Polyhedra Generation from Lattice Points

Yukiko Kenmochi[1], Atsushi Imiya[1] and Norberto F. Ezquerra[2]

[1] Dept. of Information & Computer Sciences, Chiba University
1-33 Yayoi-cho Inage-ku, Chiba 263 JAPAN
ken@icsd4.tj.chiba-u.ac.jp, imiya@ics.tj.chiba-u.ac.jp
[2] Graphics, Visualization & Usability Center, College of Computing
Georgia Institute of Technology
801 Atlantic Drive Atlanta, GA, 30332-0280 US
norberto@cc.gatech.edu

Abstract. This paper focuses on a method for generating polyhedra from a set of lattice points, such as three-dimensional (3D) medical computerized tomography images. The method is based on combinatorial topology [1] and algebraic properties of the 3D lattice space [2]. It is shown that the method can uniquely generate polyhedra from a subset of the lattice space independently of the choice of neighborhood. Furthermore, a practical algorithm is developed and experimental results using 3D medical imagery are presented.
Key words. Polyhedra, lattice space, topology, boundary detection, medical images.

1 Introduction

Representation of 3D objects is necessary when we deal with 3D objects for computer applications. Polyhedra are one of the well-known representations used in the various areas related to computer vision [3, 4, 5, 6]. This paper focuses on the generation of polyhedra from a set of lattice points, such as 3D medical computerized tomography images whose data are digitized.

Many polyhedral representations in a lattice space have been introduced; these representations can be separated into two main groups. The first group consists of interpolating representations. One of the well-known representations of this kind is the Marching Cubes Method [7]. This method generates polyhedral surfaces by interpolating densities between lattice points whose densities are given. Therefore, the polyhedral surfaces that are created pass not only through lattice points but also through points between lattice points.

The second group consists of non-interpolating representations. One of these representations uses local neighborhood structures, called the 6-, 18- and 26-neighborhoods [8]. The neighborhood structures are useful for detecting a polyhedral surface of an object from a given lattice point set [8]. All the vertices of the created polyhedral surfaces are lattice points, as compared with the aforementioned interpolating polyhedra. However, this representation sometimes disconnects objects; this problem is called the connectivity paradox [9].

There is another representation in the second group that uses global topological characteristics of objects as well as local neighborhood structures [9, 10, 11]. This representation deals not with lattice points but with unit cubes whose centers are lattice points, for the preservation of the topology of objects. Thus, an object is regarded as a set of unit cubes and its boundary is represented by a set of unit squares which enclose unit cubes.

The polyhedral representation introduced in this paper belongs to the second group, non-interpolating representations. We express an object as a set of lattice points, like one of the representations in the second group. However, our approach is based on combinatorial topology [1] and uses combinatorial topological properties in addition to local neighborhood structures. Therefore, our polyhedral representation of objects preserves topological features underlying the objects even if the objects are represented by sets of lattice points. While classical combinatorial topology has been developed in Euclidean spaces [1], combinatorial topology is also required in lattice spaces, as discussed in references [12, 13]. Note that Françon defined discrete combinatorial surfaces by using 2D cells such as squares and triangles [12], while we define digital polyhedra by using 3D cells which we call 3D discrete simplexes in reference [13] and this paper.

We also introduce an efficient algorithm for implementing our method of generating polyhedra from lattice points. Our algorithm generates polyhedra by referring to pre-calculated look-up tables and performing simple vector calculations. One of the distinctive features of our algorithm is that it can be applied in the same way for any neighborhood in a 3D lattice space, such as the 6-, 18- or 26-neighborhood [8]. Moreover, polyhedra are always uniquely generated with respect to each neighborhood. Though a similar algorithm for generating polyhedral surfaces was previously introduced [14], the uniqueness feature was not proved, and also no neighborhood structures were considered. We also discuss the relation between the three polyhedra created under the three different neighborhoods relative to a given set of lattice points.

2 Definitions

Assuming that $\mathbf{Z}$ is the set of all integers, $\mathbf{Z}^3$ can be defined as the 3D lattice space which consists of points whose three coordinates are integers. In this section, we analyze the kinds of planes that can exist in $\mathbf{Z}^3$, which we call digital planes. Then, we define digital polygons which lie in such digital planes, and finally we define digital polyhedra.

