Describing polyhedral tilings and higher dimensional polytopes by sequence of their two-dimensional components

Kengo Nishio & Takehide Miyazaki

Polyhedral tilings are often used to represent structures such as atoms in materials, grains in crystals, foams, galaxies in the universe, etc. In the previous paper, we have developed a theory to convert a way of how polyhedra are arranged to form a polyhedral tiling into a codeword (series of numbers) from which the original structure can be recovered. The previous theory is based on the idea of forming a polyhedral tiling by gluing together polyhedra face to face. In this paper, we show that the codeword contains redundant digits not needed for recovering the original structure, and develop a theory to reduce the redundancy. For this purpose, instead of polyhedra, we regard two-dimensional regions shared by faces of adjacent polyhedra as building blocks of a polyhedral tiling. Using the present method, the same information is represented by a shorter codeword whose length is reduced by up to the half of the original one. Shorter codewords are easier to handle for both humans and computers, and thus more useful to describe polyhedral tilings. By generalizing the idea of assembling two-dimensional components to higher dimensional polytopes, we develop a unified theory to represent polyhedral tilings and polytopes of different dimensions in the same light.

Partitioning a space with points into polyhedra in such a way that each polyhedron encloses exactly one point and then characterizing the polyhedral tiling is a promising strategy to study a wide range of structures. For example, in studying the atomic structure of a material, the space can be divided into the so-called Voronoi polyhedra, where each polyhedron encloses its associated atom. By using this method, for example, a way of how an atom X is surrounded by its first and second nearest-neighbour atoms is represented by the local tiling structure composed of the Voronoi polyhedra associated with the atom X and its first nearest-neighbour atoms.

Since such a local tiling structure can be regarded as a part of a four-dimensional polytope (4-polytope) called a polychoron, a method to describe how a polychoron is constructed from its building-block polyhedra can be used to study the structure of materials. For this reason, we have recently developed a theory of polytopes that is based on the hierarchy of structures of polytopes: a polyhedron (3-polytope) is a tiling by polygons, a polychoron (4-polytope) is a tiling by polyhedra (3-polytopes), and so on. Specifically, we have first created the $p_3$-code for representing polyhedra. The $p_3$-code consists of (1) an encoding algorithm for converting a way of how polygons are arranged to form a polyhedron into a $p_3$-codeword ($p_3$ for short) and (2) a decoding algorithm for recovering the original polyhedron from its $p_3$. By generalizing the $p_3$-code, we have created the $p_4$-code for representing polychora. By using the $p_4$-code, we can form a polychoron that can be converted into a $p_4$-codeword ($p_4$ for short), from which the original polychoron can be recovered. A polyhedral tiling can be characterized by distribution of $p_4$s of local tiling structures of different central polyhedra. However, $p_4$ is redundant as described below.

The $p_4$-codeword contains $p_4(1)$, $p_4(2)$, $p_4(3)$, ..., and $p_4(C)$, where $p_4(i)$ is $p_4$ of the polyhedron $i$ and $C$ is the number of polyhedra of the polychoron. Each $p_4(i)$ contains $p_4(i), p_4(i), p_4(i), ..., p_4(i)$, where each $p_4(i)$ is the number of edges of the face $j$ of the polyhedron $i$ and $F(i)$ is the number of faces of the polyhedron $i$. Here, we point out that $p_4(i)$ is of all the faces of all the polyhedra are recorded in $p_4$. However, since polyhedra are glued together face to face, the pair of faces glued each other have the same number of edges. $p_4(i)$ is thus redundant and...
lengthy. For example, if the face \( y \) of the polyhedron \( x \) is glued to the face \( w \) of the polyhedron \( v \), then \( p_s(v_w) = p_s(x) \), so that \( p_s(v_w) \) in \( p_s \) is redundant.

Redundant codewords mean the lack of knowledge of structures of polychora. In addition, redundant codewords are practically unfavourable for both humans and computers. For humans, recognizing and writing down lengthy codewords are troublesome. For computers, larger hard drives are necessary to store codewords and more computation time is necessary to determine the equivalence of codewords.

In this paper, we develop a theory to reduce the redundancy in \( p_s \). For this purpose, we exploit the fact that the polyhedra are glued together face to face. Specifically, we regard two-dimensional regions shared by faces of adjacent polyhedra as building blocks of a polychoron. To distinguish between parts of a polychoron and parts of a polyhedron, we refer the two-dimensional building blocks of a polychoron to ridges. As the distinction between edges of a polyhedron and sides of a polygon was crucial for L. Euler to find his famous polyhedral formula, \( V - E + F = 2^4 \), the distinction between ridges of a polychoron and faces of a polyhedron is crucial for our theory. To represent a polyhedron using ridges, we formulate a method to convert \( p_s \) into \( p_s^{(4)} \), where the superscript “\( rs \)” indicates the ridge-sequence. Note that \( p_s \) instructs how to construct a polychoron from its building-block polyhedra, while \( p_s^{(4)} \) instructs how to construct a polychoron from its building-block ridges. The length of \( p_s^{(4)} \) is as short as half of \( p_s \). By generalizing the method to higher-dimensional polytopes, we develop a unified theory of how a polytope is constructed from its two-dimensional components.

**Results**

**Bare essentials of the \( p_s \)-code.** We will formulate the \( p_s^{(4)} \)-code consisting of (1) an encoding algorithm for converting \( p_s \) into \( p_s^{(4)} \) and (2) an decoding algorithm for recovering the original polychoron or \( p_s \) from \( p_s^{(4)} \). We start with the brief explanation of bare essentials of the \( p_s \)-code needed to formulate the \( p_s^{(4)} \)-code. Specifically, we explain how to recover a polychoron from \( p_s \). For reader’s convenience, the encoding algorithm is described in Supplementary Note. The full details of the \( p_s \)-code has been given in the previous paper\(^{13} \). Since polychora associated with disordered structures are simple, we deal with simple polychora. By a simple polychoron, we mean a polychoron whose 0-faces are all incident with four peaks. Here, 0-faces and peaks are zero- and one-dimensional components of a polychoron, respectively. Since a simple polychoron is composed of simple polyhedra, we first explain the \( p_s \)-code for simple polyhedra. By a simple polyhedron, we mean a polyhedron whose vertices are all incident with three edges.

