
FAV File Format Specification

Author : Tomonari Takahashi ¹⁾, Atsushi Masumori ²⁾, Masahiko Fujii ¹⁾, Hiroya Tanaka ²⁾

1) Fuji Xerox Co., Ltd.

2) Social Fabrication Laboratory of Keio University at Shonan-Fujisawa Campus (SFC)

Issued on : July 12, 2016

Version : 1.0

INDEX

INTRODUCTION 3

 THE PURPOSE OF THIS SPECIFICATION DOCUMENT 3

 WORDING IN THIS DOCUMENT 3

 LICENSE 4

1. THE FAV FILE FORMAT 5

 1.1. CONCEPT 5

 1.2. CHARACTERISTICS 5

 1.3. ENVIRONMENTAL CONDITIONS 7

 1.4. STRUCTURE 8

 1.5. NOTATION METHOD 9

 1.6. SECURITY 9

 1.7. <FAV> 9

2. <METADATA> 10

 2.1. <ID> 10

 2.2. <TITLE> 11

 2.3. <AUTHOR> 11

 2.4. <LICENSE> 11

 2.5. <NOTE> 12

3. <PALETTE> 13

 3.1. <GEOMETRY> 14

 3.1.1. <shape> 15

 3.1.2. <scale> 16

 3.2. <MATERIAL> 17

 3.2.1. <material_name> 18

 3.2.2. <product_info> 18

 3.2.3. <iso_standard> 19

4. <VOXEL> 20

 4.1. <GEOMETRY_INFO> 21

 4.2. <MATERIAL_INFO> 22

 4.3. <DISPLAY> 23

 4.4. <APPLICATION_NOTE> 23

5. <OBJECT> 24

 5.1. <GRID> 26

 5.1.1. <origin> 26

 5.1.2. <unit> 27

 5.1.3. <dimension> 27

 5.2. <STRUCTURE> 28

 5.2.1. <voxel_map> 29

 5.2.2. <color_map> 31

 5.2.3. <link_map> 33

6. CONCLUSION 37

Introduction

The purpose of this specification document

This document describes the FAV (fabricatable voxel) file format specification. This document is made publicly available so that:

- the FAV format may be used to achieve efficient management of the complex structures of 3D model data such as shapes and attributes (including internal structures)
- FAV format 3D model data may be utilized in hardware and software
- FAV files may be created, edited, and utilized within systems (interoperability)

3D model data formatted in accordance with this specification can be used on systems that support FAV files. By processing FAV files on FAV-supporting systems in accordance with this specification, it is possible to utilize a range of information from 3D model data stored in this format.

We hope that use of the FAV format will enhance the range of design and expression of 3D model data and contribute to further developments in the fabrication of and communication using 3D models.

Wording in this document

● RFC2119

This document uses keywords and phrases as described in RFC 2119. Below is an excerpt from RFC 2119.

1. **MUST**

This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

2. **MUST NOT**

This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.

3. **SHOULD**

This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

4. **SHOULD NOT**

This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implication should be understood and the case carefully weighed before implementing any behavior described with this label.

5. **MAY**

This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option (except, of course, for the feature the option provides.)

In this specification document, those specifications that contain the keywords and phrases defined above (1 to 5) are the requirements deemed necessary in fulfilling the purpose of the FAV format and achieving the intended operation of systems that handle FAV files. It must be fully understood that serious problems may occur in systems handling FAV files when FAV files are used without satisfying the specified REQUIREMENTS.

The use of FAV files that do not conform to this specification REQUIRES that a system be equipped with an appropriate error handling mechanism; however, it cannot be guaranteed that any one system will have such a mechanism. Therefore, it is necessary for the users of FAV format files to take utmost care that no unintended operations occur on a system handling FAV files.

● XML

FAV files are written in XML.

In this specification document, elements in XML are enclosed in angle brackets. E.g., <element>
 The attributes of elements are enclosed in parentheses. E.g., (attribute)

In this document, XML code examples are enclosed in a box and begin with "e.g.,".

```
e.g., <element_label attribute_label=0.1>ELEMENT</element_label>
```

In this document, elements and attributes are detailed in tables as shown below.

Table 1: An example description of the specifications of an element and its attribute

Element	Attribute	Data type	Data [Default value]	Description	Condition
element_label	attribute_label	Double	A double value [0.0]	This area describes what is specified as the attribute of an element and the meaning of the attribute.	This area describes conditions such as whether the attribute is required, the number of attributes required (e.g., one or more), etc.
	-	String	A character string [-]	This area describes what is specified as the data of the element and the meaning of the specified data. "-" is written when there is no default value.	

When elements are written in angle brackets separated by a period, the element on the left side of the period indicates a parent element, and the element on the right side of the period indicates a child element.

```
e.g., <element_parent.element_child> expresses the element structure shown below.

<element_parent>
  <element_child> The data for this element is described here. </element_child>
</element_parent>
```

License

The owners of the copyright and the license of this specification document are as follows:

License information:

```
© 2016, Tomonari Takahashi*, Atsushi Masumori**, Masahiko Fujii*, and Hiroya Tanaka**,
*Fuji Xerox Co., Ltd. ,
**Keio University SFC Social Fabrication Laboratory.

Creative commons Attribution-NonCommercial-NoDerivatives 4.0 International
<https://creativecommons.org/licenses/by-nc-nd/4.0/>
```

This license statement is related to the copying and modification of this document. It is neither intended to restrict nor permit the use of data in FAV format and/or systems that handle FAV files, and does not assert any rights with regard to such data and systems.

The following are to be described in order to fulfill the obligations specified in the license that apply when this document is entirely or partially copied or used in other formats.

- The license information shown above
- The URL of this document
- Notation stating that the material is a copy or an excerpt of FAV File Format Specification Ver 1.0

1. The FAV file format

The FAV format is a file format for saving 3D model data.

1.1. Concept

FAV stands for fabricatable voxel.

In addition to a 3D object's external structure, a FAV file can store information on a range of attributes such as those defining the internal structure, materials to be used, colors, and connection strength of an object. The FAV file format enables designers to model both the exterior and interior of a 3D object right down to the finest requirements and to then save this data.

Using 3D model data in the FAV format furthers the range of expression possible with a fabricated object, thereby enabling us to conceive new ideas and fabricate new objects that have not been possible until now. These enhancements will contribute to the facilitation of better communication using 3D model data, and be especially beneficial in the creation of new applications capable of crossing between industry borders. For example, the FAV format can be expected to have a particularly large influence in the creation of applications capable of integrating the currently separate processes of image processing, management of scanned data in the medical field (MRI / CT), and fabrication of industrial products from composite materials.

3D model data in FAV format is "fabricatable", meaning that data is stored in a way that is optimized for fabrication. Data optimized for fabrication satisfies the following:

- The information required for fabrication (e.g., shape, material, color, connection strength) is clearly defined in three-dimensional space, for both the exterior and interior of an object.
- It allows the user to design (CAD), analyze (CAE), and inspect (CAT) 3D model data seamlessly in an integrated manner without having to convert data.

FAV format 3D model data satisfies all of the above.

1.2. Characteristics

FAV stands for fabricatable voxel. The FAV format expresses 3D data in the form of voxels.

Voxels are the three-dimensional equivalents of pixels. Similar to the way pixels are arranged to create a two-dimensional image, a three-dimensional object is structured by arranging voxels in three-dimensional space.

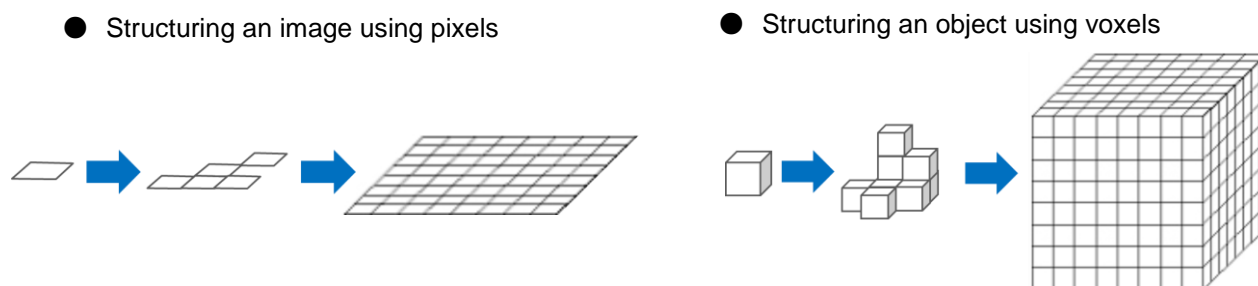


Fig. 1 : Pixels and voxels

Defining three-dimensional objects using voxels makes it possible to design 3D objects of various microstructures. This can be done by using voxels of various designs, not including voxels in certain areas, etc.

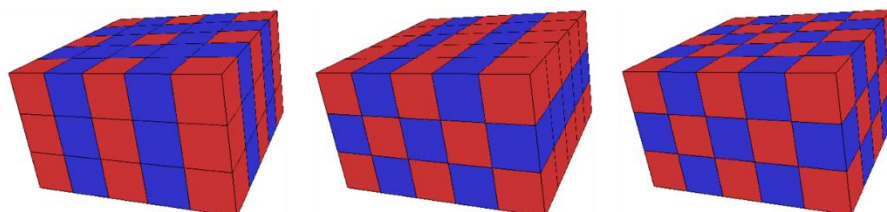
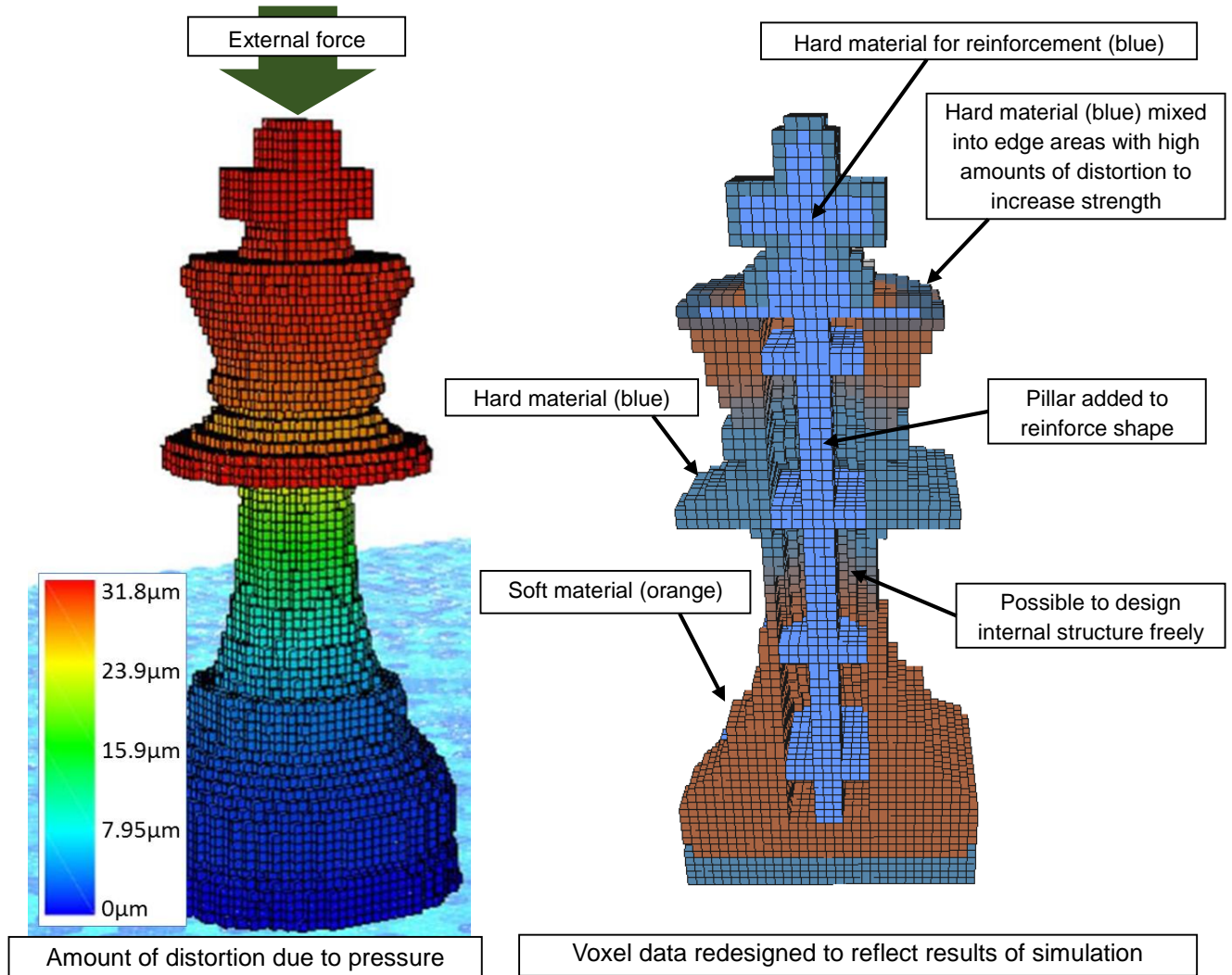


Fig. 2 : Designing intricate microstructures using voxels

The following are the benefits of defining 3D objects by arranging voxels:

- The internal structure as well as the external structure of an object can be defined.
- Various attributes such as materials, colors, and connection strength can be defined together with the overall structure of the object.
- Voxels are finite elements. Therefore, voxel-based data can be used for simulation (CAE) using the same format as that used for the design phase (CAD). Also, it is possible to modify the design of each voxel based on the results of simulations.