2.1 Digital Planes and Neighborhoods

Digital planes are defined as planes normal to the elemental direction vectors [2]. There are 26 elemental direction vectors and each vector has a direction code (i, j), as illustrated in Fig. 1. The vector with direction code (i, j) is represented by $\boldsymbol{q}_{ij}$; it can be drawn from point $(0, 0)$ to point (i, j) in Fig. 1. A direction vector $\boldsymbol{q}_{ij}$ always has an inverse vector. If a vector and its inverse vector define

a pair, then there are only 13 pairs of direction vectors, as can be seen from Fig. 1. The reduced set of 13 vectors are classified into three groups depending on their lengths as follows:

$$\boldsymbol{s} = \{\boldsymbol{q}_{11}, \boldsymbol{q}_{31}, \boldsymbol{q}_{33}\} \ , \tag{1}$$

$$\boldsymbol{r} = \{\boldsymbol{q}_{21}, \boldsymbol{q}_{41}, \boldsymbol{q}_{23}, \boldsymbol{q}_{27}, \boldsymbol{q}_{38}, \boldsymbol{q}_{32}\} \ , \tag{2}$$

$$\boldsymbol{h} = \{\boldsymbol{q}_{22}, \boldsymbol{q}_{24}, \boldsymbol{q}_{26}, \boldsymbol{q}_{28}\} \ , \tag{3}$$

where the length of each vector in the set $\boldsymbol{s}$, $\boldsymbol{r}$ or $\boldsymbol{h}$ is 1, $\sqrt{2}$ or $\sqrt{3}$, respectively.

We use the notation $C_{ij}(\boldsymbol{x})$ to describe the digital plane which has normal vector $\boldsymbol{q}_{ij}$ and passes through the point $\boldsymbol{x}$. Since $C_{ij}(\boldsymbol{x})$ is a subset of $\mathbf{Z}^3$, $C_{ij}(\boldsymbol{x})$ can be regarded as a 2D lattice space embedded in $\mathbf{Z}^3$. The shape of the lattice depends on the normal vector of $C_{ij}(\boldsymbol{x})$, that is, on $\boldsymbol{q}_{ij}$. If $\boldsymbol{q}_{ij}$ is included in $\boldsymbol{s}$, each cell of the lattice of $C_{ij}(\boldsymbol{x})$ is a square whose side lengths are 1. If $\boldsymbol{q}_{ij}$ is included in $\boldsymbol{r}$, each cell of the lattice of $C_{ij}(\boldsymbol{x})$ is a rectangle whose long side lengths are $\sqrt{2}$ and whose short side lengths are 1. If $\boldsymbol{q}_{ij}$ is included in $\boldsymbol{h}$, each cell of the lattice of $C_{ij}(\boldsymbol{x})$ is an equilateral triangle whose side lengths are $\sqrt{2}$. These three kinds of 2D lattice spaces are denoted by $S_{ij}(\boldsymbol{x})$, $R_{ij}(\boldsymbol{x})$ and $H_{ij}(\boldsymbol{x})$, resepectively; examples are shown in Fig. 1.

Let $\boldsymbol{n}(6)$, $\boldsymbol{n}(18)$ and $\boldsymbol{n}(26)$ be $\boldsymbol{s}$, $\boldsymbol{s} \cup \boldsymbol{r}$ and $\boldsymbol{s} \cup \boldsymbol{r} \cup \boldsymbol{h}$, repspectively; then the 6-, 18- and 26-neighborhoods [8] of a point $\boldsymbol{x}$ in $\mathbf{Z}^3$ are defined by

$$N_m(\boldsymbol{x}) = \{\boldsymbol{y} \mid \boldsymbol{x} - \boldsymbol{y} = \pm \boldsymbol{q}, \ \mathbf{y} \in \mathbf{Z}^3, \ \boldsymbol{q} \in \boldsymbol{n}(m)\} \ , \tag{4}$$

where $m = 6, 18,$ or $, 26$. The choice of neighborhood determines which kinds of digital planes can be defined. If the 6-neighborhood is considered, the resulting digital planes can only be $S_{ij}(\boldsymbol{x})$, while all three types of digital planes, $S_{ij}(\boldsymbol{x})$, $R_{ij}(\boldsymbol{x})$ and $H_{ij}(\boldsymbol{x})$, exist if the 18- or 26-neighborhood is considered. This is because at least two different direction vectors on a digital plane are necessary when we draw digital polygons in the digital plane and these direction vectors must be included in $\boldsymbol{n}(m)$ if the m-neighborhood is considered. For example, the digital polygons in $H_{22}(\boldsymbol{x})$ illustrated in Fig. 1 can be described by using two direction vectors, $\boldsymbol{q}_{27}$ and $\boldsymbol{q}_{38}$, which are included in $\boldsymbol{n}(18)$ and $\boldsymbol{n}(26)$. Thus, $H_{22}(\boldsymbol{x})$ is defined if we consider either the 18- or 26-neighborhood.

