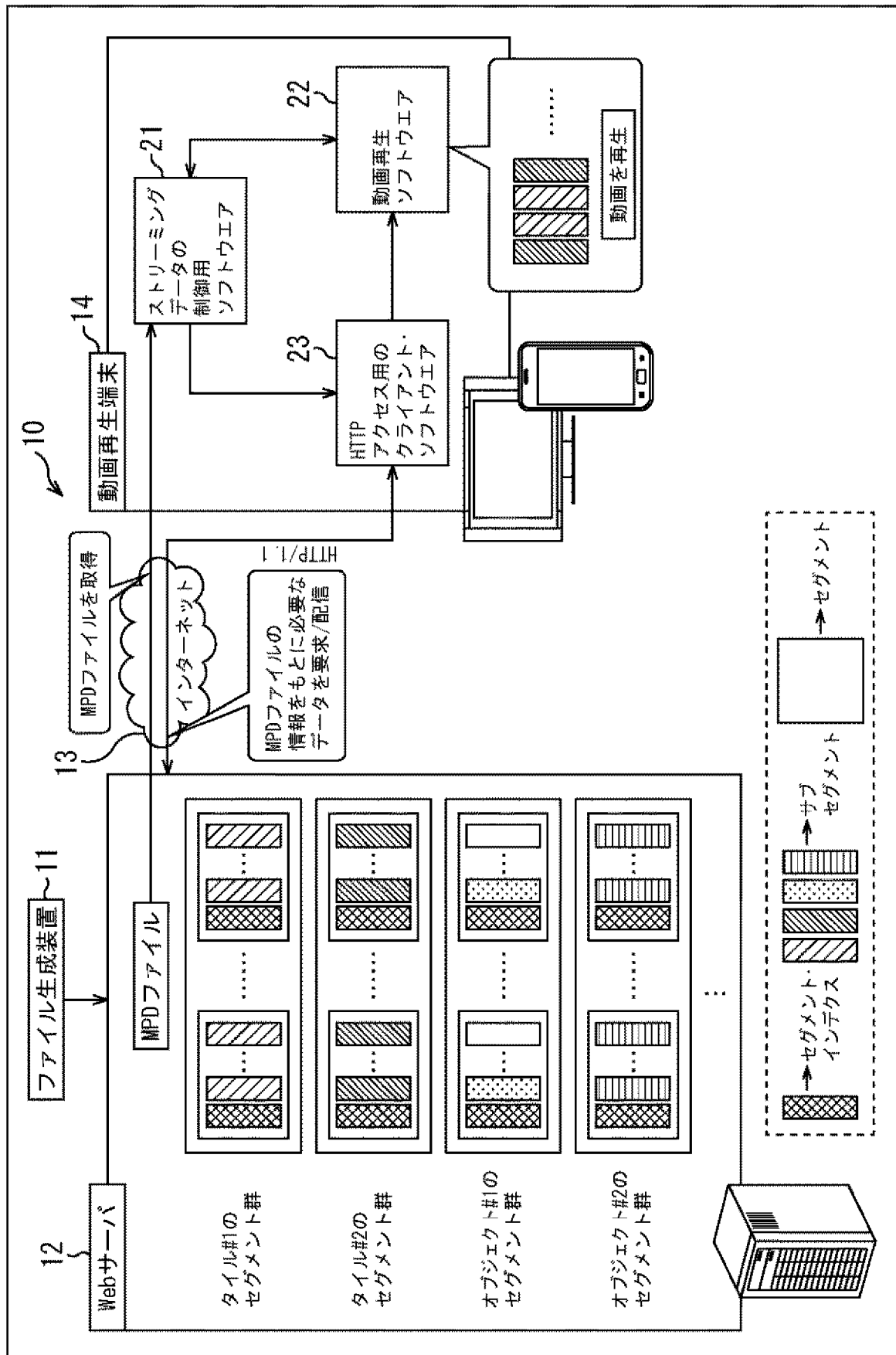


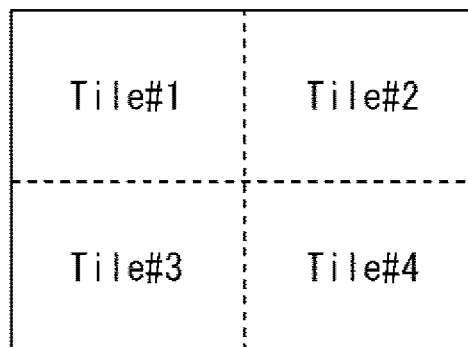
[図1]

図1



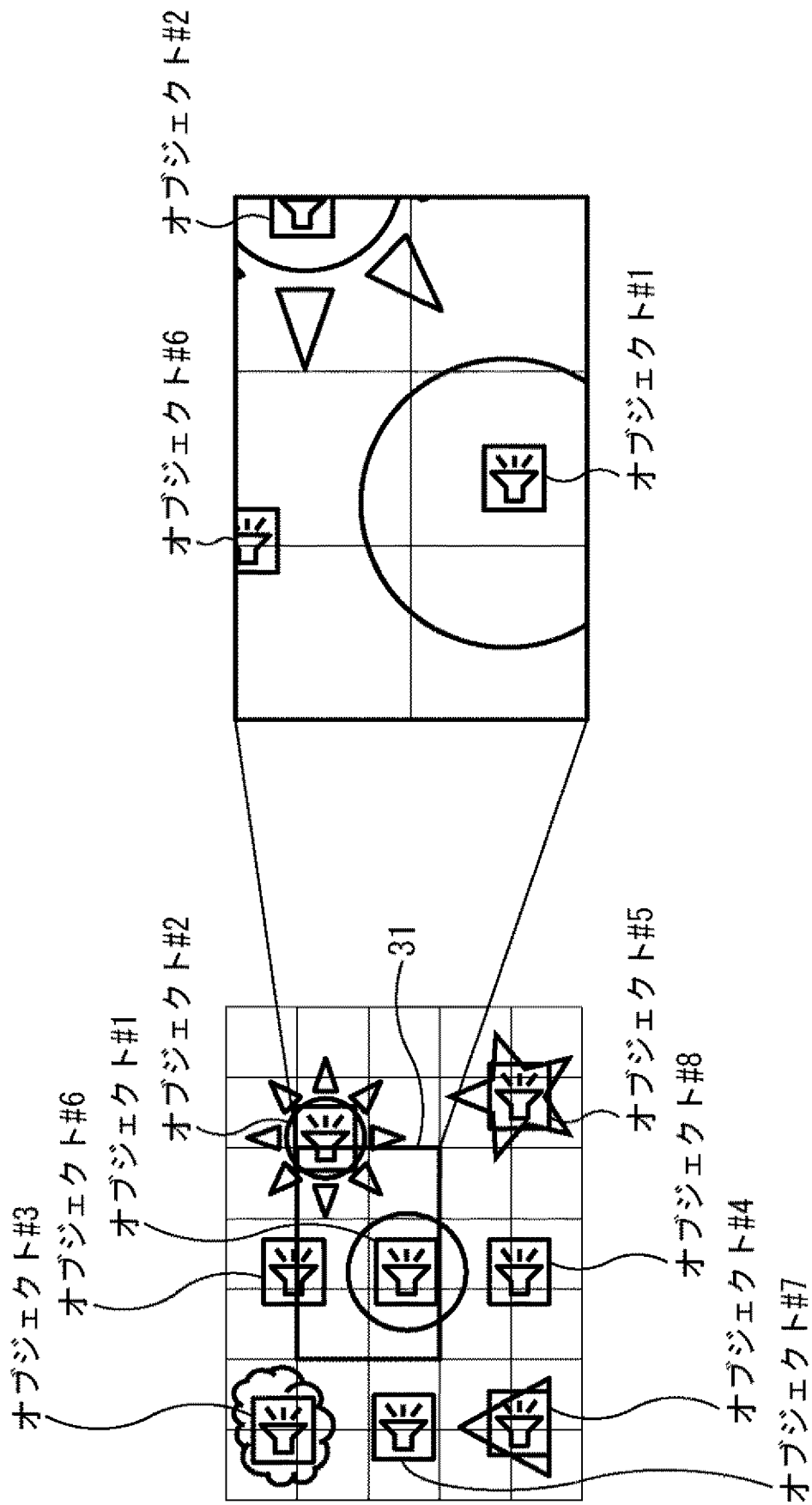
[図2]

図2



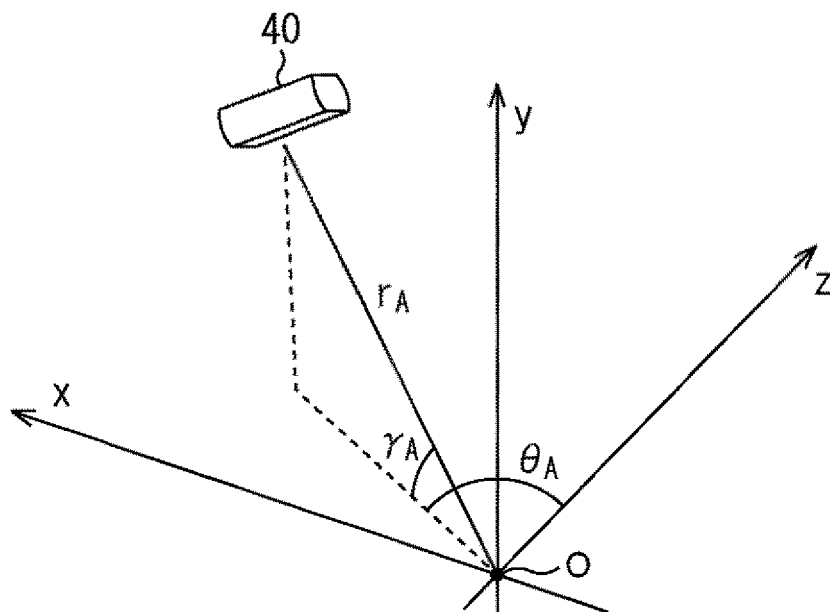
[図3]

図3



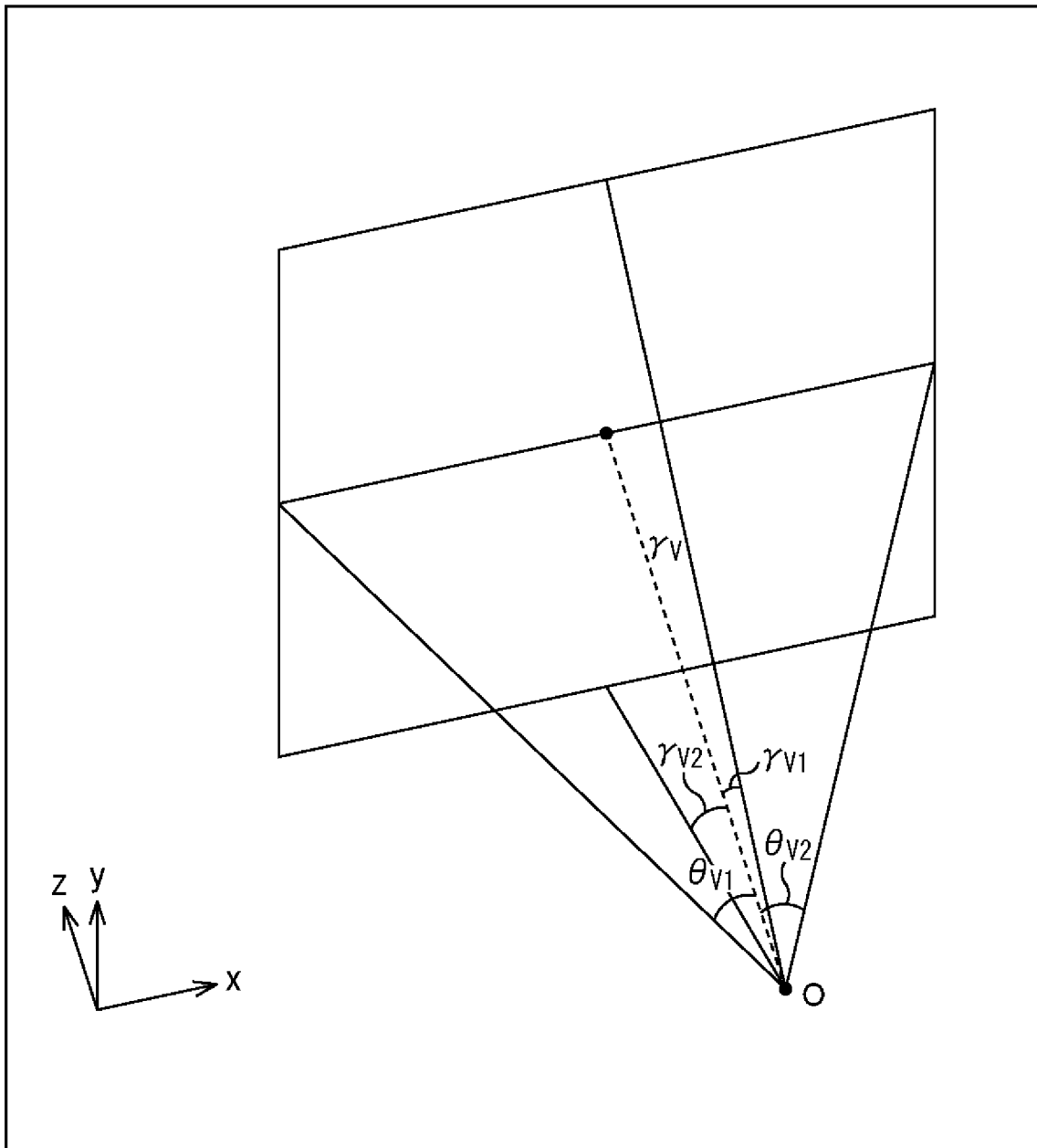
[図4]

図4

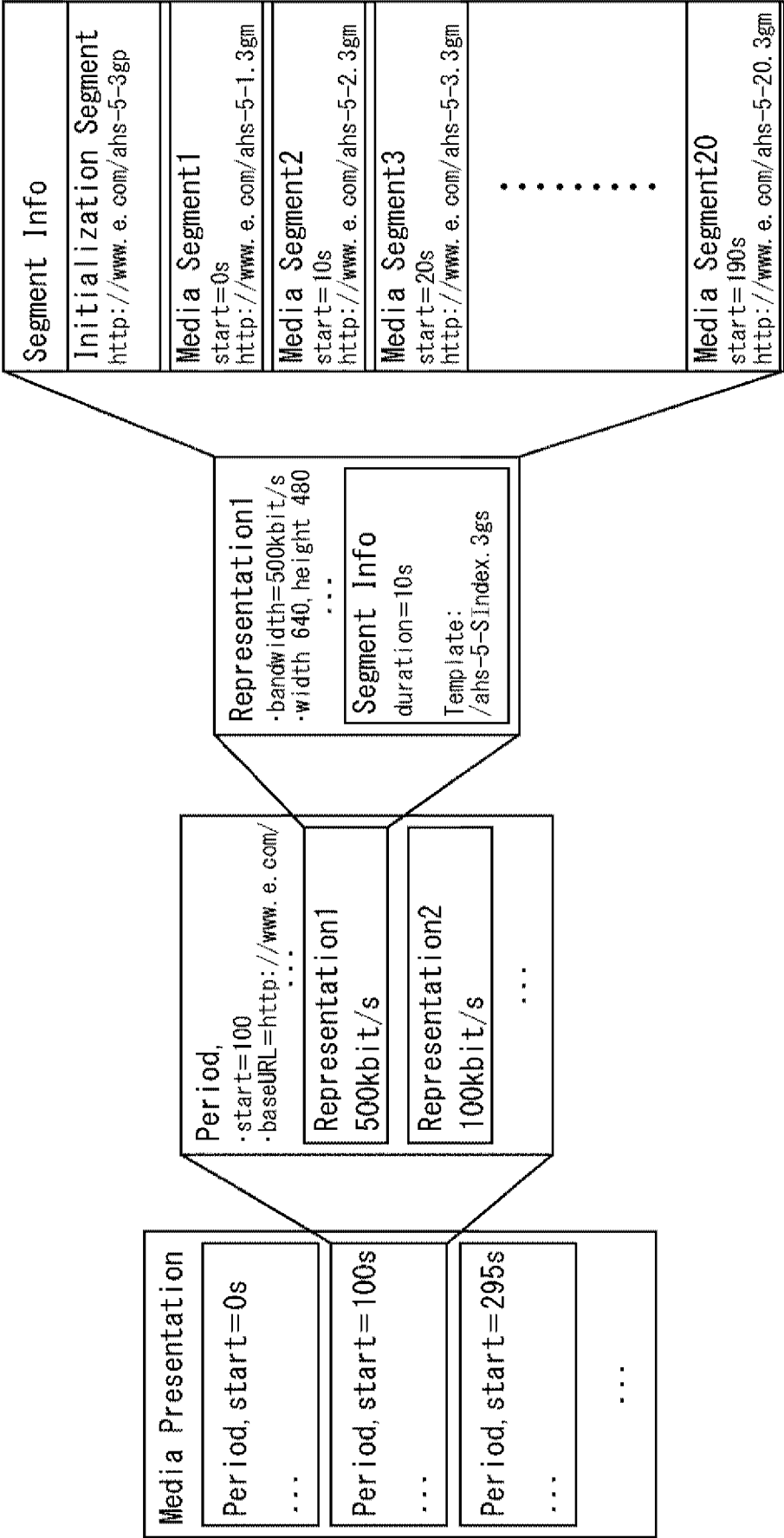


[図5]