Simulation results can be reflected in voxel data for easy design changes

- Material changes (changing to harder materials, etc.)
- Structural changes (reinforcing shape, removing internal areas, etc.)

Fig. 3 : The benefits of using voxel-based 3D model data

1.3. Environmental conditions

The specifications that follow assume that FAV-format data is handled in the environment described below.

● The coordinate system

In the coordinate system for FAV files, planes which form the layers of an object are defined on the x-y axes, while the height of an object is defined on the z-axis as a positive value. The coordinate system is REQUIRED to be right-handed.

Taking into consideration the work areas of fabrication equipment such as 3D printers where 3D models are actually constructed, it is RECOMMENDED that coordinates of 3D model data be plotted so that the z-axis value represents the height, the x-axis value the width, and the y-axis value the depth of a fabricated object when viewed from the origin point of the equipment to be used (which is equal to the origin point of the coordinates).

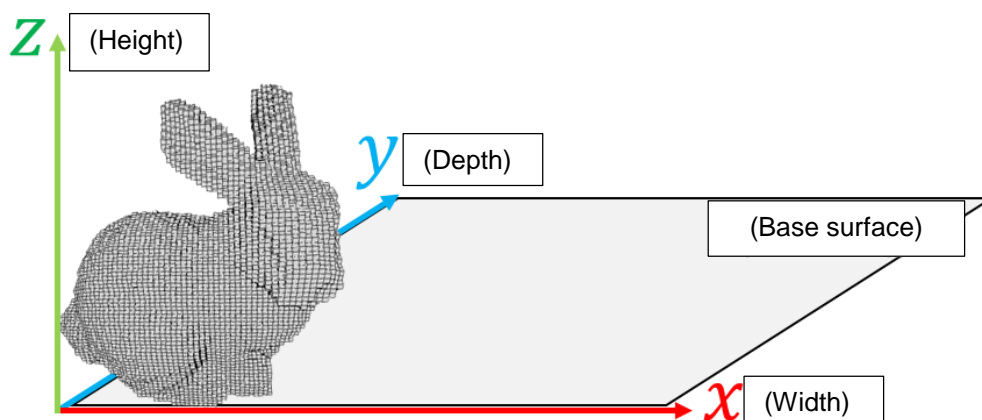


Fig. 4 : The coordinate system used for FAV files

● The units used for the coordinate system

The numerical value 1 used in the coordinate system of the FAV format is REQUIRED to represent 1 mm in a real-world object.

Different units (e.g., inches) may be used depending on the application. If a different unit is used, however, note that a coordinate value of 1 MUST still be equal to 1 mm in a real-world object when data is saved in FAV format. Note also that the same FAV file MUST NOT be interpreted in different units when handled by different environments or applications.

1.4. Structure

The elements of a FAV file form a tree structure as shown below.

<metadata >, <palette >, <voxel >, and <object > are the main four elements of a FAV file. <palette> is referenced by <voxel>, and <voxel> is referenced by <object>. Therefore, it is RECOMMENDED that elements be defined in the following order: <palette >, <voxel >, <object >.

Unless otherwise specified, the order of elements is discretionary.

See each section for details of each element and attribute.

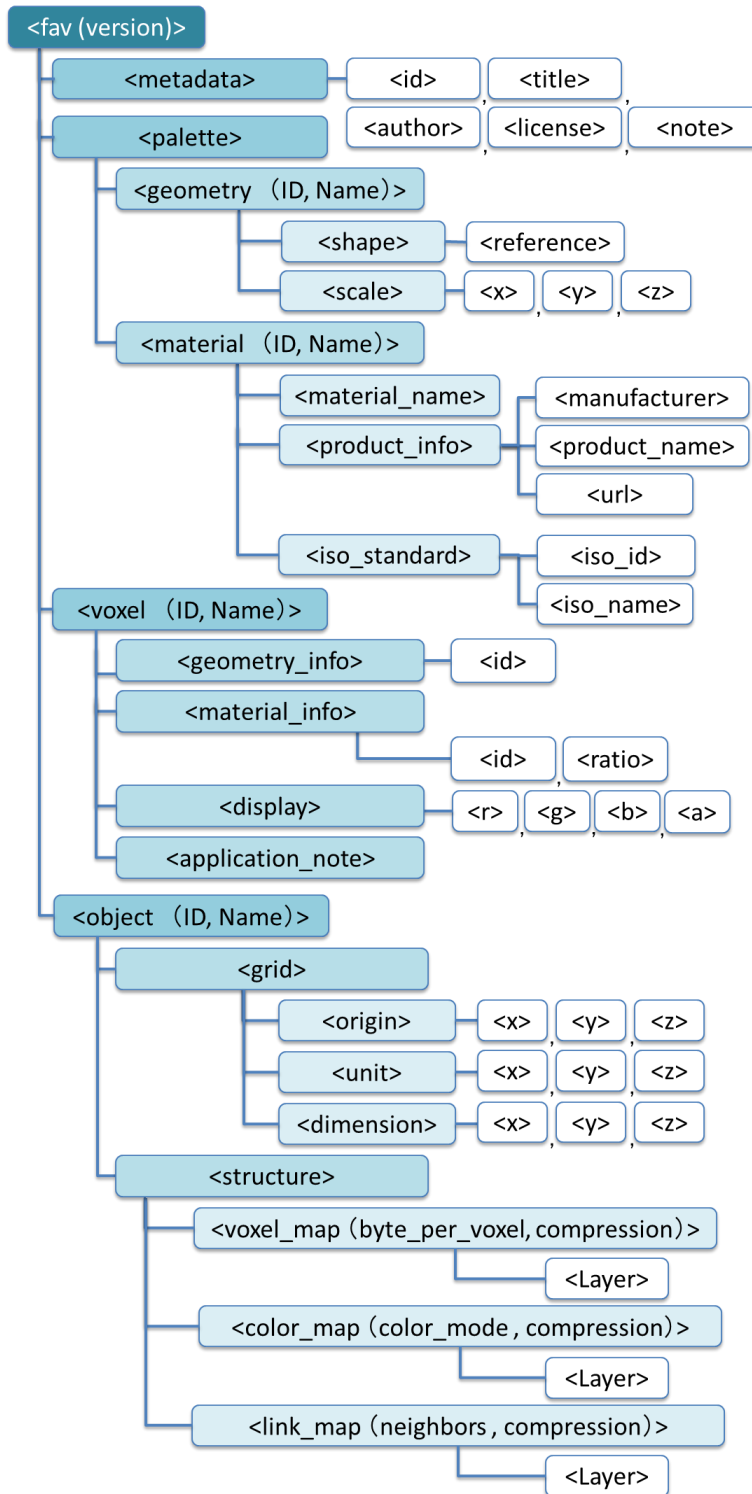


Fig. 5 : The tree structure of elements that constitute a FAV file

1.5. Notation method

The labels of elements and attributes in a FAV file SHALL be in lower case. When a label consists of multiple words, each word SHALL be concatenated by underscore.

e.g., <element_item attribute_item="ATTRIBUTE"/>

1.6. Security

Security specifications for FAV-format data do not greatly differ from other general electronic data. This means that it is possible to apply security measures such as setting passwords, adding digital signatures for detecting falsification, and allowing users to browse FAV files only when they have certain keys.

FAV files that have been modified without the permission of the author or whose author information has been falsified must not be used. It MUST be fully understood that any usage in conflict with the rules stipulated in the license may be considered a violation of the law.

1.7. <fav>

Parent element: — , **Link:** <metadata>, <palette>, <voxel>, <object>

<fav> is the root element of a FAV file. 3D model data defined in the FAV format starts from the <fav > element. All pieces of data defined and managed in the FAV format are stored under <fav>.

<fav> has the following attribute.

Table 2: The description of <fav>

Element	Attribute	Data type	Data [Default value]	Description	Condition
fav	version	String	The version number of the FAV format [—]	Specifies the character string that represents the version number of the FAV format used for the file.	Required

Shown below is an example of XML code under <fav>

e.g.,
 <fav version="1.0">
 ...
 </fav>

Fig. 6 : XML code examples under <fav>

2. <metadata>

Parent element: <fav>, Link: <material>, <object>

<metadata> defines metadata for the various types of data defined in the FAV format.

<fav>, <material>, and <object> can be defined as the parent elements of <metadata>. Regardless of the level at which the <metadata> element is defined, elements that can be defined under <metadata> are always remain the same. It is only necessary to define required elements.

In <metadata>, the following elements may be defined.

- <id>
- <title>
- <author>
- <license>
- <memo>

Shown below is an example of XML code under <metadata>

```
e.g.,
<metadata>
  <id>bc4affb5-9a53-4de7-9f27-721ef27e8f34</id>
  <title><![CDATA[FAV Ver1.0 Sample File]]></title>
  <author><![CDATA[Fuji Xerox & Keio SFC]]></author>
  <license><![CDATA[CC BY]]></license>
  <note><![CDATA[This is a sample file in FAV format ver1.0.]]></note>
</metadata>
```

Fig. 7 : XML code examples under <metadata>

2.1. <id>

Parent element: <metadata>, Link: <fav>, <material>, <object>

<id> specifies the ID used to identify the parent element of <metadata>. It is recommended that IDs that are guaranteed to be unique be used (e.g., GUID) to ensure the uniqueness of each piece of data in the FAV format.

<id> contains the following data.

Table 3: The description of <id>

Element	Attribute	Data type	Data [Default value]	Description	Condition
id	-	String	The ID of the parent element [-]	Specifies the character string of the ID that is used to identify the parent element of <metadata >.	Required

See Fig. 7 for XML code examples for <id>.

2.2. <title>

Parent element: [<metadata>](#), Link: [<fav>](#), [<material>](#), [<object>](#)

[<title>](#) specifies the title of the parent element of [<metadata>](#). For example, [<fav.metadata.title >](#) specifies the title of [<fav>](#), and [<object.metadata.title >](#) specifies the title of a single [<object>](#) element.

[<title>](#) contains the following data.

Table 4: The description of [<title >](#)

Element	Attribute	Data type	Data [Default value]	Description	Condition
title	-	CDATA	The title of the parent element [—]	Specifies the character string of the title of the parent element of <metadata> .	Required

See Fig. 7 for XML code examples for [<title>](#).

2.3. <author>

Parent element: [<metadata>](#), Link: [<fav>](#), [<material>](#), [<object>](#)

[<author>](#) specifies the author of the parent element of [<metadata>](#). For example, [<fav.metadata.author>](#) specifies the author of [<fav>](#), and [<object.metadata.author>](#) specifies the author of a single [<object>](#) element.

[<author>](#) contains the following data.

Table 5: The description of [<author>](#)

Element	Attribute	Data type	Data [Default value]	Description	Condition
Author	-	CDATA	The author of the parent element [—]	Specifies the character string of the author of the parent element of <metadata> .	Required

See Fig. 7 for XML code examples for [<author>](#).

2.4. <license>

Parent element: [<metadata>](#), Link: [<fav>](#), [<material>](#), [<object>](#)

[<license>](#) is the license information of the parent element of [<metadata>](#). For example, [<fav.metadata.license>](#) specifies the license for [<fav>](#), and [<object.metadata.license >](#) specifies the license for a single [<object>](#) element.

Some example licenses are Creative Commons (CC), GNU General Public License (GPL), BSD, and X11 (MIT). If a unique license is to be defined, write the entire text under [<license>](#) and/or provide the link to the license information.

[<license>](#) contains the following data.

Table 6: The description of [<license>](#)

Element	Attribute	Data type	Data [Default value]	Description	Condition
license	-	CDATA	The license information of the parent element [—]	Specifies the character string describing the license information of the parent element of <metadata> .	Required

See Fig. 7 for XML code examples for [<license>](#).

2.5. <note>

Parent element: <metadata>, **Link:** <fav>, <material>, <object>

<note> is a memo for the parent element of <metadata>. For example, <fav.metadata.note> specifies the memo for <fav>, and <object.metadata.note > specifies the memo for a single <object> element.

<note> contains the following data.

Table 7: The description of <note>

Element	Attribute	Data type	Data [Default value]	Description	Condition
note	-	CDATA	The memo for the parent element [–]	Specifies the character string of a memo for the parent element of <metadata>.	