A polyhedron can be regarded as a tiling by polygons of the surface of a three-dimensional object that is topologically the same as a sphere. According to the idea developed by L. Euler, A. M. Legendre, F. Möbius, and P. R. Cromwell\(^{14} \), we assume that polyhedra are glued such that (1) any pair of polygons meet only at their sides or corners and that (2) each side of each polygon meets exactly one other polygon along an edge. We stress that we distinguish between parts of a polyhedron and those of the building-block polygons. Specifically, vertices and edges are zero- and one-dimensional parts of a polyhedron, respectively. On the other hands, corners and sides are zero- and one-dimensional parts of a polygon, respectively. Since this idea plays a central role in our theory, we need a verb to briefly describe the relation between parts of a polyhedron and those of polygons. For this purpose, we use the verb “contribute”. For example, when we say that the corners contribute to the vertex or the vertex is contributed by the corners, we mean that the vertex is a point on a polyhedron at which the corners of polygons meet. We also say that a polygon (side) contributes to a vertex if one of its corners (endpoints) contributes to the vertex. Similarly, when we say that the edge is contributed by the sides, we mean that the edge is a line segment on a polyhedron along which the sides of polygons meet. The face of a polyhedron is a polygon. But when we call a polygon, we regard it as a building block of a polyhedron. So, we may say the edge of a face. But we cannot say the edge of a polygon.

Using the \( p_s \)-code, a way of how polygons are arranged to form a polyhedron can be converted into \( p_s \), which instructs how to construct the polyhedron from its building-block polygons. The \( p_s \)-codeword consists of the polygon-sequence codeword \((p_s)\) and the side-pairing codeword \((sp)\), and is denoted as

\[
p_s = p_s^{(4)};sp,
\]

where “;” is a separator. The \( p_s \)-codeword is denoted as

\[
p_s = p_s(1)p_s(2)p_s(3)\cdots p_s(F).
\]

Here, \( p_s(i) \) is the number of sides of the polygon \( i \), and \( F \) is the number of polygons of the polyhedron. We note that the number of sides of the polygon \( i \) is identical with the number of edges of the face \( i \).

If we know all information of \( p_s(i) \) and all information about which side should be glued to which side, we can construct a polyhedron by gluing polygons side to side. The \( p_s \)-codeword contains not only all information of \( p_s(i) \)'s, but also all or almost all information about which side should be glued to which side. Many polyhedra are represented just by \( p_s \), but there are some polyhedra that need additional information about which side should be glued to which side. Such additional information is recorded in \( sp \), which is denoted as

\[
sp = y(1)x(1)y(2)x(2)y(3)x(3)\cdots y(N_s)x(N_s).
\]

Here, \( y(i) \) and \( x(i) \) are the identification numbers (IDs) of sides. The pair of \( y(i)x(i) \) is what we call a non-curable additional pair (side-na-pair \( y(i)x(i) \)). By a side-na-pair \( y(i)x(i) \), we mean that the sides \( y(i) \) and \( x(i) \) should be glued together. Here, \( y(i) > x(i) \) and \( y(i) < y(i+1) \). \( N_s \) is the number of side-na-pairs.

Decoding \( p_s \) is constructing its original polyhedron by gluing together polygons side to side. To instruct which side should be glued to which side, we assign IDs to sides. We assign \( i \) to the \( j \)th side of the polygon \( i \), and the side-ID \( i \) represents an integer: \( i = j + \sum_{x=1}^{i-1} p_s(x) \). In constructing a polyhedron, if a side of a polygon of the
partial polyhedron is not glued to the other polygon, we call it a dangling side. We abbreviate the smallest-ID dangling side as the s-side. We regard that an isolated corner as well as two corners meeting at a point forms a vertex of a partial polyhedron. We also regard that a dangling side forms an edge. If the pair of dangling sides contribute to a vertex that is also contributed by three polygons, that vertex is said to be illegal. When an illegal vertex (i-vertex) is generated, we rectify it by gluing together the two dangling sides contributing to it. The polyhedron can be recovered from \( ps_sp \) as follows:

1. (a) The polygon 1 is a \( p_1(1) \)-gon.
   (b) Assign IDs \( (1, 1_2, 1_3, \ldots, 1_p(1)) \) to its sides in a clockwise (CW) direction.

2. (a) The next polygon \( i \) \((2 \leq i \leq F)\) is a \( p_1(i) \)-gon.
   (b) Assign IDs \( (i_1, i_2, i_3, \ldots, i_p(1)) \) to its sides in a CW direction.
   (c) Glue the side \( i_1 \) to the s-side of the partial polyhedron.
   (d) If \( \gamma(n) \) \((1 \leq n \leq N)\) is the side ID of the polygon \( i \), then glue the side \( \gamma(n) \) to the side \( x(n) \) of the partial polyhedron.
   (e) If i-vertices are generated, then rectify them, and repeat this procedure until no i-vertices remain.

3. (a) Repeat the procedure 2 until all polygons are placed.

   The edge IDs are assigned as follows. Given that each edge is contributed by two sides, we tentatively assign the smaller side ID to the edge, and then relabel IDs so that the edge \( i \) is the one with the \( i \)-th smallest-ID tentative ID.

   We note that the \( p_3 \)-code can be used to represent a tiling by polygons of a torus without modification. But to represent a toroidal polyhedron, we need to specify how to embed the torus in the 3-dimensional Euclidean space to form a toroidal polyhedron. The \( p_3 \)-code can also be generalized to non-orientable planes such as the Klein bottle by defining the clockwise direction for the polygon \( i \), in which IDs are assigned to sides, depending on the clockwise direction for the polygon to which the side \( i \) is glued.

   The \( p_3 \)-code is generalized to the \( p_3 \)-code for polychora as follows. We regard a polychoron as a tiling by polyhedra of the surface of a four-dimensional object that is topologically the same as a 3-sphere. We assume that polyhedra are glued together such that (1) any pair of polyhedra meet only at their faces, edges, or vertices and that (2) each face of each polyhedron meets exactly one other polyhedron along a ridge. We distinguish parts of a polychoron and parts of its building-block polyhedra. The \( 0 \)-face, peak, and ridge are a point, line segment, and area on a polychoron, where the vertices, edges, and faces of polyhedra meet, respectively. The cell of a polychoron is a polyhedron.

