively. It is clear that the boundary structure of $K_{\mathbb{F}(G)}$ can look very different depending on the generator G and on the resolution Δ. In particular, significant changes can happen with respect to the connectivity of $K_{\mathbb{F}(G)}$ in comparison to K. Hence, any computation of intrinsic volumes of K based on an approximation $K_{\mathbb{F}(G)}$ is subject to a substantial approximation error. This phenomenon is well–known especially in the case of the Euler–Poincaré characteristic; see e.g. [10], p. 220. An upper bound on the approximation error is given in Section 3.4.

3 Algorithm

3.1 Basic idea and computational efficiency

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 $K \in \mathcal{R}$. In other words, an algorithm is described that approximates the vector $V(K)$ of intrinsic volumes on the basis of the polyhedral approximation $K_{\mathbb{F}(G)}$ of K defined in Section 2.2. Thus, instead of $\rho(K)$ and $V(K)$, we will compute the corresponding approximations $\rho(K_{\mathbb{F}(G)})$ and $V(K_{\mathbb{F}(G)})$, respectively.

Recall that formulae (2.4)–(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

$\rho(K_{\mathbb{F}(G)})$ whose components are given by (2.3). Then, the vector $V(K_{\mathbb{F}(G)}) = A^{-1}_{r_0\dots r_{d-1}}\rho(K_{\mathbb{F}(G)})$ can be easily determined, where $A^{-1}_{r_0\dots r_{d-1}}$ is the inverse of the matrix $A_{r_0\dots r_{d-1}}$ 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

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

or, equivalently,

$$R_r(K_{\mathbb{F}(G)}) = \Delta^d \sum_{x\in(\partial K_{\mathbb{F}(G)}\oplus B_r(o))\cap\mathbb{L}^d} \Big(\sum_{q\in\partial(K\cap\mathbb{L}^d)\setminus\{x\}} + \sum_{q\in\partial K_{\mathbb{F}(G)}\setminus\left(\partial(K\cap\mathbb{L}^d)\cup\{x\}\right)} \Big) J(K_{\mathbb{F}(G)}\cap B_r(x), q, x)\,,$$

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

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

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

$$D_r(P) = \{x \in \mathbb{L}^d :\, J(K_{\mathbb{F}(G)}\cap B_r(x), \tau_P(x), x) \neq 0,\ \tau_P(x) \in \operatorname{int}(P),\ 0 < |x - \tau_P(x)| \leqslant r\}$$

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) interior $\operatorname{int}(P)$ 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 $\operatorname{card} D_r(P)$ 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 not be 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) sampling window $W \cap \mathbb{L}^d$, where $K \subset W \subset \mathbb{R}^d$. Then, the direct computation of the sum (3.1) would require $O(m + l^2 r^d)$ operations for each radius r, whereas the fast algorithm based on (3.2) has complexity $O(m)$; see also Section 3.3.

3.2 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 $q \in \partial(K \cap \mathbb{L}^d)$ 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. [4], [11]. 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 the window $W \cap \mathbb{L}^d$.

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. 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

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

3.3 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 the approximation $R_r(K_{\mathbb{F}(G)})$ of $\rho_r(K_{\mathbb{F}(G)})$ using (3.6). Determine the corresponding approximation $R(K_{\mathbb{F}(G)}) = (R_{r_0}(K_{\mathbb{F}(G)}), \ldots, R_{r_{d-1}}(K_{\mathbb{F}(G)}))^\top$ of the vector $\rho(K_{\mathbb{F}(G)}) = (\rho_{r_0}(K_{\mathbb{F}(G)}), \ldots, \rho_{r_{d-1}}(K_{\mathbb{F}(G)}))^\top$.
7. Compute the approximation $\widetilde{V}(K) = A^{-1}_{r_0 \ldots r_{d-1}} R(K_{\mathbb{F}(G)})$ of $V(K)$.

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 $O(m)$.

3.4 Bound on the approximation error

Introduce the maximum norm in $\mathbb{R}^d$ by $\|x\| = \max_{0 \leqslant i \leqslant d-1} |x_i|$ for $x = (x_0, \ldots, x_{d-1}) \in \mathbb{R}^d$. For any $(d \times d)$–matrix $A = (a_{ij})$, the corresponding matrix norm is $\|A\| = \max_{0 \leqslant i \leqslant d-1} \sum_{j=0}^{d-1} |a_{ij}|$. For any $x \in \mathbb{R}^d$, let $C_x = x + [-\Delta/2, \Delta/2]^d$ be the d–dimensional cube with side length Δ and centroid x. The following theorem yields an upper bound on the approximation error $\|V(K) - \widetilde{V}(K)\|$, where $\widetilde{V}(K) = A^{-1}_{r_0 \ldots r_{d-1}} R(K_{\mathbb{F}(G)})$ and the components of $R(K_{\mathbb{F}(G)})$ are given by (3.6).

Proposition 3.1. *For any $0 < r_0 < r_1 < \ldots < r_{d-1}$, it holds*

$$\|V(K) - \widetilde{V}(K)\| \leqslant \|A^{-1}_{r_0 \ldots r_{d-1}}\| \, \|\rho(K) - R(K_{\mathbb{F}(G)})\| \,, \tag{3.7}$$

where

$$\|A^{-1}_{r_0 \ldots r_{d-1}}\| \leqslant \frac{d!\, r_1 r_2^2 \ldots r_{d-1}^{d-1}}{r_0 \prod\limits_{i>j} (r_i - r_j)} \tag{3.8}$$

and

$$\|\rho(K) - R(K_{\mathbb{F}(G)})\| \leqslant \max_{r \in \{r_0, \ldots, r_{d-1}\}} \left(\Delta^d \sigma_{1,r} \operatorname{card} L_{1,r} + \Delta^d \sigma_{2,r} \operatorname{card} L_{2,r} \right) . \tag{3.9}$$

Here,

$$\begin{aligned} L_{1,r} &= \left\{ x \in (\partial K \oplus B_{r+\sqrt{d}\Delta}(o)) \cap \mathbb{L}^d : \int_{C_x} (I_r(K,y) - I_r(K,x)) \, dy \neq 0 \right\} , \\ L_{2,r} &= \left\{ x \in (\partial K \oplus B_{r+2\sqrt{d}\Delta}(o)) \cap \mathbb{L}^d : I_r(K,x) - I_r(K_{\mathbb{F}(G)},x) \neq 0 \right\} \end{aligned}$$

and

$$\sigma_{1,r} = \max_{x \in L_{1,r}} \left(\max_{y \in C_x} I_r(K,y) - \min_{y \in C_x} I_r(K,y) \right) , \quad \sigma_{2,r} = \max_{x \in L_{2,r}} |I_r(K,x) - I_r(K_{\mathbb{F}(G)},x)| \,.$$

Proof. The inequality (3.7) immediately follows from $V(K) - \widetilde{V}(K) = A^{-1}_{r_0 \ldots r_{d-1}}(\rho(K) - R(K_{\mathbb{F}(G)}))$ and from well–known properties of matrix norms. In order to show that (3.8) holds, notice that Cramer's rule gives

$$\|A^{-1}_{r_0 \ldots r_{d-1}}\| = \frac{\max\limits_{0 \leqslant i \leqslant d-1} \sum\limits_{j=0}^{d-1} |D_{ji}|}{|\det A_{r_0 \ldots r_{d-1}}|} \leqslant \frac{d \max\limits_{0 \leqslant i,j \leqslant d-1} |D_{ji}|}{|\det A_{r_0 \ldots r_{d-1}}|} \,. \tag{3.10}$$