See Fig. 7 for XML code examples for <note>.

3. <palette>

Parent element: <fav>, Link: <voxel>, <geometry>, <material>

In <palette>, basic information such as the shape and material of a voxel is registered in preparation for defining the overall structure of the 3D model data in FAV format. 3D model data is represented in the FAV format by taking <voxel> elements built from the basic information registered in <palette> and using these to define <object>.

The following elements are described at the level under <palette>.

- <geometry>
- <material>

Shown below is an example of XML code under <palette>

e.g.,

```
<palette>
  <geometry id="1" name="Normal Cube">
    <shape>cube</shape>
    <scale>
      <x>1</x>
      <y>1</y>
      <z>1</z>
    </scale>
  </geometry>
  <geometry id="2" name="Plate">
    <shape>cube</shape>
    <scale>
      <x>1</x>
      <y>1</y>
      <z>0.25</z>
    </scale>
  </geometry>
  <geometry id="3" name="Diamond">
    <shape>user_defined</shape>
    <reference><![CDATA[Diamond.stl]]</reference>
    <scale>
      <x>0.98</x>
      <y>0.98</y>
      <z>-1.05</z>
    </scale>
  </geometry>
  <material id="1" name="SoftMat1">
    <material_name><![CDATA[Some-soft-materials]]</material_name>
  </material>
  <material id="2" name="HardMat1">
    <product_info>
      <manufacturer><![CDATA[ABC Materials Co.]]</manufacturer>
      <product_name><![CDATA[ULTRA-HARD/007]]</product_name>
      <url><![CDATA[http://www.abcmaterial.com/ultra/hard/007]]</url>
    </product_info>
    <product_info>
      <manufacturer><![CDATA[ABC Materials Co.]]</manufacturer>
      <product_name><![CDATA[ULTRA-HARD/006a]]</product_name>
      <url><![CDATA[http://www.abcmaterial.com/ultra/hard/006/a]]</url>
    </product_info>
  </material>
</palette>
```

```

    <iso_standard>
      <iso_id>ISO 1043-1:2006</iso_id>
      <iso_name>ABS</iso_name>
    </iso_standard>
  </material>
</palette>

```

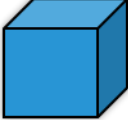





Fig. 8 : XML code examples under <palette>

3.1. <geometry>

Parent element: <palette>, Link: <voxel>, <shape>, <scale>

<geometry> defines the shape and scale of a voxel, the basic element of 3D model data. By arranging voxels that have their geometric properties defined in <geometry>, 3D model data can be structured.

Table 8: Example voxel shapes that can be defined in <geometry>

Image						
(id)	01	02	03	04	05	06
(name)	Cube01	Cube02	Plate	BigSphere	SmallSphere	Cylinder
<shape>	cube	cube	cube	sphere	sphere	user_defined
<scale>	2x2x2	1x1x1	1x1x0.3	1.5x1.5x1.5	0.25x0.25x0.25	3x1x1

<geometry> has the following attributes.

Table 9: The description of <geometry>

Element	Attribute	Data type	Data [Default value]	Description	Condition
geometry	id	positive Integer	A positive integer [1~]	Specifies the unique ID that is used to identify <geometry>. The same ID MUST NOT be used for multiple <geometry> elements.	Required
	name	String	A character string [Object1~]	Specifies the name of <geometry>. The same name SHOULD NOT be used for multiple <geometry> elements.	

The following elements are described at the level under <geometry>

- <shape>
- <scale>

Shown below is an example of XML code under `<geometry>`

```
e.g.,
<geometry id="2" name="Plate">
  <shape>cube</shape>
  <scale>
    <x>1</x>
    <y>1</y>
    <z>0.25</z>
  </scale>
</geometry>
<geometry id="3" name="Diamond">
  <shape>user_defined</shape>
  <reference><![CDATA[Diamond.stl]]</reference>
  <scale>
    <x>0.98</x>
    <y>0.98</y>
    <z>-1.05</z>
  </scale>
</geometry>
```

Fig. 9 : XML code examples under `<geometry>`

3.1.1. `<shape>`

Parent element: `<geometry>`, Link: `<voxel>`, `<grid>`

`<shape>` defines the shape of a voxel, the basic element of 3D model data. In addition to the standard shapes, it is possible to use other shapes by specifying external STL files.

When voxels with `<shape>` specified are arranged in `<grid>`, the center of each shape is positioned in the center of its cell.

`<shape>` contains the following data.

Table 10: The description of `<shape>`

Element	Attribute	Data type	Data [Default value]	Description	Condition
shape	—	String	cube / sphere / user_defined [cube]	Specifies the shape of a voxel. When "user_defined" is specified, <code><reference></code> MUST also be defined.	

When "user_defined" is specified in `<shape>`, it is **REQUIRED** that `<reference>` also be defined. The `<reference>` element specifies an external STL file that defines the shape of a voxel.

Table 11: The description of `<reference>`

Element	Attribute	Data type	Data [Default value]	Description	Condition
reference	—	CDATA	The path of the STL file defining the shape of a voxel [—]	Specifies the character string of the relative path to the external STL file defining the shape of a voxel.	

See Fig. 9 for XML code examples for `<shape>`.

3.1.2. <scale>

Parent element: <geometry>, Link: <voxel>, <shape>, <grid>

<scale> defines the scale of a voxel, the basic element of 3D model data. The scale is specified in relation to the cell size specified under <grid>, the element which defines the space in which voxels are arranged. The standard scale is 1×1×1.

When voxels with <scale> specified are arranged in <grid>, the center of each shape is positioned in the center of its cell.

<scale> contains the following elements

Table 12: The description of <scale>

Element	Attribute	Data type	Data [Default value]	Description	Condition
x	—	Double	A double value [1.0]	Specifies the scale for the x-axis of <shape> in relation to the cell size specified in <grid>. When a negative value is specified, the voxel is flipped. 0 <u>MUST NOT</u> be specified.	
y	—	Double	A double value [1.0]	Specifies the scale for the y-axis of <shape> in relation to the cell size specified in <grid>. When a negative value is specified, the voxel is flipped. 0 <u>MUST NOT</u> be specified.	
z	—	Double	A double value [1.0]	Specifies the scale for the z-axis of <shape> in relation to the cell size specified in <grid>. When a negative value is specified, the voxel is flipped. 0 <u>MUST NOT</u> be specified.	

See Fig. 9 for XML code examples for <scale>.

3.2. <material>

Parent element: <palette>, Link: <voxel>

<material> contains information on the material(s) that constitute each voxel, the basic element of 3D model data.

By arranging voxels for which the material make-up is defined under <material>, the overall material structure of the 3D model data is defined.

<material> has the following attributes.

Table 13: The description of <material>

Element	Attribute	Data type	Data [Default value]	Description	Condition
material	id	positive Integer	A positive integer [1~]	Specifies the unique ID that is used to identify <material>. The same ID MUST NOT be used for multiple <material> elements.	Required
	name	String	A character string [Material01~]	Specifies the name of <material>. The same name SHOULD NOT be used for multiple <material> elements.	

At the level under <material>, one or more of the following elements representing material information are defined.

- <material_name>
- <product_info>
- <iso_standard>

It is possible to define multiple elements under <material>. These elements are checked in the order they were defined to confirm whether the material they specify can be used. Once confirmation is obtained that a specified material can be used, all other material information is ignored.

Shown below is an example of XML code under <material>

e.g.,

```

<material id="2" name="HardMat1">
  <product_info>
    <manufacturer><![CDATA[ABC Materials Co.]]</manufacturer>
    <product_name><![CDATA[ULTRA-HARD/007]]</product_name>
    <url><![CDATA[http://www.abcmaterial.com/ultra/hard/007]]</url>
  </product_info>
  <product_info>
    <manufacturer><![CDATA[ABC Materials Co.]]</manufacturer>
    <product_name><![CDATA[ULTRA-HARD/006a]]</product_name>
    <url><![CDATA[http://www.abcmaterial.com/ultra/hard/006/a]]</url>
  </product_info>
  <iso_standard>
    <iso_id>ISO 1043-1:2006</iso_id>
    <iso_name>ABS</iso_name>
  </iso_standard>
</material>

```

Fig. 10 : XML code examples under <material>

3.2.1. <material_name>

Parent element: <material>; Link: -

<material_name> specifies a material by its name. A character string identifying the material (e.g., product name, common name, general material name such as ABS, PLA) is specified.

<material_name> contains the following data.

Table 14: The description of <material_name>

Element	Attribute	Data type	Data [Default value]	Description	Condition
material_name	-	CDATA	The material name [–]	Specifies a character string identifying the material used.	

See Fig. 10 for XML code examples for <material_name>.

3.2.2. <product_info>

Parent element: <material>; Link: -

<product_info> specifies a material by its product information. Product information identifying the material (e.g., manufacturer's name, product name, product code, product information URL) is specified.

<product_info> contains the following elements.

Table 15: The description of <product_info>

Element	Attribute	Data type	Data [Default value]	Description	Condition
manufacturer	-	CDATA	The name of the manufacturer of the material [–]	Specifies the character string of the name of the manufacturer of the material.	
product_name	-	CDATA	The product name of the material [–]	Specifies the character string of the product name, product code, etc. provided by the manufacturer.	
url	-	CDATA	The URL of the website providing material information [–]	Specifies the URL of the website providing material information.	

See Fig. 10 for XML code examples for <product_info>.

3.2.3. <iso_standard>

Parent element: <material>; Link: -

<iso_standard> specifies a material by its ISO standards. For example, when acrylonitrile butadiene styrene (ABS), which is listed in ISO 1043-1:2006, is specified as the material, the character strings "ISO 1043-1:2006" and "ABS" are to be used.

<iso_standard> has the following elements.

Table 16: The description of <iso_standard>

Element	Attribute	Data type	Data [Default value]	Description	Condition
iso_id	-	String	The ISO standard number [—]	Specifies the character string of the ISO standard number describing the material to be used.	
iso_name	-	String	The material name described in the ISO standard [—]	Specifies the character string of the material name or the abbreviation as listed in the ISO standard.	

See Fig. 10 for XML code examples for <iso_standard>.

4. <voxel>

Parent element: <fav>; Link: <geometry>, <material>, <display>, <application_note>

<voxel> is the basic element constituting 3D model data in the FAV format. By arranging voxels in three-dimensional space, the overall structure of 3D model data is defined. <voxel> references information (e.g., <geometry>, <material>) registered under <palette>. Therefore, it is possible to define attributes such as colors and materials for each voxel of a 3D model as well as the overall shape of the model.

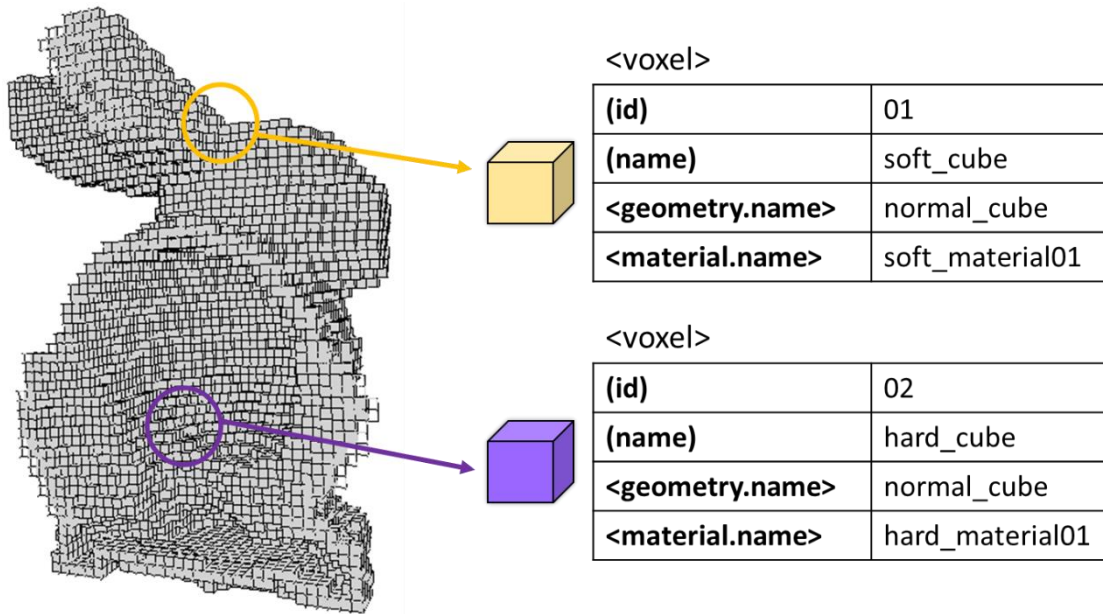


Fig. 11 : Specifying the overall shape and attributes using <voxel>

<voxel> has the following attributes.