   The \( p_3 \)-codeword consists of a polyhedron-sequence codeword \( (ps) \) and a face-pairing codeword \( (fp) \), and is denoted as

   \[
   p_3 = ps_fp \tag{4}
   \]

   Here,

   \[
   ps = p_1(1)p_2(2)p_3(3) \cdots p_3(C) \tag{5}
   \]

   \( C \) is the number of polyhedra of the polychoron. \( p_i(i) = ps_i(i); sp(i) = p_i \) of the polyhedron \( i \).

   The \( fp \)-codeword consists of what we call face-na-pairs \( wzv \), and is denoted as

   \[
   fp = w(1)z(1)v(1) \cdots w(N_f)z(N_f)v(N_f) \tag{6}
   \]

   Here, \( w(i) \) and \( v(i) \) are face IDs, \( w(i) > v(i) \) and \( w(i) < w(i + 1) \). \( z(i) \) is the global side ID of a side of the polygon \( w(i) \). The global side IDs will be explained later. By a face-na-pair \( w(i); z(i); v(i) \), we mean that the faces \( w(i) \) and \( v(i) \) should be glued together in such a way that the edge of the face \( w(i) \) contributed by the side \( z(i) \) is glued to the smallest-ID edge of the face \( v(i) \). \( N_f \) is the number of face-na-pairs of the polychoron. Note that, in order to formulate \( ps_i; fp \) of the present work is slightly modified from the original definition. For the original definition, \( z(i) \) is the edge ID of the edge of the face \( w(i) \) glued to the smallest-ID edge of the face \( v(i) \).

   In decoding \( p_3 \), if a face of a polyhedron of the partial polyhedron is not glued to the other face, we call it a dangling face. By the edge \( i \), we mean the \( j \)-th edge (face \( j \)) of the polyhedron \( i \). We abbreviate the smallest-ID dangling face as the s-face. We regard that an isolated edge as well as two edges meeting along a line segment forms a peak of a partial polyhedron. We also regard that a dangling face forms a ridge. In a partial polyhedron, if the pair of dangling faces contribute to a peak that is also contributed by three polyhedra, we call that peak an illegal peak (i-peak). When an i-peak is generated, we rectify it by gluing together the two dangling faces contributing to it. The polyhedron can be recovered from \( ps_sp \) as follows:

1. (a) Decode \( p_1(1) \) to obtain the polyhedron 1, assigning face and edge IDs.
2. (a) Decode \( p_i(i) \) to obtain the next polyhedron \( i \) \((2 \leq i \leq C)\), assigning face and edge IDs.
   (b) Glue the face \( i_1 \) to the s-face of the partial polyhedron in such a way that the edge \( i_1 \) is glued to the smallest-ID edge of the s-face.
   (c) If \( \gamma(n) \) \((1 \leq n \leq N)\) is the face ID of the polyhedron \( i \), then glue the face \( \gamma(n) \) to the face \( v(n) \) of the partial polyhedron in such a way that the edge of the face \( \gamma(n) \) contributed by the side \( z(n) \) is glued to the smallest-ID edge of the face \( v(n) \).
   (d) If i-peaks are generated, then rectify them, and repeat this procedure until no i-peaks remain.

3. (a) Repeat the procedure 2 until all polyhedra are placed.
Ridge IDs are assigned as follows. Given that each ridge is contributed by two faces, we tentatively assign the smallest face ID to the ridge. We call the ID thus assigned the tentative ridge ID. We then relabel IDs so that the ridge is the one with the $i$th smallest tentative ID. The tentative ridge IDs and ridge IDs play a key role in reducing the redundancy in $p_s$. Peak IDs are also assigned similarly.

**Preliminary arrangements.** An outline of converting $p_s$ into $p_s^{(n)}$ is illustrated in Fig. 1. We will first break $p_i$ down into its $p_s$s and $p_sp$, and reconstruct $p_s^{(n)} = p_s^{(*)}sp^s_fp^p$, which provides us a good perspective for reducing the redundancy. We will then reduce the redundancy by converting $p_s^{(n)}$ into $p_s^{(*n)} = rs^sp^s_{*fp^p}$. Finally, to make our theory more beautiful, we unify $sp^s$ and $fp$ into a part-pairing codeword ($pp$), and obtain $p_s^{(n)} = rsa_pp$.

To formulate $p_s^{(n)}$, we distinguish local IDs and global IDs. When we call the polygon $j$ of the polyhedron $i$, the number $j$ is what we call the local polygon ID associated with the polyhedron $i$. We can designate the same polygon as the polygon $i$. The symbol $i$ is what we call the global polygon ID of the polyhedron $i$. The symbol $i$ also represents the number $i = j + \sum_{k=1}^{n-1} F(k)$, where $F(k)$ is the number of polygons of the polyhedron $k$. Similarly, by the side $m_n$ of the polyhedron $i$, we mean the $n$th side of the polygon $m$ of the polyhedron $i$. The number $n$ is the local side ID associated with the polygon $m$ of the polyhedron $i$, while the symbol $m$ is the local side ID associated with the polyhedron $i$ and is also the number $m = n + \sum_{k=1}^{n-1} p_s i_k$.

Here, $p_s i_k$ is the number of sides of the polygon $i$. Using the global side ID, we can designate the side $m_n$ of the polyhedron $i$ as the side $m_n$. The symbol $m_n$ also represents the number $m_n = m_n + 2\sum_{k=1}^{n-1} E(k)$, where $E(k)$ is the number of edges of the polyhedron $k$.

The $sp^s$-codeword of $p_s^{(*)} = p_s^{(*)}sp^s_{*fp^p}$ is written using the local side IDs associated with the polyhedron $i$. Using the global side IDs, we rewrite $p_s^{(*)} = p_s^{(*)}sp^s_{*fp^p}$. Here, $sp^s_{*fp^p} = y^p(i_x)x^p(i_y)y^x(i_z)x^x(i_z)\ldots y^x(i_{N_x,i})(x(i_{N_x,i}))$ obtained by translating $sp^s_{*fp^p} = y(i_x)x(i_y)y(i_z)x(i_z)\ldots y(i_{N_x,i})(x(i_{N_x,i}))$ from local side ID into global side ID.

