

Aspects combinatoires dans l'analyse et la manipulation des formes digitales

Combinatorial aspects in digital shape analysis and manipulation

Mémoire d'habilitation à diriger des recherches
présentée et soutenue le 6 décembre 2018 par

Yukiko KENMOCHI

COMPOSITION DU JURY:

Dominique ATTALI	DR CNRS – GIPSA-Lab, CNRS	<i>Président/Examineur</i>
Valérie BERTHÉ	DR CNRS – IRIF, CNRS	<i>Rapporteur</i>
Gilles BERTRAND	PR – Émerite, LIGM, ESIEE Paris	<i>Examineur</i>
Alfred M. BRUCKSTEIN	PR – Technion, Israel	<i>Examineur</i>
David COEURJOLLY	DR CNRS – LIRIS, CNRS	<i>Examineur</i>
Reinhard KLETTE	PR – Auckland University of Technology	<i>Rapporteur</i>
Jacques-Olivier LACHAUD	PR – LAMA, Université de Savoie	<i>Examineur</i>
Sylvain LAZARD	DR INRIA – INRIA Nancy Grand Est	<i>Rapporteur</i>

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to all the three "rapporteurs" of this thesis: Valérie Berthé, Reinhard Klette and Sylvain Lazard, who have kindly accepted to devote a part of their precious time for evaluating my work. Many thanks are also due to the rest of the members of the jury: Dominique Attali, Gilles Bertrand, Alfred M. Bruckstein, David Coeurjolly and Jacques-Olivier Lachaud, for their encouragement, insightful comments and hard questions.

The research work presented in this thesis is the fruit of multiple collaborations. My sincere thanks also goes to all the people who contributed to the work described in this thesis, my colleagues, doctoral students and post-doctoral fellows.

Last but not the least, I would like to thank my family and friends for their supports.

CONTENTS

1	INTRODUCTION	1
2	DIGITAL SHAPE ANALYSIS WITH TOPOLOGICAL ASSUMPTION	5
2.1	Introduction	6
2.1.1	Background	6
2.1.2	Problem and approach	6
2.1.3	Related work	8
2.2	Polyhedral complex for digital object	8
2.2.1	Convex polyhedra and polyhedral complexes in \mathbb{R}^3	8
2.2.2	Polyhedral complex construction with vertex constraint on \mathbb{Z}^3	9
2.2.3	Combinatorial boundaries of polyhedral complexes	11
2.3	Topological characterization of digital object point	13
2.3.1	Star and link	13
2.3.2	Topological characterization by stars	14
2.3.3	Point classification	16
2.3.4	Enumeration: Algorithm and experiments	16
2.4	Local configurations on discrete surfaces	19
2.5	Algorithm of combinatorial boundary extraction	20
2.6	Thinning based on complex collapsibility	21
2.6.1	Background and approach	22
2.6.2	Collapsing	24
2.6.3	Collapsibility and simple points	25
2.6.4	Topological characteristics of simple points	26
2.6.5	Linear thinning algorithm	27
2.6.6	Setting of endpoints	29
2.7	Summary and perspectives	30
3	DIGITAL SHAPE ANALYSIS WITH GEOMETRIC ASSUMPTION	33
3.1	Analytic model of digital hyperplanes	33
3.1.1	Continuous hyperplanes	33
3.1.2	Digital hyperplanes	34
3.1.3	Arithmetic hyperplanes	35
3.2	Local geometric patterns of digital planes	36
3.2.1	Preimages of a digital plane patch	36
3.2.2	Local geometric patterns and their linearity	37
3.2.3	Linear LGP generation by arithmetic planes	38
3.2.4	Algorithm and results	39
3.2.5	Topological property of digital planes	40
3.3	Digital planar surface segmentation	41
3.3.1	Related work and our approach	42

3.3.2	Linear and non-linear LGPs	43
3.3.3	Non-linear point rejection of digital objects . . .	44
3.3.4	Linear LGPs and discrete Gaussian spheres . .	46
3.3.5	Planar surface segmentation using LGPs	52
3.4	Digital line and plane fitting	55
3.4.1	Related work	55
3.4.2	Problem formulation and its geometric interpretations	56
3.4.3	Exact fitting	58
3.4.4	Theoretical observation	62
3.4.5	Approximate fitting	63
3.4.6	Experiments	65
3.5	Summary and perspectives	70
4	DIGITAL SHAPE MANIPULATION: RIGID MOTIONS ON Z^2 AND Z^3	73
4.1	Rigid motion and digitization	75
4.1.1	Rigid motions on \mathbb{R}^n	75
4.1.2	Rotation representations and parameterizations	75
4.1.3	Rigid motions on \mathbb{Z}^n	77
4.2	Combinatorial local analysis	77
4.2.1	Related work	78
4.2.2	Image patch and its alterations under digitized rigid motions	78
4.2.3	Problem as an arrangement of hypersurfaces	79
4.2.4	Combinatorial analysis of structures	79
4.2.5	Graph representing the dual combinatorial structures	81
4.2.6	Algorithms for computing the structure of digitized rigid motions	82
4.3	Characterization of bijectivity	89
4.3.1	Combinatorial approach to characterization of bijectivity	90
4.3.2	Arithmetic approach to certification of bijectivity	96
4.4	Topology preservation	99
4.4.1	Digital topology and well-composed images	100
4.4.2	Topological invariance under 2D digitized rigid motions	102
4.4.3	Topological alterations under 3D digitized rigid motions	110
4.5	Geometry preservation	111
4.5.1	Digitized rigid motions via polygonization	112
4.5.2	Digitization and topology preservation	113
4.5.3	Convexity-preserving digitized rigid motions	116
4.5.4	Digitized rigid motions for non-convex objects	123
4.5.5	Experiments	126
4.5.6	Extension to 3D	127

4.6	Summary and Perspectives	129
5	CONCLUSION AND PERSPECTIVES	133
5.1	Conclusion	133
5.2	Ongoing and future challenges	134
5.2.1	Ongoing challenges	135
5.2.2	Future challenges	136
	Bibliography	141

INTRODUCTION

An image (a scene or an object) is mathematically represented by a function defined on a domain of \mathbb{R}^n , $n = 2, 3$. In order to store, process and manipulate such an image by computer, it is generally digitized by partitioning the support \mathbb{R}^n into regular polygons or polyhedra: for example, the square, hexagonal or triangular tessellation of \mathbb{R}^2 , and the cubical tessellation of \mathbb{R}^3 (see Figure 1). In particular, the center of each cell of the square or cubical tessellation is represented by a point of \mathbb{Z}^n , so that the support of a digital image is \mathbb{Z}^n instead of \mathbb{R}^n [37]. This operation is called *digitization*, and often defined as a rounding function, $\mathfrak{D} : \mathbb{R}^n \rightarrow \mathbb{Z}^n$.

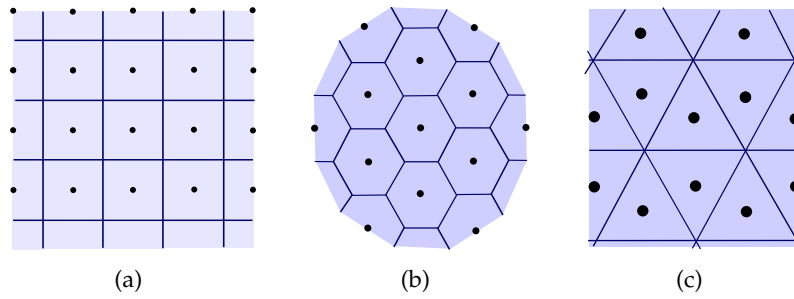


Figure 1: Regular tilings of the two-dimensional space: (a) the square tiling, (b) the hexagonal tiling, and (c) the triangular tiling.

With this operation, given a continuous object $X \subset \mathbb{R}^n$, the associated digital object is defined as $\mathfrak{D} \circ X \subset \mathbb{Z}^n$ (see Figure 2). To analyze and manipulate the shapes of such digital objects defined on \mathbb{Z}^n , one often uses classical techniques, which are based on geometric or topological models defined in the continuous frame, while their computations are performed numerically with limited precision. Consequently, the results obtained may include digitization and calculation errors that violate the geometric and topological properties of the original object, and therefore lack reliability (see Figure 2 (b) for an original shape in \mathbb{R}^2 and (d) for the topological violation after the digitization process).

In order to deal with these issues, we¹ study discrete models that allow a geometric calculation in a more reliable way for digital objects. Digital geometry is indeed the study of the geometric and topological properties of sets of discrete points in \mathbb{Z}^n , and provides us with various discrete models and useful mathematical tools [37], [101].

¹ The use of “we” throughout this thesis is purposeful. It is used to involve the reader with the thesis as recommended by [103].

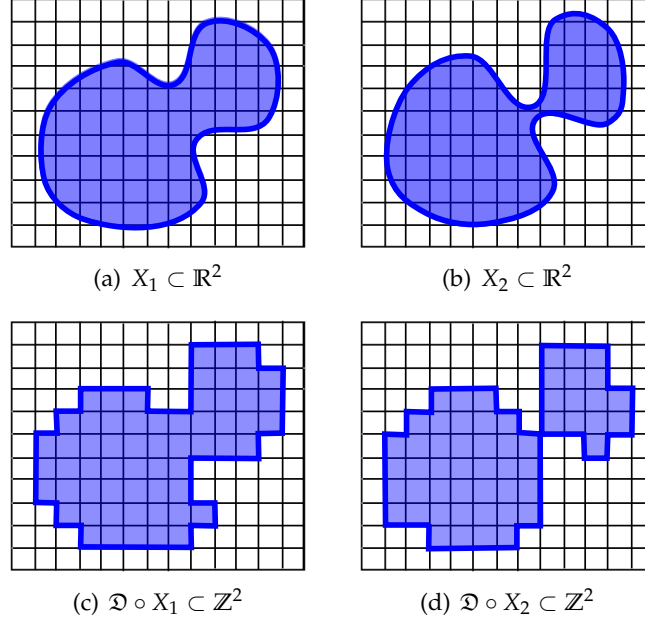


Figure 2: (a,b) Continuous shapes X_1 and X_2 in \mathbb{R}^2 (in transparent blue) and their boundaries (in blue). (c,d) The associated digitized shapes $\mathcal{D} \circ X_1$ and $\mathcal{D} \circ X_2$ in \mathbb{Z}^2 (in transparent blue), and their digital boundaries (in blue). (c) The digital boundary of $\mathcal{D} \circ X_1$ is a 1-manifold. (d) The digital boundary of $\mathcal{D} \circ X_2$ is not a 1-manifold, by contrast with that of X_2 .

Due to the discrete nature of \mathbb{Z}^n , there are often combinatorial aspects to geometric and topological problems of image analysis and synthesis. The goal of the work presented in this manuscript is to understand the combinatorial structure of the solution space of each problem, so that more efficient algorithms and properties can be proposed, with a help of digital geometry.

In this manuscript, we consider three different topics, related to analyzing and manipulating digital shapes, in different contexts: topological shape analysis (Chapter 2), geometric shape analysis (Chapter 3), and shape rigid motions (Chapter 4)². For each issue, we adopt the following common approach, which is basically fourfold:

1. determining the underlying geometric/topological assumption for the problem;
2. finding an appropriate discrete model, which discretizes the solution space;
3. studying the associated combinatorial structures and their properties; and

² The order of the topics is chronological in the research carrier of the author.

4. proposing efficient algorithms based on those structures and properties.

In the next chapter, we deal with two-dimensional surfaces in \mathbb{Z}^3 . Our assumption is that the boundaries of three-dimensional objects are surfaces. We use polyhedral complexes to interpret topology of digital objects and their boundary, and to characterize each point topologically. This characterization allows us to study the local configurations of discrete surfaces and to propose efficient algorithms for boundary extraction and thinning.

In Chapter 3, we focus on geometry. We assume that we know a priori what we observe in digital data. Here, we assume that it is a plane in a three-dimensional space. Two different problems are considered: digital planar surface segmentation and digital plane fitting. The former assumes the shape linearity locally, while the latter leads to a global optimization under the linear shape constraint. Both use the common geometric model, digital plane, formulated by a pair of linear inequalities.

Chapter 4 presents our most recent work. Here, we assume that digital objects are rigid and simply move in \mathbb{Z}^n without changing the shapes. However, in contrast to rigid motions on \mathbb{R}^n , this assumption is so strong that geometric and topological invariances are generally lost in \mathbb{Z}^n due to the digitization process. We focus on each problem independently, and develop a purely discrete framework for rigid motions on \mathbb{Z}^n based on digital geometry and topology concepts, with a help of computational geometry and computer algebra.

DIGITAL SHAPE ANALYSIS WITH TOPOLOGICAL ASSUMPTION

This chapter is dedicated to study combinatorial aspects with topological constraints, in particular, topologies of two-dimensional surfaces in a three-dimensional discrete space. Three-dimensional image analyses often require surface structures as boundaries of solid objects. For example, image segmentation is the process of partitioning a digital image into multiple objects and background. In other words, it is used to locate their boundaries, which should be generally considered as surfaces, in a three-dimensional image.

Seeing that objects in digital images are given by sets of integer points, representing such surfaces, namely 2-manifolds, as polyhedral surfaces is a natural choice when no shape knowledge is available a priori [101]. Besides, once we have polyhedral surfaces, they are useful for visualization [62], as well as calculating geometric features, such as volume, surface area, tangent, curvature, etc. in post-processing [159].

In this chapter, we do not focus on image segmentation techniques themselves, but on topological representation of each segmented region. More precisely, given a three-dimensional binary image (value 1 for a region of interest and 0 for the background), we represent the region, namely a discrete object, by a polyhedral complex. Based on this approach, we can topologically characterize every point in discrete objects in a local way. As a consequence, it is easy to characterize points on the object boundary, which forms a two-dimensional combinatorial surface (manifold). This topological characteristic enables us to develop a simple algorithm which examines whether each point of discrete objects is on a combinatorial surface. As the characterization is made locally, we find and enumerate all the local point configurations that can appear on combinatorial surfaces in the three-dimensional discrete space.

Thanks to this topological framework, we can present an efficient algorithm to directly obtain a polyhedral surface from a given discrete object. Moreover, we show that the framework is also useful for a thinning operation of a binary image.

This chapter is based on the results mainly presented in [85], [89], [116].

2.1 INTRODUCTION

2.1.1 Background

The well-known method for generating polyhedral surfaces from a three-dimensional digital image is the marching cubes method [118], [198], which simply uses a look-up table based on local point configurations. The original work was made purely for the aim of visualizing three-dimensional object in a digital image, and it is well known that there are topological problems, such as creating cracks [69], [140]. For the purpose of solving this problem, approaches making a link between the classical concept of connectivity of digital topology and polyhedral surfaces were presented [46], [93], [110].

Topological equivalence between a continuous object boundary and the polyhedral surfaces reconstructed from its digital image is also studied in [185]. The authors pointed out that if a digital image satisfies the well-composedness, proposed by Latecki [115], then the marching cubes algorithm [118], [198] does not lead to any topological problem in generated surfaces.

On the other hand, the framework of cell complex or cubical complex is very much used in the context of topological analyses of digital images [13], [16], [41], [108], [120], whose dual representation is called Khalimsky space [97]. In fact, if a digital image is well composed, the boundary of its cubical complex always leads to a topologically correct polyhedral surface [121]. Thanks to this topological guarantee, the cubical complex approach is nowadays popular if we are sure that what we treat are only three-dimensional objects.

However, when multi-dimensional representations are necessarily using only integer points for multi-dimensional shape analysis (e.g. curve or surface thinning), dual approaches of the above cubical complexes would be useful. This is because the framework of cubical complexes provides inter-voxel surfaces, namely polyhedral surfaces whose vertices are not at voxel centers but are in between, while its dual can provide polyhedral surfaces whose vertices are voxel centers, i.e. integer points. Such representation has been proposed and used, for example, for computing the Euler characteristic of a digital object [191], [197], thinning digital objects based on collapse [94], and analysing topological changes of discrete objects in the level sets of gray scale images [158], and representing surfaces in the discrete space [31], [33], [42], [126], etc.

In this chapter, we consider this latter approach.

2.1.2 Problem and approach

Let us consider a three-dimensional discrete space \mathbb{Z}^3 , consisting of points whose coordinates are all integers in a three-dimensional

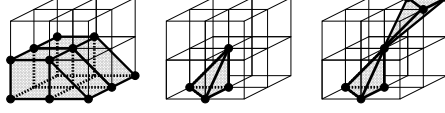


Figure 3: Examples of local configurations in a $3 \times 3 \times 3$ point region so that the central point is considered to be a border point [89], which is either a surface point (left), a surface point but not a simplicity surface point [42] (center), or a non-surface point, i.e. a singular point (right).

Euclidean space \mathbb{R}^3 . In [89], we presented a boundary extraction algorithm which provides a polygonization of a set of boundary points given by

$$\mathcal{B}r_m(\mathbf{V}) = \{p \in \mathbf{V} : \mathcal{N}_m(p) \cap \overline{\mathbf{V}} \neq \emptyset\} \quad (1)$$

where \mathbf{V} is a finite subset of \mathbb{Z}^3 , i.e., a discrete object and $\overline{\mathbf{V}}$ is its complement. $\mathcal{N}_m(p)$ is the m -neighborhood of a point p in \mathbb{Z}^3 , defined by

$$\mathcal{N}_m(p) = \{q \in \mathbb{Z}^3 : \|p - q\|_2^2 \leq r\}$$

where $r = 1, 2, 3$ for $m = 6, 18, 26$ respectively. Based on combinatorial topology [3], [186], [202], we introduce discrete polyhedral complexes [89], [116] to represent discrete objects, and then obtain topologies for boundary points. With a help of such topologies, we find that boundary points in $\mathcal{B}r_m(\mathbf{V})$ include not only surface points, i.e., points on 2-dimensional combinatorial manifolds, but also non-surface points, i.e., singular points, as shown in Fig. 3. In this chapter, we use local topological notions similarly to our work [92] to discriminate surface points from boundary points. Such notions enable us to present an algorithm to count the local point configurations appearing in discrete combinatorial surfaces for the 6-, 18- and 26-neighborhood systems; discrete combinatorial surfaces are defined for any m -neighborhood system, $m = 6, 18, 26$, such that their adjacent vertices are m -neighboring. Because there are a finite number of points in the local region, there must be a finite number of local point configurations in discrete combinatorial surfaces.

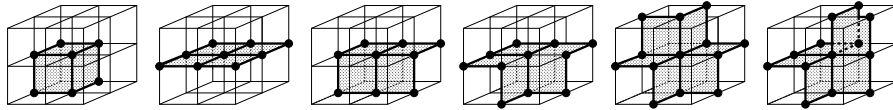


Figure 4: The 6 local configurations of $3 \times 3 \times 3$ of discrete combinatorial surfaces for the 6-neighborhood system [31], [64], [81], [83] where the five left ones appear in discrete planes [65], [90].

2.1.3 Related work

For the 6-neighborhood system, the definition of discrete combinatorial surfaces is given by Françon in [64] and he showed that there are 6 local configurations of discrete surfaces for the 6-neighborhood system as illustrated in Fig. 4. Note that similar results are obtained by using different approaches, for example, in [31], [81], [83]. The discrete deformation model based on such discrete surface configurations for 6-neighborhood system is also presented in [57]. Moreover, Françon and Kenmochi *et al.* show that there are five local configurations which appear in discrete planes, illustrated as the five left configurations in Fig. 4 [65], [90]. In other words, there is only one configuration (the most right one in Fig. 4) which does not appear in discrete planes but appears in discrete non-planar surfaces.

In [64], however, the 18- and 26-neighborhoods are not practically treated so that we do not see how to generate discrete combinatorial surfaces for the 18- and 26-neighborhood systems, even if the mathematical definition is given for any neighborhood system. Morgenthaler *et al.* defined discrete surfaces by using the point connectivity based on the Jordan surface theorem; any Jordan surface divides the space into two regions [126]. In [42], Couprie *et al.* pointed out that, for the 26-neighborhood system, Morgenthaler's discrete surfaces have only 13 local configurations while their discrete surfaces, called simplicity surfaces, have 736 configurations. However, we see that even simplicity surfaces do not give enough configurations if we would like to treat our boundary points. For example, we obtain a boundary point by applying our boundary extraction algorithm [89] as illustrated in Fig. 3 (center) and we see that it is not considered to be a simplicity surface. Ciria *et al.* also presented a graph-based notion of discrete surfaces for the 26-neighborhood system, though the number of all local point configurations has not been studied yet [33].

2.2 POLYHEDRAL COMPLEX FOR DIGITAL OBJECT

2.2.1 Convex polyhedra and polyhedral complexes in \mathbb{R}^3

For the definitions of convex polyhedra and polyhedral complexes in \mathbb{R}^3 , we follow the notions in [202]. Similar notations are also seen in [3], [186].

Definition 2.1. A convex polyhedron σ is the convex hull of a finite set of points in some \mathbb{R}^n .

The dimension of a convex polyhedron σ is the dimension of its convex hull. An n -dimensional convex polyhedron σ is abbreviated to an n -polyhedron. For instance, a point is a 0-polyhedron, a line segment is a 1-polyhedron, a triangle or a square is a 2-polyhedron, and a tetrahedron or a hexahedron is a 3-polyhedron.

A linear inequality $\mathbf{a} \cdot \mathbf{x} \leq z$ is said to be valid for σ if it is satisfied for all points $\mathbf{x} \in \sigma$. A face of σ is then defined by any set of the form

$$\delta = \sigma \cap \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{a} \cdot \mathbf{x} = z\}$$

where $\mathbf{a} \cdot \mathbf{x} \leq z$ is valid for σ . For instance, a 3-polyhedron which is a tetrahedron has four 0-polyhedra, six 1-polyhedra and four 2-polyhedra for its faces. If a k -dimensional convex polyhedron τ is a face of σ , τ is called an k -face and such a binary relation is denoted by $\tau \prec \sigma$. Note that the binary relation is reflexive so that $\sigma \prec \sigma$ for any σ and also $\emptyset \prec \sigma$ for any σ .

The point of a 0-polyhedron, the endpoints of a 1-polyhedron and the vertices of 2- and 3-polyhedra are called the vertices of each convex polyhedron.

Definition 2.2. A polyhedral complex \mathbf{K} is a finite collection of convex polyhedra such that

1. the empty polyhedron is in \mathbf{K} ,
2. if $\sigma \in \mathbf{K}$ and $\tau \prec \sigma$, then $\tau \in \mathbf{K}$,
3. if $\sigma, \tau \in \mathbf{K}$, then the intersection $\sigma \cap \tau$ is a common face of σ and τ .

The dimension of \mathbf{K} is the greatest dimension of a convex polyhedron in \mathbf{K} . It is known that \mathbf{K} is a partially ordered set, which can be identified with a topological space called a discrete space; the detail is found in Section 6 of Chapter 1 in [3].

2.2.2 Polyhedral complex construction with vertex constraint on \mathbb{Z}^3

If we have a method to construct a polyhedral complex \mathbf{K} from a finite point set \mathbf{V} in \mathbb{Z}^3 , satisfying the following three properties, we can derive a local topological characterization of points based on such a polyhedral complex. In the following, we describe a method proposed in [89], [116].

Property 2.1. A polyhedral complex \mathbf{K} is uniquely constructed from any finite subset $\mathbf{V} \subset \mathbb{Z}^3$, written by




































$$\mathbf{K} = \text{Comp}(\mathbf{V}).$$

Let $Sk_n(\mathbf{K})$ be the union of all n -polyhedra in \mathbf{K} , called an n -dimensional skeleton of \mathbf{K} . Therefore, $Sk_0(\mathbf{K})$ denotes the union of sets of vertices of all $\sigma \in \mathbf{K}$.

Property 2.2. Given a finite subset $\mathbf{V} \subset \mathbb{Z}^3$, we have

$$Sk_0(\text{Comp}(\mathbf{V})) = \mathbf{V}.$$

Table 1: All n -dimensional discrete convex polyhedra, $n = 0, 1, 2, 3$, for the m -neighborhood systems, $m = 6, 18, 26$, up to rotations and symmetries.

dim.	discrete convex polyhedra					
	N6	N18			N26	
0						
1						
2						
						
3						
						
						
						

We say that a point q is m -adjacent to p if $q \in \mathcal{N}_m(p) \setminus \{p\}$. Let $\mathcal{A}_m(p)$ be the set of all m -adjacent points of p . Then we can define all the adjacent relations in \mathbf{V} such that

$$\mathcal{A}_m(\mathbf{V}) = \bigcup_{p \in \mathbf{V}} \bigcup_{q \in \mathcal{A}_m(p)} \{\{p, q\}\}.$$

Property 2.3. *Given a finite subset $\mathbf{V} \subset \mathbb{Z}^3$, we have*











$$Sk_1(Comp(\mathbf{V})) = \mathcal{A}_m(\mathbf{V})$$

where $m = 6, 18, 26$.

In the following, we describe a method proposed in [89], [116] for construction of $Comp(\mathbf{V})$ satisfying the above three properties, called discrete polyhedral complex.

The construction is made with respect to a chosen m -neighborhood where $m = 6, 18, 26$. Let us first consider the case of $m = 26$. We consider a unit cube whose eight vertices are discrete points in \mathbb{Z}^3 .

Table 2: Five 1-point configurations of a unit cube where different discrete complexes are constructed for the 18- and 26-neighborhood systems.

	a discrete complex in a unit cube				
N_{18}					
N_{26}					

Setting the value of each point at either 1 or 0, we make a convex hull of points whose value is 1. The dimension of such a convex hull can vary from 0 to 3 and we see that every pair of adjacent vertices of any discrete convex polyhedron are 26-neighboring, as illustrated in Table 1. After generating a discrete convex polyhedron in each unit cubic region, we compute the union of all discrete convex polyhedra and their faces, and obtain a discrete polyhedral complex \mathbf{K} .

If we consider discrete convex polyhedra such that every pair of adjacent vertices are 6-neighboring, we obtain only one type of discrete convex polyhedra for each dimension as shown in Table 1. Similarly to the case of $m = 26$, for the case of $m = 6$, considering the union of all discrete convex polyhedra and their faces, we obtain a polyhedral complex \mathbf{K} .

Let us consider the case of $m = 18$. Table 1 shows that there are 22 different 1-point configurations for $m = 26$, among which there are five 1-point configurations missing for the 18-neighborhood system. This is because each of those five convex polyhedra includes a pair of adjacent vertices which are 26-neighboring but not 18-neighboring. Therefore, to construct a discrete complex in a unit cube whose 1-point configuration is one of those five for the 18-neighborhood system, we need to construct the discrete complexes as illustrated in Table 2. In addition, the last case in Table 2 may cause the conflicts depending on the 1-point configuration in its adjacent unit cube (see the left figures in Figs. 5 and 6). In those cases, we modify polyhedral complexes in both cubes as illustrated in Figs. 5 and 6 (right). We remark that such a replacement does not destroy the topology of a discrete complex. The details and the precise algorithm can be found in [89].

2.2.3 Combinatorial boundaries of polyhedral complexes

Before defining combinatorial boundaries, we give some notions for polyhedral complexes [3], [89], [116].

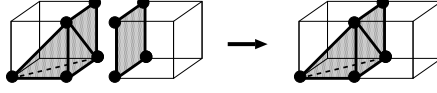


Figure 5: Additional polyhedral decomposition for the 18-neighborhood system.

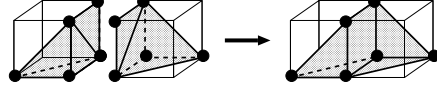


Figure 6: Modification of polyhedral complexes in two adjacent unit cubes such that each of them has the last 1-point configuration in Table 2, and they have four common 1-points and also have two non-common 1-points whose distance is $\sqrt{5}$, for the 18-neighborhood system.

Definition 2.3. An n -complex \mathbf{K} is said to be pure if there is at least one n -polyhedron $\sigma \in \mathbf{K}$ for every s -polyhedron $\tau \in \mathbf{K}$ so that $\tau \prec \sigma$.

Figure 7 shows examples of pure and non-pure discrete complexes.

The following definitions of combinatorial closure and of connectivity are necessary for topological operations in discrete polyhedral complexes.

Definition 2.4. Let \mathbf{K} be a polyhedral complex and \mathbf{H} be a subset of \mathbf{H} . The combinatorial closure of \mathbf{H} is defined as

$$Cl(\mathbf{H}) = \mathbf{H} \cup \{\tau \in \mathbf{K} : \tau \prec \sigma, \sigma \in \mathbf{H}\}.$$

Note that \mathbf{H} may not be a complex while $Cl(\mathbf{H})$ is always a complex, and $\mathbf{H} \neq Cl(\mathbf{H})$ if \mathbf{H} is not a complex. Figure 8 shows an example of the closure of a subset for a given complex.

Definition 2.5. Let \mathbf{K} be a polyhedral complex, and σ, τ be arbitrary elements in \mathbf{K} . We say that \mathbf{K} is connected if there exists a path $\sigma(= a_1), a_2, \dots, \tau(= a_n)$ in \mathbf{K} that satisfies $Cl(\{a_i\}) \cap Cl(\{a_{i+1}\}) \neq \emptyset$ for every $i = 1, 2, \dots, n-1$.

We can then define combinatorial boundary based on the notion of algebraic topology [125], [186].

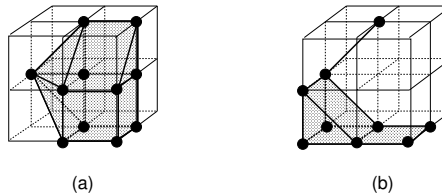


Figure 7: Examples of (a) pure and (b) non-pure 3-complexes.

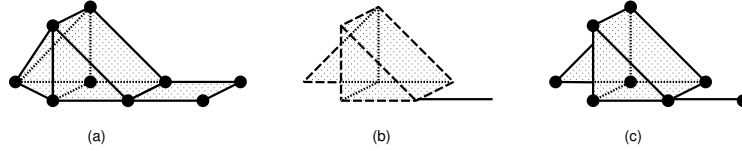


Figure 8: (a) A 3-complex \mathbf{K} , (b) its subset \mathbf{H} consisting of three convex polyhedra whose dimensions are from 1 to 3 respectively, and (c) the closure of \mathbf{H} in \mathbf{K} .

Definition 2.6. Let \mathbf{K} be a connected and pure n -complex where $n > 0$ and \mathbf{H} be the set of all $(n - 1)$ -polyhedra in \mathbf{K} each of which is a face of exactly one n -polyhedron in \mathbf{K} . The combinatorial boundary of \mathbf{K} is then defined as the pure $(n - 1)$ -complex

$$\partial\mathbf{K} = Cl(\mathbf{H}).$$

2.3 TOPOLOGICAL CHARACTERIZATION OF DIGITAL OBJECT POINT

One of the goals of this chapter is to present an algorithm to verify whether each point in $Sk_0(\mathbf{K})$ is considered to be on a discrete surface or not. For this goal, we study topological characterization of each point in $Sk_0(\mathbf{K})$ by observing its local point configuration and investigate all topological characteristics which can be maintained by points in $Sk_0(\mathbf{K})$. In \mathbb{Z}^3 , we have polyhedral complexes whose dimensions can be from zero to three. Thus, we present topological characterization of polyhedral complexes for each dimension from one to three [116] by using the notions of star and link [202] similarly to the previous work [92]. We then show that there are 12 topological types of points in $Sk_0(\mathbf{K})$. In the next sections, we will classify all points in $Sk_0(\mathbf{K})$ by their topological characteristics and study the type of points in discrete surfaces.

In this section, we do not have to distinguish the three different neighborhood systems. Thus, we abbreviate a polyhedral complex \mathbf{K}_m for $m = 6, 18, 28$ simply to \mathbf{K} .

2.3.1 Star and link

The star and the link are defined for each element in $Sk(\mathbf{K})$ as follows.

Definition 2.7. For a polyhedral complex \mathbf{K} , the star of a convex polyhedron $\sigma \in Sk(\mathbf{K})$ is defined such that

$$Star(\sigma) = \{\tau \in \mathbf{K} : \sigma \prec \tau\}.$$

Definition 2.8. For a polyhedral complex \mathbf{K} , the link of a convex polyhedron $\sigma \in Sk(\mathbf{K})$ is defined such that

$$Link(\sigma) = Cl(Star(\sigma)) \setminus Star(\sigma).$$

If we need to emphasize \mathbf{K} where a star and a link are calculated, we denote them by $star(\sigma : \mathbf{K})$ and $link(\sigma : \mathbf{K})$ respectively. Figure 9 shows examples of star and link. Note that any link is a polyhedral complex while stars are not always polyhedral complexes.

The dimension of $Star(\sigma)$ is defined as the greatest dimension of convex polyhedra belonging to $Star(\sigma)$ and denoted by $\dim(Star(\sigma))$.

2.3.2 Topological characterization by stars

For each 0-polyhedron, namely a point x , in the 0-skeleton $Sk_0(\mathbf{K})$ of a polyhedral complex \mathbf{K} , we define topological characteristics of stars [3], [116].

Definition 2.9. Let \mathbf{K} be a polyhedral complex and $x \in Sk_0(\mathbf{K})$. We say that $Star(x)$ is linear if $Link(x)$ consists of two 0-polyhedra.

Definition 2.10. Let \mathbf{K} be a polyhedral complex and $x \in Sk_0(\mathbf{K})$. We say that $Star(x)$ is semi-linear if $Link(x)$ consists of one 0-polyhedron.

Figure 10 illustrates stars which are linear and semi-linear. By using linear and semi-linear stars, we define combinatorial curves.

Definition 2.11. Let \mathbf{K} be a connected and pure 1-complex. We say that \mathbf{K} is a combinatorial curve with endpoints if the star of every 0-polyhedron in $Sk_0(\mathbf{K})$ is either linear or semi-linear and there is at least one point whose star is semi-linear in $Sk_0(\mathbf{K})$.

Definition 2.12. Let \mathbf{K} be a connected and pure 1-complex. We say that \mathbf{K} is a combinatorial closed curve if the star of every 0-polyhedron in $Sk_0(\mathbf{K})$ is linear.

By using the above definitions of combinatorial curves, we define topological characteristics of stars in two dimensions.

Definition 2.13. Let \mathbf{K} be a polyhedral complex and $x \in Sk_0(\mathbf{K})$. We say that $Star(x)$ is cyclic if $Link(x)$ is a combinatorial closed curve.

Definition 2.14. Let \mathbf{K} be a polyhedral complex and $x \in Sk_0(\mathbf{K})$. We say that $Star(x)$ is semi-cyclic if $Link(x)$ is a combinatorial curve with endpoints.

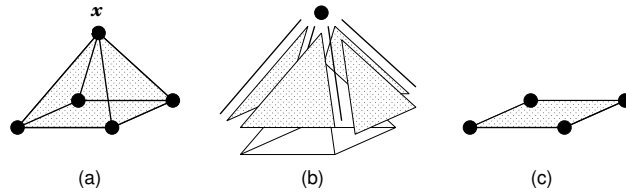


Figure 9: (a) A 3-complex \mathbf{K} ; (b) the star of $x \in Sk_0(\mathbf{K})$; (c) the link of x .

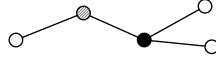


Figure 10: One-dimensional topological characterization of points whose stars are linear, semi-linear and neither of them, illustrated as grey, white and black points.

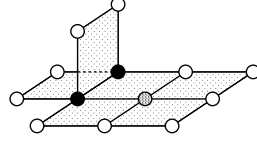


Figure 11: Two-dimensional topological characterisation of points whose stars are cyclic, semi-cyclic and neither of them, illustrated as grey, white and black points.

Figure 11 illustrates stars which are cyclic and semi-cyclic. By using cyclic and semi-cyclic stars, we define combinatorial surfaces.

Definition 2.15. Let \mathbf{K} be a connected and pure 2-complex. We say that \mathbf{K} is a combinatorial surface with edges if every 0-polyhedron in $Sk_0(\mathbf{K})$ has either a cyclic or semi-cyclic star, and there is at least one 0-polyhedron whose star is semi-cyclic in $Sk_0(\mathbf{K})$.

Definition 2.16. Let \mathbf{K} be a connected and pure 2-complex. We say that \mathbf{K} is a combinatorial closed surface if every 0-polyhedron in $Sk_0(\mathbf{K})$ has a cyclic star.

By using combinatorial surfaces and combinatorial boundary (Definition 2.6), we define topological characteristics of stars in three dimensions.

Definition 2.17. Let \mathbf{K} be a polyhedral complex and $x \in Sk_0(\mathbf{K})$. We say that $Star(x)$ is spherical if $Link(x)$ is a combinatorial closed surface.

Definition 2.18. Let \mathbf{K} be a polyhedral complex and $x \in Sk_0(\mathbf{K})$. We say that $Star(x)$ is semi-spherical if $Link(x)$ is a combinatorial surface with edges, and the edges, i.e., the combinatorial boundary $\partial(Link(x))$ is a combinatorial closed curve.

Figure 12 illustrates stars which are spherical, semi-spherical and neither of them. It also shows that a point whose star is spherical is an interior point in a 3-complex, a point whose star is semi-spherical is a boundary point of a 3-complex, and a point whose star is neither spherical nor semi-spherical is a singular point, i.e., an intersection point of the boundaries.

We present the following proposition which plays an important role in this chapter [116].

Proposition 2.1. Let \mathbf{K} be a pure 3-complex and x be a point in $Sk_0(\mathbf{K})$. If $Star(x : \mathbf{K})$ is semi-spherical, then $star(x : \partial\mathbf{K})$ is cyclic.

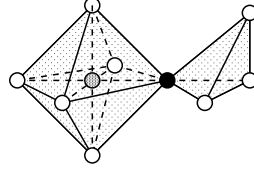


Figure 12: Three-dimensional topological characterisation of points whose stars are spherical, semi-spherical and neither of them, illustrated as grey, white and black points.

2.3.3 Point classification

Each 0-polyhedron, namely point x , in the 0-skeleton $Sk_0(\mathbf{K})$ of an n -complex \mathbf{K} where $n \leq 3$ can be classified into one of the twelve types each of which satisfies one of the following conditions [116].

TYPE 0: $\dim(\text{Star}(x)) = 0$;

TYPE 1A: $\text{Star}(x)$ is linear;

TYPE 1B: $\text{Star}(x)$ is semi-linear;

TYPE 1C: $\dim(\text{Star}(x)) = 1$ and $\text{Star}(x)$ is neither linear nor semi-linear;

TYPE 2A: $\text{Star}(x)$ is cyclic;

TYPE 2B: $\text{Star}(x)$ is semi-cyclic;

TYPE 2C: $\dim(\text{Star}(x)) = 2$, $Cl(\text{Star}(x))$ is pure and $\text{Star}(x)$ is neither cyclic nor semi-cyclic;

TYPE 2D: $\dim(\text{Star}(x)) = 2$ and $Cl(\text{Star}(x))$ is not pure;

TYPE 3A: $\text{Star}(x)$ is spherical;

TYPE 3B: $\text{Star}(x)$ is semi-spherical;

TYPE 3C: $\dim(\text{Star}(x)) = 3$, $Cl(\text{Star}(x))$ is pure and $\text{Star}(x)$ is neither spherical nor semi-spherical;

TYPE 3D: $\dim(\text{Star}(x)) = 3$ and $Cl(\text{Star}(x))$ is not pure.

2.3.4 Enumeration: Algorithm and experiments

As we mentioned above, any point is classified into one of the twelve types. We also see that these twelve types have a hierarchical structure as shown in Fig. 13.

By using the hierarchical structure, we can easily obtain an algorithm to classify a local 1-point set \mathbf{V} with respect to a type of the star of the central point $x \in \mathbf{V}$:

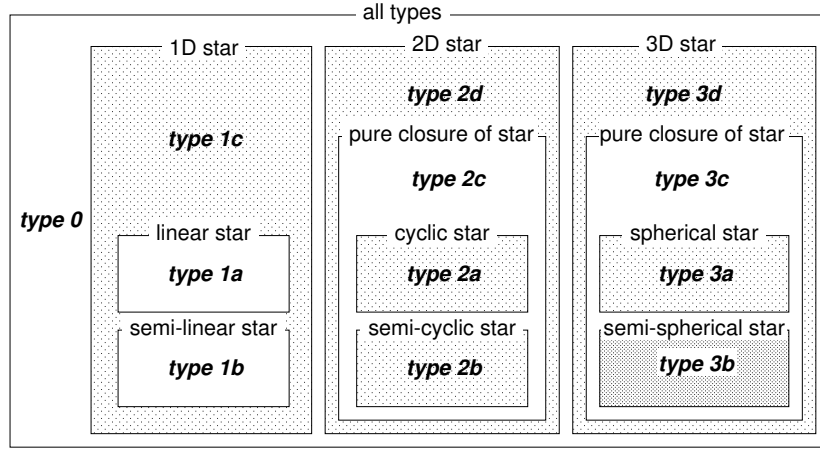


Figure 13: Hierarchical point classification by topological characterization of stars on $Comp(\mathbf{V})$ for any $\mathbf{V} \subset \mathbb{Z}^3$.

1. construct a polyhedral complex \mathbf{K}_m for $m = 6, 18$ or 26 from \mathbf{V} ;
2. obtain $Star(x)$ in \mathbf{K}_m for each $x \in \mathbf{V}$;
3. classify each $Star(x)$ by its dimension;
4. if the dimension is more than zero, classify $Star(x)$ by its topological characteristics (including the purity of $Cl(Star(x))$ for more than one dimension).

Note that x is always a 1-point for any 1-point configuration of \mathbf{V} . In addition, a discrete complex \mathbf{K}_m for $m = 6, 18, 26$ is uniquely obtained from \mathbf{V} as described in Section 2.2.2.

We apply the algorithm to every local 1-point configurations of $\mathbf{V} \subseteq \mathcal{N}_{26}(x)$ whose central point x is a 1-point. The number of all possible 1-point configurations of \mathbf{V} is $2^{26} = 67\,108\,864$, that is reduced to 1426144 up to rotations around the x -, y - and z -axes and symmetries with respect to the xy -, yz -, zx -planes. Among them, we count the number of each type of local point configurations.

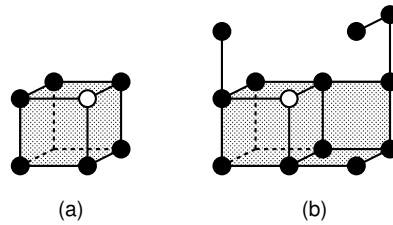


Figure 14: Examples of two different 3-complexes \mathbf{K}_6 (a) and \mathbf{K}'_6 (b) whose central points, illustrated as white points, have type 3b such that $Sk_0(\mathbf{K}_6) \neq Sk_0(\mathbf{K}'_6)$ but $Sk_0(Cl(Star(x : \mathbf{K}_6))) = Sk_0(Cl(Star(x : \mathbf{K}'_6)))$.

Table 3: The numbers of star configurations for each of the twelve types, with respect to the 6-, 18- and 26-neighborhood system, up to rotations and symmetries.

	6-neighborhood	18-neighborhood	26-neighborhood
type 0	1	1	1
type 1a	2	6	11
type 1b	1	2	3
type 1c	6	17	77
type 2a	6	80	55
type 2b	14	313	398
type 2c	123	938	3 203
type 2d	74	461	3 664
type 3a	1	21 425	23 520
type 3b	9	102 793	290 979
type 3c	11	58 532	321 371
type 3d	274	179 893	782 862
total	522	364 461	1 426 144

The type of the central point x is not determined by \mathbf{K}_m , but by $Star(x : \mathbf{K}_m)$. Therefore, we do not need to observe all points in $Sk_0(\mathbf{K}_m)$ but only those in $Sk_0(Cl(Star(x : \mathbf{K}_m)))$ for obtaining the point type of x . For example, Fig. 14 shows examples of two different 3-complexes \mathbf{K}_6 and \mathbf{K}'_6 whose central points have type 3b such that

$$Sk_0(\mathbf{K}_6) \neq Sk_0(\mathbf{K}'_6)$$

but

$$Sk_0(Cl(Star(x : \mathbf{K}_6))) = Sk_0(Cl(Star(x : \mathbf{K}'_6))). \quad (2)$$

Obviously, because of Eq. (2), they have the same type 3b, and have the same forms around the central points.

For $m = 6, 18$, the following equation does not always hold;

$$Sk_0(\mathbf{K}_m) = Sk_0(Cl(Star(x : \mathbf{K}_m))), \quad (3)$$

while it always holds for $m = 26$. For example, we see in Fig. 14 that Eq. (3) holds for \mathbf{K}_6 (a) but does not for \mathbf{K}'_6 (b). In order to avoid counting the local point configurations twice for \mathbf{K}_6 and \mathbf{K}'_6 in Fig. 14, we count different configurations of $Sk_0(Cl(Star(x : \mathbf{K}_m)))$ instead of those of $Sk_0(\mathbf{K}_m)$. Such configurations are called star configurations and we obtain Table 3. Note that $Star(x : \mathbf{K}_m)$ is not a polyhedral complex so that we make a lowest complex by using the closure function before making its skeleton. We also mention that we count configurations of $Star(x : \mathbf{K}_m)$ up to rotations and symmetries; thus no redundant configuration is contained in Table 3. We also see in Table 3 that the total numbers of different star configurations for $m = 6, 18$ are much less than that for $m = 26$.

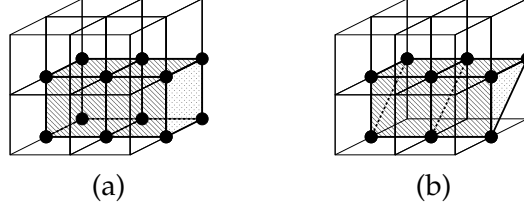


Figure 15: Two different discrete complexes \mathbf{K}_{26} (a) and \mathbf{K}'_{26} (b) around the central points x such that $\text{Star}(x : \partial\mathbf{K}_{26}) = \text{Star}(x : \partial\mathbf{K}'_{26})$.

2.4 LOCAL CONFIGURATIONS ON DISCRETE SURFACES

Our discrete combinatorial surfaces appear at the 2-dimensional combinatorial boundaries of 3-complexes, that is $\partial\mathbf{K}$, where $\dim(\mathbf{K}) = 3$. Because $\text{Star}(x : \partial\mathbf{K})$ is cyclic if $\text{Star}(x : \mathbf{K})$ is semi-spherical, from Proposition 2.1, we see that semi-spherical stars whose point type is 3b give all local configurations appearing in such discrete combinatorial surfaces.

Let us consider a set of boundary points $\mathcal{B}r_{m'}(\mathbf{V})$ of (1) for $m' = 6, 18, 26$. In [89], we see that $\mathcal{B}r_{m'}(\mathbf{V})$ includes all types of points except for spherical points (type 3a), which are interior points of \mathbf{V} . It is also observed that there are relations between the polyhedral complex \mathbf{C}_m constructed from a given \mathbf{V} and $\mathcal{B}r_{m'}(\mathbf{V})$ for the pairs $(m, m') = (6, 18), (6, 26), (18, 6), (26, 6)$ [89]. Those relations indicate that the points of $\mathcal{B}r_{m'}(\mathbf{V})$ do not always have semi-spherical stars, but also the other stars such as one-, two- and three-dimensional stars except for spherical stars depending on their local point configurations. By using our topological classification of local point configurations in the previous section, we easily discriminate semi-spherical stars from the other stars on boundaries.

As shown in Table 3, the numbers of semi-spherical star configurations of $Sk_0(Cl(\text{Star}(x : \mathbf{K}_m)))$ are still large, especially for $m = 18, 26$. What we are interested in, however, is point configurations of stars of x in combinatorial boundaries $\partial\mathbf{K}_m$ but not in \mathbf{K}_m . For example, we have different discrete complexes \mathbf{K}_m and \mathbf{K}'_m such that $Sk_0(\mathbf{K}_m) \supset Sk(\mathbf{K}'_m)$ and $\text{Star}(x : \partial\mathbf{K}_m) = \text{Star}(x : \partial\mathbf{K}'_m)$ as illustrated in Fig. 15.

Note that this does not occur for $m = 6$. However, we have the case that $\mathbf{K}_6 \neq \mathbf{K}'_6$ and $\text{Star}(x : \partial\mathbf{K}_6) = \text{Star}(x : \partial\mathbf{K}'_6)$ as shown in Fig. 16. We can also say that two discrete surfaces in Fig. 16 have the same shape but do not have the same orientation if we consider that they have two sides, the inside and outside.

In order to avoid counting twice for such two discrete complexes \mathbf{K}_m and \mathbf{K}'_m in Figs. 15, 16 respectively, we consider point configurations of $Sk_0(Cl(\text{Star}(x : \partial\mathbf{K}_m)))$ instead of those of $Sk_0(Cl(\text{Star}(x : \mathbf{K}_m)))$. Such point configurations are called surface star configurations. The results of counting different surface star configurations up to rotations

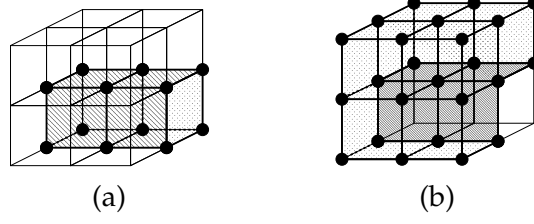


Figure 16: Two different discrete complexes K_6 (a) and K'_6 (b) around the central points x such that $Star(x : \partial K_6) = Star(x : \partial K'_6)$.

Table 4: The numbers of surface star configurations for the m -neighborhood systems for $m = 6, 18, 26$.

	$m = 6$	$m = 18$	$m = 26$
# of surface stars	6	1 412	6 028

and symmetries for $m = 6, 18, 26$ are shown in Table 4. We verified that our results for $m = 6$ are the same as those in reference [64], which are illustrated in Fig. 4.