Table 17: The description of <voxel>

Element	Attribute	Data type	Data [Default value]	Description	Condition
voxel	id	Positive Integer	A positive integer [1~]	A positive integer of 1 or larger <u>MUST</u> be specified as the (id) that is used to identify <voxel>. The same (id) <u>MUST NOT</u> be used for multiple <voxel> elements.	Required
	name	String	A character string [Voxel001~]	Specifies the name of <voxel>. The same name <u>SHOULD NOT</u> be used for multiple <voxel> elements.	

The (id) of <voxel> MUST be a positive integer. This is because the value is listed as a hexadecimal number in <voxel_map>, the element which defines the shape of a 3D object. Also, 0 MUST NOT be used for (id) because this value is reserved to denote the absence of a <voxel>.

The following elements are described at the level under <voxel>.

- <geometry_info>
- <material_info>
- <display>
- <application_note>

Shown below is an example of XML code under `<voxel>`.

```
e.g.,
<voxel id="1" name="soft_cube">
  <geometry_info>
    <id>1</id>
  </geometry_info>
  <material_info>
    <id>1</id>
    <ratio>1</ratio>
  </material_info>
</voxel>
<voxel id="2" name="hard_cube">
  <geometry_info>
    <id>1</id>
  </geometry_info>
  <material_info>
    <id>1</id>
    <ratio>0.15</ratio>
  </material_info>
  <material_info>
    <id>2</id>
    <ratio>0.85</ratio>
  </material_info>
  <application_note><![CDATA[HM-H01:Hybrid Hard Material Number 01]]></application_note>
  <application_note><![CDATA[FabAppAttr : application note]]></application_note>
</voxel>
```

Fig. 12 : XML code examples under `<voxel>`

4.1. `<geometry_info>`

Parent element: `<voxel>`; Link: `<palette>`, `<geometry>`

`<geometry_info>` specifies the shape and scale of a voxel, as defined under `<geometry>`.

The `<id>` value specified here MUST be that value which was specified in `<geometry>` registered under `<palette>`. When the shape of 3D model data is defined using voxels that have `<geometry>` (`id`)s specified under `<geometry_info>`, the shapes and scales of each voxel at their respective positions will be the shapes and scales associated with the same IDs specified under `<geometry>` (Fig. 11).

`<geometry_info>` has the following elements.

Table 18: The description of `<geometry_info>`

Element	Attribute	Data type	Data [Default value]	Description	Condition
id	-	String	The (<code>id</code>) specified in <code><geometry></code> [-]	Specifies the (<code>id</code>) of <code><geometry></code> so that the element defining the geometric properties can be referenced. The predefined (<code>id</code>) attribute in <code><geometry></code> <u>MUST</u> be specified here.	Required

See Fig. 12 for XML code examples for `<geometry_info>`.

4.2. <material_info>

Parent element: <voxel>; Link: <palette>, <material>

<material_info> specifies the material information of a voxel, as defined under <material>.

The <id> of <material_info> is specified to be the same (id) as that of <material> (registered under <palette>). When the shape of a 3D model is defined using voxels that have material IDs specified under <material_info>, the materials used at their respective voxel positions will be the materials associated with the same (id) specified under <material> (Fig. 11).

Hybrid material can also be used by defining multiple <material_info> elements under <voxel> (Fig. 13). When using hybrid material, the following elements are specified under <material_info>: <ratio>, which specifies the proportion of each material, and <id>, which references the material specified under <material>. Values MUST be specified so that the sum of all <ratio> values in <material_info> of a given <voxel> element is 1.0.

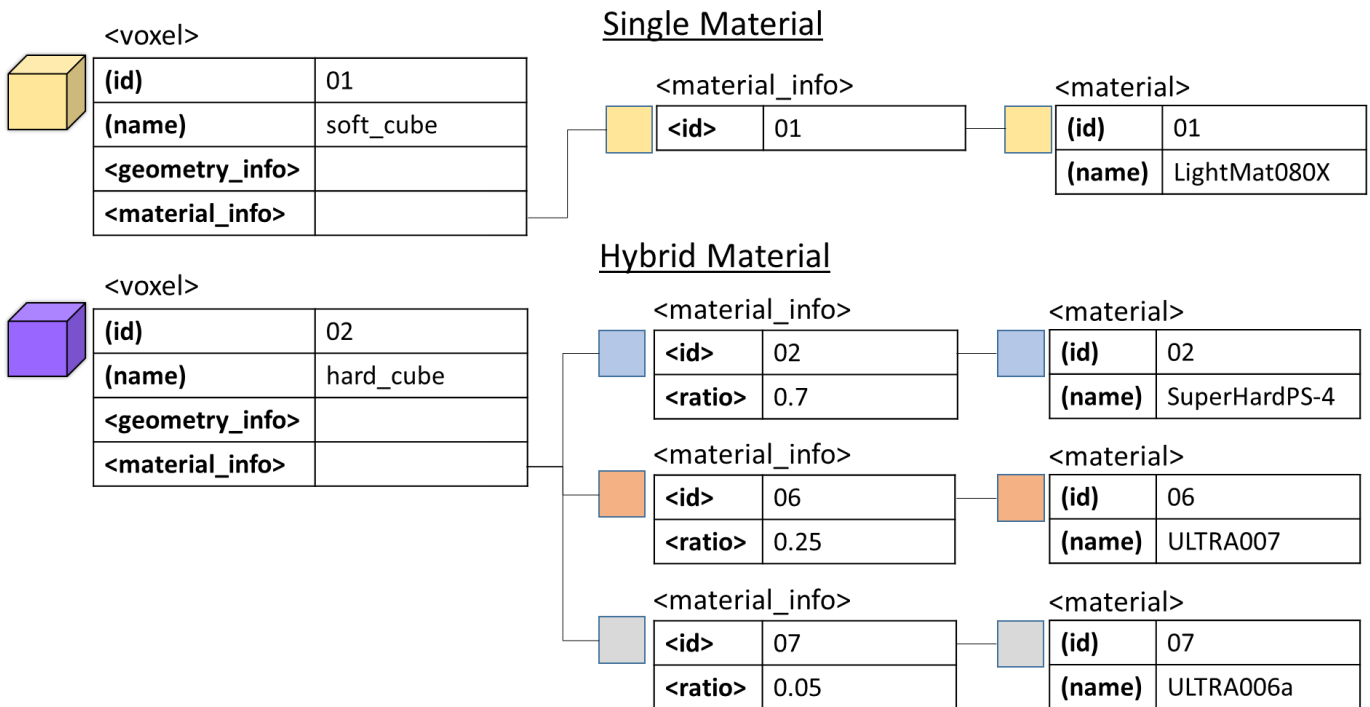


Fig. 13 : Examples of voxels made up of single and composite material

<material_info> contains the following elements.

Table 19: The description of <material_info>

Element	Attribute	Data type	Data [Default value]	Description	Condition
id	-	positive Integer	The (id) of <material> [-]	Specifies the (id) of <material> so that the element defining the material can be referenced. The (id) attribute in <material> <u>MUST</u> be specified here.	Required
ratio	-	Double	The proportion of the material to be used in the composite material [-]	A double value is used to specify what proportion the material makes up of the whole (defined under <material>). Values <u>MUST</u> be specified so that they total to 1.0 when added. The value <u>MUST</u> be larger than 0.	

See Fig. 12 for XML code examples for <material_info>.

4.3. <display>

Parent element: <voxel>; Link: <color_map>

<display> specifies the color information of <voxel>. This element is used so that differences in attributes such as the shape and material can be visually confirmed.

If the <display> element is defined under <voxel> when defining the shape of a 3D object, the color of the voxel will be the color specified under <display>. The color specified in <display> can also be used as the color of the object when a 3D object is actually fabricated. However, if many colors are to be used for an object, it is RECOMMENDED that <color_map> be used.

<display> has the following elements.

Table 20: The description of <display>

Element	Attribute	Data type	Data [Default value]	Description	Condition
r	-	nonNegativeInteger	0 - 255 [—]	Specifies the red component of the color of a <voxel>. A value smaller than 0 or a value exceeding 255 <u>MUST NOT</u> be specified.	
g	-	nonNegativeInteger	0 - 255 [—]	Specifies the green component of the color of a <voxel>. A value smaller than 0 or a value exceeding 255 <u>MUST NOT</u> be specified.	
b	-	nonNegativeInteger	0 - 255 [—]	Specifies the blue component of the color of a <voxel>. A value smaller than 0 or a value exceeding 255 <u>MUST NOT</u> be specified.	
a	-	nonNegativeInteger	0 - 255 [—]	Specifies the alpha component of the color of a <voxel>. A value smaller than 0 or a value exceeding 255 <u>MUST NOT</u> be specified.	

See Fig. 12 for XML code examples for <display>.

4.4. <application_note>

Parent element: <voxel>; Link: -

<application_note> defines properties for <voxel>.

It is necessary to add one <application_note> element for each property. Also, it is RECOMMENDED that descriptions be added to assist users (those who create and use properties) in distinguishing and selecting necessary data from multiple <application_note> elements.

Note that large data SHOULD NOT be stored and large numbers of <application_note> elements SHOULD NOT be defined unnecessarily.

<application_note> has the following data.

Table 21: The description of <application_note>

Element	Attribute	Data type	Data [Default value]	Description	Condition
application_note	—	CDATA	A property stored in <voxel> [—]	Specifies the character string of a property to be stored in <voxel>. It is <u>RECOMMENDED</u> that descriptions that may help users distinguish and select necessary data from multiple <application_note> elements be added.	

See Fig. 12 for XML code examples for <application_note>.

5. <object>

Parent element: <fav>; Link: <voxel>, <grid>, <structure>

<object> defines 3D model data in FAV format to create the actual form of the final 3D object.

Underneath <object> are the <grid> and <structure> elements. <grid> defines the space that stores 3D model data, and within this space, <structure> defines the structure of the 3D model data. 3D model data is divided into <voxel_map>, which defines the shape, <color_map>, which defines color information, and <link_map>, which defines the connection strength between voxels.

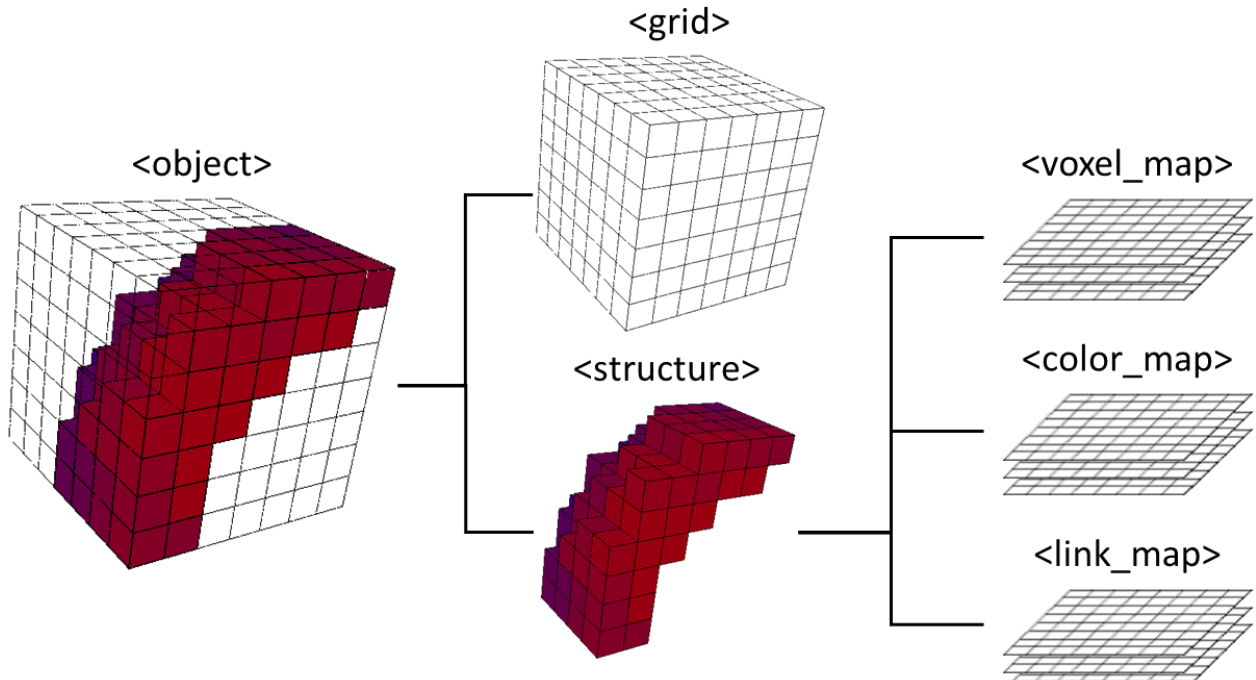


Fig. 14 : The relation between grid and structure

Multiple <object> can be defined in a FAV file. When multiple <object> elements are defined, <grid> and <structure> are defined for each <object> element.

