TMD: Modeling Data for OS Library

The TMD format contains 3D modeling data which is compatible with the PlayStation expanded graphics library (libgs). TMD data is downloaded to memory and may be passed as an argument to functions provided by LIBGS. TMD files are created using the RSDLINK utility, which reads an RSD file created by the SCE 3D Graphics Tool or a comparable program.

The data in a TMD file is a set of graphics primitives—polygons, lines, etc.—that make up a 3D object. A single TMD file can contain data for one or more 3D objects.

Coordinate Values

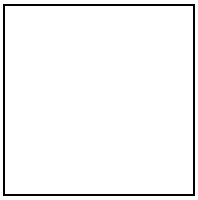
Coordinate values in the TMD file follow the 3D coordinate space handled by the 3D graphics library. The positive direction of the X axis represents the right, the Y axis the bottom, and the Z axis the depth. The spatial coordinate value of each object is a signed 16-bit integer value ranging from -32768 to +32767.

In the 3D object design phase and within the RSD format, the vertex information is stored as a floating point value. Conversion from RSD into TMD involves converting and scaling vertex values as needed. The scale used is reflected in the object structure, described later, as the reference value. This value can provide an index for mapping from object to world coordinates. The current version of LIBGS ignores the scale value.

File Format

TMD files are configured by 4 blocks. They have 3 dimensional object tables, and 3 types of data entities—PRIMITIVE, VERTEX, and NORMAL—which configure these.

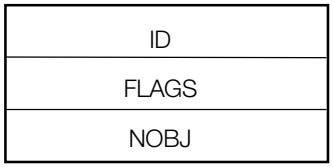
Figure 54: TMD File Format



Header

The header section is composed of three word (12 bytes) data carrying information on data structure.

Figure 55: Structure of Header



ID: Data having 32 bits (one word). Indicates the version of a TMD file. The current version is 0x00000041.

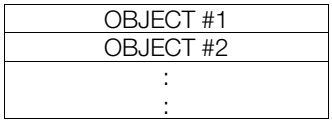
FLAGS: Data having 32 bits (one word). Carries information on TIM data configuration. The least significant bit is FIXP. The other bits are reserved and their values are all zero. The FIXP bit indicates whether the pointer value of the OBJECT structure described later is a real address. A value of one means a real address. A value of zero indicates the offset from the start.

NOBJ: Integral value indicating the number of objects

Obj Table

The OBJ TABLE block is a table of structures holding pointer information indicating where the substance of each object is stored. Its structure is as shown below.

Figure 56: OBJ TABLE structure



The object structure has the following configuration:

```
struct object
{
       u_long *vert_top;
       u_long n_vert;
       u_long *normal top;
       u_long n_normal;
       u_long *primitive top;
       u_long n_primitive;
       long scale;
}
```

(Explanation of members)

vert_top: Start address of a vertex
n_vert: Number of vertices
normal_top: Start address of a normal
n_normal: Number of normals
primitive_top: Start address of a primitive
n_primitive: Number of primitives

Among the members of the structure, the meanings of the pointer values (vert_top, normal_top, primitive_top) change according to the value of the FIXP bit in the HEADER section. If the FIXP bit is 1, they indicate the actual address, and if the FIXP bit is 0, they indicate a relative address taking the top of the OBJECT block as the 0 address.

The type of the scaling factor is "signed long", and its value raised to the second power is the scale value. That is to say, if the scaling factor is 0, the scale value is an equimultiple; if the scaling factor is 2, the scale value is 4; if the scaling factor is -1, the scale value is 1/2. Using this value, it is possible to return to the scale value at the time of design.

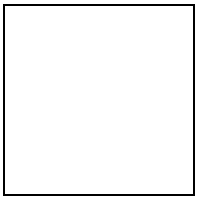
Primitive

The PRIMITIVE section is an arrangement of the drawing packets of the structural elements (primitives) of the object. One packet stands for one primitive (see Figure below).

The primitives defined in TMD are different from the drawing primitives handled by libgpu. A TMD primitive is converted to a drawing primitive by undergoing perspective transformation processing performed by the libgs functions.

Each packet is of variable length, and its size and structure vary according to the primitive type.