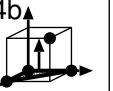
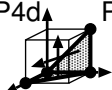
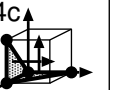




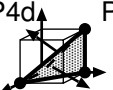
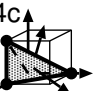
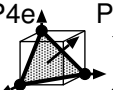






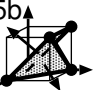


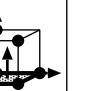


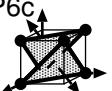


For $m = 6$, we see that surface star configurations are exactly the same as cyclic star configurations (type 2a in Table 3). However, for $m = 18, 26$, they contain more configurations than cyclic star configurations. One of the reasons is that a cyclic star requires that both interior and exterior points which are separated by a discrete surface exist in $\mathcal{N}_{26}(x)$. For $m = 18, 26$, there are some surface star configurations where there is no interior point as illustrated in Fig. 3 (center).

2.5 ALGORITHM OF COMBINATORIAL BOUNDARY EXTRACTION

In Section 2.2.3, we gave the definition of the combinatorial boundary of a pure n -complex (Definition 2.6). Here, we consider the cases of $n = 3$, so that the combinatorial boundary is a pure 2-complex. However this definition and the complex construction method presented in Section 2.2.2 do not provide a practical algorithm for extracting the combinatorial boundaries of the polyhedral complex constructed from a given finite points set \mathbf{V} . In this section, we therefore present an effective algorithm for generating $\partial Comp_m(\mathbf{V})$ directly from \mathbf{V} . The method simply refers to Table 5, which is a similar table used for the marching cubes method [118], [198], for each neighborhood system. The comparison between the marching cubes method and our method is discussed in [93]. The details of how to obtain Table 5 can be found in [89].

Some experimental results of combinatorial boundaries $\partial Comp_m(\mathbf{V})$ for the various inputs \mathbf{V} , such as digitized sphere, cube, torus and

Table 5: The look-up table which provides a one-to-one correspondence between a configuration of 1-points in a unit cubic region $Cube(x)$ and a pure polyhedral 2-complex $T_m(x)$ for the combinatorial boundary $\partial Comp_m(V)$ of the set V of all 1-points with respect to each $m = 6, 18, 26$. The arrows are oriented to the exterior of $Comp_m(V)$.

# of 1-points	2D pure subcomplex for a combinatorial boundary		
	N ₆	N ₁₈	N ₂₆
3		P3a 	
4	P4a 	P4a  P4b  P4d  P4c  P4e  P4g 	P4a  P4b  P4d  P4c  P4e  P4g 
5	P5a 	P5a  P5b  P5c 	P5a  P5b  P5c 
6	P6a  P6b 	P6a  P6b  P6c 	
7	P7 	P7 	

catenoid, with respect to $m = 6, 18, 26$ are shown in Figures 17, 18, 19 and 20. Those inputs of volume data are made by the Volgen tool.

2.6 THINNING BASED ON COMPLEX COLLAPSIBILITY

Thinning is an image operation whose goal is to reduce object points in a “topology-preserving” way. Such points whose removal does not change the topology are called simple points and they play an important role in any thinning process. For efficient computation, local characterizations have been already studied based on the concept of point connectivity for two- and three-dimensional digital images. In this section, we introduce a topological characterization of simple points based on collapsibility of polyhedral complexes. We also study their topological characteristics and propose a linear thinning algorithm. This section is based on the results in [85].

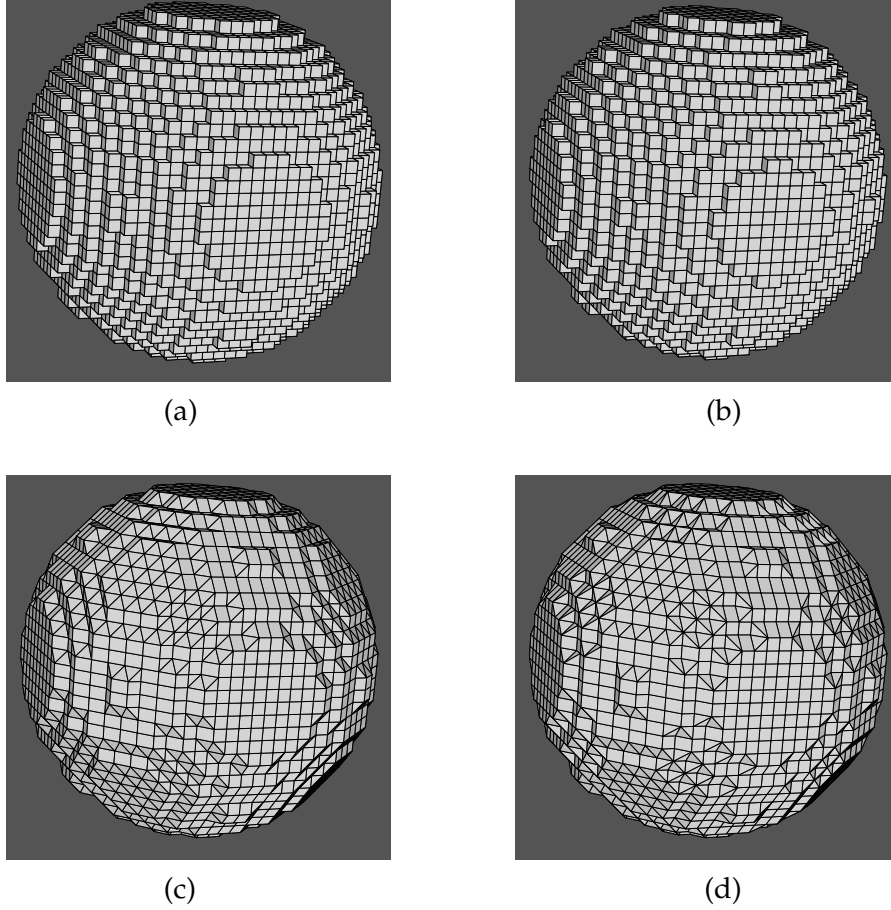


Figure 17: (a) A digitized sphere generated by Volgen and its combinatorial boundaries for (b) 6-, (c) 18- and (d) 26-neighborhood systems. Note that lattice points are located at centers of cubes in (a) whereas they are located at vertices of polyhedral complexes in (b), (c) and (d).

2.6.1 Background and approach

Mathematically, the definition of simple points is given as follows [15].

Definition 2.19. A point x in a finite subset $\mathbf{V} \subset \mathbb{Z}^3$ is said to be simple if there is a one-to-one correspondence of each connected component of \mathbf{V} and its complement $\bar{\mathbf{V}}$, and the holes of \mathbf{V} and $\bar{\mathbf{V}}$, with each connected component of $\mathbf{V} \setminus \{x\}$ and $\bar{\mathbf{V}} \cup \{x\}$, and the holes of $\mathbf{V} \setminus \{x\}$ and $\bar{\mathbf{V}} \cup \{x\}$, respectively.

Because the above global definition is not appropriate for computation, many studies on their local characterization have been made: for example, in 3D, characterizations by using connected component numbers, genus, Euler numbers, and other numbers [15], [105], [191].

Here, we introduce one of the simplest characterizations of simple points in 3D by using topological numbers, proposed in [15]. We consider the m -neighborhoods in \mathbb{Z}^3 for $m = 6, 26$. Let \mathbf{V} be a subset in \mathbb{Z}^3 , $C_m(\mathbf{V})$ be the set of all m -connected components of \mathbf{V} , and

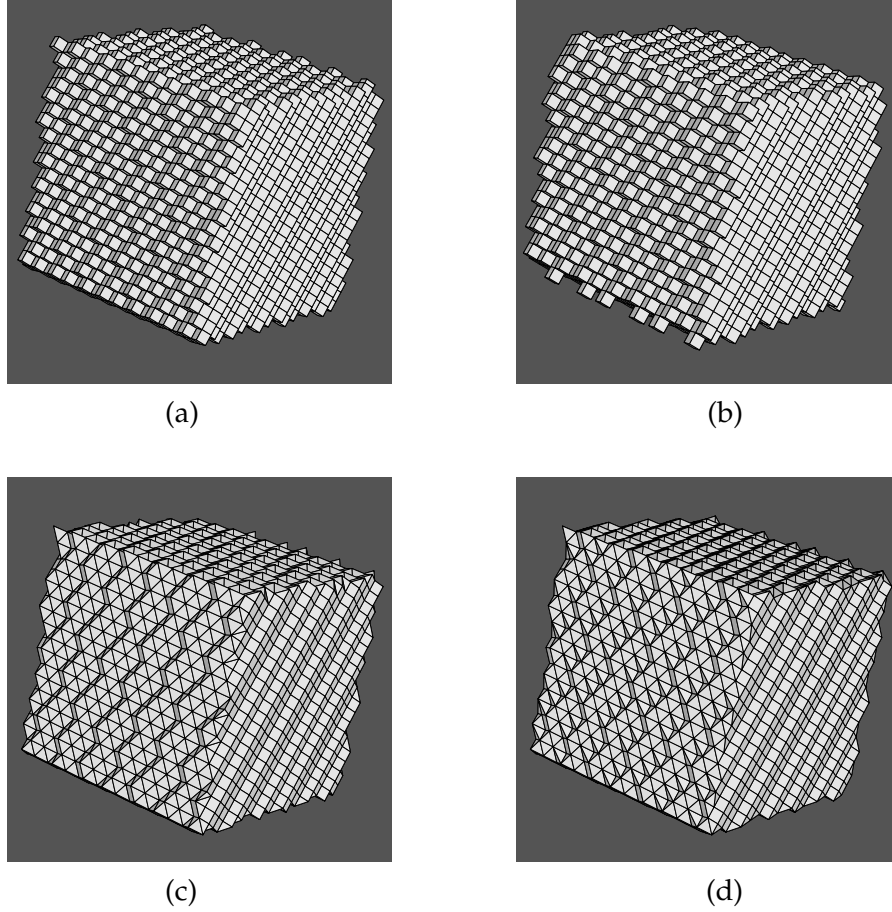


Figure 18: (a) A digitized cube with 45-degree rotations around x and y axes, generated by Volgen, and its combinatorial boundaries for (b) 6-, (c) 18- and (d) 26-neighborhood systems. Note that lattice points are located at centers of cubes in (a) whereas they are located at vertices of polyhedral complexes in (b), (c) and (d).

$C_m^a[x, \mathbf{V}]$ be the set of all components in $C_m(\mathbf{V})$ which are m -adjacent to a point x . Then, we define topological numbers

$$\begin{aligned} T_6(x, \mathbf{V}) &= \#C_6^a[x, \mathcal{N}_{18}(x) \setminus \{x\} \cap \mathbf{V}], \\ T_{26}(x, \mathbf{V}) &= \#C_{26}^a[x, \mathcal{N}_{26}(x) \setminus \{x\} \cap \mathbf{V}], \end{aligned}$$

where $\#X$ is the cardinal of a set X . By using these topological numbers, the following proposition is obtained.

Proposition 2.2. *A point $x \in \mathbf{V}$ is m -simple if and only if $T_m(x, \mathbf{V}) = T_{\bar{m}}(x, \bar{\mathbf{V}}) = 1$ for $(m, \bar{m}) = (6, 26), (26, 6)$.*

In this section, we present a topological characterization of simple points based on collapsibility of polyhedral complexes (see Section 2.6.2). We show that our characterization is also local and only needs the connectivity m of \mathbf{V} but not \bar{m} of $\bar{\mathbf{V}}$ (Section 2.6.3). Therefore, we can avoid the well-known problem of how to choose a connectivity

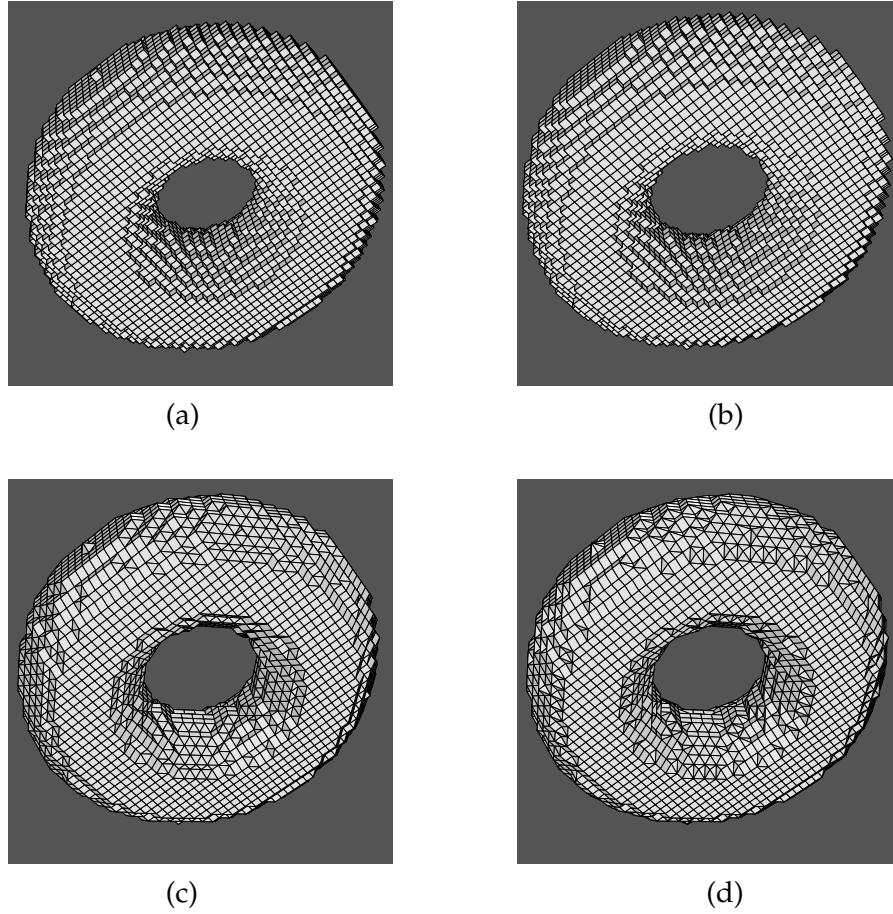


Figure 19: (a) A digitized torus generated by Volgen, and its combinatorial boundaries for (b) 6-, (c) 18- and (d) 26-neighborhood systems. Note that lattice points are located at centers of cubes in (a) whereas they are located at vertices of polyhedral complexes in (b), (c) and (d).

pair (m, \bar{m}) for \mathbf{V} and $\bar{\mathbf{V}}$. Moreover, we show topological characteristics of simple points derived from collapsibility (Section 2.6.4). We also propose a linear thinning algorithm (Section 2.6.5) and discuss on the advantages of our method and on the problems which still exist (Section 2.6.6).

2.6.2 Collapsing

We introduce a deformation retraction of a polyhedral complex, called *collapsing* [166], [186].

Let \mathbf{K} be an n -complex and σ be an r -polyhedron in \mathbf{K} where $r < n$. If there is exactly one $(r+1)$ -face $\tau \in \mathbf{K}$ such that $\sigma \prec \tau$, such a σ is called *free*. Then we say that there is an *elementary collapse* of \mathbf{K} to a subcomplex $\mathbf{K}' = \mathbf{K} \setminus \{\sigma, \tau\}$, denoted by $\mathbf{K} \searrow^e \mathbf{K}'$.

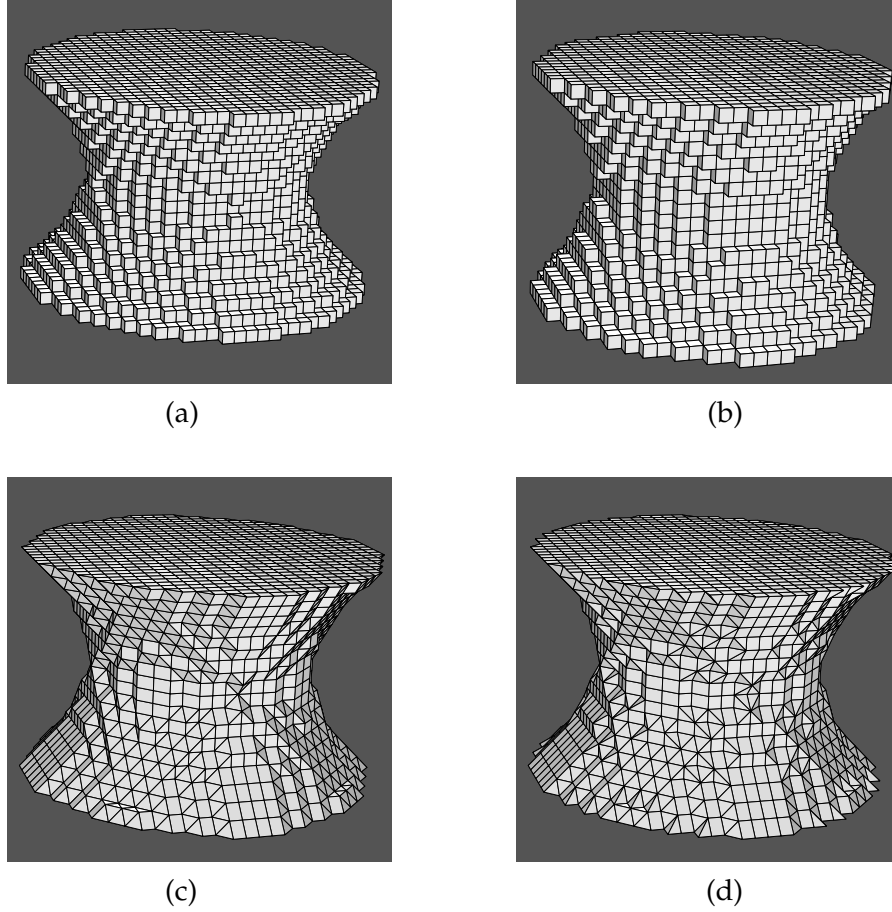


Figure 20: (a) A digitized catenoid generated by Volgen, and its combinatorial boundaries for (b) 6-, (c) 18- and (d) 26-neighborhood systems. Note that lattice points are located at centers of cubes in (a) whereas they are located at vertices of polyhedral complexes in (b), (c) and (d).

Definition 2.20. We say that \mathbf{K} collapses to a subcomplex \mathbf{L} if there is a sequence of elementary collapses

$$\mathbf{K} = \mathbf{K}_0 \searrow^e \mathbf{K}_1 \searrow^e \dots \searrow^e \mathbf{K}_k = \mathbf{L},$$

and we write $\mathbf{K} \searrow \mathbf{L}$.

It is well known that there is a homotopy equivalence between \mathbf{K} and \mathbf{L} if $\mathbf{K} \searrow \mathbf{L}$.

Definition 2.21. An n -complex \mathbf{K} is said to be collapsible if \mathbf{K} collapses to a point, and we write $\mathbf{K} \searrow 0$ in this case.

2.6.3 Collapsibility and simple points

Let \mathbf{V} be a finite point set and $\text{Comp}_m(\mathbf{V})$ be a discrete polyhedral complex constructed from \mathbf{V} for the m -neighborhood system where $m = 6, 26$ as described in Section 2.2.2.

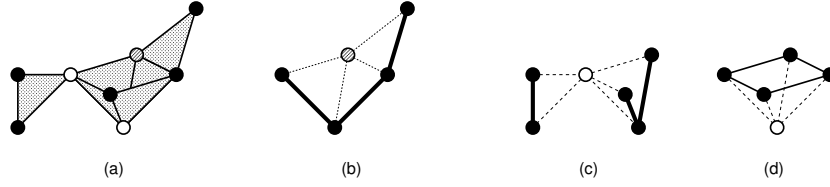


Figure 21: (a) Three points of type 2c which are colored in grey and white, (b) the collapsible link of the grey point, and (c, d) the non-collapsible links of the white points.

Proposition 2.3. *A point $x \in \mathbf{V}$ is m -simple if and only if $\text{Link}(x : \text{Comp}_m(\mathbf{V}))$ is collapsible.*

We have calculated all local point configurations such that $\text{Link}(x : \text{Comp}(\mathbf{V}))$ is collapsible, and have verified that they are the same as those of m -simple points [15]. More precisely, we obtain 550 435 different local point configurations in a $3 \times 3 \times 3$ point region for either case $m = 6, 26$. Remark that this is not a coincidence; we can derive this result from Proposition 2.2 which is obtained by another local characterization. We easily see that any m -simple point for \mathbf{V} is a \bar{m} -simple point for $\bar{\mathbf{V}}$ if we interchange \mathbf{V} with $\bar{\mathbf{V}}$.

Similar characterizations of simple points can be found in [13], [105]. Note that the topological space in [105] is dual to a discrete polyhedral complex for 6-neighborhood [89], [116] so that we have an inclusion relation \prec which is inverse.

2.6.4 Topological characteristics of simple points

In Section 2.3.3, we showed that each point in a point set \mathbf{V} can be classified into one of the twelve types by using the polyhedral-complex representation $\text{Comp}(\mathbf{V})$. In this subsection, we check which types of points are simple.

According to Proposition 2.3, we verify the collapsibility of $\text{Link}(x)$ for every type of points x and then obtain the following theorem [85].

Theorem 2.1. *Every point whose type is either type 1b, 2b or 3b is always a simple point. Contrarily, any point whose type is either 0, 1a, 1c, 2a, 2d or 3a can never be a simple point.*

From the above theorem, we see that points of types 2c, 3c and 3d, differing from the other types, have both cases which are simple and not simple. Figures 21, 22 and 23 show examples of simple and non-simple points for types 2c, 3c and 3d respectively. The examples illustrate that the connectivity of $\text{Link}(x)$ is a necessary condition but not a sufficient one for the collapsibility of $\text{Link}(x)$.

From Theorem 2.1, we also see that simple points can be of the six different types 1b, 2b, 2c, 3b, 3c, 3d. Table 6 shows the numbers of all different local configurations of simple points for each point type.

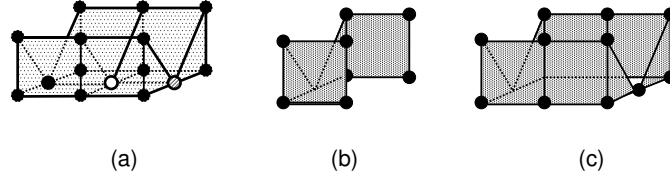


Figure 22: (a) Two points of type 3c which are colored with grey and white, (b) the collapsible link of the grey point, and (c) the non-collapsible link of the white point.

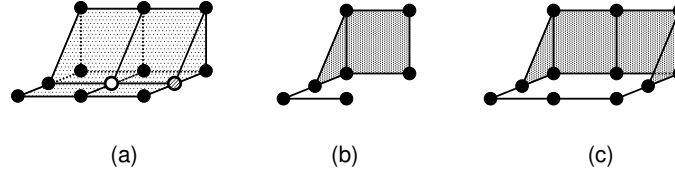


Figure 23: (a) Two points of type 3d which are colored in grey and white, (b) the collapsible link of the grey point, and (c) the non-collapsible link of the white point.

2.6.5 Linear thinning algorithm

Given a finite subset \mathbf{V} in \mathbb{Z}^3 , we present a linear algorithm for thinning \mathbf{V} . In Algorithm ??, we require a list \mathbf{P} for deletable point candidates and also a Boolean function $f : \mathbf{V} \rightarrow \mathbb{B}$ for renewing the deletability of $x \in \mathbf{V}$ after removing one of its neighboring points.

In Step 12 of Algorithm ??, we assume that points y whose types can be changed due to removing x from \mathbf{V} are in a neighborhood of x , i.e. $\mathcal{N}(x)$. If we use a method in the framework of either of a Khalimsky topology [97], a partially ordered set [13] or a discrete polyhedral complex [89], [116] for construction of a polyhedral complex $Comp(\mathbf{V})$, such a neighborhood $\mathcal{N}(x)$ can be considered to be the 26-neighborhood.

Table 6: The numbers of local point configurations of simple points for each point type with respect to 6- and 26-neighborhood systems.

	6-neighborhood	26-neighborhood
type 1b	134 280	3
type 2b	345 016	398
type 2c	28 994	1 037
type 3b	14 031	290 979
type 3c	332	28 525
type 3d	27 782	229 493
total	550 435	550 435

Algorithm 1: Thinning

```

input : a point set  $V \subset \mathbb{Z}^3$ 
output: a thinned set  $V$ 
1 begin
2   obtain the set  $P$  of simple points that are not endpoints in  $V$ ;
3   foreach  $x \in V$  do
4     if  $x \in P$  then
5        $f(x) \leftarrow \text{True}$ ;
6     else
7        $f(x) \leftarrow \text{False}$ ;
8   while  $P \neq \emptyset$  do
9     select a point  $x \in P$  and  $P \leftarrow P \setminus \{x\}$ ;
10    if  $f(x) = \text{True}$  then
11       $V \leftarrow V \setminus \{x\}$  (change the value of  $x$  from 1 to 0);
12      foreach  $y \in N(x) \cap V$  do
13        if  $f(y) = \text{True}$  and  $y$  is not simple or is an endpoint
14          then
15             $f(y) \leftarrow \text{False}$ ;
16          else if  $f(y) = \text{False}$  and  $y$  is simple but not an
17            endpoint then
18               $P \leftarrow P \cup \{y\}$  and  $f(y) \leftarrow \text{True}$ ;
19  return  $V$ ;

```

Obviously, the result of Algorithm ?? depends on a point $x \in P$ selected in Step 9. If we set no endpoint, thinning results are topologically equivalent with respect to the initial set V . Therefore, we can simply realize P as a queue in the case that we are interested in only topological results. However, if we set endpoints and our interests are not only topology but also geometry, we may need to realize P as a priority queue whose priorities depend on distances from the complement \bar{V} , for example.

Applying Algorithm ??, we can obtain a curve or surface skeleton of an initial set V , depending on the definition of endpoints. Thanks to the results of topological point classification in Section 2.3.3, we can set endpoints easily by dimensions and topological characteristics of stars. If we set endpoints to have type 1b (semi-linear), Algorithm ?? behaves as a curve thinning. Similarly, if we set endpoints to have type 1b and 2b (semi-linear and semi-cyclic), it behaves as a surface thinning.

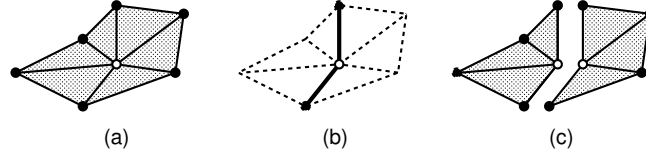


Figure 24: An example of semi-cyclic star generation for the 26-neighborhood system; (a) a cyclic star, (b) a linear star having the common central point of (a), and (c) two new semi-cyclic stars made by cutting (a) by (b).

2.6.6 Setting of endpoints

Such intuitive settings of endpoints, however, do not always meet our expectations in practice. For example, it is very rare that we use the 6-neighborhood system for the curve/surface thinning because thinning results generally contain too many small parts because of too many configurations of endpoints; see in Table 6 that there are much more configurations of types 1b and 2b for the 6-neighborhood than those for the 26-neighborhood. For the 26-neighborhood system, the curve thinning works very well (see Fig. 25) while the surface thinning does not. This is because we do not have enough endpoints (type 2b) for the 26-neighborhood system as shown in Table 6. To obtain those semi-cyclic points (type 2b), we construct a polyhedral complex as a collection of convex polyhedra each of which is locally made from a set of points in \mathbf{V} at a unit cubic region. Therefore, a constructed polyhedron tends to have three dimensions rather than less than three dimensions for any point configuration.

We therefore propose a simple method to obtain more configurations for semi-cyclic (type 2b) points. In [116], we obtain all possible configurations of discrete surfaces which appear on the boundaries of 3D discrete objects and whose central points have cyclic stars on the surfaces: 6 and 6028 configurations for the 6- and 26-neighborhood systems, respectively. We cut each cyclic star (a discrete surface) by a linear star (a discrete line) having the common central point and create additional semi-cyclic stars. Figure 24 illustrates an example for such a semi-cyclic star generation. Then we obtain 22 399 configurations instead of 398 for type 2b in Table 6.

With these new semi-cyclic points, we obtain a surface thinning result in Figs. 25, 26. In these examples, priorities of \mathbf{P} categorized by 26 directions are used. Figure 26 illustrates that we may dig a hole at the intersection of digitized planes depending on the rotation of digitized planes and the number of digitized planes. This is caused by the image discreteness: locally, we cannot distinguish between a 3D part and an intersection of two 2D parts if they have the same local point configuration. In order to solve the problem, additional topo-

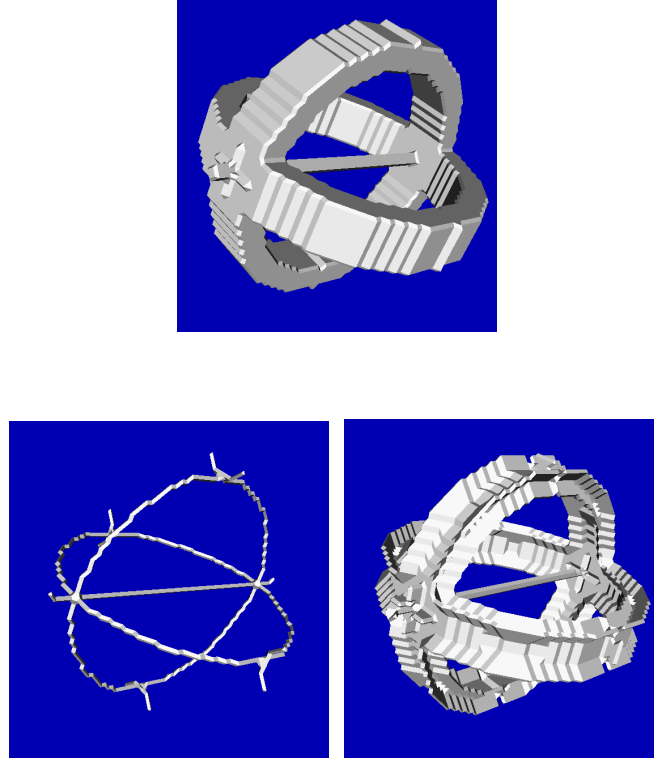


Figure 25: The original 3D image (top), its curve thinning result (bottom left), and its surface thinning result (bottom right) for the 26-neighborhood system.

logical configurations for surface intersections [117] or supplementary geometrical concepts will be necessary.

2.7 SUMMARY AND PERSPECTIVES

Given a subset $V \subseteq \mathcal{N}_{26}(x)$, we presented a method for classifying the central point x into one of the twelve types by the topological characterization of its star after constructing a polyhedral complex $K_m = \text{Comp}_m(V)$. Considering that boundary points having type 3b form discrete combinatorial surfaces as Proposition 2.1, we enumerated local configurations on discrete surfaces such as local point configurations whose central point has type 3b, and obtained 9, 102 793 and 290 979 semi-spherical star configurations, $Sk_0(Cl(Star(x : K_m)))$, and 6, 1412 and 6028 surface star configurations, $Sk_0(Cl(Star(x : \partial K_m)))$, for $m = 6, 18, 26$, respectively. The same surface star configurations for $m = 6$ are already presented in [64] and they are illustrated in Fig. 4. We see that a boundary point illustrated as the central point in Fig. 3 (center) has a surface star configuration. This explains why our

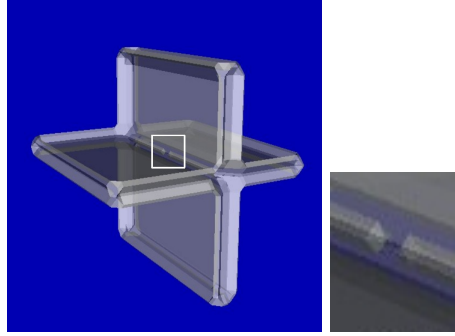


Figure 26: A surface thinning result for two intersected digitized planes for the 26-neighborhood system (left) and the magnification of the white square (right). A transparency is given to make easy to see the interior such that there are two deep holes dug from the surface edges at the intersection and only one point is connected to surfaces.

discrete surfaces have more configurations than those of simplicity surfaces [42].

Thanks to the local topological characterization of points, the proposed framework has several utilities: it enables us to propose a linear thinning algorithm of three-dimensional digital images based on collapsibility of polyhedral complexes [85], and to define topologically reasonable discrete surface patches, on each of which geometrical measures, such as normal vector, can be calculated in a finite way [87] (see the next Chapter for more details).

The major drawback of this approach is that it is not well adapted when several objects are located adjacently with sharing their boundaries; indeed, if we apply the proposed method, then each object has its own boundary and there exists a non-empty space between the two boundaries. In such a situation, inter-voxel approaches such as cubical complexes, which are considered as dual approaches of ours, are often taken [120], as already mentioned in Section 2.1. Despite such inconvenience, in case one discrete object is treated, this discrete approach is effective.

As a matter of fact, even the inter-voxel approach of cubical complex would cause topological problems, such that combinatorial boundary of a three-dimensional complex does not constitute combinatorial manifolds. To avoid such topological conflicts, for example, the notion of well-composedness has been proposed [115], and a similar notion could be found for the proposed framework. For example, a digital version of the regularity notion, which is originally proposed for preserving topology under two-dimensional rigid motions [133], can be a good starting point.

DIGITAL SHAPE ANALYSIS WITH GEOMETRIC ASSUMPTION

Given digital volume data, in this chapter, we perform shape analysis under some geometric assumption. In other words, we know a priori geometric information about what we would like to observe in data: for example, linear shapes as the simplest, namely planes in a 3D space. Two different problems are considered here: digital planar surface segmentation and digital plane fitting. The former assumes local constraints of the shape linearity, while the latter leads to a global optimization under the linear shape constraint. Both use the common geometrical model, digital plane, formulated by a pair of linear inequalities.

We first introduce the analytic (geometric) models of digital hyperplanes, as well as arithmetic hyperplanes, which are formulated by pairs of linear Diophantine inequalities. Using the analytic digital models, we solve the above geometric problems in a discrete manner. The solutions to the first problem is based on the results mainly presented in [86], [87], [90], while the second is based on those in [2], [88], [206].

3.1 ANALYTIC MODEL OF DIGITAL HYPERPLANES

Before introducing analytic models of digital hyperplanes, we give the analytic definition of continuous hyperplanes in Euclidean space.

3.1.1 Continuous hyperplanes

A hyperplane in Euclidean space \mathbb{R}^d , $d \geq 2$, is defined by

$$\mathbf{H} = \{(x_1, x_2, \dots, x_d) \in \mathbb{R}^d : \sum_{i=1}^d a_i x_i + a_{d+1} = 0\} \quad (4)$$

with $a_i \in \mathbb{R}$, $i = 1, \dots, d+1$. Note that it is common to add a normalization constraint such as

$$\sum_{i=1}^d |a_i| = 1$$

or

$$\sum_{i=1}^d a_i^2 = 1.$$

In this chapter, we use the following normalization instead of the above conventional ones:

$$\forall i \in \{1, \dots, d\}, -1 \leq a_i \leq 1 \bigwedge \max(a_i) = 1. \quad (5)$$

All of the above normalization techniques enable us to bound the range of every coefficient between -1 and 1 , except for a_{d+1} : practically a_{d+1} is also bounded by the size of an input image.

3.1.2 Digital hyperplanes

A digital hyperplane is the digitization of a hyperplane \mathbf{H} in the discrete space \mathbb{Z}^d . The following definition was proposed by Stojmenovics et al. [187].

Definition 3.1. A digital hyperplane of a hyperplane \mathbf{H} is defined by the set of discrete points satisfying two inequalities:

$$\mathfrak{D}(\mathbf{H}) = \{(p_1, p_2, \dots, p_d) \in \mathbb{Z}^d : 0 \leq \sum_{i=1}^d a_i p_i + a_{d+1} + \frac{w}{2} < w\}$$

where $a_i \in \mathbb{R}$, $i = 1, \dots, d+1$, and w is a constant, called the width of $\mathfrak{D}(\mathbf{H})$. The width is set as

$$w = \max_{i=1, \dots, d} (|a_i|).$$

This digitization operation is also seen as the grid-intersection digitization [170], which is one of popular digitization schemes of curves and surfaces [101]. Indeed, $\mathfrak{D}(\mathbf{H})$ is the set of all grid points that are closest to the intersection points of \mathbf{H} with the grid lines of \mathbb{Z}^d . Note that we always have $w = 1$ for $\mathfrak{D}(\mathbf{H})$ with the above setting if the normalization of Eq. (5) is considered for \mathbf{H} . A set of integer points is also said to be digitally flat [193] if every point of the set satisfies the inequalities of Definition 3.1 and $w = 1$ with its width setting.

If the standard digitization [8], which is a modification of the supercover (or outer Jordan) digitization due to avoiding "thicker parts" called bubbles, is considered, the width is set by

$$w = \sum_{i=1}^d |a_i|, \quad (6)$$

instead of the one in Definition 3.1.

From the definition, we obviously obtain a unique digital hyperplane $\mathfrak{D}(\mathbf{H})$ from any \mathbf{H} , but the converse is not true. Let us consider the following two types of digital hyperplanes depending on the coefficients. If $\forall i \in \{1, \dots, d\}$, $a_i \in \mathbb{Q}$, then $\mathfrak{D}(\mathbf{H})$ is called a rational digital hyperplane, and otherwise, an irrational digital hyperplane. Then, we have the following property [28], [101].

Property 3.1. *If $\mathfrak{D}(\mathbf{H})$ is irrational, then $\mathfrak{D}(\mathbf{H})$ uniquely determines all the coefficients a_i , $i = 1, \dots, d+1$. If $\mathfrak{D}(\mathbf{H})$ is rational, then $\mathfrak{D}(\mathbf{H})$ uniquely determines the coefficients a_i , $i = 1, \dots, d$ and a_{d+1} up to an interval.*

The difference between rational and irrational cases, in particular for $d = 2, 3$, has been studied by taking the approaches such as number theory and the theory of words. It is known that a rational digital hyperplane is periodic while an irrational digital hyperplane is aperiodic (or 1D periodic for $d = 3$); the latter is related to Sturmian words, which are aperiodic infinite words [25], [27]. For surveys on this topic, we refer to [101] and in particular [102] for $d = 2$ and [25] for $d = 3$.

As our observation of grid points, given as pixels in digital images, is generally limited in a finite space, treating only rational digital hyperplanes is sufficient from a viewpoint of digital image analysis. In such a case, we can also use another model for digital hyperplanes, which is called arithmetic hyperplanes.

3.1.3 Arithmetic hyperplanes

There is another model for digital hyperplanes, which is not defined through a digitization scheme as above, but defined completely in a discrete manner with only integer coefficients, i.e. with a pair of Diophantine inequalities [7], [157].

Definition 3.2. *An arithmetic hyperplane is defined as*

$$\mathcal{A} = \{(p_1, p_2, \dots, p_d) \in \mathbb{Z}^d : 0 \leq \sum_{i=1}^d \alpha_i p_i + \alpha_{d+1} < \omega\}$$

where $\alpha_i \in \mathbb{Z}$ for all $i = 1, \dots, d+1$ such that $\gcd(\alpha_1, \dots, \alpha_d) = 1$. We call ω the width of \mathcal{A} , and set

$$\omega = \max_{i=1, \dots, d} (|\alpha_i|).$$

This setting is similar to w for $\mathfrak{D}(\mathbf{H})$ in Definition 3.1. When the width is set as mentioned above, \mathcal{A} is called a naive hyperplane [7]. Similarly to Eq. (6), we can also set ω by

$$\omega = \sum_{i=1}^d |\alpha_i|. \quad (7)$$

Then \mathcal{A} is called a standard hyperplane [7].

The width setting influences the connectivity of a digital hyperplane [7], [101]. We also need the following concept of tunnels¹, in particular, for $d \geq 3$ [7].

¹ The tunnels are also called “gaps” in [24].

Definition 3.3. *An arithmetic hyperplane of coefficients $\alpha_i, i = 1, \dots, d + 1$, has a k -tunnel if there are two distinct k -neighbors, $(p_1, p_2, \dots, p_d), (q_1, q_2, \dots, q_d) \in \mathbb{Z}^d$, such that*

$$\sum_{i=1}^d \alpha_i p_i + \alpha_{d+1} < 0 \quad \wedge \quad \sum_{i=1}^d \alpha_i q_i + \alpha_{d+1} \geq \omega.$$

Using the above notion of k -tunnels, we have the following (topological) connectivity property for $d = 2, 3$ [7], [157].

Property 3.2. *Let \mathcal{A} be an arithmetic hyperplane with a width ω .*

- *If $\omega = \max_{i=1, \dots, d} (|\alpha_i|)$, \mathcal{A} is a minimal $(2d + d^d)$ -connected set (without $2d$ -tunnel for $d = 3$).*
- *If $\omega = \sum_{i=1}^d |\alpha_i|$, \mathcal{A} is a minimal $(2d)$ -connected set (without tunnel for $d = 3$).*

Here, we say that a connected set $X \subset \mathbb{Z}^d$ is minimal if there is no simple point in X . In other words, if we take out any point $p \in X$, then $X \setminus \{p\}$ has a different topology from that of X ($X \setminus \{p\}$ does not separate $\overline{X} \cup \{p\}$ into two parts anymore while X separates $\overline{X} = \mathbb{Z}^d \setminus X$ into two.).

The following proposition makes a link between digital hyperplanes and arithmetic planes [7], [157].

Proposition 3.1. *Every rational digital hyperplane is a naive hyperplane, and vice-versa.*

Thanks to this, we can enumerate all digital hyperplanes of a finite size, called linear local geometric patterns, as seen in the next section for $d = 3$.

Hereafter, we focus on the cases of $d = 2, 3$ in particular, and digital/arithmetic hyperplanes when $d = 2, 3$ are called digital/arithmetic lines and planes, respectively.

3.2 LOCAL GEOMETRIC PATTERNS OF DIGITAL PLANES

3.2.1 Preimages of a digital plane patch

Let us consider a plane $\mathbf{P} \subset \mathbb{R}^3$ and its digitization $\mathfrak{D}(\mathbf{P})$. Since we observe $\mathfrak{D}(\mathbf{P})$ in a finite grid space $\mathbf{X} \subset \mathbb{Z}^3$ in this section, \mathbf{X} is bounded such that $\mathbf{X} = \Pi_{i=1,2,3} [X_i^-, X_i^+] \cap \mathbb{Z}^3$ where X_i^-, X_i^+ are finite integers with $X_i^- < X_i^+$. Then, our digital plane $\mathfrak{D}(\mathbf{P})$ is also bounded such that

$$\mathfrak{D}_{\mathbf{X}}(\mathbf{P}) = \mathfrak{D}(\mathbf{P}) \cap \mathbf{X}, \quad (8)$$

called a digital plane patch.

Given $\mathfrak{D}_X(\mathbf{P})$, we can find a set of Euclidean planes \mathbf{P} such that the digitization of each \mathbf{P} in X is equal to $\mathfrak{D}_X(\mathbf{P})$. The set of all such \mathbf{P} is called the preimage of $\mathfrak{D}_X(\mathbf{P})$ [35]. Note that the correspondence between $\mathfrak{D}_X(\mathbf{P})$ and \mathbf{P} is not one-to-one but one-to-many, as mentioned above. Thus, the preimage of $\mathfrak{D}_X(\mathbf{P})$ is represented by a set of feasible parameters α_i , $i = 1, \dots, 4$, such that all points of $\mathfrak{D}_X(\mathbf{P})$ satisfy the inequalities in Definition 3.2. It means that the preimage is given by a convex polytope in the parameter space [35]. Because all interesting parameters in this paper are translation-invariant, we focus on the three parameters α_i , $i = 1, 2, 3$, indicating the normal vector of \mathbf{P} , distinguished from the intercept α_4 of \mathbf{P} . Further discussion is given in Section 3.3.4.

3.2.2 Local geometric patterns and their linearity

We define a local point set around a point p in \mathbb{Z}^3 , such that

$$\mathbf{Q}_k(p) = \{q \in \mathbb{Z}^3 : \|p - q\|_\infty \leq k\} \quad (9)$$

where k is a positive integer, $k \in \mathbb{Z}^+$. $\mathbf{Q}_k(p)$ is a cubical grid-point set whose edge length is $2k + 1$. Note that $\mathbf{Q}_1(p) = \mathcal{N}_{26}(p)$. Let us consider that each grid point in \mathbb{Z}^3 has a binary value such as either 1 or 0. Such a pattern of binary points in $\mathbf{Q}_k(p)$ is called local geometric patterns, abbreviated to LGP hereafter. There are $2^{(2k+1)^3-1}$ different LGP for $\mathbf{Q}_k(p)$, provided that the central point p always has a fixed value, such as 1. This indicates that p is considered to be not a background point but an object point.

In this section, we investigate, among these $2^{(2k+1)^3-1}$ LGP, which LGP can appear on digital planes. Note that we set binary values of points of $\mathbf{D}_{\mathbf{Q}_k(p)}(\mathbf{P})$ to be 1 and those of other points to be 0. This problem is mathematically written as follows. Let \mathbf{F} be a set of points whose values are 1 in $\mathbf{Q}_k(p)$. If there is a digital plane patch

$$\mathfrak{D}_{\mathbf{Q}_k(p)}(\mathbf{P}) = \mathbf{F},$$

we say that \mathbf{F} forms a digital plane patch in $\mathbf{Q}_k(p)$. Therefore, our problem is solved by looking for all possible \mathbf{F} , namely LGP, forming digital plane patches, i.e. satisfying the inequalities:

$$0 \leq \sum_{i=1}^3 a_i p_i + a_4 + \frac{w}{2} < w \quad (10)$$

in Definition 3.1 with $d = 3$. Such LGPs are called linear LGPs. Since this problem is considered to be the feasibility of the above inequalities for all $(p_1, p_2, p_3) \in \mathbf{F}$, we check if there are feasible solutions a_i , $i = 1, \dots, 4$ for each different \mathbf{F} , namely LGP. If they exist, such an LGP can appear on digital planes and becomes a linear LGP.

3.2.3 Linear LGP generation by arithmetic planes

In this section, however, in order to avoid computing the feasibility test for all $2^{(2k+1)^3-1}$ LGPs of $\mathbf{Q}_k(\mathbf{p})$, we consider another approach to generate all linear LGPs, which is based on arithmetic planes [101], [156], and similar to [47].

From the discussion in Section 3.2.1, we know that there are many possible \mathbf{P} corresponding to a given $\mathfrak{D}_{\mathbf{Q}_k(\mathbf{p})}(\mathbf{P})$ and that such preimage is represented by a set of feasible parameters α_i , $i = 1, \dots, 4$. This implies that, given $\mathfrak{D}_{\mathbf{Q}_k(\mathbf{p})}(\mathbf{P})$, we can find a corresponding \mathbf{P} with only rational parameters. In addition, the denominators of those rational numbers are bounded by the size of $\mathbf{Q}_k(\mathbf{p})$, namely k . Furthermore, it is also known from Proposition 3.1 that a digital plane $\mathfrak{D}(\mathbf{P})$ with rational slopes is equivalent to an arithmetic plane \mathcal{A} .

Based on this fact, we generate all linear LGPs by using naive planes \mathcal{A} instead of digital planes $\mathfrak{D}(\mathbf{P})$. Our algorithm mainly consists of the following three steps:

1. set parameters $\alpha_i \in \mathbb{Z}$ for $i = 1, \dots, 4$;
2. from those parameters, construct a grid point set \mathcal{A} ;
3. for each point $\mathbf{p} \in \mathcal{A}$, observe the LGP of $\mathbf{Q}_k(\mathbf{p})$ for a given k .

In the following, we detail each of steps 1 and 2.

3.2.3.1 Parameter setting for naive planes

As mentioned in Section 3.2.1, we distinguish the three parameters α_i for $i = 1, \dots, 3$ indicating a normal vector of \mathcal{A} from the intercept α_4 indicating a translation of \mathcal{A} . We are first concerned with the setting of value α_4 . It is known that \mathcal{A} always has grid points $(p, q, r) \in \mathbb{Z}^3$ satisfying the following equality:

$$\alpha_1 p + \alpha_2 q + \alpha_3 r + \alpha_4 = 0, \quad (11)$$

called leaning points [156]. Chosen a leaning point $\mathbf{p} \in \mathcal{A}$, even if we translate \mathcal{A} with a vector $-\mathbf{p}$ so that the origin becomes a leaning point, it is certain that such a translation does not influence LGPs on \mathcal{A} . We therefore simply set

$$\alpha_4 = 0 \quad (12)$$

so that we consider only naive planes \mathcal{A} where the origin is a leaning point.

Concerning the other parameters $\alpha_i \in \mathbb{Z}$, $i = 1, 2, 3$, we give the following constraints:

$$0 \leq \alpha_1 \leq \alpha_2 \leq \alpha_3, \alpha_3 \neq 0. \quad (13)$$

All naive planes which do not satisfy (13) can be generated from the naive planes with the constraints (13) by their rotations around the origin, since we set $\alpha_4 = 0$, and their symmetries with respect to the xy -, yz - and xz -planes. We can also bound α_3 such that

$$\alpha_3 \leq 8k^2 \quad (14)$$

from the size of $\mathbf{Q}_k(\mathbf{p})$ [29]. Therefore, once the value of k is given, we can automatically generate a set of all relatively-prime integer triplets $(\alpha_1, \alpha_2, \alpha_3)$, denoted by V_k , by using the Euclidean algorithm, with the constraints (13) and (14).

3.2.3.2 Constructed part of a naive plane

With the constraint (13), we see that the principal projection plane of \mathcal{A} is the xy -plane. Then, it is known that there are at most $(2k+1)^2$ different LGPs on \mathcal{A} and that all different LGPs can appear in the region which is projected in the principal projection plane, i.e. in the xy -plane, as a $(4k+1) \times (4k+1)$ squared region and whose central point is a leaning point of \mathcal{A} , i.e. the origin [195].

Concerning to z -coordinates, thanks to the periodicity of \mathcal{A} [101], we see that the maximum difference of the z -coordinates of any pair of points in the region which is projected in the xy -plane as a $(4k+1) \times (4k+1)$ rectangle does not exceed twice those of the x - and y -coordinates. Therefore, we can set a constructed part of \mathcal{A} in a finite grid space \mathbf{X}_k such that

$$\mathbf{X}_k = ([-2k, 2k] \times [-2k, 2k] \times [-4k, 4k]) \cap \mathbb{Z}^3. \quad (15)$$

3.2.4 Algorithm and results

From the above discussion, we now present Algorithm ?? for generating all linear LGPs of $\mathbf{Q}_k(\mathbf{x})$ for a given k . By executing Algorithm ??, we obtain 34 linear LGPs for $k = 1$, similarly 1574 for $k = 2$, 23 551 for $k = 3$, and 181 735 for $k = 4$, up to translations, rotations and symmetries, thanks to the constraints (13). Figure 27 shows all linear LGPs for $k = 1$ and Fig. 28 shows some examples for $k = 2$. In order to visualize their shapes in the figures, we add polyhedral meshes which are made by applying a discrete version of the marching cubes method for the 18-neighborhood system [89] to a digitized half space.

