



## Research Article

## Open Access

Ranita Biswas\*, Gaëlle Largeteau-skapin, Rita Zrour, and Eric Andres

# Digital Objects in Rhombic Dodecahedron Grid

<https://doi.org/10.1515/mathm-2020-0106>

Received February 4, 2020; accepted November 17, 2020

**Abstract:** Rhombic dodecahedron is a space filling polyhedron which represents the close packing of spheres in 3D space and the Voronoi structures of the face centered cubic (FCC) lattice. In this paper, we describe a new coordinate system where every 3-integer coordinates grid point corresponds to a rhombic dodecahedron centroid. In order to illustrate the interest of the new coordinate system, we propose the characterization of 3D digital plane with its topological features, such as the interrelation between the thickness of the digital plane and the separability constraint we aim to obtain. We also present the characterization of 3D digital lines and study it as the intersection of multiple digital planes. Characterization of 3D digital sphere with relevant topological features is proposed as well along with the 48-symmetry appearing in the new coordinate system.

**Keywords:** rhombic dodecahedron, face-centered cubic (FCC) grid, 3D coordinate system, 3D digital line, digital plane, digital sphere

**MSC:** 52C

## 1 Introduction

The cubic grid  $\mathbb{Z}^3$  is the most frequently used grid for three-dimensional images. However, in recent times, non standard, three-dimensional grids received a lot of interest, particularly with applications in networks [35], image processing [15, 27, 36], computer vision [17], and many other fields. Among non standard grids we can cite face-centered cubic (FCC), body-centered cubic (BCC), honeycomb [8], and diamond grids [32].

Unconventional grids are steadily gaining popularity in medical image data processing, specially 3D image reconstruction. The use of optimal sampling grids are more efficient with regards to the number of voxels and lead to significantly fewer samples without decreasing the quality of the reconstruction [14, 24, 26, 33]. In [9], authors proposed to repair 3D images by associating the FCC grid with the cubic grid and illustrated applications on two tasks related to the repaired image, boundary reconstruction and computation of its Euler characteristic.

In a more theoretical perspective, several researches discussed coordinate system on the 3D grids such as [23] for cubic grids, [28, 29] for hexagonal and triangular grids, [11] for BCC grids and, [10, 12, 37] for FCC grids. Line tracing and representations of other 3D objects in generalized regular grids including BCC and

---

\*Corresponding Author: **Ranita Biswas:** Institute of Science and Technology (IST) Austria, 3400 Klosterneuburg, Austria, E-mail: ranita.biswas@ist.ac.at

**Gaëlle Largeteau-skapin:** University of Poitiers, Laboratory XLIM, ASALI, UMR CNRS 7252, BP 30179, F-86962 Futuroscope Chasseneuil, France, E-mail: gaelle.largeteau.skapin@univ-poitiers.fr

**Rita Zrour:** University of Poitiers, Laboratory XLIM, ASALI, UMR CNRS 7252, BP 30179, F-86962 Futuroscope Chasseneuil, France, E-mail: rita.zrour@univ-poitiers.fr

**Eric Andres:** University of Poitiers, Laboratory XLIM, ASALI, UMR CNRS 7252, BP 30179, F-86962 Futuroscope Chasseneuil, France, E-mail: eric.andres@univ-poitiers.fr

0 This paper is an extended version of the paper titled "Rhombic Dodecahedron Grid – Coordinate System and 3D Digital Object Definitions" [7] presented in DGCI'2019.

FCC grids are discussed in [16, 19, 20]. Space filled entirely by rhombic dodecahedra may be sliced by specific planes to reveal patterns of hexagons. In [22], authors proposed a generation algorithm for discrete spheres on the FCC grid using layered discrete annuli on hexagonal grids. The general idea there is to propose digital primitive generation algorithms that are compatible with additive manufacturing techniques.

The face centered cubic grid is the densest possible packing in three dimensions [37]. The shape of the cells in an FCC grid is a rhombic dodecahedron, which is a space filling polyhedron described by 12 faces, 24 edges, and 14 vertices. The FCC grid can be seen as the union of four disjoint cubic grids [10, 21, 30, 31] which led to various coordinate systems. In such a rhombic dodecahedron grid coordinate systems, sometimes distinction is made between different cells and not all the integer coordinates correspond to a cell. These make it difficult to use geometric transforms, e.g. rotation, in such coordinate systems.

This paper is an extended version of the work presented in [7] where a new non-orthogonal basis coordinate system is proposed. In this paper, we present a slightly simplified version of such a coordinate system. This proposed system provides integer coordinates for rhombic dodecahedron centroids covering the whole set of integer points. The interest of such a system would be to facilitate work on such a grid. This will be highly beneficial considering the usage of this grid in image reconstruction and similar applications due to its correspondence with hexagonal grids in 2D, and also in additive manufacturing using spherical cells utilizing the emulation of densest sphere packing in space. The interest of the new coordinate system is illustrated in this paper with the (topological) characterization of various classes of digital planes, 3D lines, and spheres in the rhombic dodecahedron grid (RDG). The coordinate system offers a 48-symmetry which can be used to efficiently construct digital spheres and other symmetric objects.

### Organization of the paper

In Section 2, we present some preliminaries and basics as well as the rhombic dodecahedron grid and the existing Cartesian coordinate system [10] on it. In section 3, we detail our coordinate system and we present some of its properties. In Section 4, we define the digital planes, 3D lines, and spheres in this coordinate system and explore the required thickness for different models to ensure desired connectivity and separation of space. We conclude and present future perspectives in Section 5.

## 2 Preliminaries

In this section, we recall some of the basic terminologies and definitions relevant to our problem. We put forward the general definitions of digital plane, 3D digital line, and digital sphere, and the different topological models already established in literature for conventional cubic grid (CCG). We also talk about the formal notion of grids and coordinate systems and present the concept and importance of designing a rhombic dodecahedron grid.

We consider a 3D space. We call integer points the points that have integer coordinates on all the three axes. Let us denote  $d_2$  as the Euclidean distance in the regular Cartesian space. The standard definitions of digital planes, 3D digital lines, and digital spheres are as follows.

**Definition 1** (Digital Plane in CCG [2]). *The digital plane  $P$  corresponding to the Euclidean plane  $ax + by + cz + d = 0$ , with the thickness of the digital plane as  $\rho$ , is the set of integer coordinate points  $p(x, y, z)$  verifying the following inequalities:*

$$-\frac{\rho}{2} \leq ax + by + cz + d < \frac{\rho}{2}.$$

This is also called analytical plane with the thickness  $\rho$  ensuring the required topological properties in terms of connectivity and separability.

Originally, the thickness was used to characterize the connectivity of digital lines leading to so called naive and standard digital lines [34]. However, for higher dimensional objects, e.g. planes and spheres in 3D,

the thickness characterizes the separation of space into two separate connected components (in 2D, separation and connectivity are equivalent for digital lines, but not in higher dimensions anymore).

Different values of  $\rho$  in this definition leads to different digitization models, e.g. 2-separating, 0-separating, etc. A digital object is  $l$ -separating if its complements are not  $l$ -connected. Hence, a digital plane is 2-separating if the complementary half-spaces are not 2-connected. Note that, 0-, 1-, and 2-connectivity here refer to the classical 26-, 18-, 6-connectivity respectively for the cubic grid. Please consult [2, 3] for more details on digital hyperplanes and hyperspheres in cubic grid.

**Definition 2** (Digital Line in CCG [13]). *I. Debled-Rennesson proposed a 3D digital line  $L$  with direction vector  $(a, b, c)$  and  $a \geq b \geq c$  definition as the set of points  $(x, y, z)$  from  $\mathbb{Z}^3$  verifying the following Diophantine inequalities:*

$$\begin{cases} \mu \leq cx - az < \mu + \rho \\ \mu' \leq bx - ay < \mu' + \rho' \end{cases}$$

with  $\mu, \mu', \rho, \rho' \in \mathbb{Z}$ .  $\rho$  and  $\rho'$  are called arithmetical thicknesses of the line  $L$ .