Here, $D_{ij} = (-1)^{i+j} \det A^{(ij)}_{r_0 \ldots r_{d-1}}$ is the so–called cofactor of the (i,j)th matrix element of $A_{r_0 \ldots r_{d-1}}$, where $A^{(ij)}_{r_0 \ldots r_{d-1}}$ is the matrix obtained by eliminating the ith row and the jth column of $A_{r_0 \ldots r_{d-1}}$. Furthermore, using linearity and antisymmetry of determinants one gets

$$\det A_{r_0 \ldots r_{d-1}} = k_1 \ldots k_d \, r_0 \ldots r_{d-1} \, (-1)^{\lfloor d/2 \rfloor} \det \widetilde{A}_{r_0 \ldots r_{d-1}} \,,$$

where $\lfloor d/2 \rfloor$ denotes the integer part of $d/2$ and $\widetilde{A}_{r_0 \ldots r_{d-1}}$ represents Vandermonde's matrix, i.e.,

$$\widetilde{A}_{r_0 \ldots r_{d-1}} = \begin{pmatrix} 1 & r_0 & \ldots & r_0^{d-1} \\ 1 & r_1 & \ldots & r_1^{d-1} \\ \ldots & \ldots & \ldots & \ldots \\ 1 & r_{d-1} & \ldots & r_{d-1}^{d-1} \end{pmatrix} .$$

Then, applying the formula for the determinant of Vandermonde's matrix, the following equation is obtained

$$\det A_{r_0 \ldots r_{d-1}} = k_1 \ldots k_d \, r_0 \ldots r_{d-1} \, (-1)^{\lfloor d/2 \rfloor} \prod_{i>j} (r_i - r_j) \,. \tag{3.11}$$

Similarly, the cofactor D_{ji} is equal to

$$D_{ji} = \prod_{l \neq i} k_l \prod_{l \neq j} r_l \, (-1)^{\lfloor (d-1)/2 \rfloor} \widetilde{D}_{ji} \,,$$

where $\widetilde{D}_{ji}$ is the (j,i)th cofactor of $\widetilde{A}_{r_0 \ldots r_{d-1}}$. Thus, by the upper bound $|\widetilde{D}_{ji}| \leqslant (d-1)!\, r_1 \ldots r_{d-1}^{d-1}$ and the inequalities $0 < r_0 < r_1 < \ldots < r_{d-1}$, we get that $|D_{ji}| \leqslant (d-1)!\, k_1 \ldots k_d \, r_1^2 r_2^3 \ldots r_{d-1}^d$ for $0 \leqslant i, j \leqslant d-1$. By (3.10) and (3.11), this shows that (3.8) holds. To prove the upper bound (3.9), notice that

$$\|\rho(K) - R(K_{\mathbb{F}(G)})\| = \max_{i=0,\ldots,d-1} |\rho_{r_i}(K) - R_{r_i}(K_{\mathbb{F}(G)})| \,.$$

Furthermore, for any $r > 0$, we have

$$\rho_r(K) - R_r(K_{\mathbb{F}(G)}) = \widetilde{\rho}_{r,1}(K, \mathbb{F}(G)) + \widetilde{\rho}_{r,2}(K, \mathbb{F}(G)) \,, \tag{3.12}$$

where

$$\widetilde{\rho}_{r,1}(K, \mathbb{F}(G)) = \int_{\partial K \oplus B_r(o)} I_r(K, y)\, dy - \Delta^d \sum_{x \in (\partial K \oplus B_r(o)) \cap \mathbb{L}^d} I_r(K, x)$$

is the error arising from the discretization of the integral given in (2.3) and

$$\widetilde{\rho}_{r,2}(K, \mathbb{F}(G)) = \Delta^d \sum_{x \in (\partial K \oplus B_r(o)) \cap \mathbb{L}^d} I_r(K, x) - \Delta^d \sum_{x \in (\partial K_{\mathbb{F}(G)} \oplus B_r(o)) \cap \mathbb{L}^d} I_r(K_{\mathbb{F}(G)}, x)$$

is the error of approximation of K by $K_{\mathbb{F}(G)}$. Notice that

$$\widetilde{\rho}_{r,1}(K, \mathbb{F}(G)) = \sum_{x \in (\partial K \oplus B_{r+\delta}(o)) \cap \mathbb{L}^d} \int_{C_x} \left(I_r(K, y) - I_r(K, x) \right) \, dy$$

and

$$\Big| \int_{C_x} \left(I_r(K, y) - I_r(K, x) \right) \, dy \Big| \leqslant \Delta^d \widetilde{I}_r(K, C_x) \,,$$

where $\widetilde{I}_r(K, C_x) = \max_{y \in C_x} I_r(K, y) - \min_{y \in C_x} I_r(K, y)$ and $\delta = \sqrt{d}\Delta$ is chosen to satisfy

$$\partial K \oplus B_r(o) \subset \bigcup_{x \in (\partial K \oplus B_{r+\delta}(o)) \cap \mathbb{L}^d} C_x \,.$$

This gives the bound $|\widetilde{\rho}_{r,1}(K, \mathbb{F}(G))| \leqslant \Delta^d \mathrm{card}\, L_{1,r} \max_{x \in L_{1,r}} \widetilde{I}_r(K, C_x)$. Similarly, for the second summand in (3.12), we have

$$\widetilde{\rho}_{r,2}(K, \mathbb{F}(G)) = \Delta^d \sum_{x \in (\partial K \oplus B_{r+2\delta}(o)) \cap \mathbb{L}^d} \big(I_r(K, x) - I_r(K_{\mathbb{F}(G)}, x) \big)$$

and, therefore,

$$|\widetilde{\rho}_{r,2}(K, \mathbb{F}(G))| \leqslant \Delta^d \mathrm{card}\, L_{2,r} \max_{x \in L_{2,r}} |I_r(K, x) - I_r(K_{\mathbb{F}(G)}, x)| \,.$$

This completes the proof. □

3.5 Asymptotic behaviour of the approximation error as $\Delta \downarrow 0$

Since K is the union of finitely many convex bodies, i.e. $K = \bigcup_{i=1}^m K_i$, it is not difficult to see that the quantity $\sigma_{1,r}$ introduced in Proposition 3.1 is uniformly bounded for $\Delta \downarrow 0$. Furthermore, the following inequality holds.

Proposition 3.2. *For each $r > 0$, there exists a constant $c < \infty$ such that for any $\Delta > 0$*

$$\operatorname{card} L_{1,r} \leqslant \frac{c}{\Delta^{d-1}} \,. \tag{3.13}$$

Proof. Because of the additivity of $I_r(.,x)$, the inclusion–exclusion principle yields

$$I_r(K,y) - I_r(K,x) = \sum_{i=1}^m (-1)^{i-1} \sum_{j_1<\ldots<j_i} \big[I_r(K_{j_1} \cap \ldots \cap K_{j_i}, y) - I_r(K_{j_1} \cap \ldots \cap K_{j_i}, x) \big] \,.$$

Thus, if $I_r(K_{j_1} \cap \ldots \cap K_{j_i}, y) - I_r(K_{j_1} \cap \ldots \cap K_{j_i}, x) = 0$ holds for all $y \in C_x$, for all i and for all $j_1 < \ldots < j_i$, then $\int_{C_x} \big(I_r(K,y) - I_r(K,x) \big)\, dy = 0$ is valid, too. Let $L = (\partial K \oplus B_{r+\sqrt{d}\Delta}(o)) \cap \mathbb{L}^d$. The complement of the set $L_{1,r}$ in L can be described by

$$\begin{aligned} L \setminus L_{1,r} &= \{x \in L : \int_{C_x} (I_r(K,y) - I_r(K,x))\, dy = 0\} \\ &\supseteq \bigcap_{i=1}^m \bigcap_{j_1<\ldots<j_i} \{x \in L :\, I_r(K_{j_1} \cap \ldots \cap K_{j_i}, y) = I_r(K_{j_1} \cap \ldots \cap K_{j_i}, x) \;\forall\, y \in C_x\} \,. \end{aligned}$$