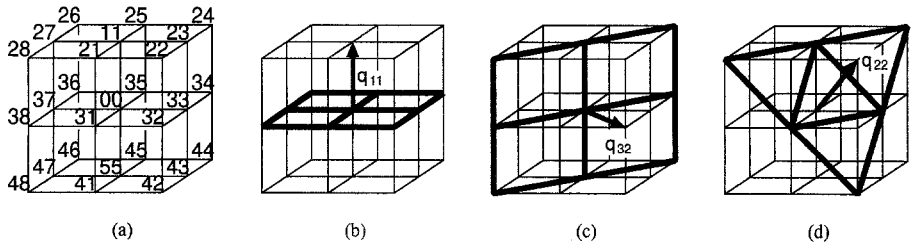

Fig. 1. (a) Direction codes (i, j) of 26 elemental direction vectors in $\mathbf{Z}^3$. (b) A 2D lattice space, $S_{11}(\boldsymbol{x})$. (c) A 2D lattice space, $R_{32}(\boldsymbol{x})$. (d) A 2D lattice space, $H_{22}(\boldsymbol{x})$.

2.2 Digital Polygons

A digital polygon is defined on a digital plane. Its definition is given in the same way as the definition of a polygon in 3D Euclidean space [15]. Let $perp(\boldsymbol{q}_{ij})$ be the set of all direction vectors perpendicular to $\boldsymbol{q}_{ij}$.

Definition 1. A digital polygon is a finite region surrounded by a sequence of p line segments, $\boldsymbol{x}_2 - \boldsymbol{x}_1$, $\boldsymbol{x}_3 - \boldsymbol{x}_2$, ..., $\boldsymbol{x}_1 - \boldsymbol{x}_p$, lying in $C_{ij}(\boldsymbol{x}_1)$, which satisfies the relation

$$\frac{1}{k}(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i) \in (perp(\boldsymbol{q}_{ij}) \cap \boldsymbol{n}(m)) \tag{5}$$

for $i = 1, 2, 3, \ldots, p$ and some $k \in \mathbf{Z}$, where $\boldsymbol{x}_{p+1} = \boldsymbol{x}_1$ and $m = 6, 18$ or 26. This digital polygon is denoted by

$$S = (\boldsymbol{x}_1 \boldsymbol{x}_2 \boldsymbol{x}_3 \ldots \boldsymbol{x}_p). \tag{6}$$

Note that S involves p points $\boldsymbol{x}_i$, the p line segments $\boldsymbol{x}_{i+1} - \boldsymbol{x}_i$, and the region enclosed by them. The points and the line segments are called vertices and edges, respectively. We shall require that the p line segments generate a simple closed curve, such that it has no intersections with itself, and also that the points are numbered in a counterclockwise fashion with respect to the inside of the simple closed curve. Because it is assumed that the polygon is viewed from the direction of the normal vector of the digital plane in which the polygon lies, the ordering of the points is uniquely determined by the normal vector.

We can extend the notion of digital polygons to treat not only simply connected polygons [15], but also polygons that contain holes, such as that illustrated in Fig. 2. If a digital polygon has n holes, the inside of the digital polygon is encircled by $n+1$ simple closed curves because each hole is encircled by a simple closed curve. Clearly, the holes themselves are also digital polygons, although the ordering of the points should be opposite from that in a simply connected digital polygon. Namely, the order is clockwise with respect to the inside of the polygon encircling a hole because the inside of the hole is regarded as the outside of the polygon. The extended notation for a digital polygon with n holes is

$$S = (\boldsymbol{x}_1 \boldsymbol{x}_2 \ldots \boldsymbol{x}_p;\ \boldsymbol{y}_{1(1)} \boldsymbol{y}_{1(2)} \cdots \boldsymbol{y}_{1(q1)};\ \cdots;\ \boldsymbol{y}_{n(1)} \boldsymbol{y}_{n(2)} \cdots \boldsymbol{y}_{n(qn)}) \tag{7}$$

where $(\boldsymbol{x}_1 \boldsymbol{x}_2 \ldots \boldsymbol{x}_p)$ is a simply connected digital polygon and $(\boldsymbol{y}_{i(1)} \boldsymbol{y}_{i(2)} \cdots \boldsymbol{y}_{i(qi)})$, for $i = 1, 2, \ldots, n$, is a hole in $(\boldsymbol{x}_1 \boldsymbol{x}_2 \ldots \boldsymbol{x}_p)$.

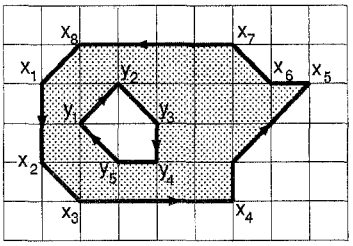

Fig. 2. An example of a digital polygon with a hole in a digital plane $S_{ij}(\boldsymbol{x})$.

2.3 Digital Polyhedra

Because we will define digital polyhedra in $\mathbf{Z}^3$ using concepts of combinatorial topology [1], it is necessary to introduce 3D volumetric elements in $\mathbf{Z}^3$, which we call 3D discrete simplexes [13]; 3D volumetric elements in Euclidean space are called 3D simplexes [1]. The dimensions of the possible simplexes in a 3D space range from 0 to 3; we abbreviate nD discrete simplexes to n-simplexes.