By putting together $sp^s_{*fp^p}$, $sp^s_{*fp^p}$ is defined as follows:

$$sp^s_{*fp^p} = sp^s_{*fp^p} = sp^s(1)sp^s(2)sp^s(C)$$

$$y^x(1_x)x^y(1_y)y^x(1_z)x^y(1_{N_x,1})\ldots y^x(1_{N_x,i_x})x^y(1_{N_x,1})$$

$$y^x(2_x)x^y(2_y)y^x(2_z)x^y(2_{N_x,2})\ldots y^x(2_{N_x,i_x})x^y(2_{N_x,2})$$

$$\vdots$$

$$y^x(C_x)x^y(C_y)y^x(C_z)x^y(C_{N_x,1})\ldots y^x(C_{N_x,i_x})x^y(C_{N_x,1})$$

$$= y^x(1_x)x^y(1_y)y^x(2_x)x^y(2_y)\ldots y^x(Sum_N)\times x^y(Sum_N).$$

(7)

Here, $Sum_{N_x} = \sum_{i_x=1}^{C} N_x(i_x)$.

Similarly, $ps^s_{*fp^p}$ is defined by putting together $ps^s_{*fp^p}$ as follows:

**Figure 1. Overview of the $p_s^{(*)}$-code.** Three-dimensional Schlegel diagrams (a projection from four- to three-dimensional space) are used to illustrate the polyhedron. Note that the interior of the polyhedron $abcd$ on the polyhedron in four-dimensional space (not shown) is mapped to the exterior of the outside polyhedron $abcd$ on the Schlegel diagram.
\[
\begin{align*}
ps_2^* &= ps_2(1)ps_2(2)\cdots ps_2(C) \\
&= p_2(1_1)p_2(1_2)\cdots p_2(1_{F(1)}) \\
&\quad \vdots \\
&= p_2(C_1)p_2(C_2)\cdots p_2(C_{F(C)}) \\
&= p_2(1)p_2(2)\cdots p_2(2R) \\
\end{align*}
\]

(8)

In the last transformation, we translated the symbols \( i \) into the corresponding numbers. \( R \) is the number of ridges of the polychoron.

By putting \( ps_2^* \), \( sp^e \), and \( fp \) together, \( p^*_1 = ps_2^*sp^e.fp \). For example, for a polychoron \( A \) shown in Fig. 1, \( p^*_1[A] = ps_2^*[A] = 33334434344334443343333333.

We can recover \( p_1 \) from \( p^*_1 \) as follows. By construction, the first \( F(1) \) digits of \( ps_2^* \) form \( ps_2(1) \). However, we do not know \( F(1) \) beforehand. To find out \( F(1) \), we regard \( ps_2^*sp^e \) as \( p_2 \), and decode it until a polyhedron is completed. Suppose that a polyhedron is completed when the 0th digit of \( ps_2^* \) is decoded. Then \( F(1) = \alpha \), and \( ps_2(1) = p_1(1)p_2(2)p_2(3)\cdots p_2(\alpha) \). If the sides of the polyhedron are found in \( sp^e \), record them in \( sp^e(1) \). We then remove \( ps_2(1) \) from \( ps_2^* \), and obtain \( ps_2^{e(-1)} = p_2(\alpha + 1)p_2(\alpha + 2)p_2(\alpha + 3)\cdots p_2(2R) \). As with \( ps_2^* \), the first \( F(2) \) digits of \( ps_2^{e(-1)} \) form \( ps_2(2) \). To find out \( F(2) \), we decode \( ps_2^{e(-1)}sp^e \) using the \( p_2 \)-code. Suppose that a polyhedron is completed when the \( \beta \)th digit of \( ps_2^{e(-1)} \) is decoded. Then \( F(2) = \beta \), and \( ps_2(2) = p_1(1)p_2(\alpha + 2)p_2(\alpha + 3)\cdots p_2(\alpha + \beta) \). If the sides of the polyhedron are found in \( sp^e(2) \), record them in \( sp^e(2) \). We then remove \( ps_2(2) \) from \( ps_2^{e(-1)} \), and obtain \( ps_2^{e(-2)} = p_2(\alpha + \beta + 1)p_2(\alpha + \beta + 2)p_2(\alpha + \beta + 3)\cdots p_2(2R) \). By repeating this procedure, we can determine \( p_2^e(1) \), \( p_2^e(2) \), \( p_2^e(3) \), \ldots, and therefore \( p_2 \). As an example, we illustrate the procedures of recovering \( p_1[A] \) from \( p^*_1[A] \) in Supplementary Note.

**Reveal and remove redundancy in \( p^*_1 \).** To reveal the redundancy in \( p^*_1 \), we observe how the polychoron \( A \) shown in Fig. 1 is recovered from \( p^*_1[A] = ps_2^*[A] \). After determining \( p_1[A] \), we construct the polyhedron 1 (3333) and polyhedron 2 (34443) and then glue them together in such a way that the face 2 is glued to the face 1. Therefore, \( p_1(2) \) must be equal to \( p_1(1) \). Thus, \( p_1(2) \) is redundant. Next we attach the polyhedron 3 (34443) to the partial polyhedron in such a way that the face 3 is glued to the face 1 of the partial polyhedron. Therefore, \( p_1(3) \) must be equal to \( p_1(1) \), and \( p_1(3) \) is redundant. In addition, to rectify an i-peak, we glue together the faces 3 and 2. Therefore, \( p_1(3) \) must be equal to \( p_1(2) \), and \( p_1(3) \) is redundant.

In general, when two faces \( i_1 \) and \( m_i (m > i) \) are glued together, \( p_1(m_i) \) is redundant, while \( p_1(i) \) is essential. Since the face \( m_i \) meets the face \( i \) along the ridge with the tentative ID \( i, p_1(m_i) = p_1(i) \). Here, \( r(x) \) is the number of peaks of the ridge with the tentative ID \( x \). Thus, the number of peaks of every ridge is doubly recorded in \( ps_2^* \).