Remark that our LGP around a point $\mathbf{p} = (p_1, p_2, p_3)$ in \mathcal{A} for a given k is slightly different from the $(2k+1, 2k+1)$ -cube [195], which is defined as a set of points $\mathbf{q} = (q_1, q_2, q_3)$ in \mathcal{A} such that $|p_1 - q_1| \leq 1$ and $|p_2 - q_2| \leq 1$, under the constraints (13). A linear LGP of $\mathbf{Q}_k(\mathbf{p})$ can be smaller than a $(2k+1, 2k+1)$ -cube, so that there are less linear LGPs than $(2k+1, 2k+1)$ -cubes; for example, there are 40 $(3, 3)$ -cubes, called tricubes [47], [195], against the 34 linear LGPs for $k = 1$.

Algorithm 2: Generation of all linear LGPs

```

input : a size  $k$  of  $\mathbf{Q}_k(x)$ 
output: a set  $T$  of linear LGPs
1 begin
2   initialize a set  $T$ ;
3   make a set of integer normal vectors  $V_k$  with the constraints
      (13) and (14) ;
4   foreach  $(\alpha_1, \alpha_2, \alpha_3) \in V_k$  do
5     construct a finite grid-point set  $\mathcal{A}$  in  $\mathbf{X}_k$  ;
6     foreach  $x \in \mathcal{A} \cap \mathbf{X}_k$  do
7       if LGP of  $\mathbf{Q}_k(x)$  is not included in  $T$  then put it in  $T$ ;
8   return  $T$ ;

```

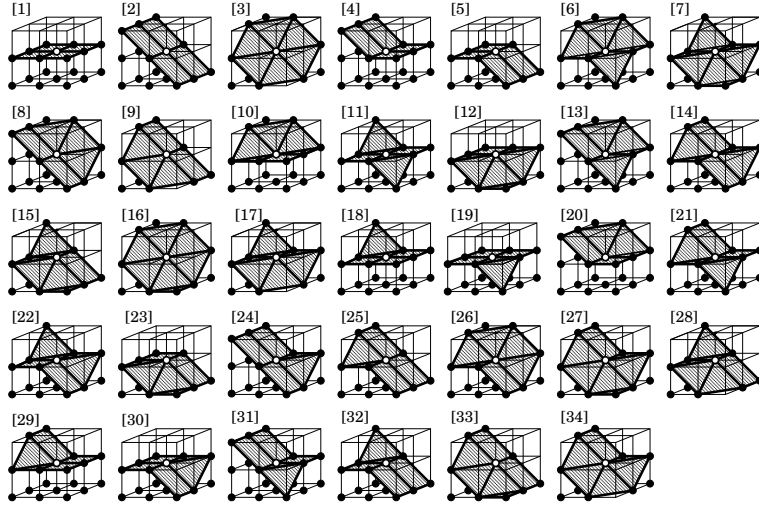


Figure 27: The 34 linear LGPs for $k = 1$ up to translations, rotations and symmetries, with polyhedral meshes made by a discrete version of the marching cubes method [89].

3.2.5 Topological property of digital planes

For $d = 3$ in particular, there is also the interesting property on topological characterization of grid points on naive planes [91].

Let us consider a naive plane \mathcal{A} with integer parameters α_i , $i = 1, \dots, 4$, and its associated digitized half-space \mathbf{I} such that

$$\mathbf{I} = \{(p_1, p_2, p_3) \in \mathbb{Z}^3 : \sum_{i=1}^3 \alpha_i p_i + \alpha_4 \geq 0\}.$$

To this \mathbf{I} , we apply the algorithm of constructing the polyhedral complex, $\text{Comp}_m(\mathbf{I})$ with setting either $m = 18$ or 26 , presented in Section 2.2. Because of the geometry of \mathbf{I} , we have the following lemma.

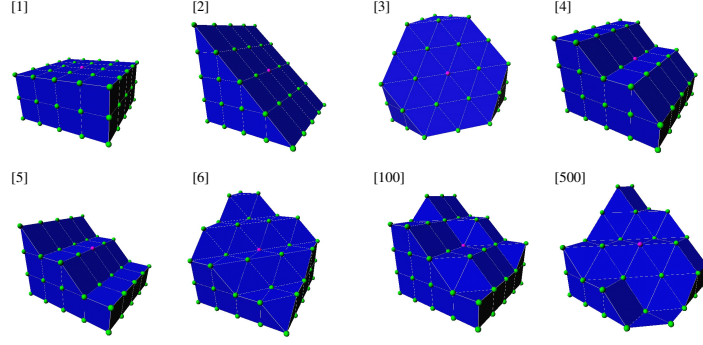


Figure 28: Examples of the linear LGPs for $k = 2$ up to translations, rotations and symmetries, with polyhedral meshes made by a discrete version of the marching cubes method [89].

Lemma 3.1. *For any digitized half-space \mathbf{I} , its polyhedral complex $\text{Comp}_m(\mathbf{I})$ with either $m = 18$ or 26 is a pure polyhedral 3-complex.*

Then, we also have the following lemma concerning the combinatorial boundary.

Lemma 3.2. *For any digitized half-space \mathbf{I} , the combinatorial boundary of the polyhedral complex of \mathbf{I} , $\partial\text{Comp}_m(\mathbf{I})$, is also pure, namely, a pure polyhedral 2-complex.*

We then have the following relations [91].

Property 3.3. *For any naive plane \mathcal{A} , which associates a half-space \mathbf{I} , we have:*

$$\mathcal{A} = \text{Sk}_0(\partial\text{Comp}_{18}(\mathbf{I})) = \text{Sk}_0(\partial\text{Comp}_{26}(\mathbf{I})).$$

This property together with Lemma 3.2 and the observation of Figure 27 lead us to the next (combinatorial) topological property of naive planes [90].

Property 3.4. *Given a naive plane \mathcal{A} , let us consider its associated topological structure $\partial\text{Comp}_m(\mathbf{I})$. Then, for any discrete point $\mathbf{p} \in \mathcal{A}$, $\text{Star}(\mathbf{p} : \partial\text{Comp}_m(\mathbf{I}))$ is cyclic. In other words, $\text{Star}(\mathbf{p} : \text{Comp}_m(\mathbf{I}))$ is semi-spherical.*

This result is not against intuition. This also implies that if we need a topological structure of a naive plane, we can apply the discrete version of the marching cubes algorithm to the associated digitized half-space in order to obtain the polyhedral surface $\partial\text{Comp}_m(\mathbf{I})$, which can be considered as a combinatorial 2-manifold.

3.3 DIGITAL PLANAR SURFACE SEGMENTATION

In this section, we focus on $d = 3$ in particular. We first characterize local geometric patterns on digital planes and then present a method

for segmenting a 3D grid-point cloud into planar surfaces by using those patterns. The latter segmentation problem is called surface segmentation, which is one of classical problems in computer vision.

3.3.1 *Related work and our approach*

In computer vision, conventional methods for surface segmentation are classified into three categories: region-based, edge-based, and hybrid methods. The first ones merge points having similar region properties calculated from their neighboring points such as normal vectors and curvatures [19]. As calculated properties are sensitive to noise and quantization errors, they cause over-segmentation. The second methods search edges that separate regions by using depth discontinuities [201]. As edges are not always extracted as connected curves, they cause under-segmentation. The third methods are hybrid between the two [183], [200]. In particular, for the case where our interesting object is polyhedral, a hybrid method using locally planar points is proposed [183]. In that method, points not locally planar are considered to be potential edge points.

In discrete geometry, a digital plane is defined as a set of grid points lying between two parallel planes with a small distance [101]. Local geometric patterns (LGPs) appearing on digital planes are called linear LGPs, and their number is finite. It is known that linear LGPs are related to arithmetic planes [156]. Their arithmetic properties were studied for digital plane recognition [47], [195], and used to develop region-based methods for digital planar surface segmentation [177]. However, those region-based methods require an incremental plane recognition process, which causes another problem of incremental point tracking.

Here, to avoid plane recognition involving incremental point tracking for segmentation, we present a discrete version of the hybrid methods, consisting of an edge-based and a region-based parts, using linear LGPs. Our idea is similar to [183]. We first generate all linear LGPs in a cubic region of $(2k + 1) \times (2k + 1) \times (2k + 1)$ grid points for an arbitrary size k , by using their arithmetic properties. We then reject a point from a grid-point set if its LPG is not linear; since rejected points define candidates of edge points (the edge-based part), the remaining non-rejected points define candidates of planar points. For the region-based part, we use the normal vectors of linear LPGs. Each linear LPG possesses a set of feasible normal vectors, called preimages [35]. We merge non-rejected points whose linear LGPs have common normal vectors to obtain digital planar surfaces. We show our experimental results demonstrating that our method is robust against not only quantization errors but also noise. As our method gives a rough segmentation result with less computation, it may be

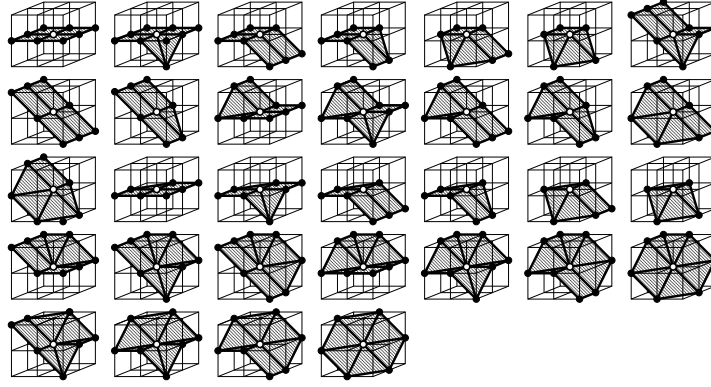


Figure 29: The 32 local star configurations which appear in discrete combinatorial planes for the 26-neighborhood system.

useful to apply our method to obtain an initial segmentation before applying plane recognition.

3.3.2 Linear and non-linear LGPs

From the above results, we see that there are a few linear LGPs relatively to non-linear ones. However, we have learned from experience that many border points of a digital object have linear LGP, if its object surface is very smooth. Indeed, this is not difficult to understand, since any local surface patch on a smooth surface can be approximated to a planar surface when the size of the patch becomes small. In other words, even if a point has a linear LGP, we are uncertain whether such a point appears on a planar surface or a non-planar surface. Contrarily, if a point has a non-linear LGP, it is certain that such a point can never appear on a planar surface.

Together with topological point characterizations based on polyhedral complexes, presented in Chapter 2, we also lead that there are 5 and 32 different configurations of stars which appear in discrete combinatorial planes for $m = 6, 26$, respectively [65], [90]. Such planar stars for $m = 6$ are shown as the five left configurations in Fig. 4. We also illustrate the 32 configurations of planar stars for $m = 26$ in Fig. 29. Note that oriented surfaces are considered in [65], [90] so that 8 and 34 configurations are obtained for $m = 6, 26$.

Figure 30 shows that, for example, each boundary point appearing at the faces of a digitized cube has one of the 32 planar stars illustrated in Fig. 29. On the other hand, around the vertices and edges of a digitized cube, boundary points have non-planar stars. Figure 30 also shows that many boundary points on non-planar surfaces such as a sphere, a one-sheet hyperboloid and a hyperbolic paraboloid have planar stars rather than non-planar stars. From such experiments, we consider that the study of local configurations of boundary points

in the 26-neighborhood system might be useful for shape analysis of three-dimensional images. We remark that the same shape analysis works for $m = 18$, while it would not be worth doing for $m = 6$ because most of all boundary points of three-dimensional digitized objects have planar stars as illustrated in Fig. 31.

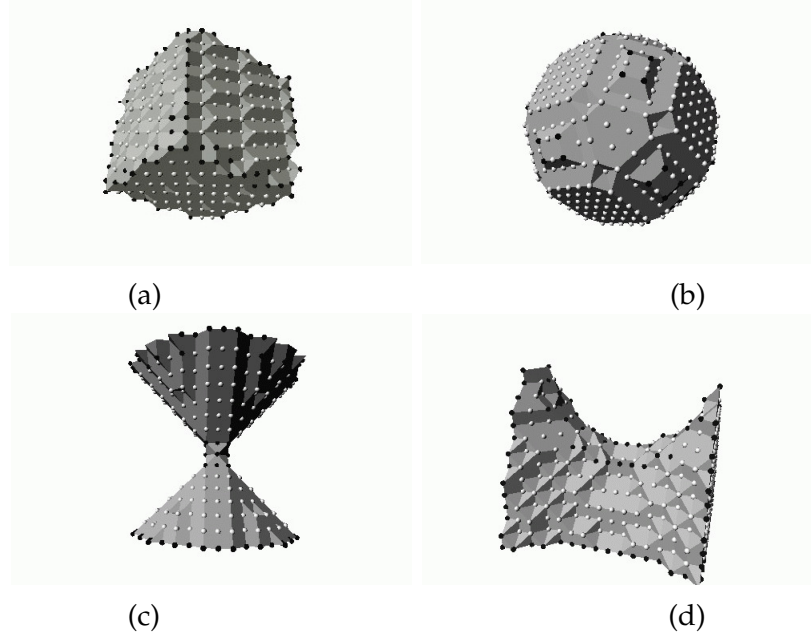


Figure 30: Boundary points of three-dimensional digitized objects, such as a cube, a sphere, a one-sheet hyperboloid and a hyperbolic paraboloid, are classified into two types in the 26-neighborhood system: they are illustrated as white and black points if the stars are planar and non-planar, respectively.

3.3.3 Non-linear point rejection of digital objects

By simply checking the LGP linearity, we can therefore reject non-linear points from a grid-point set, since we know that non-linear points never appear on any discrete plane. In other words, the linear LGPs play an important role in filtering linear points. Note that it is realized by looking up the binary table of LGPs (linear or not).

3.3.3.1 From a point cloud to a grid point set

Before executing the non-linear point rejection to a grid-point set, we describe how to transform a 3D point cloud into a grid-point set. Our input in this paper is a range image represented by a 2D digital image, such that each pixel $(p, q) \in [X_1, X_2] \times [Y_1, Y_2]$ of \mathbb{Z}^2 has a depth information $d(p, q) \in \mathbb{R}^+$ from a 3D scanner to an object surface. We transform such a range image into a 3D triple-valued

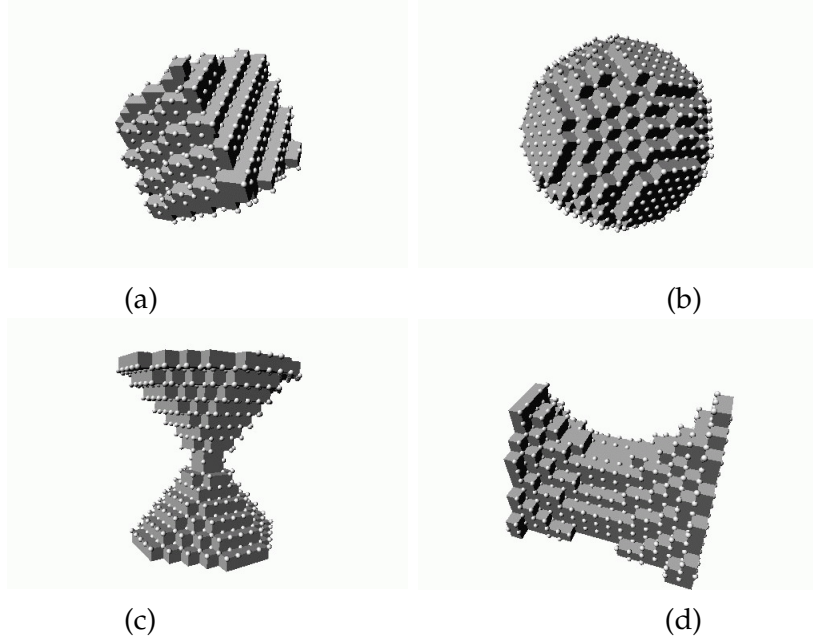


Figure 31: Most boundary points of three-dimensional digitized objects, for example, a cube, a sphere, a one-sheet hyperboloid, and a hyperbolic paraboloid have planar surface stars in the 6-neighborhood system illustrated as white points.

image by re-quantizing a depth $d(x, y)$ as follows: for each point (p, q, r) in a finite subset $\mathbf{X} = [X_1, X_2] \times [Y_1, Y_2] \times [Z_1, Z_2]$ of \mathbb{Z}^3 , we define a triple-valued function such that

$$t(p, q, r) = \begin{cases} 2 & \text{if } r = \lfloor \frac{d(p, q)}{s} + \frac{1}{2} \rfloor, \\ 1 & \text{if } r > \lfloor \frac{d(p, q)}{s} + \frac{1}{2} \rfloor, \\ 0 & \text{otherwise,} \end{cases} \quad (16)$$

where s is a sampling interval for depths.

Grid points whose values are 2 are closer to input points $(p, q, d(p, q)) \in \mathbb{Z}^2 \times \mathbb{R}^+$ so that they are considered to be discrete surface points and to be visible from a 3D scanner. Thus, we call them visible surface points and define a set of visible surface points such that

$$\mathbf{V} = \{(p, q, r) \in \mathbf{X} : t(p, q, r) = 2\}. \quad (17)$$

Concerning grid points whose values are 1, they are invisible from a 3D scanner so that we do not know whether they are surface points or not. Therefore, we simply call them invisible points. Since the rest of grid points whose values are 0 are visible and background points, a set of potential points for an object is defined as a union of visible surface points and invisible points such that

$$\mathbf{W} = \{(p, q, r) \in \mathbf{X} : t(p, q, r) \neq 0\}.$$

Then, the set of surface points is obtained as the border point set $\mathcal{Br}_6(\mathbf{W})$ of Eq. (1). Note that $\mathbf{V} \subseteq \mathcal{Br}_6(\mathbf{W})$ and the equality does not always hold.

A visible surface point set \mathbf{V} can be considered to be a digitization of a point cloud, whereas a surface point set $\mathcal{Br}_6(\mathbf{W})$ is necessary for making binary patterns of LGPs; the binary value of a point p is set to be 1 if $p \in \mathcal{Br}_6(\mathbf{W})$; otherwise, it is set to be 0. This is why we also need $\mathcal{Br}_6(\mathbf{W})$ as well as \mathbf{V} .

3.3.3.2 Experiments

For the experiment, we use a 3D point cloud taken by a 3D scanner Konica-Minolta VIVID 910 with a resolution 320×240 . We first re-quantized the z -coordinates with a similar interval r to those of the x - and y -coordinates from Eq. (16), and obtained two finite grid-point sets, namely, a visible surface point set \mathbf{V} and a surface point set $\mathcal{Br}_6(\mathbf{W})$, from Eqs. (17) and (1). Note that the LGP linearity is checked for every point in \mathbf{V} even if binary patterns for LGPs are made from $\mathcal{Br}_6(\mathbf{W})$.

Figure 32 shows an example of locally linear and non-linear points for $k = 1$, colored in light green and black respectively, in a 3D point cloud taken by a 3D scanner Konica-Minolta VIVID 910 with a resolution 320×240 . We see in the figure that points appearing around polyhedral-face edges are rejected as well as isolated points that are considered to be noise. However, we also observe that some points around edges are not rejected, because they are considered to be locally linear even if they are not linear in a larger region than their LGPs. This fact implies that a simple post-processing, such as the connected component labeling [101] of a non-rejected point set, does not always give satisfactory results for planar surface segmentation.

3.3.4 Linear LGPs and discrete Gaussian spheres

In the continuous framework, each smooth surface point has a unique normal vector, and a mapping from a surface point to its normal vector on the Gaussian sphere is called the Gaussian image. In [75], the extended Gaussian image is presented as a mapping from a point \mathbf{n} on the Gaussian sphere to the area of the surface whose normal vector is \mathbf{n} in order to represent surface shapes. In this article, we need a discrete version of the extended Gaussian image, called the unified discrete Gaussian image, which will be used for planar surface segmentation in the following section. In the discrete framework, each linear LGP has a set of unit normal vectors, called a preimage, and such a preimage is represented as a set of points on the Gaussian sphere. In this section, we first investigate a preimage for each linear LGP, and study how all linear LGPs divide the Gaussian sphere. Such a divided Gaussian sphere is called a discrete Gaussian sphere. We

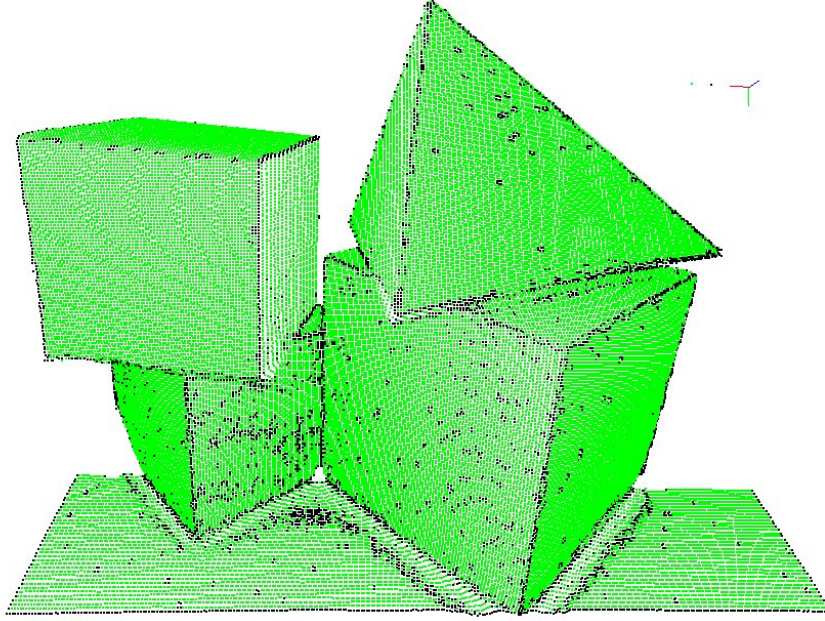
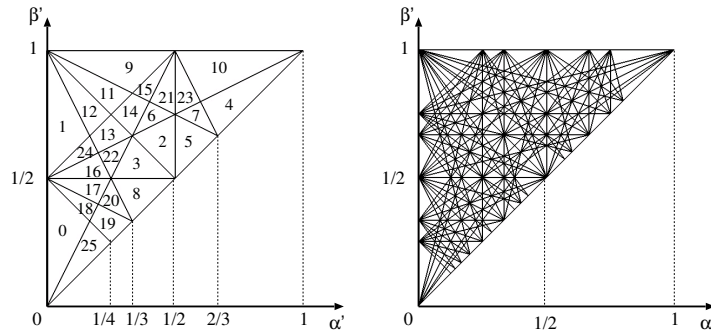


Figure 32: An experimental example of non-linear point rejection.

Figure 33: Normal cells in the parameter space (α', β') for $k = 1$ (left) and $k = 2$ (right).

then define our unified discrete Gaussian image using a discrete Gaussian sphere.

3.3.4.1 Feasible normal vectors of each LGP

From the discussion in the previous subsection, if a surface point p of a digital object has a linear LGP of $\mathbf{Q}_k(p)$, we can say that such p is locally linear with the size of $\mathbf{Q}_k(p)$. In other words, the value of k indicates the absolute size of a planar surface around p . Based on this fact, we therefore calculate normal vectors at p by using (8) from a set of points on a digital plane in $\mathbf{Q}_k(p)$.

From each linear LGP, we obtain $\mathbf{D}_{\mathbf{Q}_k(p)}(\mathbf{P})$ in $\mathbf{Q}_k(p)$, so that for all $(p, q, r) \in \mathbf{D}_{\mathbf{Q}_k(p)}(\mathbf{P})$, we obtain a set of linear inequalities (8), namely,

$$0 \leq \alpha p + \beta q + \gamma r + \delta < \omega \quad (18)$$

where $\alpha, \beta, \gamma, \delta \in \mathbb{R}$. Because we assumed the constraints (13) for generating the linear LGPs in Section 3.3, we also consider the similar constraints such that

$$0 \leq \alpha \leq \beta \leq \gamma, \gamma \neq 0. \quad (19)$$

Note that, thanks to these constraints, we have $\omega = \gamma$ in (18) from its definition. Arbitrary choosing pairs of different points $(p_i, q_i, r_i), (p_j, q_j, r_j) \in \mathbf{D}_{\mathbf{Q}_k(\mathbf{x})}(\mathbf{P})$, we thus obtain:

$$0 \leq \alpha p_i + \beta q_i + \gamma r_i + \delta < \gamma, \quad (20)$$

$$0 \leq \alpha p_j + \beta q_j + \gamma r_j + \delta < \gamma. \quad (21)$$

Because $\gamma \neq 0$ from the constraints (19), we divide both of Ineqs. (20) and (21) by γ and substitute α', β' and γ' for α, β and γ such that

$$\alpha' = \frac{\alpha}{\gamma}, \beta' = \frac{\beta}{\gamma}, \gamma' = \frac{\delta}{\gamma}. \quad (22)$$

In order to eliminate γ' by Fourier-Motzkin elimination [202], we then obtain

$$-p_i \alpha' - q_i \beta' - r_i \leq \gamma' < -p_i \alpha' - q_i \beta' - r_i + 1, \quad (23)$$

$$-p_j \alpha' - q_j \beta' - r_j \leq \gamma' < -p_j \alpha' - q_j \beta' - r_j + 1, \quad (24)$$

and finally derive

$$(p_i - p_j) \alpha' + (q_i - q_j) \beta' + r_i - r_j + 1 > 0 \quad (25)$$

for any pair $(p_i, q_i, r_i), (p_j, q_j, r_j) \in \mathbf{D}_{\mathbf{Q}_k(\mathbf{x})}(\mathbf{P})$. The solution gives a feasible region which is a convex polygon in the space (α', β') . Remark that all calculations are done by using only integers, i.e. they cause no rounding errors. Figure 33 illustrates that some of those convex polygons share the region in (α', β') and that such common regions make triangular or quadrilateral cells, called normal cells. Note that the constraints (19) make our interesting space (α', β') narrow down to a triangle bounded by linear inequalities $0 \leq \alpha' \leq \beta' \leq 1$. Table 7 shows a set of normal cells which represents a set of feasible normal vectors for each linear LGP of $\mathbf{Q}_1(\mathbf{x})$ depicted in Fig. 27.

We can find a similar figure to the left one of Fig. 33 in [195], which is generated by an approach based on 2D Farey series in number theory. It is clear that our normal cells are related to 2D Farey series since the values of our inputs $p_i - p_j, q_i - q_j$ and $r_i - r_j$ of (25) are bounded by the size of $\mathbf{Q}_k(\mathbf{x})$. However, our result is slightly different from that in [195]. It is caused by the difference between the definition of linear LGPs and that of $(2k+1, 2k+1)$ -cubes [195], as we already discussed at the end of Section 3.3.2. The difference can be seen in Fig. 33 (left), such that the normal cells “o” and “1o” are not symmetric with respect to a line of $\alpha' + \beta' = 1$, while the cells in [195] are symmetric.

Table 7: Normal cells, illustrated in Fig. 33 (left), corresponding to each linear LGP for $k = 1$, illustrated in Fig. 27, with the constraints (19).

linear LGP for $k = 1$	corresponding normal cells
1	0 25
2	1 9 11 12
3	4 5 7 10 23
4,5	0 1 16 17 18 24
6,17	2 3 4 5 7 8
7	2 3 5 8
8,9	6 9 10 11 14 15 21 23
10,12	8 19 20 25
11	8 17 18 19 20
13,28	2 3 4 5 6 7 9 10 11 12 13 14 15 21 22 23
14	2 3 6 13 14 15 16 21 22 24
15	2 3 6 11 12 13 14 22
16	4 5 7 10 23
18,19	0 18 19 25
20,23	0 1 3 8 12 13 16 17 18 19 20 22 24 25
21,22	3 8 16 17 20 22
24,25	1 9 11 12 13 14 15 24
26,34	2 4 5 6 7 10 21 23
27	2 5 6 7 21 23
29,30	0 17 18 19 20 25
31,32	1 12 13 16 22 24
33	6 9 11 14 15 21

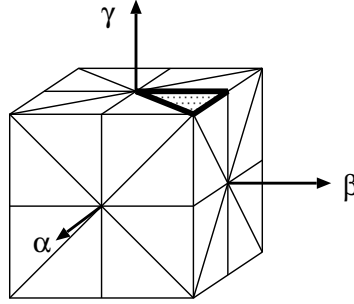


Figure 34: The cubical Gaussian sphere.

3.3.4.2 Discrete Gaussian spheres

The 26 and 910 normal cells in Fig. 33 are generated with the constraints (19). We embed these normal cells into the 3D space (α, β, γ) by using (22) with $\gamma = 1$. The triangle surrounded by thick lines in Fig. 34 (right) corresponds to the triangular region which is the union of normal cells in Fig. 33. Once the normal cells are embedded into the space (α, β, γ) , we make the congruous ones by applying to them 48 transformations of rotations and symmetries of a cube of edge length 2, centered at the origin of the 3D space. We see, in Fig. 34, that there are the 48 triangles on the cube, so that the whole cube contains 1248 and 43 680 normal cells for $k = 1, 2$, respectively. Such a cube is called a cubical Gaussian sphere.

We now project normal cells tiled on the cubical Gaussian sphere onto a unit sphere centered at the origin, as illustrated in Fig. 35.

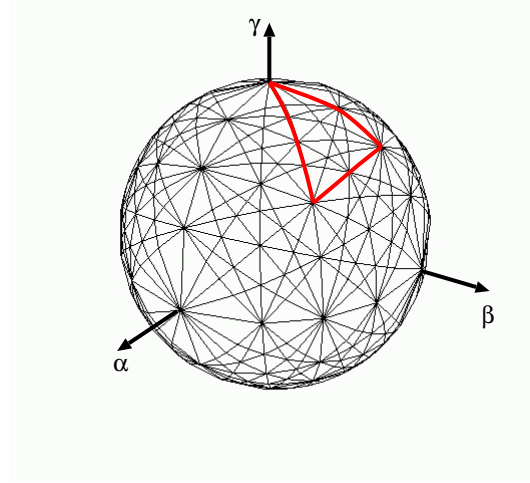


Figure 35: The discrete Gaussian sphere.

The unit sphere separated by projected normal cells is called the discrete Gaussian sphere, because the size of normal cells indicates the resolution of digitized normal vectors calculated from linear LGPs. The triangle surrounded by red lines in Fig. 35 corresponds to the triangle surrounded by thick lines in Fig. 34, which is the union of normal cells in Fig. 33. In the remainder, we denote \mathbf{G} the set of all normal cells on the discrete Gaussian sphere. Remark that we use only integer or rational numbers to calculate all normal cells, which are related to the cubical Gaussian sphere.

3.3.4.3 Unified discrete Gaussian images

By using the discrete Gaussian sphere, we give a discrete version of extended Gaussian images [75], called unified discrete Gaussian images. Let us first consider a discrete version of the Gaussian image. Let \mathbf{V} be a set of object surface grid-points. For a point $p \in \mathbf{V}$, we define a discrete Gaussian image $\mathbf{I}_k(p)$ as the set of normal cells corresponding to the linear LGP of p , if its LGP is linear with respect to k ; otherwise, $\mathbf{I}_k(p)$ is defined as empty. Because we are interested in points which have linear LGP in this section, we set $\mathbf{I}_k(p) = \emptyset$ when p has a non-linear LGP. For each cell $c \in \mathbf{G}_k$, we consider a point subset of \mathbf{V} such that

$$\mathbf{R}(c) = \{p \in \mathbf{V} : c \in \mathbf{I}_k(p)\}. \quad (26)$$

We then obtain the number of points in $\mathbf{R}(c)$ for every $c \in \mathbf{G}_k$, called a unified discrete Gaussian image, such that

$$u(c) = |\mathbf{R}(c)|. \quad (27)$$

The concept of unified discrete Gaussian images is similar to that of extended Gaussian images [75]. The differences from extended Gaussian images are the followings: the function (27) is defined with

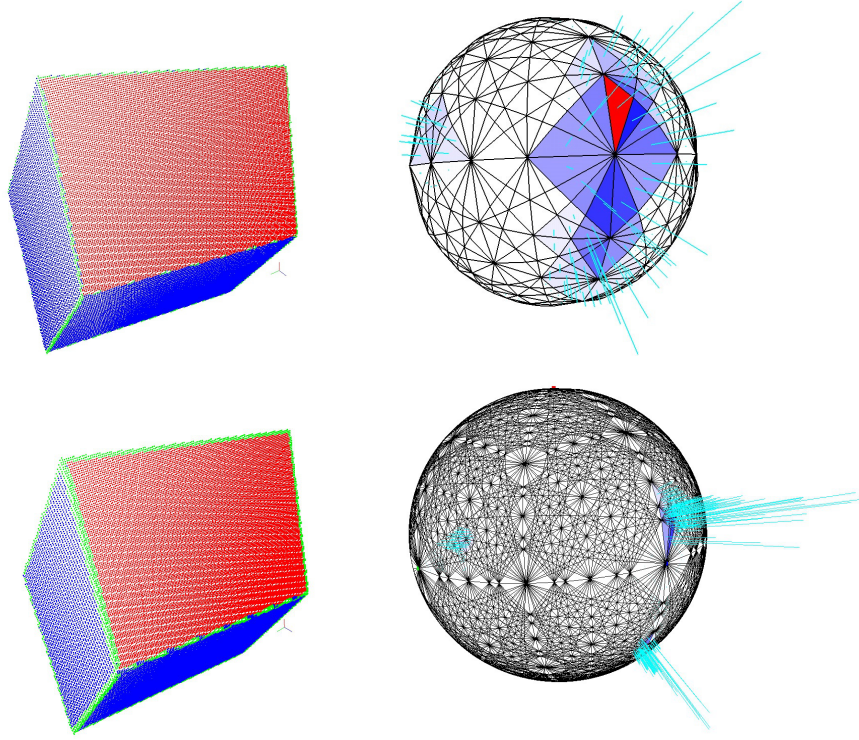


Figure 36: A synthetic 3D image of a box (left) and its unified discrete Gaussian image (right) for $k = 1$ (top) and $k = 2$ (bottom). Concerning cell colors on the discrete Gaussian sphere (right), the darker the blue cell, the larger the value of $u(c)$, and the red cell has the maximum value. The length of the pale blue needle for each cell c also corresponds to the value of $u(c)$. On a digitized box (left), red and blue points have linear LGPs for $k = 1$, and $k = 2$, while green points have non-linear LGPs. Note that red points correspond to the red cell.

respect to a normal cell c on the discrete Gaussian sphere \mathbf{G}_k , instead of a point \mathbf{n} on the Gaussian sphere; the value of (27) is the number of grid points \mathbf{p} such that $\mathbf{I}_k(\mathbf{p})$ includes c , instead of the area of the surface whose normal vector is \mathbf{n} . From the definition, we see that our unified discrete Gaussian image represents a distribution of normal cells of a digital object surface.

Figure 36 shows examples of unified discrete Gaussian images for a digitized box. In the figures, we see that we can extract a set of grid points which belong to a digital plane $\mathbf{D}(\mathbf{P})$ by choosing a “good” cell, for example, a red one. This is based on the following fact; if (α, β, γ) is a normal vector of $\mathbf{D}(\mathbf{P})$, (α, β, γ) is included in the common cell(s) of $\mathbf{I}_k(\mathbf{p})$ for all $\mathbf{p} \in \mathbf{D}(\mathbf{P})$.

3.3.5 Planar surface segmentation using LGPs

By using the unified discrete Gaussian image $u(c)$ and the point sets $\mathbf{R}(c)$, we present our algorithm for planar surface segmentation from a locally linear point set \mathbf{V}' . Our problem is formulated as follows; each point $p \in \mathbf{V}'$ is assigned to one of sets \mathbf{S}_i for $i = 1, 2, \dots$ such that the points in each \mathbf{S}_i constitutes a connected planar-surface set. From the previous discussions, our method is founded on the following hypothesis: if there is a connected point subset $\mathbf{S} \subseteq \mathbf{V}'$ such that they have a common normal cell for all $p \in \mathbf{S}$, \mathbf{S} may constitute a discrete plane.

3.3.5.1 Algorithm

Based on this hypothesis, we present Algorithm ?? . We look for the largest connected set \mathbf{S}_i , whose points have a common normal cell, by using $u(c)$ and $\mathbf{R}(c)$. As each point has several normal cells, our method cannot be processed in parallel with respect to normal cells. It must be an iterative procedure; once we obtain \mathbf{S}_i , we remove all points of \mathbf{S}_i from every $\mathbf{R}(c)$, modify $u(c)$, and repeat this procedure after the increment of i . Practically, we would like to avoid obtaining a very small surface patch, so that we set a parameter s that is the minimal size for \mathbf{S}_i .

Algorithm ?? is thus a loop procedure of seeking planar surfaces \mathbf{S}_i . Each \mathbf{S}_i is a maximally connected point set, whose points have a common normal cell. Once we find \mathbf{S}_i , we check the size of \mathbf{S}_i in Step 11, and if $|\mathbf{S}_i| \geq s$, we remove all points of \mathbf{S}_i from every $\mathbf{R}(c)$ and also modify $u(c)$ in Step 13. After such modification and incrementing i , we seek a new \mathbf{S}_i . For finding each \mathbf{S}_i , we look for the maximum connected component \mathbf{C} of each $\mathbf{R}(c)$, and then set \mathbf{S}_i to be the maximum among all \mathbf{C} . In order to reduce the frequency of calculation of connected components, which is a global operation, we make a priority queue D_k of normal cells with $u(c)$ in Step 4. We then repeat dequeue of a normal cell h from D_k to obtain the maximum connected component \mathbf{C} of $\mathbf{R}(h)$ in Step 8. Comparing the size of \mathbf{C} with the maximum among those of other normal cells that are already dequeued from D_k , we finally obtain the currently maximum point set \mathbf{S}_i in Step 9. Note that this loop is repeated until the size of $\mathbf{R}(h)$ is less than s or more than the size of \mathbf{S}_i as described in Step 7. For calculating the maximally connected component of $\mathbf{R}(h)$, we apply a simple method based on a depth-first strategy by using a queue [101]. The time complexity is linear with respect to the size of $\mathbf{R}(h)$.

3.3.5.2 Experimental results

For the experiment, we used six range images of the same blocks, which are taken by a 3D scanner Konica-Minolta VIVID 910 from two

Algorithm 3: Planar surface segmentation

input : a unified discrete Gaussian image $u(c)$, point sets $\mathbf{R}(c)$,
and a minimum surface size s

output: planar-surface point sets \mathbf{S}_i for $i = 1, 2, 3, \dots$

```

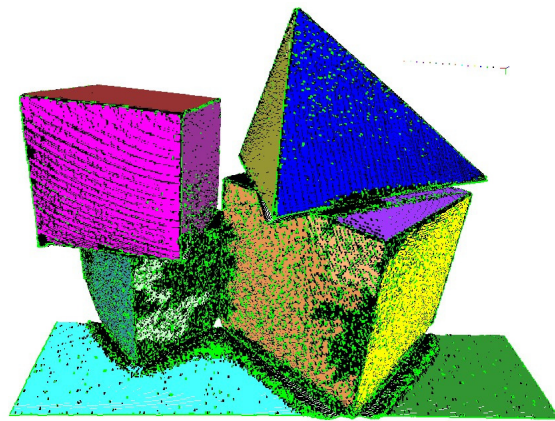
1 begin
2   initialize a label such that  $l = 0$ ;
3   repeat
4     make a queue  $D_k$  of normal cells with priorities of values
        $u(c)$ ;
5     increment  $l$  and initialize  $\mathbf{S}_l = \emptyset$ ;
6     set  $h$  to be the highest priority cell in  $D_k$  and remove it
       from  $D_k$ ;
7     while  $|\mathbf{R}(h)| > \max(s - 1, |\mathbf{S}_l|)$  do
8       set  $\mathbf{C}$  to be the maximal connected component of
          $\mathbf{R}(h)$ ;
9       if  $|\mathbf{C}| > |\mathbf{S}_l|$  then set  $\mathbf{S}_l = \mathbf{C}$ ;
10      reset  $h$  to be the highest priority normal cell in  $D_k$ 
        and remove it from  $D_k$ ;
11    if  $|\mathbf{S}_l| \geq s$  then
12      forall  $c$  such that  $u(c) \neq 0$  and  $\mathbf{R}(c) \cap \mathbf{S}_l \neq \emptyset$  do
13        reset  $\mathbf{R}(c) = \mathbf{R}(c) \setminus \mathbf{S}_l$  and  $u(c) = |\mathbf{R}(c)|$ ;
14  until  $|\mathbf{S}_l| < s$ ;
15  return  $\mathbf{S}_i$  for  $i = 1, 2, \dots, l - 1$ ;

```

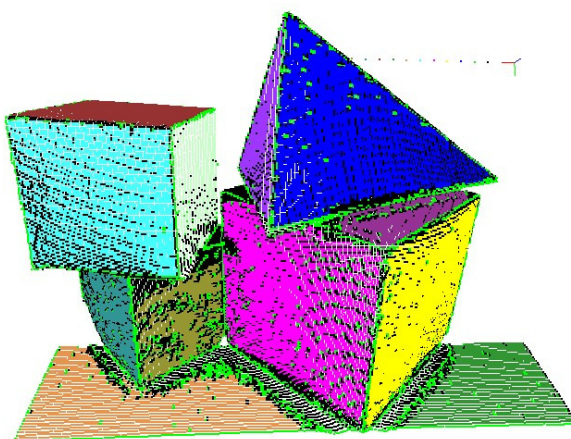
different viewpoints with three different resolutions. The range images were transformed into grid-point sets by following the explanation in Subsection 3.3.3.1. First, we rejected all non-linear points, as described in Section 3.3.3, and then applied Algorithm ???. The results are illustrated in Fig. 37. In the cases of Fig. 37, the numbers of valid (measured) points are 207 459 for (a), 51 739 for (b) and 12 859 for (c). Among those valid points, we have 184 682 locally linear points for (a), 47 093 for (b), and 11 346 for (c), respectively.

We see in Fig. 37 that non-linear points, colored in light green, appear around edges of block faces, and sometimes appear in faces because of small bumps in faces or noise in the range images. As we set the minimum surface size s , there are locally linear points that construct no planar surface whose size is not less than s around the points, colored in black in the figures. Note that we use 2D connected component labeling in Algorithm ??, instead of 3D connected component labeling, because locally linear points are sparsely distributed in the 3D space, but not in the 2D space.

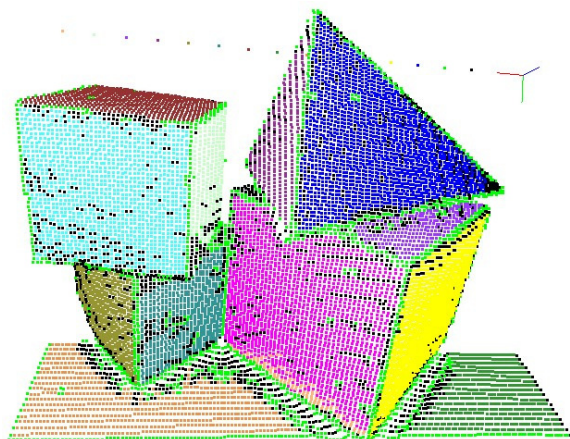
There are physically 12 visible planar surfaces in Fig. 37; there are actually 11 planes because a table face is separated into two parts with a right cube. Figure 37 shows that all planar surfaces are segmented by



(a)



(b)



(c)

Figure 37: Planar surface segmentation results from range images of blocks, which are taken from the same viewpoint, with different resolutions: the image sizes are 640×480 (a), 320×240 (b), and 160×120 (c). The minimum surfaces sizes s are set to be 1000 (a), 500 (b), and 100 (c), respectively.

our simple algorithm, which requires neither complicated parameter setting nor parameter estimation. We should mention that it may bring us rather over-segmentation results when the resolution of an input image is high. For example, the orange and cream points in Fig. 37 (a) should be considered to be in the same region, even if they are separately segmented. We also see that our method is less sensitive to image noise in lower image resolutions; for example, in Fig. 37 (a), there do not exist many linear points on the left cubic face colored in moss green, while more olive and turquoise points are found in Figs. 37 (b) and (c).

As we discussed in Section 3.3.3, we can use larger-size LGPs for the planar surface segmentation. If we use larger-size LGPs, then we will have more normal cells on the discrete Gaussian sphere [86]. This means that each normal cell becomes relatively small so that we can see smaller differences between normal vectors for their distinction. However, we may have a risk of obtaining over-segmentation results. Furthermore, as mentioned before, there are other problems such as obtaining less locally linear points because larger LGPs are more sensitive to noise, and finding a good data structure.

3.4 DIGITAL LINE AND PLANE FITTING

In the previous sections, we considered the local geometric constraints. In this section, we are interested in a global geometric constraint and consider the following fitting problem: given an arbitrary set of N points in a bounded grid in dimension d , find a digital hyperplane that contains the largest possible number of points. We focus on the cases of $d = 2, 3$ in particular. Such line fitting and plane fitting are essential tasks in the field of image analysis and computer vision. For instance, the procedures are used for shape approximation [20], [179], image registration [173], [203], and image segmentation [87], [106].

3.4.1 *Related work*

In the fields of computer vision and image processing, most commonly used methods derive from a continuous hyperplane model, which is defined as a set of Euclidean points satisfying the linear equation of Eq. (4). Such hyperplane fitting is an essential task, and the problem can be viewed as a parameter estimation method [73]. Fitting is typically carried out through optimizing various cost functions. For instance, least-square fitting minimizes the sum of geometric distances from all given points to the model, while least-absolute-value fitting uses the vertical distances [23]. However, these are not robust to the presence of outliers. Conversely, Least Median of Squares regression, which minimizes the median of the vertical/geometric distances, is robust as long as fewer than half of the given points are outliers [167].

Well-known and most used methods for approximate hyperplane fitting include the Hough Transform (HT) [53], [76], RANSAC [60] and associated variations [32]. The Hough Transform uses an accumulative approach in the discretized dual space and an ad-hoc detection of high accumulated values. The computational complexity of traditional HT is $O(N\delta^{d-1})$, where N is the number of given points and δ is the (discrete) size of image sides. This is linear for a fixed δ and a fixed d . RANSAC and its variation consider random d -tuples of points (i.e. pairs and triplets respectively for 2D or 3D) within the pixel set S , forming respectively a candidate hyperplane, and compute a distance from S to the candidate hyperplane. Many distances can be considered, including robust versions that may or may not be true distances. A distance or score is associated with every candidate hyperplane. After a number of candidate trials that depend on the size of S , the best-fitting hyperplane featuring the minimum distance or score is given as the output. The computational complexity of RANSAC is linear in the size N of S if the number of inliers is a constant fraction of N . Both HT and RANSAC are efficient, linear-time complexity algorithms. However, neither can claim to find a global, or even error-bounded approximate, solution to an associated optimization problem.

3.4.2 Problem formulation and its geometric interpretations

Because the problem is inherently discrete, it is useful to consider a purely discrete formulation of the problem using a digital hyperplane of Definition 3.1, instead of a continuous hyperplane of (4).

Given a finite set S of discrete points in \mathbb{Z}^d , the problem is to find a digital hyperplane that contains a maximum number of points, called an optimal digital hyperplane $\mathfrak{D}(\mathbf{H})$. Discrete points that are contained in the fitted digital hyperplane $\mathfrak{D}(\mathbf{H})$ are called inliers; the complement points are called outliers. Our problem is then equivalent to finding a digital hyperplane $\mathfrak{D}(\mathbf{H})$ such that the number of inliers be maximum. Figure 38 (left) depicts an example of digital line fitting. Note that we hereafter set $a_d = 1$ of $\mathfrak{D}(\mathbf{H})$ in Definition 3.1 for simplicity. In other words, we mainly deal with the following linear inequalities

$$0 \leq \sum_{i=1}^{d-1} a_i x_i + x_d + a_{d+1} < w \quad (28)$$

instead of the ones in Definition 3.1. Thus, the above optimization problem is formulated as follows:

$$\begin{aligned} & \underset{C \in 2^S}{\text{maximize}} && |C| \\ & \text{subject to} && \forall (x_1, x_2, \dots, x_d) \in C, 0 \leq \sum_{i=1}^{d-1} a_i x_i + x_d + a_{d+1} < w \end{aligned}$$

where 2^S is the power set of S . As $C = \mathfrak{D}(\mathbf{H}) \cap S$, C can be considered as the consensus set for a digital hyperplane $\mathfrak{D}(\mathbf{H})$.

In order to solve this combinatorial optimization problem, we focused on a geometric property of consensus sets for digital hyperplanes, implying that all consensus sets C can be generated from a given point set S . We introduce the following definition for geometric understanding:

Definition 3.4. *A slab is the region on and in between two parallel hyperplanes in \mathbb{R}^d . An ω -slab is a slab of size ω , meaning that the distance between the two hyperplanes is ω .*

If the distance ω is taken in the x_d -axis direction in the space (x_1, x_2, \dots, x_d) , i.e. in the primal space, and $\omega = w$, then $\mathfrak{D}(\mathbf{H})$ can be geometrically interpreted as a w -slab. Thus, finding the optimal digital hyperplane for a given S is equivalent to finding a w -slab that contains the maximum number of points in S (see Fig. 38 (left)).

Let us now consider the dual-space interpretation of a digital hyperplane in the primal space, defined by (28). A digital hyperplane is regarded as a set of parallel hyperplanes whose coefficients of x_i are a_i for $i = 1, \dots, d$ where $a_d = 1$ and whose x_d -intercepts are between $-a_{d+1}$ and $w - a_{d+1}$. This corresponds, in the dual space, to a line segment in the a_{d+1} -axis direction of length w , as illustrated in Fig. 39 (left). Because points in the primal space are represented by hyperplanes in the dual space, the problem of finding the optimal consensus set in the primal space is equivalent to searching the best position of the vertical line segment of length w such that it intersects with as many hyperplanes as possible in the dual space, as illustrated in Fig. 39 (left).