We can define n-simplexes in $\mathbf{Z}^3$ using neighborhood structures [13]. A 0-simplex is defined as an isolated point in $\mathbf{Z}^3$, and a 1-simplex is defined as a line segment whose endpoints neighbor each other. Thus, the 1-simplexes depend on the assumed neighborhood, as shown in the first line of Fig. 3. A 2-simplex is defined as a minimum non-zero area encircled by 1-simplexes; the possible 2-simplexes are shown in the second line of Fig. 3. A 3-simplex, which is necessary for our definition of a digital polyhedron, is defined as a minimum non-zero volume enclosed by 2-simplexes. They are shown in the last line of Fig. 3.

The 0-, 1- and 2-simplexes included in a 3-simplex are called the vertices, edges and faces of the 3-simplex, respectively. If the intersection of two 3-simplexes is a common vertex, edge or face of the 3-simplexes, we say that the 3-simplexes are adjacent. If, for a set of 3-simplexes, there exists a sequence of 3-simplexes between two arbitrary 3-simplexes in the set such that consecutive simplexes in the set are adjacent, the set is said to be connected. The definition of a digital polyhedron can now be given.

Definition 2. A digital polyhedron is a union of connected 3-simplexes.

This definition includes many cases that are not regarded as classical polyhedra [15], such as a union of two 3-simplexes which share only one vertex.

Because the boundaries of 3-simplexes are 2-simplexes, the surface of a digital polyhedron is described by a set of 2-simplexes. Note that the 2-simplexes in Fig. 3 are all in digital planes. If we combine 2-simplexes which are connected in a digital plane, a digital polygon is created from the connected 2-simplexes, as illustrated in Fig. 2. Therefore, a digital polyhedron can be represented by a set of digital polygons which are created by combining connected 2-simplexes in a digital plane. The notation for a digital polyhedron bounded by p digital polygons, $S_1, S_2, \ldots, S_p$, is

$$P = \{S_1, S_2, \ldots, S_p\} \ . \tag{8}$$

The counterclockwise ordering of the vertices in each digital polygon determines the side of the digital polygon which faces the exterior of the digital polyhedron P. The digital polygons which enclose P, their edges and their vertices are called the faces, edges and vertices of P, respectively.

We can derive the next proposition, which is helpful for the following discussion, from Definition 2. The proof is omitted since it can be derived from reference [1].

Proposition 1 *If two digital polyhedra, P_1 and P_2, have at least one common vertex, then $P_1 \cup P_2$ is also a digital polyhedron, P.*

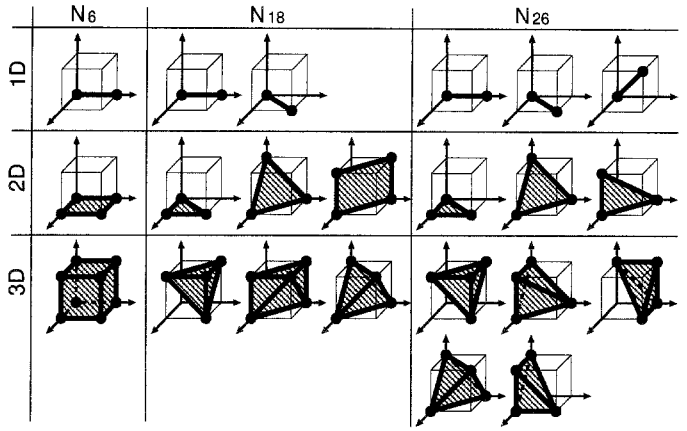

Fig. 3. All possible 1D, 2D and 3D simplexes for the 6-, 18- and 26-neighborhoods.

3 A Method for Generating Polyhedra

This section introduces a method for generating digital polyhedra from a subset of $\mathbf{Z}^3$, denoted by $\mathbf{V}$. Assume that every point in $\mathbf{Z}^3$ is assigned a value of 1 or 0, i.e., $\mathbf{Z}^3$ is binarized, and that the value of every point in $\mathbf{V}$ is 1 while that of every point in the complement of $\mathbf{V}$ is 0. Points in $\mathbf{V}$ and its complement are called 1-points and 0-points, respectively. Our method is divided into two stages, which are shown separately. We also give an algorithm as a summary of the method, as well as some properties of digital polyhedra.

3.1 Digital Polyhedra in Unit Cubes

Every 3-simplex illustrated in Fig. 3 is included in a unit cube, that is, the size of every 3-simplex is smaller than a unit cube. This fact enables us to embed 3-simplexes in each unit cube such that all vertices of the 3-simplexes are 1-points. We embed 3-simplexes into a unit cube following the rule that the volume occupancy of the 3-simplexes in the cube is a maximum; this rule helps to uniquely determine the regions that should be occupied by the 3-simplexes. These embedded 3-simplexes can be combined into a digital polyhedron in the unit cube according to Definition 2. Such a digital polyhedron in a unit cube will be called a unit digital polyhedron.