Returning to the polyhedron \( A \), we will remove all the redundant \( p_1(m_i) \)'s from \( ps_2^*[A] \) and construct the sequence of essential \( p_1(i) \)'s,

\[
\begin{align*}
p_1(1)p_2(1_2)p_2(1_3)p_2(1_4)p_2(2_2)p_2(2_3)p_2(2_4)p_2(3_2)p_2(3_3)p_2(3_4)p_2(4_2)p_2(4_3)p_2(5_2) \\
&= 3333443443433333.
\end{align*}
\]

(9)

The sequence of essential \( p_1(i) \) is identical with the sequence of \( r(i) \),

\[
\begin{align*}
p_1(1)p_2(1_2)p_2(1_3)p_2(1_4)p_2(2_2)p_2(2_3)p_2(2_4)p_2(3_2)p_2(3_3)p_2(3_4)p_2(4_2)p_2(4_3)p_2(5_2) \\
&= r(1_1)r(1_2)r(1_3)r(1_4)r(1_5)r(2_1)r(2_2)r(2_3)r(2_4)r(2_5) \\
&= r(3_1)r(3_2)r(3_3)r(3_4)r(3_5). \\
\end{align*}
\]

(10)

By rewriting the sequence using the ridge IDs (not tentative ridge IDs), we obtain what we call the ridge-sequence codeword (rs),

\[
\begin{align*}
r(1_1)r(1_2)r(1_3)r(1_4)r(1_5)r(2_1)r(2_2)r(2_3)r(2_4)r(2_5)r(3_1)r(3_2)r(3_3)r(3_4)r(3_5) \\
r(4_1)r(4_2)r(4_3)r(4_4)r(4_5) \\
r(5_1)r(5_2)r(5_3)r(5_4)r(5_5) \\
(10)r(1_1)r(1_2)r(1_3)r(1_4)r(1_5) \\
(12)r(2_1)r(2_2)r(2_3)r(2_4)r(2_5) \\
(13)r(3_1)r(3_2)r(3_3)r(3_4)r(3_5) \\
(14)r(4_1)r(4_2)r(4_3)r(4_4)r(4_5) \\
(10)r(1_1)r(1_2)r(1_3)r(1_4)r(1_5) \\
(12)r(2_1)r(2_2)r(2_3)r(2_4)r(2_5) \\
(13)r(3_1)r(3_2)r(3_3)r(3_4)r(3_5) \\
(14)r(4_1)r(4_2)r(4_3)r(4_4)r(4_5) \\
= rs[A].
\end{align*}
\]

(11)

Here, \( r(i) \) is the number of peaks of the ridge with the ID \( i \). The number of peaks of every ridge is recorded just once in \( rs \), and the redundancy is thus reduced. In general, the redundancy can be reduced by modifying \( p_1 \) into \( p^{(rs)*}_1 = rs;sp^e;fp \). Here, \( rs = r(1_1)r(1_2)r(1_3)r(1_4)r(1_5) \).

**How to recover \( p_1 \) from \( p^{(rs)*}_1 \).** The \( p^{(rs)*}_1 \)-codeword contains information about how to assemble ridges to form a polyhedron in the sense that \( p_1 \) can be recovered from \( p^{(rs)*}_1 \). As is summarized in Fig. 2, to recover \( p_1 \), we determine \( p_1(1), p_1(2), p_1(3), \ldots, p_1(F(A)) \) step-by-step. To determine \( p_1(i) \), we deduce \( p_1(i) \), \( p_1(1) \), \( p_1(2) \), \( p_1(3) \), \( \ldots, p_1(F(A)) \) step-by-step. To deduce \( p_1(i) \), we examine whether the face \( i \) should be glued to an existing face of the partial polyhedron or create a new ridge.
We first describe how to determine $p_i(1) = ps_i(1); sp_i(1)$ from $p_4^{(n)}$. All faces of polyhedron 1 create new ridges. By construction, the first $F(1)$ digits of $r(1)r(2)r(3) \ldots r(R)$ form $ps_i(1)$. However, we do not know $F(1)$ beforehand. To find out $F(1)$, we regard $r(1)r(2)r(3) \ldots r(R); sp_i^{(n)}$ as $p_i$, and decode it until a polyhedron is completed. The polyhedron thus obtained is the polyhedron 1. Suppose that a polyhedron is completed when the $\alpha$th face is decoded. Then $F(1) = \alpha$, and $ps_i(1) = r(1)r(2)r(3) \ldots r(\alpha)$. Every time we recover a polygon in the decoding process, we search $sp_i^{(n)}$ for side-na-pairs of the polyhedron 1. If side-na-pairs are found, we record their corresponding local side IDs in $sp_i(1)$. By combining $ps_i(1)$ and $sp_i(1)$, we obtain $p_i(1) = ps_i(1); sp_i(1)$.

For $2 \leq i$, $p_i(1)$ can be determined from $p_i^{(n)}$ and $p_i(1); p_i(2); \ldots; p_i(i-1)$. Our first task is to deduce $p_i(1)$. For this purpose, we construct a partial polyhedron $D_i(i-1)$, which is obtained by decoding $p_i(1); p_i(2); \ldots; p_i(i-1); fp_i$. Since the face 1 of the polyhedron $i$ should be glued to the s-face of $D_i(i-1)$, $p_i(1) = ps_i(D_i(i-1); i-1)$. Here, $ID_{c-face}(k)$ is the global face ID of the s-face of $D_i(k)$.

For $2 \leq j$, $p_i(1)$ can be determined from $p_i^{(n)}$, $p_i(1); p_i(2); \ldots; p_i(i-1)$, and $p_i(1); p_i(j); p_i(j+1); \ldots; p_i(i-1)$. To deduce $p_i(1)$, we examine whether the face $j$ should be glued to an existing face or create a new ridge. We first search the $w$th part of $fp_i$ for $i$. If it is found, the faces $j = (i, w)$ and $v$ form a face-na-pair. Since those faces should be glued together, $p_i(1) = p_i(v)$. If $i$ is not found, we construct a partial polyhedron $D_i(i-1)$, which is obtained by decoding $p_i(1); p_i(2); \ldots; p_i(i-1); sp_i^{(n)}$ using the $p_i$-code. Then we glue the face $j$ of $(D_i(j))$ to the s-face of $D_i(i-1)$ in such a way that the edge $i$ is glued to the s-face ID of the s-face. If $w(n)$ is the face ID of $D_i(i-1)$, then glue the face $w(n)$ to the face $v(n)$. Thus, $p_i(1)$ is the global face ID of the s-face. $p_i(1)$ can be determined from $p_i^{(n)}$ for $1 \leq i \leq n$. If $p_i(1)$ is the number of ridges contributed by two faces.