It is noteworthy that with this definition, a 3D digital line represents the intersection between two digital planes, each one being the orthogonal extrusion of a 2D digital line included in one of the coordinate planes. The definitions for coefficients ordered in a different way are obtained by permuting  $x, y, z$  as well as the coefficients.

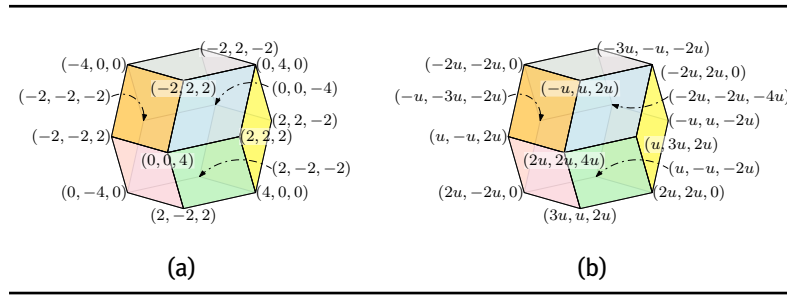
In [1, 4], authors gave the analytical definition of the supercover 3D lines based on three 2D line projections (respectively in the planes  $xy, yz,$  and  $zx$ ) or three 3D planes. Note that, the supercover of an object  $A$  corresponds to all the digital cells that are intersected by  $A$ . The authors proved that two planes are, generally, not enough and this is the major difference with Debled-Rennesson's definition 2. They stated that the intersection of three digital planes defines a digital line in CCG and create a symmetry that does not exist in the classical definition. In this paper, we follow the approach from [1] and use the correct number of digital planes to ensure exact digitization of the 3D line in RDG.

Several literatures describe incremental line drawing algorithms on FCC grid and other non-orthogonal regular grids. In [16], the authors proposed a line drawing algorithm on the FCC grid following the 3D cubic Bresenham algorithm. Very alike to the system we propose here, [16] also uses non-orthogonal basis vectors to traverse directional neighborhood while generating the discrete line. However, their coordinate system is spanned by the axes which are mutually at acute angles; whereas, we have two axes orthogonal to each other and the remaining axis at an obtuse angle with respect to the other two. This makes our system closer to the Cartesian coordinate system, and thus needs simpler transformations to switch between systems (see Section 2).

The line drawing approach we use in this paper, enables us to identify all voxels intersected by the real line and thus also produce the supercover of the line along with the thinner models, as required. The Bresenham line drawing approach as adopted in [16] chooses only the voxels closest to the real line and only maintaining minimal connectivity, thus producing a particular (naive) model of the digital line. [19] also explains similar techniques of line drawing as in [16]. They uses the same basis vectors for the FCC grid, and also give similar basis vectors for the BCC grid; and describes Bresenham like integer algorithm for line drawing suitable for ray-tracing purposes. In [20], the authors generalize the approach from [19] to arbitrary dimension and any regular grid. It utilizes the concept of generalized basis vectors according to the chosen neighborhood of voxels in a possibly non-orthogonal regular grid and uses the directional vectors in designing a generalized Bresenham line drawing algorithm.

**Definition 3** (Digital Sphere in CCG [3]). *The digital sphere  $S$  centered in  $c$ , of radius  $r$  and thickness  $\rho$ , is the set of integer coordinate points  $p$  verifying the following inequalities:*

$$\left(r - \frac{\rho}{2}\right) \leq d_2(c, p) < \left(r + \frac{\rho}{2}\right).$$



**Figure 1:** (a) Combinatorial coordinates for vertices in the face centered cubic grid ([10]). (b) Vertex coordinates in the proposed non-orthogonal coordinate system, where  $u = \frac{1}{4}$ .

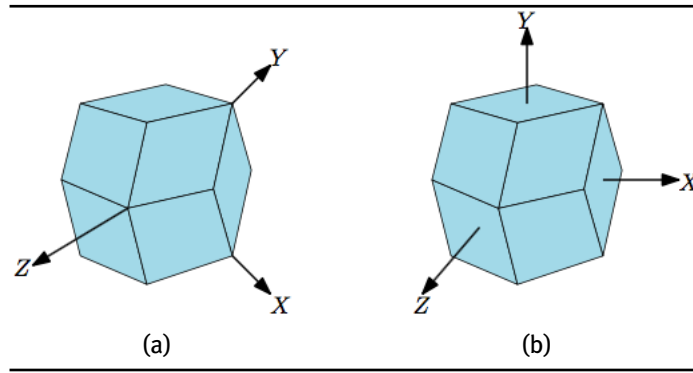
The thickness  $\rho$  of a digital sphere also specifies the separability property of the constructed object exactly similar to the definition of digital plane. To ensure topological minimality, we need to use more complex formulations as described in [5, 6].

In this paper, we consider a grid based on rhombic dodecahedrons, which are space-filling polyhedrons similar to cubes. A rhombic dodecahedron has 12 face-connected neighbors and 6 strictly vertex-connected neighbors. A consistent coordinate system for the RDG will help us to utilize this grid in 3D with similar advantages as hexagonal grid gives over square grid in 2D, specially in the domain of imaging, tomography, etc. As rhombic dodecahedron space filling arrangement emulates the close sphere packing in space, RDG can be used for additive manufacturing using spherical material particle powder.

The main problem of the unit dodecahedron is that neither its vertices nor its centroids form an integer coordinate grid. It is however simple to get integer coordinates for either centroids or vertices or both using simple transformation as presented in [10], shown in Figure 1(a). In the mentioned paper [10], a combinatorial 3-coordinate system for cells in the face centered cubic grid is presented and some of its properties are detailed; authors made a rescaling with factor 2, in order to assign to each cell the Cartesian coordinates of its center and to have only integer coordinates for faces and edges. However, the main drawback of this system, which we addressed in [7], is that this coordinate system leads to two different and intricate grids, and any integer point in this space may either represent a centroid or a vertex. This will be a major cause of anomaly while doing transformation operations such as rotation, shear etc. and construction of different digital objects in this grid. Other approaches have been proposed such as using four coordinates with a linear dependency between the integer coordinates [18] which of course, does not allow a complete system where any four coordinates correspond to a grid point.

### 3 Rhombic Dodecahedron Grid

In this section, we discuss a non-orthogonal coordinate system called *RDGCS* (for Rhombic Dodecahedron Grid Coordinate System) to represent the rhombic dodecahedron close packing grid such that each integer point represents the centroid of a rhombic dodecahedron cell. The system proposed here is similar to the one proposed in [7]. There are however some minor differences. In [7], there was an extra rotation involved because the central rhombic dodecahedron of coordinates  $(0, 0, 0)$  was not oriented in the same way. The coordinate system presented in this paper is basic and the reader shouldn't have a problem adapting the proposed formulas for another grid orientation. We talk about the conversion of this new coordinate system to and from the Cartesian coordinate system (as described in [10]) and define the corresponding distance metric. We also show that there exists a 48-symmetry in RDG using this new coordinate system which can be utilized further to propose efficient algorithms for geometric object construction in this type of grid.



**Figure 2:** Coordinate system transformation: (a) Axes in Cartesian coordinate system [10], (b) After required transformation to get the non-orthogonal coordinate system.

### 3.1 Non-Orthogonal Coordinate System

The coordinate system RDGCS we present here is a simpler version of the one proposed in [7]. We transform the well-known Cartesian space to obtain non-orthogonal grid vectors. Figure 2(a) shows the initial positioning of the rhombic dodecahedron centered at the origin with respect to the coordinate axes. All three axes are orthogonal to each other and pass through the vertices of the rhombic dodecahedron.