Let us consider all possible patterns of 1-points in a unit cube. Because each unit cube consists of eight lattice points and every lattice point is either a 1- or 0-point, the number of possible patterns is 256. By considering rotational symmetry, these patterns can be reduced to 23. These 23 possible patterns are shown in Fig. 4.

It is sufficient to consider the unit digital polyhedra defined by these 23 patterns. Table 1 shows the unit digital polyhedron defined by each 1-point pattern in a unit cube with respect to the 6-, 18- and 26-neighborhoods. Because

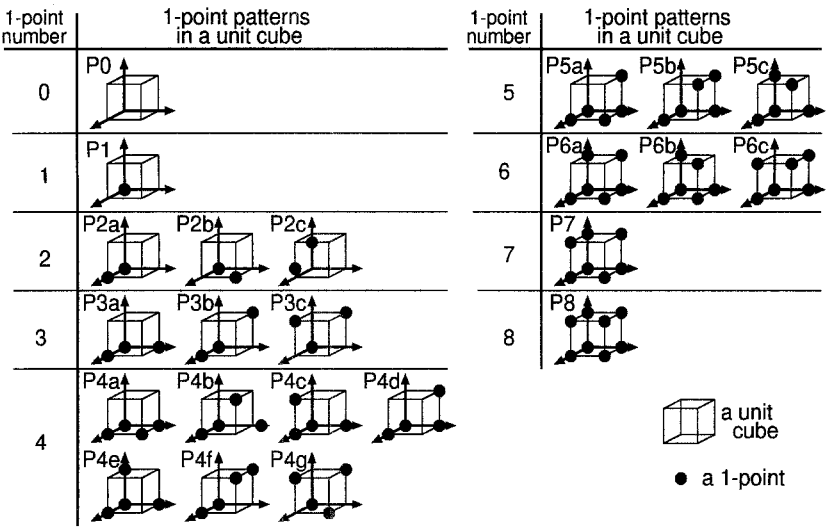

Fig. 4. Twenty-three possible patterns of 1-points in a unit cube; the set of all possible patterns is reduced by considering rotations in the 3D lattice space.

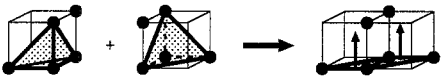

Fig. 5. The exceptional case in which two adjacent unit digital polyhedra cannot be combined with each other, and the elimination of the polyhedra in this case. This case occurs only under the 18-neighborhood.

the unit digital polyhedra created for the 18- and 26-neighborhoods are the same if the number of 1-points in the unit cube is 6, 7 or 8, their lines in Table 1 are given in common. The correspondence between the patterns of 1-points and the unit digital polyhedra is one-to-one, as shown in Table 1 for each neighborhood. Note that the table shows the unit digital polyhedra only if the correspondence exists. For instance, a unit digital polyhedron for pattern P8 is shown only in the case of the 6-neighborhood, because no pattern except for P8 corresponds to a unit digital polyhedron. This yields the fact that parts which are smaller than 3-simplexes, such as isolated points and sets of points forming needle-like and wall-like shapes, are excluded.

3.2 Combining Unit Digital Polyhedra

According to Proposition 1, two adjacent unit digital polyhedra can be combined into a digital polyhedron if they share a face, edge or vertex. We can observe that all possible pairs of adjacent unit digital polyhedra in Table 1 can be combined without contradiction except for the case illustrated in Fig. 5 under the 18-neighborhood. In that case, we retain the digital polygons in the horizontal plane of the base of the two unit cubes, and eliminate the rest of the unit polyhedra, as shown in Fig. 5.

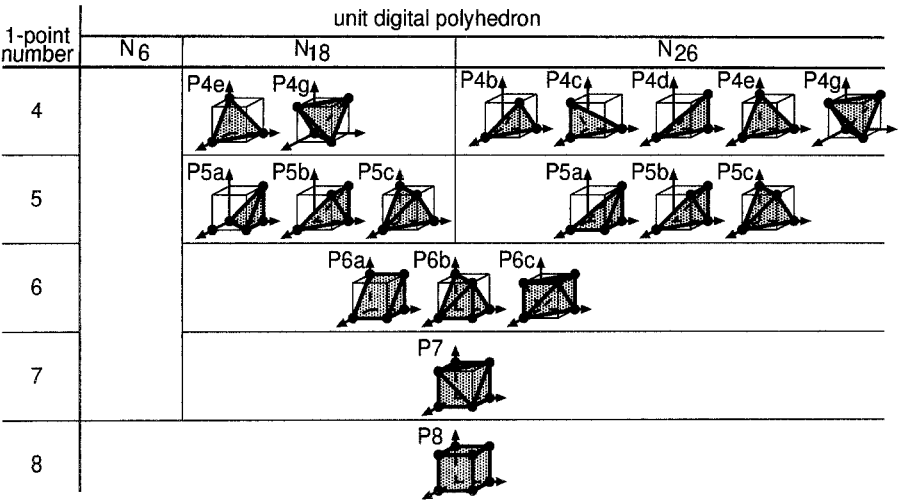

Table 1. The look-up table which provides the one-to-one correspondence between a pattern of 1-points and a unit digital polyhedron for the 6-, 18- and 26-neighborhoods.