To form the $pp$-codeword $p_4^{(n)}$, we introduce the notion of parts. We regard the sides of a polygon are parts of that polygon. We also regard the polygon itself is the part of that polygon. We define the set of parts of the polygon $i$ as

$$S[polygon i] = \{Polygon i, side i_1, \ldots, side i_p(1)\}$$

Similarly, for $j > 2$, we define the set of parts of the polyhedron $j$ as

$$S[polyhedron j] = \{polyhedron j, S[polygon j_1], \ldots, S[polygon j_p(1)], edge j_1, \ldots, edge j_E(1)\}. \quad (13)$$

We assign IDs to parts such that we can identify side-na-pairs $xy$ and face-na-pairs $wv$ in recovering the original polyhedron. To meet this requirement, we assign IDs to parts of the polyhedron in the order of $S[polygon 1]$, polyhedron 1, $S[polygon 2], \ldots, S[polygon F(1)]$, edges of polyhedron 1, polyhedron 1, $S[polyhedron 2], \ldots, S[polyhedron C_i]$, ridges of polyhedron 1, peaks of polyhedron 1.

The $pp$-codeword is obtained as follows. We first translate $sp_i^{(n)}$ $fp_i$ into part ID. Then we remove the separator “;”. Finally, we obtain

$$p_i^{(n)} = rs; pp. \quad (14)$$

The side- and face-na-pairs can be identified from $pp$ as follows. Let $p(i)$ be the ith digit of $pp$. If the part $p(i)$ is a side $Y$, the part $p(i+1)$ is a side $X$, and $Y > X$, then the pair $p(i); p(i+1)$ is a side-na-pair. If the part $p(i)$ is a face $W$ and the part $p(i+2)$ is a face $V$, then the pair $p(i); p(i+1); p(i+2)$ is a face-na-pair.

Note that the amount of tasks needed to generate $p_i^{(n)}$ is comparable to that needed to generate $p_i$. This is because converting $p_i$ to $p_i^{(n)}$ amounts to just assigning IDs to ridges and parts of the polyhedron. We also note that the length of $p_i^{(n)}$ is shorter than that of $p_i$, by $R'$, where $R'$ is the number of ridges contributed by two faces. $R' = R$ for a polyhedron, while $R' < R$ for a partial polyhedron. Therefore, the compression efficiency gets worse for partial polyhedron. As described above, by converting $3333-34443-34443-34443-34333$ into $3333444344434433$, $data$ of the polyhedron is compressed to half. On the other hand, for example, $p_3$ and $p_3^{(n)}$ of a
partial polychoron composed of one 3333-polyhedron and one 34443-polyhedron are 3333 and 34443, respectively. Just one number "3" is removed by converting $p_4$ into $p_{rs}^4$.

The $p_4$-codeword can be recovered from $p_{rs}^4$ by modifying the procedure for recovering $p_4$ from $p_{rs}^4$ as follows (Fig. 3):

1. Determine $p_4(1) = p_{rs}2(1); sp(1)$ as follows:
   (a) Decode $r_1 r_2 r_3 \ldots r(R); pp$ using the the $p_4$-code.
   (b) If a polyhedron is completed when the $\alpha$th face is decoded, then $p_{rs}2(1) = r_1 r_2 r_3 \ldots r(\alpha)$.
   (c) If side-na-pairs of the polyhedron are found in $pp$, record their corresponding local side IDs in $sp(1)$.
2. Determine the next \( p_i = p_i(i) \cdot s(i)(2 \leq i) \) as follows:

(a) \( p_i(i) = p_i(ID_{c-dim}(i - 1)) \).

(b) To determine the next \( p_i(i) (2 \leq j) \), we search \( pp \) for the part ID of the face \( i \). Here, two cases arise:

(I) If the part \( p(k) \) is the face \( i \) and the part \( p(k + 2) \) is a face, then let \( m_n \) be the face-ID of the part \( p(k + 2) \) and \( p_i(i) = p_i(m_n) \).

(II) Otherwise, we examine the k-peak, and then additional two cases arise:

(i) If the k-peak is contributed by three polyhedra, then \( p_i(i) = p_i(ID_{c-dim}(j, i)) \).

(ii) Otherwise, \( p_i(i) = r(N_{ridge}(j, i) + 1) \).

(c) Decode \( p_i(i)p_i(i_2)p_i(i_3) \ldots p_i(i)pp \) using the the \( p_c \)-code. Two cases then arise:

(i) If a polyhedron is completed, then \( p_s = p_s(i)p_s(i_2)p_s(i_3) \ldots p_s(i) \). If side-na-pairs of the polyhedron are found in \( pp \), record their corresponding local side IDs in \( sp \). Thus, \( p_4 = p_s(fp) \) is determined.

(ii) Otherwise, repeat the procedure 2b.

3. Decode \( p_i(1)p_i(2)p_i(3) \ldots p_i(i)pp \) using the the \( p_c \)-code. Two cases then arise:

(a) If a polyhedron is completed, then \( p_s = p_s(i)p_s(2)p_s(3) \ldots p_s(i) \). If face-na-pairs are found in \( pp \), record their corresponding global face, side, and face IDs in \( fp \). Thus, \( p_4 = p_s(fp) \) is determined.

(b) Otherwise, repeat the procedure 2.

As an example, we illustrate how to recover \( p_i[A] \) from \( p_i^{(rs)}[A] = rs[A] = 3333444344343333 \) as follows:

1. We decode \( rs[A] \) using the \( p_c \)-code. When the 4th digit is decoded, a 3333-polyhedron is obtained, thereby it turns out \( p_i(1) = 3333 \) (Fig. 4).

2. We determine \( p_i(2) \) as follows:

(a) The 3333-polyhedron is the partial polyhedron \( D_4(1) \). The s-face of \( D_4(1) \) is the face 11 (Fig. 4). Since the face 1 of the polyhedron 2 will be glued to the face 11; \( p_i(2) = p_i(1) = 3 \).

(b) We construct the partial polyhedron \( D_4(2) \), glue it to the partial polyhedron \( D_4(1) \), and obtain the partial polyhedron \( D_4(2) \) & \( D_4(1) \) (Fig. 5). Since the k-peak \( ab \) is contributed by two polyhedra (polyhedra 1 and \( D_4(2) \)), the face 2; will create a new ridge. Since \( D_4(2) \) & \( D_4(1) \) has four ridges \( abc, bad, cdb, \) and \( acd \), \( N_{ridge}(2) = 4 \). Thus, \( p_i(2) = r(N_{ridge}(2) + 1) = r(4) = 4 \).