図5

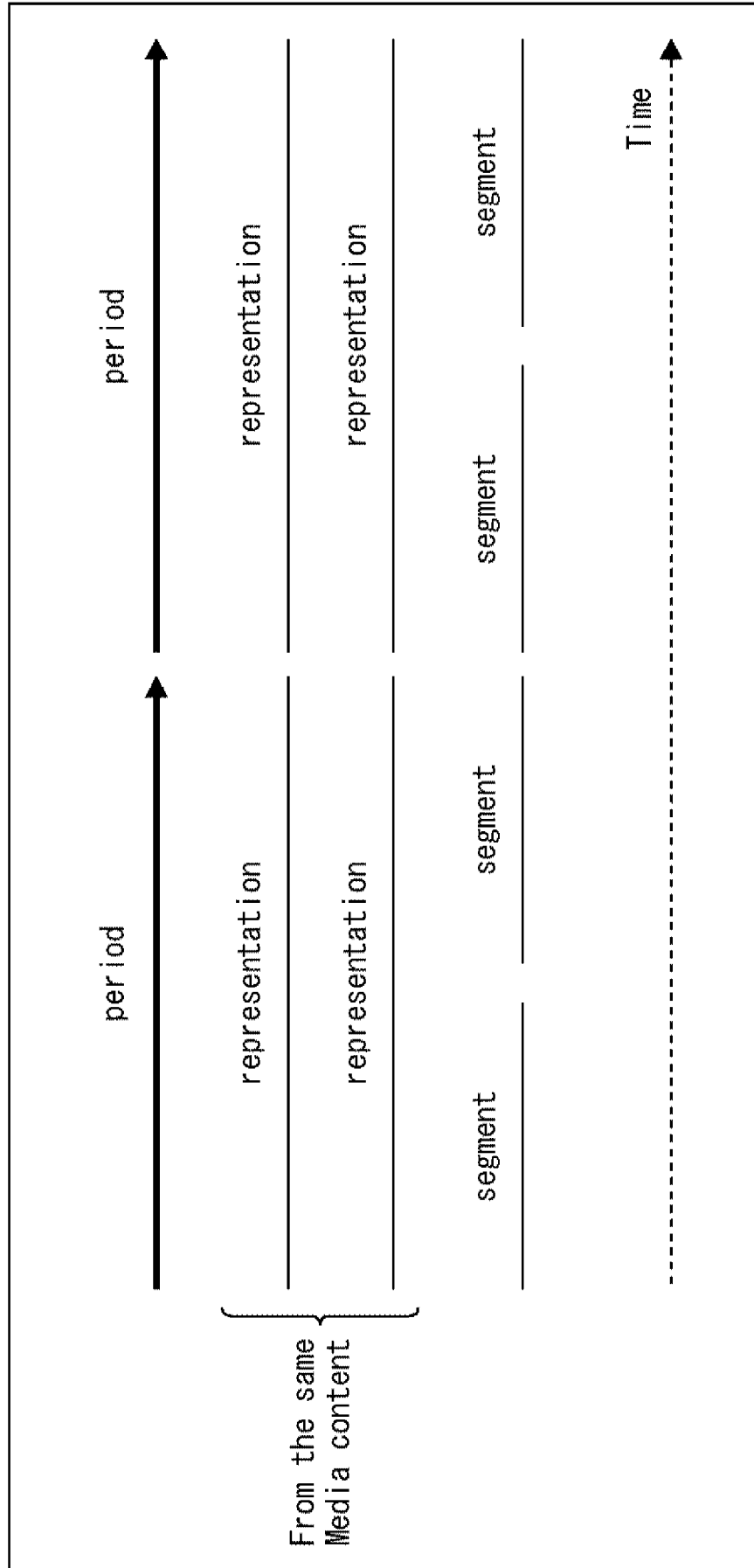


[図6]
図6



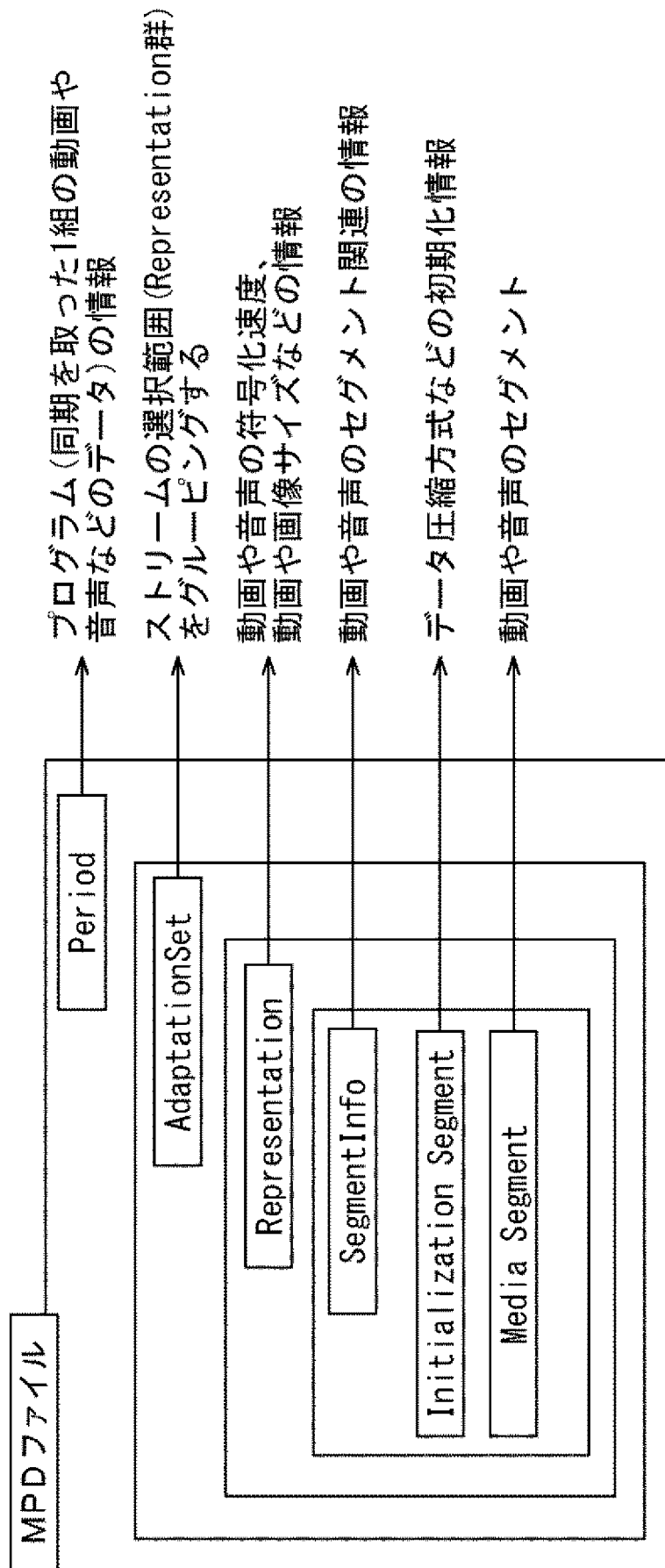
[図7]

図7

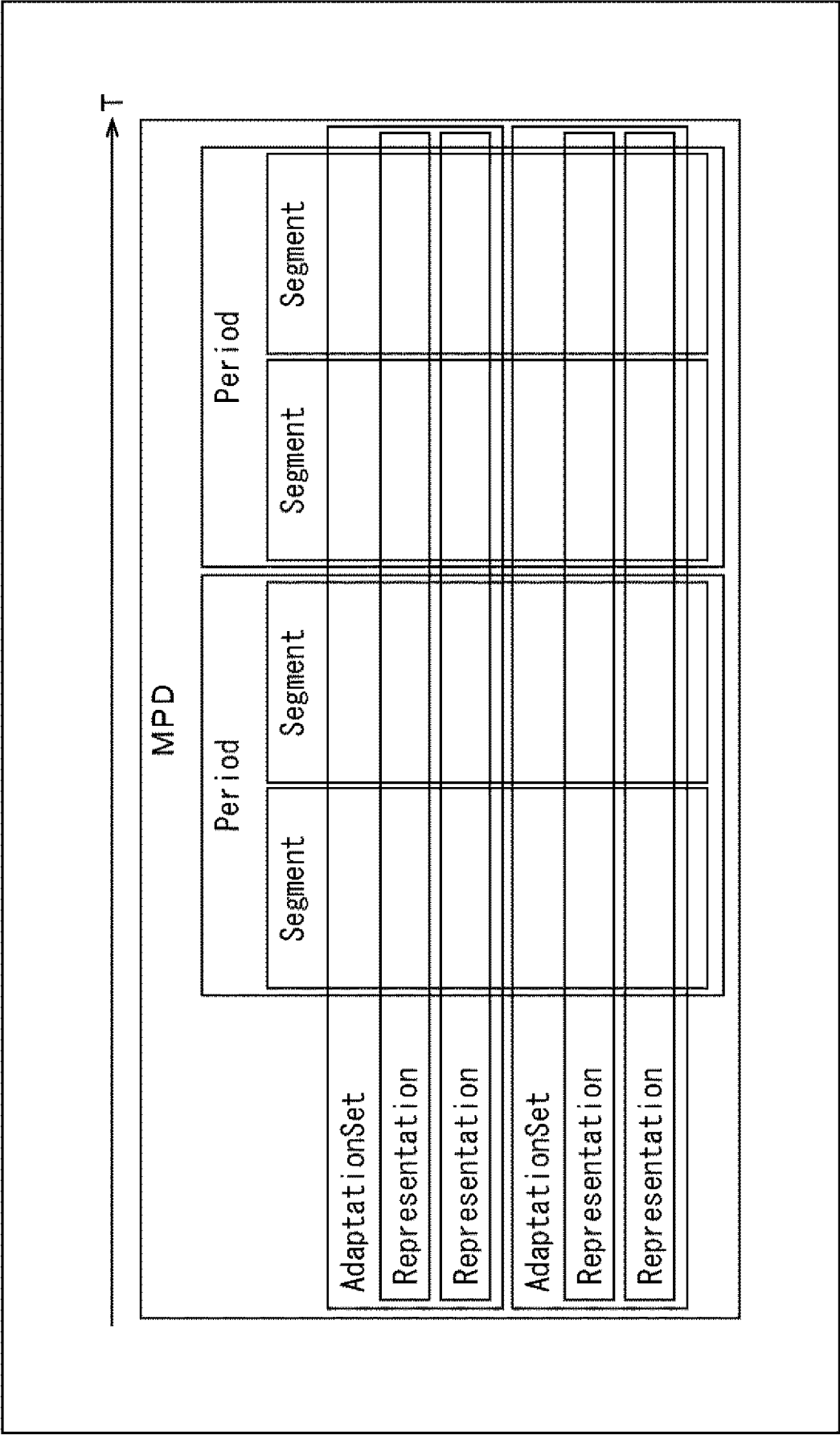


[図8]

図8

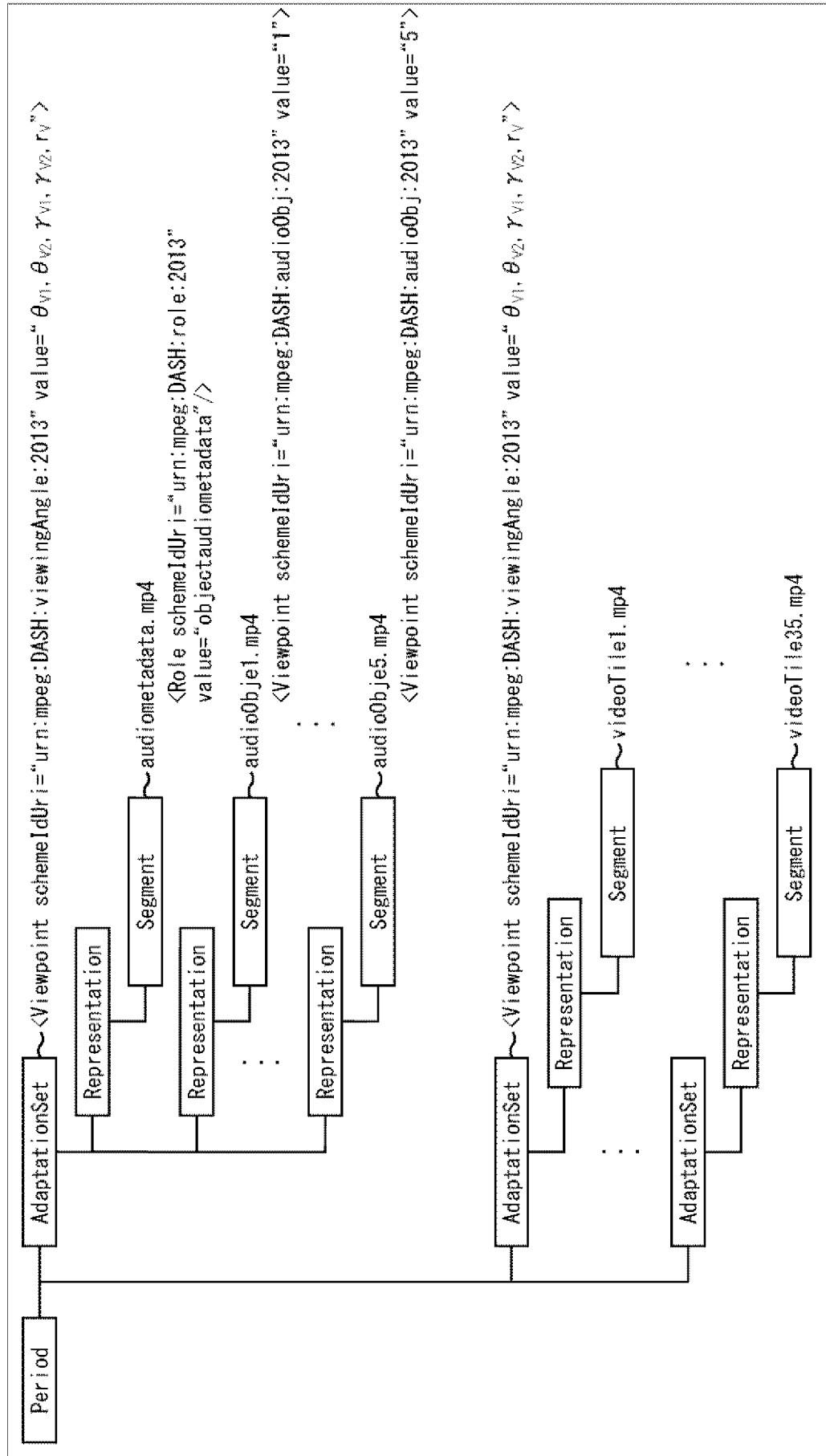


[図9]
[図9]