That implies the following relation for $L_{1,r}$:

$$\begin{aligned} L_{1,r} &= (L \setminus L_{1,r})^c \cap L \\ &\subseteq \bigcup_{i=1}^m \bigcup_{j_1<\ldots<j_i} \{x \in L : \exists\, y \in C_x \text{ with } I_r(K_{j_1} \cap \ldots \cap K_{j_i}, y) \neq I_r(K_{j_1} \cap \ldots \cap K_{j_i}, x)\} \\ &\subseteq \bigcup_{i=1}^m \bigcup_{j_1<\ldots<j_i} \{x \in (\partial(K_{j_1} \cap \ldots \cap K_{j_i}) \oplus B_{r+\sqrt{d}\Delta}(o)) \cap \mathbb{L}^d : \exists\, y \in C_x \text{ with} \\ &\qquad\qquad I_r(K_{j_1} \cap \ldots \cap K_{j_i}, y) \neq I_r(K_{j_1} \cap \ldots \cap K_{j_i}, x)\} \,. \end{aligned}$$

From this it is easily seen that the cardinality of $L_{1,r}$ is bounded from above, i.e.,

$$\begin{aligned} \operatorname{card} L_{1,r} \leqslant \sum_{i=1}^m \sum_{j_1<\ldots<j_i} \operatorname{card} \{&x \in (\partial(K_{j_1} \cap \ldots \cap K_{j_i}) \oplus B_{r+\sqrt{d}\Delta}(o)) \cap \mathbb{L}^d : \exists\, y \in C_x \text{ with} \\ &I_r(K_{j_1} \cap \ldots \cap K_{j_i}, y) \neq I_r(K_{j_1} \cap \ldots \cap K_{j_i}, x)\} \,. \end{aligned}$$

Due to the convexity of $K_{j_1} \cap \ldots \cap K_{j_i}$, it is sufficient to show that

$$\operatorname{card} \{x \in (\partial K \oplus B_{r+\sqrt{d}\Delta}(o)) \cap \mathbb{L}^d : \exists\, y \in C_x \text{ with } I_r(K,y) \neq I_r(K,x)\} \leqslant c/\Delta^{d-1}$$

holds for a convex body $K \in \mathcal{K}$ and some constant $c < \infty$. Recall that for $K \in \mathcal{K}$ the computation of $I_r(K, x)$ simplifies to $I_r(K, x) = \mathbf{1}(x \in (K \oplus B_r(o)) \setminus K)$. Therefore, the inclusion

$$\{x \in (\partial K \oplus B_{r+\sqrt{d}\Delta}(o)) \cap \mathbb{L}^d : \exists\, y \in C_x \text{ with } I_r(K, y) \neq I_r(K, x)\} \\ \subseteq \{x \in \mathbb{L}^d : C_x \cap \partial K \neq \emptyset\} \cup \{x \in \mathbb{L}^d : C_x \cap \partial(K \oplus B_r(o)) \neq \emptyset\}$$

holds. The cardinality of the latter union of sets is bounded from above by

$$2 \cdot 2^d \cdot \frac{\sum_{i=1}^{d} \tau_{e_i^\perp}(K) + \tau_{e_i^\perp}(K \oplus B_r(o))}{\Delta^{d-1}} = \frac{c}{\Delta^{d-1}} \,,$$

where $e_1, \ldots, e_d$ is an orthonormal basis of $\mathbb{R}^d$ and $\tau_{e_i^\perp}(K)/\Delta^{d-1}$ is the number of lattice points lying in the orthogonal projection of K onto the $(d-1)$–dimensional plane perpendicular to e_i. Note that exactly 2^d cubes C_x share a common vertex in $\mathbb{R}^d$. □

Thus, by the uniform boundedness of $\sigma_{1,r}$, the inequality (3.13) implies that the first summand in (3.9) tends to zero as $\Delta \downarrow 0$. However, for general polyconvex sets K, the second summand $\Delta^d \sigma_{2,r} \operatorname{card} L_{2,r}$ of the upper bound in (3.9) need not converge to zero. This happens e.g. if the structural properties of ∂K are changed by the discretization of K with respect to the lattice $\mathbb{L}^d$. For instance, this is the case if the boundary of K contains lower dimensional parts such as isolated points, pieces of curves or surfaces, etc. that are not grasped by the sequence of lattices $\mathbb{L}^d$ with $\Delta \downarrow 0$. Another example of such problematic sets K is a union of at least two convex bodies touching each other in one single point that does not belong to any lattice from the lattice sequence; see Figure 6.1. However, for all polyconvex sets K satisfying the conditions

$$L_{2,r} \subseteq \partial K \oplus B_{r+2\sqrt{d}\Delta}(o) \setminus \partial K \oplus B_{r-h\Delta}(o) \tag{3.14}$$

for some constant $h > 0$, and

$$\sup_{\Delta > 0} \sigma_{2,r} < \infty \,, \tag{3.15}$$

the expression $\Delta^d \sigma_{2,r} \operatorname{card} L_{2,r}$ converges to zero as $\Delta \downarrow 0$. Indeed, it can be shown by similar arguments as in the proof of Proposition 3.2 that by (3.14) we have $\operatorname{card} L_{2,r} \leqslant c_1/\Delta^{d-1}$ for some constant $c_1 < \infty$ and for any $\Delta > 0$. Two–dimensional polyconvex sets K that satisfy (3.14) and (3.15) will be considered in detail in Section 4.4.

3.6 Appropriate choice of dilation radii

Computer experiments showed that the accuracy of the algorithm heavily depends on the choice of the d-tuple of dilation radii $r_0, \ldots, r_{d-1}$, where $\Delta < r_0 < r_1 < \ldots < r_{d-1}$. In particular, the error $\|V(K) - \widetilde{V}(K)\|$ is substantial for small radii $r_i \approx \Delta$. On the other hand, by (3.8), we have

$$\|A^{-1}_{r_0 \ldots r_{d-1}}\| \leqslant \frac{d!\, r_1 r_2^2 \ldots r_{d-1}^{d-1}}{r_0 \prod_{i>j} (r_i - r_j)} \longrightarrow 0$$

if $r_0 \to \infty$ and $r_i/r_{i+1} \leqslant 1-\varepsilon$ for all $i = 0, \ldots, d-2$ and some $\varepsilon > 0$. Furthermore, in many cases, the bound in (3.9) tends to zero as $\Delta \to 0$ provided that the radii $r_0, \ldots, r_{d-1}$ are fixed; see the discussion at the end of Section 3.4. Hence, one might expect that the accuracy of the algorithm is much better for larger r_i. Notice that there are no algorithmic restrictions on r_i from above. Thus, the largest value r_{d-1} of the radii r_i can be chosen in such a way that the run times of the algorithm are still acceptable. In practice, this could be $r_{d-1} \approx 10000$.