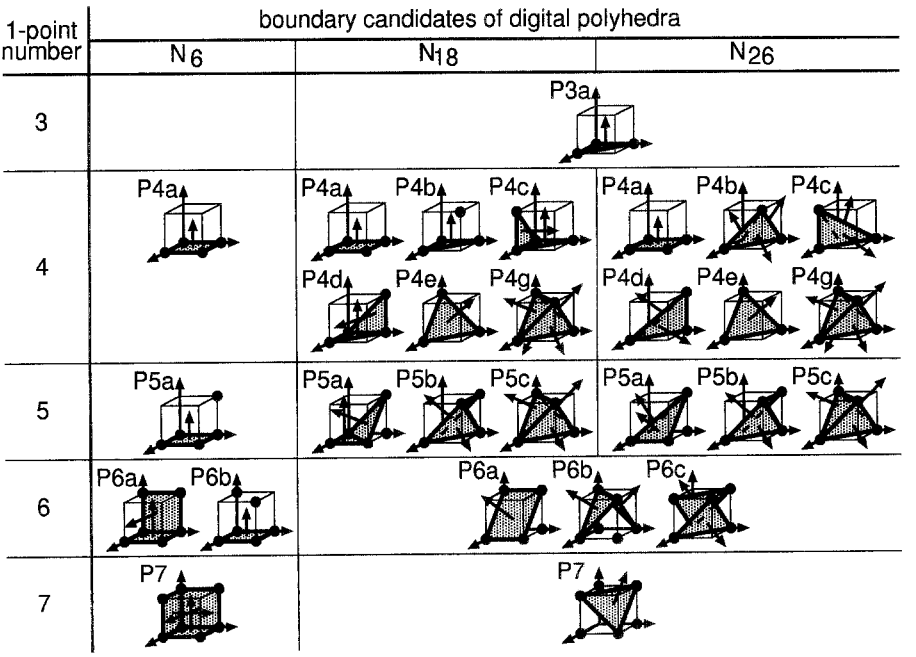

Table 2. The look-up table which provides the one-to-one correspondence between a pattern of 1-points and the boundary candidates of the combined digital polyhedra for the 6-, 18- and 26-neighborhoods.

If two adjacent unit digital polyhedra that share a face are combined into a single digital polyhedron, their shared face is not a face of the new digital polyhedron, but is interior to it. The process of combining unit digital polyhedra defines a new set of digital polygons which constitute the boundary of the new digital polyhedron. After iteration of the process of combining a set of connected unit digital polyhedra, a set of digital polygons which constitute the boundary of the combined digital polyhedron is obtained. We can directly obtain the set of digital polygons which are the boundary candidates of the combined digital polyhedra by using Table 2. The arrow attached to each digital polygon in the table indicates the direction to the exterior of the combined digital polyhedron. Table 2, which is based on Table 1, is created so that the exterior side of every digital polygon which is a boundary candidate must face the interior of the unit cube. Table 2 may create a pair of digital polygons whose vertices are the same but with arrows pointing in opposite directions. If such a pair arises, the polygons need to be erased from the set of boundary candidates.

We thus obtain a set of digital polygons which are the boundary candidates of the digital polyhedra which should be created from the given **V**. These digital polygons have to be merged if they share edges and lie in a digital plane, and if their arrows point in the same direction. If the digital polygon $S = (\boldsymbol{x}_1 \boldsymbol{x}_2 \ldots \boldsymbol{x}_p)$ lies in the plane $\boldsymbol{n} \cdot \boldsymbol{x} = \tau$, $\boldsymbol{n}$ and τ can be calculated by

$$\boldsymbol{n} = \frac{(\boldsymbol{x}_2 - \boldsymbol{x}_1) \times (\boldsymbol{x}_3 - \boldsymbol{x}_2)}{|(\boldsymbol{x}_2 - \boldsymbol{x}_1) \times (\boldsymbol{x}_3 - \boldsymbol{x}_2)|} \ , \tag{9}$$

$$\tau = \boldsymbol{n} \cdot \boldsymbol{x}_1 \ . \tag{10}$$

Let S_1, S_2, ..., S_p be digital polygons in the plane $\boldsymbol{n} \cdot \boldsymbol{x} = \tau$. If $S_i = (\boldsymbol{x}_1 \boldsymbol{x}_2 \ldots \boldsymbol{x}_p)$ and $S_j = (\boldsymbol{y}_1 \boldsymbol{y}_2 \ldots \boldsymbol{y}_q)$ are adjacent, then there exists at least one sequence of vertices common to S_i and S_j:

$$\boldsymbol{x}_s = \boldsymbol{y}_{(t+r) \oslash q}, \ \boldsymbol{x}_{(s+1) \oslash p} = \boldsymbol{y}_{(t+r-1) \oslash q}, \ \ldots, \ \boldsymbol{x}_{(s+r) \oslash p} = \boldsymbol{y}_t \tag{11}$$

where $a \oslash b$ is the remainder of the division of a by b and r is greater than 1 and less than both p and q. Because the numbering of the vertices of a digital polygon must be counterclockwise, the orders of the common vertices are different in S_i and S_j. If there is only one sequence of common vertices, the digital polygons S_i and S_j can then be combined into a digital polygon denoted by

$$S' = (\boldsymbol{x}_1 \ldots \boldsymbol{x}_{s-1} \boldsymbol{y}_{t+1} \ldots \boldsymbol{y}_q \boldsymbol{y}_1 \ldots \boldsymbol{y}_{(t+r) \oslash q} \boldsymbol{x}_{(s+r) \oslash p} \ldots \boldsymbol{x}_p) \ . \tag{12}$$

Even if S_i and S_j have more than one sequence of common vertices, the merging process can be applied to each sequence; the result will be more than one sequence, representing a polygon with holes. Moreover, the merging process can be applied to all adjacent pairs of digital polygons in the same digital plane. Because the S' which is finally obtained may not be correctly represented as a digital polygon, the vertices of S' must be reduced such that

$$\boldsymbol{x}_{i+2} - \boldsymbol{x}_{i+1} \neq s(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i) \tag{13}$$

where s is a rational number. Finally, we obtain a set of digital polygons, S'_1, ..., S'_h, for all digital planes. These polygons bound one or more digital polyhedra.

3.3 Algorithm

The algorithm for generating digital polyhedra from a given set of lattice points $\mathbf{V}$ for the 6-, 18- and 26-neighborhoods is as follows:

Algorithm 1

input: *A set of lattice points,* $\mathbf{V}$.
output: *Digital polyhedra,* $P_1, P_2, \ldots, P_k$.
begin

1. *For each unit cube generate a set of digital polygons* P *of boundary candidates by lookup in Table 2.*
2. *If the* 18*-neighborhood is considered, check whether the exceptional case illustrated in Fig. 5 occurs. If so, exclude these digital polygons from* P *and add the new flat digital polygons in Fig. 5 to* P.
3. *Calculate the vector* $\boldsymbol{n}$ *and the scalar* τ *of a digital plane on which each digital polygon in* P *lies using* (9) *and* (10).
4. *If two digital polygons consist of the same vertices and their* $\boldsymbol{n}$*s point in the opposite directions, exclude them from* P.
5. *Select digital polygons with the same values of* $\boldsymbol{n}$ *and* τ *from* P *and merge them if they share edges.*
6. *Extract connected components from* P *and number them as* $P_1, P_2, \ldots, P_k$.

end

This algorithm leads to the next theorem.

Theorem 1 *Digital polyhedra are uniquely generated from* $\mathbf{V}$ *for a given neighborhood.*

Proof. Table 2 in step 1 provides a one-to-one correspondence between $\mathbf{V}$ and P. The other steps in the algorithm provide one-to-one modifications of P. Therefore, $P_1, P_2, \ldots, P_k$ are uniquely obtained from $\mathbf{V}$.

3.4 Properties of Digital Polyhedra

According to Theorem 1, digital polyhedra are uniquely generated from $\mathbf{V}$ for each neighborhood. The relation between the digital polyhedra for the different neighborhoods is described in the following theorem.

Theorem 2 *If* $P[6]$, $P[18]$ *and* $P[26]$ *are the digital polyhedra generated from a lattice-point set* $\mathbf{V}$ *for the* 6-, 18- *and* 26*-neighborhoods, respectively, then*

$$P[6] \subseteq P[18] \subseteq P[26] \ . \tag{14}$$

Proof. According to our method, $P[6]$, $P[18]$ and $P[26]$ consist of 3-simplexes as given in Table 1, for the 6-, 18- and 26-neighborhoods, respectively. Table 1 shows us the differences between the unit digital polyhedra depending on the neighborhoods, which are denoted by $U[6]$, $U[18]$ and $U[26]$, in a unit cube. By

comparison between the shapes of $U[6]$, $U[18]$ and $U[26]$, the following relation is established for all the 23 patterns in Fig. 4:

$$U[6] \subseteq U[18] \subseteq U[26] \ . \tag{15}$$

Because for each unit cube (15) is satisfied and $P[m]$ is a union of $U[m]$s where $m = 6, 18, 26$, the relation (14) is obtained.