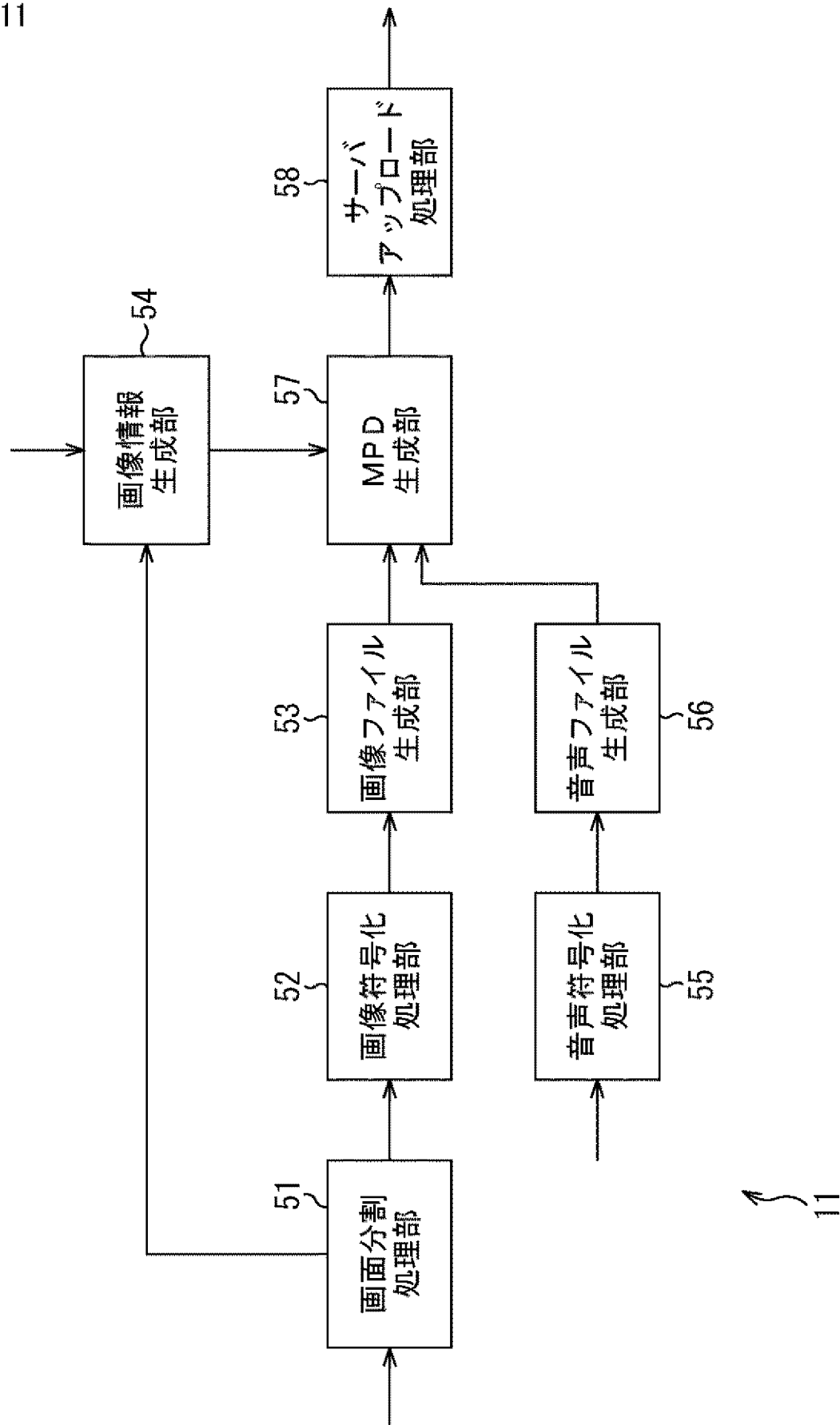
[図10]

[図10]



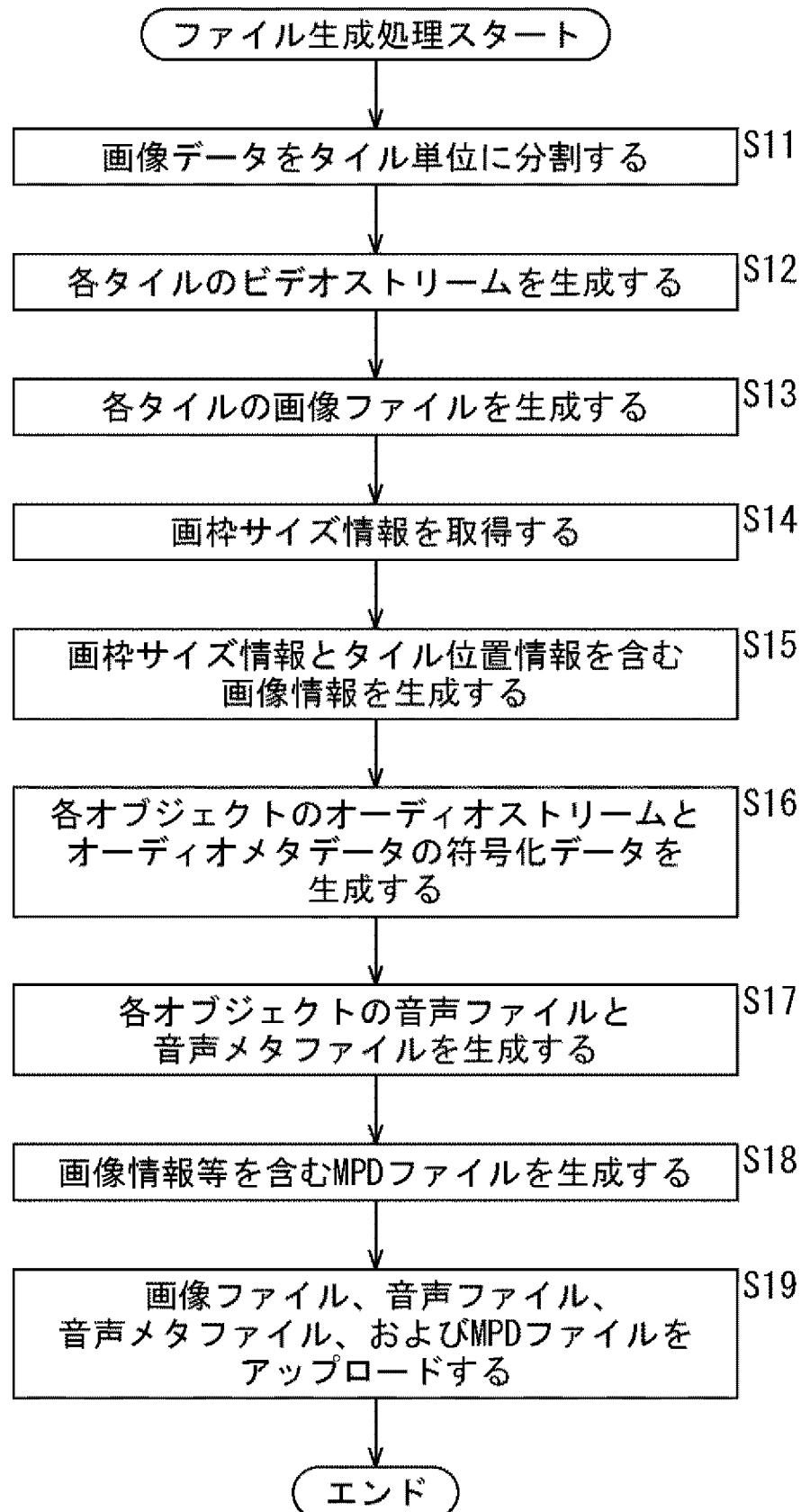
[図11]

図11



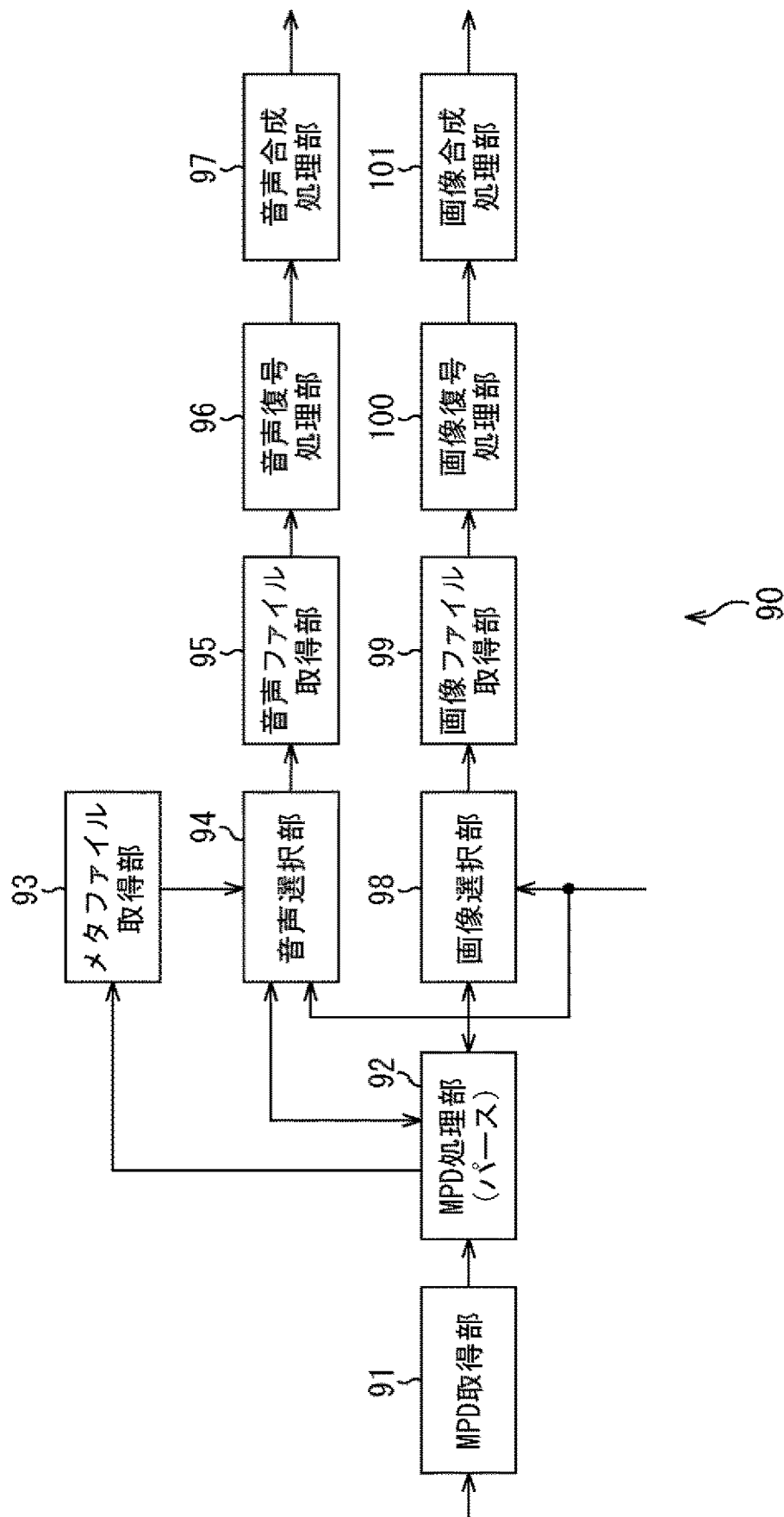
[図12]

図12



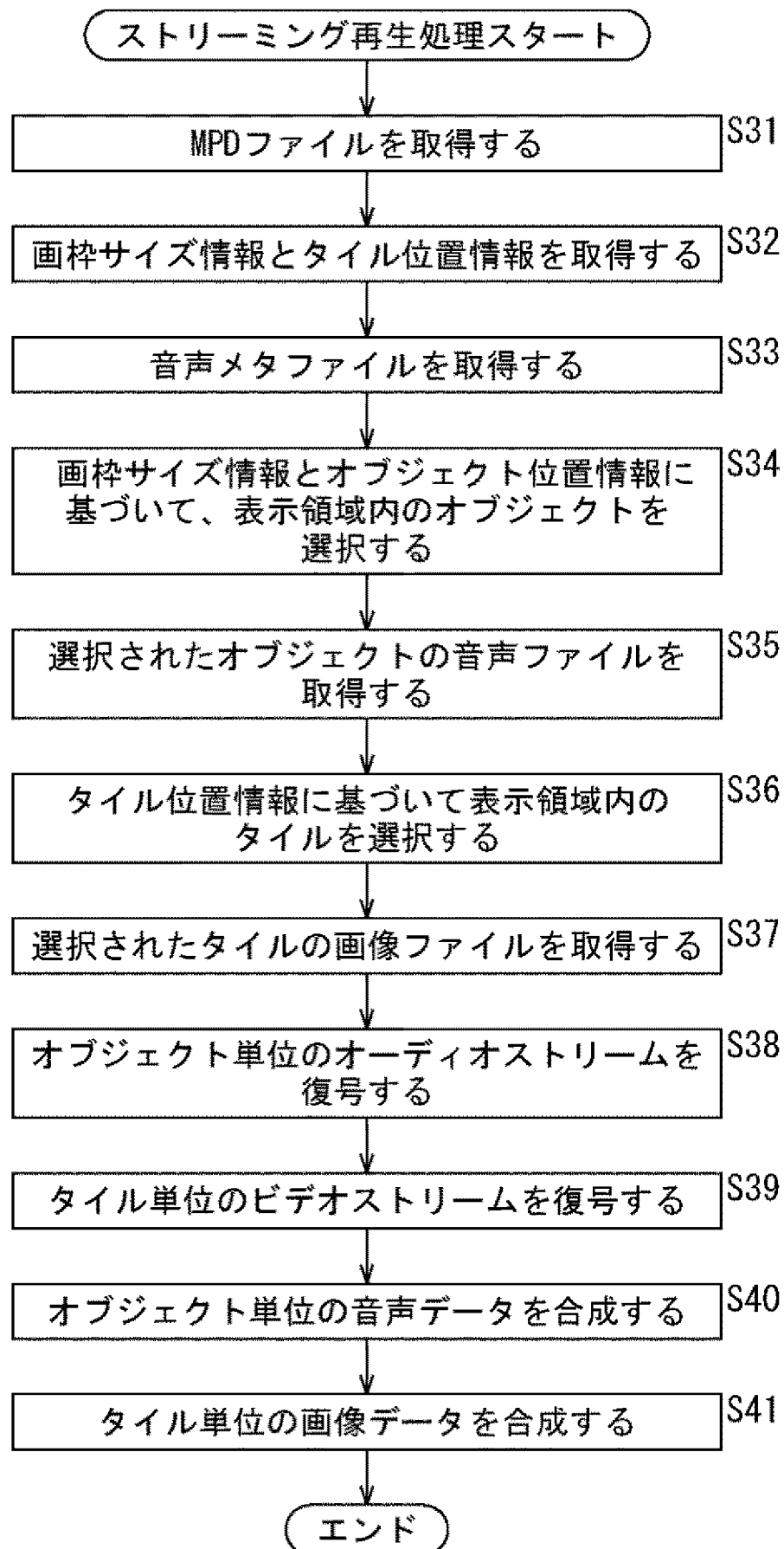
[図13]