(c) For the same reason, \( p_i(4) = r(6) = 4 \), \( p_i(6) = r(7) = 4 \), and \( p_i(2) = r(8) = 3 \).

(d) When we decode \( p_i(2)p_i(4)p_i(5)p_i(6) \) \( p_i(2)p_i(3) = 444433 \), a polyhedron is completed, thereby it turns out \( p_i(2) = 344433 \).

3. We determine \( p_i(3) \) as follows:

(a) We construct \( D_3(2) \) from the partial \( p_c \)-codeword: \( p_i(1)p_i(2) = 3333444344433333 \) (Fig. 6). The s-face of \( D_3(2) \) is the face 1, Therefore, \( p_i(3) = p_i(1) = 3 \).

(b) We construct \( D_3(3) \) & \( D_3(2) \) by gluing \( D_3(3) \) to \( D_3(2) \) (Fig. 7). Since the k-peak is contributed by three polyhedra (polyhedra 1 and 2, and \( D_3(3) \)), the face 3; should be glued to the face ID_{c-dim}(3) (face abf). Thus, \( p_i(3) = p_i(ID_{c-dim}(3)) = p_i(2) = 4 \).

(c) We construct \( D_3(3) \) from the partial \( p_c \)-codeword 34, glue it to \( D_3(2) \), and obtain \( D_3(3) \) & \( D_3(2) \) (Fig. 8). Since the k-peak is contributed by two polyhedra (polyhedra 1 and \( D_3(3) \)), the face 3; will create a new ridge. Thus, \( p_i(3) = r(N_{ridge}(3) + 1) = r(9) = 4 \).

(d) For the same reason, \( p_i(3) = r(10) = 4 \), and \( p_i(3) = r(11) = 3 \).

(e) When we decode \( p_i(3)p_i(2)p_i(3)p_i(2)p_i(3) = 3444334444333333 \), a polyhedron is completed, thereby it turns out \( p_i(3) = 3444334444333333 \).

4. In a similar way, \( p_i(4) \) is determined to be \( p_i(1)p_i(2)r(12)p_i(3)r(13) = 344433p_i(5) = p_i(1)p_i(2)p_i(3)p_i(2)p_i(4)r(14) = 344433p_i(6) = p_i(2)p_i(3)p_i(5)p_i(6) = 3444333333 \).

5. When we decode \( p_i(3)p_i(2)p_i(3)p_i(3)p_i(2)p_i(3)p_i(6) \), a polyhedron is completed, thereby it turns out \( p_i[A] = 3333444344433333333333333333 \).

**Generalization to higher dimensional polytopes.** The \( p_c \)-code can be generalized to the \( p_n \)-code for \( n \)-polytopes (see Supplementary Note and Supplementary Table S1). The \( p_n \)-code instructs how to construct the \( n \)-polytope from its building block \((n-1)\)-polytopes. However, as in the case of \( p_n \), \( p_n \) is redundant. By reducing the redundancy, we can obtain \( p_n^{(rs)} = f_{s2}^{(s)} \). Here, \( p_n^{(rs)} \) is the \( n \)-dimensional generalization of \( p_n^{(rs)} \). The superscript ”\( f_{s2} \)” indicates the 2-face-sequence codeword. The i-face is the \( i \)-dimensional face of an \( n \)-polytope. For example, a 2-face of a polyhedron is a ridge, and a 1-face of a polyhedron is a peak.

As an example, we explain \( p_n \) for \( n \)-dimensional cubes (\( n \)-cubes), and then demonstrate how the \( p_n \) are converted into their corresponding \( p_n^{(rs)} \)’s. The 3-cube is an ordinary cube, and \( p_3 \{3\text{-cube}\} = 44444444 = 4^n \). The 4-cube
is composed of eight 3-cubes, and \( p_4(4\text{-cube}) = 4^8 \). The 5-cube is composed of ten 4-cubes, and \( p_5(5\text{-cube}) = p_4(4\text{-cube})^5 \). In general, an \( n\text{-cube} \) consists of \( 2^n (n - 1) \) - cubes\(^{20} \), and \( p_n[n\text{-cube}] = p_{n-1}[(n - 1) - \text{cube}]^2 \) (for \( n \geq 3 \)). The number of 1-faces of each 2-face of an \( n\text{-polytope} \) is \( (n - 2)! \) times recorded in \( p_{n+1} \), so that reducing the redundancy has a greater impact for higher dimensional polytopes. The redundancy can be reduced by using \( f_{22} \) (see Supplementary Note and Supplementary Figure S1), which is denoted as

\[
f_{22} = f_2(1)f_2(2)f_2(3)\cdots f_2(N_2).
\]  

Figure 3. Procedures for recovering \( p_4 \) from \( p_4^{(rs)} = rspp \). The differences from the algorithm for \( p_4^{(rs)} \) is highlighted in yellow.
Here, \( f_2(i) \) is the number of 1-faces of the 2-face \( i \). \( N_2 \) is the number of 2-faces of the \( n \)-polytope. For example, \( p_{\text{cub}}(n) \) can be recovered from \( n \frac{n(n-1)}{2} \). Here, \( N_2(n \text{-cube}) = n(n-1)2^{n-2} \).

Moreover, we can rewrite \( p_{\text{cub}}(n) \) as \( p \). In other words, we unify \( p_{\text{cub}}(1), p_{\text{cub}}(2), p_{\text{cub}}(3), \ldots \) into \( p \). Although the subscript "\( n \)" is removed, the dimension \( n \) of the polytope can be determined as a result of decoding \( p \). We stress that polytopes of different dimensions can be represented by codewords of the same format, namely two number sequences separated by ",".

**Discussion**

E. A. Lazar, et al. introduced the Weinberg code to describe single Voronoi polyhedra\(^4\). But the Weinberg code does not allow for describing complexes of Voronoi polyhedra. On the other hand, our \( p \)-code allows us to describe complexes of Voronoi polyhedra, which would reveal the longer-range order of amorphous materials that cannot be seen from single Voronoi polyhedra. Our methods can be used to study a wide range of systems which are represented by polytopal tilings such as atoms in materials, grains in crystals, foams, galaxies in the universe, hyperspheres in higher-dimensional spaces, etc.\(^1-12,21\).