Figure 57: Drawing Packet General Structure

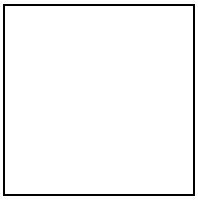


Each item in the figure above is as follows:

Mode (8 bit)

Mode indicates the type of primitive and added attributes. They have the following bit structure:

Figure 58: Mode



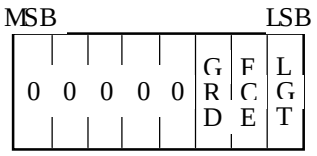
CODE: 3 bit code expressing entities
001 = Polygon (triangle, quadrilateral)
010 = Straight line
011 = Sprite

OPTION: Varies with the option, bit and CODE values
(Listed with the list of packet data configurations described later)

Flag (8 bit)

Flag indicates option information when rendering and has the following bit configuration:

Figure 59: Flag



GRD: Valid only for the polygon not textured, subjected to light source calculation
1: Gradation polygon
0: Single-color polygon

FCE: 1: Double-faced polygon
0: Single-faced polygon
(Valid, only when the CODE value refers to a polygon.)

LGT: 1: Light source calculation not carried out
0: Light source calculation carried out

Ilen (8 bit)

Indicates the length, in words, of the packet data section.

Olen (8 bit)

Indicates the word length of the 2D drawing primitives that are generated by intermediate processing.

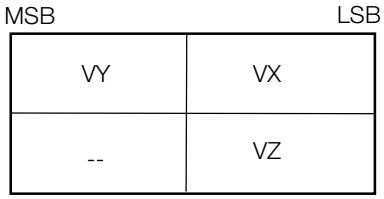
Packet Data

Parameters for vertices and normals. Content varies depending on type of primitive. Please refer to “Packet data configuration” which will be discussed later.

Vertex

The vertex section is composed of a set of structures representing vertices. The following gives the format of one structure.

Figure 60: Vertex Structure

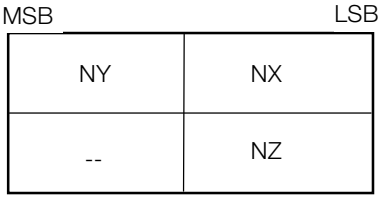


VX, VY, XZ: x, y and z values of vertex coordinates (16-bit integer)

Normal

The normal section is composed of a set of structures representing normals. The following gives the format of one structure.

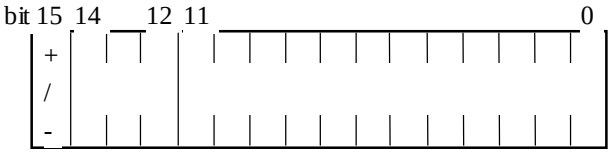
Figure 61: Normal Structure



NX, NY, NZ: x, y and z components of a normal (16-bit fixed-point value)

NX, NY and NZ values are signed 16-bit fixed-point values where 4096 is considered to be 1.0.

Figure 62: Fixed-Point Format



Sign: 1 bit

Integral part: 3 bits

Decimal part: 12 bits

Packet Data Composition Table

This section lists packet data configurations for each primitive type.

The following parameters are contained in the packet data section:

Vertex(n)

Index value of 16-bit length pointing to a vertex. Indicates the position of the element from the start of the vertex section for an object covering the polygon.

Normal(n)

Index value of 16-bit length pointing to a normal. Same as Vertex.

Un, Vn

X and Y coordinate values on the texture source space for each vertex

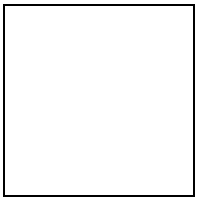
Rn, Gn, Bn

RGB value representing polygon color being an unsigned 8-bit integer. Without light source calculation, the predetermined brightness value must be entered.

TSB

Carries information on a texture/sprite pattern.

Figure 63: TSB



TPAGE: Texture page number (0 to 31)

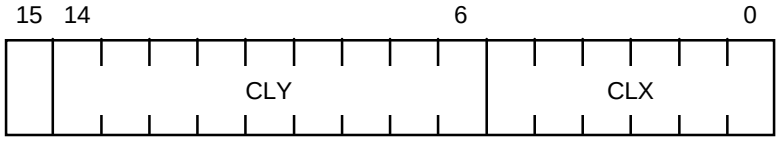
ABR: Semi-transparency rate (Mixture rate).
Valid, only when ABE is 1.
00 50%back + 50%polygon
01 100%back + 100%polygon
10 100%back - 100%polygon
11 100%back + 25%polygon

TPF: Color mode
00 4 bit
01 8 bit
10 15 bit

CBA

Indicates the position where CLUT is stored in the VRAM.

Figure 64: CBA



CLX: Upper six bits of 10 bits of X coordinate value for CLUT on the VRAM

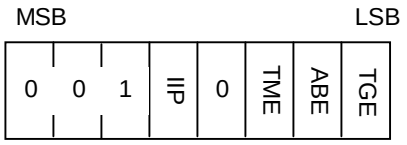
CLY: Nine bits of Y coordinate value for CLUT on the VRAM

3 Vertex Polygon with Light Source Calculation

Bit Configuration of Mode Value

The primitive section mode value bit configuration is shown below. For the value of each bit please refer to "3 vertex polygon with light source calculation."