<object> has the following attributes.

Table 22: The description of <object>

Element	Attribute	Data type	Data [Default value]	Description	Condition
object	id	positive Integer	A positive integer [1~]	Specifies the (id) that is used to identify <object>. The same (id) MUST NOT be used for multiple <object> elements.	Required
	name	String	A character string [Object001~]	Specifies the name of <object>. The same (name) SHOULD NOT be used for multiple <object> elements.	

The following elements **MUST** be described at the level under <object>.

- <grid>
- <structure>

Shown below is an example of XML code under `<object>`

```
e.g.,
<object id="1" name="SampleObject">
  <grid>
    <origin>
      <x>28.5</x>
      <y>-30</y>
      <z>0</z>
    </origin>
    <unit>
      <x>1</x>
      <y>1</y>
      <z>1</z>
    </unit>
    <dimension>
      <x>7</x>
      <y>7</y>
      <z>7</z>
    </dimension>
  </grid>
  <structure>
    <voxel_map compression="none" bit_per_voxel="8">
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
    </voxel_map>
    <color_map compression="none" color_mode="RGB">
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
    </color_map>
    <link_map compression="none" neighbors="6">
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
      <layer><![CDATA[.....]]></layer>
    </link_map>
  </structure>
</object>
```

Fig. 15 : XML code examples under `<object>`

5.1. <grid>

Parent element: <object>; Link: <voxel>, <shape>, <scale>

<grid> defines the space in which 3D model data in FAV format is defined. The space is divided into cells of uniform size. The structure of 3D model data is defined based on the definition of <grid>. When multiple <object> are defined in a FAV file, <grid> is defined for each <object> element.

The following elements are specified at the level under <grid>.

- <origin>
- <unit>
- <dimension>

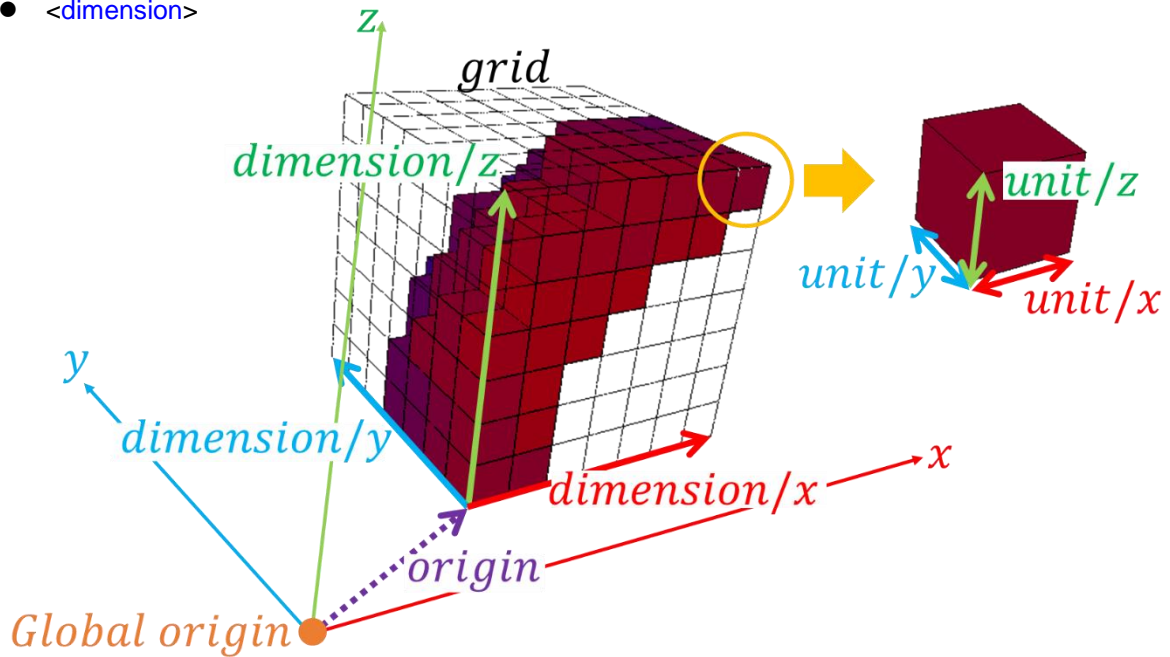


Fig. 16 : Visual description of each element in <grid>

5.1.1. <origin>

Parent element: <grid>; Link: -

<origin> defines the distance from the <origin> of the global coordinate system to the <origin> of the coordinate system of <grid>, in so specifying the position of an object in the global coordinate system.

<origin> has the following elements.

Table 23: The description of <origin>

Element	Attribute	Data type	Data [Default value]	Description	Condition
x	-	Double	A double value [0.0]	Specifies the distance from the origin of the global coordinate to the origin of the coordinate of <grid> for the x-axis.	
y	-	Double	A double value [0.0]	Specifies the distance from the origin of the global coordinate to the origin of the coordinate of <grid> for the y-axis.	
z	-	Double	A double value [0.0]	Specifies the distance from the origin of the global coordinate to the origin of the coordinate of <grid> for the z-axis.	

See Fig. 15 for XML code examples for <origin>.

5.1.2. <unit>

Parent element: <grid>; Link: <voxel>, <geometry>, <shape>

<unit> specifies the size of a cell within <grid>. The size can be specified for each of the axes: x, y, and z.

<unit> has the following elements.

Table 24: The description of <unit>

Element	Attribute	Data type	Data [Default value]	Description	Condition
x	-	Double	A double value [1.0]	Specifies the length of a single cell in the direction of the x-axis. The value MUST be larger than 0.	
y	-	Double	A double value [1.0]	Specifies the length of a single cell in the direction of the y-axis. The value MUST be larger than 0.	
z	-	Double	A double value [1.0]	Specifies the length of a single cell in the direction of the z-axis. The value MUST be larger than 0.	

See Fig. 15 for XML code examples for <unit>.

5.1.3. <dimension>

Parent element: <grid>; Link: <voxel>, <unit>

<dimension> defines the overall size of a <grid>. The size is specified by determining the maximum number of <voxel> that can be arranged in a <grid>. The actual size of a grid in the global coordinate system can be calculated by multiplying the cell size defined in <unit> by the number of cells defined in <dimension>.

<dimension> has the following elements

Table 25: The description of <dimension>

Element	Attribute	Data type	Data [Default value]	Description	Condition
x	-	Positive Integer	A positive integer [-]	Specifies the maximum number of voxels that can be arranged in a <grid> in the x-axis direction.	Required
y	-	Positive Integer	A positive integer [-]	Specifies the maximum number of voxels that can be arranged in a <grid> in the y-axis direction.	Required
z	-	Positive Integer	A positive integer [-]	Specifies the maximum number of voxels that can be arranged in a <grid> in the z-axis direction.	Required

See Fig. 15 for XML code examples for <dimension>.

5.2. <structure>

Parent element: <object>; Link: <grid>, <voxel>, <voxel_map>, <color_map>, <link_map>

<structure> defines the structure of 3D model data in FAV format. The shape of a 3D object is defined by arranging voxels (defined by <voxel>) within a three-dimensional grid (defined by <grid>). Also, information such as the color, material, connection strength of each arranged <voxel> is defined.

The following elements are described at the level under <structure>.

- <voxel_map>
- <color_map>
- <link_map>

The following sections explain each of the elements located under <structure>. To do so, an example is used in which the following 3D model data is defined.

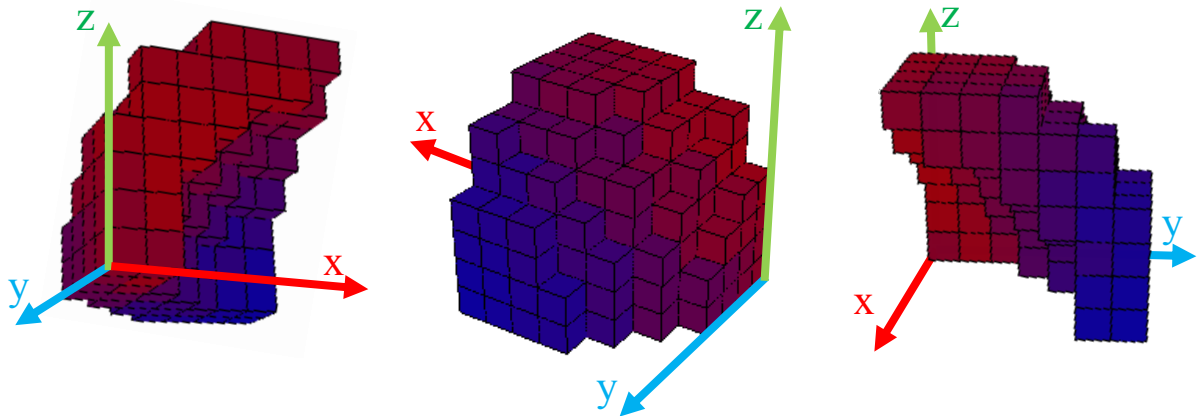


Fig. 17 : Example 3D model data defined in the FAV format

See Fig. 27 for XML code examples for <structure>.

5.2.1. <voxel_map>

Parent element: <structure>; Link: <grid>, <voxel>

<voxel_map> defines the overall shape of 3D model data in FAV format. In a <voxel_map>, it is necessary to define whether a voxel is present or not at each voxel position of an x-y layer (horizontal layer defined on the x and y axes), for each layer of a three-dimensional <grid>. By defining the voxel arrangement in the voxel map for each layer and repeating this for the number of layers comprising the height of the object, the overall shape of a 3D object is defined.

<voxel_map> has the following attributes.

Table 26: The description of <voxel_map>

Element	Attribute	Data type	Data [Default value]	Description	Condition
voxel_map	bit_per_voxel	positive Integer	4 / 8 / 16 [—]	Specifies a numerical value of 4, 8, or 16 (representing the length of hexadecimal characters) to define the number of bits in one voxel. (4, 8, and 16 each indicate one, two, and four hexadecimal characters respectively.)	Required
	compression	String	none / zlib / base64 [none]	Selects the compression method for <layer> from the following: none (no compression), zlib, and base64.	

<layer> elements are defined at the level under <voxel_map>. The number of <layer> elements to be defined is determined by <dimension.z>. <layer> elements are stored starting from the lowest layer on the z-axis (i.e., the base of the 3D object).

One voxel map is defined for each x-y layer (<layer>). When the (id) of <voxel> is written in a specific position in voxel map, it indicates that the <voxel> associated to that (id) is present at that specified position in the <grid>. When no voxel is to be present in a position, 0 is written in the corresponding position in voxel map.

<voxel_map.layer> has the following data.

Table 27: The description of <voxel_map.layer>

Element	Attribute	Data type	Data [Default value]	Description	Condition
layer	-	CDATA	The character string listing all the (id) values of each <voxel> [—]	Lists the (id) values in <voxel> using the format specified in (bit_per_voxel). Each (id) value is concatenated into one line (the number of IDs to be concatenated can be calculated by multiplying <dimension.x> by <dimension.y>). When (compression) in <voxel_map> is set to that other than "none", the data in <layer> is compressed using the specified method.	

Shown below is the procedure for defining the <voxel_map> for the first layer of the 3D model data shown in Fig. 17.

(The cells of the voxel maps to the right of the figures below are colored solely for the purpose of improving readability, and have no effect on the actual 3D model.)

- ① The presence and absence of voxels is defined for each position along the x-axis (<dimension.x>) in <grid> by specifying the (id) values of <voxel>. A character string is formed by concatenating the (id) values of each <voxel> element. 0 is specified to denote the absence of a voxel at a certain position.

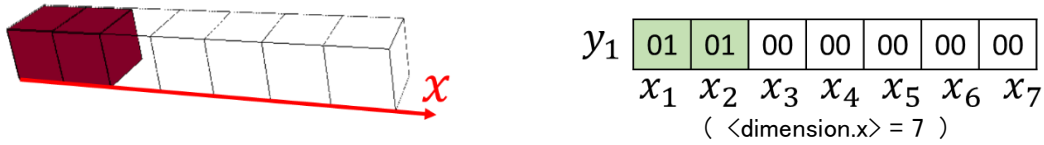


Fig. 18 : <voxel_map> for the x-axis (<dimension.x>)

- ② The process in ① (specifying (id) values for <dimension.x>) is carried out for each row on the y-axis (i.e., the number of rows defined by <dimension.y>) to create the <voxel_map>.