図13



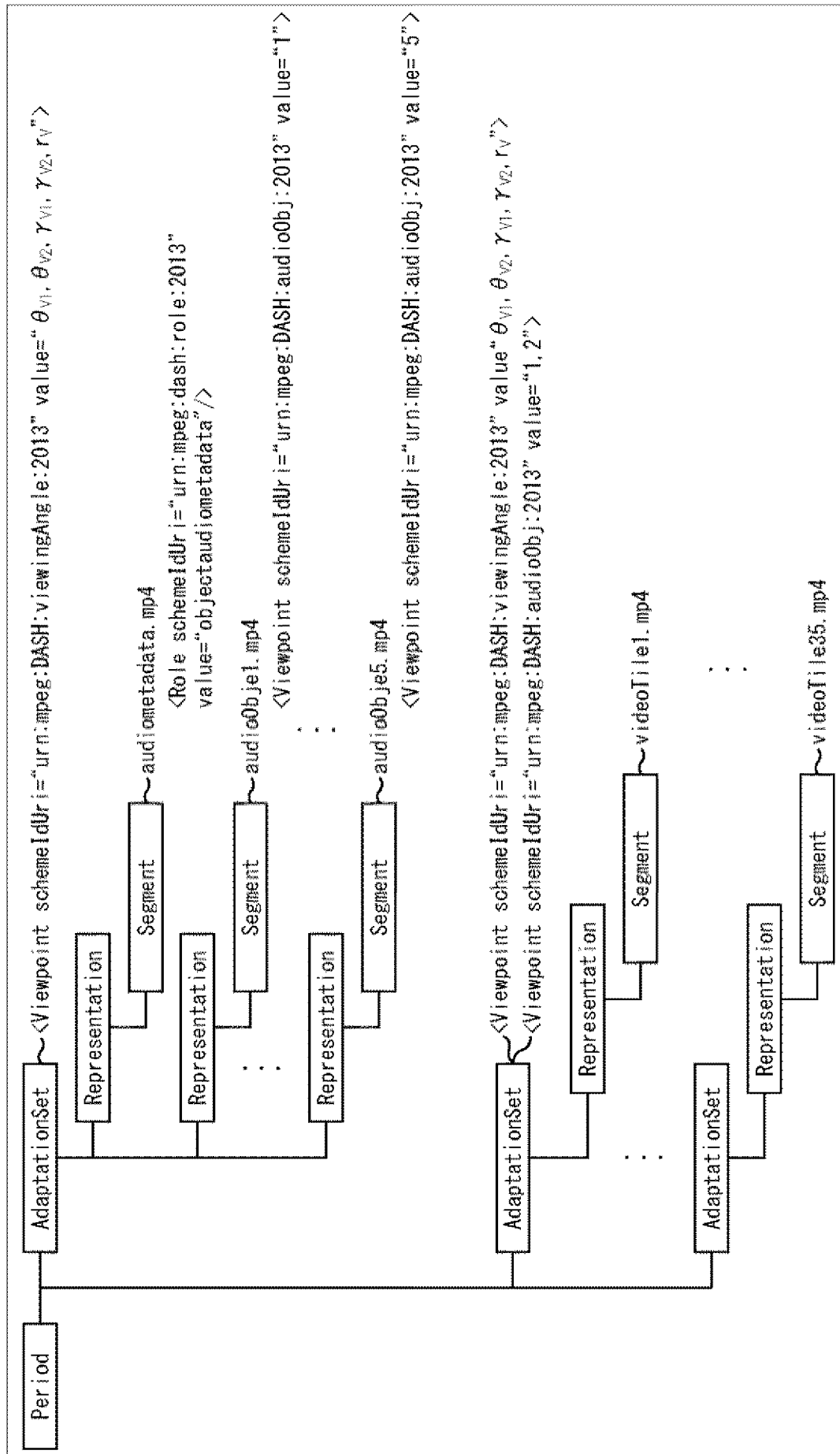
[図14]

図14



[図15]

図15



[図16]
図16

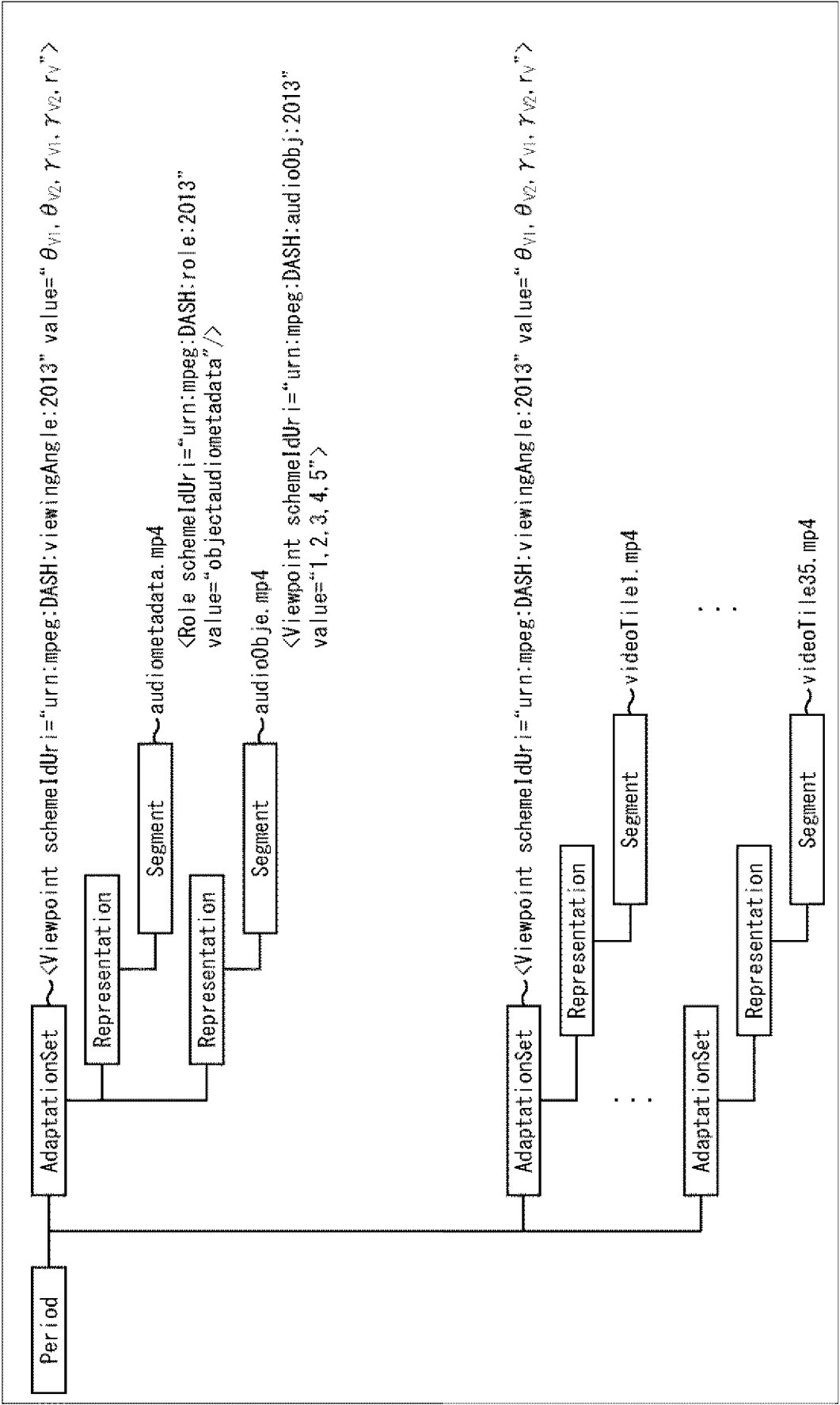
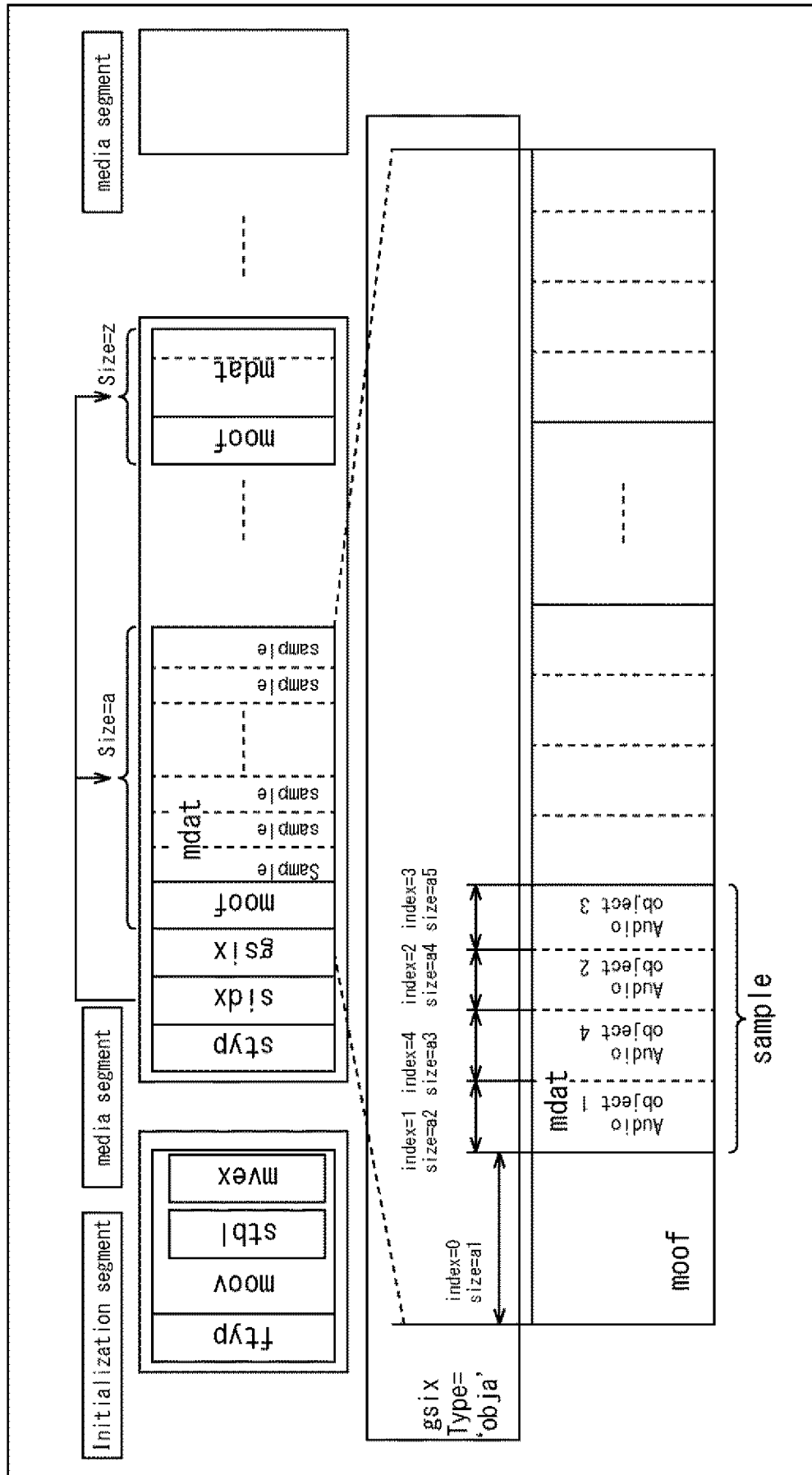


图 17



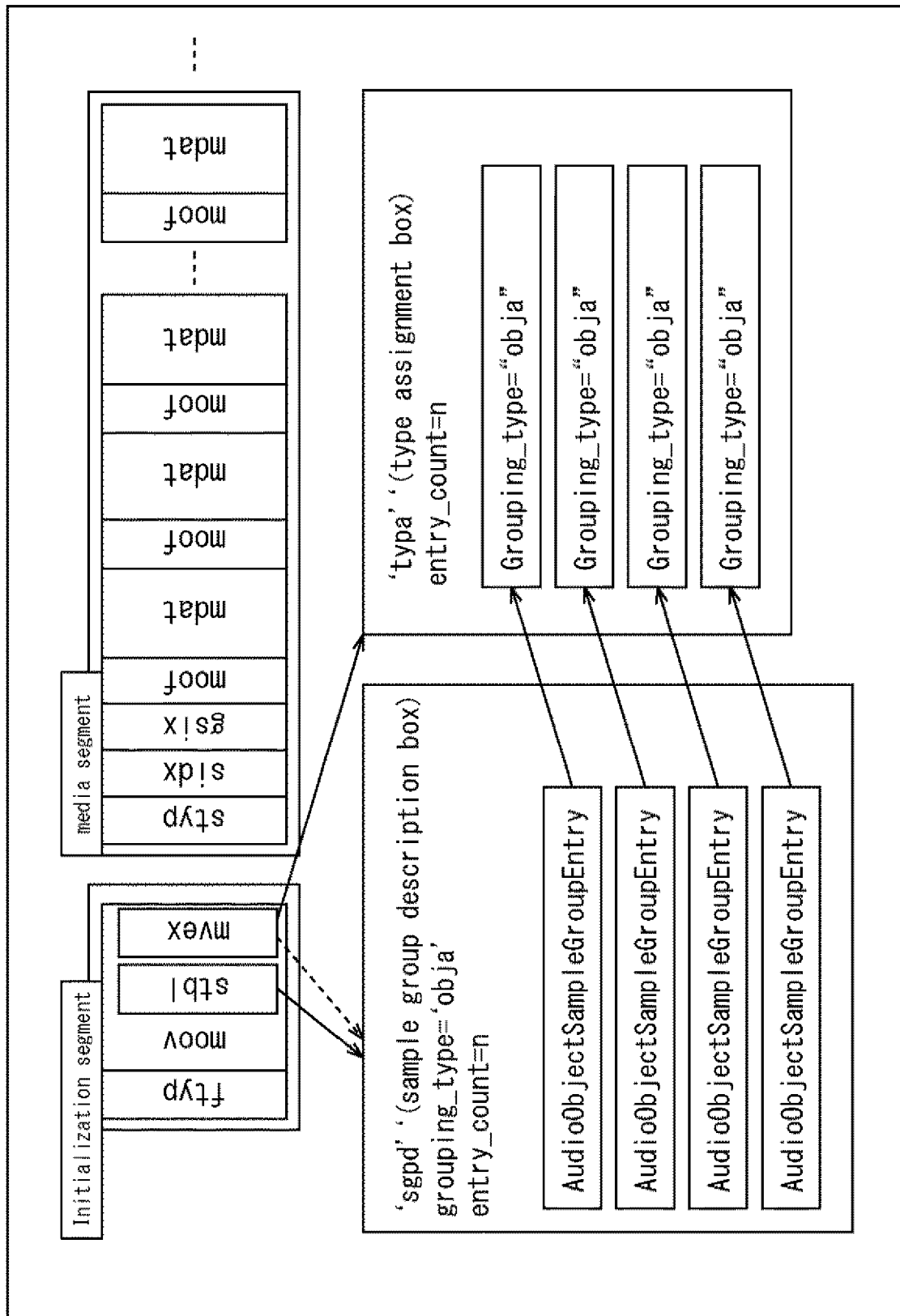
[図18]