Moreover, the computational results can be significantly improved if, instead of taking d dilation radii $r_0, \ldots, r_{d-1}$, 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 polyconvex set $K \in \mathcal{R}$ is analyzed for $n \geqslant 1$ 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 ith d–tuple $r^{(i)}$ of radii; $i = 1, \ldots, n$. Then, the sample $(V^{(1)}(K_{\mathbb{F}(G)}), \ldots, V^{(n)}(K_{\mathbb{F}(G)}))$ 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. This leads to the following overdetermined system of linear equations, which corresponds to (2.5):

$$\begin{pmatrix} \rho_{r_0}(K_{\mathbb{F}(G)}) \\ \vdots \\ \rho_{r_{n-1}}(K_{\mathbb{F}(G)}) \end{pmatrix} = \begin{pmatrix} r_0^d k_d & r_0^{d-1} k_{d-1} & \ldots & r_0^2 k_2 & r_0 k_1 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ r_{n-1}^d k_d & r_{n-1}^{d-1} k_{d-1} & \ldots & r_{n-1}^2 k_2 & r_{n-1} k_1 \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_{d-1} \end{pmatrix}, \tag{3.16}$$

or, in matrix form, $\rho(K_{\mathbb{F}(G)}) = A_{r_0 \ldots r_{n-1}}\, x$, where $\rho(K_{\mathbb{F}(G)}) = \big(\rho_{r_0}(K_{\mathbb{F}(G)}), \ldots, \rho_{r_{n-1}}(K_{\mathbb{F}(G)})\big)^\top$ 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 (3.16) exactly. However, it is well known that the vector

$$V^*(K_{\mathbb{F}(G)}) = \big(A_{r_0 \ldots r_{n-1}}^\top A_{r_0 \ldots r_{n-1}}\big)^{-1} A_{r_0 \ldots r_{n-1}}^\top \rho(K_{\mathbb{F}(G)})$$

is the unique solution of the minimization problem

$$\big|\, \rho(K_{\mathbb{F}(G)}) - A_{r_0 \ldots r_{n-1}}\, V^*(K_{\mathbb{F}(G)}) \,\big| = \min_{x \in \mathbb{R}^d} \big|\, \rho(K_{\mathbb{F}(G)}) - A_{r_0 \ldots r_{n-1}}\, x \,\big|$$

and, therefore, can be regarded as an approximation of $V(K_{\mathbb{F}(G)}) = (V_0(K_{\mathbb{F}(G)}), \ldots, V_{d-1}(K_{\mathbb{F}(G)}))^\top$.

4 The planar case

In this section, the algorithm given above for general dimensions d and generators G will be illustrated in the 2D case for $G = G_{\min}$ and $G = G_{\max}$.

4.1 Neighborhood configurations of boundary points

We first consider the maximal approximation system $\mathbb{F}(G_{\max})$ 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. Altogether, there are $2^9 = 512$ possible neighborhood configurations for the 8–neighborhood. 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. 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 4.6–4.7. 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 2.4. After scanning the image, the frequencies of occurrence of these neighborhood configurations are summed up to $h_{S,7}$.

Considering the minimal approximation system $\mathbb{F}(G_{\min})$, 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 maximal approximation system $\mathbb{F}(G_{\max})$

4.2.1 Polygonal approximation

Consider the maximal approximation system $\mathbb{F}(G_{\max})$ in $\mathbb{R}^2$. Then, for each type of 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 4.6–4.7 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 4.8. 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 4.6–4.7. Thus, in accordance with (3.5), we can write

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

where J_j denotes the partial sum in (3.5) which corresponds to a boundary element of type $j = 1, \dots, 11$ 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.

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 $\mathbb{F}(G_{\max})$

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 4.8. Introduce the set

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

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 without one point ($j = 1$). By the definition

of the index function given in (2.2), 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* $IS_j = H(q) \cap \mathbb{L}^2 \cap B_r(q)$. For the 11 possible types of boundary elements, their index sectors are marked red in Figure 4.8. The dashed parts of the boundary do not belong to the index sectors whereas the solid parts do. Then, J_j can be rewritten as

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

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 $a > 0$.

j	$\operatorname{card} IS_j$	j	$\operatorname{card} IS_j$
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 $\mathbb{F}(G_{\max})$

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 3.2, 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 $N_\Gamma(y)$. 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 minimal approximation system $\mathbb{F}(G_{\min})$

The polygonal approximation of 50 neighborhood configurations on the basis of $\mathbb{F}(G_{\min})$ yields 5 different types of boundary elements; see Figure 4.9. 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

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

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

j	$\operatorname{card} IS_j$
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 $\mathbb{F}(G_{\min})$

the case of system $\mathbb{F}(G_{\max})$, we can write $S_{r,i} = \sum_{j=1}^{5} \omega_{ij} J_j$ for $i = 1, \ldots, 50$, where J_j denotes the partial sum in (3.5) which corresponds to a boundary element of type $j = 1, \ldots, 5$ 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 $J_j = j_0 \operatorname{card} IS_j$ are given in Table 4.5. All possible index sectors IS_j, $j = 1, \ldots, 5$ are marked red in Figure 4.9. 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.

4.4 Polyconvex sets with small discretization error

In this section, we give simple examples of two-dimensional polyconvex sets satisfying the conditions (3.14) and (3.15).

First of all, let us show that these conditions hold for any nonempty two–dimensional convex set K without lower dimensional parts, i.e., $K = \overline{\mathrm{int}(K)}$. Then, it holds for any $r \in \{r_0, \dots, r_d\}$ that

$$I_r(K, x) = 0, \quad x \in K \qquad \text{and} \qquad I_r(K_{\mathbb{F}(G_{\max})}, x) = 0, \quad x \in K_{\mathbb{F}(G_{\max})} \tag{4.3}$$

for all resolutions $\Delta > 0$. This implies that

$$I_r(K, x) = I_r(K_{\mathbb{F}(G_{\max})}, x), \quad x \in \mathbb{L}^2 \cap K\,. \tag{4.4}$$

Additionally, one can easily see that

$$\chi\big(K \cap B_r(x)\big) = \chi\big(K_{\mathbb{F}(G_{\max})} \cap B_r(x)\big), \quad x \in \mathbb{L}^2 \cap \big(K \oplus B_{r-h\Delta}(o)\big) \setminus K \tag{4.5}$$

holds for all $\Delta > 0$ and some $h > 0$. Thus, it follows from the relation $I_r(C, x) = \chi\big(C \cap B_r(x)\big)$ for any $C \in \mathcal{R}$ and $x \notin K$ (cf. [8], p. 224) that

$$I_r(K, x) = I_r(K_{\mathbb{F}(G_{\max})}, x), \quad x \in \mathbb{L}^2 \cap \big(K \oplus B_{r-h\Delta}(o)\big) \setminus K\,. \tag{4.6}$$

Hence, by (4.4) and (4.6), we have shown that $I_r(K, x) = I_r(K_{\mathbb{F}(G_{\max})}, x)$ for any $x \in \mathbb{L}^2 \cap \big(K \oplus B_{r-h\Delta}(o)\big)$, i.e., the inclusion (3.14) holds. As for condition (3.15), it is easily seen that

$$\sigma_{2,r} \leqslant \max\{1, \max_{x \in \mathbb{R}^2} \sup_{\Delta > 0} \chi\big(K_{\mathbb{F}(G_{\max})} \cap B_r(x)\big)\} < \infty.$$

Indeed, for each $x \in \mathbb{R}^2$, the Euler number $\chi\big(K_{\mathbb{F}(G_{\max})} \cap B_r(x)\big)$ is bounded as a function of Δ due to the convexity of K. Its maximum value is the maximum number of disconnected components of $K_{\mathbb{F}(G_{\max})} \cap B_r(x)$. Since K is convex, this number is uniformly bounded in $x \in \mathbb{R}^2$.