There is another primal-space interpretation of the digital hyperplane fitting problem. For each point of S , we first consider the line segment in the x_{d+1} -axis direction of length w . Given the set of the line segments of all points of S , we then look for a hyperplane that stabs the maximum number of such line segments. Figure 38 (right) illustrates the problem of stabbing line segments, which is equivalent to the original problem in Fig. 38 (left).

The dual-space interpretation of the latter primal problem is then as follows. Using standard geometric duality induced by (28), every point \mathbf{p} in the primal space is mapped to a hyperplane \mathbf{H} in the dual space. Then, a line segment, containing \mathbf{p} as an endpoint, in the x_{d+1} -axis direction of length w in the primal space is mapped to a w -slab (distance w in the a_{d+1} -axis direction) one of whose sides is equal to \mathbf{H} , in the dual space. As a hyperplane in the primal is mapped to the point in the dual, the problem of stabbing line segment in the primal (see Fig. 38 (right)) can be considered in the dual space, given a set of w -slabs for all points of S , to find the divided region that is covered by the maximum number of w -slabs (see Fig. 39 (right)).

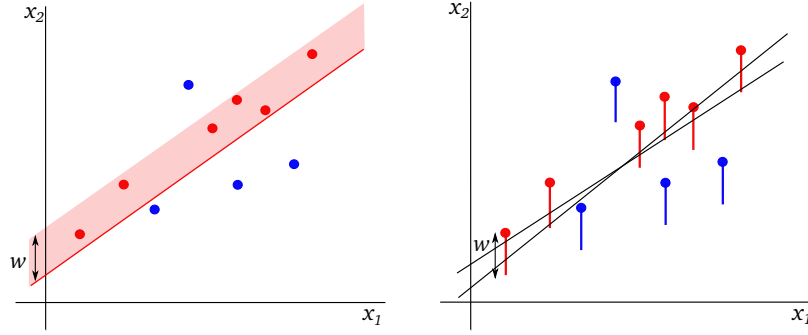


Figure 38: Digital line fitting (left) and its equivalent problem of stabbing line segments (right) in the primal space.

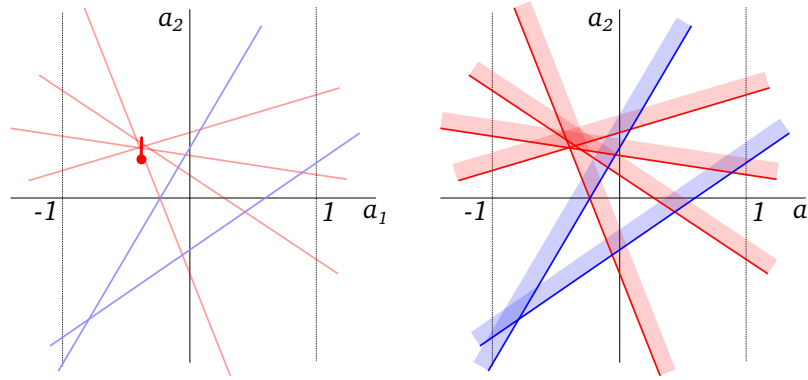


Figure 39: Digital line fitting (left) and its equivalent problem of stabbing line segment (right) in the dual space. A slab in the primal space in Fig. 38 (left) is associated with a vertical line segment in the dual space while each vertical line segment in Fig. 38 (right) is associated with a slab in the dual space.

In the following sections, the problem of digital hyperplane fitting will be considered either in the primal space or in the dual space. Working in the dual space of such a duality transform is a classical approach in computational geometry and computer vision [12], [73].

3.4.3 Exact fitting

The discrete optimization-based framework for the digital hyperplane fitting problem was initially proposed in [207] for 2D lines and 3D planes. Even though the pair of hyperplanes form a convex set, the problem is combinatorial in nature, and a non-polynomial, branch-and-bound approach was initially suggested to find the optimal solution [207]. This was later solved with an $O(N^d \log N)$ algorithm for $d = 2, 3$ in [204]–[206], and improved in [88] with an $O(N^2)$ solution in 2D using a topological sweep method. While a polynomial solution of degree equal to the dimension of the problem is useful, it is still too inefficient for many application. Typically, the problem is currently solvable for $N = 10^3$ but impractical for $N = 10^6$ in 3D.

3.4.3.1 Algorithms

As mentioned above, we have presented two kinds of exact algorithms for digital hyperplane fitting [88], [206], only one of which is applicable so far for 3D [206]. Here we explain this latter one, which searches efficiently the optimal solution with time complexity $O(N^d \log N)$ for dimension d , where $d = 2, 3$, together with space complexity $O(N)$ where N is the size of a given set S of grid points. The algorithm is inspired by the algorithm of LMS [182] working in the dual space, as explained in the previous section.

The approach is focusing on inlier sets, also called consensus sets. Since the size of S is finite and each element $x \in S$ has finite coordinates, we easily notice that the number of different consensus sets $C \in 2^S$ for the digital hyperplane fitting of S is finite as well. Thus, if we can find all different consensus sets C from a given S , which is relatively smaller than $|2^S|$, we just need to verify the size of each C and find the maximum one (ones if there are several) as the optimal solution. Then the following question comes up naturally: is it possible to find the set of all the consensus sets of S , denoted by F ? If the answer is positive, how can we generate F ? We will answer these questions as follows. Some notions related to digital hyperplanes are given: two parallel hyperplanes defined by the equations, by which the inequalities in (28) are replaced, are called the support hyperplanes of a digital hyperplane. Discrete points that are on support hyperplanes are called critical points of a digital hyperplane. The proposed method is then based on the following property of consensus sets of digital hyperplanes [206].

Property 3.5 (Zrour et al. [206]). *Given a finite set $S \subset \mathbb{Z}^d$ where $d = 2, 3$, let C be a consensus set of S for an arbitrary hyperplane. It is then always possible to find a digital hyperplane whose consensus set is the same as C such that it has at least d critical points.*

From this property, we see that we can find a digital hyperplane $\mathfrak{D}(\mathbf{H})$ for any consensus set C of S such that it has at least d critical points. This is intuitively understandable, because when we move a digital hyperplane $\mathfrak{D}(\mathbf{H})$ in the primal space, its consensus set C will change when a critical point goes out from $\mathfrak{D}(\mathbf{H})$, namely, becomes an outlier, due to the motion of the hyperplane. Indeed, such a digital hyperplane $\mathfrak{D}(\mathbf{H})$ can be constructed from a set of d points chosen from S such that they become critical points of $\mathfrak{D}(\mathbf{H})$. Consequently, we can generate the set F of all C from those $\mathfrak{D}(\mathbf{H})$ constructed from sets of d points in S . Thus, the naive method is to choose d points from S , to construct $\mathfrak{D}(\mathbf{H})$ from the d points and to verify for each point x of S if x is an inlier or not. This method leads to the time complexity $O(N^{d+1})$.

The key point of the proposed method is to reduce this complexity to $O(N^d \log N)$ by using the geometrical structure in the dual

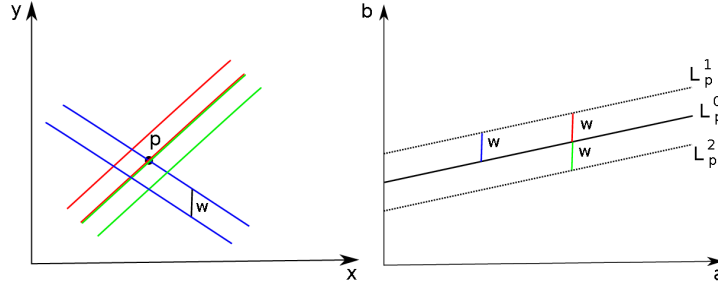


Figure 40: Digital lines on which a point p is a critical point in the primal space (left), and those corresponding vertical line segments of length w in the dual space (right).

space, which allows to make the verification incrementally. Let $x = (x_1, x_2, \dots, x_d)$ be a point in the primal space, then the dual of x is the hyperplane of (4) in the dual space $(a_1, a_2, \dots, a_{d-1}, a_{d+1})$. Here we suppose that $a_d = 1$ due to the normalization explained in Section 3.4.2. Likewise, the dual of a nonvertical hyperplane $a \cdot x = 0$ in the primal space where $a = (a_1, a_2, \dots, a_{d-1}, a_{d+1})$ is the point a in the dual space. Now, let us consider the dual-space interpretation of a digital hyperplane in the primal space: a digital hyperplane corresponds, in the dual space, to a vertical line segment of length w , as illustrated in Fig. 39 (left). Because points in S in the primal space are represented by hyperplanes in the dual space, the problem of finding the optimal consensus set in the primal space is equivalent to searching the best position of the vertical line segment of length w such that it intersects with as many hyperplanes as possible in the dual space, as described in Section 3.4.2.

Obviously, we cannot search everywhere in the dual space to find the best line segment. From Property 3.5, we know that, for any consensus set, there exists a digital hyperplane that features at least d critical points. Therefore, we first take $d - 1$ points and consider all the digital hyperplanes on which all the $d - 1$ points are critical points on the same support hyperplane.

For simplification, let us explain the 2D case first. We take one point $p \in S$ and consider it to be the first critical point of such a fitted digital line. Because p corresponds to a line L_p^0 in the dual space, all digital lines for which p is a critical point correspond to the set of all the vertical line segments of length w having one of its endpoints on L_p^0 in the dual space, as shown in Fig. 40. The set of such digital hyperplanes, therefore, forms two w -slabs in the dual space; one of them is bounded by L_p^0 and $L_p^1 = L_p^0 + we_d$ where $e_d = (0, 0, \dots, 0, 1)$, and another is bounded by L_p^0 and $L_p^2 = L_p^0 - we_d$, as illustrated in Fig. 40. For simplification, we focus on the w -slab bounded by L_p^0 and $L_p^1 = L_p^0 + we_d$, because the following discussion is also valid for another slab bounded by L_p^0 and $L_p^0 - we_d$.

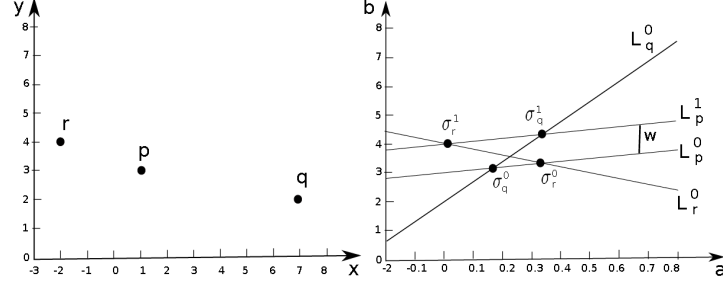


Figure 41: Three points p, q, r in the primal space (left), and the corresponding hyperplanes L_p^i for $i = 0, 1$, L_q^0 and L_r^0 in the dual space, with their intersections.

According to Property 3.5, we choose a point $q \in S \setminus \{p\}$ to be the second critical point of a fitted digital hyperplane such that the x_1 -coordinate of q is different from that of p , i.e., $x_q \neq x_p$; the case of $x_q = x_p$ will be discussed later. Any point q in the primal space is represented by the line L_q^0 in the dual space, as shown in Fig. 41. We see in this figure that L_q^0 intersects each of the slab boundaries, L_p^0 and $L_p^1 = L_p^0 + we_d$, if it is not parallel to L_p^0 ; the parallel case occurs when $x_q = x_p$, and it will be dealt with separately as a degenerate case without exceeding the computational complexity (see [206] for details). The intersections between L_q^0 and L_p^i , σ_q^i , for $i = 0, 1$, are calculated. Geometrically, the vertical line segment in the w -slab, one of whose endpoints is one of the intersections σ_q^i , in the dual space corresponds to a digital line with the two critical points p and q in the primal space. This shows that the digital lines corresponding to the vertical line segments between the two intersections σ_q^0 and σ_q^1 in the slab always contain q as an inlier.

In order to know the number of inliers within the digital lines with a critical point p , we check the intersections σ_q^0 and σ_q^1 of L_q^0 for all $q \in S \setminus \{p\}$ with the w -slab boundaries, L_p^0 and L_p^1 . We increment a counter when L_q^0 enters the slab and reduces the counter by -1 when L_q^0 goes out from the slab. Algorithmically, once all the intersections σ_q^i for $i = 0, 1$ are calculated for all $q \in S \setminus \{p\}$, they are sorted with respect to the a_1 values. With this sorting order, the counter is incrementally calculated so that the maximum value is found, together with the optimal consensus set. The degenerate cases such that many lines enter or leave a slab at the same a_1 value is described in [206].

For digital plane fitting in the three-dimensional space, we need a search procedure for an optimal segment similarly to the above algorithm for digital line fitting. Thanks to Property 3.5, we know that, for any consensus set, there exists a digital plane featuring at least three critical points, among which at least two are on one of the support planes. Thus, taking two different points p, q from S in the primal space, we first consider all the digital planes on which both p

fitting problem is as hard to obtain as that of $O(N^d)$ problems; for $d = 2$, we call such a class of problems the 3SUM problem. For the sake of simplicity, we start our discussion in the 2D plane.

3SUM is the following computational problem, introduced by Gajentaan and Overmars [68] and conjectured to require roughly quadratic time complexity: given a set T of n integers, are there elements a, b, c in T such that $a + b + c = 0$? A problem is called 3SUM-hard if solving it in subquadratic time implies a subquadratic time algorithm for 3SUM.

Observation 3.1 (Aiger *et al.* [2]). *The problem of digital line fitting is 3SUM-hard.*

The extension of 3SUM problem to higher dimensions is considered as the problem of detecting affine degeneracy of a given collection of N hyperplanes, i.e. finding a subset of $d + 1$ hyperplanes intersecting in a common point. This is conjectured to require $O(N^d)$ time. Here, we do not seek a proof since the case for $d = 2$ is sufficient for our purpose, as we seek a linear algorithm.

3.4.5 Approximate fitting

As solving the problem exactly in arbitrary dimension is likely at least quadratic, we next suggest two approximations. The first has a theoretically proven characteristics but is not easy to implement. The second is a more practical algorithm that features a proven worst case runtime and can be easily implemented. Moreover, its practical runtime can be much better than what is suggested by its worst case analysis.

3.4.5.1 Approximation with bounded error in number of inlier points

Here, we show that if the optimal number of inlier points is not too small (i.e. $\Omega(N)$), an approximation of the optimal digital hyperplane can be found in linear time, with respect to N and the runtime also depends on the given approximation value ε . We use the dual space and make a simple use of the tool to solve approximately the problem of linear programming with violations, presented in [10]. In this approximation, we do not use the fact that the points lie on a grid and it is correct for any set of points in \mathbb{R}^d .

We start with the results of Aronov *et al.* [10] on linear programming with violations. They obtained a randomized algorithm which is correct with high probability. Afshani *et al.* also [1] obtained a Las Vegas algorithm (i.e. one that either provides the correct answer or informs about failure).

Theorem 3.1 (Aronov *et al.* [10]). *Let L be a linear program with n constraints in \mathbb{R}^d , and let f be the objective function to be minimized. Let k_{opt} be the minimum number of constraints that must be violated to make*

L feasible, and let v be the point minimizing $f(v)$ with k_{opt} constraints violated. Then one can output a point $u \in \mathbb{R}^d$ such that u violates at most $(1 + \varepsilon)k_{opt}$ constraints of L , and $f(u) \leq f(v)$. The results returned are correct with high probability. The expected running time (which also holds with high probability) of this algorithm is $O(n + \varepsilon^{-4} \log n)$ for $d = 2$, and $O(n(\varepsilon^{-2} \log n)^{d+1})$ for larger d .

In our case, we are only interested in finding a point in the dual space that is covered by the maximum number of w -slabs. We reduce this problem to the problem of linear programming with violations and solve it using the result of [10]. The following observation is a result of replacing each w -slab with two halfspaces that have the w -slab as their intersection, represented by (28). We thus have $2N$ constraints and a point in space is violating (i.e. not covered by) k w -slabs among N , if and only if it is violating k halfspaces among the $2N$. Therefore, the tool for finding a point violating the minimum number of halfspaces finds also a point that is covered by the maximum number of w -slabs. Let n_{opt} be the maximum possible number of w -slabs that can contain a point in d -space (i.e. inliers) and let k_{opt} an optimal number of violations, $N = k_{opt} + n_{opt}$. For the approximation parameter, we observe that since we find a point that violates at most $(1 + \varepsilon)k_{opt}$ slabs, we actually find a point that is covered by at least $N - (1 + \varepsilon)k_{opt}$ w -slabs. Let n be the number of points that are covered by the w -slab we just found. If $n_{opt} = \Omega(N)$ we have $n > (1 - c\varepsilon)n_{opt}$ for some fixed c . We now observe the following:

Observation 3.2 (Aiger et al. [2]). *Given a set of N points in the d -dimensional space and some $\varepsilon > 0$, we can find a w -slab that contains at least $(1 - \varepsilon)n_{opt}$ points, where n_{opt} is the maximum possible number of points that can be found in a w -slab, assuming $n_{opt} = \Omega(N)$. The runtime is $O(N + \varepsilon^{-4} \log N)$ for $d = 2$ and $O(N(\varepsilon^{-2} \log N)^{d+1})$ for larger d .*

This result can be immediately used for our original problem of finding the optimal digital hyperplane by using the set of grid points.

3.4.5.2 Approximation with bounded error in digital hyperplane width

Here, we show another kind of approximation that makes use of the fact that the input points are in a bounded grid as well. The meaning of this approximation is slightly different from the previous one. The advantage of this version is that it is easy to implement, unlike the previous one that is mainly of theoretical interest and is probably hard to implement. Moreover, using this approximation, we can do both: prove its worst case runtime and also allow to get optimal solution with practical good runtime or to have an approximation to any desired level in better runtime than the worst case. We need the following result from da Fonseca and Mount [63].

Theorem 3.2 (da Fonseca and Mount [63]). *For a set of N points in the unit d -dimensional cube and some $\varepsilon > 0$, one can build a data structure with $O(\varepsilon^{-d})$ storage space, in $O(N + \varepsilon^{-d} \log^{O(1)}(\varepsilon^{-1}))$ time, such that for a given query hyperplane \mathbf{H} , the number of points on and below \mathbf{H} can be approximately reported in $O(1)$ time, in the following sense: all the points (below \mathbf{H}) that have a larger distance² than ε from \mathbf{H} are counted. Points that are closer to \mathbf{H} on both sides may or may not be reported.*

Then, the main theorem of this section is stated as follows.

Theorem 3.3 (Aiger et al. [2]). *Given a set of N points on a grid $[0, \delta]^d$, and some $\varepsilon > 0$, $w > 0$, a digital hyperplane of width $w + 5\varepsilon$ that contains $n \geq n_{\text{opt}}$ points, can be found in $O(N + (\frac{\delta}{\varepsilon})^d \log^{O(1)}(\frac{\delta}{\varepsilon}))$ time where n_{opt} is the maximum number of points that any digital hyperplane of width w in $[0, \delta]^d$ can contain.*

For the proof, please see [2]. A typical use for digital hyperplanes would be to use $w = 1$ and $0 < \varepsilon < 0.5$.

The detail of building a specially-designed data structure for the query is described as follows. The algorithm is very simple to implement using recursion. For every box in space we first build the range counting data structure recursively. We first find the points that lie inside each one of the 8 children of the box (this is performed in logarithmic time using orthogonal range searching) and then we create the range counting data structure for each one of them recursively using the box of size ε as the smallest box. Then for building the data structure for the current box we create the set of all possible digital planes (depending on ε) and query each one of them in all children, summing the number of points.

We now go over all planes in the outer box (the bounding box of the set of input points) and for each plane, we query the number of points below this plane and below a parallel plane of distance $w + 5\varepsilon$ as described in the previous section. We simply take the pair (which is a slab) in which the number of points resulting from the subtraction of the two is the largest. A practical problem however is the memory requirement, since the algorithm has space (and roughly time) complexity $O((\delta/\varepsilon)^3)$, so in 3D it may require large amounts of memory depending on the approximation parameter.

3.4.6 Experiments

In this section, we experimentally compare the exact algorithm [206] with time complexity $O(N^3 \log N)$, explained in Section 3.4.3.1, and the approximated method with bounded error in digital plane width, explained in Section 3.4.5.2. We used two types of 3D digital images

² The ℓ_2 distance is used in [63] while the ℓ_1 distance is used in this section. However, the theorem still holds since the ℓ_1 distance is always greater than or equal to the ℓ_2 distance.

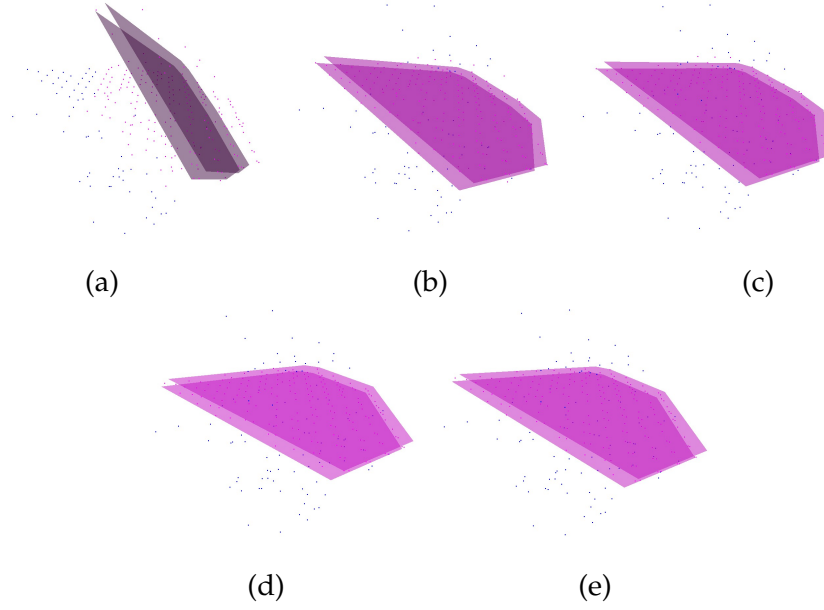


Figure 44: Experimental results of digital plane fitting for a 3D synthetic volume data generated by 400 points in a digital plane and 100 randomly generated points (outliers). The approximation algorithm with bounded error in digital plane width is applied with the value of ε set to be 2.0 (a), 1.0 (b), 0.5 (c), and 0.25 (d): rose points are inliers and blue points are outliers. A fitted digital plane is visualized as a pair of rose parallel planes. The optimal solution (e) obtained by the exact algorithm [206] is also illustrated.

for the experiments: some synthetic test data and three real data: a 3D discrete point cloud, an X-ray micro-tomography image, and an electron nano-tomography image.

3.4.6.1 Noisy digital image of digital planes

The 3D synthetic data was created such that 400 grid points were samples from a digital plane formulation of Definition 3.1 with setting $w = 1$, and 100 grid points were added randomly. For the approximated method, four different ε values were used: 2.0, 1.0, 0.5 and 0.25.

As seen in Table 8 and Fig. 44, the computation time of the exact algorithm is long, however it does yield the optimal solution containing 406 inlier points (i.e. the 400 expected points plus 6 random ones). On the other hand, the approximation results indicate that the smaller the value of ε , the more precise the solution; when $\varepsilon = 2$, a solution relatively far from the optimal was obtained. Table 8 also shows that two different numbers of points were obtained for each value of ε : the first one is the number of points in a fitted digital plane with width w , and the second one is the number of points in a fitted digital plane with width $w + 5\varepsilon$. Note that the second number is guaranteed to be greater than or equal to the optimal number of inlier points by design,

Table 8: The runtimes, parameters and numbers of points of fitted digital planes in Fig. 44.

	runtime	parameters			nb. of points	
		a_1	a_2	a_4	w	$w + 5\epsilon$
Approx. fitting						
$\epsilon = 2.0$	31 msec	-0.0625	-0.0625	-10.9	110	485
$\epsilon = 1.0$	266 msec	-0.5	-0.5	-3.0002	300	435
$\epsilon = 0.5$	2172 msec	-0.5	-0.5	-3.0002	300	421
$\epsilon = 0.25$	16 640 msec	-0.5625	-0.546875	-2.55002	362	410
Exact fitting						
exact comp.	35 min 29.109 sec	-47/81	-43/81	-203081/81000	406	
float comp.	4 min 36.908 sec	-0.580247	-0.530864	-2.507173		

and it indeed converges to the optimum as ϵ is decreasing (see Table 8). We can observe as well from Table 8 that the smaller the value of ϵ , the higher the runtime. Therefore, it is necessary in practice to find an appropriate value for ϵ , which provides a sufficiently approximate solution within a reasonable timeframe.

3.4.6.2 Real data: 3D discrete point cloud

The first real data example is a set of 3D discrete point clouds in Fig. 37 (c), which is obtained after the planar surface segmentation of a range image of blocks (see Section 3.3.5.2). The number of points in the cloud is 12 859, which are segmented into 13 planar surfaces, as illustrated in Fig. 37 (c) with points in different colors, except for those colored in light green that are detected as edge points. For each of these 13 sets, we optimally fitted a digital plane. We see the corresponding planes in Fig. 45, and the number of points for each segmented surface and the size of its optimal consensus set in Table 9.

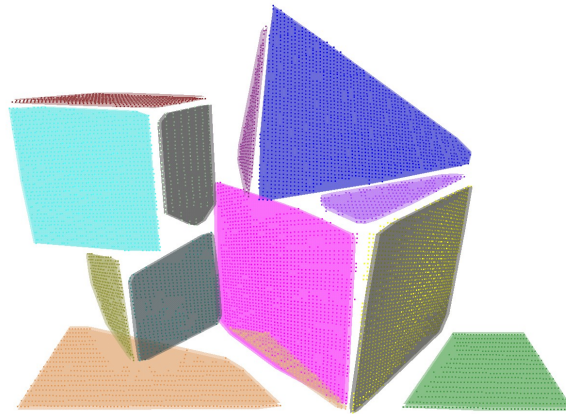


Figure 45: Results of exact digital plane fitting of segmented planar surfaces in Fig. 37.

Table 9: The number of points for each segmented planar surface in Fig. 37 and the size of its optimal consensus set for digital plane fitting.

	Nb. of points	Opt. consensus set size
Blue	1770	1401
Yellow	1578	1195
Pink	1523	935
Pale blue	1191	922
Orange	699	693
Green	573	573
Brown	545	544
Turquoise	536	512
Olive	440	405
Purple	248	245
Violet	232	206
Moss green	223	223
Cream	101	97

3.4.6.3 Real data: X-ray micro-tomography image

We also applied the exact algorithm to a 3D image extracted from a polymer foam observed in X-ray micro-tomography, on which homotopic thinning and surface decomposition were applied [56]. Figure 46 shows a cross section of the original image and Fig. 47 shows a 3D binary image obtained after homotopic thinning and surface decomposition; the image is cut into two parts for visualization. Among around 400 sets of points forming surfaces in the entire image, we choose a part, as illustrated in Fig. 48, including 17 decomposed surfaces for digital plane fitting. We show the fitted planes in Fig. 49, and the number of points and the optimal consensus set size for each segmented surface in Table II. For both the examples, we set $w = 1$.

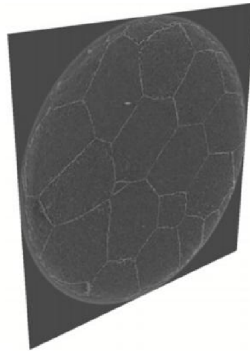


Figure 46: A cross section of a 3D image extracted from a polymer foam observed in X-ray micro-tomography.



Figure 47: The 3D binary image obtained after homotopic thinning and surface decomposition applied on the image in Fig. 46: the image is cut into two part for visualization.

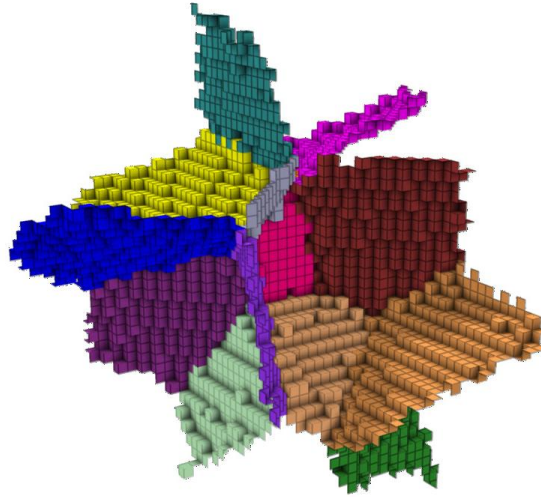


Figure 48: Selected decomposed surfaces, which is a part of the 3D binary image in Fig. 47, for digital plane fitting.

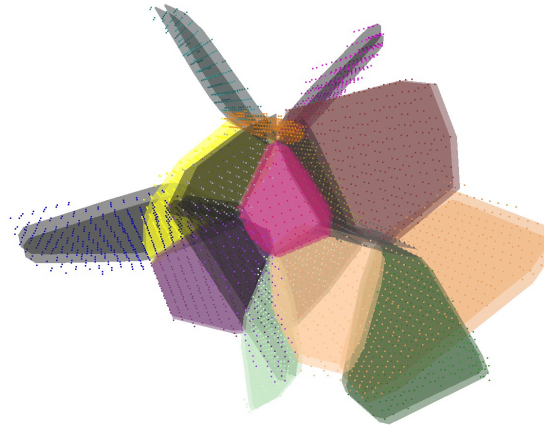


Figure 49: Optimal fitted digital planes for decomposed surfaces shown in Fig. 48.

3.4.6.4 Real data: electron nano-tomography image

The third real data we used was a 3D binary image generated from a electron nano-tomography image containing a cubical crystal. The

Table 10: The number of points and the optimal consensus set size for each decomposed surface in Fig. 49.

Nb. of points	Opt. consensus set size
541	269
512	233
439	208
427	196
427	200
405	208
377	159
335	206
333	169
309	141
308	168
258	76
220	104
200	90
198	61
163	98
104	71

image is very noisy due to the dimension of the sample and the physically-constrained tomography reconstruction method. The original image is of $512 \times 511 \times 412$ with gray values. After binarizing the image by a thresholding, we detected the boundary points by using the 6-neighborhood and extracted the largest connected component by using the 26-connectivity. Finally, we obtained 205 001 points in the $512 \times 511 \times 412$ grid. This number of points is too large to apply the exact algorithm: it would have required on the order of 10^{18} operations, i.e. many months of runtime. Instead we ran the approximate algorithm with some adjustments of the values for ε and w . We set $\varepsilon = 4$ to obtain its runtime around 12 seconds with $w = 1$; the result is illustrated in Fig. 50 (a). As we saw that the cube wall is very noisy, we also performed a fitting with $w = 25$ to obtain a thicker digital plane (see Fig. 50 (b)).

3.5 SUMMARY AND PERSPECTIVES

In this chapter, we assumed that what we observe from data has linear shapes, namely, form lines in 2D and planes in 3D. In this sense, the common geometrical model, digital hyperplane, formulated in Definition 3.1, has been used through the whole chapter. The two different types of digital geometric problems were considered: digital planar surface segmentation, which assumes rather locally the linear shape constraint, and digital plane fitting, which is a global optimization problem under the linear shape constraint.

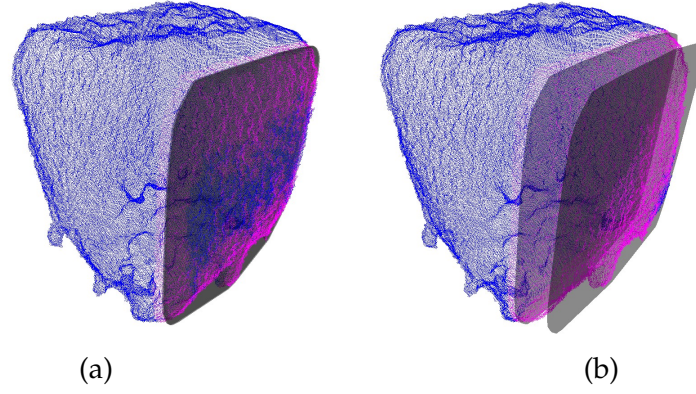


Figure 50: Results of digital plane fitting for a pre-processed 3D binary nanotomography image containing 205 001 points. The approximation algorithm with bounded error in digital plane width is applied with $\varepsilon = 4$ for $w = 1$ (a) and $w = 25$ (b).

In order to solve the first problem of digital planar segmentation, we present a discrete version of the hybrid method. Our method simply requires two types of look-up tables, i.e., the binary LGP table (linear or non-linear) and the normal cell list with respect to each linear LGP, and does not require any parameter setting/estimation. The experimental results in Fig. 37 show that our method is useful for planar surface segmentation from a point cloud. This comes from the fact that our method takes into account not only quantization errors but also noise. The proposed method is practical to filter out noise and to classify points with respect to similar directions of the normal vectors based on the unified discrete Gaussian image. From this reason, the results of this first part were also used as inputs of the second problem of this chapter.

In the second part of this chapter, we have exposed a new method for line and plane fitting on discrete data such as bitmap images using a digital geometry approach. The digital geometry approach allows practitioners to separate effects due to digitization on the one hand and noise on the other. Using our approach, we first have proposed an optimal fitting method from the point of view of the maximal consensus set: we are guaranteed to fit a digital line or plane with the least amount of outliers. The 3D algorithm is based on the same idea as the 2D algorithm. The 2D and 3D algorithms has a complexity that are identical to parameter-less traditional plane-fitting algorithms such as LMS regression [167], but allows us to define a digital line or plane exactly, in the presence of outliers. Nevertheless, solving the problem exactly in dimension d is likely at least quadratic, we have proposed two approximate discrete hyperplane fitting methods with outliers. The first uses an approach based on linear programming with violations. It is continuous in nature and features interesting complexities but is difficult to implement. The second is discrete in

nature, it uses an accumulation and query data structure and is easy to implement. This method features bounded error defined in this way: for a given N points, a width w and an error factor ε , the hyperplane found contains in a width equal to $w + 5\varepsilon$ at least as many points as the optimum would with a width of w . It features a computational complexity in $O(N + (\frac{\delta}{\varepsilon})^d)$, where δ is the distance between two neighbours in the hypergrid. The algorithm is therefore linear in the number of points in the set being considered, but exponential in the approximation factor with d , the geometric dimension. Memory requirements are also exponential with ε and d in the same way.

Nonetheless, as we show in this chapter that the exact solution is 3SUM-hard, this method is useful. The computational complexity is given in the worst case, in practice it can be much better. The algorithm is in particular well-behaved when there are few outliers.

Future work will include more complete applications such as optimal polygonization or polyhedrization by choosing a good value for w automatically, and image registration considering all feasible digital line and plane parameters.

DIGITAL SHAPE MANIPULATION: RIGID MOTIONS ON \mathbb{Z}^2 AND \mathbb{Z}^3

Geometric transformations are often involved in image registration, object tracking, and shape matching and recognition in two- and three-dimensional digital images [11], [199], [203]. A geometric transformation is a bijection of a set having some geometric structure to itself. They are classified according to the geometric properties they preserve: for example,

- displacements preserve angles, distances and orientations;
- isometries preserve angles and distances;
- similarities preserve angles and ratios between distances;
- affine transformations preserve parallelism;
- projective transformations preserve collinearity.

Among these geometric transformations, displacements obtained by composing translations and rotations, are simple and fundamental transformations in digital image processing. In order to apply such a displacement, namely a rigid motion, to digital images, it must be digitized. However, while the geometry and topology are preserved by any rigid motion in \mathbb{R}^n , in the discrete domain, i.e. \mathbb{Z}^n but not \mathbb{R}^n , these geometric and topological invariances are generally lost because of the discontinuities induced by the digitization processes of the function (see Figure 51). Indeed, these defects could be caused by the loss of the bijectivity of the digitized function (see Figure 52).

This problem is of crucial importance, particularly when images contain rich contents such as a large number of objects structured in a complex way, for example in biomedical and geomaterial applications. At present, this issue is addressed through ad hoc strategies in most cases. Thus, rigid motions are generally applied in \mathbb{R}^n , after the digital images have been plunged from \mathbb{Z}^n into this continuous space. The results are then re-digitized in \mathbb{Z}^n . This succession of spatial transpositions without topological and geometric consideration induces numerical results which necessarily lead to topological and geometric alterations.

In order to avoid losing such control of geometry and topology on \mathbb{Z}^n , we aim to develop a purely discrete framework for rigid motions on \mathbb{Z}^n based on digital geometry and topology concepts. This new framework will allow to manipulate digital objects in a “rigid” manner

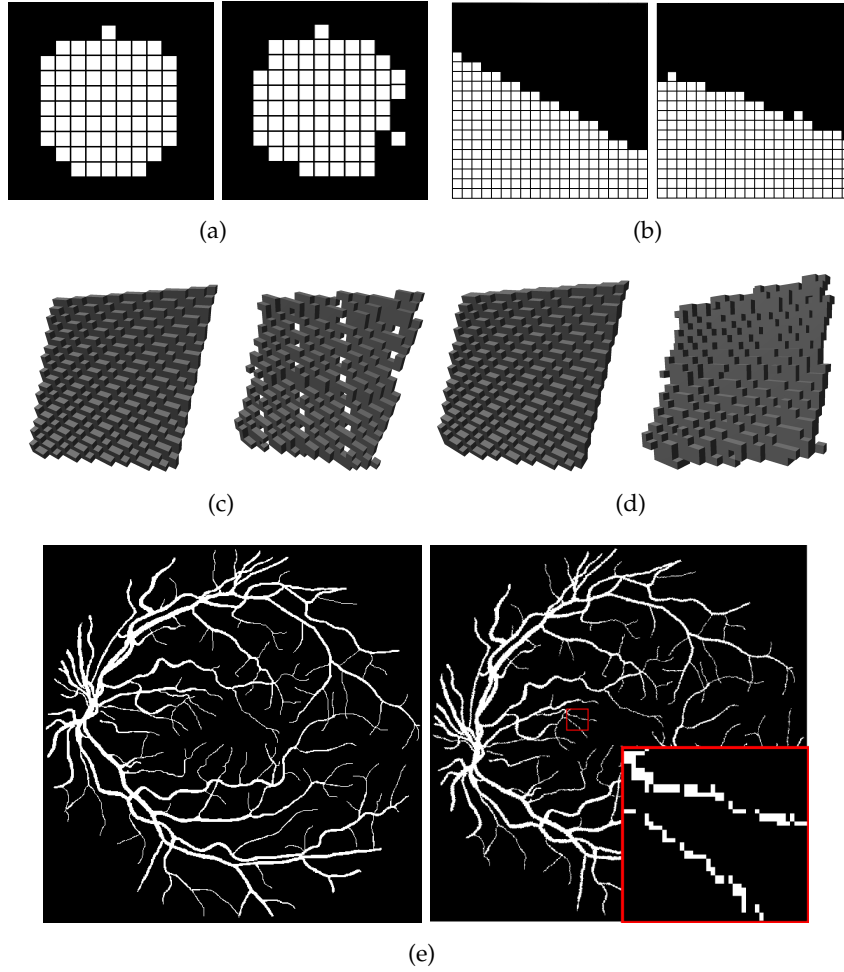


Figure 51: Digital images (left) and their rigid motions (right). (a) A disk preserves neither topology nor geometry, while (b) a half-plane preserves its topology but not its geometry. (c) A thin digital plane preserves neither topology nor geometry, while (d) a thicker digital plane preserves topology (connected without hole) but is not planar shape. (e) A binary retinal image here preserves neither topology nor geometry.

via “pixel-by-pixel” operations. In this chapter, we consider the discrete model of the exact digitized rigid motions and seek solutions to the following sub-problems of digitized rigid motions: combinatorial analysis of local behaviors [39], [149] (see Section 4.2), characterisation of bijectivity [150], [152], [153] (see Section 4.3), preservation of topology [138], [148] and geometry [131] (see Sections 4.4 and 4.5).

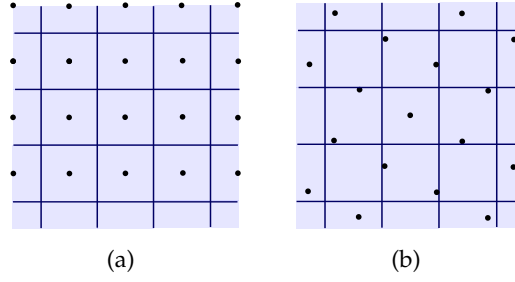


Figure 52: After a rigid motion of points of \mathbb{Z}^2 in the square tiling that are initially positioned as illustrated in (a), there exist square cells containing no preimage or rather two preimages, as illustrated in (b); the transformation is no longer bijective.

4.1 RIGID MOTION AND DIGITIZATION

4.1.1 Rigid motions on \mathbb{R}^n

Let us consider an object X in the Euclidean space \mathbb{R}^n , $n = 2, 3$, as a bounded and closed connected subset of \mathbb{R}^n . Rigid motions on \mathbb{R}^n are defined by a mapping:

$$\left| \begin{array}{lcl} \mathfrak{T} & : & \mathbb{R}^n \rightarrow \mathbb{R}^n \\ & & x \mapsto Rx + t \end{array} \right. \quad (29)$$

where R is a rotation matrix and $t \in \mathbb{R}^n$ is a translation vector. Such bijective transformation \mathfrak{T} is isometric and orientation-preserving. In other words, $\mathfrak{T}(X)$ has the same shape as X (i.e. preserves the geometry and topology).

4.1.2 Rotation representations and parameterizations

Let us here focus on rotational parts. Rotation matrices in \mathbb{R}^n are $n \times n$ square matrices, with real entries. More specifically, they can be characterized as orthogonal matrices with determinant 1.

In \mathbb{R}^2 , any rotation matrix has the following form:

$$R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad (30)$$

which rotates points on the plane counterclockwise by an angle θ about the origin of the Cartesian coordinate system. Indeed, there is one parameter θ , which describes the magnitude of the rotation about the origin, in \mathbb{R}^2 .

In \mathbb{R}^3 , the number of rotational degrees of freedom is three¹, and there are several representations parameterizing a rotation. The

¹ The number of rotational degrees of freedom in n dimensions is $\frac{n(n-1)}{2}$, which comes from the dimensions of the rotation group.

axis-angle representation parameterizes a rotation by a unit vector w indicating the direction of an axis of rotation and an angle θ . Then Rodrigues' rotation formula gives a method to obtain R from w and θ [128]:

$$R = I + (\sin \theta)W + (1 - \cos \theta)W^2$$

where I is 3×3 identity matrix and W denotes the cross-product matrix for the unit vector $w = (w_x, w_y, w_z)$,

$$W := \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}$$

with $w_x^2 + w_y^2 + w_z^2 = 1$.

We can also parametrize a rotation in \mathbb{R}^3 by three reals $a, b, c \in \mathbb{R}$, which constitute the skew-symmetric matrix A :

$$A := \begin{bmatrix} 0 & c & -b \\ -c & 0 & a \\ b & -a & 0 \end{bmatrix}. \quad (31)$$

Then Cayley transform [30] allows us to obtain almost any rotation matrix R such that

$$\begin{aligned} R &= (I - A)(I + A)^{-1} \quad (32) \\ &= \frac{1}{1 + a^2 + b^2 + c^2} \begin{bmatrix} 1 + a^2 - b^2 - c^2 & 2(ab - c) & 2(b + ac) \\ 2(ab + c) & 1 - a^2 + b^2 - c^2 & 2(bc - a) \\ 2(ac - b) & 2(a + bc) & 1 - a^2 - b^2 + c^2 \end{bmatrix}. \end{aligned} \quad (33)$$

Indeed, rotations by π around any axis can only be obtained by the Cayley transform as a limit: angles of rotation converge to π when a, b, c tend to infinity [174]. In practice, this constraint is negligible and does not affect generality of our study. Note that rotations in \mathbb{R}^2 can be also written by Eq. (32), setting

$$A := \begin{bmatrix} 0 & -a \\ a & 0 \end{bmatrix}. \quad (34)$$

In other words, 2D rotations are parametrized by $a \in \mathbb{R}$, instead of $\theta \in (-\pi, \pi)$, and we have

$$\cos \theta = \frac{1 - a^2}{1 + a^2}, \quad \sin \theta = \frac{2a}{1 + a^2}.$$

Rather than rotation matrices, we can use quaternion operations for rotations in \mathbb{R}^3 . Let us represent any point in \mathbb{R}^3 by a pure imaginary quaternion: $x = (x_1, x_2, x_3) \simeq x_1i + x_2j + x_3k$. Then, any rotation can be written as

$$x \mapsto qxq^{-1}, \quad (35)$$

where $x \in \mathbb{R}^3$ [84], [194]. The quaternion q is uniquely determined up to multiplication by a nonzero real number, and if $|q| = 1$, up to a sign change: $qxq^{-1} = (-q)x(-q)^{-1}$; hence the correspondence between unit quaternions and rotation matrices is two-to-one. Note that, for any unit norm quaternion $q = a + bi + cj + dk$, a rotation angle θ and an axis of rotation w are given as $\theta = 2 \cos^{-1} a$, and $w = \frac{(b,c,d)}{|(b,c,d)|}$, respectively. We refer the reader unfamiliar with quaternions to [40], [84], [194].

We use one of the above rotation representations properly according to each of the following problems.

4.1.3 Rigid motions on \mathbb{Z}^n

If we simply apply a rigid motion \mathfrak{T} of (29) to every point in \mathbb{Z}^n , we generally have $\mathfrak{T}(\mathbb{Z}^n) \not\subset \mathbb{Z}^n$. In order to get the points back on \mathbb{Z}^n , we then need a digitization operator

$$\left| \begin{array}{ll} \mathfrak{D} : \mathbb{R}^n & \rightarrow \mathbb{Z}^n \\ (x_1, \dots, x_n) & \mapsto (\lfloor x_1 + \frac{1}{2} \rfloor, \dots, \lfloor x_n + \frac{1}{2} \rfloor) \end{array} \right.$$

where $\lfloor s \rfloor$ denotes the largest integer not greater than s . A discrete analogue of \mathfrak{T} is then obtained by

$$\mathcal{T} := \mathfrak{D} \circ \mathfrak{T}|_{\mathbb{Z}^n}, \quad (36)$$

so that the discrete analogue of the point-wise rigid motion of a digital object X on \mathbb{Z}^n is given by $\mathcal{T}(X)$. Due to the behavior of \mathfrak{D} that maps \mathbb{R}^n onto \mathbb{Z}^n , digitized rigid motions are, most of the time, non-bijective, so that they guarantee neither topology nor geometry preservation, as illustrated in Figures 51 and 52.

4.2 COMBINATORIAL LOCAL ANALYSIS

As such alterations happen locally, due to digitization, discrete motion maps have been studied for small image patches, in order to understand such defects at local scale [142], [143]. For such a local analysis, one wishes to generate all possible images of an image patch under digitized rigid motions. In digital geometry and combinatorics, complexity analyses of such a problem has been made for certain geometric transformations. However, there are few algorithms available for generating all the transformed images from a given image patch. In this chapter, we study combinatorial and local structures of 2D and 3D digitized rigid motions, mainly published in [39], [149]. In this section, we consider the following issues:

1. the complexity of digitized rigid transformations of a finite image patch of \mathbb{Z}^n ,

2. the graphical modelling of the structure of digitized rigid transformations, and
3. the algorithm for computing the structure.

4.2.1 Related work

Similar study on combinatorial analysis of 2D image transformations, where theoretical complexities of some digitized geometric transformations are summarized in Table 11, were made for rotations [5], [189], scalings [4], [6], combined scalings and rotations [79], affine transformations [77], [80], projective and linear transformations [78].

Classes of transformations	Complexity
Rotations [5], [189]	$\mathcal{O}(r^3)$
Scalings [4], [6]	$\mathcal{O}(r^3)$
Rotations and scalings [79]	$\mathcal{O}(r^6)$
Linear transformations [78]	$\mathcal{O}(r^{12})$
Affine transformations [77], [80]	$\mathcal{O}(r^{18})$
Projective transformations [78]	$\mathcal{O}(r^{24})$

Table 11: Space complexities of different classes of transformations on a finite subset of \mathbb{Z}^2 of radius r .

On the other hand, few algorithms which enable the construction of the combinatorial structures of those digitized geometric transformations were proposed: 2D rotations [142]; 3D rotations around a given rational axis [190]; 2D rigid motions [39], [153], 3D rigid motions (without the structure) [149] and 2D affine transformations [80]. In the sections below, we first explain the theoretical complexity for digitized geometric transformations and then the algorithm for constructing the combinatorial structures of rigid motions of a finite subset of \mathbb{Z}^2 , proposed in [39], and the one for generating all the rigid motions of a finite subset of \mathbb{Z}^3 , proposed in [149].

4.2.2 Image patch and its alterations under digitized rigid motions

Let us consider a finite set $S \subset \mathbb{Z}^n$, called an *image patch*, whose center c and radius r of S are respectively given by

$$c = \frac{1}{|S|} \sum_{p \in S} p$$

and

$$r = \max_{p \in S} \|p - c\|.$$

Note that, in this section, we consider c as the origin for simplicity, and study the evolution of such an image patch S under digitized rigid motions \mathcal{T} .

The digitized rigid motions $\mathcal{T} = \mathcal{D} \circ \mathfrak{T}$ are piecewise constant, and thus non-continuous, which is a consequence of the nature of the digitization operator \mathcal{D} . In particular, the image $\mathcal{T}(p)$ of a given point p may remain constant as the parameters of \mathcal{T} vary, and then suddenly jump from one point of \mathbb{Z}^n to another. In other words, an image patch S evolves non-continuously, under digitized rigid motions, in accordance with the parameters of \mathfrak{T} that underlies \mathcal{T} . Hereafter, without loss of generality, we assume that $\mathcal{T}(c)$ stays in the digitization cell of c , namely $\mathcal{T}(c) = c$, since translation by an integer vector would not change the geometry of S . Under this assumption we have that

$$t \in \left(-\frac{1}{2}, \frac{1}{2}\right)^n.$$

Studying the non-continuous evolution of an image patch S is equivalent to studying the discontinuities of $\mathcal{T}(p)$ for every $p \in S \setminus \{c\}$, which occur when $\mathcal{T}(p)$ is on the *half-grid* plane, namely a boundary of a digitization cell. This is formulated by

$$R_i p + t_i = k_i - \frac{1}{2} \quad (37)$$

where $k_i \in \mathbb{Z} \cap [-r', r']$, R_i is the i -th row of the rotation matrix for $i = 1, \dots, n$ and r' is the longest radius of $\mathfrak{T}(S)$ for all \mathfrak{T} , so that $r' = r + \sqrt{n}$.