To obtain our new non-orthogonal RDGCS, we rotate the  $X$ - and  $Y$ -axes while bending the  $Z$ -axis away from the positive  $X$ - and  $Y$ -axes to make all three axes pass through the center of the faces of the rhombic dodecahedron (Figure 2(b)). During this transformation, a scaling is also required to make the rhombic dodecahedron cells unit in size; we consider our cells to have unit distance between opposite faces and do not show any explicit scaling step to obtain the same. Notice that, to simplify the visualization, we have kept the rhombic dodecahedron fixed while transforming the coordinate axes.

The following lemma states the transformation needed for a point in the RDGCS to and from the Cartesian coordinate system.

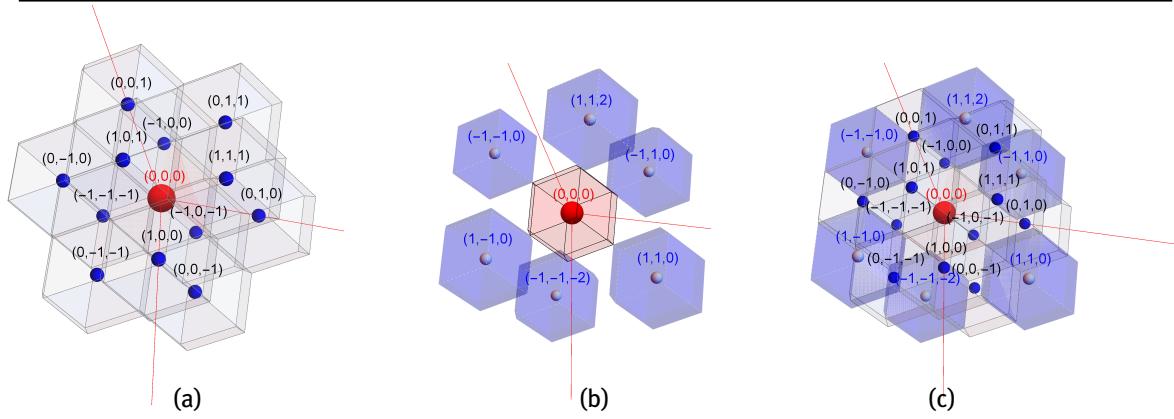
**Lemma 1.** *Coordinate conversion*

- The transformation of a point  $p(x, y, z)$  from the Cartesian coordinate system to the new RDGCS is given by:  $\mathcal{N}(p) = \left( \frac{x-y+z}{2}, \frac{x+y+z}{2}, z \right)$ .
- The transformation of a point  $p(x, y, z)$  from the new RDGCS to the Cartesian coordinate system is given by,  $\mathcal{C}(p) = (x + y - z, -x + y, z)$ .

*Proof.* To obtain the new coordinates for a point  $p(x, y, z)$  we use a transformation to rotate the  $X$ -,  $Y$ -axes and to make the  $Z$ -axis non-orthogonal with respect to the  $XY$ -plane. This transformation can be calculated as follows.

$$\begin{aligned} \mathcal{N}(p) &= \begin{bmatrix} 1/2 & -1/2 & 1/2 \\ 1/2 & 1/2 & 1/2 \\ 0 & 0 & 1 \end{bmatrix} (x, y, z) \\ &= \left( \frac{x-y+z}{2}, \frac{x+y+z}{2}, z \right). \end{aligned}$$

To get the Cartesian coordinate values for a point  $p(x, y, z)$  in the new coordinate space, we need to do the reverse transformation. Therefore, we have to make the  $Z$ -axis orthogonal to the  $XY$ -plane. Following



**Figure 3:** The twelve face-connected neighbors and the six strictly vertex-connected neighbors in the proposed non-orthogonal coordinate system.

calculation can be used to obtain this transformation.

$$\begin{aligned} \mathcal{C}(p) &= \begin{bmatrix} 1 & 1 & -1 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} (x, y, z) \\ &= (x + y - z, -x + y, z). \end{aligned}$$

□

**Lemma 2** (Euclidean Distance). *Let  $p(x, y, z)$  and  $p'(x', y', z')$  be two points in the RDGCS. The Euclidean distance between  $p$  and  $p'$  is given by,*

$$d_2(p, p') = \sqrt{2((x - x')^2 + (y - y')^2 + (z - z')^2 - (x - x')(z - z') - (y - y')(z - z'))}.$$

*Proof.* The proof follows directly from Lemma 1.

$$\begin{aligned} d_2(p, p') &= \|\mathcal{C}(p) - \mathcal{C}(p')\| \\ &= \|(x + y - z, -x + y, z) - (x' + y' - z', -x' + y', z')\| \\ &= \sqrt{(x - y - x' + y')^2 + (z - z')^2 + (x + y - z - x' - y' + z')^2} \\ &= \sqrt{2((x - x')^2 + (y - y')^2 + (z - z')^2 - (x - x')(z - z') - (y - y')(z - z'))}. \end{aligned} \tag{1}$$

□

We simplify the distance calculation by dropping the constant factor from the Euclidean distance as calculated in Lemma 2, and define our new distance which is to be used for further calculations.

**Definition 4** (RDG Distance). *Let  $p(x, y, z)$  and  $p'(x', y', z')$  be two points in the RDGCS. The RDG distance between  $p$  and  $p'$  is defined as,*

$$\begin{aligned} d(p, p') &= \frac{1}{\sqrt{2}} d_2(p, p') \\ &= \sqrt{(x - x')^2 + (y - y')^2 + (z - z')^2 - (x - x')(z - z') - (y - y')(z - z')}. \end{aligned}$$

A rhombic dodecahedron has 12 face-connected neighbors and 6 strictly vertex-connected neighbor as shown in Figure 3. We make the following observations regarding the neighborhood of each rhombic dodecahedron cell.

**Observation 1** (Neighbors). *Rhombic dodecahedron cell  $p$  and  $p'$  are face (respectively strictly vertex) connected neighbors iff the RDG distance between  $p$  and  $p'$  i.e.  $d(p, p')$  is equal to 1 (respectively  $\sqrt{2}$ ).*

*Proof.* From the positioning of the rhombic dodecahedron cell with respect to the non-orthogonal coordinate axes, as shown in Figure 2(c), following are the face-connected neighbors of a cell  $(x, y, z)$ :  $(x + 1, y, z)$ ,  $(x - 1, y, z)$ ,  $(x, y + 1, z)$ ,  $(x, y - 1, z)$ ,  $(x, y, z + 1)$ ,  $(x, y, z - 1)$ ,  $(x + 1, y + 1, z + 1)$ ,  $(x - 1, y - 1, z - 1)$ ,  $(x, y + 1, z + 1)$ ,  $(x + 1, y, z + 1)$ ,  $(x - 1, y, z - 1)$ ,  $(x, y - 1, z - 1)$ . By using the definition of  $d(p, p')$ , we can see that the RDG distance between  $p$  and  $p'$  is always 1 when they are face-connected neighbors.

Similarly, we can have the strictly vertex-connected neighbors of  $(x, y, z)$ , which are  $(x + 1, y + 1, z)$ ,  $(x + 1, y - 1, z)$ ,  $(x - 1, y + 1, z)$ ,  $(x - 1, y - 1, z)$ ,  $(x + 1, y + 1, z + 2)$ ,  $(x - 1, y - 1, z - 2)$ , and the RDG distance between two strictly vertex-connected neighbors comes out to be  $\sqrt{2}$ . □

### 3.2 48-Symmetry

The rhombic dodecahedron belongs to the octahedral symmetry group. It therefore has 24 rotational symmetries and 48 symmetries when considering transformations that combine reflection with rotation. Table 1 shows the 48 symmetric points corresponding to a point  $(x, y, z)$  in the orthogonal coordinate system.