Suppose now that $K = K_1 \cup \ldots \cup K_n$ is a polyconvex set with nonempty convex components K_i satisfying the condition

$$K_{i_1} \cap \ldots \cap K_{i_k} = \overline{\mathrm{int}(K_{i_1} \cap \ldots \cap K_{i_k})}, \quad 1 \leqslant i_1 < \ldots < i_k \leqslant n, \quad k = 1, \dots, n. \tag{4.7}$$

It is clear that there are examples of polyconvex sets such that (4.7) does not hold; see Figure 6.1.

By induction on n, it can be proved that the condition (4.7) is sufficient for (3.14) and (3.15). Let us show this for $n = 2$. Using the additivity of I_r, we have

$$\begin{aligned} I_r(K, x) - I_r\big(K_{\mathbb{F}(G_{\max})}, x\big) &= I_r(K_1, x) - I_r\big((K_1)_{\mathbb{F}(G_{\max})}, x\big) + I_r(K_2, x) - I_r\big((K_2)_{\mathbb{F}(G_{\max})}, x\big) \\ - \Big(I_r(K_1 \cap K_2, x) - I_r\big((K_1 \cap K_2)_{\mathbb{F}(G_{\max})}, x\big)\Big) &- \Big(I_r\big((K_1 \cup K_2)_{\mathbb{F}(G_{\max})}, x\big) - I_r\big((K_1)_{\mathbb{F}(G_{\max})} \cup (K_2)_{\mathbb{F}(G_{\max})}, x\big)\Big) \\ &- \Big(I_r((K_1 \cap K_2)_{\mathbb{F}(G_{\max})}, x) - I_r\big((K_1)_{\mathbb{F}(G_{\max})} \cap (K_2)_{\mathbb{F}(G_{\max})}, x\big)\Big)\,. \end{aligned} \tag{4.8}$$

Since the sets K_1, K_2 and $K_1 \cap K_2$ are convex it follows from the above reasoning that the set of lattice points where the first three differences in the equation (4.8) are not zero is "thin" in the sense of condition (3.14).

The boundaries of $(K_1 \cup K_2)_{\mathbb{F}(G_{\max})}$ and $(K_1)_{\mathbb{F}(G_{\max})} \cup (K_2)_{\mathbb{F}(G_{\max})}$ as well as of $(K_1 \cap K_2)_{\mathbb{F}(G_{\max})}$ and $(K_1)_{\mathbb{F}(G_{\max})} \cap (K_2)_{\mathbb{F}(G_{\max})}$ can slightly differ only in the vicinity of the points in $\partial K_1 \cap \partial K_2$ where this difference becomes smaller with decreasing Δ. Arguing similarly as in the proof of (4.3) and (4.5) one can show that the domain of lattice points $x \in \mathbb{L}^2$ where this difference affects the values of $I_r(\cdot, x)$ is also "thin" in the sense of condition (3.14). Hence, one can conclude that the condition (3.14) is satisfied for $K = K_1 \cup K_2$.

As for the condition (3.15), it follows from (4.8) that

$$\begin{aligned}
\sup_{\Delta>0} \max_{x \in L_{2,r}} |I_r(K,x) - I_r\big(K_{\mathbb{F}(G_{\max})}, x\big)| \leqslant \sup_{\Delta>0} \max_{x \in L_{2,r}} \Big| I_r(K_1,x) - I_r\big((K_1)_{\mathbb{F}(G_{\max})}, x\big) \Big| \\
+ \sup_{\Delta>0} \max_{x \in L_{2,r}} \Big| I_r(K_2,x) - I_r\big((K_2)_{\mathbb{F}(G_{\max})}, x\big) \Big| + \sup_{\Delta>0} \max_{x \in L_{2,r}} \Big| I_r(K_1 \cap K_2,x) - I_r\big((K_1 \cap K_2)_{\mathbb{F}(G_{\max})}, x\big) \Big| \\
+ \sup_{\Delta>0} \max_{x \in L_{2,r}} \Big| I_r\big((K_1 \cup K_2)_{\mathbb{F}(G_{\max})}, x\big) - I_r\big((K_1)_{\mathbb{F}(G_{\max})} \cup (K_2)_{\mathbb{F}(G_{\max})}, x\big) \Big| \\
+ \sup_{\Delta>0} \max_{x \in L_{2,r}} \Big| I_r((K_1 \cap K_2)_{\mathbb{F}(G_{\max})}, x\big) - I_r\big((K_1)_{\mathbb{F}(G_{\max})} \cap (K_2)_{\mathbb{F}(G_{\max})}, x\big) \Big| ,
\end{aligned}$$

where the first three terms in the right–hand side are finite due to the convexity of K_1, K_2 and $K_1 \cap K_2$. The last two terms are finite as well. The proof is similar to the convex case. Thus, the condition (3.15) holds for $K = K_1 \cup K_2$. Hence, for polyconvex sets satisfying (4.7) the expression $\Delta^d \sigma_{2,r} \operatorname{card} L_{2,r}$ in the upper bound in (3.9) converges to zero as $\Delta \downarrow 0$; see Section 3.5.

5 Challenge of three dimensions

In the 3D case, the implementation of the algorithm gets more complex. The main reason is the necessity to deal with 26–neighborhoods instead of 8–neighborhoods in the 2D case. This implies that the number of possible voxel neighborhood configurations is equal to $2^{27} = 134217728$.

For a detailed analysis of neighborhood voxel configurations, each of them can be modelled separately by a graph. Then, the depth–first search algorithm (see [9]) is used to detect connected components of white or black voxels within the configuration. If white connected components of all neighborhood voxel configurations are known, all 324454 different boundary elements for $\mathbb{F}(G_{\max})$ can be determined. Unlike the 2D case with 11 different boundary elements, it is impossible to give a direct formula for the cardinality of the index sector for each element explicitly — this has to be done automatically using a complex algorithm containing strategies like backtracking, depth–first search, etc. In the 2D case, we have $J(K_{\mathbb{F}(G_{\max})}, q, x) = j_0 \in \{1, -1\}$ for each x of the index sector. In three dimensions, the index function takes values $1, 0, -1, -2, -3$. Again, the correct value must be determined automatically for each boundary element.

In the 2D case, there are only two different types of segments of the boundary for $\mathbb{F}(G_{\max})$. In three dimensions, three different types of segments (edges, face diagonals and spatial diagonals of the unit cell) as well as three different types of triangles forming the surface of the polytope $K_{\mathbb{F}(G_{\max})}$ must be taken into consideration for the calculation of the sum M_r.

Similarly to the two–dimensional case, the 3D–algorithm yields precise results if the dilation radii are chosen large enough. However, the obvious drawback is its complexity that makes software tests much more difficult. A remedy for this can be the use of another related method based on the principle kinematic formula of integral geometry as considered in [12].

6 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 test purposes, we use planar polyconvex sets with known Minkowski functionals.

We consider several examples of polyconvex sets $K \subset \mathbb{R}^2$ 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. 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 3.6 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 minimal and the maximal approximation system, 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 minimal approximation system $\mathbb{F}(G_{\min})$, 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 maximal approximation system $\mathbb{F}(G_{\max})$, 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 [3]. In Figures 6.5 and 6.6, 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 6.1. 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.

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 numerical examples considered below, 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 6.1–6.7, 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 6.2, where these values are rounded up to the 6th digit. For the algorithm described in Sections 3 and 4, we used the dilation radii $r_0 = 5000$, $r_{i+1} = r_i + 20.3$, $i = 0, \ldots, 999$ combined with the least–squares method.