Figure 65: Mode Value of 3 Vertex Polygon with Light Source Calculation



IIP: Shading mode
0: Flat shading
1: Gouraud shading

TME: Texture specification
0: Off
1: On

ABE: Translucency processing
0: Off
1: On

TGE: Brightness calculation at time of texture mapping
0: On
1: Off (Draws texture as is)

Packet Configuration

Figure 66: Packet Configuration of 3 Vertex Polygon with Light Source Calculation

Flat, No-Texture (solid)

0x20	0x00	0x03	0x04
0x20*	B	G	R
Vertex 0		Normal 0	
Vertex 2		Vertex 1	

Gouraud, No-Texture (solid)

0x30	0x00	0x04	0x06
0x30*	B	G	R
Vertex 0		Normal 0	
Vertex 1		Normal 1	
Vertex 2		Normal 2	

Flat, No-Texture (gradation)

0x20	0x04	0x05	0x06
0x20*	B0	G0	R0
—	B1	G1	R1
—	B2	G2	R2
Vertex 0		Normal 0	
Vertex 2		Vertex 1	

Gouraud, No-Texture (gradation)

0x30	0x04	0x06	0x06
0x30*	B0	G0	R0
—	B1	G1	R1
—	B2	G2	R2
Vertex 0		Normal 0	
Vertex 1		Normal 1	
Vertex 2		Normal 2	

Flat, Texture

0x24	0x00	0x05	0x07
CBA		V0	U0
TSB		V1	U1
—	—	V2	U2
Vertex 0		Normal 0	
Vertex 2		Vertex 1	

Gouraud, Texture

0x34	0x00	0x06	0x09
CBA		V0	U0
TSB		V1	U1
—	—	V2	U2
Vertex 0		Normal 0	
Vertex 1		Normal 1	
Vertex 2		Normal 2	

* same value as mode

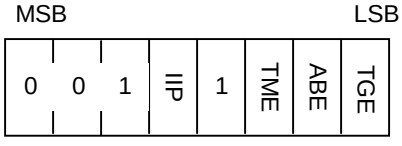
In the above example, the values of mode and flag indicate a single-faced polygon and semi-transparency processing not carried out.

4 Vertex polygon with Light Source Calculation

Bit Configuration of Mode Value

The primitive section mode value bit configuration is shown below. For the value of each bit please refer to “3 vertex polygon with light source calculation.”

Figure 67: Mode Value of 4 Vertex Polygon with Light Source Calculation



(bit 3 is set to designate a 4-vertex primitive)

Packet Configuration

Figure 68: Packet Configuration for 4 Vertex Polygon with Light Source Calculation

Flat, No-Texture (solid)

0x28	0x00	0x04	0x05
0x28*	B	G	R
Vertex 0		Normal 0	
Vertex 2		Vertex 1	
——		Vertex 3	

Gouraud, No-Texture (solid)

0x38	0x00	0x05	0x08
0x38*	B	G	R
Vertex 0		Normal 0	
Vertex 1		Normal 1	
Vertex 2		Normal 2	
Vertex 3		Normal 3	

Flat, No-Texture (gradation)

0x28	0x04	0x07	0x08
0x28*	B0	G0	R0
—	B1	G1	R1
—	B2	G2	R2
—	B3	G3	R3
Vertex 0		Normal 0	
Vertex 2		Vertex 1	
——		Vertex 3	

Gouraud, No-Texture (gradation)

0x38	0x04	0x08	0x08
0x38*	B0	G0	R0
—	B1	G1	R1
—	B2	G2	R2
—	B3	G3	R3
Vertex 0		Normal 0	
Vertex 1		Normal 1	
Vertex 2		Normal 2	
Vertex 3		Normal 3	

Flat, Texture

0x2c	0x00	0x07	0x09
CBA		V0	U0
TSB		V1	U1
—	—	V2	U2
—	—	V3	U3
Vertex 0		Normal 0	
Vertex 2		Vertex 1	
——		Vertex 3	

Gouraud, Texture

0x3c	0x00	0x08	0x0c
CBA		V0	U0
TSB		V1	U1
—	—	V2	U2
—	—	V3	U3
Vertex 0		Normal 0	
Vertex 1		Normal 1	
Vertex 2		Normal 2	
Vertex 3		Normal 3	

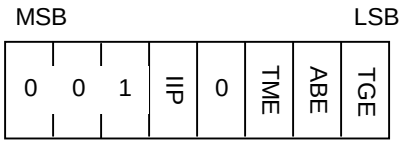
* same value as mode

3 Vertex Polygon with No Light Source Calculation

Bit Configuration of Mode Value

The primitive section mode value bit configuration is shown below. For the value of each bit please refer to “3 vertex polygon with light source calculation.”