4.2.3 Problem as an arrangement of hypersurfaces

For any image patch S , the $\frac{n(n+1)}{2}$ -dimensional parameter space of \mathfrak{T} is partitioned by a set of hypersurfaces given by (37) into a finite number of connected subsets. Although points in each $\frac{n(n+1)}{2}$ -dimensional open cell induce different rigid motion $\mathfrak{T}|_S$, their digitized rigid motions $\mathcal{T}|_S = \mathcal{D} \circ \mathfrak{T}|_S$ are identical.

In order to construct a combinatorial model of the local behavior of displacements on \mathbb{Z}^n , the transformations can be classified according to their effect on a digital image patch. This classification can be interpreted as specific problems of hypersurface arrangements in the parameter space of the transformations.

4.2.4 Combinatorial analysis of structures²

Let \mathcal{C} be a collection of hypersurfaces defined by Eq. (37) for the arrangement. For the combinatorial analysis of the structure of the arrangement \mathcal{C} , there are three important elements that we should consider: the number m of hypersurfaces, the number d of hypersurface parameters, and the maximum degree b of hypersurfaces which are in algebraic surfaces.

² Special thanks goes to Xavier Goaoc for the fruitful discussions and variable advices.

Let us here consider the rotation representation by Cayley transform, described in Section 4.1.2. Then, each of the hypersurfaces defined by (37) is written in the form:

$$P(a_1, a_2, \dots, a_d) = 0$$

where $P(a_1, a_2, \dots, a_d)$ is a polynomial function with

$$d = \frac{n(n+1)}{2} \quad (38)$$

where n is the dimension of digital image space, namely $n = 2$ or 3 here. From (32), (34)/(31) and (37), we also obtain

$$b \leq 3$$

for any P , and thus b is a small constant.

Given an image patch S of radius r , let us now consider the collection of the associated hypersurfaces defined by (37). From the definition, we have the following lemma since (37) varies for $p \in S \setminus \{c\}$ and for $k_i \in \mathbb{Z} \cap [-r', r']$ where $r' = r + \sqrt{n}$.

Lemma 4.1. *Given an n -dimensional image patch S of radius r , the complexity of the collection of the associated hypersurfaces defined by (37) is $O(r^{n+1})$.*

Applying Theorem 21.1.4 in [72] to this specific problem and taking into consideration Lemma 4.1, we have the following complexity of the arrangement \mathcal{C} . Note that the complexity of intersection points and that of d -dimensional cells are equal.

Corollary 4.1. *Given an n -dimensional image patch S of radius r , let \mathcal{C} be a collection of m hypersurfaces with d parameters defined by (37). The maximum combinatorial complexity of the arrangement of \mathcal{C} is $O(m^d) = O(r^{\frac{n(n+1)^2}{2}})$.*

Similar combinatorial analyses can be done for other geometric transformations, which are for instance listed in Table 11. The corollary is then generalized as follows.

Corollary 4.2. *Given an n -dimensional image patch S of radius r , let \mathcal{C} be a collection of hypersurfaces with d parameters, defined from critical hypersurfaces of digitized geometric transformations in the form of affine/projective transforms. The maximum combinatorial complexity of the arrangement of \mathcal{C} is $O(r^{(n+1)d})$.*

This enables us to obtain the complexity analyses in Table 12, parts of which, particularly in 2D, were same as those in Table 11 given above.

Classes of transformations	n (dim.)	d (para. #)	Complexity
2D translations	0	2	$O(r^2)$
2D scalings	2	1	$O(r^3)$
2D rotations	2	1	$O(r^3)$
2D rotations and scalings	2	2	$O(r^6)$
2D rigid transformations	2	3	$O(r^9)$
2D linear transformations	2	4	$O(r^{12})$
2D affine transformations	2	6	$O(r^{18})$
2D projective transformations	2	8	$O(r^{24})$
3D translations	0	3	$O(r^3)$
3D scalings	3	1	$O(r^4)$
3D axis-fixed rotations	3	1	$O(r^4)$
3D rotations	3	3	$O(r^{12})$
3D rotations and scalings	3	4	$O(r^{16})$
3D rigid transformations	3	6	$O(r^{24})$
3D linear transformations	3	9	$O(r^{36})$
3D affine transformations	3	12	$O(r^{48})$
3D projective transformations	3	15	$O(r^{60})$

Table 12: Combinatorial analyses of complexity of different classes of transformations on a subspace of \mathbb{Z}^n , $n = 2, 3$, of radius r , obtained from Collorary 4.2.

4.2.5 Graph representing the dual combinatorial structures

An arrangement of $(d - 1)$ -dimensional hypersurfaces defined by (37) subdivides the parameter space \mathbb{R}^d (see Fig. 53 (a) for an example of 2D digitized rigid motions). By mapping each d -dimensional open cell onto a vertex, and each separating $(d - 1)$ -dimensional hypersurface onto an edge – following a Voronoi / Delaunay duality – we can model such a subdivision of the parameter space, as a graph, called *discrete rigid transformation (DRT) graph* (see Fig. 53 (b) for an example).

Definition 4.1. Given a collection \mathcal{C} of critical hypersurfaces defined by (37) in \mathbb{R}^d , the dual graph $G = (V, E)$, called DRT graph, is defined such that:

1. any vertex $v \in V$ models a d -dimensional open cell of the arrangement of \mathcal{C} ;
2. any edge $e = (v, w) \in E$ connects two vertices $v, w \in V$ sharing a $(d - 1)$ -dimensional hypersurface of \mathcal{C} .

This DRT graph G represents the dual structure of the subdivision of the parameter space \mathbb{R}^d of rigid transformations. In particular, G associated to a digital image patch S of finite size (according to (37), the size is described by the radius of S denoted by r) is a finite data-structure that describes *all* the possible digital rigid transformations of S .

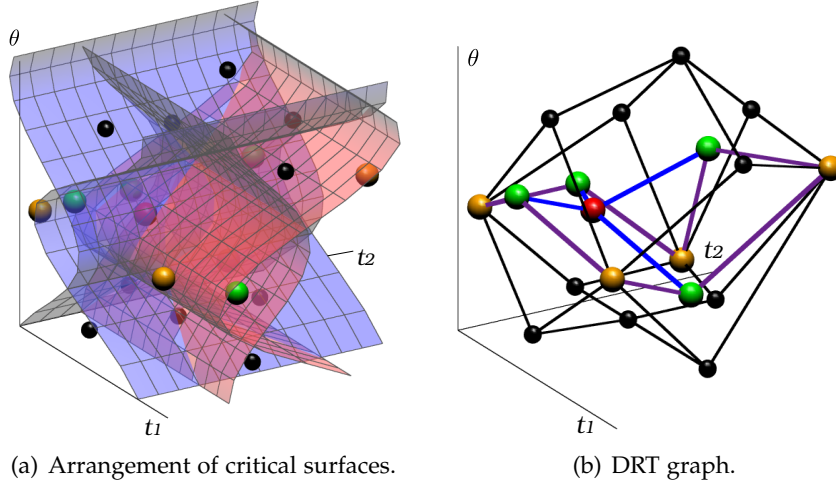


Figure 53: (a) Subdivision of the (t_1, t_2, θ) parameter space of 2D rigid motions into 3D cells by critical surfaces, which are vertical (in blue) and horizontal (in red), and (b) the associated DRT graph.

It should be mentioned that the DRT graph G does not contain any geometric parameters of rigid transformations but only the *topological* information, that models the relationship between any neighbouring transformed images. More precisely, each vertex $v \in V$ of G is associated with a unique transformed image generated by any rigid transformation in the dual open cell of v . It is shown in [39] that two connected transformed images in G differ by at most *one* over the $O(r^n)$ pixels/voxels of S . Thus, each edge $e = (v, w) \in E$ of G is –implicitly– associated to a function indicating the *only* modification that differs between the transformed images corresponding to v and w . This allows us to use the DRT graph to produce all the transformed images via successive *elementary* (i.e., single-pixel) modifications. This property opens a way of involving the DRT graph in digital image processing and analysis tasks.

4.2.6 Algorithms for computing the structure of digitized rigid motions

The theoretical combinatorial analysis above gave us the complexities $O(r^9)$ and $O(r^{24})$ for two- and three-dimensional digitized rigid motions, respectively. The corresponding arrangement problems can be solved by methods of computational geometry. For the calculation, despite the existence of degenerate cases, the algorithm based on plane sweeping is proposed for $n = 2$, and it can be applied to an image patch whose cardinality is greater than 100 [39] (see Section 4.2.6.1). Note that topological information of the arrangement, namely DRT graph, is obtained. On the other hand, working in \mathbb{Z}^3 increases considerably the complexity of the problem, and thus we set the simpler goal for $n = 3$ such as obtaining at least one sample point for

each 6-dimensional open cell in the parameter space \mathbb{R}^6 , than that of constructing the DRT graph G for $n = 2$. We tackle this problem in two stages: (1) the calculation of sampling points in an arrangement of quadrics in the parameter space of rotations; (2) the retrieval of information on the other parameters of translations [149] (see Section 4.2.6.2).

4.2.6.1 Computing an arrangement for 2D digitized rigid motions

Here we consider a finite collection \mathcal{C} of critical surfaces defined by (37) with the rotation parameterisation by angle θ , namely (30), with a given value r . Note that there are two sets of critical surfaces, denoted by \mathcal{C}_i for $i = 1, 2$, called vertical and horizontal respectively, so that $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$. As illustrated in Fig. 53 (a), any critical surface of \mathcal{C}_i is orthogonal to the (t_i, θ) -plane. We explain an efficient algorithm for constructing the DRT graph G , with a low computational cost, which is linear with respect to its space complexity. The algorithm, which handles real (*i.e.*, non-rational) values related to the continuous transformations associated to the discrete ones, is however defined in a fully discrete form, allowing for exact computation, and avoiding in particular any approximations related to floating point arithmetics.

The algorithm consists of using a plane orthogonal to the θ -axis, denoted by Γ , swept along this axis, from 0 to 2π . From the definition of critical surfaces (37), the critical surfaces of \mathcal{C} intersect and subdivide Γ into at most $(|\mathcal{C}_1| + 1) \times (|\mathcal{C}_2| + 1)$ rectangular cells, as illustrated in Fig. 54. Each rectangle corresponds to a vertex, while each frontier between two rectangles corresponds to an edge of the DRT graph. When swept Γ reaches an intersection of critical surfaces of \mathcal{C} , some rectangles disappear while new ones appear right after; new vertices and edges are generated and added into the DRT sub-graph G . In order to construct incrementally G , we need to maintain a set of sorted intersection points of critical surfaces with respect to θ in $[0, 2\pi[$, and to make Γ progress in the increasing order (see [39] for more details). Note that this algorithm has a complexity that depends on the number of these intersections, namely $O(|\mathcal{C}|^2)$ [39].

One of the key points of this incremental construction of G is exact sorting of intersections of vertical (or horizontal) critical surfaces along the θ -axis. Note that we can use for the comparison the values $\cos \theta$ and $\sin \theta$, which are both quadratic irrationals³ in this specific problem [39]. It is known in [162] that two quadratic irrationals can be compared by an exact method. In fact, a quadratic irrational can be represented exactly using a periodic continued fraction modelled by a sequence of integers, and this representation is unique. Moreover the comparison of periodic continued fractions can be performed in constant time [61] (see Appendix A of [39] for more details), so

³ A quadratic irrational is an irrational number that is a solution of some quadratic equations.

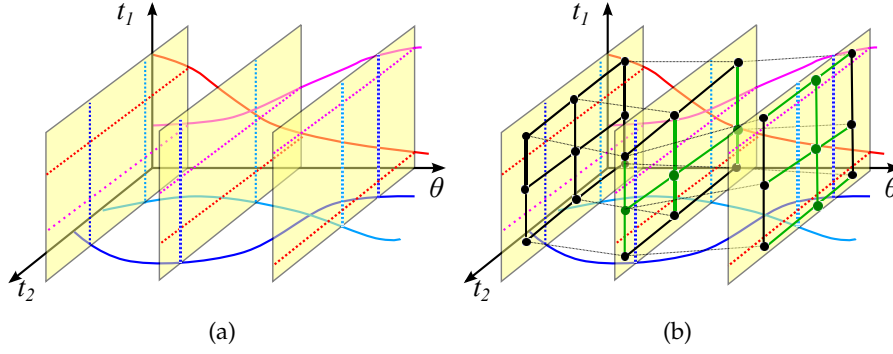


Figure 54: DRT graph construction by the sweeping plane algorithm, with 2 vertical (blue, cyan) and 2 horizontal (red, magenta) critical surfaces. (a) 3×3 rectangular cells generated by the tipping surfaces in each sweeping plane. (b) The associated part of the DRT graph in each plane (in green: new vertices and edges in the second and third planes).

that sorting all intersections requires $O(N \log N)$ times of such a comparison, where N is the number of intersections, namely $N = O(|\mathcal{C}|^2)$ as mentioned above.

Moreover, from Fig. 54, one remarks that the critical surfaces intersect the sweeping plane Γ with a specific order along each t_i -axis. At each intersection, this order evolves, but only between consecutive critical surfaces involved in the intersection. Therefore, instead of calculating intersections between *all* critical surfaces in \mathcal{C} , we can calculate only those of consecutive critical surfaces in their ordered structure in Γ along each t_i -axis. In consequence, we consider at most $|\mathcal{C}| - 2$ intersections at each update of Γ . The next intersection of critical surfaces for updating the planar subdivision is the closest among these $|\mathcal{C}| - 2$. After such intersection, the order of the critical surfaces generating the intersection in Γ is swapped.

The overall procedure starts at $\theta = 0$ with the ordered structure in Γ with \mathcal{C} along each t_i -axis and, at each intersection, calculates $|\mathcal{C}_i|$ new vertices and their associated $(3|\mathcal{C}_i| + 2)$ edges, that are integrated into G , while the ordered structure in Γ is updated. This is repeated until Γ reaches 2π . After each update, the modifications of such intersections can be performed in constant time.

In summary, the initial graph has a complexity $O(|\mathcal{C}|) \times O(|\mathcal{C}|)$, and the graph update can be made in $O(|\mathcal{C}|)$ at each intersection, of which the total number is $O(|\mathcal{C}|^2)$. Thus the complexity of the overall procedure is $O(|\mathcal{C}|^3)$. From Lemma 4.1, we have $|\mathcal{C}| = O(r^3)$, and thus this leads to $O(r^9)$, which is equal to the space complexity shown in Corollary 4.1.

The similar sweeping strategy can be also used for constructing a local structure of the DRT graph G [134]; given a vertex v , compute the neighboring vertices and edges of v . This can be made in $O(kr^2)$

where k is the maximum distance of the neighbors from v such that the length of every edge is considered to be 1. This local information is, for example, used for handling image registration [134].

4.2.6.2 Computing an arrangement for 3D digitized rigid motions

Here we consider a finite collection \mathcal{C} of critical surfaces defined by (37) with the rotation parameterisation by angle θ , namely (33), with a given value r and for $n = 3$. From (38), the number of variables of this specific problem is 6, of which 3 variables are for rotations and the other 3 are for translations. With this parametrisation, the parameter space is

$$\Omega = \left\{ (a, b, c, t_1, t_2, t_3) \in \mathbb{R}^6 \mid a, b, c \geq 0, -\frac{1}{2} < t_i < \frac{1}{2} \text{ for } i = 1, 2, 3 \right\}$$

for a 3-dimensional image patch S . In order to avoid dealing with the hypersurface arrangement in this 6-dimensional space, we uncouple the parameters in Ω as follows. By considering the differences between the hypersurfaces given in (37) for different $p \in S$ and $k \in H(S)^3 = \mathbb{Z} \cap [-r - \sqrt{3}, r + \sqrt{3}]$, we can reduce the problem to the study of an arrangement of surfaces in the (a, b, c) -space, and then lift the solution to the six dimensional space.

Let us consider a rigid motion defined by R and t . The condition for having $\mathcal{T}(p) = k = (k_1, k_2, k_3) \in \mathbb{Z}^3$ where $p \in S$ is

$$k_i - \frac{1}{2} < R_i p + t_i < k_i + \frac{1}{2}$$

for $i = 1, 2, 3$. Equivalently,

$$k_i - \frac{1}{2} - R_i p < t_i < k_i + \frac{1}{2} - R_i p. \quad (39)$$

Let us call a *configuration* a list of couples (p, k) , which describe how the image patch S is transformed. This configuration can be described as a function

$$\left| \begin{array}{ll} F : S & \rightarrow H(S)^3 \\ p = (p_1, p_2, p_3) & \mapsto k = (k_1, k_2, k_3). \end{array} \right.$$

We would like to ascertain whether a given configuration F arises from some digitized rigid motion \mathcal{T} , i.e. corresponds to some parameters a, b, c, t_1, t_2, t_3 . Then the inequalities (39) state precisely the necessary and sufficient conditions for the existence of the translation part t of such a rigid motion, assuming that a, b, c are already known. Let us now remark that all these inequalities can be summed up in *three* inequalities indexed by i :

$$\max_{p \in S} \left(F(p)_i - \frac{1}{2} - R_i p \right) < \min_{p \in S} \left(F(p)_i + \frac{1}{2} - R_i p \right), \quad (40)$$

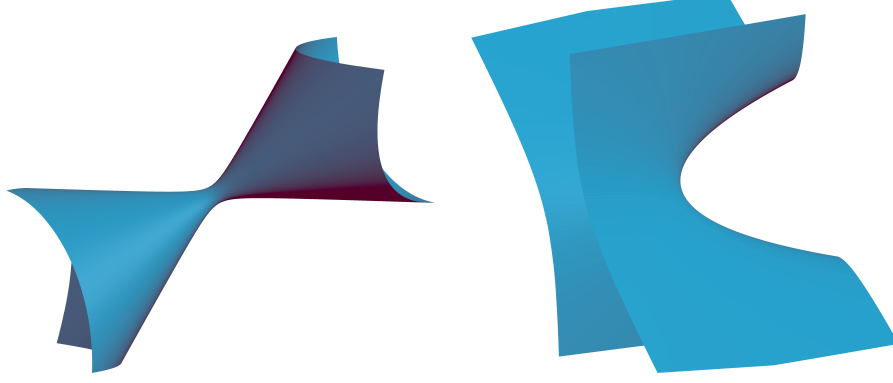


Figure 55: Examples of the zero sets of quadratic polynomials of Q .

or equivalently to the following list of inequalities

$$\forall p, p' \in S, \quad F(p')_i - \frac{1}{2} - R_i p' < F(p)_i + \frac{1}{2} - R_i p. \quad (41)$$

The key observation is that we have eliminated the variables t_1, t_2, t_3 and have reduced to a subsystem of inequalities in a, b, c . Moreover, due to the rational expression in the Cayley transform (33), we may use the following polynomials of degree 2:

$$q_i[p, k_i](a, b, c) = (1 + a^2 + b^2 + c^2)(2k_i - 1 - 2R_i p), \quad (42)$$

for $i = 1, 2, 3$. Then (41) can be rewritten as the quadratic polynomial inequalities

$$\forall p, p' \in S, \quad Q_i[p, p', F(p)_i, F(p')_i](a, b, c) > 0,$$

where

$$Q_i[p, p', k_i, k'_i](a, b, c) = q_i[p, k_i](a, b, c) + 2(1 + a^2 + b^2 + c^2) - q_i[p', k'_i](a, b, c), \quad (43)$$

for $i = 1, 2, 3$. The set of quadratic polynomials for this specific problem is then given by

$$Q = \{Q_i[p, p', k_i, k'_i](a, b, c) \mid i = 1, 2, 3, p, p' \in S, k_i, k'_i \in H(S)\}.$$

Figure 55 illustrates the zero sets of some quadratic polynomials in Q .

Computation of the arrangement of quadrics $Q(a, b, c) = 0$ for all $Q \in Q$ can be made in a similar way to the strategy proposed in [127] (see [149] for more details). One of the main differences is that we do not store information about all cells different from sample points of full-dimensional connected components. Another difference is that we compute a sample point per open connected component of a quadric arrangement using a projection along a non-generic direction, using the theory of *generalized critical values* [82], [109], [155], while the

original sweeping strategy is proposed along a generic direction [127]. Let us here consider the a -axis for such a non-generic direction. We then consider not only the following three types of critical values:

TYPE A: values s of a where the sweeping plane $a = s$ is tangent to a quadric of Q ;

TYPE B: values s of a where the curve of intersection of two quadrics of Q is tangent to the sweeping plane;

TYPE C: values s of a where the sweeping plane contains an intersecting point of three quadrics of Q ;

as illustrated in Figs. 56, 57 and 58 respectively, but also the following two types of asymptotic critical values:

TYPE A_∞ : values s of a where the sweeping plane $a = s$ is tangent to a quadric at infinity;

TYPE B_∞ : values s of a where the sweeping plane is tangent to a curve defined by the intersection of two quadrics at infinity,

as illustrated in Figs. 59 and 60 respectively.

The critical values are obtained as systems of quadratic equations (see [149] for the details) and are represented by real algebraic numbers, namely roots of univariate polynomials. We can make comparison between two algebraic numbers, which is necessary to sort the critical values, similarly to [127].

Once detecting and sorting all the critical values at which topology of an arrangement changes, we sweep by a plane the set of quadrics along a chosen direction, which is here the a -axis. The sweeping plane stops between each pair of consecutive critical values and take a sample value corresponding to the position of the sweeping plane. We then consider the intersection of the quadrics and the sweeping plane. This intersection reduces the problem into the subproblem of

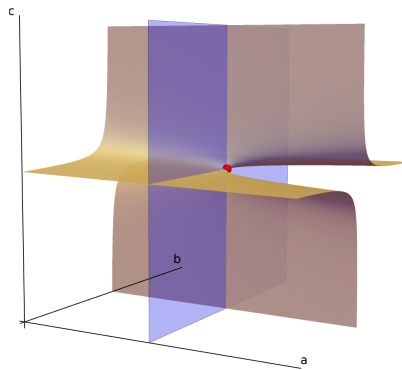


Figure 56: Example of critical value of the type A – a sweeping plane $a = 0$ (depicted in violet) tangent at a point (depicted in red) to a quadric $Q = bc - a$.

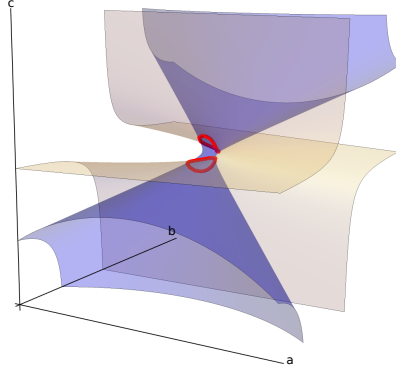


Figure 57: Example of critical value of the type B – a sweeping plane is tangent to two curves (depicted in red) given by an intersection of quadrics $Q_1 = bc + a$ and $Q_2 = a^2 + b^2 - 4bc + c^2 + 4a + 1$.

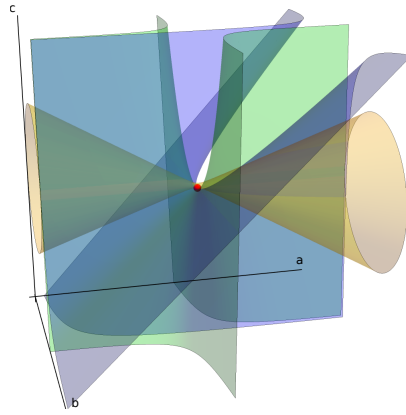


Figure 58: Example of critical value of the type C – a point (depicted in red) given by an intersection of quadrics $Q_1 = ab - c$, $Q_2 = ab - ac - b - c$ and $Q_3 = a^2 - ab + c^2 - c$.

a 2D arrangement of conics on the sweeping plane (see Fig. 61). We then compute a 3D sample point for each 2D cell of the arrangement on the plane.

It is obvious that this method can create more than one sample point in a 3D cell of the quadric arrangement. However, this method guarantees that we do not overlook a 3D cell where no sample point is generated. In order to reduce such excess of sample points, we can also consider an incremental approach, which requires instead more complex data structures, as presented in [127].

After this procedure, for each sample point we recover the translation part of the parameter space of digitized 3D rigid motions from (39). Notice that the proposed approach could be also applied to solve the problem in 2D, as described in the previous section, i.e. generation of the different images of a 2D image patch under 2D digitized rigid motions, which leads the same results obtained in [39].

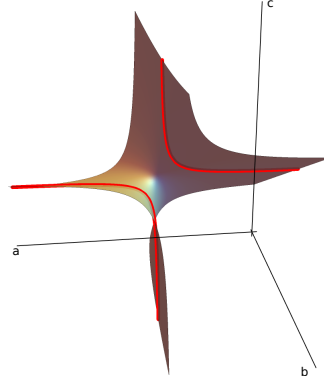


Figure 59: Example of asymptotic critical value of A_∞ type. There exist a plane $a = 0$ tangent to an asymptote—the red curves in the surface $Q = ab - 2ac - 2b - c$ —in a point at infinity.

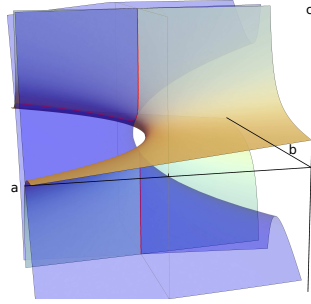


Figure 60: Example of asymptotic critical event of B_∞ type. The intersection of quadrics: $Q_1 = bc + a$ (the orange-green surface) and $Q_2 = b^2 - bc + a - 1$ (the blue surface) leads to the red curves which exhibit an asymptotic behavior at $a = \frac{1}{2}$. The a -, b - and c - axes are re-oriented and the origin is changed for a better visualization effect.

4.3 CHARACTERIZATION OF BIJECTIVITY

Although the displacements are bijective and isometric in \mathbb{R}^n , these properties are lost during the digitization in \mathbb{Z}^n (see Figure 51). To study these defects, we extend a combinatorial model of the local behavior, originally defined for rotations [144], to displacements in \mathbb{Z}^2 [153]. This model allows us to study the bijective displacements in \mathbb{Z}^2 , and to propose algorithms for verifying if a given displacement is bijective restricted to a fixed finite subset of \mathbb{Z}^2 [153]. It should be mentioned that the same characterization can be obtained by an arithmetic approach based on the properties of Gaussian integers [168].

On the other hand, in the case of displacements in \mathbb{Z}^3 , more precisely, of discrete rotations, a similar characterization does not exist. However, an arithmetic algorithm to certify the (global) bijectivity of digitized rational rotations was obtained using the properties of Lipschitz quaternions [150]. This algorithm also makes it possible to show

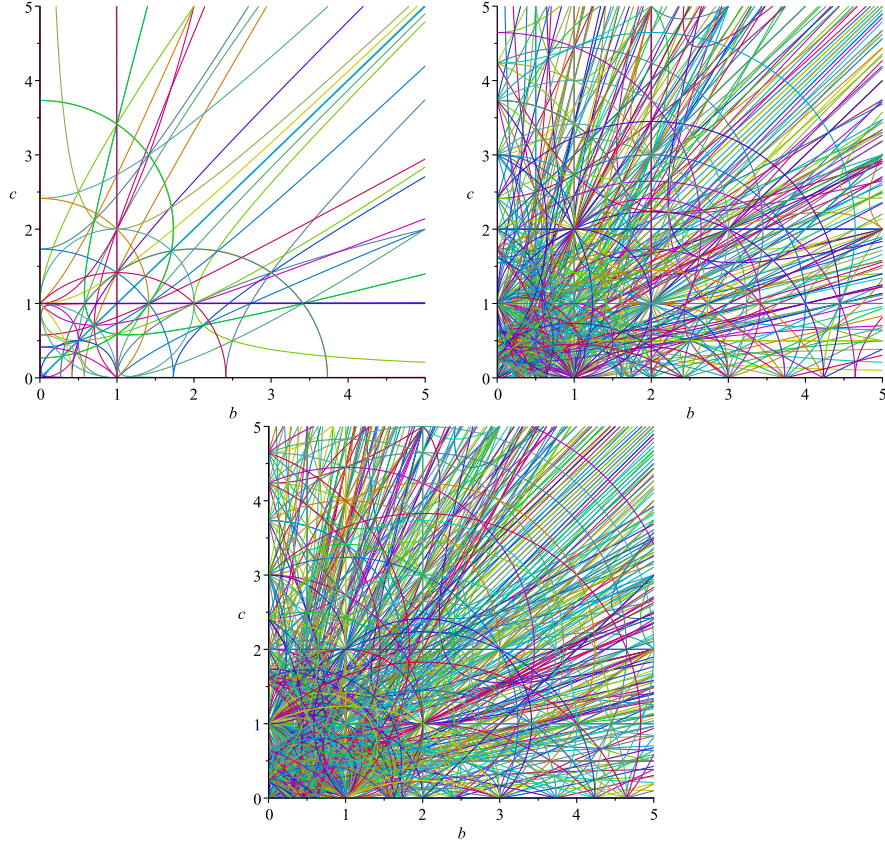


Figure 61: The 2D arrangement of 81, 513 and 741 quadrics for an image patch of radius $r = 1, \sqrt{2}, \sqrt{3}$, respectively.

the existence of bijective digitized rotations whose axis of rotation does not correspond to one of the coordinate axes.

Both the combinatorial and arithmetic approaches can also handle other regular tilings, such as hexagonal ones [151], [152]. Here we present the combinatorial and arithmetic approaches to study the bijectivity of displacements on \mathbb{Z}^2 and \mathbb{Z}^3 , respectively.

4.3.1 Combinatorial approach to characterization of bijectivity

4.3.1.1 Neighborhood motion map

In order to track the local behavior of displacements \mathcal{T} in \mathbb{Z}^2 , we introduce the notion of a neighborhood motion map that is defined as a set of vectors, each representing information about a neighbor after rigid motion. This notion was originally proposed by Nouvel and Rémila to study local behavior of 2D digitized rotations [143].

Let us first consider the neighborhood of $\mathbf{p} \in \mathbb{Z}^2$ of squared radius $r \in \mathbb{R}_+$, defined as

$$\mathcal{N}_r(\mathbf{p}) = \{\mathbf{p} + \mathbf{d} \in \mathbb{Z}^2 \mid \|\mathbf{d}\|^2 \leq r\}.$$

Then the neighborhood motion map of $\mathcal{N}_r(\mathbf{p})$ is defined as follows.

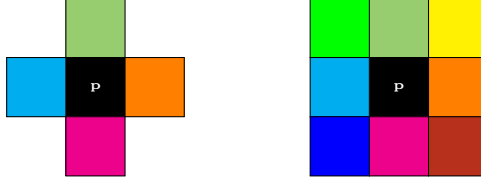


Figure 62: The reference label maps \mathcal{L}_1 (left) and \mathcal{L}_2 (right).

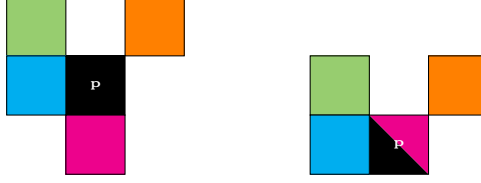


Figure 63: Examples of label maps $\mathcal{L}_1^T(p)$. Each point contains at most one label (left): the rigid motion \mathcal{T} is then locally injective. One point contains two labels (right): \mathcal{T} is then non-injective.

Definition 4.2 (Neighborhood motion map). *Given a digitized rigid motion \mathfrak{T} , the neighborhood motion map of $p \in \mathbb{Z}^2$ is defined as*

$$\left| \begin{array}{lcl} \mathcal{G}_r^T(p) & : & \mathcal{N}_r(\mathbf{0}) \rightarrow \mathcal{N}_{r'}(\mathbf{0}) \\ & & d \mapsto \mathcal{T}(p + d) - \mathcal{T}(p) \end{array} \right.$$

where $r' \geq r$.

The map $\mathcal{G}_r^T(p)$ associates to each relative position of an integer point $q = p + d$ in the neighborhood of p , the relative position of the image $\mathcal{T}(q)$ in the neighborhood of $\mathcal{T}(p)$.

For a visualization, $\mathcal{G}_r^T(p)$ is represented by label maps. A reference map \mathcal{L}_r associates a specific label to each point $d = q - p$ of $\mathcal{N}_r(\mathbf{0})$ for a given squared radius r (see Fig. 62, for the maps \mathcal{L}_1 and \mathcal{L}_2). A second map $\mathcal{L}_r^T(p)$ associates to each point r of $\mathcal{N}_{r'}(\mathbf{0})$ the labels of all the points q such that $\mathcal{T}(q) - \mathcal{T}(p) = r$. Such a set of labels for each r may contain 0, 1 or 2 labels, due to the possible mappings of integer points under digitized rigid motions (see examples in Fig. 63).

4.3.1.2 Partitioning remainder range and neighboring motion maps

Due to the definition, the neighboring motion map $\mathcal{G}_r^T(p)$ evolves non-continuously. Our purpose is now to express how $\mathcal{G}_r^T(p)$ evolves.

Let us denote by $\mathcal{C}(p)$ the digitization cell centered at a grid point $p = (p_1, \dots, p_n) \in \mathbb{Z}^n$:

$$\mathcal{C}(p) = \prod_{i=1}^n \left[p_i - \frac{1}{2}, p_i + \frac{1}{2} \right).$$

Instead of studying the whole source and target spaces of rigid motions \mathfrak{T} , we study the set of remainders defined by the remainder map:

$$\left| \begin{array}{lcl} \rho : & \mathbb{Z}^n & \rightarrow \mathcal{C}(\mathbf{0}) \\ & p & \mapsto \mathfrak{T}(p) - \mathfrak{D} \circ \mathfrak{T}(p). \end{array} \right.$$

Let us now consider an integer point $p + d$ in the neighborhood $\mathcal{N}_r(p)$ of p . From (29), we have

$$\mathfrak{T}(p + d) = Rd + \mathfrak{T}(p),$$

which is rewritten as

$$\mathfrak{T}(p + d) = Rd + \rho(p) + \mathfrak{T}(p).$$

Without loss of generality—and up to translations in \mathbb{Z}^2 —we can assume that $\mathcal{T}(p)$ is the origin of a local coordinate frame of the image space, $\mathcal{T}(p) \in \mathcal{C}(\mathbf{0})$. In these local coordinates frames, the former equation rewrites as

$$\mathfrak{T}(d + p) = Rd + \rho(p). \quad (44)$$

Now, studying the non-continuous evolution of the neighborhood motion map $\mathcal{G}_r^{\mathcal{T}}(p)$ is equivalent to studying the behavior of $\mathcal{T}(d) = \mathfrak{D} \circ \mathcal{T}(p + d)$ for $d \in \mathcal{N}_r(\mathbf{0})$ and $p \in \mathcal{C}(\mathbf{0})$, with respect to the motion parameters. In particular, the discontinuities of $\mathcal{T}(p + d)$ occur when $\mathfrak{T}(p + d)$ is on the boundary of a digitization cell. Setting $\rho(p) = (x_1, x_2) \in \mathcal{C}(\mathbf{0})$ and $d \in \mathcal{N}_r(\mathbf{0})$, this is formulated by one of the equations:

$$x_i + R_i d = k_i - \frac{1}{2} \quad (45)$$

where $k_i \in \mathbb{Z}$, $i = 1, 2$. For a given d and k_i , (45) defines a vertical or horizontal line in the remainder range $\mathcal{C}(\mathbf{0})$, called a critical line. These critical lines with different d and k_i subdivide the remainder range $\mathcal{C}(\mathbf{0})$ into rectangular regions called *frames*. As long as coordinates of $\rho(p)$ belong to a same frame, the associated neighborhood motion map $\mathcal{G}_r^{\mathcal{T}}(p)$ remains constant.

Proposition 4.1. *For any $p, q \in \mathbb{Z}^2$, $\mathcal{G}_r^{\mathcal{T}}(p) = \mathcal{G}_r^{\mathcal{T}}(q)$ iff $\rho(p)$ and $\rho(q)$ are in the same frame.*

This proposition is shown originally in [143] for the case $r = 1$ and rotations without translation: the above is an extension for general cases, such that $r \geq 1$, and rigid motions. An example of the remainder range partitioning for $r = 2$ is presented in Fig. 64, and the corresponding neighboring motion map for each frame in Fig. 64 can be found in Fig. 65 (for further examples, see Appendices of [153]).

4.3.1.3 Characterization of globally bijective digitized rigid motions

Observing such neighboring motion maps, we can find the frames in which the neighborhood motion maps have points with two or zero preimages, implying non-surjectivity or non-injectivity [144]. Those frames are called non-surjective and non-injective frames, depicted as f_*^0 and f_*^2 in Fig. 64 (see [153] for details). Therefore, if there exists $p \in \mathbb{Z}^2$ such that $\rho(p)$ is in one of non-surjective (resp. non-injective) frames, then the digitized rigid motion \mathcal{T} is not surjective (resp. injective). In other words, we have the following proposition.

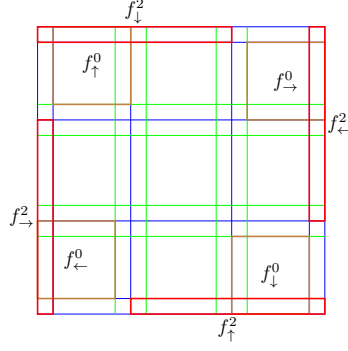


Figure 64: Example of remainder range partitioning for $r = 2$ for $\theta \in (0, \alpha_1)$, where α_1 is the smallest positive critical angle, together with non-injective zones f_*^2 and non-surjective zones f_*^0 are illustrated by red and brown rectangles, respectively.

Proposition 4.2. *A digitized rigid motion is bijective if and only if there is no $\rho(p)$ for all $p \in \mathbb{Z}^2$ in non-surjective nor non-injective frames of the remainder range $\mathcal{C}(\mathbf{0})$.*

We now characterize bijective rigid motions on \mathbb{Z}^2 while investigating those local conditions. Let us start with the rotational part of the motion. We know from [144] that rotations with any angle of irrational sine or cosine are non-bijective; indeed, such rotations have a dense image by ρ (there exists $p \in \mathbb{Z}^2$ such that $\rho(p)$ lies in a non-surjective or non-injective frame of $\mathcal{C}(\mathbf{0})$). This result is also applied to \mathcal{T} , whatever translation part is added.

Therefore, we focus on rigid motions for which both cosine and sine of the angle θ are rational. Such angles are called *Pythagorean angles* [144] and are defined by *primitive Pythagorean triples* $(a, b, c) = (p^2 - q^2, 2pq, p^2 + q^2)$ with $p, q \in \mathbb{Z}$, $p > q$ and $p - q$ is odd, such that (a, b, c) are pairwise coprime, and cosine and sine of such angles are $\frac{a}{c}$ and $\frac{b}{c}$, respectively. The image of \mathbb{Z}^2 by ρ , when \mathcal{T} is a digitized rational rotation, corresponds to a cyclic group \mathcal{G} on the remainder range $\mathcal{C}(\mathbf{0})$, which is generated by $\psi = (\frac{p}{c}, \frac{q}{c})$ and $\omega = (-\frac{q}{c}, \frac{p}{c})$ and whose order is equal to $c = p^2 + q^2$ [144] (see Fig. 66 for an example of \mathcal{G}). When \mathcal{T} contains a translation part, the image of ρ in $\mathcal{C}(\mathbf{0})$, which we denote by \mathcal{G}' , is obtained by translating \mathcal{G} (modulo \mathbb{Z}^2), and $|\mathcal{G}'|$ is equal to the order of \mathcal{G} , its underlying group. Note as well the following result, which was shown by Nouvel and Rémila [144] and, more recently, by Roussillon and Cœurjolly [168] with an arithmetical approach.

Proposition 4.3 ([144], [168]). *A digitized rational rotation is bijective (the intersection of \mathcal{G} with non-injective and non-surjective regions is empty) iff its angle comes from a twin Pythagorean triple—a primitive Pythagorean triple with the additional condition $p = q + 1$.*

Our question is then whether a digitized rigid motion can be bijective, even when the corresponding rotation is not. In order to answer

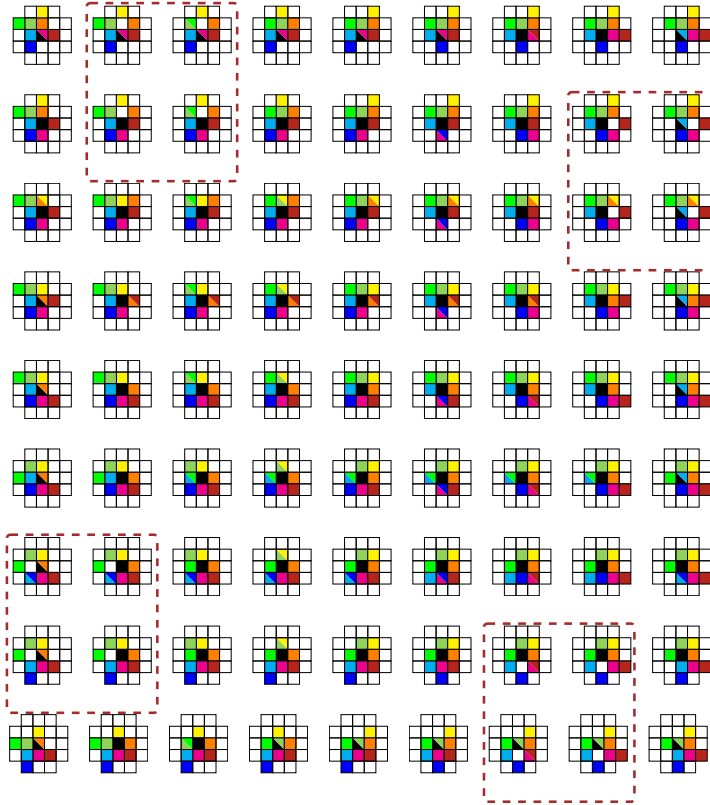


Figure 65: Neighborhood motion maps \mathcal{G}_2^T , represented by label maps, corresponding to each frame in Fig. 64. Neighborhood motion maps which correspond to non-injective zones are marked by brown, dashed, frames.

this question, we use the following equivalence property: digitized rational rotations are bijective if they are surjective or injective [144]. Indeed, this allows us to focus only on non-surjective zones, since they are squares hence provide symmetry, then presenting interesting properties in terms of exact computing.

Lemma 4.2. *A digitized rigid motion whose rotational part is given by a non-twin Pythagorean primitive triple is always non-surjective.*

If, on the contrary, the rotational part is given by a twin Pythagorean triple, i.e. is bijective, then the rigid motion is also bijective, under the following condition.

Proposition 4.4. *A digitized rigid motion is bijective if and only if it is composed of a rotation by an angle defined by a twin Pythagorean triple (a, b, c) and a translation $t = t' + \mathbb{Z}\psi + \mathbb{Z}\omega$, where $t' \in (-\frac{1}{2c}, \frac{1}{2c})^2$.*

4.3.1.4 Verification of locally bijective digitized rigid motions

The globally bijective digitized rigid motions, though numerous, are not dense in the set of all digitized rigid motions. Thus, we may

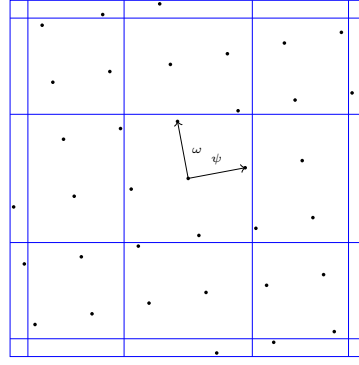


Figure 66: An example of remainder range partitioning together with the cyclic group \mathcal{G} obtained for the rotation defined by the primitive Pythagorean triple $(12, 35, 37)$. Note that the associated digitized rotation is non-bijective.

generally expect defects, such as points with two preimages. However, in practical applications, the bijectivity of a given \mathcal{T} on the whole \mathbb{Z}^2 is not the main issue; rather, one usually works on a finite subset of the plane (e.g., a rectangular digital image). The relevant question is then: “given a finite subset $S \subset \mathbb{Z}^2$, is \mathcal{T} restricted to S bijective?”. Actually, the notion of bijectivity in this question can be replaced by the notion of injectivity, since the surjectivity is trivial, due to the definition of \mathcal{T} that maps S to $\mathcal{T}(S)$.

The basic idea for such local bijectivity verification is quite natural. Because of its quasi-isometric property, a digitized rigid motion \mathcal{T} can send at most two neighbors (in a \mathcal{N}_1) onto a same point. Thus, the lack of injectivity is a purely local matter, suitably handled by the neighborhood motion maps via the remainder map. Indeed, \mathcal{T} is non-injective, with respect to S iff there exists $p \in S$ such that $\rho(p)$ lies in the union $\mathcal{F} = f_{\downarrow}^2 \cup f_{\uparrow}^2 \cup f_{\leftarrow}^2 \cup f_{\rightarrow}^2$ of all non-injective frames. We propose two algorithms making use of the remainder map information, as an alternative to a brute force verification.

The first—*forward*—algorithm, verifies for each point $p \in S$, the inclusion of $\rho(p)$ in one of the non-injective frames of \mathcal{F} . The second—*backward*—algorithm first finds all points w in $\mathcal{G}' \cap \mathcal{F}$, called the *non-injective remainder set*, and then verifies if their preimages $\rho^{-1}(w)$ are in S . See [153] for further details of both algorithms.

Both algorithms apply to rational motions, i.e., with a Pythagorean angle given by a primitive Pythagorean triple and a rational translation vector. We capture essentially the behavior for all angles and translation vectors, since rational motions are dense. These assumptions guarantee the exact computations of the algorithms, which are based on integer numbers. Methods for angle approximation by Pythagorean triples up to a given precision is found in [9].

4.3.2 Arithmetic approach to certification of bijectivity

We now present an algorithm for certifying the bijectivity of 3D digitized rational rotations using the arithmetic properties of the Lipschitz quaternions.

4.3.2.1 Remainder set and bijectivity

Here we use quaternions for representing 3D rotations such as (35). Let us consider the remainder defined by

$$\left| \begin{array}{ll} S_q : \mathbb{Z}^3 \times \mathbb{Z}^3 & \rightarrow \mathbb{R}^3 \\ (x, y) & \mapsto qxq^{-1} - y. \end{array} \right.$$

Then, the bijectivity of digitized rotation \mathcal{T} defined by a quaternion q is expressed as

$$\forall y \in \mathbb{Z}^3 \exists! x \in \mathbb{Z}^3, S_q(x, y) \in \mathcal{C}(\mathbf{0}),$$

which is equivalent to the “double” surjectivity relation, used in [168]:

$$\left\{ \begin{array}{ll} \forall y \in \mathbb{Z}^3 \exists x \in \mathbb{Z}^3, S_q(x, y) \in \mathcal{C}(\mathbf{0}) \\ \forall x \in \mathbb{Z}^3 \exists y \in \mathbb{Z}^3, S_q(x, y) \in q\mathcal{C}(\mathbf{0})q^{-1} \end{array} \right. \quad (46)$$

provided that both sets $S_q(\mathbb{Z}^3, \mathbb{Z}^3) \cap \mathcal{C}(\mathbf{0})$ and $S_q(\mathbb{Z}^3, \mathbb{Z}^3) \cap q\mathcal{C}(\mathbf{0})q^{-1}$ coincide; in other words,

$$S_q(\mathbb{Z}^3, \mathbb{Z}^3) \cap ((\mathcal{C}(\mathbf{0}) \cup q\mathcal{C}(\mathbf{0})q^{-1}) \setminus (\mathcal{C}(\mathbf{0}) \cap q\mathcal{C}(\mathbf{0})q^{-1})) = \emptyset.$$

Hereafter, we shall rely on (46), and in the study of the bijectivity of digitized rotation \mathcal{T} , we focus on the values of S_q . More precisely, we study the group \mathcal{G} spanned by values of S_q :

$$\begin{aligned} \mathcal{G} = \mathbb{Z}q \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} q^{-1} + \mathbb{Z}q \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} q^{-1} + \mathbb{Z}q \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} q^{-1} \\ + \mathbb{Z} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \mathbb{Z} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \mathbb{Z} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \end{aligned} \quad (47)$$

4.3.2.2 Dense subgroups and non-injectivity

The key to understanding the conditions that ensure the bijectivity of \mathcal{T} is the structure of \mathcal{G} . For this reason, we start by looking at the image \mathcal{G} of S_q , and discuss its density.

Proposition 4.5. *If one or more generators of \mathcal{G} have an irrational term, then $\mathcal{G} \cap V$ is dense for some nontrivial subspace V . We say that \mathcal{G} has a dense factor.*

On the contrary, we have the following result.

Proposition 4.6. *If all generators of \mathcal{G} have only rational terms, then there exist vectors $\sigma, \phi, \psi \in \mathcal{G}$ which are the minimal generators of \mathcal{G} .*

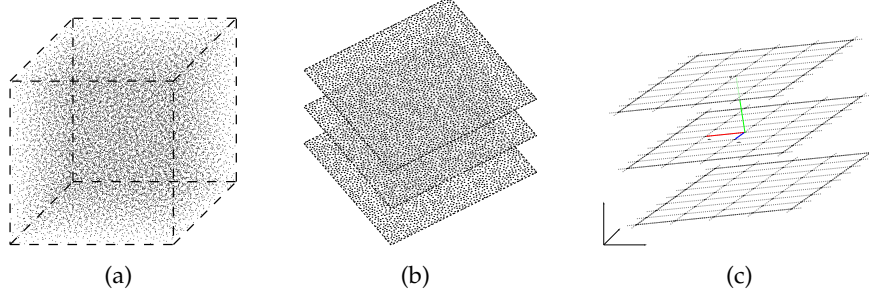


Figure 67: Illustration of a part of \mathcal{G} that: is dense (a); has a dense subgroup – the set of points at each plane is dense while the planes are spaced by a rational distance (b); is a lattice (c). In the case of (a) and (b), only some random points are presented, for the sake of visibility. In (c), vectors σ, ϕ, ψ are marked in red, blue and green, respectively.