**Conclusion**

We have developed a unified theory for representing polyhedral tilings and polytopes of different dimensions by brief codewords. Specifically, we have first formulated a method to deduce how to assemble ridges to form a polyhedral tiling or a polychoron from \( rs = r(1)r(2)r(3)\ldots r(R) \). This has been achieved by reducing the redundancy in \( p_r \). Many polychora can be constructed just from \( rs \), but there are some polychora that need \( pp \) which contains additional information about how to assemble ridges. It is remarkable that a mere sequence of \( r(i) \)'s contains all or almost all information about how to assemble \( r(i) \)-gonal ridges to form a polychoron. Since a polychoron can be constructed from \( p_r^{(n)} = rs; pp \), the polychoron can be represented by \( p_r^{(n)} \). The local tiling structure composed of...
a central polyhedron and polyhedra surrounding the central polyhedron can also be represented by $p_{rs}^4$ for it can be regarded as a part of a polychoron. Therefore, a polyhedral tiling can be characterized by distribution of $p_{rs}^4$ of different central polyhedra. The idea of assembling two-dimensional components has been generalized to higher dimensional polytopes. Using the present method, $p_n$ of an $n$-polytope can be converted into $p$ whose length is as long as $1/(n-2)!$ times of that of $p_n$. Therefore, the impact of the present method factorially increases
as the dimension of a polytope increases. The amount of tasks needed to convert $p_n$ to $p$ is negligible compared to that needed to generate $p_n$. We stress that no subscript "n" that indicates the dimension of a polytope is attached to $p$. The dimension of the polytope is determined as a result of decoding $p$. In other words, the $p_3$-code, $p_4$-code, $p_5$-code, ..., and $p_n$-code have been unified into the $p$-code. Since shorter codewords are easier to handle for both humans and computers, our unified theory of polytopes would be a powerful tool to study a wide range of structures such as atoms in materials, grains in crystals, foams, galaxies in the universe, hyperspheres in higher-dimensional spaces, etc1–12,21.
References
1. Bernal, J. D. The Bakerian Lecture, 1962. The Structure of Liquids. Proc. R. Soc. Lond. A 280, 299–322 (1964).
2. Finney, J. Random Packings and Structure of Simple Liquids. 1. Geometry of Random Close Packing. Proc. R. Soc. Lond. A 319, 479–493 (1970).
3. Finney, J. Random Packings and Structure of Simple Liquids. 2. Molecular Geometry. Proc. R. Soc. Lond. A 319, 495–507 (1970).
4. Yonezawa, F. Glass Transition and Relaxation of Disordered Structures. Solid State Phys. 45, 179–254 (1991).
5. Sheng, H. W., Luo, W. K., Alamgir, F. M., Bai, J. M. & Ma, E. Atomic packing and short-to-medium-range order in metallic glasses. Nature 439, 419–425 (2006).
6. Hirata, A. et al. Geometric Frustration of Icosahedron in Metallic Glasses. Science 341, 376–379 (2013).
7. Nishio, K., Kôga, J., Yamaguchi, T. & Yonezawa, F. Confinement-Induced Stable Amorphous Solid of Lennard–Jones Argon. J. Phys. Soc. Jpn. 73, 627–633 (2004).
8. Nishio, K., Miyazaki, T. & Nakamura, H. Universal Medium-Range Order of Amorphous Metal Oxides. Phys. Rev. Lett. 111, 155502 (2013).
9. Lazar, E. A., Han, J. & Srolovitz, D. J. Topological framework for local structure analysis in condensed matter. PNAS 112, E5769–E5776 (2015).
10. Kraynik, A. M., Reinelt, D. A. & van Swol, F. Structure of random monodisperse foam. Phys. Rev. E 67, 031403 (2003).
11. Lazar, E. A., Mason, J. K., MacPherson, R. D. & Srolovitz, D. J. Complete Topology of Cells, Grains, and Bubbles in Three-Dimensional Microstructures. Phys. Rev. Lett. 109, 095505 (2012).
12. Ramella, M., Boschin, W., Fadda, D. & Nonino, M. Finding galaxy clusters using Voronoi tessellations. Astron. Astrophys. 368, 11 (2001).
13. Nishio, K. & Miyazaki, T. How to describe disordered structures. Scientific Reports 6, 23455 (2016).
14. Cromwell, P. R. Polyhedra, Ch. 5, 181–218 (Cambridge University Press, 1999).
15. Wilson, R. & Stewart, I. Four Colors Suffice: How the Map Problem Was Solved, Ch. 3 and 4, 28–54 (Princeton University Press, 2013).
16. Wilson, R. J. Introduction to Graph Theory, Ch. 1, 8–31 (Pearson, 2012).
17. Coxeter, H. S. M. Introduction to Geometry, Ch. 10, 148–159; Ch. 22, 396–414 (Wiley, 1989).
18. Ziegler, G. M. Lectures on Polytopes, Ch. 5, 127–148 (Springer, 2012).
19. Richeson, D. S. Euler’s Gem: The Polyhedron Formula and the Birth of Topology, Ch. 16, 156–172 (Princeton University Press, 2012).
20. McMullen, C. The Visual Guide To Extra Dimensions: The Physics Of The Fourth Dimension, Compactification, And Current And Upcoming Experiments, Ch. 3, 65–98 (Create Space Independent Publishing Platform, 2009).
21. Skoge, M., Donev, A., Stillinger, F. H. & Torquato, S. Packing hyperspheres in high-dimensional Euclidean spaces. Phys. Rev. E 74, 041127 (2006).

Acknowledgements
K.N. wishes to thank Motoko Kotani for fruitful discussions on toroidal polyhedra.

Author Contributions
K.N. conceived the original idea of code for polytopes. K.N. and T.M. polished up the idea.

Additional Information
Supplementary information accompanies this paper at http://www.nature.com/srep

Competing financial interests: The authors declare no competing financial interests.

How to cite this article: Nishio, K. and Miyazaki, T. Describing polyhedral tilings and higher dimensional polytopes by sequence of their two-dimensional components. Sci. Rep. 7, 40269; doi: 10.1038/srep40269 (2017).

Publisher’s note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This work is licensed under a Creative Commons Attribution 4.0 International License. The images or other third party material in this article are included in the article’s Creative Commons license, unless indicated otherwise in the credit line; if the material is not included under the Creative Commons license, users will need to obtain permission from the license holder to reproduce the material. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/

© The Author(s) 2017