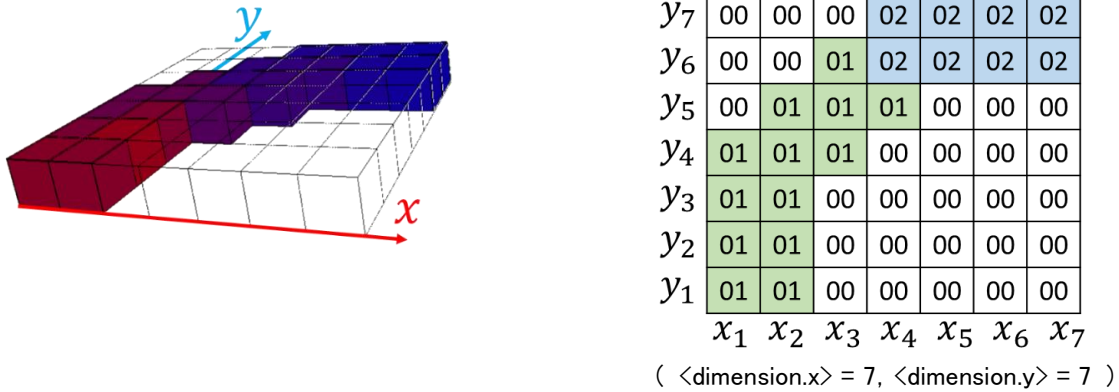


Fig. 19 : <voxel_map> for one layer (<dimension.x> x <dimension.y>)

- ③ The data for a single <layer> element is obtained by concatenating the values defined for each position on the x-y layer (<dimension.x> x <dimension.y>) in <voxel_map>

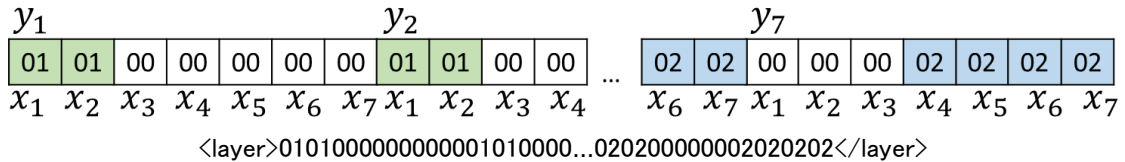


Fig. 20 : The data of a single <layer> obtained by concatenating the values of <voxel_map>

- ④ The process shown above for creating a voxel map for a single layer is to be carried out for the number of layers comprising the z-axis (i.e., the number of layers defined by <dimension.z>), in so defining the overall shape and structural information of the 3D model data.

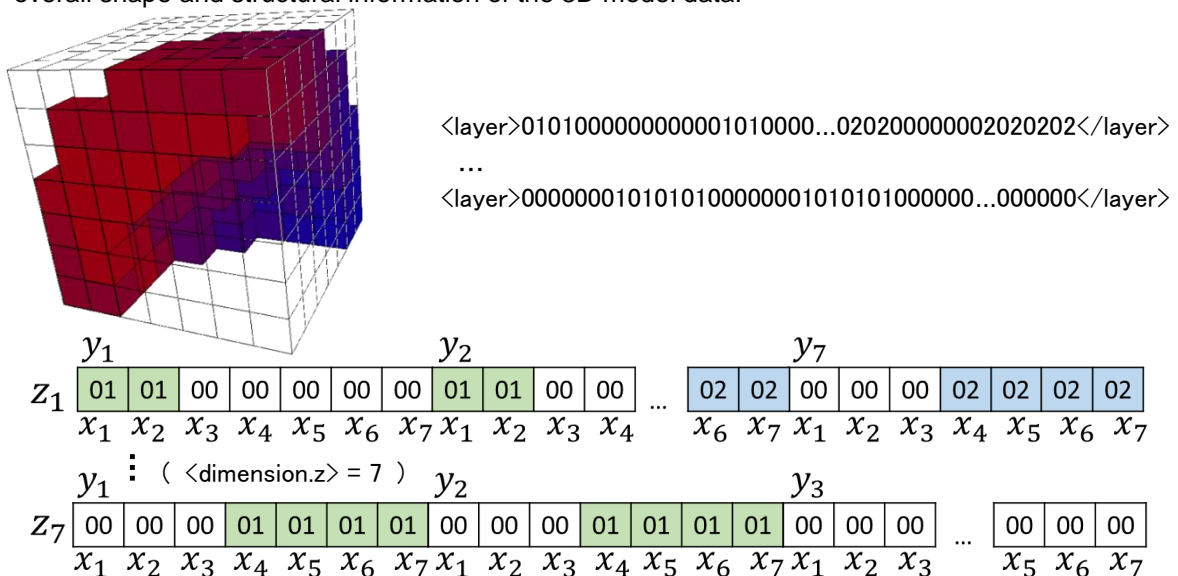


Fig. 21 : <voxel_map> for the overall 3D model data

5.2.2. <color_map>

Parent element: <structure>; Link: <grid>, <voxel>, <display>

<color_map> defines the color information of 3D model data in FAV format. Color information is specified for each <voxel> element listed in <voxel_map>.

<color_map> has the following attributes.

Table 28: The description of <color_map>

Element	Attribute	Data type	Data [Default value]	Description	Condition
color_map	color_mode	String	GrayScale / GrayScale16 / RGB / RGBA / CMYK [–]	Selects the color mode from the following: GrayScale, GrayScale16, RGB, RGBA, CMYK.	Required
	compression	String	none / zlib / base64 [none]	Selects the compression method for <layer> from the following: none (no compression), zlib, and base64.	

The following are the color formats available for (color_mode).

- GrayScale (white to black)
One byte per channel is used, allowing the use of 256 shades. The color information for one voxel is written in one byte (two hexadecimal characters).
- GrayScale16 (white to black)
Two bytes per channel are used, allowing the use of 512 shades. The color information for one voxel is written in two bytes (four hexadecimal characters).
- RGB
One byte is used for each channel of red, green, and blue, allowing the use of 256 shades per channel. The color information for one voxel is written in three bytes (six hexadecimal characters).
- RGBA
One byte is used for each channel of red, green, blue, and alpha, allowing the use of 256 shades per channel. The color information for one voxel is written in four bytes (eight hexadecimal characters).
- CMYK
One byte is used for each channel of cyan, magenta, yellow, and key plate, allowing the use of 256 shades per channel. The color information for one voxel is written in four bytes (eight hexadecimal characters).

<layer> elements are defined at the level under <color_map>. The number of <layer> elements to be defined is determined by <dimension.z>. One <color_map> element is defined for each x-y layer (<layer>). In <color_map>, color information for each voxel listed in <voxel_map> is specified in accordance with the mode specified by (color_mode). When 0 is specified for a <voxel> in <voxel_map> (denoting the absence of a voxel at a certain position), the color information of the <voxel> is omitted in <color_map> and the values are left-aligned.

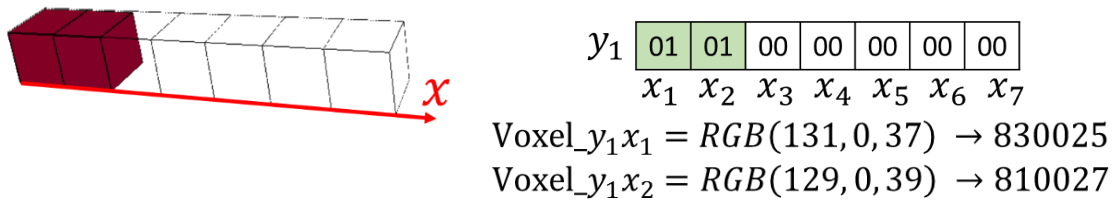
<color_map.layer> has the following data.

Table 29: The description of <color_map.layer>

Element	Attribute	Data type	Data [Default value]	Description	Condition
layer	-	CDATA	A hexadecimal character string [-]	Specifies the color information of each <voxel> in <voxel_map> in the mode specified by (color_mode). When (compression) of <color_map> is set to that other than "none", data in <layer> is compressed using the specified method.	1 or more

Shown below is the procedure for defining the color map for the first layer of the 3D model data shown in Fig. 17. (The cells of the color maps to the right of the figures below are colored solely for the purpose of improving readability, and have no effect on the actual 3D model.)

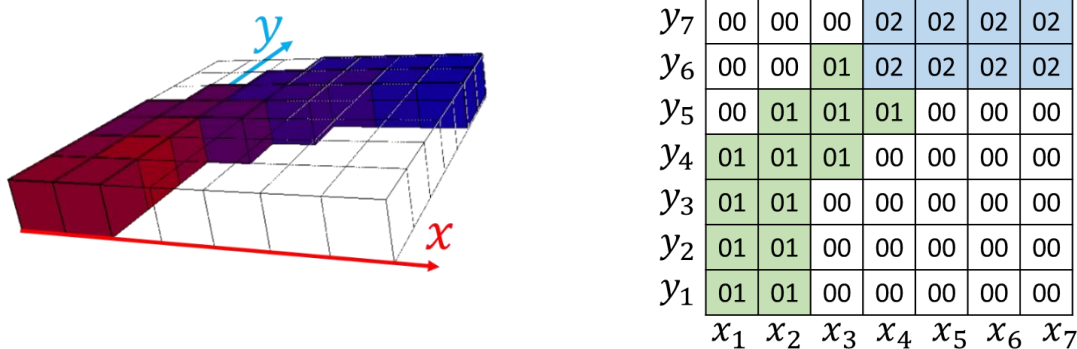
- ① The color information of each voxel on the x-axis (<dimension.x>) in a <grid> is defined as specified by (color_mode). When this is specified as 0 (denoting the absence of a voxel), color information is omitted.



The remaining voxel color information is omitted as the values are 0.
 The value of <color_map> for the x-axis (<dimension.x>) is 830025810027.

Fig. 22 : <color_map> for the x-axis (<dimension.x>)

- ② The process in ① (specifying color information for <dimension.x>) is carried out for each row of the y-axis (i.e., the number of rows defined by <dimension.y>) to create the color map. The data for a single <layer> is obtained by concatenating the values defined on the x-y layer (<dimension.x> x <dimension.y>) of <color_map>.



An example of the color map data for 21 voxels in one layer
 <Layer>8300258100277600329100176400457c002d5e004a5c004c5000595600523300753700713000782f007a3100771800900f00991f00891c008c1300960c009c</Layer>

Fig. 23 : <color_map> for one layer (<dimension.x> x <dimension.y>)

- ③ The process above for creating a color map for a single layer is to be carried out for the number of layers comprising the z-axis (i.e., the number defined by <dimension.z>), in so defining the color information for the 3D model data overall.

5.2.3. <link_map>

Parent element: <structure>; Link: <grid>, <voxel>

<link_map> defines the link information of 3D model data in FAV format. Link information is that information which defines the degree of relation between each voxel (e.g., connection strength). Link information can be used to achieve more precise structure analysis and to generate tool paths of 3D printers, etc. Link information is defined for each voxel listed in <voxel_map>.

<link_map> has the following attributes.

Table 30: The description of <link_map>

Element	Attribute	Data type	Data [Default value]	Description	Condition
link_map	neighbors	positive integer	6 / 18 / 26 [—]	Specifies a numerical value of 6, 18, or 26 to define the number of neighboring voxels for which link information is retained.	Required
	compression	String	none / zlib / base64 [none]	Selects the compression method for <layer> from the following: none (no compression), zlib, and base64.	

The following are three options for specifying the number of voxels that neighbor a certain voxel (**neighbors**) for which link information is retained.

- 6 neighboring voxels
Retains the link information between a specific voxel and the six neighboring voxels which its six faces are in full and complete contact with. Link information for a voxel is written in six bytes (12 hexadecimal characters).
- 18 neighboring voxels
Retains the link information between a specific voxel and the 18 neighboring voxels which the voxel in question is in contact with by face OR edge (excludes neighbors only in contact by corner). Link information for a voxel is written in 18 bytes (36 hexadecimal characters).
- 26 neighboring voxels
Retains the link information between a specific voxel and the 26 neighboring voxels which the voxel in question is in contact with by face OR edge OR corner (includes neighbors only in contact by corner). Link information for a voxel is written in 26 bytes (52 hexadecimal characters).

<layer> elements are defined at the level under <link_map>. The number of <layer> elements to be defined is determined by <dimension.z>. One <link_map> element is defined for each x-y layer (<layer>). In <link_map>, link information for each <voxel> listed in <voxel_map> is specified for the number of bytes specified by (**neighbors**). When 0 is specified for a voxel in <voxel_map> (denoting the absence of a voxel at a certain position), the link information of the <voxel> is omitted in <link_map> and the values are left-aligned.

When there are no neighboring voxels, 0 is specified for link information. Link information MUST be listed for each voxel defined as a neighbor, i.e., the number of bytes specified by (**neighbors**).