Lemma 4.3. *Whenever \mathcal{G} is dense, the corresponding 3D digitized rotation is not bijective.*

When \mathcal{G} is dense (see Fig. 67(a)), the reasoning of Nouvel and Rémila, originally used to discard 2D digitized irrational rotations as being bijective [144], shows that a corresponding 3D digitized rotation cannot be bijective as well. What differs from the 2D case is the possible existence of non-dense \mathcal{G} with a dense factor (see Fig. 67(b)). In this context, we state the following conjecture.

Conjecture 4.1. *Whenever \mathcal{G} has a dense factor, the corresponding digitized rotation is not bijective.*

Henceforth, we assume that \mathcal{G} is generated by rational vectors, and forms therefore a lattice (see Fig. 67(c)). In other words, corresponding rotations are considered as *rational*. The question now remains of comparing the (finitely many) points in $S_q(\mathbb{Z}^3, \mathbb{Z}^3) \cap \mathcal{C}(\mathbf{0})$ and $S_q(\mathbb{Z}^3, \mathbb{Z}^3) \cap q\mathcal{C}(\mathbf{0})q^{-1}$.

4.3.2.3 Lipschitz quaternions and bijectivity

For representing 2D rational rotations, Gaussian integers are used in [168]. In \mathbb{R}^3 , rational rotations are characterized as follows [44].

Proposition 4.7. *There is a two-to-one correspondence between the set of Lipschitz quaternions $\mathbb{L} = \{a + bi + cj + dk \mid a, b, c, d \in \mathbb{Z}\}$ such that the greatest common divisor of a, b, c, d is 1, and the set of rational rotations.*

Working in the framework of rational rotations allows us to turn to integers: $|q|^2\mathcal{G}$ is an integer lattice. As integer lattices are easier to work with from the computational point of view, we do scale \mathcal{G} by $|q|^2$ in order to develop a certification algorithm.

Similarly to the former discussion, after scaling \mathcal{G} by $|q|^2$, we consider the finite set of remainders, obtained by comparing the lattice

$q\mathbb{Z}^3\bar{q}$ with the lattice $|q|^2\mathbb{Z}^3$, and applying the scaled version of the map S_q defined as

$$\left| \begin{array}{ll} \check{S}_q : \mathbb{Z}^3 \times \mathbb{Z}^3 & \rightarrow \mathbb{Z}^3 \\ (x, y) & \mapsto q \times \bar{q} - q \bar{q} y. \end{array} \right. \quad (48)$$

Indeed, (46) is rewritten by

$$\left\{ \begin{array}{ll} \forall y \in \mathbb{Z}^3 & \exists x \in \mathbb{Z}^3, \check{S}_q(x, y) \in |q|^2\mathcal{C}(\mathbf{0}) \\ \forall x \in \mathbb{Z}^3 & \exists y \in \mathbb{Z}^3, \check{S}_q(x, y) \in q\mathcal{C}(\mathbf{0})\bar{q}. \end{array} \right. \quad (49)$$

Note that the right hand sides of (48) and (49) are left multiples of q . As a consequence, we are allowed to divide them by q on the left, while keeping integer-valued functions. Let us define

$$\left| \begin{array}{ll} S'_q : \mathbb{Z}^3 \times \mathbb{Z}^3 & \rightarrow \mathbb{Z}^4 \\ (x, y) & \mapsto x\bar{q} - \bar{q}y. \end{array} \right.$$

Then, bijectivity of \mathcal{T} is ensured when

$$\left\{ \begin{array}{ll} \forall y \in \mathbb{Z}^3 & \exists x \in \mathbb{Z}^3, S'_q(x, y) \in \bar{q}\mathcal{C}(\mathbf{0}) \\ \forall x \in \mathbb{Z}^3 & \exists y \in \mathbb{Z}^3, S'_q(x, y) \in \mathcal{C}(\mathbf{0})\bar{q}, \end{array} \right. \quad (50)$$

provided that both sets $S'_q(\mathbb{Z}^3, \mathbb{Z}^3) \cap \bar{q}\mathcal{C}(\mathbf{0})$ and $S'_q(\mathbb{Z}^3, \mathbb{Z}^3) \cap \mathcal{C}(\mathbf{0})\bar{q}$ coincide.

4.3.2.4 Bijectivity certification

We now present an idea for verifying whether a digitized rational rotation given by a Lipschitz quaternion is bijective or not. The strategy consists of checking whether there exists $w \in ((\bar{q}\mathcal{C}(\mathbf{0}) \cup \mathcal{C}(\mathbf{0})\bar{q}) \setminus (\bar{q}\mathcal{C}(\mathbf{0}) \cap \mathcal{C}(\mathbf{0})\bar{q})) \cap \mathbb{Z}^4$ such that $w = S'_q(x, y)$. If this is the case, then the rotation given by q is not bijective, and conversely.

Because q is a Lipschitz quaternion, the values of S'_q span a sublattice $\check{\mathcal{G}} \subset \mathbb{Z}^4$. Therefore, given a Lipschitz quaternion $q = a + bi + cj + dk$, solving $S'_q(x, y) = w$ with $x, y \in \mathbb{Z}^3$ for $w \in \check{\mathcal{G}}$ leads to solving the following linear Diophantine system:

$$Az = w \quad (51)$$

where $z^t = (x, y) \in \mathbb{Z}^6$ and

$$A = \begin{bmatrix} b & c & d & -b & -c & -d \\ a & -d & c & -a & -d & c \\ d & a & -b & d & -a & -b \\ -c & b & a & -c & b & -a \end{bmatrix}.$$

The minimal basis $(\check{\sigma}, \check{\phi}, \check{\psi})$ of $\check{\mathcal{G}}$ can be obtained from the columns of the Hermite normal form of the matrix A . Since the rank of A is 3, we have $\check{\mathcal{G}} = \mathbb{Z}\check{\sigma} + \mathbb{Z}\check{\phi} + \mathbb{Z}\check{\psi}$.

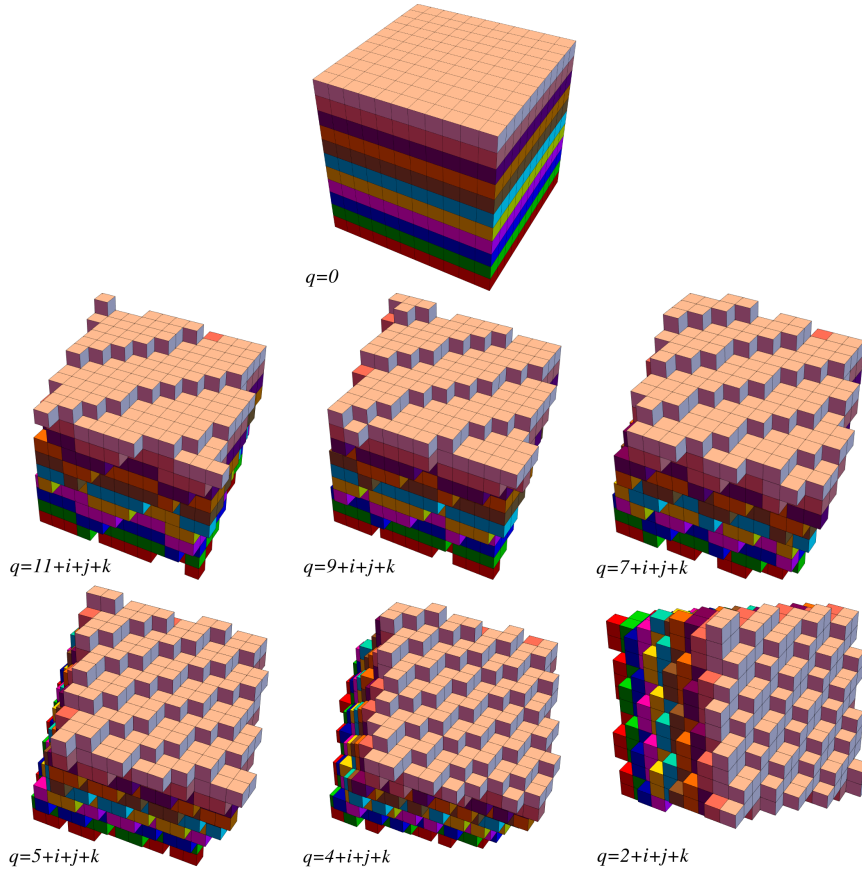


Figure 68: Examples of 3D bijective digitized rotations, represented by Lipschitz quaternions q , of a digitized cube for $q = 0$.

Therefore, the problem amounts to: (i) finding the minimal basis $(\check{\sigma}, \check{\phi}, \check{\psi})$ of the group $\check{\mathcal{G}}$ by reducing the matrix A to its Hermite normal form; (ii) checking whether there exists a linear combination of these basis vectors $w = u\check{\sigma} + v\check{\phi} + w\check{\psi}$, for $u, v, w \in \mathbb{Z}$ such that $w \in (\bar{q}\mathcal{C}(\mathbf{0}) \cup \mathcal{C}(\mathbf{0})\bar{q}) \setminus (\bar{q}\mathcal{C}(\mathbf{0}) \cap \mathcal{C}(\mathbf{0})\bar{q})$. For further details of the algorithm, see [150].

This method allows us to obtain some Lipschitz quaternions q that generate non-simple 3D bijective digitized rotations⁴, as illustrated in Fig. 68.

4.4 TOPOLOGY PRESERVATION

Similarly to the bijection, the topology is not always preserved under rigid motions on \mathbb{Z}^n (see Figure 51). This problem is studied in \mathbb{Z}^2 ; the class of two-dimensional images that preserve their topological properties during displacements – called regular images – is identified, as well as methods allowing such “regularization” [138] (see Section

⁴ A complete list of Lipschitz quaternions in the range $[-10, 10]^4$, inducing bijective 3D digitized rotations can be downloaded from: http://perso.esiee.fr/~plutak/download/lipschitz_quaternion_bijective_-10_10.csv

4.4.2). Nevertheless, these strategies in \mathbb{Z}^2 rely on topological models that are not intrinsically transposable to higher dimensions [148]; we show some counterexamples in Section 4.4.3. Before attacking these topological issues, we first give the framework of well-composed images [114], in which our study is developed (see Section 4.4.1).

4.4.1 Digital topology and well-composed images

Digital topology [105] provides a simple framework for handling the topology of binary images in \mathbb{Z}^n . It is also compliant [120] with other discrete models (e.g., Khalimsky grids [96] and cubical complexes [108]) but also with continuous notions of topology [122].

Practically, digital topology relies on two adjacency relations, noted \curvearrowright_{2n} and \curvearrowright_{3^n-1} , defined by

$$(p \curvearrowright_{2n} q) \iff (\|p - q\|_1 = 1) \quad (52)$$

$$(p \curvearrowright_{3^n-1} q) \iff (\|p - q\|_\infty = 1) \quad (53)$$

for any $p, q \in \mathbb{Z}^n$. In the case of \mathbb{Z}^2 , we retrieve the well-known 4- and 8-adjacency relations.

Let $\Omega \subseteq \mathbb{Z}^2$. We say that $p, q \in \mathbb{Z}^2$ are 4- (resp. 8-) adjacent, if $p \curvearrowright_4 q$ (resp. $p \curvearrowright_8 q$). From the reflexive-transitive closure of \curvearrowright_4 (resp. \curvearrowright_8) on Ω , we derive the 4- (resp. 8-) connectedness relation \sim_4 (resp. \sim_8) on Ω ; we say that p, q are 4- (resp. 8-) connected in Ω , if $p \sim_4 q$ (resp. $p \sim_8 q$). It is plain that \sim_4 (resp. \sim_8) is an equivalence relation on Ω ; the equivalence classes Ω/\sim_4 (resp. Ω/\sim_8) are called the 4- (resp. 8-) connected components of Ω .

Let \mathcal{GM} be the set of all finite binary images. A finite set $\Omega \subset \mathbb{Z}^2$ can be modeled as a binary image $I \in \mathcal{GM}$, defined by $I^{-1}(\{1\}) = \Omega$ and $I^{-1}(\{0\}) = \bar{\Omega} = \mathbb{Z}^2 \setminus \Omega$, or vice versa. The topological handling of I cannot easily rely on a single adjacency relation for both Ω and $\bar{\Omega}$, due to paradoxes related to the discrete version of the Jordan theorem [119]. Such paradoxes are avoided by considering distinct adjacencies for Ω and $\bar{\Omega}$, leading to the dual adjacency model [164] (Fig. 69(e-g)).

Definition 4.3 (Dual adjacency [164]). *Let $I \in \mathcal{GM}$. Let $\Omega = I^{-1}(\{1\})$ and $\bar{\Omega} = I^{-1}(\{0\})$. We say that I is a (8,4)- (resp. a (4,8)-) image if Ω is equipped with \curvearrowright_8 (resp. \curvearrowright_4), while $\bar{\Omega}$ is equipped with \curvearrowright_4 (resp. \curvearrowright_8). We define the set of the connected components of the (8,4)- (resp. (4,8)-) image I as*

$$\begin{aligned} \mathcal{C}^{(8,4)}[I] &= I^{-1}(\{1\})/\sim_8 \cup I^{-1}(\{0\})/\sim_4 \\ (\text{resp. } \mathcal{C}^{(4,8)}[I] &= I^{-1}(\{1\})/\sim_4 \cup I^{-1}(\{0\})/\sim_8) \end{aligned}$$

For the sake of concision, we will often write (k, \bar{k}) as a unified notation for (8,4) and (4,8).

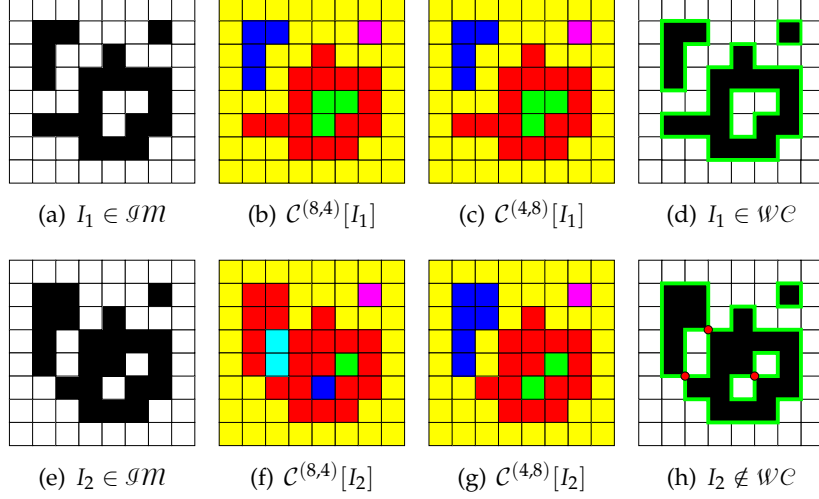


Figure 69: (a) A binary image $I_1 \in \mathcal{IM}$. (b) If we consider I_1 as a $(8,4)$ -image, the 8-connected components of $\Omega = I_1^{-1}(\{1\})$ are depicted in blue, purple and red, while the 4-connected components of $\bar{\Omega} = I_1^{-1}(\{0\})$ are depicted in yellow and green. (c) If we consider I_1 as a $(4,8)$ -image, the 4-connected components of Ω are depicted in blue, purple and red, while the 8-connected components of $\bar{\Omega}$ are depicted in yellow and green. Note that since I_1 has the same topological structure as a $(8,4)$ - and as a $(4,8)$ -image, it can also be considered in the well-composedness model: the boundaries shared by its foreground and background regions, depicted in green (d), are 1-manifolds. (e) A binary image $I_2 \in \mathcal{IM}$. (f) If we consider I_2 as a $(8,4)$ -image, the 8-connected components of $\Omega = I_2^{-1}(\{1\})$ are depicted in purple and red, while the 4-connected components of $\bar{\Omega} = I_2^{-1}(\{0\})$ are depicted in yellow, cyan, blue and green. (g) If we consider I_2 as a $(4,8)$ -image, the 4-connected components of $\Omega = I_2^{-1}(\{1\})$ are depicted in blue, purple and red, while the 8-connected components of $\bar{\Omega} = I_2^{-1}(\{0\})$ are depicted in yellow and green. Note that I_2 does not have the same topological structure as a $(8,4)$ - and as a $(4,8)$ -image. Thus, I_2 is ill-composed: the boundaries shared by its foreground and background regions, depicted in green, are not 1-manifolds (see the red dots in (h)). (a,d,e,h) Ω is depicted in black, and $\bar{\Omega}$ in white. (b,c,f,g) For the sake of readability, each connected component is represented in a different colour.

Alternatively, both Ω and $\overline{\Omega}$ may be equipped with \sim_4 , provided one considers only images that avoid the issues related to the Jordan theorem, i.e. those for which \sim_4 and \sim_8 are equivalent for both Ω and $\overline{\Omega}$, thus leading to the well-composedness model [114] (Fig. 69(a-c)).

Definition 4.4 (Well-composedness [114]). *Let $I \in \mathcal{IM}$. We say that I is a well-composed (or a wc-) image if*

$$\forall v \in \{0, 1\}, I^{-1}(\{v\}) / \sim_8 = I^{-1}(\{v\}) / \sim_4$$

We define the set of the connected components of the wc-image I as

$$\mathcal{C}^{wc}[I] = I^{-1}(\{1\}) / \sim_4 \cup I^{-1}(\{0\}) / \sim_4$$

The set of the finite well-composed binary images is noted \mathcal{WC} .

Remark 4.1. *When interpreting digital topology in a continuous framework [120], an image is well-composed iff the boundaries shared by the foreground and background regions are manifolds [114] (Fig. 69(d,h)).*

Remark 4.2. *The well-composedness model is more restrictive than dual adjacency. Indeed, any $I \in \mathcal{IM}$ can be considered in the dual adjacency model, but not necessarily in the well-composedness one, i.e.*

$$\mathcal{WC} \subset \mathcal{IM}$$

4.4.2 Topological invariance under 2D digitized rigid motions

Given a binary image $I \in \mathcal{IM}$, a transformation $\mathcal{T} : \mathbb{Z}^2 \rightarrow \mathbb{Z}^2$, and the transformed image $I_{\mathcal{T}} \in \mathcal{IM}$ obtained from I and \mathcal{T} , a frequent question in image analysis is: “Does \mathcal{T} preserve the topology between I and $I_{\mathcal{T}}$?”. It is generally answered by observing the topological invariants of these images.

Note that we consider here the backwards transformation model such that

$$\mathcal{T} := \mathfrak{D} \circ (\mathfrak{T}^{-1})|_{\mathbb{Z}^2} \quad (54)$$

rather than (36), so that \mathcal{T} is surjective. It means that the transformed image $I_{\mathcal{T}}$, defined by

$$I_{\mathcal{T}} = I \circ \mathfrak{D} \circ (\mathfrak{T}^{-1})|_{\mathbb{Z}^2} = I \circ \mathcal{T},$$

has no pixel that has either no or double values.

Among the simplest topological invariants are the Euler-Poincaré characteristic and the Betti numbers. However, these are too weak to accurately model “topology preservation” between images [119]. It is necessary to consider stronger topological invariants, e.g., the (digital) fundamental group [104], the homotopy-type (considered via notions of simple points / sets [14], [18], [146], [160]), or the adjacency tree [163].

Our first goal is to provide conditions under which 2D digital images preserve their topological properties under arbitrary rigid transformations. A crucial issue is the choice of the topological invariant used to formalize this problem. Any of those evoked above describe topology preservation in a *global* fashion, and do not model accurately the possible *local* modifications of the image topological structure. Indeed, I and $I_{\mathcal{T}}$ may have identical fundamental group, homotopy-type or adjacency tree while still retaining some topological differences between regions of I and $I_{\mathcal{T}}$ that are in correspondence with respect to \mathcal{T} . (A classical example that illustrates this assertion is the “scorpion” configuration illustrated, e.g., in [17], where the removal of a point from a 3D object removes a tunnel while simultaneously creating another, thus producing a new object with the same global topological invariants. However this procedure changes the local topological structure in the neighbourhood of these two tunnels.)

In the sequel, we propose some conditions to achieve this first goal. Our conjecture is that these conditions are *necessary and sufficient* to *locally* preserve image topological properties under arbitrary rigid transformations. However, in this section, we only establish that they are *sufficient* to *globally* preserve image topological properties under any rigid transformation.

We consider the adjacency tree [163] as a (global) topological invariant. The motivation of this choice is twofold: (i) this topological invariant is probably easier to understand than others; and (ii) in the 2D case, its preservation is equivalent [165] to the preservation of the homotopy-type, that is the most commonly used topological invariant in image processing. We now recall the definition of the adjacency tree.

Let $I \in \mathcal{IM}$ (resp. \mathcal{WC}). Let $\Omega_1, \Omega_2 \in \mathcal{C}^{(k, \bar{k})}[I]$ (resp. $\mathcal{C}^{wc}[I]$), with $\Omega_1 \neq \Omega_2$. We note $\Omega_1 \curvearrowright_I^{(k, \bar{k})} \Omega_2$ (resp. $\Omega_1 \curvearrowright_I^{wc} \Omega_2$) if there exist $p \in \Omega_1$ and $q \in \Omega_2$ such that $p \curvearrowright_4 q$. It is plain that $\curvearrowright_I^{(k, \bar{k})}$ (resp. \curvearrowright_I^{wc}) is an adjacency relation, and that $\Omega_1 \curvearrowright_I^{(k, \bar{k})} \Omega_2$ implies that $\Omega_1 \in I^{-1}(\{1\})/\sim_k$ and $\Omega_2 \in I^{-1}(\{0\})/\sim_{\bar{k}}$ or vice versa. We define the (k, \bar{k}) - (resp. *wc*-) *adjacency graph* of I as $\mathcal{G}^{(k, \bar{k})}(I) = (\mathcal{C}^{(k, \bar{k})}[I], \curvearrowright_I^{(k, \bar{k})})$ (resp. $\mathcal{G}^{wc}(I) = (\mathcal{C}^{wc}[I], \curvearrowright_I^{wc})$). This graph is connected and acyclic, and is indeed a tree. It can be equipped with a root that is the (only) infinite connected component of $\mathcal{C}^{(k, \bar{k})}[I]$ (resp. $\mathcal{C}^{wc}[I]$), thus leading to the following definition.

Definition 4.5 (Adjacency tree [163]). *Let $I \in \mathcal{IM}$ (resp. \mathcal{WC}). The (k, \bar{k}) - (resp. *wc*-) adjacency tree of I is the triplet*

$$\begin{aligned} \mathcal{T}^{(k, \bar{k})}(I) &= (\mathcal{C}^{(k, \bar{k})}[I], \curvearrowright_I^{(k, \bar{k})}, B_I^{(k, \bar{k})}) \\ (\text{resp. } \mathcal{T}^{wc}(I) &= (\mathcal{C}^{wc}[I], \curvearrowright_I^{wc}, B_I^{wc})) \end{aligned} \quad (55)$$

where $B_I^{(k, \bar{k})} \in \mathcal{C}^{(k, \bar{k})}[I]$ (resp. $B_I^{wc} \in \mathcal{C}^{wc}[I]$) is the unique infinite connected component of I .

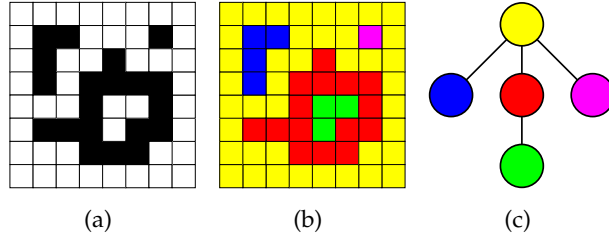


Figure 70: (a) The binary image of Fig. 69(a). (b) The connected components of the background (in yellow and green) and foreground (in red, blue and purple) of (a). (c) The adjacency tree associated to (a) in which each coloured node corresponds to a connected component of (a,b), while each edge corresponds to an adjacency link between two components. The root of the tree is the yellow node, that corresponds to the infinite background component in (a,b).

An adjacency tree example is given in Fig. 70.

We are now ready to present our definition of topology preservation under rigid transformation.

Definition 4.6 (Topological invariance [138]). *Let $I \in \mathcal{IM}$ (resp. \mathcal{WC}). We say that I is (k, \bar{k}) - (resp. wc -) topologically invariant if $I \circ \mathcal{T} \in \mathcal{IM}$ (resp. \mathcal{WC}) and if any digitized rigid motion \mathcal{T} induces an isomorphism between $\mathcal{T}^{(k, \bar{k})}(I)$ (resp. $\mathcal{T}^{wc}(I)$) and $\mathcal{T}^{(k, \bar{k})}(I \circ T)$ (resp. $\mathcal{T}^{wc}(I \circ T)$). We note $\mathcal{INV}^{(k, \bar{k})}$ (resp. \mathcal{INV}^{wc}) the set of all the (k, \bar{k}) - (resp. wc -) topologically invariant binary images.*

4.4.2.1 Image classes and topological invariance

We first introduce a notion of *singularity*, and we establish that *singular* images cannot be topologically invariant, thus reducing the image subspace to consider.

Definition 4.7 ((Non-)singular image). *Let $I \in \mathcal{IM}$. We say that I is a singular image if*

$$\exists p \in \mathbb{Z}^2, \forall q \in \mathbb{Z}^2, (q \curvearrowright_4 p) \implies (I(p) \neq I(q)) \quad (56)$$

otherwise I is non-singular. We note \mathcal{NS} the set of the well-composed images that are non-singular.

Examples of (non-)singular images are given in Fig. 71. The non-topological invariance of singular images is derived from the non-surjectivity of some digitized rigid transformations [132], [142]. Indeed some such transforms may remove connected components composed of exactly one pixel. More precisely, we have the following proposition.

Proposition 4.8.

$$\mathcal{INV}^{wc} \subseteq (\mathcal{INV}^{(k, \bar{k})} \cap \mathcal{WC}) \subseteq \mathcal{NS}$$

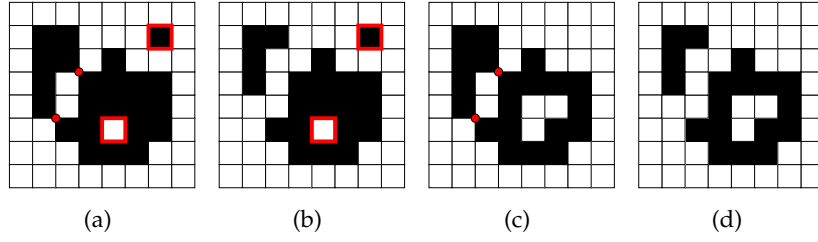


Figure 71: Examples of images being (a) singular and ill-composed, (b) singular and well-composed, (c) neither singular nor well-composed, and (d) not singular but well-composed. Red dots identify ill-composedness, while red boundaries identify singularity.

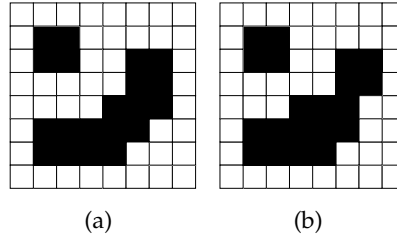


Figure 72: (a) A digitally regular image. (b) An image that is neither 1- nor 0-regular, but that is however opened by a structuring element \boxplus , both for black and white points (see Remark 4.3).

The study of topological invariance is then carried out within the set of well-composed non-singular images, independently from the considered (dual adjacency or well-composedness) model.

Let us now introduce a new notion that strengthens the notion of well-composedness.

Definition 4.8 (Digital image regularity [138]). Let $I \in \mathcal{NS}$. Let $v \in \{0, 1\}$. We say that I is v -regular if for any $p, q \in I^{-1}(\{v\})$, we have

$$(p \curvearrowright_4 q) \implies (\exists \boxplus \subseteq I^{-1}(\{v\}), p, q \in \boxplus)$$

where $\boxplus = \{x, x+1\} \times \{y, y+1\}$, for $(x, y) \in \mathbb{Z}^2$. We say that I is regular if it is both 0- and 1-regular. We note \mathcal{REG}^1 (resp. \mathcal{REG}^0 , resp. \mathcal{REG}) the set of all the 1-regular (resp. 0-regular, resp. regular) binary images.

An example of a regular binary image is given in Fig. 72(a).

Remark 4.3. Following mathematical morphology terminology [130, Ch. 1], if I is 1- (resp. 0-) regular, then $\Omega = I^{-1}(\{1\})$ (resp. $I^{-1}(\{0\})$) is opened by any structuring element \boxplus , i.e.

$$\gamma_{\boxplus}(\Omega) = \Omega \ominus \boxplus \oplus \boxplus = \Omega$$

The converse is not true, as illustrated in Fig. 72(b).

We finally establish our main theoretical result that states that digital regularity implies topological invariance, for binary images.

Theorem 4.1 (Digital regularity condition [138]).

$$\begin{aligned}\mathcal{REG}^0 &\subseteq \mathcal{INV}^{(8,4)} \\ \mathcal{REG}^1 &\subseteq \mathcal{INV}^{(4,8)} \\ \mathcal{REG} &\subseteq \mathcal{INV}^{wc}\end{aligned}$$

This result is straightforwardly extended to grey-level and label images. See [138] for the proofs and further details.

4.4.2.2 Characterization of regular images

Regular images are necessarily well-composed. A prerequisite is then to characterise \mathcal{WC} . This is tractable by considering a specific 2×2 pattern [114] (see Figure 73 (a)).

Theorem 4.2 ([114]). *Let $I \in \mathcal{IM}$. We have $I \notin \mathcal{WC}$ iff there exist distinct points $p, q, r, s \in \mathbb{Z}^2$, with $p \frown_4 q \frown_4 r \frown_4 s \frown_4 p$, that verify*

$$I(p) \neq I(q) \neq I(r) \neq I(s)$$

We now propose a pattern-based characterization of regular binary images.

Proposition 4.9 ([138]). *Let $I \in \mathcal{WC}$. We have $I \notin \mathcal{REG}^1$ (resp. \mathcal{REG}^0) – and a fortiori \mathcal{REG} – iff there exists $p \in I^{-1}(\{1\})$ (resp. $I^{-1}(\{0\})$) that satisfies at least one of the following two conditions (up to $\pi/2$ rotations and symmetries)*

$$\begin{aligned}I(p - (1, 0)) &\neq I(p) \neq I(p + (1, 0)) \\ I(p + (0, 1)) &= I(p) \neq I(p - (1, 0)) = I(p + (1, 1))\end{aligned}$$

Indeed, the first condition is a rewriting of $I \notin \mathcal{NS}$, while the second condition is a rewriting of the negation of the equation in Definition 4.8.

The characterization of regular binary images can then be carried out by simply checking that they do not contain the forbidden patterns induced by Fig. 73.

Let us consider the four binary images depicted in Fig. 74(a), that are well-composed, but neither 1- nor 0-regular. The identified forbidden patterns corresponding to this default of regularity are illustrated in Fig. 74(b). Their presence indicates that the topological structure of these images is likely to be altered when applying a rigid transformation, as exemplified in Fig. 74(c–e).

This characterization can be straightforwardly extended to grey-level and label images [138].

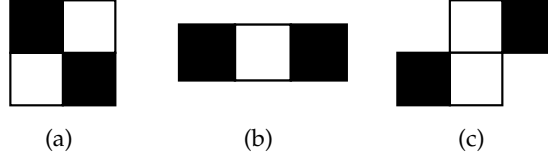


Figure 73: Forbidden patterns in \mathcal{WC} (a) and in \mathcal{REG}^0 (a–c), up to $\pi/2$ rotations and symmetries. The patterns forbidden in \mathcal{REG}^1 are obtained from (a–c) by value inversion. Black (resp. white) points have value 1 (resp. 0).

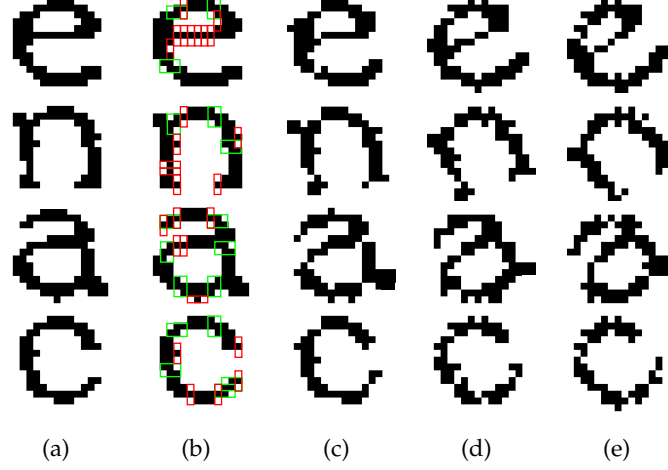


Figure 74: (a) Four well-composed, but neither 1- nor 0-regular images. Pixels of value 1 and 0 are depicted in black and white, respectively. (b) Patterns that forbid regularity. In red: patterns of Fig. 73(b); in green: patterns of Fig. 73(c). (c–e) Three examples of rigid transformations where the four images are topologically altered in comparison to (a). In particular, the black part of the images, that are 4-connected in (a), are split into several 4-connected components. Moreover, the 8-connected components forming the holes inside the “a” and “e” letters in (a) are merged to the background.

4.4.2.3 Image regularization

We now propose two strategies for preprocessing images in order to obtain regular – and thus topologically invariant – versions, before further rigid transformation. Even though both can be applied to grey-level and label images, we only show binary cases; see [138] for the other cases.

1) Iterative homotopic regularization: The strategy starts from the image I , which is defined on a finite set $S \in \mathbb{Z}^2$ in real applications, and iteratively eliminates forbidden configurations given in Theorem 4.2 and Proposition 4.9 by modifying the value of one point $p \in S$ at each iteration, until stability. This value modification can be interpreted either as a background-to-foreground or a foreground-to-background sweep. The choice of p is guided (i) by a cost function

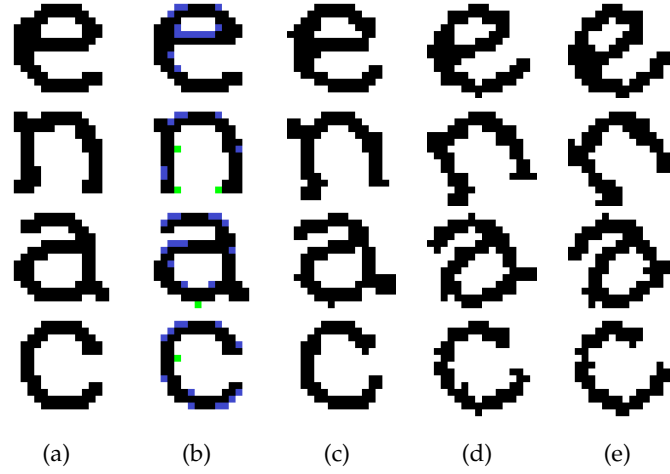


Figure 75: (a) Regular images obtained from Fig. 74(a) after iterative homotopic regularization. (b) Difference between (a) and Fig. 74(a). In blue: pixels switched from white to black; in green: pixels switched from black to white. (c–e) Three examples of rigid transformations where the four images are topologically preserved in comparison to (a). The transformation parameters are the same as in Fig. 74(c–e).

that describes a distance with respect to I , e.g., by following a gradient descent approach, and (ii) by choosing p as a simple point so that the modification preserves the topological properties [41].

For example, starting from the four images of Fig. 74(a), this strategy swaps the value of simple points until a regular image with the same homotopy-type, and sufficient geometric similarity is obtained. In the results illustrated in Fig. 75(a,b), the number of modified pixels is 14, 10, 15 and 14, for each image, respectively, and the Hausdorff distance between the initial and regularized images is 1 in each case. It can be observed in Fig. 75(c–e), that for rigid transformations with the same parameters as those of Fig. 74(c–e), the obtained results now have the same topological structure as in Fig. 74(a).

The obtained algorithm can be seen as an extension of those presented in [165] for well-composedness recovery, to the case of regularity recovery. In particular, it presents the same strengths and weaknesses. Indeed, in most application cases, it will converge in linear time with respect to the number of forbidden configurations, that are often sparsely distributed within images. Nevertheless, in the worst cases (e.g., in presence of fine textures, Figure 76), it may not converge, or even fail. To deal with this issue, we propose an alternative up-sampling regularization strategy.

2) Upsampling regularisation: Even before the issue of regularization, it may happen that I cannot be modified into a topologically-equivalent well-composed image, when using a strategy such as presented above. It is then possible to oversample I by explicitly rep-

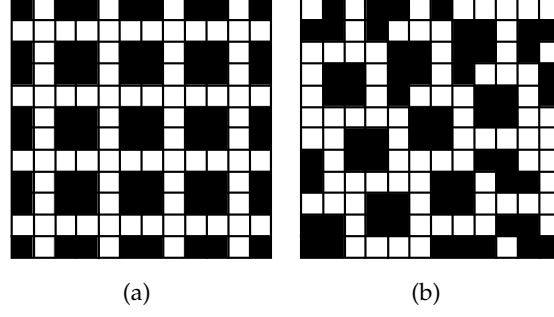


Figure 76: (a) A well-composed binary image that is not 0-regular. Pixels of value 1 and 0 are depicted in black and white, respectively. This image cannot be regularized without up-sampling, due to fine texture effects. (b) An image obtained from (a) after a rigid transformation. It is topologically altered, in comparison to (a).

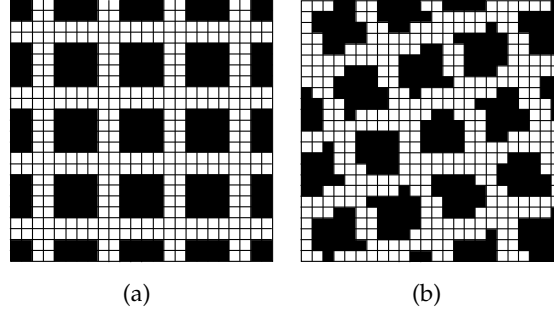


Figure 77: (a) A regular image obtained from Fig. 76(a) after the up-sampling regularization. (b) An image obtained from (a) after a rigid transformation. It is topologically preserved, by comparison to (a). The transformation parameters are the same as in Fig. 76(b).

representing its interpixel topological structure. This can be done by embedding I into the Khalimsky space [96], then leading to a new image $I_K^{(k,\bar{k})}$ defined as

$$\begin{array}{lcl}
 I_K^{(4,8)} : \mathbb{Z}^2 & \rightarrow & \{0,1\} \\
 2.\mathbf{p} & \mapsto & I(\mathbf{p}) \\
 2.\mathbf{p} + (0,1) & \mapsto & \bigvee^{\leq} I(\mathbf{p} + \{0\} \times \{0,1\}) \\
 2.\mathbf{p} + (1,0) & \mapsto & \bigvee^{\leq} I(\mathbf{p} + \{0,1\} \times \{0\}) \\
 2.\mathbf{p} + (1,1) & \mapsto & \bigvee^{\leq} I(\mathbf{p} + \{0,1\} \times \{0,1\})
 \end{array} \quad (57)$$

(The image $I_K^{(8,4)}$ is defined by substituting \bigwedge to \bigvee in Eq. (57).) The following result straightforwardly derives from these definitions.

Proposition 4.10. *Let $I \in \mathcal{IM}$. We have $I_K^{(k,\bar{k})} \in \mathcal{WC}$. Moreover, $I_K^{(k,\bar{k})}$ and image I have the same homotopy-type, when considered as (k,\bar{k}) -images.*

From now on, we then assume that $I \in \mathcal{WC}$. As stated before, even in this case, the image I may still not be modified into a regular

image when using homotopic iterative regularization. Once again, an oversampling strategy can be alternatively proposed. This strategy no longer relies on Khalimsky space embedding, but on a 2×2 up-sampling approach. More precisely, from $I \in \mathcal{WC}$, we can define a new image

$$\left| \begin{array}{lll} I_{2 \times 2} : \mathbb{Z}^2 & \rightarrow & \{0, 1\} \\ p = (x, y) & \mapsto & I(\lfloor x/2 \rfloor, \lfloor y/2 \rfloor) \end{array} \right. \quad (58)$$

The following result straightforwardly derives from this definition.

Proposition 4.11. *Let $I \in \mathcal{WC}$. We have $I_{2 \times 2} \in \mathcal{REG}$. Moreover, $I_{2 \times 2}$ and image I have the same homotopy-type when considered as (k, \bar{k}) - (resp. wc -) images.*

Finally, Eqs. (57)–(58) provide a global up-sampling strategy that enables to re-cast any $(8, 4)$ -, $(4, 8)$ -, or wc -image as regular, and thus topologically invariant. This strategy has the advantages of being deterministic and geometrically preserving (up to the thickening of the interpixel space due to the up-sampling process). Its main drawback, in comparison to the first strategy, is its higher spatial cost, as it models an image of size $|S|$ as a new one of size $4 \cdot |S|$ (and $16 \cdot |S|$ in the worst cases). This may remain however acceptable for many applications, considering the memory specifications and progress of current computers.

Let us consider the three images depicted in Fig. 76(a), that are well-composed, but not regular, with topological consequences when applying rigid transformations, as illustrated in Fig. 76(b). For such images, the iterative homotopic regularisation may not converge, or even fail, due to fine texture effects. It is then relevant to consider the second, up-sampling regularization strategy. Since the three considered images are already well-composed, it is not necessary to carry out the first step of the regularization, namely the Khalimsky grid embedding (Eq. (57)). After the application of the second step, namely the 2×2 up-sampling approach (Eq. (58)), we obtain new images, depicted in Fig. 77 (a), that are regular, and therefore topologically invariant, as illustrated in Fig. 77 (b).

4.4.3 Topological alterations under 3D digitized rigid motions

The straightforward extension of the above 2D approach to 3D is considered first of all; digital image regularity in Definition 4.8 can be extended for 3D images by replacing the 2×2 square \boxplus by the $2 \times 2 \times 2$ cube. Then, we quickly find some counterexamples of the 3D version of Theorem 4.1 [148]: for example, see Figure 78. Figure 79 also shows a position of a 6-neighborhood after a rigid motion, which generates such a topological alteration after the motion; only the center point is in the object while none of its 6-adjacent points

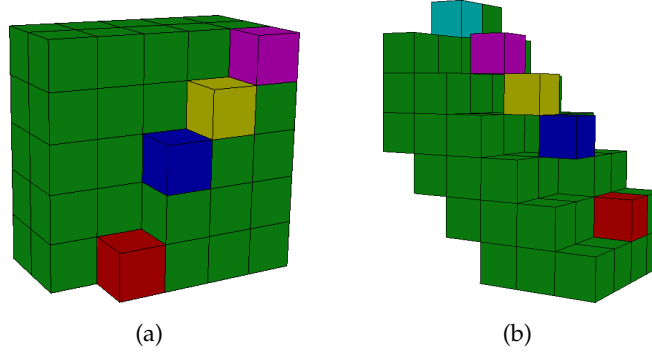


Figure 78: (a) A part of digitally regular image containing a flat shape, which is 6-connected for both the foreground and background, and (b) its rigidly transformed image, which is not anymore 6-connected for the foreground.

are in the object. Note that the initial object satisfies the digitally regularity.

This concludes that a simple extension of the 2D approach of image regularization is not useful for more than two dimensions. Consequently, it will be essential to develop new strategies more generic and thus valid in any dimension.

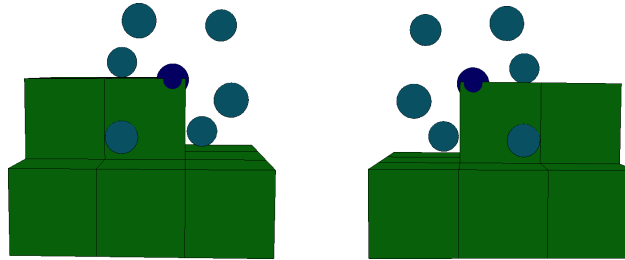


Figure 79: A position of a 6-neighborhood after a rigid motion, which can generate a topological alteration after the motion; only the center point is in the object while none of its 6-adjacent points are in the object.

4.5 GEOMETRY PRESERVATION

In this section, we focus on geometric issues under rigid transformation on \mathbb{Z}^2 . Indeed, digital image regularity, which preserves connectivity under any digitized rigid motions, as seen previously, does not preserve geometry simultaneously. See Figure 51 (b) for an example: the binary image containing a digitized half-plane is regular (Figure 51 (b) left) although its digitized rigid motion does not always preserve the linear separability (right). In order to overcome this problem, we propose a rigid motion scheme that preserves geometric properties as well as topology of transformed digital objects: a con-

nected object will remain connected, and some geometric properties (e.g. convexity, area and perimeter) will be preserved. This section is based on the work of [137].

4.5.1 Digitized rigid motions via polygonization

To reach that goal, we propose to represent a digital object of \mathbb{Z}^2 as a continuous –but discrete– object, namely a polygon of \mathbb{R}^2 . This strategy has several advantages. First, it allows us to apply the rigid motion in \mathbb{R}^2 , with the geometric and topological guarantees within this space. Second, since a polygon remains a discrete object, it can be processed without numerical error, by considering transformations based on integers (or, equivalently, rationals).

In this context, our assumption is that the polygon has to relevantly capture the geometry of the digital object. In particular, this means that the Gauss digitization of the polygon has to get us back to the initial digital object; this property is called reversibility. In other words, the global shape of the digital object, namely the succession of the convex and concave parts of its boundary, has to be captured by the polygonization process. In particular, this means that a digitally convex object of \mathbb{Z}^2 will lead to a convex polygon. In that case, we will choose as relevant polygon model its convex hull. In the other cases, the polygon will depend on the user's polygonization policy.

Based on these hypotheses, we propose, as a first contribution, an algorithmic framework for rigid motion of digital objects of \mathbb{Z}^2 . It relies on three successive steps, as illustrated in Figure 80: (1) the polygonization of the digital object; (2) the transformation of the intermediate piecewise affine object (polygon) of \mathbb{R}^2 ; and (3) the digitization of the transformed polygon for recovering a result within \mathbb{Z}^2 . In the case of an initial object being digitally convex, our framework is proved to provide a final digital object which is also digitally convex (see Section 4.5.3). In the other cases, it is experimentally observed that the shape of objects are correctly preserved (see Section 4.5.4). More precisely, such an observation can be done qualitatively and quantitatively in which geometric properties, for example area and perimeter, are measured.

Generally, preserving the geometry also implies to preserve the topology. This implication is mostly offered in \mathbb{R}^2 , while it is hardly obtained in \mathbb{Z}^2 . This is the motivation for our second contribution. Indeed, we propose a new notion of *quasi- (r, r') -regularity*, defined on continuous objects, including particular polygons (see Section 4.5.2.3). It provides sufficient conditions to be fulfilled by a continuous object for guaranteeing topology preservation during its digitization.

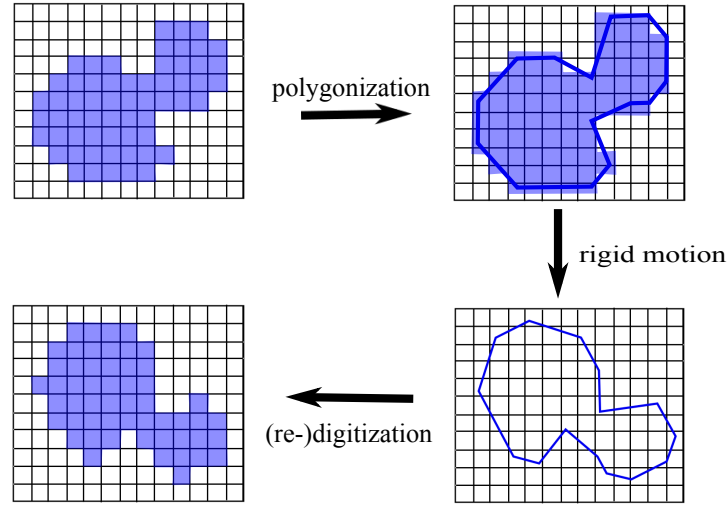


Figure 80: Rigid motion scheme of 2D digital shape via polygonization.

4.5.2 Digitization and topology preservation

As seen above, we have a re-digitization process after the rigid motion of a polygonal shape. We first explain the digitization scheme and the related topological issues.

4.5.2.1 Digitization model and topological issues

Let X be a continuous object in the Euclidean space \mathbb{R}^2 . (In the sequel, we will implicitly consider that X is bounded and connected.) A digital object $X \subset \mathbb{Z}^2$ is generally the result of a digitization process applied on a continuous object $X \subset \mathbb{R}^2$. We consider the Gauss digitization [101], which is simply the intersection of a continuous object X with \mathbb{Z}^2 :

$$X = X \cap \mathbb{Z}^2. \quad (59)$$

The object X is a subset of \mathbb{Z}^2 ; but from an imaging point of view, it can also be seen as a subset of pixels, i.e. unit squares defined as the Voronoi cells of the points of X within \mathbb{R}^2 . Based on these different models, the structure of X can be defined in various topological frameworks which are mainly equivalent [121] to that of digital topology [105]. However, this digital topology of X is often non-coherent with the continuous topology of X . This fact is illustrated in Figure 81, where a connected continuous object X leads, after Gauss digitization, to a disconnected digital object X .

4.5.2.2 Regular sets and their digitization

In the literature, various studies proposed conditions for guaranteeing the preservation of topology of digitized objects [113], [172], [184]. In particular, in [147] Pavlidis introduced the notion of r -regularity.