**Table 1:** 48-symmetry in RDG using orthogonal coordinate system.

$(x, y, z)$	$(x, z, y)$	$(y, x, z)$	$(y, z, x)$
$(z, x, y)$	$(z, y, x)$	$(-x, y, z)$	$(-x, z, y)$
$(y, -x, z)$	$(y, z, -x)$	$(z, -x, y)$	$(z, y, -x)$
$(x, -y, z)$	$(x, z, -y)$	$(-y, x, z)$	$(-y, z, x)$
$(z, x, -y)$	$(z, -y, x)$	$(x, y, -z)$	$(x, -z, y)$
$(y, x, -z)$	$(y, -z, x)$	$(-z, x, y)$	$(-z, y, x)$
$(-x, -y, z)$	$(-x, z, -y)$	$(-y, -x, z)$	$(-y, z, -x)$
$(z, -x, -y)$	$(z, -y, -x)$	$(x, -y, -z)$	$(x, -z, -y)$
$(-y, x, -z)$	$(-y, -z, x)$	$(-z, x, -y)$	$(-z, -y, x)$
$(-x, y, -z)$	$(-x, -z, y)$	$(y, -x, -z)$	$(y, -z, -x)$
$(-z, -x, y)$	$(-z, y, -x)$	$(-x, -y, -z)$	$(-x, -z, -y)$
$(-y, -x, -z)$	$(-y, -z, -x)$	$(-z, -x, -y)$	$(-z, -y, -x)$

In our case, we are interested in the 48-symmetry in the RDGCS. In order to generate the 48-symmetry in RDGCS, one should convert them in such a system. Let us take the point  $P(x, y, z)$  in the RDGCS, in order to obtain the 48-Symmetry in such a system, let us first transfer the point  $P$  in the orthogonal cubic coordinate system, this will gives us the point  $P'(x + y - z, y - x, z)$  (see Lemma 1). Let us now apply the 48-symmetry shown in Table 1 on the point  $P'$ . Let us take a concrete example: if we apply the symmetry on the first line second value of the Table 1 that gives us the symmetry  $(x, z, y)$  of the point  $(x, y, z)$  we obtain by applying this symmetry on our point  $P'$  the point  $(x + y - z, z, y - x)$ . Now, we need to re-transform the point to RDGCS (See Lemma 1) which will gives us  $(\frac{x+y-z-z+y-x}{2}, \frac{x+y-z+z+y-x}{2}, y - x) = (y - z, y, y - x)$ . In the same manner, we do the computation for all the lines of the Table 1 and we obtain the 48-symmetry in the RDGCS of the point  $(x, y, z)$  as shown in Table 2. This symmetry can be utilized while constructing symmetrical geometric objects, e.g. spheres, cones, cylinders, etc., in this digital space.

**Table 2:** 48-symmetry in RDG using proposed RDG Coordinate System.

$(x, y, z)$	$(y - z, y, -x + y)$	$(-x + z, y, z)$	$(y - z, y, x + y - z)$
$(-x + z, y, -x + y)$	$(x, y, x + y - z)$	$(-y + z, -x + z, z)$	$(-x, -x + z, -x + y)$
$(y, -x + z, z)$	$(-x, -x + z, -x - y + z)$	$(y, -x + z, -x + y)$	$(-y + z, -x + z, -x - y + z)$
$(y, x, z)$	$(x - z, x, x - y)$	$(-y + z, x, z)$	$(x - z, x, x + y - z)$
$(-y + z, x, x - y)$	$(y, x, x + y - z)$	$(x - z, y - z, -z)$	$(y, y - z, -x + y)$
$(-x, y - z, -z)$	$(y, y - z, x + y - z)$	$(-x, y - z, -x + y)$	$(x - z, y - z, x + y - z)$
$(-x + z, -y + z, z)$	$(-y, -y + z, x - y)$	$(x, -y + z, z)$	$(-y, -y + z, -x - y + z)$
$(x, -y + z, x - y)$	$(-x + z, -y + z, -x - y + z)$	$(y - z, x - z, -z)$	$(x, x - z, x - y)$
$(-y, x - z, -z)$	$(x, x - z, x + y - z)$	$(-y, x - z, x - y)$	$(y - z, x - z, x + y - z)$
$(-y, -x, -z)$	$(-x + z, -x, -x + y)$	$(y - z, -x, -z)$	$(-x + z, -x, -x - y + z)$
$(y - z, -x, -x + y)$	$(-y, -x, -x - y + z)$	$(-x, -y, -z)$	$(-y + z, -y, x - y)$
$(x - z, -y, -z)$	$(-y + z, -y, -x - y + z)$	$(x - z, -y, x - y)$	$(-x, -y, -x - y + z)$

## 4 Digital Objects: Definition and Topological Analysis

In this section, we define digital models of three primitive geometric objects, namely, digital plane, 3D digital line, and digital sphere. We define these using the proposed RDGCS and use the defined RDG distance to control the thickness of the objects resulting in different topological models.

### 4.1 Digital Plane in RDG

We define digital plane in the RDGCS using the classical definition. However, note that, the real plane in consideration here is given in the RDGCS and not in the Cartesian coordinate system.

**Definition 5** (Digital Plane). *Let  $ax + by + cz + d = 0$  be a plane equation in the RDGCS. Let  $\rho$  be the thickness of the digital plane. The digitization of the plane in the dodecahedron grid is the set of dodecahedron  $(x, y, z)$  satisfying the following inequality.*

$$-\rho/2 \leq ax + by + cz + d < \rho/2. \tag{2}$$

Following we give the formulas for converting between plane equations from the RDGCS to the Cartesian coordinate system and reverse.

**Proposition 1** (RDGCS to Cartesian Plane Equation). *Let  $ax_n + by_n + cz_n + d = 0$  be a plane equation in the RDGCS. The equation of the same plane in the Cartesian coordinate system is given by:*

$$2d + 2cz_c + a(x_c + z_c - y_c) + b(x_c + y_c + z_c) = 0 \tag{3}$$

**Proposition 2** (Cartesian to RDGCS Plane Equation). *Let  $ax_c + by_c + cz_c + d = 0$  be a plane equation in the Cartesian coordinate system. The equation of the same plane in the RDGCS is given by:*

$$d - bx_n + by_n + a(x_n + y_n - z_n) + cz_n = 0 \tag{4}$$



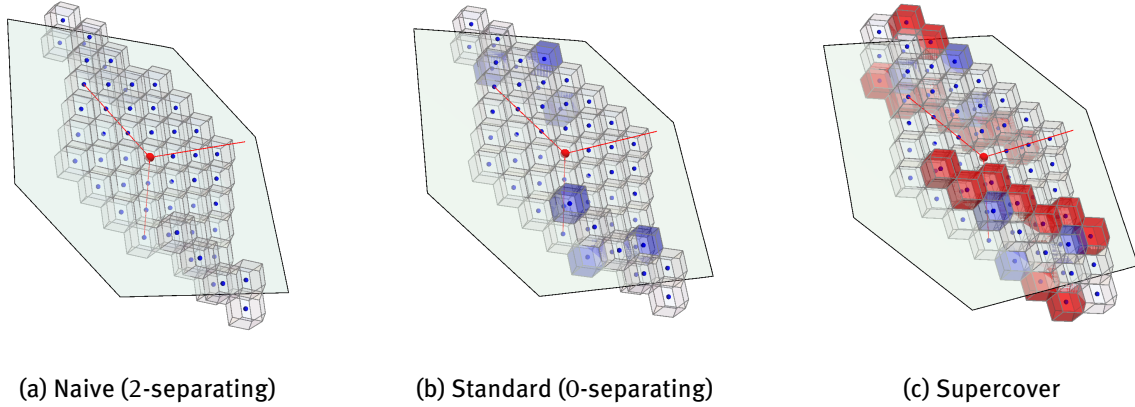


Figure 4: Dodecahedron digitizations for the plane  $3x - 18y - z = 0$  in RDGCS.