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 6.1: Weights for the basic types of squares

Figure	G	$2\,V_1(K_{\mathbb{F}(G)})$	$V_0(K_{\mathbb{F}(G)})$	$2\,V_1^*(K_{\mathbb{F}(G)})$	$V_0^*(K_{\mathbb{F}(G)})$
6.1	$G_{\min}$	756.0	3	755.998293	3.0
6.1	$G_{\max}$	758.828427	2	758.827287	2.0
6.2	$G_{\min}$	1084.0	1	1083.99943	1.0
6.2	$G_{\max}$	1040.651804	1	1040.651169	1.0
6.3	$G_{\min}$	632.0	2	631.998862	2.0
6.3	$G_{\max}$	528.901587	2	528.900294	2.0
6.4	$G_{\min}$	1346.0	1	1345.99943	1.0
6.4	$G_{\max}$	1151.217388	-2	1151.218213	-2.0
6.7	$G_{\min}$	3428.0	0	3427.999997	4.009811E-11
6.7	$G_{\max}$	2970.584053	-5	2970.586176	-5.0

Table 6.2: 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 6.1 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 6.4 produces three little holes of side length Δ at the intersection points of their bounding circles. For convenience, in Figure 6.4, 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 6.7, a discretized set is considered that contains the five possible boundary elements for $\mathbb{F}(G_{\min})$, which have been described in Figure 4.9, and the 11 possible boundary elements for $\mathbb{F}(G_{\max})$ given in Figure 4.8. If the computations for the image in Figure 6.7 are based on the minimal approximation system $\mathbb{F}(G_{\min})$, 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.

Acknowledgements

The authors are grateful to Kurt Mehlhorn, Dietrich Stoyan, and the anonymous referee for their suggestions that improved the paper. Furthermore, they would like to thank Ralph Guderlei for his help in

implementing and testing the algorithms in Java.

References

[1] GeoStoch (2006). Java library, Departments of Stochastics and Applied Information Processing, University of Ulm. http://www.geostoch.de.

[2] Gruber, P.M. and Wills, J.M. (1993) *Handbook of Convex Geometry.* North–Holland, Amsterdam.

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

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

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

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

[7] Schmidt, V. and Spodarev, E. (2005). Joint estimators for the specific intrinsic volumes of stationary random sets. *Stochastic Processes and their Applications* **115**, 959–981.

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

[9] Segewick, R. (1992). *Algorithmen in C.* Addison-Wesley, Bonn.

[10] Serra, J. (1982). *Image Analysis and Mathematical Morphology.* Academic Press, London.

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

[12] Spodarev, E. and Schmidt, V. (2005). On the local connectivity number of stationary random closed sets. In: C. Ronse, L. Najman, and E. Fernandière (eds.) *Proceedings to the 7th International Symposium on Mathematical Morphology.* Kluwer, Dordrecht, 343–356.

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

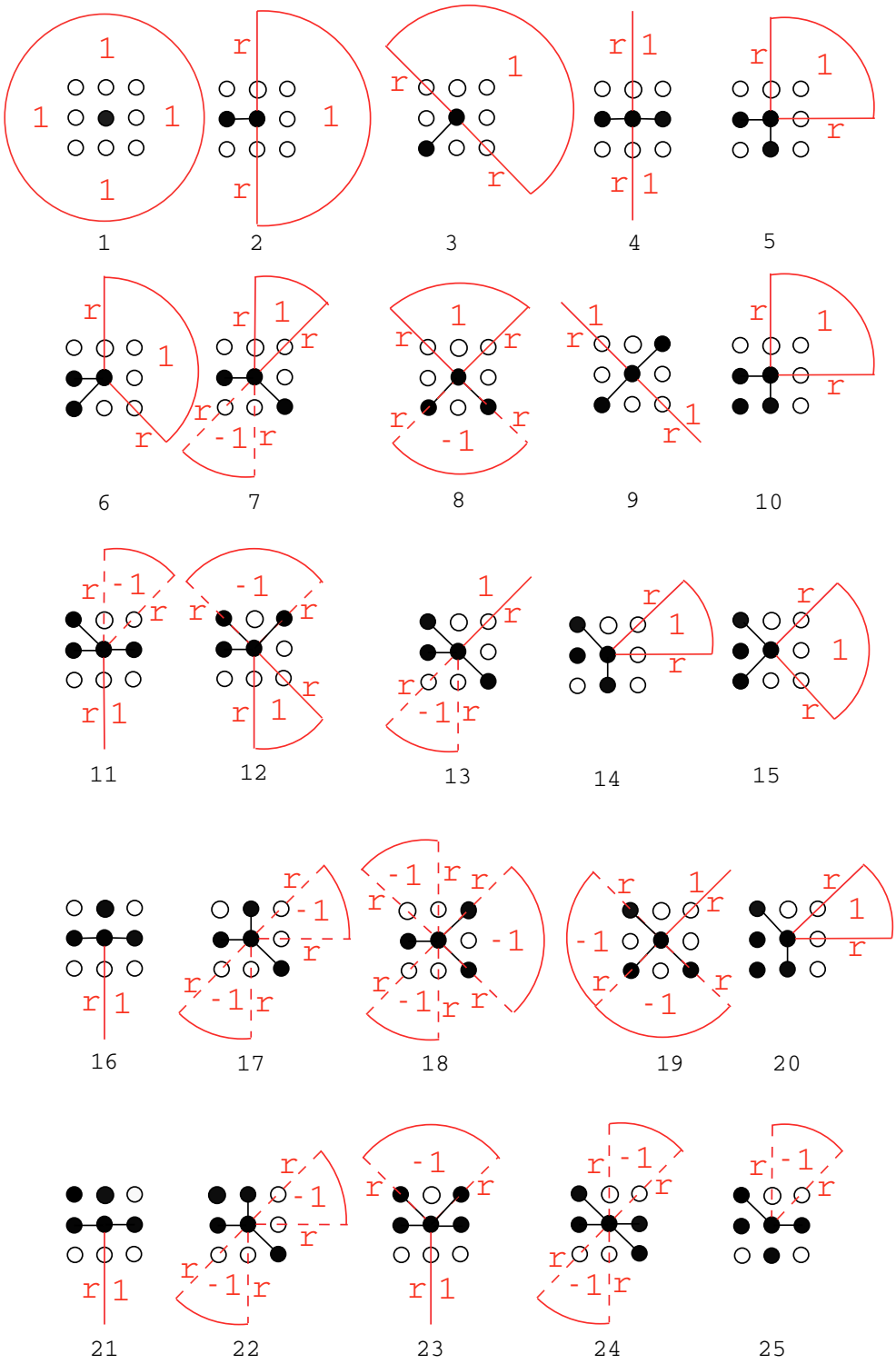

Figure 4.6: Polygonal approximation of neighborhood configurations for $\mathbb{F}(G_{\max})$