図18

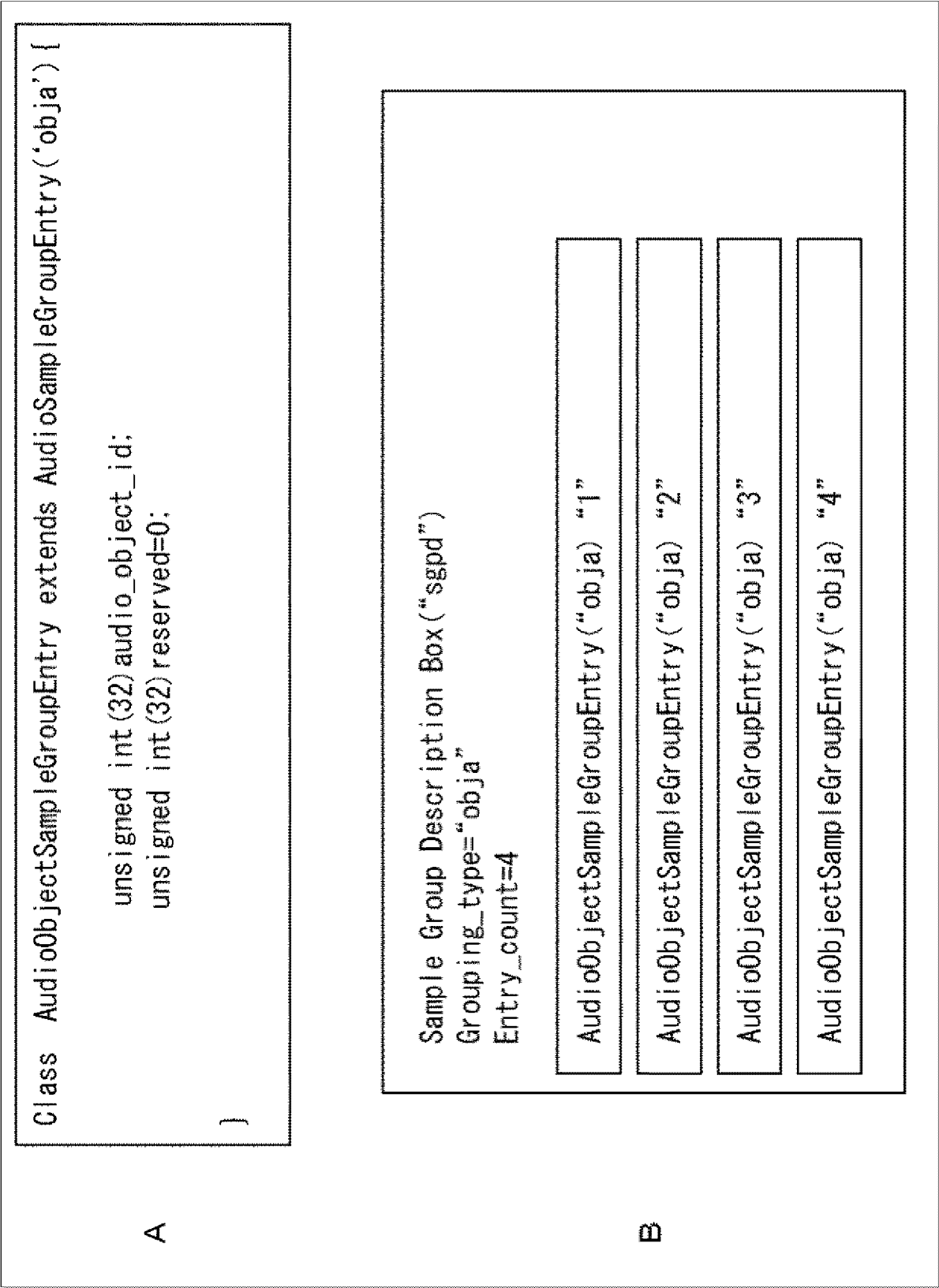
```
aligned(8) class GeneralSubsegmentIndexBox extends FullBox('gsix', 0, 0) {
    grouping_type int(32);
    unsigned int(32) subsegment_count;
    for (i=1; i<=subsegment_count; i++)
    {
        unsigned int(32) entry_count;
        for (j=1; j<= entry_count; j++) {
            unsigned int(8) entry_index;
            unsigned int(24) range_size;
        }
    }
}
```

[図19]

図19



[図20]
図20



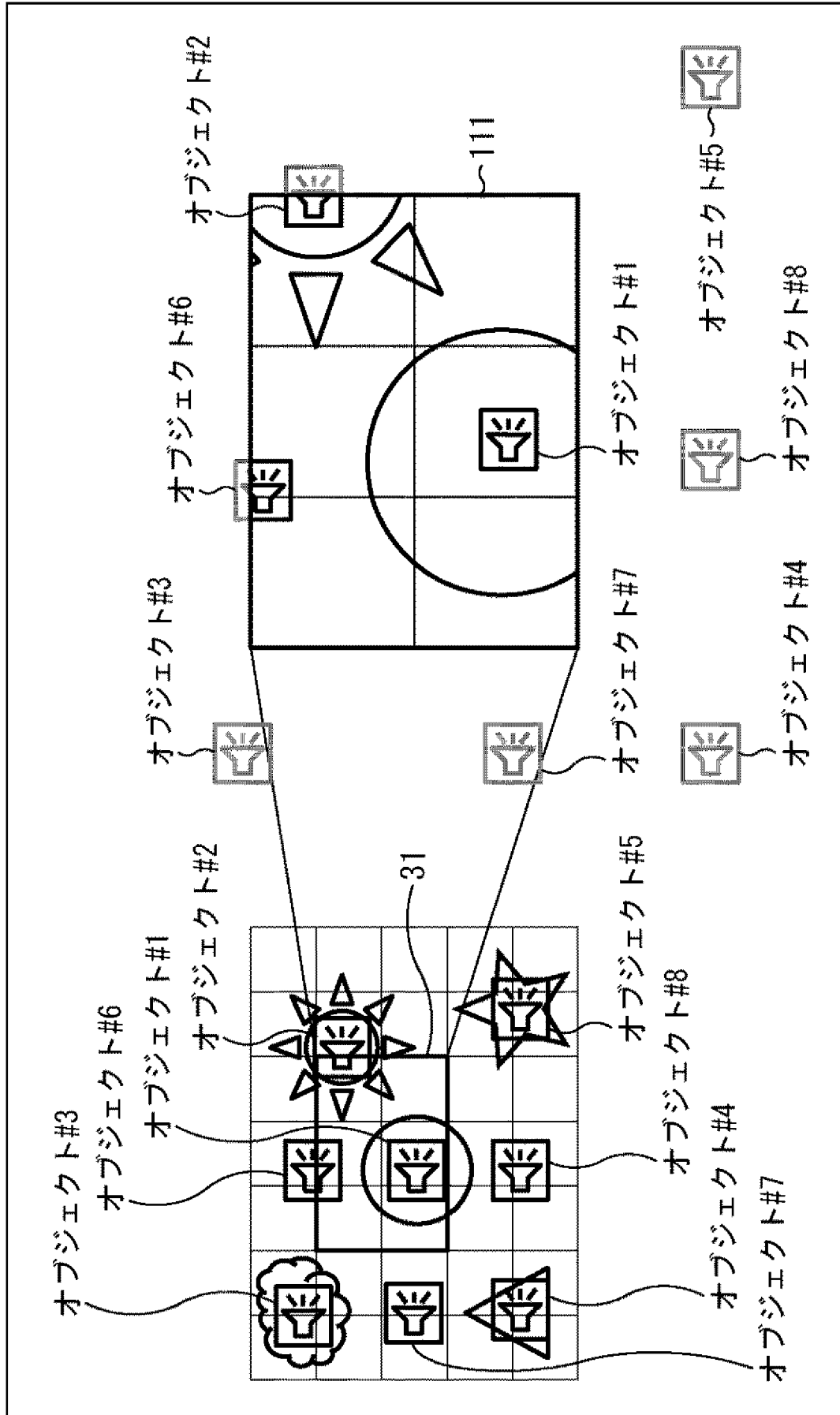
[図21]

図21

```
aligned(8) class TypeAssignmentBox extends FullBox('typa', 0, 0)
{
    unsigned int(8) entry_count;
    for (j=1; j<= entry_count; j++) {
        unsigned int(32) track_id;
        unsigned int(7) assignment_type;
        if(assignment_type == 0) {
            unsigned int(32) grouping_type;
        }
        else if (assignment_type == 1) {
            unsigned int(32) grouping_type;
            unsigned int(32) grouping_type_parameter;
        }
    }
    // other assignment_type values are reserved
}
```

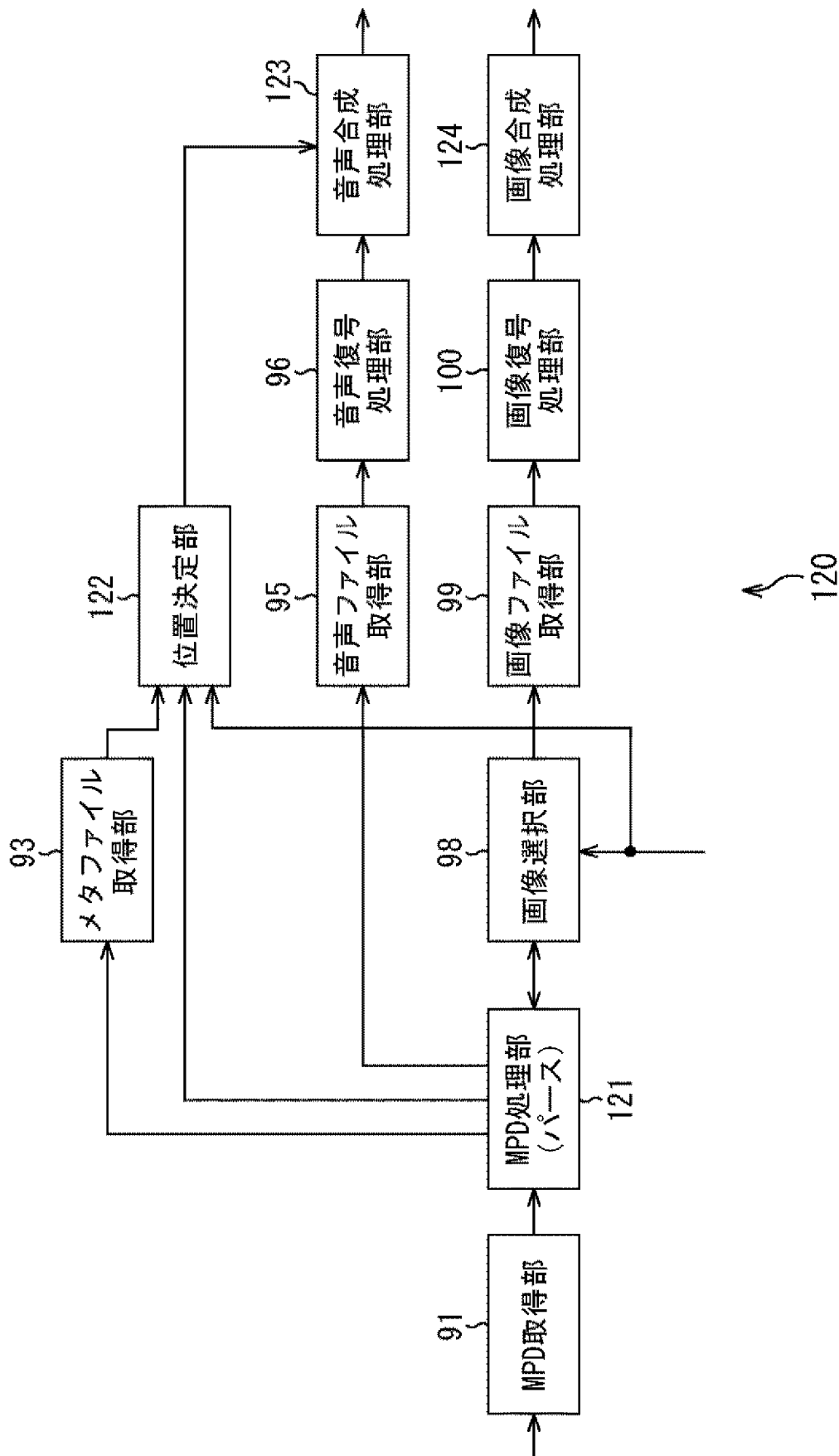
[図22]

図22



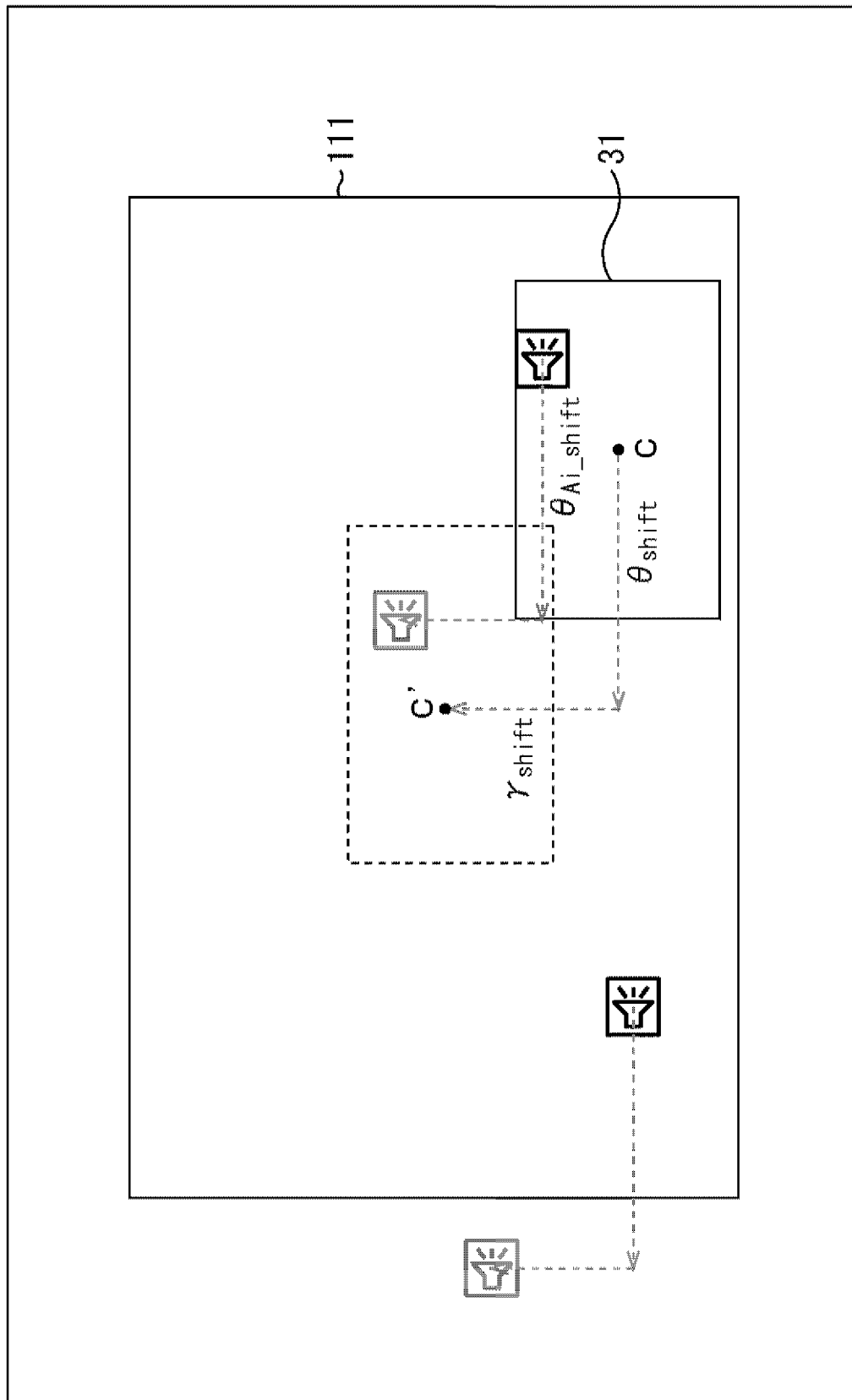
[図23]