We can control the topology of the digital plane by changing the value of  $\rho$  as mentioned in Proposition 3 below.

**Proposition 3** (Topology and Separation). *Consider the Euclidean plane of equation  $ax + by + cz + d = 0$  in RDGCS. The supercover plane is obtained using:*

$$\rho_{Sup} = \max \left( |a + b|, |b - a|, |a + b + 2c|, \frac{1}{2}(|a + b| + |b - a| + |a + b + 2c|) \right). \quad (5)$$

The 0-separating minimal plane is obtained using:

$$\rho_{Std} = \max (|a|, |b|, |c|, |a + b + c|, |a + c|, |b + c|, |a + b|, |a - b|, |a + b + 2c|). \quad (6)$$

The 2-separating minimal digital plane in the dodecahedron grid is obtained for

$$\rho_N = \max (|a|, |b|, |c|, |a + b + c|, |a + c|, |b + c|). \quad (7)$$

*Proof.* The proof for the supercover is the same as in [25] using the following plane equation in the Cartesian coordinate system:  $Ax + By + Cz + D = 0$  where  $A = \frac{a+b}{2}$ ,  $B = \frac{b-a}{2}$ ,  $C = \frac{a+b+2c}{2}$ ,  $D = d$ . The coefficient values directly follow from Proposition 1.

It can also be proven by considering the thickness of the thickest directions we can have in the dodecahedron, i.e. the distance between opposite vertices or opposite faces as necessary. To obtain this, we consider the coordinates of the vertices and face centers of the rhombic dodecahedron. To determine the 0-separating minimal thickness, only vertex coordinates are considered; and for the 2-separating minimal thickness, only face center coordinates are considered. For supercover thickness, both have to be taken into account.

Next, we calculate their (vertices and/or face centers) individual distances to the Euclidean plane, and the final thickness is calculated considering the maximum of these distances. This ensures that the chosen thickness gives us an appropriate neighborhood around the real plane fulfilling the desired separability criteria, and in the case of the supercover this neighborhood contains centroids of all the rhombic dodecahedrons intersected by the real plane. A detailed calculation in the Cartesian coordinate system for the supercover model can be found in [25]. We use the same method for the other two models and use the values in RDGCS.  $\square$

Note that, the 0-separating minimal plane, which is also known as the standard digitization, is the close to the supercover plane, which is also 0-separating, but not necessarily minimal. The 2-separating minimal plane is widely referred to as the naive digitization. Figure 4 presents the three different digitizations for the plane  $3x - 18y - z = 0$  in the dodecahedron non-orthogonal grid. Figure 4(a) shows the naive plane for the thickness defined by Equation 7. In Figure 4(b), we can see the standard plane obtained with Equation 6.

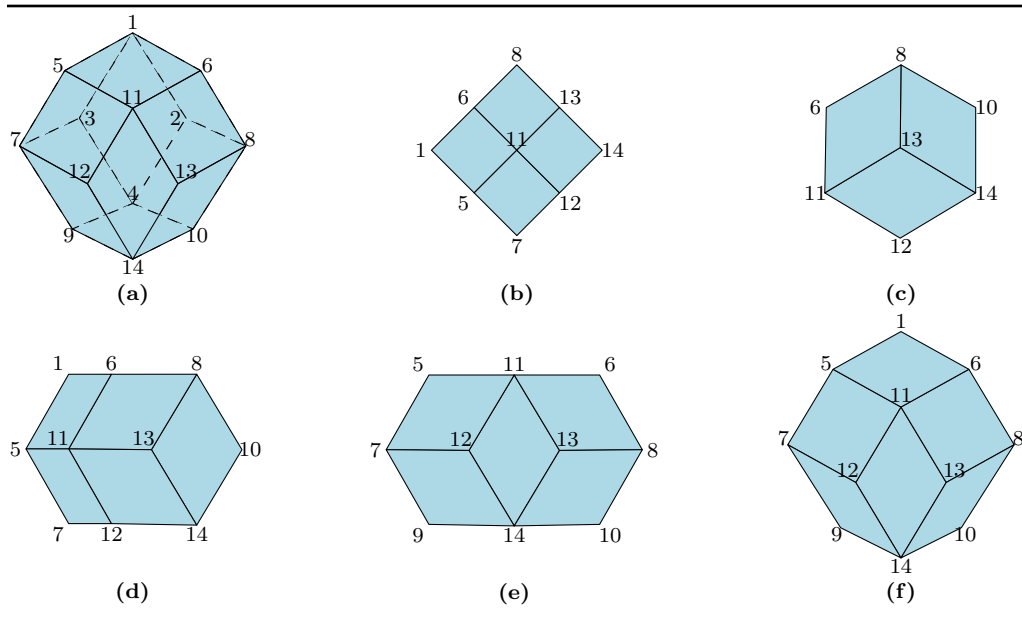
The additional dodecahedra compared to the naive digitization are highlighted in blue. In Figure 4(c), we present the thickest plane, i.e. the supercover plane which uses Equation 5 for the thickness. The additional dodecahedra compared to the two other digitizations are highlighted in red.

### 4.2 Digital Lines in 3D

A straight line in  $\mathbb{R}^3$  can be represented by the intersection of any two distinct planes passing through the line. This is because in  $\mathbb{R}^3$ , two non-parallel planes always intersect in a single line. However, in  $\mathbb{Z}^3$ , this is not true. Two non-parallel planes centered on the same continuous straight line result in an infinite number of different objects which can be a very larger set of voxels if their orientations are not chosen wisely. And even if we can fix the relative orientation of the digital planes (say, two orthogonal planes), we may still get more voxels than expected. The correct approach is to choose the set of planes according to the direction (vector) of the real line we want to digitize. The shape of the voxel in the chosen grid system also plays its role in choosing the necessary digital planes to be intersected.

For instance, in the cubic grid, as explained in [1], we may need as many as three digital planes. In [25], the proposed solutions with three digital planes in RDG did, however, not cover all directions. In this paper, we show that according to the orientation of the rhombic dodecahedron voxel with respect to the direction of the line to be digitized, we may need as many as four different digital planes to define a line in the RDG. The proof of the same follows from the observation depicted in [1].

According to the direction of the real line, if we change the orientation of the rhombic dodecahedron cell, we get five different view points. We show these different views in Figure 5. Note that, following [1], we need intersection of as many digital planes as the number of parallel pairs of edges on the boundary of each such view. Collinear edges on the boundary are considered as one; they appears in Figure 5(b), (d), and (e). Therefore, in only one of these five views (Figure 5(b)), it suffices to get the correct intersection for the digital line using just two digital planes. Three of the views (Figure 5(c, d, e)) need at least three different planes to get the correct intersection, and one of the five views (Figure 5(f)) requires four planes.



**Figure 5:** The five views of the dodecahedron corresponding to the different direction of the line: (a) vertex numbering, (b)  $a > 0$  and  $b = c = 0$ , (c)  $a = b = c$ , (d)  $0 < b < a$  and  $b = c$ , (e)  $0 \leq |c| < a$  and  $a = b$ , (f)  $0 \leq |c| < b < a$ .

To get the corresponding plane equations for each of the different views, we identify the parallel pairs of edges on the boundary of the views. The plane equations are then obtained by choosing planes passing through the real line (which is to be digitized) and having its normal parallel to each such edge pairs on the boundary. The calculations are straight forward in the Cartesian coordinate system when we know the equation of the real line and the edge vectors of the rhombic dodecahedron view. Therefore, we do that first and obtain the set of plane equations for each of the five views.