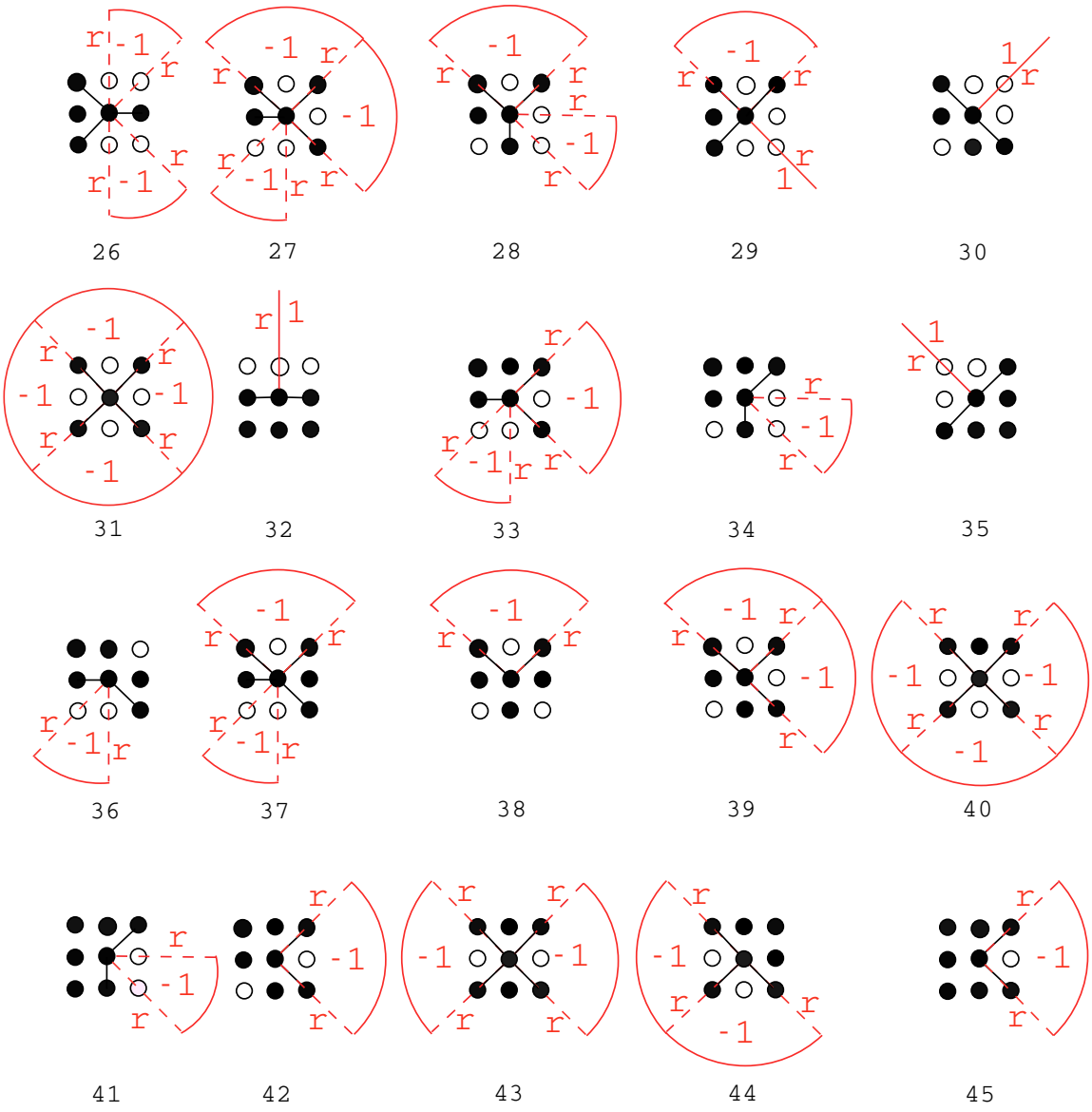

Figure 4.7: Polygonal approximation of neighborhood configurations for $\mathbb{F}(G_{\max})$ (continuation)

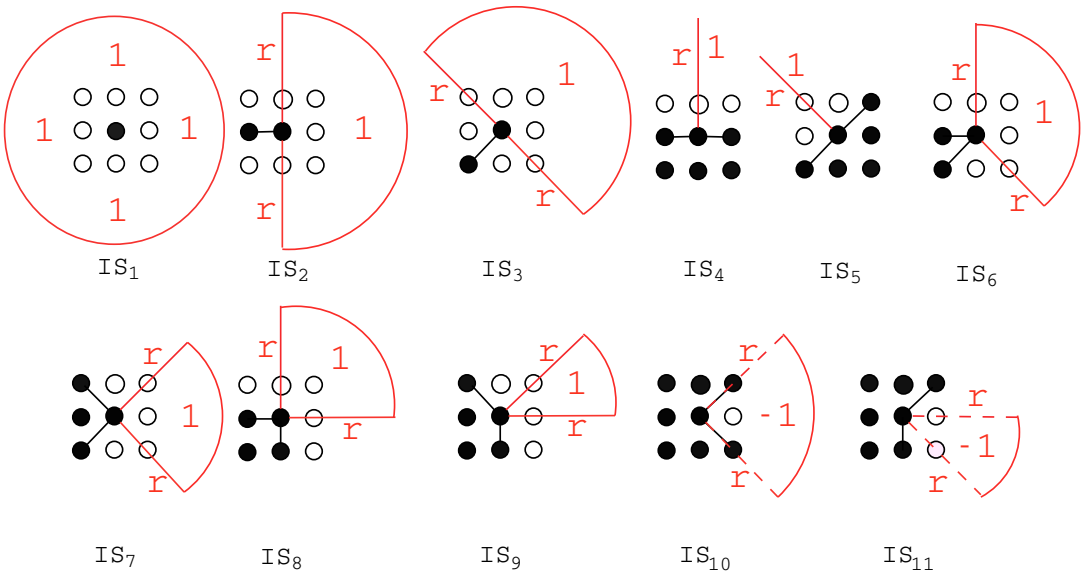

Figure 4.8: Boundary elements with their index sectors for $\mathbb{F}(G_{\max})$

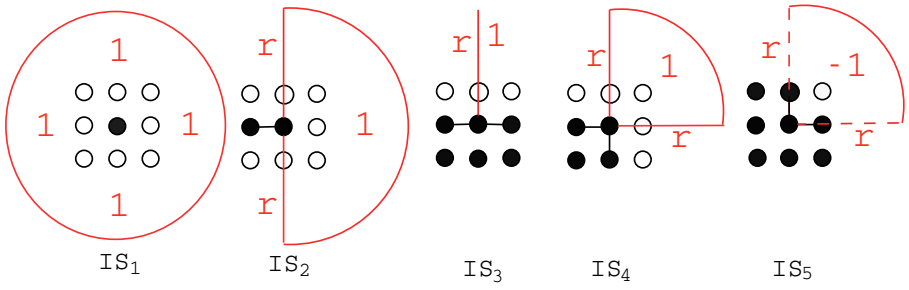

Figure 4.9: Boundary elements with their index sectors for $\mathbb{F}(G_{\min})$