図23



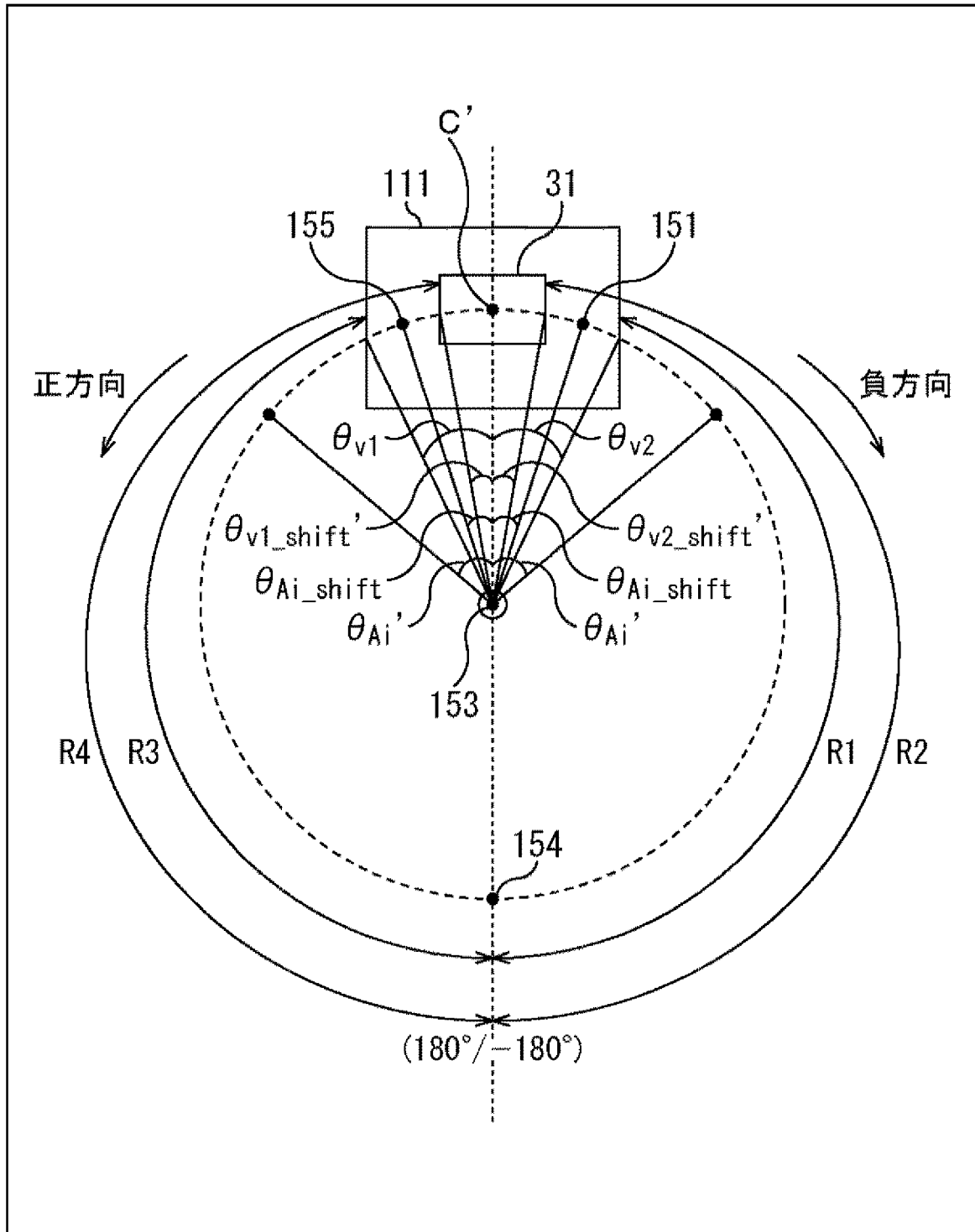
[図24]

图 24



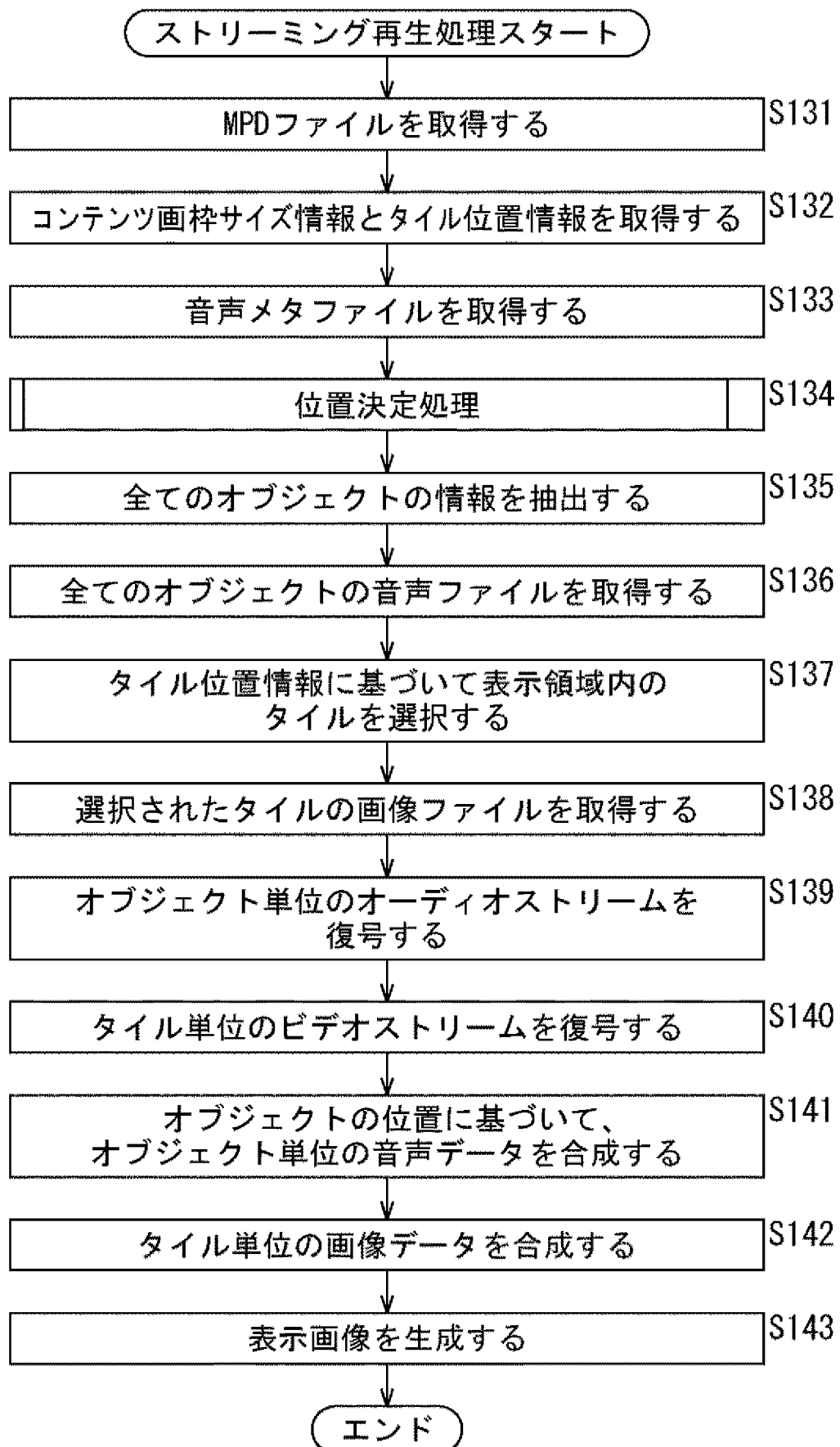
[図26]

図26



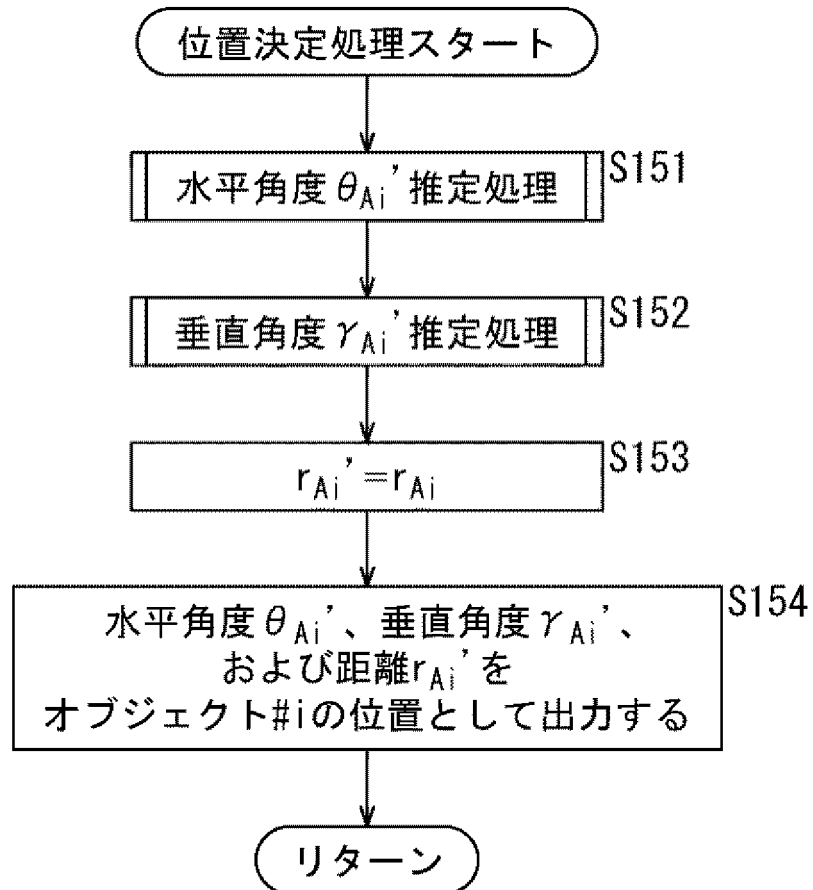
[図28]

図28



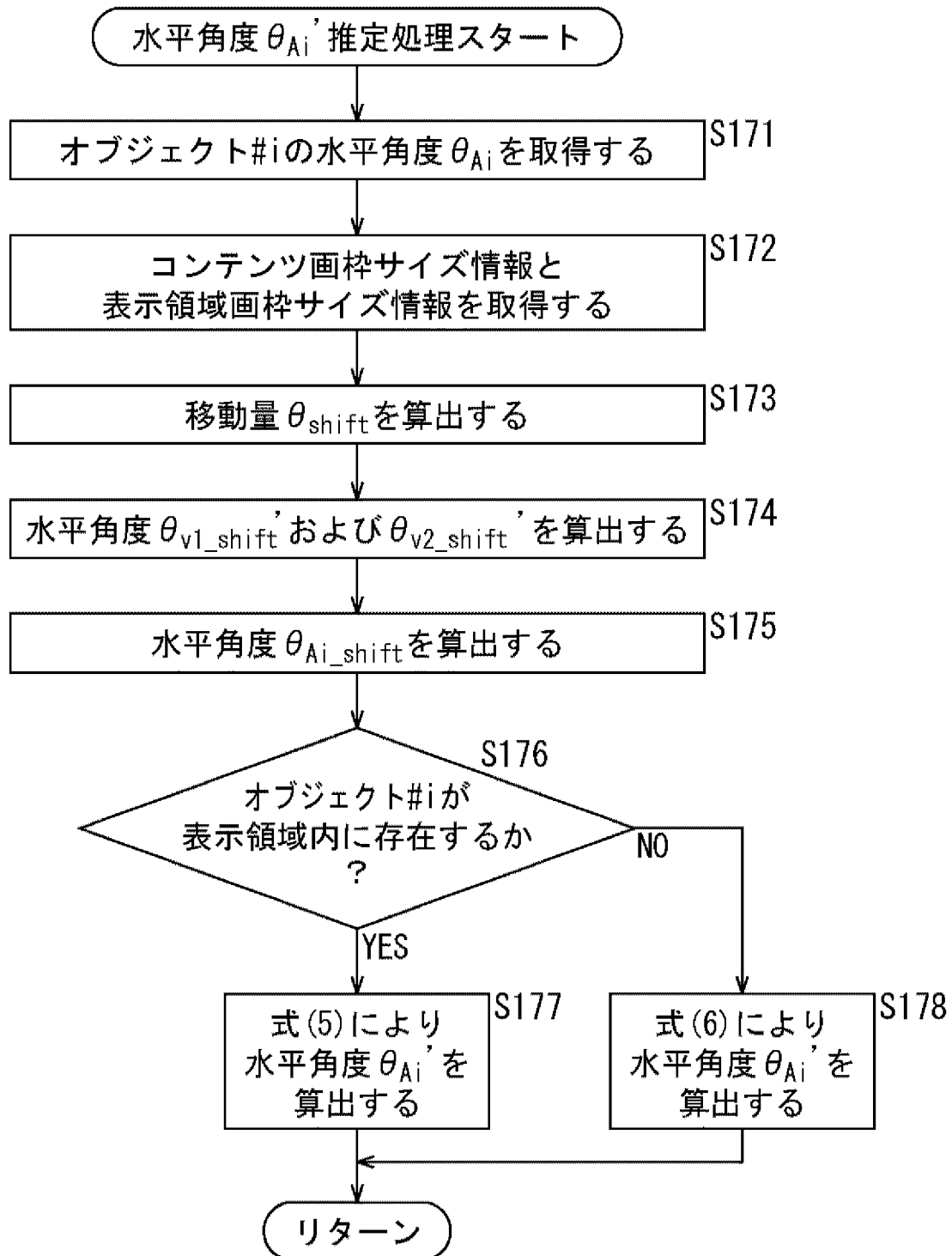
[図29]

図29



[図30]

図30



[図31]