The plane equations in the Cartesian coordinate system, with  $(a, b, c)$  in the Cartesian coordinate system as well, are given by:

Case 1:  $a > b = c = 0$

$$2\sqrt{2}(b-c)y_c + (2a - \sqrt{2}(b-c))z_c - \sqrt{2}a(-2x_c + z_c) + 4a = 0$$

$$2\sqrt{2}(b+c)y_c - (2a - \sqrt{2}(b+c))z_c - \sqrt{2}a(-2x_c + z_c) + 4a = 0$$

Case 2:  $a = b = c > 0$

$$\sqrt{2}(b-c)y_c + (a+b)z_c - ((b-c)z_c)/\sqrt{2} + ((-a-c)(-2x_c + z_c))/\sqrt{2} + 2(a+b) = 0$$

$$\sqrt{2}(b+c)y_c + (-a-b)z_c - ((b+c)z_c)/\sqrt{2} + ((-a+c)(-2x_c + z_c))/\sqrt{2} + 2(a+b) = 0$$

$$\sqrt{2}(b+c)y_c + (-a+b)z_c - ((b+c)z_c)/\sqrt{2} + ((-a-c)(-2x_c + z_c))/\sqrt{2} + 2(a+c) = 0$$

Case 3:  $a > b = c > 0$

$$2\sqrt{2}(b-c)y_c + 2az_c - \sqrt{2}(b-c)z_c - \sqrt{2}a(-2x_c + z_c) + 4a = 0$$

$$\sqrt{2}(b+c)y_c + (-a-b)z_c - ((b+c)z_c)/\sqrt{2} + ((-a+c)(-2x_c + z_c))/\sqrt{2} + 2(a+b) = 0$$

$$\sqrt{2}(b+c)y_c + (-a+b)z_c - ((b+c)z_c)/\sqrt{2} + ((-a-c)(-2x_c + z_c))/\sqrt{2} + 2(a+c) = 0$$

Case 4:  $a = b > c \geq 0$

$$\sqrt{2}(-b+c)y_c + (-a+b)z_c - ((-b+c)z_c)/\sqrt{2} + ((a-c)(-2x_c + z_c))/\sqrt{2} + 2(b-c) = 0$$

$$\sqrt{2}(b+c)y_c + (-a-b)z_c - ((b+c)z_c)/\sqrt{2} + ((-a+c)(-2x_c + z_c))/\sqrt{2} + 2(a+b) = 0$$

$$\sqrt{2}(-b+c)y_c + (-a-b)z_c - ((-b+c)z_c)/\sqrt{2} + ((a+c)(-2x_c + z_c))/\sqrt{2} + 2(a+b) = 0$$

Case 5:  $a > b > c \geq 0$

$$\sqrt{2}(b+c)y_c + (-a+b)z_c - ((b+c)z_c)/\sqrt{2} + ((-a-c)(-2x_c + z_c))/\sqrt{2} + 2(a+c) = 0$$

$$\sqrt{2}(-b+c)y_c + (-a+b)z_c - ((-b+c)z_c)/\sqrt{2} + ((a-c)(-2x_c + z_c))/\sqrt{2} + 2(a-c) = 0$$

$$\sqrt{2}(-b+c)y_c + (-a-b)z_c - ((-b+c)z_c)/\sqrt{2} + ((a+c)(-2x_c + z_c))/\sqrt{2} + 2(a+b) = 0$$

$$\sqrt{2}(b+c)y_c + (-a-b)z_c - ((b+c)z_c)/\sqrt{2} + ((-a+c)(-2x_c + z_c))/\sqrt{2} + 2(a+b) = 0$$

The planes equation in RDGCS, with  $(a, b, c)$  in RDGCS as well, are given by:

Case 1:  $a > b = c = 0$

$$2(a+b-c)x_n - 2(a+b-c)y_n - 2(a-b+c)(x_n + y_n - z_n) + 2(a+b-c)z_n = 0$$

$$2(a+b-c)x_n - 2(a+b-c)y_n + 2(-a+b+c)(x_n + y_n - z_n) - 2(a+b-c)z_n = 0$$

Case 2:  $a = b = c > 0$

$$-(-a-b)x_n + (-a-b)y_n + (-a+b-c)(x_n + y_n - z_n) + (2b-c)z_n = 0$$

$$-(-a-b+2c)x_n + (-a-b+2c)y_n + (-a+b+c)(x_n + y_n - z_n) + (-2b+c)z_n = 0$$

$$-(-a-b)x_n + (-a-b)y_n + (-a+b+c)(x_n + y_n - z_n) + (-2a+c)z_n = 0$$

Case 3:  $a > b = c > 0$

$$\begin{aligned} & 2(a + b - c)x_n - 2(a + b - c)y_n - 2(a - b + c)(x_n + y_n - z_n) + 2(a + b - c)z_n = 0 \\ & -(a - b + 2c)x_n + (-a - b + 2c)y_n + (-a + b + c)(x_n + y_n - z_n) + (-2b + c)z_n = 0 \\ & -(-a - b)x_n + (-a - b)y_n + (-a + b + c)(x_n + y_n - z_n) + (-2a + c)z_n = 0 \end{aligned}$$

Case 4:  $a = b > c \geq 0$

$$\begin{aligned} & -(a + b - 2c)x_n + (a + b - 2c)y_n + (a - b + c)(x_n + y_n - z_n) + (-2a + c)z_n = 0 \\ & -(a - b + 2c)x_n + (-a - b + 2c)y_n + (-a + b + c)(x_n + y_n - z_n) + (-2b + c)z_n = 0 \\ & -(a + b)x_n + (a + b)y_n + (a - b + c)(x_n + y_n - z_n) + (-2b + c)z_n = 0 \end{aligned}$$

Case 5:  $a > b > c \geq 0$

$$\begin{aligned} & -(-a - b)x_n + (-a - b)y_n + (-a + b + c)(x_n + y_n - z_n) + (-2a + c)z_n = 0 \\ & -(a + b - 2c)x_n + (a + b - 2c)y_n + (a - b + c)(x_n + y_n - z_n) + (-2a + c)z_n = 0 \\ & -(a + b)x_n + (a + b)y_n + (a - b + c)(x_n + y_n - z_n) + (-2b + c)z_n = 0 \\ & -(-a - b + 2c)x_n + (-a - b + 2c)y_n + (-a + b + c)(x_n + y_n - z_n) + (-2b + c)z_n = 0 \end{aligned}$$

To transform the plane equation from the Cartesian one to the RDGCS, we use Lemma 1 and the conversion formulas from Section 4.1. While generating the digital line, we take the intersection of the naive digitization of the planes. This ensures an appropriate tubular neighborhood around the real line which includes centroids of all the rhombic dodecahedrons intersected by the real line and nothing extra. The approach is the same as in [1].

We have implemented two algorithms in Mathematica to compute the 3D lines in the dodecahedron grid. The first one (Algorithm 1) computes the tubular neighborhood corresponding to the line using the different views we presented and, for every integer coordinate point in a given window, tests whether the point is inside this neighborhood or not. The result in this case is the supercover digitization of the line and may not be the thinnest digitization. As all points inside the chosen 3D window is checked with a constant number of planes, this brute-force algorithm runs in cubic time.