The order in which link information is defined in `<link_map>` is as shown below Fig. 24. Link information is defined starting from the voxel with the lowest coordinate values under the axis ordering of z, x, y. The red voxel in the figure below indicates a reference voxel, and the numbers illustrate the order in which link information is defined

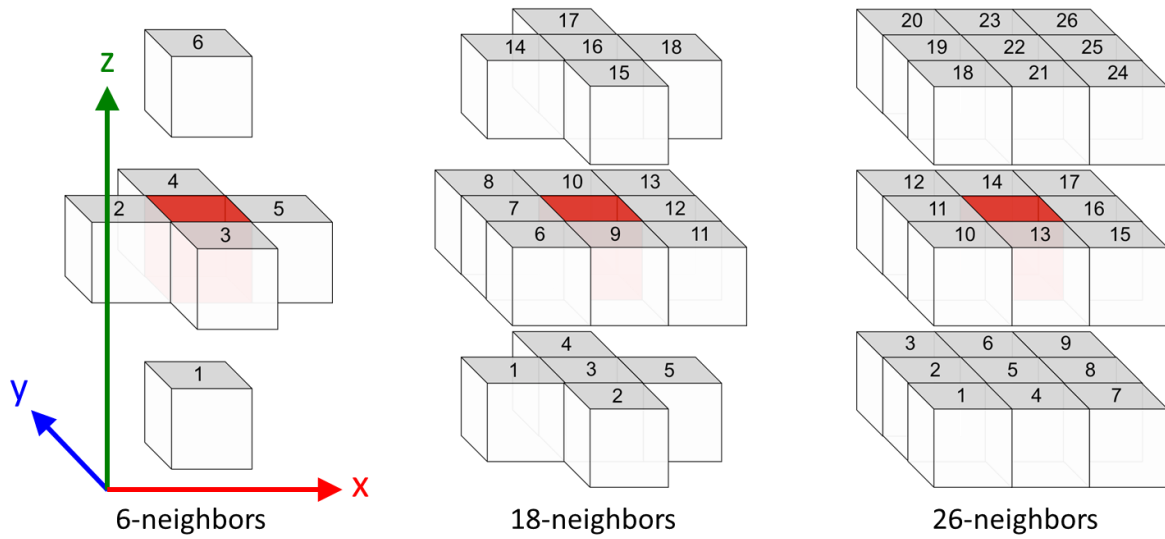


Fig. 24: The order in which link information is defined

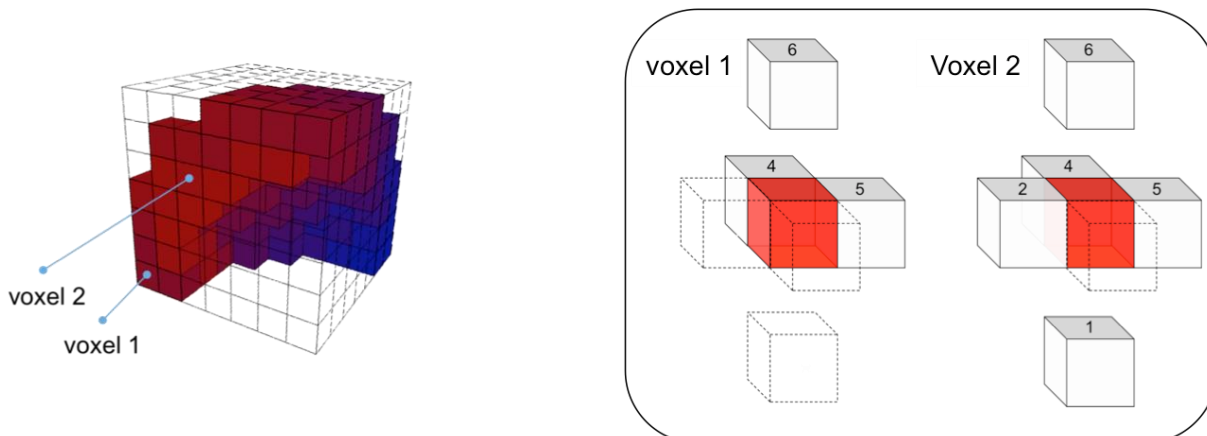
`<link_map.layer>` has the following data.

Table 31: The description of `<link_map.layer>`

Element	Attribute	Data type	Data [Default value]	Description	Condition
layer	-	CDATA	A hexadecimal character string [-]	Specifies the link information in the order shown in Fig. 24 for each <code><voxel></code> in <code><voxel_map></code> as specified by <code>(neighbors)</code> . When <code>(compression)</code> of <code><link_map></code> is set to that other than "none", data in <code><layer></code> is compressed using the specified method.	1 or more

Shown below is an example of the link map data specified for two voxels in the 3D model data of Fig. 17.

When (*neighbors*) is set to six and the x, y, and z coordinate values of the link information are 100, 200, and 255 respectively, the values of link information for voxel 1 and voxel 2 in Fig. 25 will be as shown below.

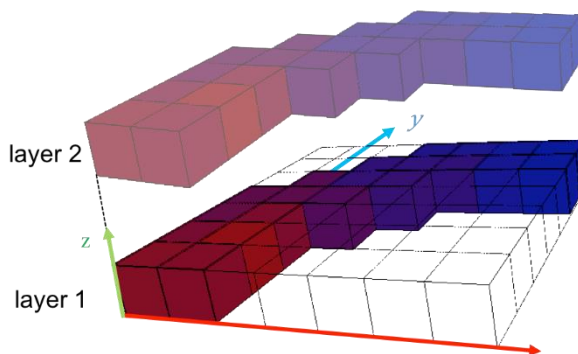


Voxel1 = LINK(0,0,0,200,100,255) → 000000c864ff

Voxel2 = LINK(255,100,0,200,100,255) → ff6400c864ff

Fig. 25 : Link information for neighboring voxels

The link information for a single layer is defined by specifying the link information for each voxel in the layer. The number of voxels for which link information is defined can be calculated by multiplying <dimension.x> by <dimension.y>. The data of a single <layer> is obtained by concatenating the link information values of each voxel. To define the link information of the overall 3D model data, the link information map created for a single layer is to be created for each layer comprising the z-axis (i.e., the number defined by <dimension.z>).



An example of the link map data for 21 voxels in layer 1

```
<layer>000000c864ff006400c800ff0000c8c864ff0064c8c800ff0000c8c864ff0064c8c800ff0000c8c864ff0064c8c800ff0000c8c864ff0064c8c800ff0000c80064ff0064c8c864ff006400c800ff0000c80064ff0064c8c864ff006400c864ff006400c864ff006400c800ff0000c80064ff0064c80064ff0064c80064ff0064c80000ff</layer>
```

Fig. 26 : <link_map> for one layer (<dimension.x> x <dimension.y>)

6. Conclusion

Shown below is the list of all elements and attributes in the FAV format described in this document

Table 32: All elements and attributes in the FAV format

Element (The further left an element is positioned on the table, the further up that element is in the tree structure, and the further right, the further down.)		Attribute	Data type	Data [Default value]	Description	Condition	
—	fav	-	-	-	This is the root element of a FAV file. <metadata>, <palette>, <voxel>, <object> elements are described at the level under <fav>.	Required	
		version	String	The version number of the FAV format [—]	Specifies the character string representing the version number of the FAV format used for the file.	Required	
fav / material / object	metadata	→	-	-	In <metadata>, <id>, <title>, <author>, <license>, <note> elements may be defined.		
		id	-	String	The id of the parent element [—]	Specifies the character string of the ID that is used to identify the parent element of <metadata>.	Required
		title	-	CDATA	The name of the parent element [—]	Specifies the character string of the name of the parent element of <metadata>.	Required
		author	-	CDATA	The author of the parent element [—]	Specifies the character string of the author of the parent element of <metadata>.	Required
		license	-	CDATA	The license information of the parent element [—]	Specifies the character string describing the license information of the parent element of <metadata>.	Required

			note	-	CDATA	The memo for the parent element [-]	Specifies the character string of a memo for the parent element of <metadata>.		
fav	palette	→		-	-	-	<geometry>, <material> elements are described at the level under <palette>.		
		geometry	→		-	-	-	<shape>, <scale> elements are described at the level under <geometry>.	Unbounded
			id		positive Integer	A positive integer [1~]	Specifies the unique ID that is used to identify <geometry>. The same ID <u>MUST NOT</u> be used for multiple <geometry> elements.	Required	
			name		String	A character string [Object1~]	Specifies the name of <geometry>. The same (name) <u>SHOULD NOT</u> be used for multiple <geometry> elements.		
		shape		-	String	cube / sphere / user_defined [cube]	Specifies the shape of a voxel. When "user_defined" is specified, <reference> <u>MUST</u> also be defined.		
		reference		-	CDATA	The path of the STL file defining the shape of a voxel [-]	Specifies the character string of the relative path to the external STL file defining the shape of a voxel.		
		scale	x	-	Double	A double value [1.0]	Specifies the scale for the x-axis of <shape> in relation to the cell size specified in <grid>. When a negative value is specified, the voxel is flipped. 0 <u>MUST NOT</u> be specified.		

fav	palette	geometry	scale	y	-	Double	A double value [1.0]	Specifies the scale for the y-axis of <shape> in relation to the cell size specified in <grid>. When a negative value is specified, the voxel is flipped. 0 <u>MUST NOT</u> be specified.			
				z	-	Double	A double value [1.0]	Specifies the scale for the z-axis of <shape> in relation to the cell size specified in <grid>. When a negative value is specified, the voxel is flipped. 0 <u>MUST NOT</u> be specified.			
	material	→			-	-	-	-	One of the following elements representing material information is defined.	Unbounded	
					id	positive Integer	A positive integer [1~]	Specifies the unique ID that is used to identify <material>. The same ID <u>MUST NOT</u> be used for multiple <material> elements.	Required		
					name	String	A character string [Material01~]	Specifies the name of <material>. The same (name) <u>SHOULD NOT</u> be used for multiple <material> elements.			
		material_name	→	-	CDATA	The material name [-]	Specifies a character string identifying the material used.	Unbounded			
		product_info	→			-	-	-	-	This has the following elements.	Unbounded
						manufacturer	-	CDATA	The name of the manufacturer of the material [-]	Specifies the character string of the name of the manufacturer of the material.	
						product_name	-	CDATA	The product name of the material [-]	Specifies the character string of the product name, product code, etc. provided by the manufacturer.	

fav	palette	material	product_info	url	-	CDATA	The URL of the website providing material information [-]	Specifies the URL of the website providing material information.		
			iso_standard	→	-	-	-	-	This has the following elements.	Unbounded
				iso_id	-	String	The ISO standard number [-]	Specifies the character string of the ISO standard number describing the material to be used.		
				iso_name	-	String	The material name described in the ISO standard [-]	Specifies the character string of the material name or the abbreviation as listed in the ISO standard.		
	voxel	→			-	-	-	<geometry_info>, <material_info>, <display>, <application_note> elements are described at the level under <voxel>.	Unbounded	
		→			id	positive Integer	A positive integer [1~]	A positive integer of 1 or larger <u>MUST</u> be specified as the ID identifying <voxel>. The same (id) <u>MUST NOT</u> be used for multiple <voxel> elements.	Required	
		→			name	String	A character string [Voxel001~]	Specifies the name of <voxel>. The same (name) <u>SHOULD NOT</u> be used for multiple <geometry> elements.		
		geometry_info	→			-	-	-	This has the following elements.	
			→	id	-	String	The (id) specified in <geometry> [-]	Specifies the (id) of <geometry> so that the element defining the geometric properties can be referenced. The predefined (id) attribute in <geometry> <u>MUST</u> be specified here.	Required	

fav	palette	material_info	→	-	-	-	This has the following elements.	Unbounded
			id	-	positive Integer	The (id) of <material> [-]	Specifies the (id) of <material> so that the element defining the material can be referenced. The (id) attribute in <material> <u>MUST</u> be specified here.	Required
			ratio	-	Double	The proportion of the <material> to be used in the composite <material> [-]	A double value is used to specify what proportion the material makes up of the whole (defined under <material>). Values <u>MUST</u> be specified so that they total to 1.0 when added. The value <u>MUST</u> be larger than 0.	
		display	r	-	nonNegativeInteger	0 - 255 [-]	Specifies the red component of the color of a <voxel>. A value smaller than 0 or a value exceeding 255 <u>MUST NOT</u> be specified.	
			g	-	nonNegativeInteger	0 - 255 [-]	Specifies the green component of the color of a <voxel>. A value smaller than 0 or a value exceeding 255 <u>MUST NOT</u> be specified.	
			b	-	nonNegativeInteger	0 - 255 [-]	Specifies the blue component of the color of a <voxel>. A value smaller than 0 or a value exceeding 255 <u>MUST NOT</u> be specified.	
	a		-	nonNegativeInteger	0 - 255 [-]	Specifies the alpha component of the color of a <voxel>. A value smaller than 0 or a value exceeding 255 <u>MUST NOT</u> be specified.		
	application_note	→	-	CDATA	A property stored in <voxel> [-]	Specifies the character string of a property to be stored in <voxel>. It is <u>RECOMMENDED</u> that descriptions that may help users distinguish and select necessary data from multiple <application_note> elements be added.		