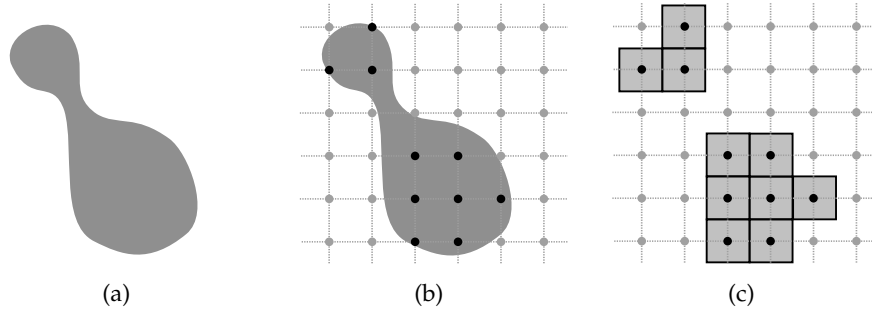


Figure 81: (a) A continuous object X in \mathbb{R}^2 . (b) A Gauss digitization of X , leading to the definition of X which is composed by the black points of \mathbb{Z}^2 within X . (c) The digital object X represented as a set of pixels. The objects X and X are not topologically equivalent: the digitization process led to a disconnection, due to the resolution of the discrete grid, not fine enough for catching the shape of X .

Definition 4.9 (r -regularity [147]). *An object $X \subset \mathbb{R}^2$ is r -regular if for each boundary point of X , there exist two tangent open disks of radius r , lying entirely in X and its complement \bar{X} , respectively*

The notion of r -regularity is based on classical concepts of differential geometry. In particular, r -regularity is strongly related to bounded values of curvature, parameterized by the resolution of the digitization sampling. Pavlidis proved the topological equivalence of an r -regular continuous, smooth, object X and its digital counterpart X , for a dense sampling.

Proposition 4.12 ([147]). *An r -regular object $X \subset \mathbb{R}^2$ has the same topological structure as its digitized version $X = X \cap \mathbb{Z}^2$ if $r \geq \frac{\sqrt{2}}{2}$.*

Remark 4.4. In [147], “the same topological structure” between two objects means that there exists an homeomorphism between both. In the sequel, we will consider the same paradigm. However, it is worth mentioning that in the 2D case and for digital objects whose continuous analogues have a manifold boundary (this will be our case with well-composed objects, see below), most topological invariants are indeed equivalent, namely homotopy type, adjacency tree and homeomorphism [22], [163], [165].

It was shown that the digitization process of an r -regular object yields a well-composed object [113], whose definition relies on standard concepts of digital topology, as shown in Section 4.4.1. Here, we consider the notion of well-composedness [114] focusing on a subset of \mathbb{Z}^2 , and characterize such digital objects whose structure intrinsically avoids the topological issues of a digital version of the Jordan curve theorem.

Definition 4.10 (Well-composed sets [114]). *A digital object $X \subset \mathbb{Z}^2$ is well-composed if each 8-connected component of X and of its complement \bar{X} is also 4-connected.*

This definition implies that the boundary⁵ of X is a set of 1-manifolds whenever X is well-composed (see Figure 82). In particular, there exists a strong link between r -regularity and well-composedness.

Proposition 4.13 ([113]). *If an object $X \subset \mathbb{R}^2$ is r -regular, with $r \geq \frac{\sqrt{2}}{2}$, then $X = X \cap \mathbb{Z}^2$ is a well-composed digital object.*

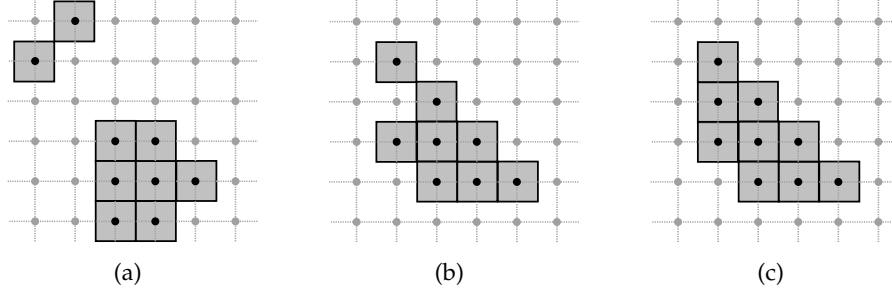


Figure 82: (a) $X \subset \mathbb{Z}^2$ (in grey) is neither connected, nor well-composed. (b) X is 8-connected, but neither 4-connected nor well-composed. (c) X is 4-connected and well-composed.

4.5.2.3 Quasi-regular sets and their digitization

Clearly, polygons are not r -regular as their boundaries are not differentiable at the polygonal vertices. In order to deal with polygons, we generalize the concept of r -regularity for non-differentiable shapes.

For this purpose, let us first recall some notations and a few mathematical morphology notions. [74], [161], [172]. We denote by \oplus and \ominus the classical operators of dilation and erosion, corresponding to the Minkowski addition, and its associated subtraction

$$X \oplus Y = \bigcup_{y \in Y} X_y = \bigcup_{x \in X} Y_x \quad (60)$$

$$X \ominus Y = \bigcap_{y \in Y} X_{-y} \quad (61)$$

where $X_y = \{x + y \mid x \in X\}$ and, in our case, $X, Y \subset \mathbb{R}^2$. We also denote by \circ the composition of erosion and dilation, called opening, that is

$$X \circ Y = (X \ominus Y) \oplus Y. \quad (62)$$

We denote by B_r a close disk of \mathbb{R}^2 of radius $r > 0$ and centered on $(0, 0) \in \mathbb{R}^2$. Then, we can rewrite Definition 4.9 for a bounded, simply connected set $X \subset \mathbb{R}^2$ as follows: X is r -regular if:

- $X \ominus B_r$ is non-empty and connected;

⁵ The boundary of X is defined here as the boundary of the continuous object obtained as the union of the closed Voronoi cells associated to the points of X , in \mathbb{R}^2 .

- $\overline{X} \ominus B_r$ is connected;
- $X = X \ominus B_r \oplus B_r = X \circ B_r$; and
- $\overline{X} = \overline{X} \ominus B_r \oplus B_r = \overline{X} \circ B_r$.

We are now ready to introduce the notion of *quasi- (r, r') -regularity*. Intuitively, a quasi- (r, r') -regular object X of \mathbb{R}^2 presents sufficient conditions for guaranteeing that its connectedness will not be affected by a Gauss digitization process.

Definition 4.11 (Quasi- (r, r') -regularity [137]). *Let $r' \geq r > 0$. Let $X \subset \mathbb{R}^2$ be a bounded, simply connected (i.e., connected and with no holes) set. We say that X is quasi- (r, r') -regular if it satisfies the following four properties:*

- $X \ominus B_r$ is non-empty and connected;
- $\overline{X} \ominus B_r$ is connected;
- $X \subseteq X \ominus B_r \oplus B_{r'}$; and
- $\overline{X} \subseteq \overline{X} \ominus B_r \oplus B_{r'}$;

Remark 4.5. *This definition does not require specific assumption on the boundary of X . In particular, it does not need to be differentiable.*

Remark 4.6. *Comparing the two notions of quasi- (r, r') -regularity and of Pavlidis' r -regularity, we observe that the principal difference between both notions is the fact that the matching between X (resp. \overline{X}) and its opening need to be perfect in the case of r -regularity, namely $r = r'$, while a “margin”, $r' - r \neq 0$, is authorized in the case of quasi- (r, r') -regularity, thus allowing for non-smooth (for instance, non-differentiable, noisy, etc.) boundary. Examples of quasi- $(1, \sqrt{2})$ -regular and non-quasi- $(1, \sqrt{2})$ -regular objects are given in Figure 83.*

Proposition 4.14 ([137]). *Let $X \subset \mathbb{R}^2$ be a bounded, simply connected (i.e., connected and with no holes) set. If X is quasi- $(1, \sqrt{2})$ -regular, then $X = X \cap \mathbb{Z}^2$ and $\overline{X} = \overline{X} \cap \mathbb{Z}^2$ are both 4-connected. In particular, X is then well-composed.*

This notion of quasi- (r, r') -regularity will be used in the following sections for guaranteeing the preservation of topological properties of digital shapes during rigid motions, via their (continuous) polygonal representation.

4.5.3 Convexity-preserving digitized rigid motions

Here we deal with a specific case of digital sets, namely the convex ones. For rigid motion purpose, we build a continuous polygon corresponding to the convex hull of the input digital set. Then, we

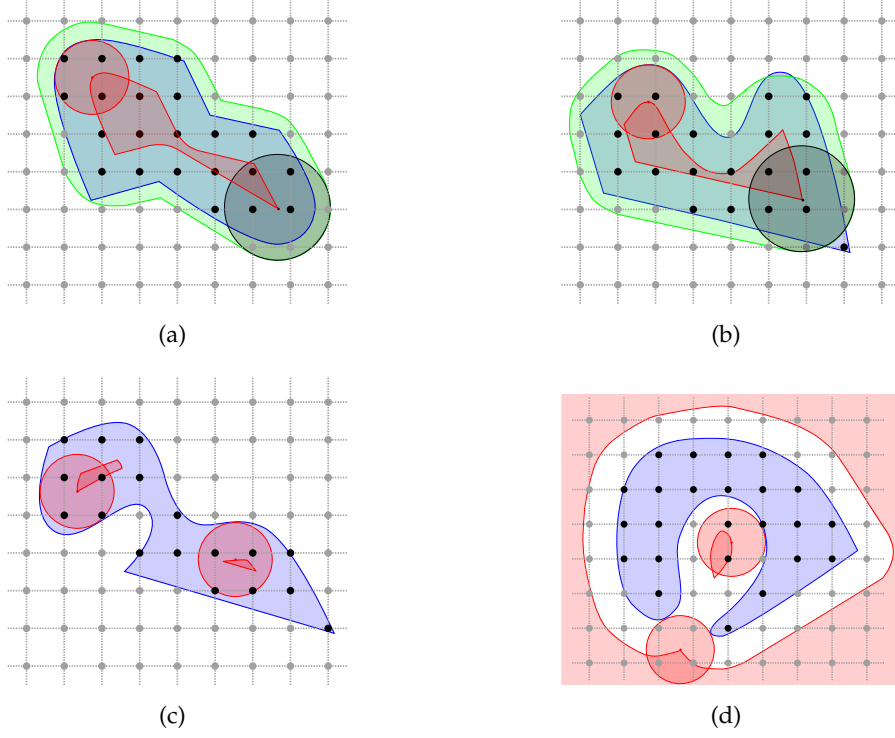


Figure 83: Examples of quasi- $(1, \sqrt{2})$ -regular (a) and non-quasi- $(1, \sqrt{2})$ -regular (b,c,d) sets X : (b) $X \not\subseteq X \ominus B_1 \oplus B_{\sqrt{2}}$; (c) $X \ominus B_1$ is not connected; (d) $\bar{X} \ominus B_1$ is not connected. The sets $X \subset \mathbb{R}^2$ are in blue, the disks B_1 are in red and the disks $B_{\sqrt{2}}$ are in black, the erosions $X \ominus B_1$ are in red and the openings $X \ominus B_1 \oplus B_{\sqrt{2}}$ are in green.

move this continuous polygon, and finally digitize it for retrieving the final transformed digital set. We show that, by this process, the digital convexity is preserved if the convex hull is quasi- $(1, \sqrt{2})$ -regular.

4.5.3.1 Digital convexity

In \mathbb{R}^2 , a set X is said to be convex if, for any pair of points $x, y \in X$, the line segment joining x and y

$$[x, y] = \{\lambda x + (1 - \lambda)y \in \mathbb{R}^2 \mid 0 \leq \lambda \leq 1\} \quad (63)$$

is included in X . However, this intuitive continuous notion cannot be directly transposed to digital sets of \mathbb{Z}^2 . Indeed, given a connected set X in \mathbb{Z}^2 , for $p, q \in X$ we generally have $[p, q] \not\subset \mathbb{Z}^2$.

In order to tackle this problem, various extensions of the notion of convexity have been proposed for \mathbb{Z}^2 . We can cite, for instance: MP-convexity [124] which is a straightforward extension of the continuous notion; S-convexity [180] which uses convex objects in \mathbb{R}^2 to determine the convexity of objects in \mathbb{Z}^2 ; H-convexity [98] which is a geometrical

version of S-convexity, using the convex-hull of digital objects; and D-convexity [99] which is based on the notion of digital line.

In the case of 4-adjacency modeling of digital objects, MP- and H-convexities have been proved equivalent [98, Theorem 5]. Similar results under the assumption of 8-adjacency can be found in [55], via the chord property, which relate the MP-, H- and D-convexities. Under the condition that X has no isolated point (i.e., no point adjacent to one other point within X), it was then proved that X is H-convex iff it is S-convex [98, Theorem 4]. A more complete description on various notions of digital convexity can be found in [45, Chapter 9].

In this section, the notion of H-convexity was chosen. This is motivated, on the one hand, by its compliance with the other kinds of convexities in the case of 4-connected (and, a fortiori, well-composed) digital objects. On the other hand, the notion of H-convexity relies on the explicit definition of the convex hull of the digital object. Such polygonal object provides us with a continuous model that can be involved in the continuous part of our rigid motion algorithmic process.

We recall hereafter the definition of the convex hull of a digital object $X \subset \mathbb{Z}^2$, denoted by $Conv(X)$. Then, we provide the formal definition of H-convexity.

$$Conv(X) = \left\{ x = \sum_{i=1}^{|X|} \lambda_i p_i \in \mathbb{R}^2 \mid \sum_{i=1}^{|X|} \lambda_i = 1 \right. \\ \left. \wedge \forall i \in \{1, \dots, |X|\}, (\lambda_i \geq 0 \wedge p_i \in X) \right\} \quad (64)$$

Definition 4.12 (H-convexity [98]). *A digital object $X \subset \mathbb{Z}^2$ is H-convex if*

$$X = Conv(X) \cap \mathbb{Z}^2$$

i.e., if X is equal to the digitization of its continuous polygonal convex hull.

Remark 4.7. *An H-convex object is not necessarily connected. This is exemplified in Figure 84.*

It is important to notice that, similarly to continuous convexity, H-convexity remains stable by intersection. In particular, we have the following property.

Property 4.1. *Let X and Y be two digital objects in \mathbb{Z}^2 . If X and Y are H-convex, then $X \cap Y$ is H-convex.*

4.5.3.2 Polygonization of H-convex digital objects

Let us hereafter consider that the input is an H-convex digital object X . Then, one of the simplest polygonizations, which is the first step of the algorithmic process computing the rigid motion of X as illustrated in Figure 80, is the polygonal convex hull.

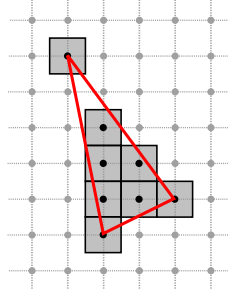


Figure 84: A digital object X that is H-convex, but not connected. This is due, here, to the acute angle at the highest vertex of the convex hull $\text{Conv}(X)$ that allows the induced polygon to “pass between” two 4-adjacent points of the background of X .

If X contains at least three non-colinear points, then its convex hull $\text{Conv}(X)$ is a non-trivial convex polygon whose vertices are some points of X . As these vertices are grid points of \mathbb{Z}^2 , the polygon $\text{Conv}(X)$ is defined as the intersection of closed half-planes with integer coefficients

$$\text{Conv}(X) = \bigcap_{H \in \mathcal{R}(X)} H \quad (65)$$

where $\mathcal{R}(X)$ is the smallest set of closed half-planes that include X . This set is finite and sufficient for defining $\text{Conv}(X)$. Each closed half-plane H of this subset is defined as

$$H = \{(x, y) \in \mathbb{R}^2 \mid ax + by + c \leq 0\} \quad (66)$$

with $a, b, c \in \mathbb{Z}$ and $\gcd(a, b) = 1$. Note that the integer coefficients of H are obtained by a pair of consecutive vertices of $\text{Conv}(X)$, denoted by $u, v \in \mathbb{Z}^2$, which are in the clockwise order, such that

$$(a, b) = \frac{1}{\gcd(w_x, w_y)} (-w_y, w_x) \quad (67)$$

$$c = (a, b) \cdot u \quad (68)$$

where $(w_x, w_y) = v - u \in \mathbb{Z}^2$.

Many algorithms can be used to compute the convex hull of a digital object. In [48], a linear time algorithm determines whether a given polyomino is convex and, in that case, it returns its convex-hull. This method relies on the incremental digital straight line recognition algorithm [49], and uses the geometrical properties of leaning points of maximal discrete straight line segments on the contour. The algorithm scans the contour curve and decomposes it into discrete segments whose extremities must be leaning points. The tangential cover of the curve [58] can be used to obtain this decomposition. Alternatively, an approach presented in [26] uses tools of combinatorics on words to study contour words: the linear Lyndon factorization algorithm [54] and the Christoffel words. A linear time algorithm decides convexity

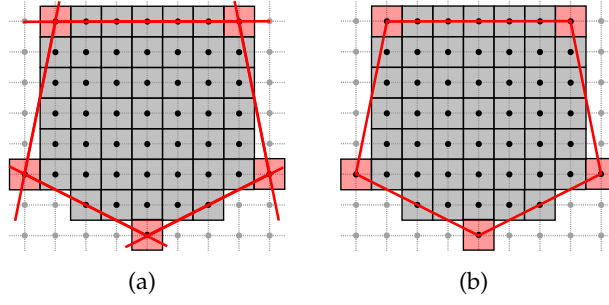


Figure 85: A digital H-convex object X of \mathbb{Z}^2 (black dots and grey pixels). (a) The half-plane representation of X , depicted by the 5 red support lines. The red points/pixels are those required to define these closed half-spaces. (b) The convex hull $\text{Conv}(X)$ in \mathbb{R}^2 , defined as the polygon whose vertices are these red points.

of polyominoes and can also compute the convex hull of a digital object (it is presented as a discrete version of the classical Melkman algorithm [123]).

The half-planes can then be deduced from the consecutive vertices of the computed convex hull from (66)–(68). An example of convex hull and half-plane modeling of an H-convex digital object is illustrated in Figure 85.

4.5.3.3 Rational rigid motions of convex polygons

In order to perform rigid motions of convex polygons without any numerical approximation, one can consider only rigid motions with rational parameters. Doing so, only exact computations with integers can be involved. This does not constitute an applicative restriction, due to the density of “relevant” rational values within the rotation and translation parameter space.

Thus, we assume hereafter that all the parameters of a rigid motion \mathfrak{T} of (29) are rational, such that the rotation matrix R is defined as $\frac{1}{r} \begin{pmatrix} p & -q \\ q & p \end{pmatrix}$ where $p, q, r \in \mathbb{Z}$ constitute a Pythagorean triple, i.e., $p^2 + q^2 = r^2$, $r \neq 0$ and the translation vector is defined as $(t_1, t_2) \in \mathbb{Q}^2$. This assumption is fair, as we can always find rational parameter values as close as desired from any real values [9] for defining such a Pythagorean triple.

A half-plane H , as defined in (66), is transformed by such rational rigid motion \mathfrak{T} as follows

$$\mathfrak{T}(H) = \{(x, y) \in \mathbb{R}^2 \mid \alpha x + \beta y + \gamma \leq 0\} \quad (69)$$

where $\alpha, \beta, \gamma \in \mathbb{Q}$ are given by $(\alpha \ \beta)^T = R(a \ b)^T$ and $\gamma = c + \alpha t_1 + \beta t_2$. This leads to a rational half-plane, which can be easily rewritten as an integer half-plane in the form of (66).

4.5.3.4 Digitization of convex polygons: geometric issues

Since an H-convex digital object X is represented by a finite set of digital half-planes, we can define the rigid motion \mathcal{T}_{Conv} of X on \mathbb{Z}^2 via its continuous polygonal convex hull as follows:

$$\mathcal{T}_{Conv}(X) = \mathfrak{T}(Conv(X)) \cap \mathbb{Z}^2 = \mathfrak{T}\left(\bigcap_{H \in \mathcal{R}(X)} H\right) \cap \mathbb{Z}^2. \quad (70)$$

This constitutes an alternative to the standard pointwise rigid motion defined in (54). Note that the backwards model is considered here thanks to the surjection. In order to distinguish this standard pointwise rigid motion \mathcal{T} from \mathcal{T}_{Conv} , we write \mathcal{T}_{Point} instead of \mathcal{T} .

We have

$$\begin{aligned} \mathfrak{T}\left(\bigcap_{H \in \mathcal{R}(X)} H\right) \cap \mathbb{Z}^2 &= \left(\bigcap_{H \in \mathcal{R}(X)} \mathfrak{T}(H)\right) \cap \mathbb{Z}^2 \\ &= \bigcap_{H \in \mathcal{R}(X)} (\mathfrak{T}(H) \cap \mathbb{Z}^2), \end{aligned} \quad (71)$$

and the digitization of any continuous half-space of \mathbb{R}^2 is H-convex. Then, from (70–71), $\mathcal{T}_{Conv}(X)$ is expressed as the intersection of a finite number of H-convex digital objects. The following proposition is then a corollary of Property 4.1.

Corollary 4.3. *Let X be a digital object of \mathbb{Z}^2 . Let \mathcal{T}_{Conv} be the polygon-based rigid motion induced by a rigid motion \mathfrak{T} with rational parameters. If X is H-convex, then $\mathcal{T}_{Conv}(X)$ is H-convex.*

The polygon corresponding to the convex hull of $\mathcal{T}_{Conv}(X)$ is not equal, in general, to the transformed convex hull of X . However, we have the following inclusion relation.

Property 4.2. *With the same hypotheses as in Corollary 4.3, we have*

$$Conv(\mathcal{T}_{Conv}(X)) \subseteq \mathfrak{T}(Conv(X)).$$

The proof of this property derives from the fact that $\mathcal{T}_{Conv}(X) = \mathfrak{T}(Conv(X)) \cap \mathbb{Z}^2$. Thus we have $\mathcal{T}_{Conv}(X) \subseteq \mathfrak{T}(Conv(X))$, and this inclusion also holds for the convex hull of $\mathcal{T}_{Conv}(X)$.

This inclusion indicates first that the cardinality of $\mathcal{T}_{Conv}(X)$ is lower (often strictly) than that of X . In other words, \mathcal{T}_{Conv} is a decreasing operator with respect to the cardinality of the input digital object; these facts are exemplified in Figure 86. A straightforward consequence is that \mathcal{T}_{Conv} is not bijective, in general. Second, this implies that the polygons of the two convex hulls of the input and output digital objects may be distinct, with respect to their number and size of edges, and angles at vertices. However, the convexity of the objects is preserved, which was the fundamental property to satisfy.

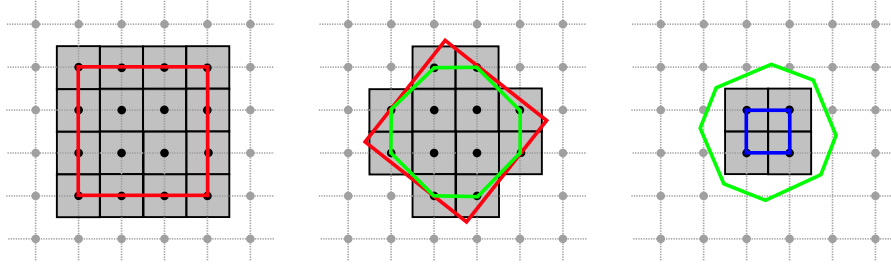


Figure 86: A sequence of transformations \mathcal{T}_{Conv} on an H-convex object X . The convex hull of $\mathcal{T}_{Conv}(X)$ is included in the transformed convex hull of X and the cardinality of $\mathcal{T}_{Conv}(X) \cap \mathbb{Z}^2$ is lower than that of X .

4.5.3.5 Digitization of convex polygons: topological issues

In \mathbb{R}^2 , the continuous definition of convexity intrinsically implies connectedness. By contrast, in \mathbb{Z}^2 the notion of H-convexity (such as various other notions of digital convexity) does not always offer guarantees of connectedness, e.g. with respect to 4- and 8-adjacencies.

In order to illustrate that fact, let us consider the example of Figure 84. The digital object X , composed of 8 points/pixels, is H-convex. Indeed, its convex hull contains only digital points that belong to X . However, X is not connected (neither with 4- nor 8-adjacencies). Such phenomenon is mainly caused by angular or metric factors: whenever an angle of the convex hull polygon is too acute, or when an edge is too short, such disconnections may happen.

Then, in addition to providing geometry guarantees of convexity—via the H-convexity of digital objects—when performing rigid transformations of a digital object, it is desirable to also provide topology guarantees, and more precisely connectedness guarantees.

To reach that goal, we use the notion of *quasi- (r, r') -regularity* introduced in Section 4.5.2.3. This additional notion provides us with sufficient conditions for ensuring that a digital H-convex object will remain not only H-convex but also connected after any rigid motion. The next corollary is obtained from Proposition 4.14 and Corollary 4.3.

Corollary 4.4. *Let $X \subset \mathbb{Z}^2$ be an H-convex 4-connected set. If $Conv(X)$ is quasi- $(1, \sqrt{2})$ -regular, then $\mathcal{T}_{Conv}(X)$ is H-convex, 4-connected and well-composed.*

Remark 4.8. *If $Conv(X)$ is quasi- $(1, \sqrt{2})$ -regular, then the initial digital set X is also 4-connected and well-composed.*

The experiment of rigid motions was carried out on a digitized ellipse X , which is H-convex (see Figure 87 (a)). Figure 87 presents the results of \mathcal{T}_{Point} and \mathcal{T}_{Conv} applied to X . It should be mentioned that \mathcal{T}_{Conv} preserves the H-convexity of X as shown in Corollary 4.3 and Figure 87 (c). By contrast, \mathcal{T}_{Point} hardly preserves the H-convexity as illustrated in Figure 87 (b).

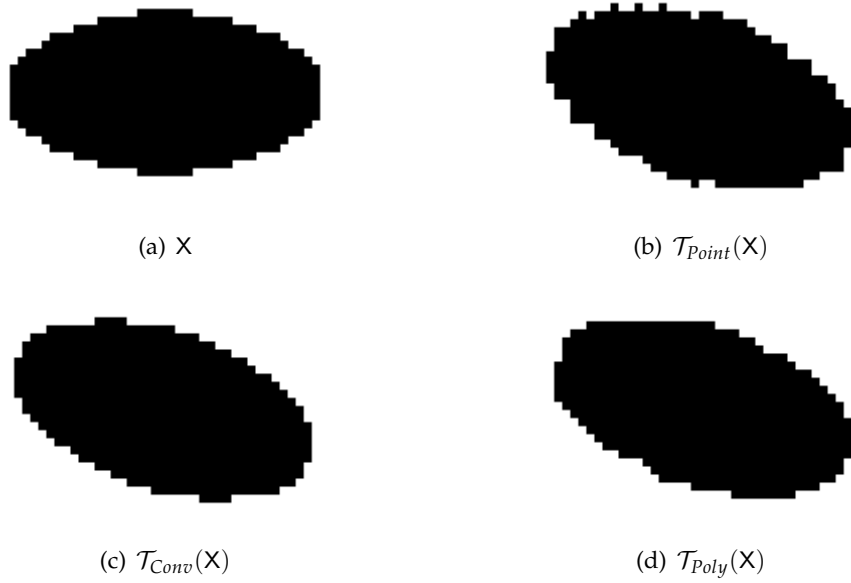


Figure 87: Comparison between the digitized rigid motions \mathcal{T}_{Point} (b), \mathcal{T}_{Conv} (c) and \mathcal{T}_{Poly} (d), of a digitized ellipse (a), which is H-convex, with rotation angle of $\frac{\pi}{10}$ and translation of $(t_x, t_y) = (0.1, 0.2)$. H-convexity is preserved by \mathcal{T}_{Conv} while it is not by \mathcal{T}_{Point} and \mathcal{T}_{Poly} .

4.5.4 Digitized rigid motions for non-convex objects

We now deal with rigid motions of digital objects without convexity hypothesis. We propose and show experimentally how geometric measures such as area and perimeter are preserved.

4.5.4.1 Polygonization of a non-convex digital object

There exist various methods for polygonizing a digital object. In the field of digital geometry, numerous approaches used the contour curves extracted from the digital objects; each method computes a polygonal representation of the digital object with particular properties. In [50], [175], invertible methods enable us to compute Euclidean polygons whose digitization is equal to the original discrete boundary. These methods use the Vittone algorithm [196] in the preimage space for straight line recognition. In [51], [52], [59], [169] the arithmetical recognition algorithm [49] is used to decompose a discrete contour and deduce a polygonal representation. These methods rely on the tangential cover of the contour [58], composed of the sequence of its maximal discrete straight segments. It was proved in [59] that all polygonal representations of the contour can be deduced from its tangential cover, leading to a linear algorithm which computes the polygon with minimal integral summed squared error. In [51], [52], [169], the goal was different. It consisted of determining a

reversible polygon that faithfully represents the convex and concave parts of the boundary of a digital object. The polygonization method proposed in [135], [139] also exploits the idea of maximal straight segment primitives. It allows to identify the characteristic points on a contour, called dominant points, and to build a polygon representing the given contour. Another technique presented in [70] is the curve decomposition. It uses the analytical primitives, called digital level layers, to decompose a given contour and to obtain an analytical representation. Another algorithm is proposed in [178] to compute the polygonal simplification of a curve such that the Fréchet distance [66] between the simplified polygon and the original curve is lower than a given error.

It should be mentioned that, for a given digital object, different results can be obtained from these various polygonization techniques. In other words, the polygonal representation of a digital object is not unique. However, the crucial property to be satisfied is that the polygon $P(X)$ computed for a digital object X has to be coherent with respect to digitization, i.e. $P(X) \cap \mathbb{Z}^2 = X$. This property is called reversibility. A second important property, in our framework of discrete geometry and exact calculus, is that the vertices of $P(X)$ have rational coordinates, which we call rationality.

To the best of our knowledge, none of the above methods guarantees both of these properties for any X if they are directly applied to the boundary of X . In fact, most of them compute a simplified polygon from a digital curve, which must be generated from X as its boundary. However, it is not obvious to obtain a “nice” digital curve from any X , which leads to a “nice” $P(X)$. On the other hand, we can find very simple polygonization methods that respect both of the properties, for example:

1. making a rectilinear polygon as the boundary of the cell complex of X [108],
2. making a polygonal curve generated by the marching square method, which is the 2D version of marching cube [118], [198],

under the condition that X is well-composed. Those simple polygonization methods, however, may provide “poor” geometric properties.

Here, we adapt a polygonization strategy based on [135], [139], which guarantees the above two properties, for the first trial. See [137] for the details of this polygonization scheme.

4.5.4.2 Rational rigid motion of a non-convex polygon

As the polygon $P(X)$ may not be convex, we cannot use the half-plane representation, as it was done in Section 4.5.3.2 for convex polygons. Here, we use a standard vertex representation, by modeling a polygon via a sequence of successive vertices of its boundary.

Note that the vertices of $P(X)$ should be integer or rational points, and those of $\mathfrak{T}(P(X))$ are rational points, if the rigid motion \mathfrak{T} is given by a rational matrix and a rational translation vector (see Section 4.5.3.3). Then, for each vertex of the polygon $P(X)$, we simply apply the rigid motion \mathfrak{T} of (29) with such rational setting, as described in Section 4.5.3.3, and preserve the order of the vertex sequence.

4.5.4.3 Digitization of a polygon and geometric/topological issues

Once the polygon $\mathfrak{T}(P(X))$ has been computed, the resulting object, denoted by $\mathcal{T}_{poly}(X)$ can be deduced. Similarly to the case of H-convex digital objects (see (70)), this is done by embedding $\mathfrak{T}(P(X))$ in \mathbb{Z}^2 via the Gauss digitization

$$\mathcal{T}_{poly}(X) = \mathfrak{T}(P(X)) \cap \mathbb{Z}^2 \quad (72)$$

Various ways exist for carrying out this digitization in an exact way. For instance, it is possible to decompose $\mathfrak{T}(P(X))$ into a partition of triangles whose vertices are (rational-coordinate) vertices of the boundary of $\mathfrak{T}(P(X))$. Each of such triangles being defined as a convex region modeled by three half-planes with rational parameters, the points of \mathbb{Z}^2 contained herein can be determined without numerical error.

In order to ensure the connectedness preservation of X , we require, as for the H-convex case, that the polygon $P(X)$ of X , is quasi- $(1, \sqrt{2})$ -regular.

Proposition 4.15. *Let $X \subset \mathbb{Z}^2$ be a digital object. Let $P(X) \subset \mathbb{R}^2$ be a polygon such that $P(X) \cap \mathbb{Z}^2 = X$. If $P(X)$ is quasi- $(1, \sqrt{2})$ -regular, then $\mathcal{T}_{poly}(X)$ is 4-connected and well-composed.*

Remark 4.9. *Beyond topological guarantees such as 4-connectedness and well-composedness, the notion of quasi- $(1, \sqrt{2})$ -regularity also presents some geometric properties. Indeed, any point of X is either part of $P(X) \circ B_1$ (i.e. the “smooth” opening of a polygon) or part of the (noisy) boundary in $P(X) \setminus (P(X) \circ B_1)$. But, in this second case, this point is necessarily at a distance not greater than $\sqrt{2} - 1 < 0,5$ (i.e. the half of a pixel size) from this opening $P(X) \circ B_1$. In other words, quasi- $(1, \sqrt{2})$ -regularity describes objects with boundaries that may not be completely smooth (in particular, they may be non-differentiable), but that will be, in the worst cases, only slightly noisy, by contrast with results of standard pointwise rigid motions \mathcal{T}_{Point} . This is illustrated in Table 13.*

Note that $P(X)$ can be non-convex even if X is H-convex, depending on a chosen polygonization method, as illustrated in Figure 88. Therefore, \mathcal{T}_{poly} does not guarantee the H-convexity of the transformed object (see Figure 87 (d) for an example).

As stated above, $P(X)$ can be defined by following various policies. Then, there exist many (actually an infinite number of) polygons

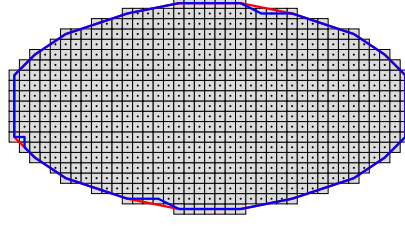


Figure 88: Polygon (in blue) and convex hull (in red) of the digital object X of Figure 87 (a).

whose digitization leads to X . In particular, it may happen that $P(X)$ is not quasi- $(1, \sqrt{2})$ -regular while X and $\mathcal{T}_{Poly}(X)$ are indeed 4-connected and well-composed. This statement emphasizes the importance of choosing wisely a polygonization policy. In this context, various properties may be relevantly targeted.

4.5.5 Experiments

The experiments on a digitized square of size 21×21 , which is H-convex, was made as follows: using the three transformation models, \mathcal{T}_{Point} , \mathcal{T}_{Conv} and \mathcal{T}_{Poly} , we rotate the object with angle $\frac{\pi}{10}$ around its center repeatedly, such that the rotation is first applied on the input image, and then the transformed image is used as input for the next rotation, and so on. Table 13 illustrates the visual results of successive rotated images with the number of the object grid points, which is interpreted as the area of the rotated shape, after each rotation. We observe that the \mathcal{T}_{Point} alter not only the digital topology of the object boundary but also the H-convexity. By contrast, \mathcal{T}_{Conv} preserves the topology together with the H-convexity since $Conv(X)$ is quasi- $(1, \sqrt{2})$ -regular, as discussed in Section 4.5.3.5, while the number of the object points is decreasing as seen in Remark 4.2. We observe that \mathcal{T}_{Poly} avoids this shrinking effect without topological alteration thanks to the quasi- $(1, \sqrt{2})$ -regularity of $P(X)$ for this example.

In order to quantify experimentally the accuracy and stability of geometric measurements using the three models of rigid motions on convex digital objects, we observe two measures: area and perimeter. The area is computed simply as the number of digital points within the transformed objects [100] and the perimeter is calculated based on curve segmentation by maximal digital standard segment [107] of the 4-connected curves extracted from the transformed objects. It has been proven that these estimators have multigrid convergence property [34]. The experiments were done with rotations for angles θ varying from 0 to 2π . Figure 89 reports some quantitative comparisons of those geometric measures between rotations by \mathcal{T}_{Point} , \mathcal{T}_{Conv} and \mathcal{T}_{Poly} on the input images given in Table 13. We can observe that \mathcal{T}_{Point} and \mathcal{T}_{Conv} do not preserve well the perimeter of the transformed objects since \mathcal{T}_{Point} alters the boundary of the objects and \mathcal{T}_{Conv} is a

















	 #Points=441				
	$\theta = \frac{\pi}{10}$	$\theta = \frac{2\pi}{10}$	$\theta = \frac{3\pi}{10}$	$\theta = \frac{4\pi}{10}$	$\theta = \frac{\pi}{2}$
$\mathcal{T}_{Point}(X)$	 #Points=445	 #Points=437	 #Points=437	 #Points=445	 #Points=441
$\mathcal{T}_{Conv}(X)$	 #Points=397	 #Points=385	 #Points=373	 #Points=357	 #Points=349
$\mathcal{T}_{Poly}(X)$	 #Points=397	 #Points=409	 #Points=409	 #Points=397	 #Points=441

Table 13: Comparison of the three different models for digitized rigid motion, \mathcal{T}_{Point} , \mathcal{T}_{Conv} and \mathcal{T}_{Poly} , on a digitized square of size 21×21 . A rotation of angle $\frac{\pi}{10}$ around the square center is applied repeatedly, so that the accumulated rotation angle θ is given.

decreasing operator. By construction, \mathcal{T}_{Poly} uses a polygon that fits the input digital object for the transformation; thus it preserves better the perimeter. For the same reasons, \mathcal{T}_{Conv} does not preserve well the area, contrary to \mathcal{T}_{Poly} . Since \mathcal{T}_{Point} is defined on a point-by-point model, it also preserves well the area.

The last experiments were made to non-H-convex digital objects, as illustrated in Figure 90. Here we compared the two transformation models, \mathcal{T}_{Point} and \mathcal{T}_{Poly} , with respect to the estimations of area and perimeter, similarly to those for the H-convex case. The results are respectively shown in Figure 91. We can observe that both of \mathcal{T}_{Point} and \mathcal{T}_{Poly} have a stable behaviour with respect to the area measurement, while \mathcal{T}_{Poly} preserves better the perimeter than \mathcal{T}_{Point} .

4.5.6 Extension to 3D

The extension of the digitized rigid motions via polygonization for 2D to 3D is straightforward, in particular for convex cases. The important

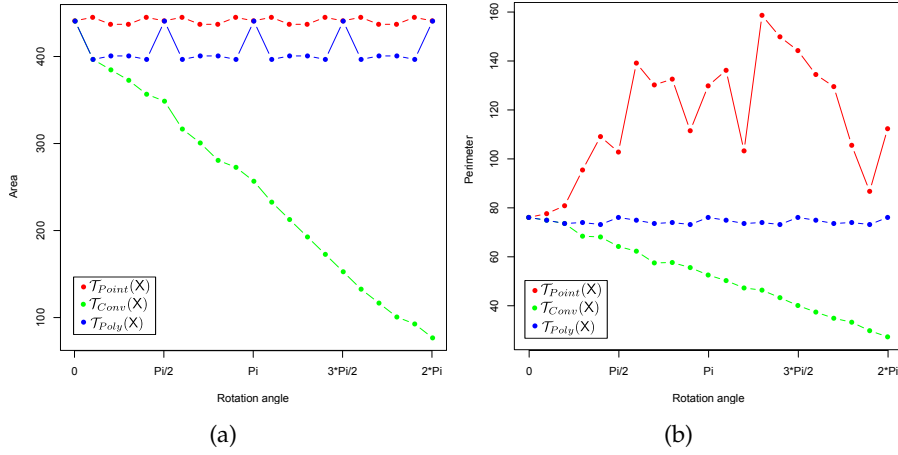


Figure 89: Estimated area (a) and perimeter (b) variations after applying rotations of angles varying from 0 to 2π to the digitized square of size 21×21 used in Table 13.

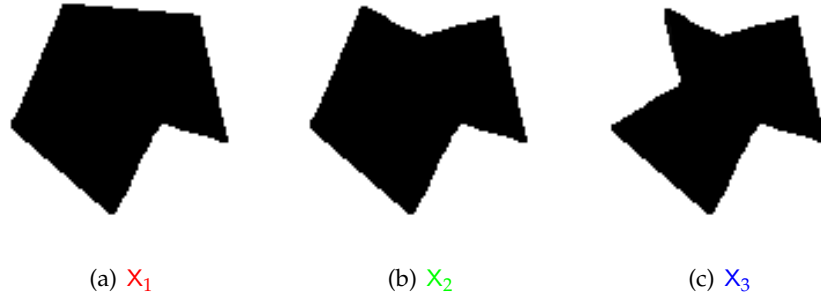


Figure 90: Non-convex digital objects used for the quantitative experiments.

elements, such as H-convexity, quasi- (r, r') -regularity and convex hull, can be defined for 3D as well. Thus, the following corollary, which is similar to Corollary 4.4 for 2D, can be also obtained for 3D [136].

Corollary 4.5 (Ngo et al. [136]). *Let $X \subset \mathbb{Z}^3$ be a finite H-convex and 6-connected set. If $\text{Conv}(X)$ is quasi- $(1, \frac{2}{\sqrt{3}})$ -regular, then $\mathcal{T}_{\text{Conv}}(X)$ is both H-convex and connected.*

An example of digital object, whose convex hull is quasi- $(1, \frac{2}{\sqrt{3}})$ -regular, is given in Figure 92 (a). The behaviour of $\mathcal{T}_{\text{Conv}}$ compared to $\mathcal{T}_{\text{point}}$ when applied on the H-convex object is illustrated in Figure 92. One can observe the better preservation of the shape of the transformed object, and in particular the preservation of H-convexity by $\mathcal{T}_{\text{Conv}}$.

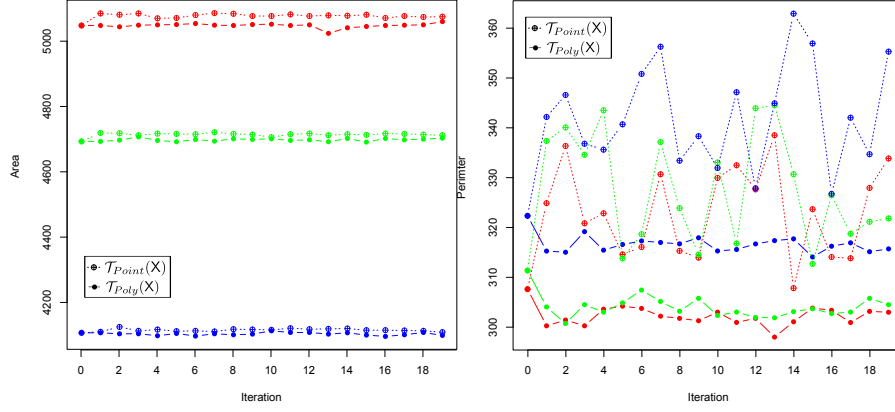


Figure 91: Area (left) and perimeter (right) evolution of the three digital objects X_1 , X_2 and X_3 (see Figure 90), under successive rigid motions \mathcal{T}_{Point} and \mathcal{T}_{Poly} .

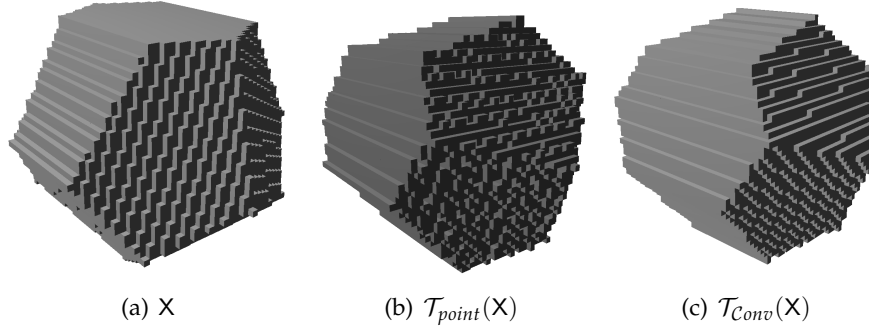


Figure 92: An H-convex digital object X such that $Conv(X)$ is quasi-1-regular (a) and its digital rigid motions by \mathcal{T}_{point} (b) and by \mathcal{T}_{Conv} (c).

4.6 SUMMARY AND PERSPETIVES

In this chapter, we tackled the following problems of digitized rigid motions:

1. combinatorial analysis of local configurations;
2. characterization of bijective rigid motions;
3. topological invariance/alteration;
4. geometric invariance/alteration.

In order to solve the first problem, we first introduced a combinatorial structure represented by a graph for modelling the parameter space of digitized rigid transformations. In the dual structure, this problem can be seen as decomposing the parameter space for a given image patch of finite size. We analyzed the space complexity of such combinatorial structures in 2D and 3D cases. Besides, algorithms based on sweeping plane were proposed for constructing the combinatorial structure in linear time with respect to this space complexity

for the 2D case, and for computing only sample points for the 3D case. This structure can be also calculated locally in the 2D case, and allows us a fully discrete framework of image registration [134] and geometric transform. The similar problem in the 3D case should be also treated in near future.

Concerning the second problem, we have seen two approaches: the combinatorial and arithmetic ones. In the first approach, we first extended the neighborhood motion maps to rigid motions, previously proposed by Nouvel and Rémila [143] for digitized rotations on \mathbb{Z}^2 . This led us to the necessary and sufficient condition of bijective rigid motions on \mathbb{Z}^2 . This combinatorial approach could be extended to digitized rigid motions on \mathbb{Z}^3 with a help of uncoupling the parameters of 3D rigid motions [149], and results would be helpful for characterizing bijective digitized rigid motions on \mathbb{Z}^3 . On the other hand, the second arithmetic approach, which is already admitted as a useful tool for characterizing bijective 2D digitized rotations [168], allows us to show the existence of non-simple 3D bijective digitized rotations—ones for which a given rotation axis does not correspond to any of the coordinate axes; we do not succeed yet to fully characterize the set of 3D bijective digitized rotations using Lipschitz quaternions, which play a similar role to Gaussian integers for the 2D case. This is due to the non-commutative nature of quaternions and their two-to-one relation with 3D rotations.

In order to tackle the third problem, we first focused on the 2D problem. Based on theoretical results established in the digital topology framework, we proposed the notion of digital regularity, based on which we derived efficient algorithms for analyzing and preprocessing such images. The genericity of these results and methods, in terms of topological models (dual adjacency and well-composedness) and values (binary, grey-level and label images), authorize their actual use in real applications [138]. As a priority, it should be first sought whether the notion of digital regularity provides not only sufficient, but also *necessary* conditions for topological invariance. We will also investigate the link between the notions of digital regularity and quasi-regularity, that intrinsically merges digital and discrete (and even continuous) frameworks. Indeed, the rigid transformation of a digital image can be interpreted as the re-digitization of its associated continuous pixel-based, thus polygonal representation. This link may help us to explain why digital regularity does not provide a sufficient condition for topological invariance in 3D to the contrary to 2D.

The last problem was solved in an algorithmic fashion; we first proposed an algorithmic process for performing rigid motions of well-composed sets on \mathbb{Z}^2 , while preserving their global shapes. This shape preservation was expressed in terms of geometry, but also in terms of topology, since the object should not be erroneously disconnected due to the discrete structure of \mathbb{Z}^2 . In order to tackle these issues, we con-

sidered an intermediate continuous model of the digital object, namely a polygonal model, and proposed a new notion of quasi- r -regularity that provides sufficient conditions for guaranteeing topological preservation when digitizing a continuous object, which can be seen as a generalization of the classic notion of r -regularity. This notion of quasi- r -regularity was indeed required to correctly handle the mandatory digitization step induced by the use of an intermediate continuous polygonal model. This work opens the way to various perspectives. First we will investigate how this rigid motion scheme can be extended to the 3D case, i.e. to digital objects defined in \mathbb{Z}^3 . Such an extension cannot be straightforward as topological (and geometric) properties of Gauss digitization for more than two dimensions are different and more complex than those in two dimensions [111]. Second, from a practical point of view, we will investigate the relevance of different polygonization approaches, in order to identify those that are the best fitted to the proposed transformation approach.

CONCLUSION AND PERSPECTIVES

5.1 CONCLUSION

In this manuscript, we considered the following three topics, related to analyzing and manipulating shapes in \mathbb{Z}^2 and \mathbb{Z}^3 , in different contexts: topological shape analysis (Chapter 2), geometric shape analysis (Chapter 3), and shape rigid motions (Chapter 4). For each problem, we adopted a combinatorial approach, which consists of

- studying the combinatorial structure of the solution space and its properties, and
- proposing some solutions and efficient algorithms based on these properties.

In Chapter 2, we first proposed a framework based on polyhedral complexes such that the vertices are in \mathbb{Z}^3 and any adjacency between polygonal vertices are the m -adjacency, $m = 6, 18, 26$. Given a finite subset $X \subset \mathbb{Z}^3$, we presented a method for constructing a polyhedral complex and classified each point of X into one of the twelve types by the topological characterization of its star, as shown in Figure 13, among which type 3b (semi-spherical star) corresponds to the local configuration on combinatorial surfaces. We also enumerated such local configurations.

Thanks to this local topological characterization of points, the proposed framework has several utilities: it allowed us to propose a discrete version of the marching cubes method for boundary extraction (see Section 2.5) and a linear thinning algorithm of three-dimensional digital images based on collapsibility of polyhedral complexes [85] (see Section 2.6), and to define topologically reasonable discrete surface patches, on each of which geometrical measures, such as normal vector, can be calculated in a finite way [87] (see Chapter 3).

In Chapter 3, two different problems were considered: digital planar surface segmentation and digital plane fitting. The former problem was solved by a discrete version of the hybrid method, based on the previously obtained topologically reasonable discrete surface patches: filtering out non-linear points and classifying points with respect to normal vectors in the unified discrete Gaussian image. The experimental results show that our method is useful for planar surface segmentation from a point cloud if the input is sufficiently close to an ideal image, i.e., if it involves only quantization errors. However, such ideal images are rarely obtained in practice as input images generally contain noise in addition to quantization errors. In order to eliminate

such noise, we reduce image resolutions before applying our method for example.