図31

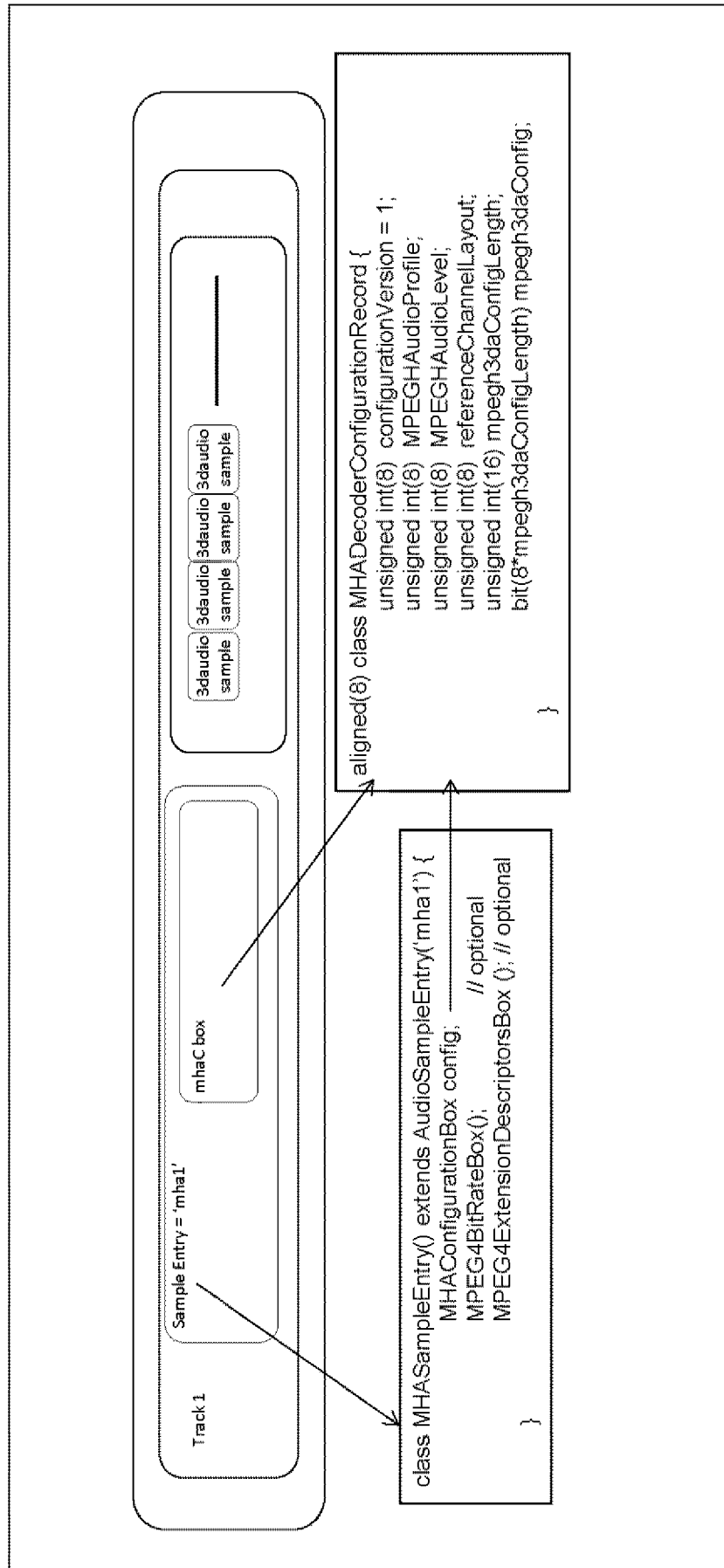
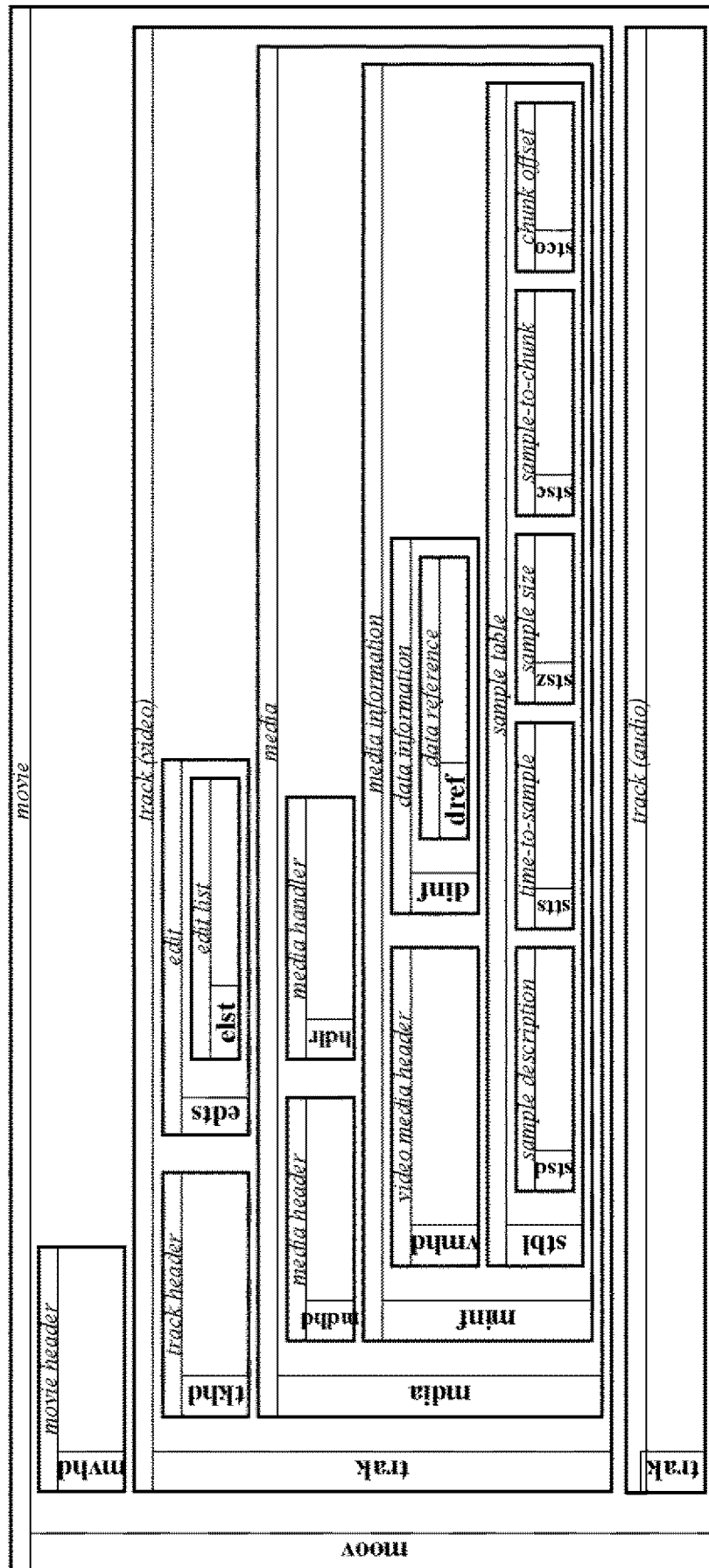
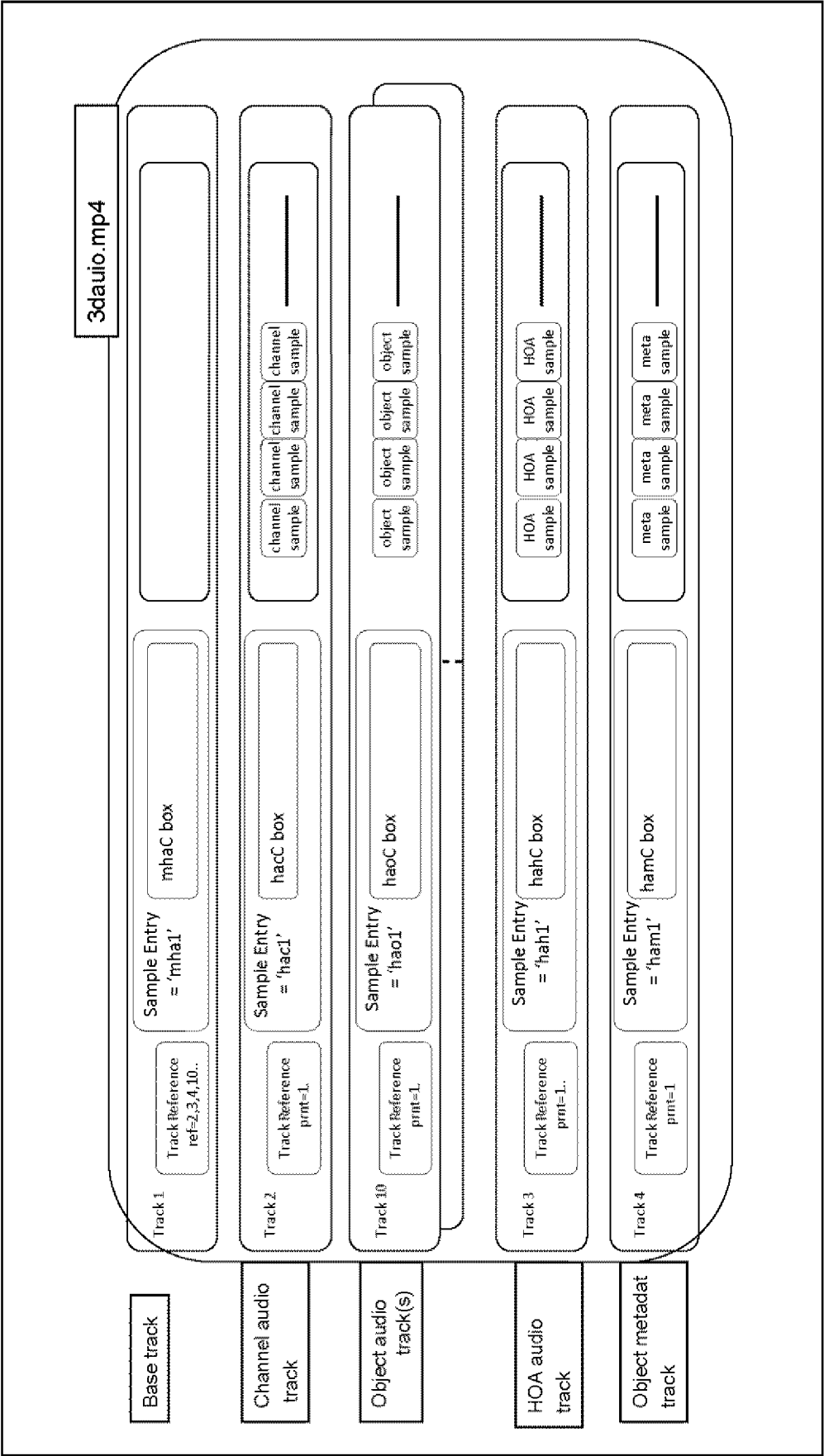


图32



[図33]
図33



[図34]
図34



[図35]

図35

```

class MHACChannelAudioSampleEntry() extends AudioSampleEntry('hac1') {
    MHACChannelAudioConfigurationBox config;           // optional
    MPEG4BitRateBox();                                 // optional
    MPEG4ExtensionDescriptorsBox ();
}

class MHACChannelAudioConfigurationBox() extends Box('hacC') {
    MHACChannelAudioDecoderConfigurationRecord() MHACACConfig;
}

aligned(8) class MHACChannelAudioDecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    unsigned int(8) MPEG4AudioProfile;
    unsigned int(8) MPEG4AudioLevel;
}

```

Channel Audioのdecodeに必要なProfile/Level情報

[図37]

図37

```

class MHAHOAAudioSampleEntry() extends AudioSampleEntry('hah1') {
    MHAHOAAudioConfigurationBox config;           // optional
    MPEG4BitRateBox();                             // optional
    MPEG4ExtensionDescriptorsBox ();               // optional
}

class MHAHOAAudioConfigurationBox() extends Box('hahC') {
    MHAHOAAudioDecoderConfigurationRecord() MHAHACConfig;
}

aligned(8) class MHAHOAAudioDecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    unsigned int(8) MPEG4AudioProfile;
    unsigned int(8) MPEG4AudioLevel;
}

```

このtrackに格納されている
HOA audioに必要な
Profile/Level情報

[図38]

図38

```

class MHAMetadataSampleEntry() extends AudioSampleEntry('ham1') {
    MHAMetadataConfigurationBox config;           // optional
    MPEG4BitRateBox();                             // optional
    MPEG4ExtensionDescriptorsBox ();              // optional
}

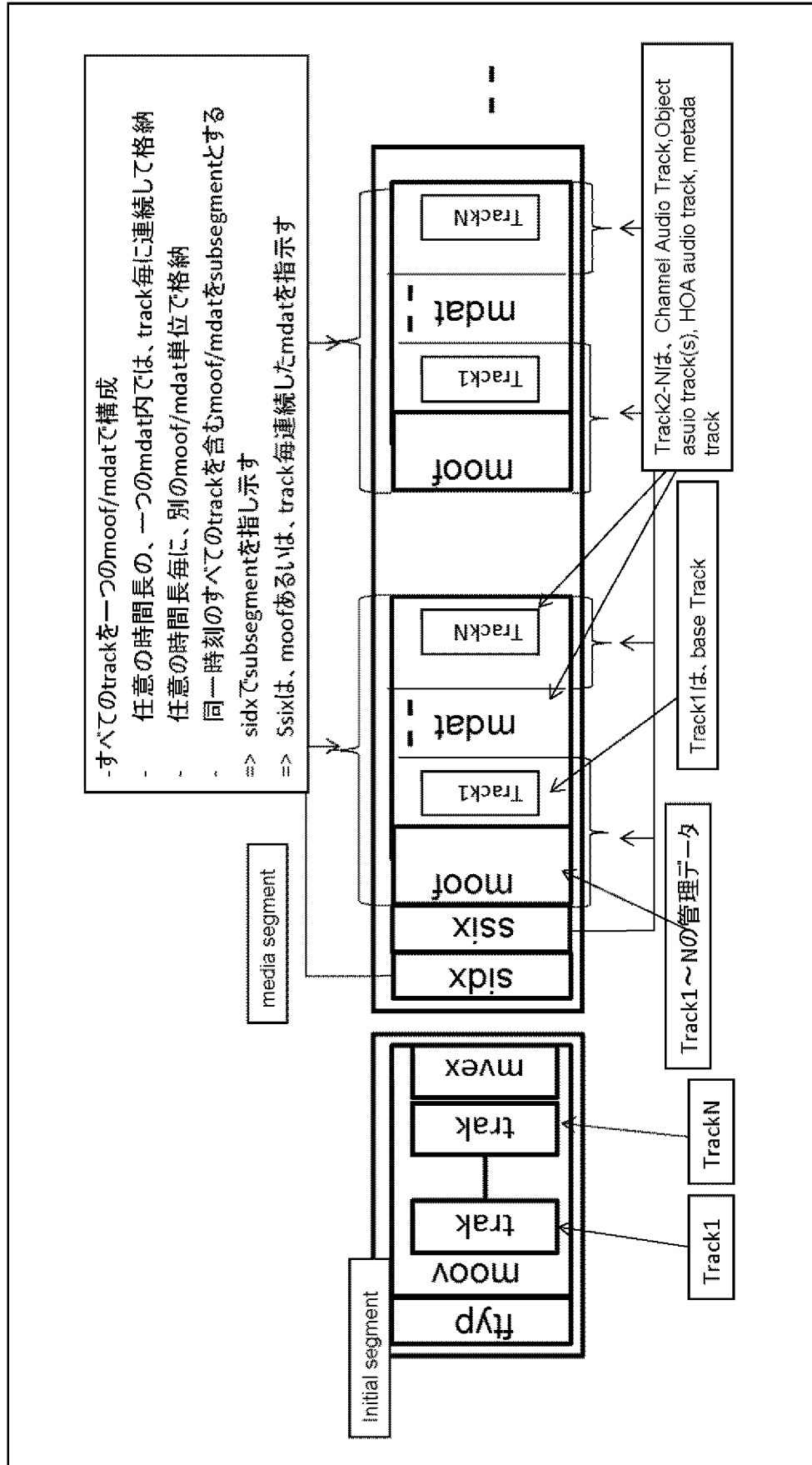
class MHAMetadataConfigurationBox() extends Box('hahC') {
    MHAMetadataConfigurationRecord() MHAMConfig;
}

aligned(8) class MHAMetadataConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    unsigned int(16) reserved
}