4 Implementation

The data set we used was obtained from a 3D single-photon emission computer tomography (SPECT) cardiac image. It includes 32 slices, each of which has 64×64 pixels. Each pixel has 2-byte gray values. Because our method requires binary data, the data was binarized by Otsu's threshold method [16]. Figure 6 shows some of the 32 slices of the binarized SPECT cardiac images. White and black pixels have values 1 and 0, respectively. White pixels are part of the myocardium while black pixels fill the background. Figure 6 indicates that there is a hollow bounded by the myocardium. The three digital polyhedra generated from this data set by our algorithm are shown in Fig. 7 for the 6-, 18- and 26-neighborhoods.

Fig. 6. Some of the 32 slices of the binarized SPECT cardiac image.

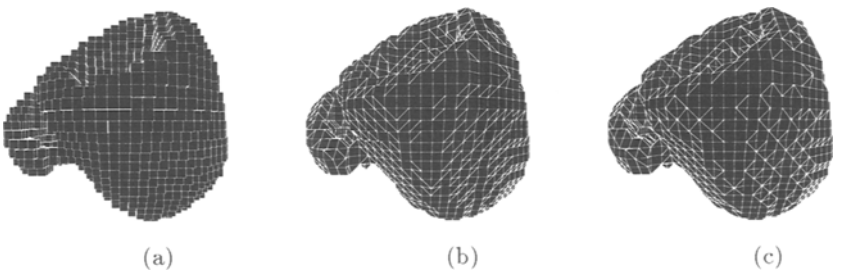

Fig. 7. The results of applying our polyhedra generation algorithm to the SPECT cardiac image for the 6- (a), 18- (b), and 26-neighborhoods (c).

5 Conclusions

This paper has introduced an algorithm for uniquely generating polyhedra in a 3D lattice space from a subset of the space, using the 6, 18- or 26-neighborhood. The three types of polyhedra have different shapes and satisfy the inclusion relations (14). Our algorithm was successfully implemented and applied to a

set of medical images. Our polyhedra are not 2-manifolds [12]. However, they can be modified into 2-manifolds if we use a combinatorial topology property, such as umbrella [12] or star [1, 13]. The first author expresses much thanks for the advice of Professor A. Rosenfeld at the Center for Automation Research, University of Maryland. This work was supported in part by National Library of Medicine Grant R29LM04692 and NIH grant 1R01HL42052.

References

1. P. S. Aleksandrov, *Combinatorial Topology*, Vol. 1, Graylock Press, Rochester, NY, 1956.
2. A. Imiya, Structure of 3-dimensional Digital Neighborhood and Its Application (in Japanese), Information Processing Society Journal, Vol. 34, pp. 2153-2164, Information Processing Society of Japan, 1993.
3. A. A. G. Requicha and H. B. Voelcker, Solid Modeling: A Historical Summary and Contemporary Assessment, IEEE Computer Graphics and Applications, Vol. 2, No. 2, pp. 9-24, 1982.
4. M. Chao, Y. Shen and W. Zhao, Solid Modeling Based on Polyhedron Approach, Computers and Graphics, Vol. 16, pp. 101-105, 1992.
5. S. N. Srihari, Representation of Three Dimensional Digital Images, Computing Surveys, Vol. 13, pp. 399-424, 1981.
6. J. D. Foley, A. V. Dam, S. T. Feiner and J. F. Hughes, *Computer Graphics*, Addison-Wesley, Reading, MA, 1995.
7. W. E. Lorensen and H. E. Cline, Marching Cubes: A High-resolution 3D Surface Construction Algorithm, Computer Graphics (SIGGRAPH '87), Vol. 21, pp. 163-169, 1988.
8. T. Y. Kong and A. Rosenfeld, Digital Topology: Introduction and Survey, Computer Vision, Graphics, and Image Processing, Vol. 48, pp. 357-393, 1989.
9. V. A. Kovalevsky, Finite Topology as Applied to Image Analysis, Computer Vision, Graphics, and Image Processing, Vol. 46, pp. 141-161, 1989.
10. G. T. Herman, Discrete Multidimensional Jordan surfaces, CVGIP: Graphical Models and Image Processing, Vol. 54, pp. 507-515, 1992.
11. J. K. Udupa, Multidimensional Digital Boundaries, CVGIP: Graphical Models and Image Processing, Vol. 56, pp. 311-323, 1994.
12. J. Françon, Discrete Combinatorial Surfaces, CVGIP: Graphical Models and Image Processing, Vol. 57, pp. 20-26, 1995.
13. Y. Kenmochi, A. Imiya and A. Ichikawa, Discrete Combinatorial Geometry (submitted).
14. E. Lee, Y. Choi and K. H. Park, A Method of 3D Object Reconstruction from a Series of Cross-Sectional Images, IEICE Transactions on Information and Systems, Vol. E77-D, pp. 996-1004, 1994.
15. H. S. M. Coxeter, *Regular Polytopes*, Dover Publications, New York, 1973.
16. N. Otsu, A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-9, pp. 62-66, 1979.