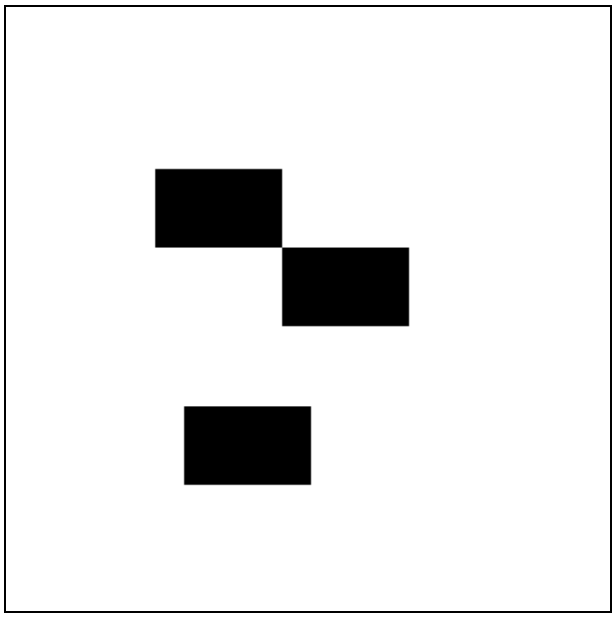

Figure 6.1: Union of non-overlapping rectangles

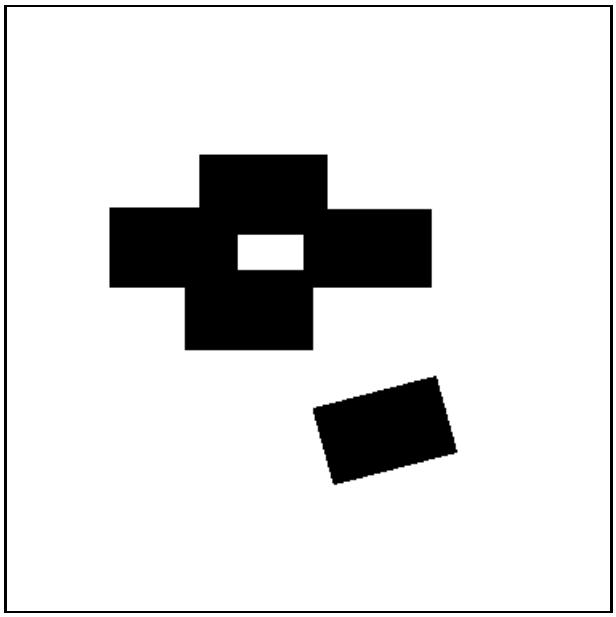

Figure 6.2: Rotated, overlapping rectangles

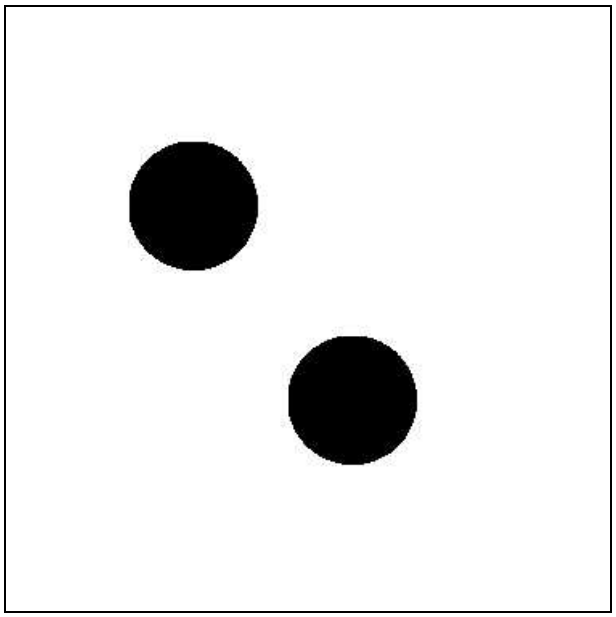

Figure 6.3: Unions of non-overlapping balls

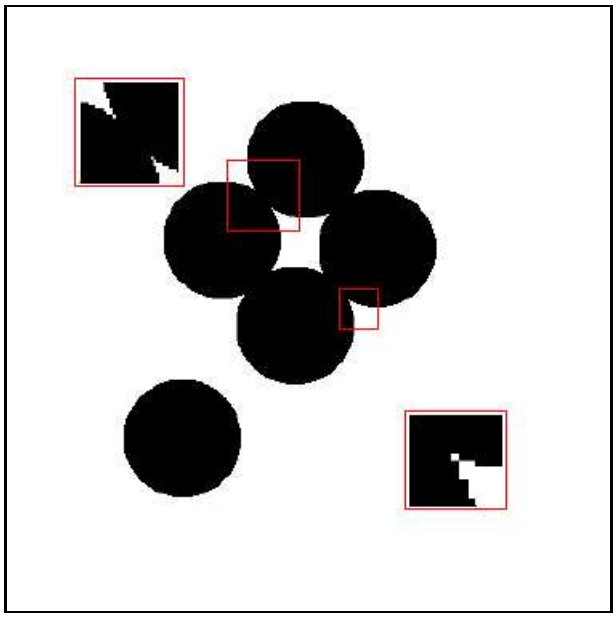

Figure 6.4: Overlapping balls

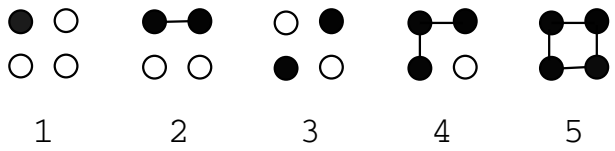

Figure 6.5: Basic squares for $\mathbb{F}(G_{\min})$

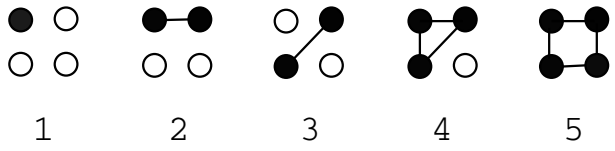

Figure 6.6: Basic squares for $\mathbb{F}(G_{\max})$

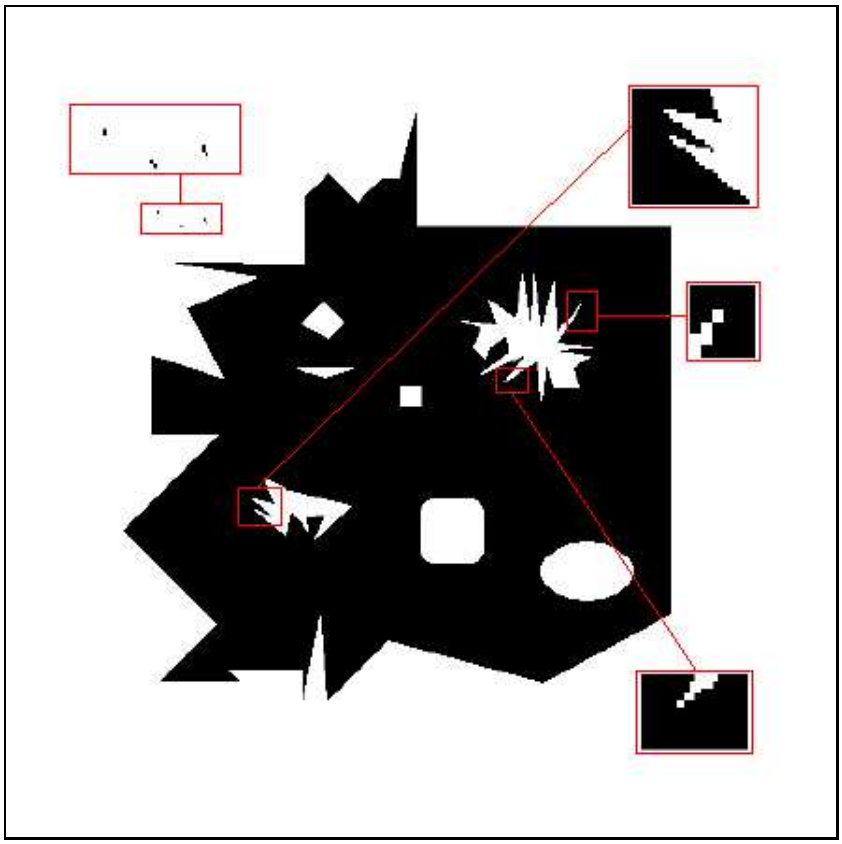

Figure 6.7: Image containing all possible boundary elements