In order to find a solution even for such noisy images, we also exposed a method for digital line and plane fitting in the presence of outliers in the second part of Chapter 3. A digital hyperplane model was used to develop the discrete optimization-based framework. The exact fitting problem was solved with an $O(N^d \log N)$ algorithm for $d = 2, 3$, which was later improved with an $O(N^2)$ solution in 2D using a topological sweep method. While a polynomial solution of degree equal to the dimension of the problem is useful, it is still too inefficient for many applications; the problem is solvable for $N = 10^3$ but intractable for $N = 10^6$ in 3D. For more practical use, we therefore proposed an approximate digital hyperplane fitting method, which uses an accumulation and query data structure and is easy to implement. The method features bounded error, which trades off the algorithm running time and memory requirements.

In Chapter 4, we first constructed a combinatorial model of the local behavior of displacements on \mathbb{Z}^2 by classifying the transformations according to their effect on a digital image patch. This classification can be interpreted as specific problems of hypersurface arrangements in the parameter space of the transformations, which were solved by methods of computational geometry and computer algebra.

This combinatorial model then allowed us to characterize the bijective transformations on \mathbb{Z}^2 . In the 3D case, on the other hand, we adopted an arithmetic approach, which led an algorithm to certify the bijective digitized rotations; the bijectivity characterization is not yet made.

Concerning the topological problem, the class of two-dimensional images that preserve their topological properties during rigid motions – called digitally regular images – has been identified, as well as methods allowing such “regularization”. Nevertheless, these strategies are not intrinsically transposable to higher dimensions.

Finally, the geometric problem was solved in an algorithmic fashion in \mathbb{Z}^2 . We considered an intermediate continuous model of the digital object, namely a polygonal model, and proposed a notion of quasi-regularity that provides sufficient conditions for topological invariance when digitizing a polygonal object. With this condition, any rigid motion can be applied to a polygonal object without topological alteration of the digitized polygon. Some geometric properties are intrinsically maintained via the polygonal representation. The extension of this approach to 3D would be straightforward.

5.2 ONGOING AND FUTURE CHALLENGES

There are various interesting themes, which are related to the presented work in this manuscript. In particular, many 3D problems are

still left as the difference between 2D and 3D is so large that methods that work perfectly in 2D are often not applicable to 3D. As the recent work is mainly presented in Chapter 4, most of the following topics are also related to those in the last chapter.

5.2.1 *Ongoing challenges*

Topologically invariant rigid motions on \mathbb{Z}^3

As we discussed in Section 4.4.3, digital regularity does not guarantee topological properties of digital shapes under rigid motions on \mathbb{Z}^3 , in contrast to the case of \mathbb{Z}^2 (see Section 4.4.2). In order to understand this difference between 2D and 3D, quasi-regular sets, which were proposed in Section 4.5 in order to guarantee topological invariance under rigid motions via polygonization/polyhedrization, would be useful as they can make a link between continuous and digital sets.

Collaboration with: Phuc Ngo (LORIA), Nicolas Passat (CReSTIC), Kacper Pluta (Technion, Israel)

Polygonization/Polyhedrization of a subset of \mathbb{Z}^n

As observed in the previous argument, techniques for polygonization/polyhedrization of a subset X of \mathbb{Z}^n is one of the key issues for topologically (and geometrically) invariant rigid motions on \mathbb{Z}^n (see Section 4.5). The following three criteria should be considered for constructed polygons/polyhedra $P(X)$, which are possibly non-convex: (1) Gaussian digitization of $P(X)$ has to get back to an initial digital object X (reversibility); (2) the coordinates of all the vertices of $P(X)$ must be rational (rationality); (3) both $P(X)$ and $P(\bar{X})$ have the common boundary where \bar{X} is the complement of X (auto-duality). We are investigating such polygonization/polyhedrization techniques.

Collaboration with: Isabelle Debled-Rennesson (LORIA), Phuc Ngo (LORIA), Nicolas Passat (CReSTIC)

Measuring geometric alteration on digital shape boundary under rigid motions

It will be pertinent to explore also questions of discrete differential geometry, such as the behavior of discrete curvature under displacements. In this context, new notions of “geometric” and “topological” preservation must be proposed. For example, the approach of “integral geometry” [171], which allows us to define a measure of the continuous shapes digitized in the same digital shape, can be a good mathematical tool in order to search for displacements in \mathbb{Z}^n preserving this type of measure. There are also the tools of combinatorics on words, which allow us

to measure the balancedness or the equilibrateness of a word, which represents a rectilinear digital curve [188].

Collaboration with: David Coeurjolly (LIRIS), Lama Tarsissi (LIGM, Université Paris-Est), Pascal Romon (LAMA, Université Paris-Est)

Tangent estimation of 3D digital curves

In the field of digital geometry, many discrete differential geometry results can be found for digital curves in 2D space, while less results are found for digital curves in 3D space. However, there are practical demands, which have motivated us to estimate tangents along a 3D digital curve. The experimental results were already published in [154], and we are currently deepening properties such as multi-grid convergence.

Collaboration with: Jacques-Olivier Lachaud (LAMA, Université de Savoie), Kacper Pluta (Technion, Israel)

Approximate digital polynomial function fitting to a subset of \mathbb{Z}^n

Analytical models of digital geometry have been studied not only for digital lines and planes but also for non-linear curves and surfaces [8]. We tackle the problem of non-linear curve / surface using such digital analytical models. With a careful consideration, this problem can be seen as stabbing polytopes with a hyperplane. With a help of tools of computational geometry, we will apply the techniques of random sampling and ϵ -approximation [129] in order to develop efficient algorithms which would be useful in practice even if the dimension of the parameter space is relatively high.

Collaboration with: Eric Andres (XLIM, Université de Poitiers), Gaëlle Largeteau-Skapin (XLIM, Université de Poitiers), Nabil Mustafa (LIGM, Université Paris-Est), Rita Zrour (XLIM, Université de Poitiers)

5.2.2 Future challenges

Characterization of 3D bijective digitized rigid motions

This is still an open problem, as we already discussed in Section 4.3. As we did not yet find any clue with the arithmetic approach (see Section 4.3.2), adopting the combinatorial approach, presented in Section 4.3.1, for this 3D problem, would help us to observe and analyze the neighborhood motion maps. This extension can be made with a help of uncoupling the parameters of 3D rigid motions, whose technique was already used in Section 4.2.6.2

Optimal transport for finding bijective discrete transformations

This problem was initially posed by Yan Gérard (ISIT, Université Clermont Auvergne) during the meeting of CoMeDic Poject held

in July 2018. We consider all bijections T from \mathbb{Z}^2 to \mathbb{Z}^2 , which are called (digital) transports. The problem is to find the optimal transport, whose total cost, $\sum_{p \in \mathbb{Z}^2} c(p, T(p))$, is the least of all possible transports. The question is: can results obtained by solving such optimal transport problems be as bijective digitized rigid motions?

Local search for 3D rigid motion registration

The method for computing an arrangement for 3D digitized rigid motions proposed in Section 4.2.6.2 is simply aimed at obtaining at least one sampling point for each full-dimensional cell; thus, no topological information of the arrangement is computed. However, there is sometimes a demand for exploring the combinatorial structure of digitized rigid transformations along the dual graph structure, for example, when discrete local search is used for rigid image registration [134]. In this case, we need to construct the combinatorial structure containing the topological information, which is not necessarily globally but only locally. For this aim, we will extend the discrete local graph-search method for 2D rigid image registration [134] to 3D, with a help of computer algebra tools [149].

Extension to other geometric transformations

In computer vision and image processing/analysis, other geometric transformations are also used. Similar study to rigid motions should be made with consideration of more general frameworks.

Link to the quasi-affine transformation

There are generic approaches based on the quasi-affine transformations [21], [36], which show that the digitization of a continuous operation is sometimes unsatisfactory (loss of bijection), but sometimes very rich with the associated arithmetic structure (tilings, enumeration, etc.). It will be naturally interesting to link these approaches to the study of displacements in \mathbb{Z}^n .

Extension to other digitization models

In this manuscript, we consider only the Gaussian digitization [101]. However, there are other digitization models, such as the inner and outer Jordan digitization models [101], the grid-intersection digitization [67], the flake digitization models [192], etc. We will study how to adapt our framework of digitized rigid transformation based on Gaussian digitization to other models.

As-rigid-as-possible deformation of digital shapes

As-rigid-as-possible deformation, i.e. deformation which is locally rigid, is used in various scenes [181]. It would be useful

to formulate such deformation as a combinatorial optimization problem using the discrete framework of digitized rigid transformations, and to verify its utility to practical problems, such as image registration and object motion/deformation tracking.

Topologically correct conversion between meshes and voxel sets

Here, we consider scenes where there are several objects exist in a same space; such voxel data are given by labeled images generated from a 3D medical image containing several organs, for example. A voxel set corresponding to each label is often converted for its visualization to a mesh using the marching cubes method [118], [198], which is often followed by mesh simplification [145]. In a multi-object scene, the topological structure between the objects in the scene should be preserved during the conversion as well as the geometric precision.

On the contrary, the conversion from meshes to voxel sets is also useful to repair and analyze the complex structures of meshes [141]. Many methods for mesh voxelization exist. However, most of the methods are interested in voxelizing a surface [38], [112], but not in voxelizing a solid [141]. From the viewpoint of digital topology, solid voxelization is, indeed, more relevant to the Gaussian digitization model, which is generally used for full-dimensional objects co-existing in a space. If we would like to guarantee coherent topology between meshes and voxel sets wherever the meshes are positioned, considering the quasi-regularity condition on each mesh, presented in Section 4.5.1, will be a good starting point.

Hierarchical surface segmentation with digital geometric features

Polyhedrization of a voxel set is known as a very difficult problem [176]. Here, we re-revisit this problem as a segmentation problem and apply a hierarchical image segmentation approach. We recently worked on graph-based hierarchical image segmentation, whose principle is region merging with a criterion based on region dissimilarity [43], [71]. As polyhedral surfaces constructed from the method proposed in Section 2.5 have graph structures, we can apply this method to such discrete surfaces accompanied with digital geometric features, such as quantified normal vectors presented in Chapter 3. Specific dissimilarity should be considered according to the discrete nature of digital geometric features.

Hierarchical image vectorization

We recently have exploited various digital-contour representations and proposed a method of greyscale image vectorization, which allows a geometric quality control [95]. On the other hand,

as mentioned above, we have worked on graph-based hierarchical image segmentation, whose principle is region merging with a criterion based on region dissimilarity [43], [71]. In order to improve the quality of the greyscale or labeled image vectorization, we consider involving the hierarchical image structure in the procedure.

BIBLIOGRAPHY

- [1] P. Afshani and T. M. Chan, "On approximate range counting and depth," *Discrete & Computational Geometry*, vol. 42, no. 1, pp. 3–21, 2009.
- [2] D. Aiger, Y. Kenmochi, H. Talbot, and L. Buzer, "Efficient robust digital hyperplane fitting with bounded error," in *Discrete Geometry for Computer Imagery: Proceedings of 16th International Conference, DGCI 2011*, I. Debled-Rennesson, E. Domenjoud, B. Kerautret, and P. Even, Eds.
- [3] P. S. Alexandrov, *Combinatorial Topology*. Graylock Press, 1956, vol. 1.
- [4] A. Amir, A. Butman, M. Lewenstein, and E. Porat, "Real two dimensional scaled matching," *Algorithmica*, vol. 53, no. 3, pp. 314–336, 2009.
- [5] A. Amir, O. Kapah, and D. Tsur, "Faster two-dimensional pattern matching with rotations," *Theoretical Computer Science*, vol. 368, no. 3, pp. 196–204, 2006.
- [6] A. Amir, G. M. Landau, and U. Vishkin, "Efficient pattern matching with scaling," *Journal of Algorithms*, vol. 13, no. 1, pp. 2–32, 1992.
- [7] E. Andrès, R. Acharya, and C. Sibata, "Discrete analytical hyperplanes," *Graphical Models and Image Processing*, vol. 59, pp. 302–309, 1997.
- [8] E. Andres, "Digital analytical geometry: how do i define a digital analytical object?" In *Combinatorial Image Analysis: 17th International Workshop, IWCIA 2015, Kolkata, India, November 24-27, 2015. Proceedings*, R. P. Barneva, B. B. Bhattacharya, and V. E. Brimkov, Eds. Cham: Springer International Publishing, 2015, pp. 3–17.
- [9] W. S. Anglin, "Using Pythagorean triangles to approximate angles," *American Mathematical Monthly*, vol. 95, no. 6, pp. 540–541, 1988.
- [10] B. Aronov and S. Har-Peled, "On approximating the depth and related problems," *SIAM Journal of Computing*, vol. 38, no. 3, pp. 899–921, 2008.
- [11] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

- [12] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry : Algorithms and Applications*, Third. Springer, 2008.
- [13] G. Bertrand, "New notions for discrete topology," in *Discrete Geometry for Computer Imagery, Proceedings of 8th International Conference, DGCI'99*, vol. 1568 of LNCS, 1999, pp. 218–228.
- [14] —, "On P-simple points," *Comptes Rendus de l'Académie des Sciences, Série Math*, vol. I, no. 321, pp. 1077–1084, 1995.
- [15] —, "Simple points, topological numbers and geodesic neighborhoods in cubic grids," *Pattern Recognition Letters*, vol. 15, pp. 1003–1011, 1994.
- [16] G. Bertrand and M. Couprie, "A model for digital topology," in *Discrete Geometry for Computer Imagery, Proceedings of 8th International Conference, DGCI'99*, vol. 1568 of LNCS, 1999, pp. 229–241.
- [17] G. Bertrand, M. Couprie, and N. Passat, "A note on 3-D simple points and simple-equivalence," *Inform Process Lett*, vol. 109, no. 13, pp. 700–704, 2009.
- [18] G. Bertrand and G. Malandain, "A new characterization of three-dimensional simple points," *Pattern Recognition Letters*, vol. 15, no. 2, pp. 169–175, 1994.
- [19] P. J. Besl and R. C. Jain, "Segmentation through variable-order surface fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 2, pp. 167–192, 1988.
- [20] P. Bhowmick and B. B. Bhattacharya, "Fast polygonal approximation of digital curves using relaxed straightness properties," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1590–1602, 2007.
- [21] V. Blot and D. Coeurjolly, "Quasi-affine transformation in higher dimension," in *Discrete Geometry for Computer Imagery, 15th IAPR International Conference, DGCI 2009, Montréal, Canada, September 30 - October 2, 2009. Proceedings*, 2009, pp. 493–504.
- [22] N. Boutry, T. Géraud, and L. Najman, "A tutorial on well-composedness," *Journal of Mathematical Imaging and Vision*, vol. 60, no. 3, pp. 443–478, 2018.
- [23] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.
- [24] V. Brimkov, D. Coeurjolly, and R. Klette, "Digital planarity—a review," *Discrete Applied Mathematics*, vol. 155, no. 4, pp. 468–495, 2007.

- [25] V. E. Brimkov and R. P. Barneva, "Plane digitization and related combinatorial problems," *Discrete Applied Mathematics*, vol. 147, no. 2, pp. 169–186, 2005, Advances in Discrete Geometry and Topology.
- [26] S. Brlek, J. Lachaud, X. Provençal, and C. Reutenauer, "Lyndon + Christoffel = digitally convex," *Pattern Recognition*, vol. 42, no. 10, pp. 2239–2246, 2009.
- [27] R. Brons, "Linguistic methods for the description of a straight line on a grid," *Computer Graphics and Image Processing*, vol. 3, no. 1, pp. 48–62, 1974.
- [28] A. M. Bruckstein, "Self-similarity properties of digitized straight lines," *Vision Geometry, Contemporary Mathematics*, vol. 119, pp. 1–20, 1991.
- [29] L. Buzer, "A composite and quasi linear time method for digital plane recognition," in *Discrete Geometry for Computer Imagery, Proceedings of 13th International Conference, DGCI2006*, vol. 4245 of LNCS, Springer-Verlag, 2006, pp. 331–342.
- [30] A. Cayley and A. Forsyth, *The Collected Mathematical Papers of Arthur Cayley*. The University Press, 1898, vol. 1.
- [31] L. Chen, *Discrete Surfaces and Manifolds*. Scientific & Practical Computing, 2004.
- [32] O. Chum, "Two-view geometry estimation by random sample and consensus," PhD thesis, Czech Technical University, Prague, Czech Republic, 2005.
- [33] J. C. Ciria, A. de Miguel, E. Domínguez, A. R. Francés, and A. Quintero, "Local characterization of a maximum set of digital $(26, 6)$ -surfaces," in *Discrete Geometry for Computer Imagery, Proceedings of 12th International Conference, DGCI2005*, vol. 3429 of LNCS, Springer-Verlag, Berlin Heidelberg, 2005, pp. 161–171.
- [34] D. Coeurjolly, J.-O. Lachaud, and T. Roussillon, "Multigrid convergence of discrete geometric estimators," in *Digital Geometry Algorithms*, V. E. Brimkov and R. P. Barneva, Eds., ser. Lecture Notes in Computational Vision and Biomechanics. 2012, vol. 2, pp. 395–424.
- [35] D. Coeurjolly, I. Sivignon, F. Dupont, F. Feschet, and J.-M. Chassey, "On digital plane preimage structure," *Discrete Applied Mathematics*, vol. 151, no. 1–3, pp. 78–92, 2005.
- [36] D. Coeurjolly, V. Blot, and M. J. Col, "Quasi-affine transformation in 3-d: theory and algorithms," in *Combinatorial Image Analysis, 13th International Workshop, IWCIA 2009, Playa del Carmen, Mexico, November 24–27, 2009. Proceedings*, 2009, pp. 68–81.

- [37] D. Coeurjolly, A. Montanvert, and J.-M. Chassery, *Géométrie discrète et images numériques*, J.-M. C. David Coeurjolly Annick Montanvert, Ed. Hermès, 2007, Traité IC2, série signal et image. ISBN 13 : 978-2-7462-1643-3.
- [38] D. Cohen-Or and A. Kaufman, "Fundamentals of surface voxelization," *Graphical Models and Image Processing*, vol. 57, no. 6, pp. 453–461, 1995.
- [39] "Combinatorial structure of rigid transformations in 2d digital images," *Computer Vision and Image Understanding*, vol. 117, no. 4, pp. 393–408, 2013.
- [40] J. Conway and D. Smith, *On Quaternions and Octonions*, ser. Ak Peters Series. Taylor & Francis, 2003.
- [41] M. Couprie and G. Bertrand, "New characterization of simple points in 2d, 3d and 4d discrete spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 637–648, 2009.
- [42] —, "Simplicity surface: a new definition of surfaces in \mathbb{Z}^3 ," in *Vision Geometry VII, Proceedings of SPIE*, vol. 3454, 1998, pp. 40–51.
- [43] J. Cousty, L. Najman, Y. Kenmochi, and S. Guimarães, "Hierarchical segmentations with graphs: quasi-flat zones, minimum spanning trees, and saliency maps," *Journal of Mathematical Imaging and Vision*, vol. 60, no. 4, 479–502, 2018.
- [44] J. Cremona, "Letter to the editor," *American Mathematical Monthly*, vol. 94, no. 8, pp. 757–758, 1987.
- [45] G. Cristescu and L. Lupsa, *Non-Connected Convexities and Applications*. Dordrecht: Kluwer Academic Publishers, 2002.
- [46] X. Daragon, M. Couprie, and G. Bertrand, "Discrete surfaces and frontier orders," *Journal of Mathematical Imaging and Vision*, vol. 23, no. 1, pp. 379–399, 2005.
- [47] I. Debled-Rennesson, "Etude et reconnaissance des droites et plans discrets," PhD thesis, Université Louis Pasteur, Strasbourg, 1995.
- [48] I. Debled-Rennesson, J. L. Rémy, and J. Rouyer-Degli, "Detection of the discrete convexity of polyominoes," *Discrete Applied Mathematics*, vol. 125, no. 1, pp. 115–133, 2003.
- [49] I. Debled-Rennesson and J. Reveillès, "A linear algorithm for segmentation of digital curves," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 9, no. 4, pp. 635–662, 1995.

- [50] M. Dexet, D. Cœurjolly, and E. Andres, "Invertible polygonalization of 3d planar digital curves and application to volume data reconstruction," in *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin, *et al.*, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 514–523.
- [51] H. Dörksen-Reiter and I. Debled-Rennesson, "Convex and concave parts of digital curves," in *Geometric Properties for Incomplete data*, ser. Computational Imaging and Vision, R. Klette, R. Kozera, L. Noakes, and J. Weickert, Eds., vol. 31, Springer-Verlag, 2005, pp. 145–159.
- [52] H. Dörksen-Reiter and I. Debled-Rennesson, "A linear algorithm for polygonal representations of digital sets," in *IWCIA Proceedings*, ser. Lecture Notes in Computer Science, vol. 4040, 2006, pp. 307–319.
- [53] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, pp. 11–15, 1972.
- [54] J. Duval, "Factorizing words over an ordered alphabet," *Journal of Algorithms*, vol. 4, no. 4, pp. 363–381, 1983.
- [55] U. Eckhardt, "Digital lines and digital convexity," in *Digital and Image Geometry: Advanced Lectures*, G. Bertrand, A. Imiya, and R. Klette, Eds., 2001, pp. 209–228.
- [56] P. V. Erwan Plougonven Dominique Bernard, "Quantitative analysis of the deformation of polypropylene foam under dynamic loading," in *Proceedings of the SPIE, Developments in X-Ray Tomography V*, vol. 6318, 2006, p. 631 813.
- [57] A. Esnard, J.-O. Lachaud, and A. Vialard, "Discrete deformable boundaries for 3d image segmentation," LaBRI, University of Bordeaux 1, Tech. Rep. 1270-02, 2002.
- [58] F. Feschet and L. Tougne, "Optimal time computation of the tangent of a discrete curve: application to the curvature," in *Discrete Geometry for Computer Imagery, Proceedings of DGCI*, ser. Lecture Notes in Computer Science, vol. 1568, Springer, 1999, pp. 31–40.
- [59] F. Feschet, "Fast guaranteed polygonal approximations of closed digital curves," in *SCIA Proceedings*, ser. Lecture Notes in Computer Science, vol. 3540, 2005, pp. 910–919.
- [60] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

- [61] P. Flajolet and B. Vallée, "Continued fractions, comparison algorithms, and fine structure constants," in *Constructive, Experimental, and Nonlinear Analysis*, M. Théra Ed., vol. Volume in the honour of Jonathan Borwein, CMS Conf. Proc., 27, Amer. Math. Soc., Providence, 2000, pp. 53–82.
- [62] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1990.
- [63] G. D. da Fonseca and D. M. Mount, "Approximate range searching: the absolute model," *Computational Geometry*, vol. 43, pp. 434–444, 2010.
- [64] J. Françon, "Discrete combinatorial surfaces," *Graphical Models and Image Processing*, vol. 57, no. 1, pp. 20–26, 1995.
- [65] —, "Sur la topologie d'un plan arithmétique," *Theoretical Computer Science*, vol. 156, pp. 159–176, 1996.
- [66] M. Fréchet, "Sur quelques points du calcul fonctionnel," *Rendiconti del Circolo Matematico di Palermo*, vol. 22, pp. 1–74, 1906.
- [67] H. Freeman, "Computer processing of line-drawing images," *ACM Computing Survey*, vol. 6, no. 1, pp. 57–97, Mar. 1974.
- [68] A. Gajentaan and M. H. Overmars, "On a class of $O(n^2)$ problems in computational geometry," *Comput. Geom.*, vol. 5, pp. 165–185, 1995.
- [69] A. V. Gelder and J. Wilhelms, "Topological considerations in isosurface generation," *ACM Transactions on Graphics*, vol. 13, pp. 337–375, 1994.
- [70] Gérard, Yan and Provot, Laurent and Feschet, Fabien, "Introduction to digital level layers," in *Discrete Geometry for Computer Imagery*, ser. LNCS, Springer, 2011, pp. 83–94.
- [71] S. Guimarães, Y. Kenmochi, J. Cousty, Z. Patrocínio, and L. Najman, "Hierarchizing graph-based image segmentation algorithms relying on region dissimilarity: the case of the Felzenszwalb-Huttenlocher method," *Mathematical Morphology - Theory and Applications*, vol. 2, pp. 55–75, 2017.
- [72] D. Halperin, "Arrangements," in *Handbook of Discrete and Computational Geometry*, 2nd Ed. J. E. Goodman and J. O'Rourke, Eds., Chapman and Hall/CRC, 2004, pp. 529–562.
- [73] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2003.
- [74] H. J. A. M. Heijmans and C. Ronse, "The algebraic basis of mathematical morphology. i dilations and erosions," *CVGIP: Image Understanding*, vol. 50, no. 3, pp. 245–295, 1990.
- [75] B. K. P. Horn, "Extended gaussian images," *Proceedings of the IEEE*, vol. 72, no. 12, pp. 1671–1686, 1984.

- [76] P. Hough, "Machine analysis of bubble chamber pictures," in *Proc. Int. Conf. High Energy Accelerators and Instrumentation*, 1959.
- [77] C. Hundt, "Affine image matching is uniform TC0-complete," in *Combinatorial Pattern Matching, CPM, Proceedings*, ser. Lecture Notes in Computer Science, vol. 6129, Springer, 2010, pp. 13–25.
- [78] C. Hundt and M. Liśkiewicz, "Combinatorial bounds and algorithmic aspects of image matching under projective transformations," in *Mathematical Foundations of Computer Science, MFCS, Proceedings*, ser. Lecture Notes in Computer Science, vol. 5162, Springer, 2008, pp. 395–406.
- [79] C. Hundt, M. Liśkiewicz, and N. Ragnar, "A combinatorial geometrical approach to two-dimensional robust pattern matching with scaling and rotation," *Theoretical Computer Science*, vol. 410, no. 51, pp. 5317–5333, 2009.
- [80] C. Hundt and M. Liskiewicz, "On the Complexity of Affine Image Matching,," in *STACS*, ser. Lecture Notes in Computer Science, vol. 4393, Springer, 2007, pp. 284–295.
- [81] A. Imiya and U. Eckhardt, "The euler characteristics of discrete objects and discrete quasi-objects," *Computer Vision and Image Understanding*, vol. 75, no. 3, pp. 307–318, 1999.
- [82] Z. Jelonek and K. Kurdyka, "Quantitative generalized Bertini-Sard theorem for smooth affine varieties," *Discrete & Computational Geometry*, vol. 34, no. 4, pp. 659–678, 2005.
- [83] P. P. Jonker and S. Svensson, "The generation of n dimensional shape primitives," in *Discrete Geometry for Computer Imagery, Proceedings of 11th International Conference, DGCI2003*, vol. 2886 of LNCS, Springer-Verlag Heidelberg, 2003, pp. 420–433.
- [84] K. Kanatani, *Understanding Geometric Algebra: Hamilton, Grassmann, and Clifford for Computer Vision and Graphics*. CRC Press, 2015.
- [85] Y. Kenmochi, "Topological characterization of simple points by complex collapsibility," in *Meeting on Image Recognition and Understanding (MIRU'06)*, Japan.
- [86] Y. Kenmochi, L. Buzer, A. Sugimoto, and I. Shimizu, "Digital planar surface segmentation using local geometric patterns," in *Discrete Geometry for Computer Imagery: Proceedings of 14th IAPR International Conference, DGCI 2008, Lyon, France*, D. Coeurjolly, I. Sivignon, L. Tougne, and F. Dupont, Eds.
- [87] —, "Discrete plane segmentation and estimation from a point cloud using local geometric patterns," *International Journal of Automation and Computing*, vol. 5, no. 3, pp. 246–256, 2008.

- [88] Y. Kenmochi, L. Buzer, and H. Talbot, in *20th International Conference on Pattern Recognition (ICPR)*, Istanbul, Turkey.
- [89] Y. Kenmochi and A. Imiya, "Combinatorial boundary of a 3d lattice point set," *Journal of Visual Communication and Image Representation*, vol. 17, no. 4, pp. 738–766, 2006.
- [90] —, "Combinatorial topologies for discrete planes," in *Discrete Geometry for Computer Imagery: Proceedings of 11th International Conference, DGCI 2003, Naples, Italy*, I. Nyström, G. Sanniti di Baja, and S. Svensson, Eds.
- [91] —, "Naive planes as discrete combinatorial surfaces," in *Discrete Geometry for Computer Imagery: Proceedings of 9th International Conference, DGCI 2000, Uppsala, Sweden*, G. Borgefors, I. Nyström, and G. S. di Baja, Eds.
- [92] Y. Kenmochi, A. Imiya, and A. Ichikawa, "Discrete combinatorial geometry," *Pattern Recognition*, vol. 30, no. 10, pp. 1719–1728, 1997.
- [93] Y. Kenmochi, K. Kotani, and A. Imiya, "Marching cubes method with connectivity," in *Proceedings of 1999 International Conference on Image Processing (ICIP 99)*, Kobe, Japan, vol. 4, 1999, pp. 361–365.
- [94] Y. Kenmochi and K. Ushigusa, "Complicial thinning for three-dimensional digital images: recursive definition of reductive points and their topological characteristics," in *Meeting on Image Recognition and Understanding (MIRU'05)*, 2005, pp. 120–127.
- [95] B. Kerautret, P. Ngo, Y. Kenmochi, and A. Vacavant, "Greyscale image vectorization from geometric digital contour representations," in *Discrete Geometry for Computer Imagery: 20th IAPR International Conference, DGCI 2017, Vienna, Austria, September 19 – 21, 2017, Proceedings*, W. G. Kropatsch, N. M. Artner, and I. Janusch, Eds., Cham: Springer International Publishing, 2017, pp. 319–331.
- [96] E. Khalimsky, "Topological structures in computer science," *Journal of Applied Mathematics and Simulation*, vol. 1, no. 1, pp. 25–40, 1987.
- [97] E. Khalimsky, R. Kopperman, and P. R. Meyer, "Computer graphics and connected topologies on finite ordered sets," *Topology and its Applications*, vol. 36, no. 1, pp. 1–17, 1990.
- [98] C. E. Kim, "On the cellular convexity of complexes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, no. 6, pp. 617–625, 1981.
- [99] C. E. Kim and A. Rosenfeld, "Digital straight lines and convexity of digital regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, no. 2, pp. 149–153, 1982.

- [100] R. Klette and J. Žunić, "Multigrid convergence of calculated features in image analysis," *Journal of Mathematical Imaging and Vision*, vol. 13, pp. 175–191, 2000.
- [101] R. Klette and A. Rosenfeld, *Digital Geometry: Geometric Methods for Digital Picture Analysis*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004.
- [102] —, "Digital straightness—a review," *Discrete Applied Mathematics*, vol. 139, no. 1, pp. 197–230, 2004.
- [103] D. E. Knuth, T. Larrabee, and P. M. Roberts, *Mathematical Writing*. Washington, DC, USA: Mathematical Association of America, 1989.
- [104] T. Y. Kong, "A digital fundamental group," *Computers Graphics*, vol. 13, no. 2, pp. 159–166, 1989.
- [105] T. Y. Kong and A. Rosenfeld, "Digital topology: introduction and survey," *Computer Vision, Graphics, and Image Processing*, vol. 48, pp. 357–393, 1989.
- [106] K. Koster and M. Spann, "Mir: an approach to robust clustering-application to range image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 5, pp. 430–444, 2000.
- [107] V. Kovalevsky and S. Fuchs, "Theoretical and experimental analysis of the accuracy of perimeter estimates," in *Robust Computer Vision*, W. Förster and S. Ruwiedel, Eds. 1992, pp. 218–242.
- [108] V. A. Kovalevsky, "Finite topology as applied to image analysis," *Computer Vision, Graphics, and Image Processing*, vol. 46, no. 2, pp. 141–161, 1989.
- [109] K. Kurdyka, P. Orro, S. Simon, *et al.*, "Semialgebraic Sard theorem for generalized critical values," *Journal of differential geometry*, vol. 56, no. 1, pp. 67–92, 2000.
- [110] J.-O. Lachaud and A. Montanvert, "Continuous analogs of digital boundaries: a topological approach to iso-surfaces," *Graphical Models and Image Processing*, vol. 62, pp. 129–164, 2000.
- [111] J.-O. Lachaud and B. Thibert, "Properties of gauss digitized shapes and digital surface integration," *Journal of Mathematical Imaging and Vision*, vol. 54, no. 2, pp. 162–180, 2016.
- [112] S. Laine, "A topological approach to voxelization," *Computer Graphic Forum*, vol. 32, no. 4, pp. 77–86, 2013.
- [113] L. J. Latecki, C. Conrad, and A. Gross, "Preserving topology by a digitization process," *Journal of Mathematical Imaging and Vision*, vol. 8, no. 2, pp. 131–159, 1998.

- [114] L. J. Latecki, U. Eckhardt, and A. Rosenfeld, "Well-composed sets," *Computer Vision and Image Understanding*, vol. 5, no. 1, pp. 70–83, 1995.
- [115] L. J. Latecki, *Discrete Representation of Spatial Objects in Computer Vision*, ser. Computational Imaging and Vision. Springer, 1998, vol. 11.
- [116] "Local configurations in discrete combinatorial surfaces," *Image and Vision Computing*, vol. 25, no. 10, pp. 1657–1670, 2007.
- [117] C. Lohou and G. Bertrand, "Poset approach to 3d parallel thinning," in *SPIE Vision Geometry VIII*, vol. 3811, 1999, pp. 45–56.
- [118] W. E. Lorensen and H. E. Cline, "Marching cubes: a high-resolution 3d surface construction algorithm," *Computer Graphics (SIGGRAPH '87)*, vol. 21, no. 4, pp. 163–169, 1987.
- [119] C. R. F. Maunder, *Algebraic Topology*. Dover, 1996.
- [120] L. Mazo, N. Passat, M. Couprie, and C. Ronse, "Digital imaging: a unified topological framework," *Journal of Mathematical Imaging and Vision*, vol. 44, no. 1, pp. 19–37, 2012.
- [121] —, "Topology on digital label images," *Journal of Mathematical Imaging and Vision*, vol. 44, no. 3, pp. 254–281, 2012.
- [122] L. Mazo, N. Passat, M. Couprie, and C. Ronse, "Paths, homotopy and reduction in digital images," *Acta Applicandae Mathematicae*, vol. 113, no. 2, pp. 167–193, 2011.
- [123] A. A. Melkman, "On-line construction of the convex hull of a simple polyline," *Information Processing Letters*, vol. 25, no. 1, pp. 11–12, 1987.
- [124] M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry*. Reading, MA: MIT Press, 1969.
- [125] E. E. Moise, *Geometric Topology in Dimensions 2 and 3*. Springer-Verlag, 1977.
- [126] D. G. Morgenthaler and A. Rosenfeld, "Surfaces in three-dimensional digital images," *Information and Control*, vol. 51, pp. 227–247, 1981.
- [127] B. Mourrain, J. P. Tecourt, and M. Teillaud, "On the computation of an arrangement of quadrics in 3D," *Computational Geometry*, vol. 30, no. 2, pp. 145–164, 2005.
- [128] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
- [129] N. H. Mustafa and K. R. Varadarajan, "Epsilon-approximations and epsilon-nets," in *Handbook on Discrete and Computational Geometry*, 3rd edition, 2017.

- [130] L. Najman and H. Talbot, *Mathematical Morphology: From Theory to Applications*. ISTE/J. Wiley & Sons, 2010.
- [131] P. Ngo, Y. Kenmochi, I. Debled-Rennesson, and N. Passat, "Convexity-preserving rigid motions of 2d digital objects," in *Discrete Geometry for Computer Imagery: 20th IAPR International Conference, DGCI 2017, Vienna, Austria, September 19 – 21, 2017, Proceedings*, W. G. Kropatsch, N. M. Artner, and I. Janusch, Eds., Cham: Springer International Publishing, 2017, pp. 69–81.
- [132] P. Ngo, Y. Kenmochi, N. Passat, and H. Talbot, "Sufficient conditions for topological invariance of 2d images under rigid transformations," in *Discrete Geometry for Computer Imagery: Proceedings of 17th IAPR International Conference, DGCI 2013, Seville, Spain, R. Gonzalez-Diaz, M.-J. Jimenez, and B. Medrano, Eds.*
- [133] —, "Topology-preserving conditions for 2d digital images under rigid transformations," *Journal of Mathematical Imaging and Vision*, vol. 49, no. 2, pp. 418–433, 2014.
- [134] P. Ngo, Y. Kenmochi, A. Sugimoto, H. Talbot, and N. Passat, "Discrete rigid registration: a local graph-search approach," *Discrete Applied Mathematics*, vol. 216, Part 2, pp. 461–481, 2017.
- [135] P. Ngo, H. Nasser, and I. Debled-Rennesson, "Efficient dominant point detection based on discrete curve structure," in *Proceedings of IWCI, ser. LNCS*, vol. 9448, 2015, pp. 143–156.
- [136] P. Ngo, N. Passat, Y. Kenmochi, and I. Debled-Rennesson, "Convexity invariance of voxel objects under rigid motions," in *Proceedings of International Conference on Pattern Recognition, ICPR 2018, Beijing, August 20–24, 2018, Beijing, China, 2018*.
- [137] —, "Geometric preservation of 2D digital objects under rigid motions," to appear in *Journal of Mathematical Imaging and Vision*, 2018.
- [138] P. Ngo, N. Passat, Y. Kenmochi, and H. Talbot, "Topology-preserving rigid transformation of 2d digital images," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 885–897, 2014.
- [139] T. P. Nguyen and I. Debled-Rennesson, "A discrete geometry approach for dominant point detection," *Pattern Recognition*, vol. 44, no. 1, pp. 32–44, 2011.
- [140] G. M. Nielson and B. Hamann, "The asymptotic decider: resolving the ambiguity in marching cubes," in *Visualization '91, Proceedings of IEEE Conference on Visualization*, 1991, pp. 83–91.
- [141] F. S. Nooruddin and G. Turk, "Simplification and repair of polygonal models using volumetric techniques," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 2, pp. 191–205, 2003.

- [142] B. Nouvel and E. Rémila, "Configurations induced by discrete rotations: Periodicity and quasi-periodicity properties," *Discrete Applied Mathematics*, vol. 147, no. 2–3, pp. 325–343, 2005.
- [143] —, "On colorations induced by discrete rotations," in *DGCI, Proceedings*, ser. Lecture Notes in Computer Science, vol. 2886, Springer, 2003, pp. 174–183.
- [144] B. Nouvel and E. Rémila, "Characterization of bijective discretized rotations," in *Combinatorial Image Analysis, 10th International Workshop, IWCIA 2004, Auckland, New Zealand, December 1–3, 2004, Proceedings*, 2004, pp. 248–259.
- [145] E. Ovreiu, "Accurate 3D mesh simplification," Theses, INSA de Lyon, Dec. 2012.
- [146] N. Passat and L. Mazo, "An introduction to simple sets," *Pattern Recognition Letters*, vol. 30, no. 15, pp. 1366–1377, 2009.
- [147] T. Pavlidis, *Algorithms for Graphics and Image Processing*. Berlin: Springer and Rockville: Computer Science Press, 1982.
- [148] K. Pluta, Y. Kenmochi, N. Passat, H. Talbot, and P. Romon, "Topological alterations of 3D digital images under rigid transformations," Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge UMR 8049, Technical Report, Nov. 2014.
- [149] K. Pluta, G. Moroz, Y. Kenmochi, and P. Romon, "Quadric arrangement in classifying rigid motions of a 3D digital image," in *Proceedings of the 18th International Workshop on Computer Algebra in Scientific Computing, CASC 2016, Bucharest, Romania*.
- [150] K. Pluta, P. Romon, Y. Kenmochi, and N. Passat, "Bijectivity certification of 3d digitized rotations," in *Computational Topology in Image Context: Proceedings of 6th International Workshop, CTIC 2016, Marseille, France*, A. Bac and J.-L. Mari, Eds.
- [151] —, "Honeycomb geometry: rigid motions on the hexagonal grid," in *Discrete Geometry for Computer Imagery: 20th IAPR International Conference, DGCI 2017, Vienna, Austria, September 19 – 21, 2017, Proceedings*, W. G. Kropatsch, N. M. Artner, and I. Janusch, Eds., Cham: Springer International Publishing, 2017, pp. 33–45.
- [152] K. Pluta, T. Roussillon, D. Coeurjolly, P. Romon, Y. Kenmochi, and V. Ostromoukhov, "Characterization of bijective digitized rotations on the hexagonal grid," *Journal of Mathematical Imaging and Vision*, vol. 60, no. 5, pp. 707–716, 2018.
- [153] K. Pluta, P. Romon, Y. Kenmochi, and N. Passat, "Bijective digitized rigid motions on subsets of the plane," *Journal of Mathematical Imaging and Vision*, vol. 59, no. 1, pp. 84–105, 2017.

- [154] M. Postolski, M. Janaszewski, Y. Kenmochi, and J.-O. Lachaud, in *21st International Conference on Pattern Recognition (ICPR)*, Tsukuba, Japan.
- [155] P. J. Rabier, "Ehresmann fibrations and Palais-Smale conditions for morphisms of Finsler manifolds," *Annals of Mathematics*, pp. 647–691, 1997.
- [156] J.-P. Reveillès, "Combinatorial pieces in digital lines and planes," in *Vision Geometry IV*, vol. 2573, SPIE, 1995, pp. 23–34.
- [157] J.-P. Reveillès, "Géométrie discrète, calcul en nombres entiers et algorithmique," PhD thesis, Université Louis Paster, Dec. 1991.
- [158] V. Robins, P. J. Wood, and A. P. Sheppard, "Theory and algorithms for constructing discrete morse complexes from grayscale digital images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1646–1658, 2011.
- [159] P. Romon, *Introduction à la géométrie différentielle discrète*, ser. Références sciences. Ellipses, Dec. 2013, p. 216.
- [160] C. Ronse, "A topological characterization of thinning," *Theoretical Computer Science*, vol. 43, no. 1, pp. 31–41, 1986.
- [161] C. Ronse and H. J. A. M. Heijmans, "The algebraic basis of mathematical morphology: ii. openings and closings," *CVGIP: Image Understanding*, vol. 54, no. 1, pp. 74–97, 1991.
- [162] K. H. Rosen, *Elementary Number Theory and its Applications*, 3rd. Addison-Wesley, 1992.
- [163] A. Rosenfeld, "Adjacency in digital pictures," *Information and Control*, vol. 26, no. 1, pp. 24–33, 1974.
- [164] —, "Digital topology," *The American Mathematical Monthly*, vol. 86, no. 8, pp. 621–630, 1979.
- [165] A. Rosenfeld, T. Y. Kong, and A. Nakamura, "Topology-preserving deformations of two-valued digital pictures," *Graphical Models and Image Processing*, vol. 60, no. 1, pp. 24–34, 1998.
- [166] C. P. Rourke and B. J. Sanderson, *Introduction to Piecewise-Linear Topology*. Berlin Heidelberg New York: Springer-Verlag, 1972.
- [167] P. J. Rousseeuw, "Least median of squares regression," *Journal of the American statistical association*, vol. 79, no. 388, pp. 871–880, 1984.
- [168] T. Roussillon and D. Coeurjolly, "Characterization of bijective discretized rotations by Gaussian integers," LIRIS UMR CNRS 5205, Research Report, Jan. 2016.
- [169] T. Roussillon and I. Sivignon, "Faithful polygonal representation of the convex and concave parts of a digital curve," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2693–2700, 2011.

- [170] P. Sankar, "Grid intersect quantization schemes for solid object digitization," *Computer Graphics and Image Processing*, vol. 8, no. 1, pp. 25–42, 1978.
- [171] L. A. Santaló and M. Kac, *Integral geometry and geometric probability*, ser. Cambridge mathematical library. Cambridge: Cambridge University Press, 2004, 1re édition London : Addison-Wesley, 1976.
- [172] J. Serra, *Image Analysis and Mathematical Morphology*. Orlando, FL, USA: Academic Press, 1983.
- [173] H.-Y. Shum, K. Ikeuchi, and R. Reddy, "Principal component analysis with missing data and its application to polyhedral object modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 9, pp. 854–867, Sep. 1995.
- [174] P. Singla and J. L. Junkins, *Multi-resolution methods for modeling and control of dynamical systems*. CRC Press, 2008.
- [175] I. Sivignon, R. Breton, F. Dupont, and E. Andres, "Discrete analytical curve reconstruction without patches," *Image and Vision Computing*, vol. 23, no. 2, pp. 191–202, 2005.
- [176] I. Sivignon and D. Coeurjolly, "Minimum decomposition of a digital surface into digital plane segments is np-hard," *Discrete Applied Mathematics*, vol. 157, no. 3, 558–570, 2008.
- [177] I. Sivignon, F. Dupont, and J.-M. Chassery, "Discrete surfaces segmentation into discrete planes," in *Proceedings of IWCIA*, vol. 3322 of LNCS, Springer-Verlag, 2004, pp. 458–473.
- [178] I. Sivignon, "A Near-Linear Time Guaranteed Algorithm for Digital Curve Simplification under the Fréchet Distance," in *Discrete Geometry for Computer Imagery: Proceedings of DGCI 2011*, 2011, pp. 333–345.
- [179] I. Sivignon, F. Dupont, and J.-M. Chassery, "Decomposition of a three-dimensional discrete object surface into discrete plane pieces," *Algorithmica*, vol. 38, no. 1, pp. 25–43, 2004.
- [180] J. Sklansky, "Recognition of convex blobs," *Pattern Recognition*, vol. 2, no. 1, pp. 3–10, 1970.
- [181] O. Sorkine and M. Alexa, "As-rigid-as-possible surface modeling," in *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, ser. SGP '07, Eurographics Association, 2007, pp. 109–116.
- [182] D. L. Souvaine and J. M. Steele, "Time- and space-efficient algorithms for least median of squares regression," *Journal of the American Statistical Association*, vol. 82, no. 399, pp. 794–801, 1987.

- [183] I. Stamos and P. K. Allen, "3d model construction using range and image data," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2000, pp. 531–536.
- [184] P. Stelldinger and U. Köthe, "Towards a general sampling theory for shape preservation," *Image and Vision Computing*, vol. 23, no. 2, pp. 237–248, 2005.
- [185] P. Stelldinger, L. J. Latecki, and M. Siqueira, "Topological equivalence between a 3d object and the reconstruction of its digital image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 126–140, 2007.
- [186] J. Stillwell, *Classical Topology and Combinatorial Group Theory*. Springer, 1993.
- [187] I. Stojmenovic and R. Tomic, "Digitization schemes and the recognition of digital straight lines, hyperplanes, and flats in arbitrary dimensions," *Contemporary Mathematics*, vol. 119, pp. 197–212, 1991.
- [188] L. Tarsissi, "Balance properties on christoffel words and applications," PhD thesis, Université Grenoble Alpes, 2017.
- [189] Y. Thibault, "Rotations in 2d and 3d discrete spaces," PhD thesis, University Paris-Est, 2010.
- [190] Y. Thibault, A. Sugimoto, and Y. Kenmochi, "3d discrete rotations using hinge angles," *Theoretical Computer Science*, vol. 412, no. 15, pp. 1378–1391, 2011.
- [191] J. Toriwaki, *Three-dimensional digital image processing (in Japanese)*. Shokodo, 2002.
- [192] J.-L. Toutant, E. Andres, G. Largeteau-Skapin, and R. Zrour, "Implicit digital surfaces in arbitrary dimensions," in *Discrete Geometry for Computer Imagery*, E. Barcucci, A. Frosini, and S. Rinaldi, Eds., Springer International Publishing, 2014, pp. 332–343.
- [193] P. Veelaert, "On the flatness of digital hyperplanes," *Journal of Mathematical Imaging and Vision*, vol. 3, no. 2, pp. 205–221, 1993.
- [194] J. Vince, *Quaternions for Computer Graphics*. Springer, 2011.
- [195] J. Vittone and J.-M. Chassery, " (n, m) -cubes and farey nets for naive planes understanding," in *Discrete Geometry for Computer Imagery, Proceedings of International Conference, DGCI'99*, ser. LNCS, vol. 1568, Springer-Verlag, 1999, pp. 76–87.
- [196] J. Vittone and J. Chassery, "Recognition of digital naive planes and polyhedrization," in *Discrete Geometry for Computer Imagery, Proceedings of International Conference, DGCI 2000*, ser. Lecture Notes in Computer Science, vol. 1953, 2000, pp. 296–307.
- [197] K. Voss, *Discrete images, objects, and functions in Z^n* , ser. Algorithms and combinatorics. Springer-Verlag, 1993.

- [198] G. Wyvill, C. McPheeters, and B. Wyvill, "Data structure for soft objects," *The Visual Computer*, vol. 2, no. 4, pp. 227–234, 1986.
- [199] A. Yilmaz, O. Javed, and M. Shah, *ACM computing surveys*, no. 4,
- [200] N. Yokoya and M. D. Levine, "Range image segmentation based on differential geometry: a hybrid approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp. 643–649, 1989.
- [201] D. Zhao and X. Zhang, "Range-data-based object surface segmentation via edges and critical points," *IEEE Transactions on Image Processing*, vol. 6, no. 6, pp. 826–830, 1997.
- [202] G. M. Ziegler, *Lectures on Polytopes*. Springer, 1998.
- [203] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003.
- [204] R. Zrou, Y. Kenmochi, H. Talbot, L. Buzer, Y. Hamam, I. Shimizu, and A. Sugimoto, "Optimal Consensus Set for Digital Line Fitting," in *Combinatorial Image Analysis: Proceedings of 13th International Workshop, IWCIA 2009, Playa del Carmen, Mexico*.
- [205] R. Zrou, Y. Kenmochi, H. Talbot, L. Buzer, Y. Hamam, I. Shimizu, and A. Sugimoto, in *2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*.
- [206] —, "Optimal consensus set for digital line and plane fitting," *International Journal of Imaging Systems and Technology*, vol. 21, no. 1, pp. 45–57, 2011.
- [207] R. Zrou, Y. Kenmochi, H. Talbot, I. Shimizu, and A. Sugimoto, "Combinatorial optimization for fitting of digital line and plane," in *International Workshop on Computer Vision and Its Application to Image Media Processing, Satellite Workshop of the 3rd Pacific-Rim Symposium on Image and Video Technology*, Japan, Jan. 2009, pp. 35–41.