---

**Algorithm 1:** Brute-force Algorithm 3D Line

---

```

input : Line direction  $v$  and 3D window  $W$ 
output: Dodecahedron centroid list forming the digital 3D line  $L$ 
1 begin
2    $L = \{\}$  ▷ initialization
3    $P = \text{getPlanes3Dline}(v)$  ▷ set of plane equations as per different cases in Sec. 4.2
4    $T = \text{getNaiveThicknesses}(P)$  ▷ array of thicknesses for the planes using Equation 7
5   foreach  $point$  in  $W$  do ▷ for each  $point$  inside the chosen 3D window
6      $test = \text{True}$  ▷ initialize  $test$  variable
7     foreach  $p$  in  $P$  do
8       if  $point \notin \text{Plane}(p, T[p])$  then ▷ if  $point$  not in digital plane using Equation 2
9          $test = \text{False}; \text{break}$  ▷ break loop if one plane doesn't contain  $point$ 
10      if  $test$  then
11         $L = L \cup \{point\}$  ▷ add  $point$  to digital line
12    return  $L$ 
13 end

```

---

**Algorithm 2:** Incremental Algorithm 3D Line

---

```

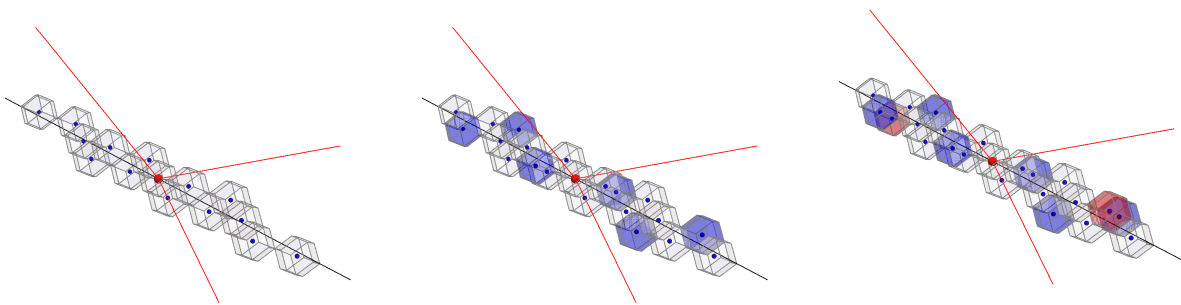
input : Line direction  $v$ , starting point  $sp$ , and 3D window  $W$ 
output: Dodecahedron centroid list forming the digital 3D line  $L$ 
1 begin
2    $L = \{\}$ ;  $point = sp$  ▷ initialization
3    $P = \text{getPlanes3Dline}(v)$  ▷ set of plane equations as per different cases in Sec. 4.2
4    $T = \text{getNaiveThicknesses}(P)$  ▷ array of thicknesses for the planes using Equation 7
5   while  $point$  in  $W$  do ▷ while  $point$  is inside the chosen 3D window
6      $L = L \cup \{point\}$  ▷ add  $point$  to digital line
7      $N = \text{getNeighbors}(point, v)$  ▷ subset of neighbors at a particular direction
8     foreach  $n$  in  $N$  do
9        $test = \text{True}$  ▷ initialize  $test$  variable
10      foreach  $p$  in  $P$  do
11        if  $n \notin \text{Plane}(p, T[p])$  then ▷ if  $n$  not in digital plane using Equation 2
12           $test = \text{False}$ ; break ▷ break loop if one plane doesn't contain  $n$ 
13        if  $test$  then
14           $point = n$ ; break ▷ break loop if one neighbor is found as next point
15      return  $L$ 
16 end

```

---

The second algorithm (Algorithm 2) starts from a known integer point on the digital line and incrementally builds the line while testing a subset (according to the direction vector of the line) of the face or vertex connected neighbors. Only one neighbor is chosen at each step leading to a minimal face-connected (standard) or vertex-connected (naive) 3D digital line. This incremental algorithm runs in linear time, because at each step a constant number of neighbors are checked with a constant number of planes.

Figure 6 presents an example of a 3D line with direction vector  $(a = 14, b = 7, c = 3)$  in the rhombic dodecahedron grid using different thicknesses and the RDGCS. The left one is the naive line constructed using Algorithm 2 by choosing vertex connected neighbors at Line 7. The middle one is the standard line constructed the same way but choosing only face connected neighbors; this is slightly thicker than the naive line. The additional dodecahedra have been highlighted in blue. The last one is the supercover of the line, i.e. all the dodecahedra that are intersected by the real line. The additional dodecahedra in this step are colored in red. The supercover is generated using Algorithm 1, but can also be generated using a slightly modified version of Algorithm 2 which considers all neighbors and maintains a queue of next points at each step.



**Figure 6:** Naive, standard, and supercover digitization of the 3D line with direction vector  $(14, 7, 3)$ .

### 4.3 Digital Sphere

We define a digital sphere in the proposed RDGCS using the same thickness notion as used while defining Andres Analytical Sphere (recollected in Definition 3). However, we use our RDG distance measure as defined in Definition 4.

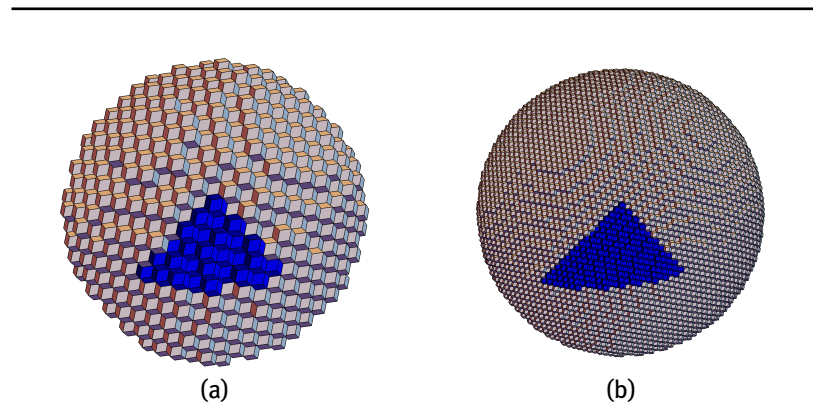
**Definition 6** (Digital Sphere in RDG). *The set of Dodecahedron centroids belonging to the digitization of a Sphere centered on  $c$  of radius  $r$  and thickness  $\rho$  is given by:  $S_{(c,r)} = \{p \in \mathbb{Z}^3 \mid r - \frac{\rho}{2} \leq d(c, p) < r + \frac{\rho}{2}\}$*

To control the topology of the resulting sphere, we use one of the two following values for the thickness.

**Proposition 4** (Topology and Separation). *The thickness  $\rho = 1$  provides a 2-separating sphere. The thickness  $\rho = \sqrt{2}$  provides a 0-separating sphere.*

*Proof.* The proof for the 2-separating value comes directly from the distance between a dodecahedron and its face neighbors which is 1 (leading to a thickness 1). The distance  $\sqrt{2}$  between the dodecahedron and its vertex neighbors leads to the 0-separating thickness of  $\sqrt{2}$ .  $\square$

The spheres presented in Figure 7 are obtained using a brute-force algorithm implemented in Mathematica (similar to Algorithm 1) that tests every integer coordinate point (i.e. dodecahedron centroid) in the bounding box with the sphere inequality using the 2-separating thickness 1. An optimization of this algorithm could be implemented using a 48<sup>th</sup> of the bounding box (see blue regions in Figure 7) and build the rest of the sphere using the symmetries we have previously defined in Section 3.2.



**Figure 7:** Digital sphere of (a) radius 8 and (b) radius 23 in the rhombic dodecahedron grid, with one of the 48<sup>th</sup> parts shown in blue, computed using the proposed RDG Coordinate System.

## 5 Conclusions

In this paper, we have described a new coordinate system for the rhombic dodecahedron grid. The novelty of this system is that unlike the previous coordinate system [10], every single one of the integer coordinate points is the centroid of a rhombic dodecahedron. Using this system, we have proposed the characterization of 3D digital plane with its topological features and a characterization of 3D digital lines. We have also presented a characterization of 3D digital sphere, together with its 48 symmetries. In future, we want to

explore the potential of this coordinate system through the study of graphical transforms such as the rotation.

**Acknowledgement:** This work has been partially supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant no. 788183, and the DFG Collaborative Research Center TRR 109, 'Discretization in Geometry and Dynamics', Austrian Science Fund (FWF), grant no. I 02979-N35.