Figure 69: Mode Value of 3 Vertex Polygon with No Light Source Calculation



(bit 3 is set to designate a 4-vertex primitive)

Packet Configuration

Figure 70: Packet configuration for 3 Vertex Polygon with No Light Source Calculation

Flat, No-Texture

0x21	0x01	0x03	0x04
0x21*	B	G	R
Vertex 1		Vertex 0	
——		Vertex 2	

Gradation, No-Texture

0x31	0x01	0x05	0x06
0x31*	B0	G0	R0
—	B1	G1	R1
—	B2	G2	R2
Vertex 1		Vertex 0	
——		Vertex 2	

Flat, Texture

0x25	0x01	0x06	0x07
CBA		V0	U0
TSB		V1	U1
—	—	V2	U2
—	B	G	R
Vertex 1		Vertex 0	
——		Vertex 2	

Gradation, Texture

0x35	0x01	0x08	0x09
CBA		V0	U0
TSB		V1	U1
—	—	V2	U2
—	B0	G0	R0
—	B1	G1	R1
—	B2	G2	R2
Vertex 1		Vertex 0	
——		Vertex 2	

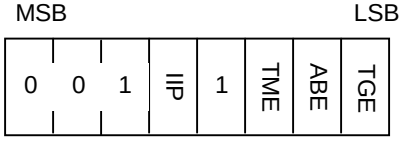
* same value as mode

4 Vertex Polygon with No Light Source Calculation

Bit Configuration of Mode Value

The primitive section mode value bit configuration is shown below. For the value of each bit please refer to “3 vertex polygon with light source calculation.”

Figure 71: Mode Value of 4 Vertex Polygon with No Light Source Calculation



(bit 3 is set to designate a 4-vertex primitive)

Packet Configuration

Figure 72: Packet Configuration for 4 Vertex Polygon with No Light Source Calculation

Flat, No-Texture

0x29	0x01	0x03	0x05
0x29*	B	G	R
Vertex 1		Vertex 0	
Vertex 3		Vertex 2	

Gradation, No-Texture

0x39	0x01	0x06	0x08
0x39*	B0	G0	R0
—	B1	G1	R1
—	B2	G2	R2
—	B3	G3	R3
Vertex 1		Vertex 0	
Vertex 3		Vertex 2	

Flat, Texture

0x2d	0x01	0x07	0x09
CBA		V0	U0
TSB		V1	U1
—	—	V2	U2
—	—	V3	U3
—	B	G	R
Vertex 1		Vertex 0	
Vertex 3		Vertex 2	

Gradation, Texture

0x3d	0x01	0x0a	0x0c
CBA		V0	U0
TSB		V1	U1
—	—	V2	U2
—	—	V3	U3
—	B0	G0	R0
—	B1	G1	R1
—	B2	G2	R2
—	B3	G3	R3
Vertex 1		Vertex 0	
Vertex 3		Vertex 2	

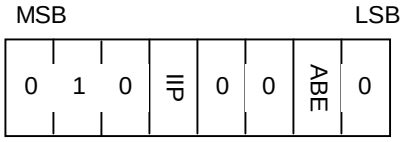
* same value as mode

Straight Line

Bit Configuration of Mode Value

The primitive section mode value bit configuration is as follows:

Figure 73: Mode Value of Straight Line



IIP: With or without gradation
0: Gradation off (Monochrome)
1: Gradation on

ABE: Translucency processing on/off
0: off
1: on

Packet Configuration

Figure 74: Packet Configuration for “Straight Line”

Gradation OFF

0x40	0x01	0x02	0x03
0x40*	B	G	R
Vertex 1		Vertex 0	

Gradation ON

0x50	0x01	0x03	0x04
0x50*	B0	G0	R0
—	B1	G1	R1
Vertex 1		Vertex 0	

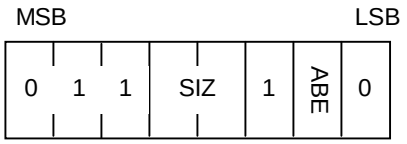
* same value as mode

3 Dimensional Sprite

A 3 dimensional sprite is a sprite with 3-D coordinates and the drawing content is the same as a normal sprite.

Bit Configuration of Mode Value

Figure 75: Mode Value of 3D Sprite



SIZ: Sprite size
00: Free size (Specified by W, H)
01: 1 x 1
10: 8 x 8
11: 16 x 16

ABE: Translucency processing
0: Off
1: On

Packet Data Configuration

Figure 76: Packet Configuration for Sprites