fav	object			-	-	-	<grid> , <structure> <u>MUST</u> be described at the level under <object>.	Required , Unbounded	
				id	positive Integer	A positive integer [1~]	Specifies the (id) that is used to identify <object>. The same (id) <u>MUST NOT</u> be used for multiple <object> elements.	Required	
				name	String	A character string [Object001~]	Specifies the name of <object>. The same (name) <u>SHOULD NOT</u> be used for multiple <object> elements.		
	grid			→	-	-	-	<origin>, <unit>, <dimension> elements are described at the level under <grid>.	
				origin	x	-	Double	A double value [0.0]	Specifies the distance from the origin of the global coordinate system to the origin of the coordinate system of <grid> for the x-axis.
		y	-		Double	A double value [0.0]	Specifies the distance from the origin of the global coordinate system to the origin of the coordinate system of <grid> for the y-axis.		
		z	-		Double	A double value [0.0]	Specifies the distance from the origin of the global coordinate system to the origin of the coordinate system of <grid> for the z-axis.		
		unit	x	-	Double	A double value [1.0]	Specifies the length of a single cell in the direction of the x-axis. The value <u>MUST</u> be larger than 0.		
			y	-	Double	A double value [1.0]	Specifies the length of a single cell in the direction of the y-axis. The value <u>MUST</u> be larger than 0.		

fav	object	grid		z	-	Double	A double value [1.0]	Specifies the length of a single cell in the direction of the z-axis. The value <u>MUST</u> be larger than 0.		
			dimension	x	-	positive Integer	A positive integer [—]	Specifies the maximum number of voxels that can be arranged in a <grid> in the x-axis direction.	Required	
				y	-	positive Integer	A positive integer [—]	Specifies the maximum number of voxels that can be arranged in a <grid> in the y-axis direction.	Required	
				z	-	positive Integer	A positive integer [—]	Specifies the maximum number of voxels that can be arranged in a <grid> in the z-axis direction.	Required	
		structure	→		-	-	-	-	<voxel_map>, <color_map>, <link_map> elements are described at the level under <structure>.	
			voxel_map	→		-	-	-	This has the following elements.	
				bit_per_voxel	positive Integer	4 / 8 / 16 [—]	Specifies a numerical value of 4, 8, or 16 (representing the length of hexadecimal characters) to define the number of bits in one voxel. (4, 8, and 16 each indicate one, two, and four hexadecimal characters respectively.)	Required		
				compression	String	none / zlib / base64 [none]	Selects the compression method for <layer> from the following: none (no compression), zlib, and base64.			
			layer	-	CDATA	The character string listing all the (id) values of each <voxel> [—]	Lists the (id) values in <voxel> using the format specified in (bit_per_voxel). Each (id) value is concatenated into one line (the number of IDs to be concatenated can be calculated by multiplying <dimension.x> by <dimension.y>). When (compression) in <voxel_map> is set to that other	1 or more		

fav	object	structure						than "none", the data in <layer> is compressed using the specified method.	
			color_map	→	-	-	-	This has the following elements.	
					color_mode	String	GrayScale / GrayScale16 / RGB / RGBA / CMYK [-]	Selects the color format from the following: GrayScale, GrayScale16, RGB, RGBA, CMYK.	Required
				compression	String	none / zlib / base64 [none]	Selects the compression method for <layer> from the following: none (no compression), zlib, and base64.		
			layer	-	CDATA	A hexadecimal character string [-]	Specifies the color information of each <voxel> in <voxel_map> in the format specified by (color_mode). When (compression) of <color_map> is set to that other than "none", data in <layer> is compressed using the specified method.	1 or more	
			link_map	→	-	-	-	This has the following elements.	
					neighbors	positive Integer	6 / 18 / 26 [-]	Specifies a numerical value of 6, 18, or 26 to define the number of neighboring voxels for which link information is retained.	Required
				compression	String	none / zlib / base64 [none]	Selects the compression method for <layer> from the following: none (no compression), zlib, and base64.		
			layer	-	CDATA	A hexadecimal character string [-]	Specifies the link information in the order shown in Fig. 24 for each <voxel> in <voxel_map> as specified by (neighbors). When (compression) of <link_map> is set to that other than "none", data in <layer> is compressed using the specified method.	1 or more	

e.g.,

```

<?xml version="1.0" encoding="utf-8"?>
<fav version="1.0">
  <metadata>
    <id>bc4affb5-9a53-4de7-9f27-721ef27e8f34</id>
    <title><![CDATA[FAV Ver1.0 Sample File]]></title>
    <author><![CDATA[Fuji Xerox & Keio SFC]]></author>
    <license><![CDATA[CC BY]]></license>
    <note><![CDATA[This is a sample file in FAV format ver1.0.]]></note>
  </metadata>
  <palette>
    <geometry id="1" name="NormalCube">
      <shape>cube</shape>
      <scale>
        <x>1</x>
        <y>1</y>
        <z>1</z>
      </scale>
    </geometry>
    <geometry id="2" name="Plate">
      <shape>cube</shape>
      <scale>
        <x>1</x>
        <y>1</y>
        <z>0.25</z>
      </scale>
    </geometry>
    <geometry id="3" name="Diamond">
      <shape>user_defined</shape>
      <reference><![CDATA[Diamond.stl]]></reference>
      <scale>
        <x>0.98</x>
        <y>0.98</y>
        <z>-1.05</z>
      </scale>
    </geometry>
  </palette>
</fav>

```

```
<material id="1" name="SoftMat1">
  <material_name><![CDATA[Some-soft-materials]]></material_name>
</material>
<material id="2" name="HardMat1">
  <product_info>
    <manufacturer><![CDATA[ABC Materials Co.]]></manufacturer>
    <product_name><![CDATA[ULTRA-HARD/007]]></product_name>
    <url><![CDATA[http://www.abcmaterial.com/ultra/hard/007]]></url>
  </product_info>
  <product_info>
    <manufacturer><![CDATA[ABC Materials Co.]]></manufacturer>
    <product_name><![CDATA[ULTRA-HARD/006a]]></product_name>
    <url><![CDATA[http://www.abcmaterial.com/ultra/hard/006/a]]></url>
  </product_info>
  <iso_standard>
    <iso_id>ISO 1043-1:2006</iso_id>
    <iso_name>ABS</iso_name>
  </iso_standard>
</material>
</palette>
<voxel id="1" name="soft_cube">
  <geometry_info>
    <id>1</id>
  </geometry_info>
  <material_info>
    <id>1</id>
    <ratio>1</ratio>
  </material_info>
</voxel>
<voxel id="2" name="hard_cube">
  <geometry_info>
    <id>1</id>
  </geometry_info>
  <material_info>
    <id>1</id>
    <ratio>0.15</ratio>
  </material_info>
```

```

<material_info>
  <id>2</id>
  <ratio>0.85</ratio>
</material_info>
<application_note><![CDATA[HM-H01:Hybrid Hard Material Number 01]]></application_note>
<application_note><![CDATA[FabAppAttr : application note]]></application_note>
</voxel>
<object id="1" name="SampleObject">
  <metadata>
    <id>cafed8bd-3bd9-4d7a-a67d-2df635d2d8f8</id>
    <title><![CDATA[]]></title>
    <author><![CDATA[Mr. Sample Creator]]></author>
    <license><![CDATA[No rights reserved]]></license>
  </metadata>
  <grid>
    <origin>
      <x>28.5</x>
      <y>-30</y>
      <z>0</z>
    </origin>
    <unit>
      <x>1</x>
      <y>1</y>
      <z>1</z>
    </unit>
    <dimension>
      <x>7</x>
      <y>7</y>
      <z>7</z>
    </dimension>
  </grid>
  <structure>
    <voxel_map compression="none" bit_per_voxel="8">
      <layer><![CDATA[00000001010101...00000000000000]]></layer>
      <layer><![CDATA[00000101010101...00000000000000]]></layer>
      <layer><![CDATA[00010101000000000101010000000001010101000000000101010100000000010101000000000101010000000000000]]></layer>
      <layer><![CDATA[01010100000000010101000000000001010100000000010101010000000001010101000000010101010000000010101]]></layer>
    </voxel_map>
  </structure>
</object>

```

```
<layer><![CDATA[0101000000000010100000000001010100000000001010100000000010101010000000001010101000000001010101]]></layer>
<layer><![CDATA[01010000000000101000000000010100000000001010100000000001010100000000001010101000000001010101]]></layer>
<layer><![CDATA[0101000000000010100000000001010000000000101010000000000101010000000001010101000000001010101]]></layer>
</voxel_map>
<color_map compression="none" color_mode="RGB">
  <layer><![CDATA[1f00891c008c1300960c009c1f00891e008b14009410009827008119008f19008f2c007c2100881f0089]]></layer>
  <layer><![CDATA[3000782f007a3100771800900f00993b006d2c007c2400842d007a1700913900703800702700812300863000793d006b2d007c220086200089430
0662a007e27008139006f]]></layer>
  <layer><![CDATA[5600523300753700715500534000684200665300554900604200673d006b59004f3e006a4100684100674100676100475d004b55005342006639
006f5a004e5300554f0059]]></layer>
  <layer><![CDATA[5e004a5c004c5000595b004e5900505500537700315800515a004e7500336e003b6f003a58005172003562004759004f6300465b004e5c004c6f0
03a6d003c5800506f003a6d003c5a004f]]></layer>
  <layer><![CDATA[6400457c002d6300466800408100277b002d6c003c7f00297200367000397500337300347000386b003d8700218600237700316f00396d003c800
0287d002b7000386e003a]]></layer>
  <layer><![CDATA[7600329100177400348e001a8500237f002a8100278000288100279200169200168a001e8f001988001f8c001c7b002d7a002e8100267e002a7400
33720037]]></layer>
  <layer><![CDATA[8300258100278200269700118900209a000e9300159800109b000e99000f9700119a000e9000189000188a001e9700129600128f00198a001f860
023800028]]></layer>
</color_map>
<link_map compression="none" neighbors="6">
  <layer><![CDATA[000000c864ff000000c800ff000000c864ff000000c800ff0000c8c864ff0000c8c800ff0000c80064ff0000c8c864ff006400c800ff0000c80064ff0064
c8c864ff006400c800ff0000c80064ff0064c8c864ff006400c864ff006400c864ff006400c864ff006400c800ff0000c80064ff0064c80064ff0064c80000ff]]></layer>
  <layer><![CDATA[000000c864ff000000c800ff000000c864ff000000c800ff0000c8c864ff0000c8c800ff0000c80064000000c8c864ff006400c800ff0000c80064ff006
4c8c864ff006400c800ff0000c80064ff0064c8c864ff006400c864ff006400c864ff006400c864ff006400c800ff0000c80064000064c80064ff0064c80064ff0064c80000ff]]></layer>
  <layer><![CDATA[ff0000c864ffff0000c800ffff0000c864ffff0000c800ffff00c8006400ff00c8c864ff006400c800ffff00c8c864ffff64c8c864ff006400c800ffff00c80064
00ff64c8c864ffff64c8c864ff006400c800ffff00c8006400ff64c80064ffff64c8c864ffff6400c864ffff6400c800ffff00c80064ffff64c80064ffff64c80000ff]]></layer>
  <layer><![CDATA[ff0000c86400ff0000c864ff006400c800ffff000006400ff0000c864ff006400c800ffff00c8c864ffff64c8c864ff006400c800ffff00c8006400ff64c8c8
64ffff64c8c864ff006400c800ffff00c80064ffff64c8c864ffff64c8c864ff006400c864ff006400c864ff006400c800ffff00c8006400ff64c8c864ffff64c8c800ffff00c8006400ff64c80000
ff]]></layer>
  <layer><![CDATA[ff0000c864ffff6400c864ff006400c800ffff0000c86400ff6400c864ff006400c800ffff00c8006400ff64c8c864ffff64c8c864ff06400c800ffff00c8c86
400ff64c8c864ffff64c8c864ff006400c864ff006400c800ffff00c8006400ff64c8006400ff64c8c864ffff64c8c864ffff64c8c800ffff00c8006400ff64c8006400ff64c80000
ff]]></layer>
  <layer><![CDATA[ff0000006400ff6400c86400ff6400c864ff006400c864ff006400c800ffff0000c86400ff6400c864ff006400c864ff006400c864ff006400c800ffff00c8
006400ff64c8c864ffff64c8c864ff0064c8c864ff0064c8c800ffff00c8006400ff64c8c864ffff64c8c864ffff64c8c800ffff00c8006400ff64c8006400ff64c80000ff00c80
0000]]></layer>
  <layer><![CDATA[ff0000c86400ff6400c86400ff6400c86400006400c80000ff0000c86400ff6400c86400ff6400c86400ff6400c80000ff00c8006400ff64c8c86400ff64
c8c86400ff64c8c80000ff00c8006400ff64c8006400ff64c8000000]]></layer>
```



```
</link_map>  
</structure>  
</object>  
</fav>
```

Fig. 27 : XML code examples of overall FAV framework.