## References

- [1] Eric Andres. *Modélisation Analytique Discrète d'Objets Géométrique*. Habilitation à diriger des recherches, UFR Sciences Fondamentale et Appliquées, Université de Poitiers (France), Décembre 2000.
- [2] Eric Andres, Raj Acharya, and Claudio Sibata. Discrete Analytical Hyperplanes. *Graphical Models and Image Processing*, 59(5):302–309, 1997.
- [3] Eric Andres and Marie-Andrée Jacob. The Discrete Analytical Hyperspheres. *IEEE Transactions on Visualization and Computer Graphics*, 3:75–86, 1997.
- [4] Eric Andres, Philippe Nehlig, and Jean Françon. Supercovers of straight lines, planes and triangles. In *Discrete Geometry for Computer Imagery*, pages 243–254, Berlin, Heidelberg, 1997.
- [5] Ranita Biswas and Partha Bhowmick. From prima quadraginta octant to lattice sphere through primitive integer operations. *Theoretical Computer Science*, 624:56–72, 2016.
- [6] Ranita Biswas and Partha Bhowmick. On the Functionality and Usefulness of Quadraginta Octants of Naive Sphere. *Journal of Mathematical Imaging and Vision*, 59(1):69–83, 2017.
- [7] Ranita Biswas, Gaëlle Largeteau-Skapin, Rita Zrour, and Eric Andres. Rhombic Dodecahedron Grid – Coordinate System and 3D Digital Object Definitions. In *DGCI'2019*, volume 11414, pages 27–37, Paris, France, 2019.
- [8] Valentin E. Brimkov and Reneta P. Barneva. Analytical honeycomb geometry for raster and volume graphics. *The Computer Journal*, 48(2):180–199, 2005.
- [9] Lidija Comic and Paola Magillo. Repairing 3D Binary Images Using the FCC Grid. *Journal of Mathematical Imaging and Vision*, 61:1301–1321, 2019.
- [10] Lidija Comic and Benedek Nagy. A combinatorial 3-coordinate system for the face centered cubic grid. *2015 9th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 298–303, 2015.
- [11] Lidija Comic and Benedek Nagy. A combinatorial coordinate system for the body-centered cubic grid. *Graphical Models*, 87:11–22, 2016.
- [12] Lidija Comic and Benedek Nagy. A topological 4-coordinate system for the face centered cubic grid. *Pattern Recognition Letters*, 83:67–74, 2016.
- [13] Isabelle Debled-Rennesson. *Reconnaissance des droites et plans discrets*. PhD thesis, Université Louis Pasteur, Strasbourg, France, 1995.
- [14] Bernhard Finkbeiner, Usman R. Alim, Dimitri Van De Ville, and Torsten Moller. High-Quality Volumetric Reconstruction on Optimal Lattices for Computed Tomography. *Computer Graphics Forum*, 28(3):1023–1030, 2009.
- [15] Artyom M. Grigoryan and Sos S. Agaian. 2D hexagonal quaternion Fourier transform in color image processing. In *Mobile Multimedia/Image Processing, Security, and Applications 2016*, volume 9869, page 98690N. International Society for Optics and Photonics, 2016.
- [16] Li-Jun He, Yong-Kui Liu, and Shi-Chang Sun. A Line Generation Algorithm on 3D Face-Centered Cubic Grid. *Chinese Journal of Computers*, 33(12):2407–2416, 2010.
- [17] Xiangjian He and Wenjing Jia. Hexagonal Structure for Intelligent Vision. In *Information and Communication Technologies, 2005. ICT 2005. First International Conference on*, pages 52–64. IEEE, 2005.
- [18] Innchyn Her. Description of the F.C.C. Lattice Geometry Through a Four-Dimensional Hypercube. *Acta Cryst.*, A51:659–662, 1995.
- [19] Luis Ibáñez, Chafiaâ Hamitouche, and Christian Roux. Ray-Tracing and 3-D Objects Representation in the BCC and FCC Grids. In *Discrete Geometry for Computer Imagery*, pages 235–241, Berlin, Heidelberg, 1997.
- [20] Luis Ibáñez, Chafiaâ Hamitouche, and Christian Roux. A Vectorial Algorithm for Tracing Discrete Straight Lines in N-Dimensional Generalized Grids. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):97–108, 2001.
- [21] Charles Kittel. *Introduction to Solid State Physics*. Wiley, 8 edition, 2004.
- [22] Girish Koshti, Ranita Biswas, Gaëlle Largeteau-Skapin, Rita Zrour, Eric Andres, and Partha Bhowmick. Sphere Construction on the FCC Grid Interpreted as Layered Hexagonal Grids in 3D. In *Combinatorial Image Analysis*, pages 82–96. Springer International Publishing, 2018.
- [23] Vladimir A. Kovalevsky. *Geometry of Locally Finite Spaces: Computer Agreeable Topology and Algorithms for Computer Imagery*. Berlin: Editing House Dr. Barbel Kovalevski, 2008.

- [24] Marie-Ange Lebre, Antoine Vacavant, Manuel Grand-Brochier, Hugo Rositi, Robin Strand, Hubert Rosier, Armand Abergel, Pascal Chabrot, and Benoît Magnin. A robust multi-variability model based liver segmentation algorithm for CT-scan and MRI modalities. *Computerized medical imaging and graphics : the official journal of the Computerized Medical Imaging Society*, 76:101635, September 2019.
- [25] Troung Kieu Linh, Atsushi Imiya, Robin Strand, and Gunilla Borgefors. Supercover of Non-square and Non-cubic Grids. In *Combinatorial Image Analysis*, pages 88–97, Berlin, Heidelberg, 2004. IEEE, Springer.
- [26] S. Matej and R. M. Lewitt. Efficient 3D grids for image reconstruction using spherically-symmetric volume elements. *IEEE Transactions on Nuclear Science*, 42(4):1361–1370, 1995.
- [27] Kazi Mostafa and Innchyn Her. An Edge Detection Method for Hexagonal Images. *International Journal of Image Processing (IJIP)*, 10(4):161–173, 2016.
- [28] Benedek Nagy. Cellular Topology on the Triangular Grid. In *Combinatorial Image Analysis. IWCIA 2012*, volume 7655, pages 143–153. Lecture Notes in Computer Science, 2012.
- [29] Benedek Nagy. Cellular topology and topological coordinate systems on the hexagonal and on the triangular grids. *Annals of Mathematics and Artificial Intelligence*, 75:117–134, October 2015.
- [30] Benedek Nagy and Robin Strand. A connection between  $\mathbb{Z}^n$  and generalized triangular grids. *Advances in Visual Computing*, pages 1157–1166, 2008.
- [31] Benedek Nagy and Robin Strand. Non-traditional grids embedded in  $\mathbb{Z}^n$ . *International Journal of Shape Modeling*, 14:209–228, 2008.
- [32] Benedek Nagy and Robin Strand. Neighborhood sequences in the diamond grid: Algorithms with two and three neighbors. *International Journal of Imaging Systems and Technology*, 19(2):146–157, 2009.
- [33] Gergely Ferenc Racz and Balazs Csébfalvi. Cosine-weighted b-spline interpolation on the face-centered cubic lattice. *Computer Graphics Forum*, 37(3):503–511, 2018.
- [34] Jean-Pierre Reveillès. *Calcul en Nombres Entiers et Algorithmique*. PhD thesis, Université Louis Pasteur, Strasbourg, France, 1991.
- [35] Ivan Stojmenović. Honeycomb network. In *Proc. Math. Foundations of Computer Science - MFCS '95, Lecture Notes in Computer Science*, volume 969, pages 267–276, 1995.
- [36] Robin Strand and Gunilla Borgefors. Resolution Pyramids on the FCC and BCC Grids. In *Discrete Geometry for Computer Imagery*, pages 68–78, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
- [37] Robin Strand, Benedek Nagy, and Gunilla Borgefors. Digital distance functions on three-dimensional grids. *Theoretical Computer Science*, 412(15):1350–1363, 2011.