```

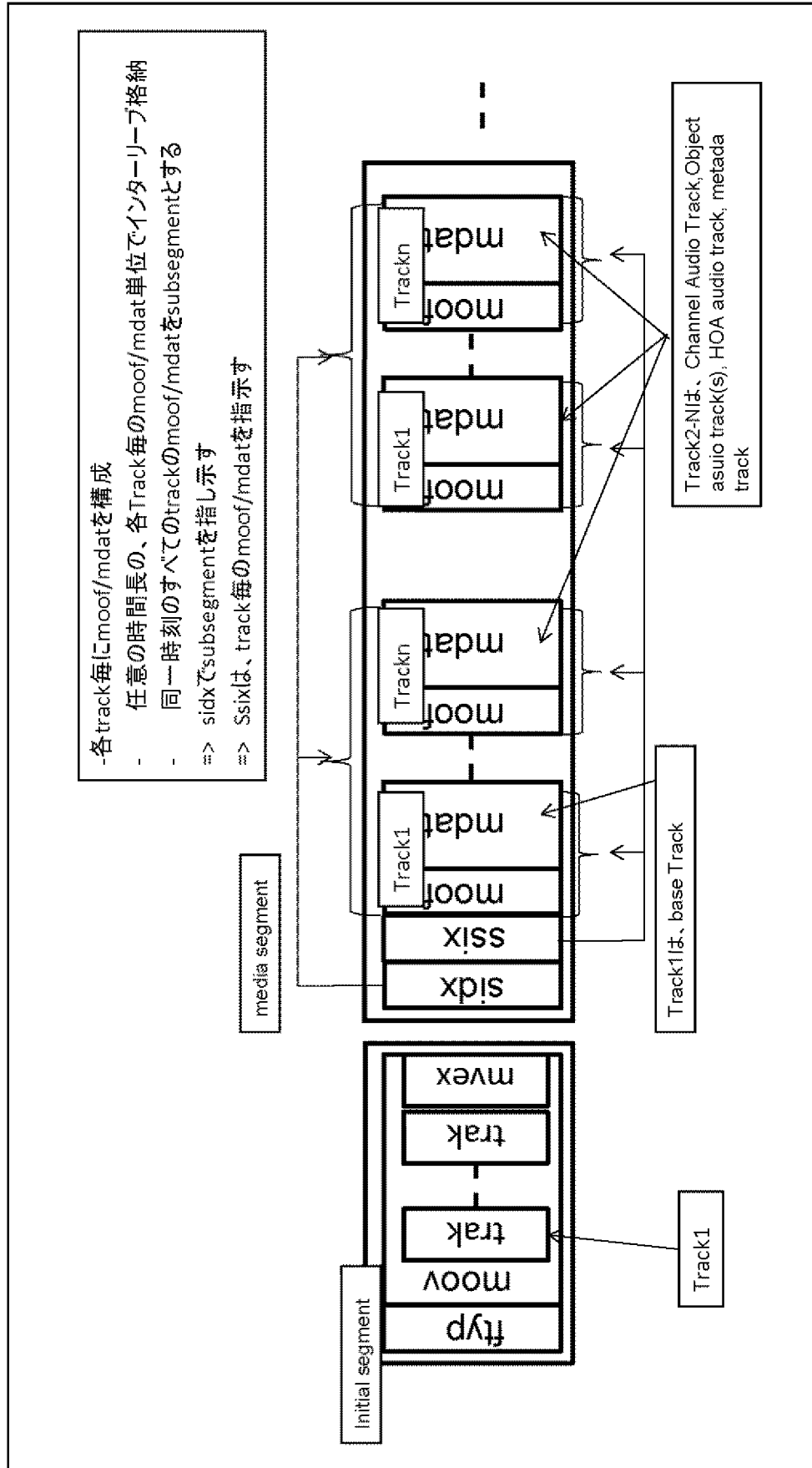
[図39]

図39



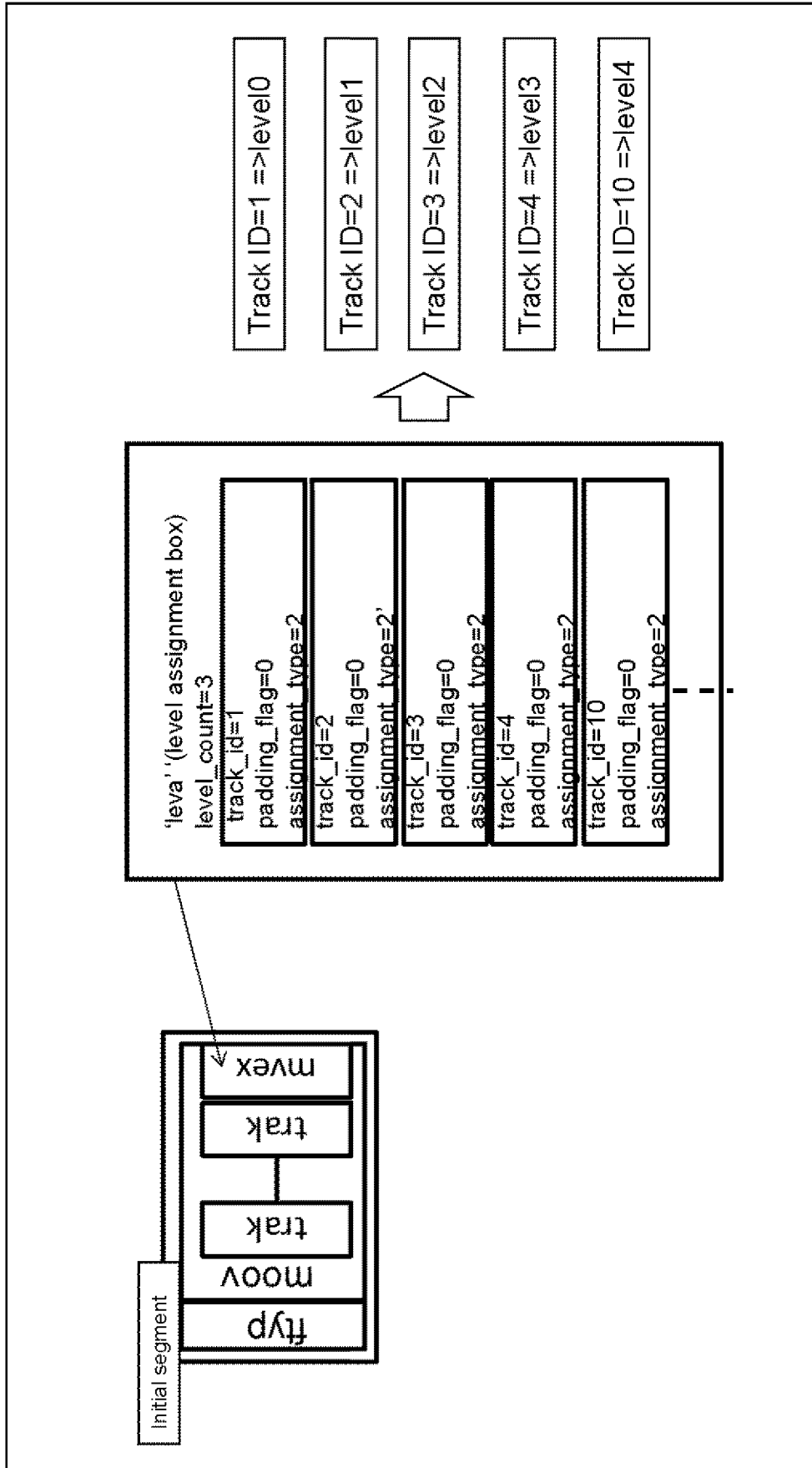
[図40]

図40



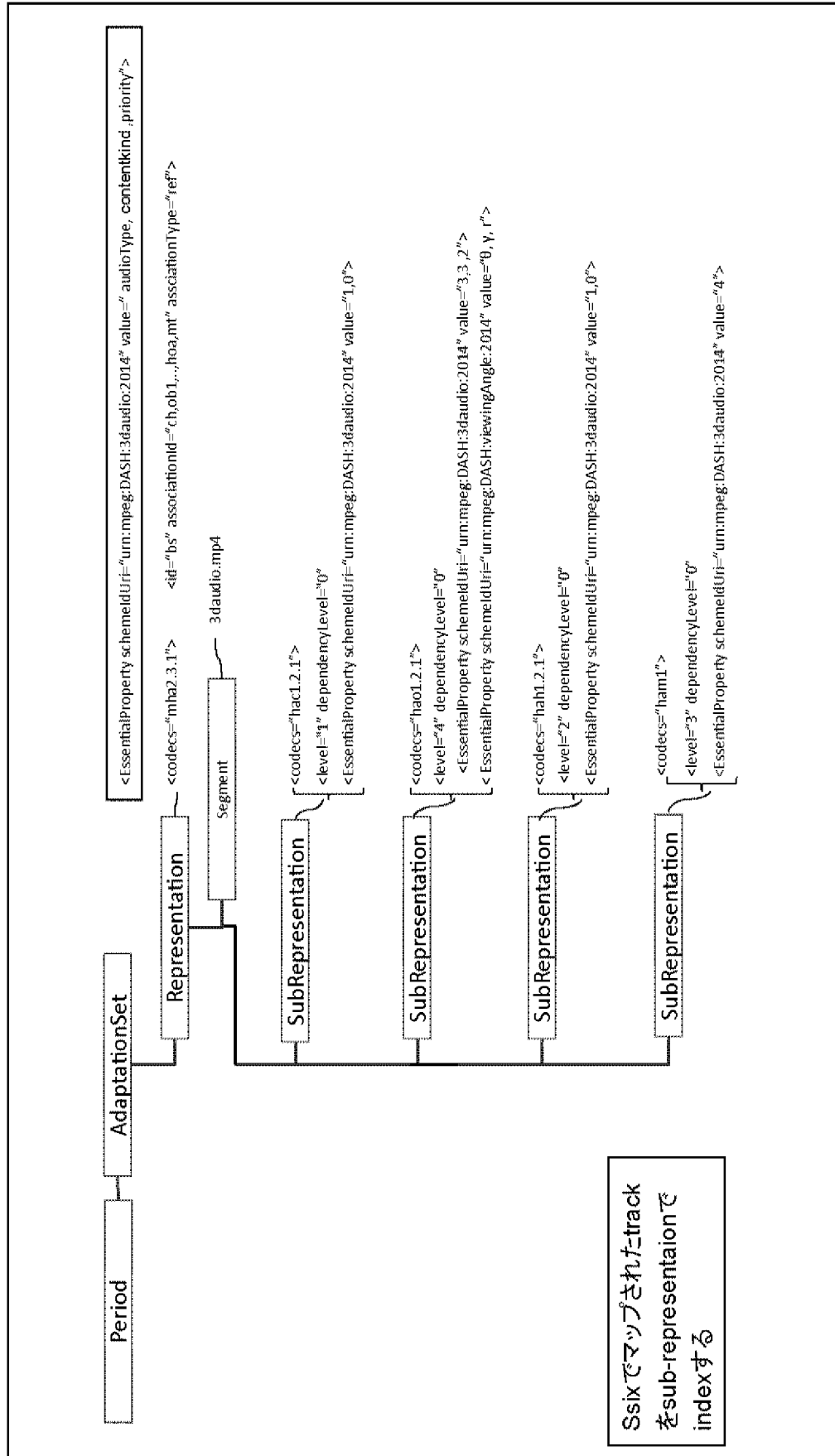
[図41]

図41



[図42]

図42

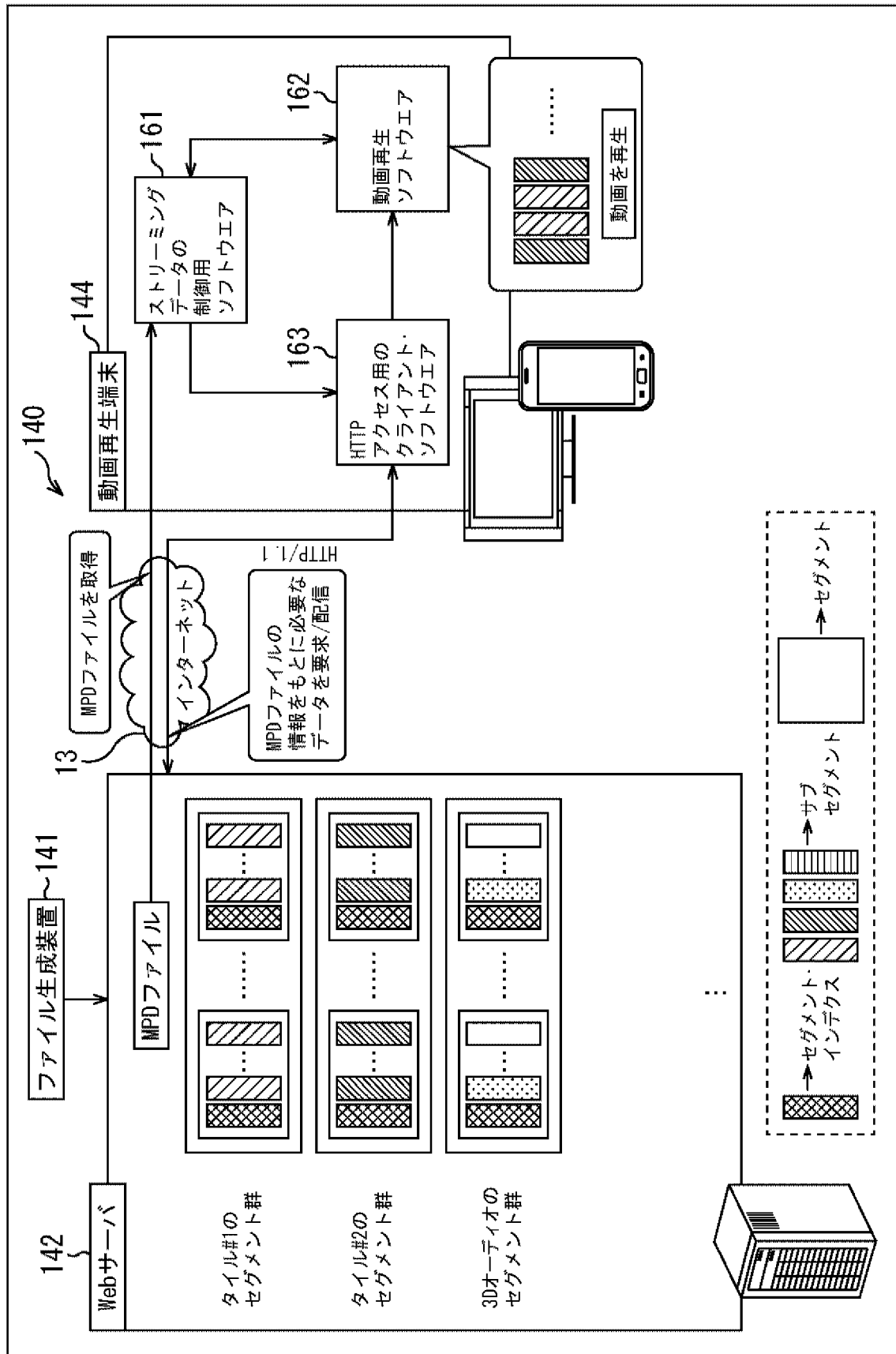


[図43]
図43

AudioType 1 channel audio 2 HOA audio 3 object audio 4 metadata		Priorityは 23008-3で定義して いるもの	
		contentKind	description
		0	undefined
		1	complete main
		2	dialogue
		3	music
		4	effect
		5	mixed
		6	LFE
		7	voiceover
		8	spokensubtitle
		9	audiodescription/visually impaired
		10	commentary
		11	hearing impaired
		12	emergency
		13-15	reserved

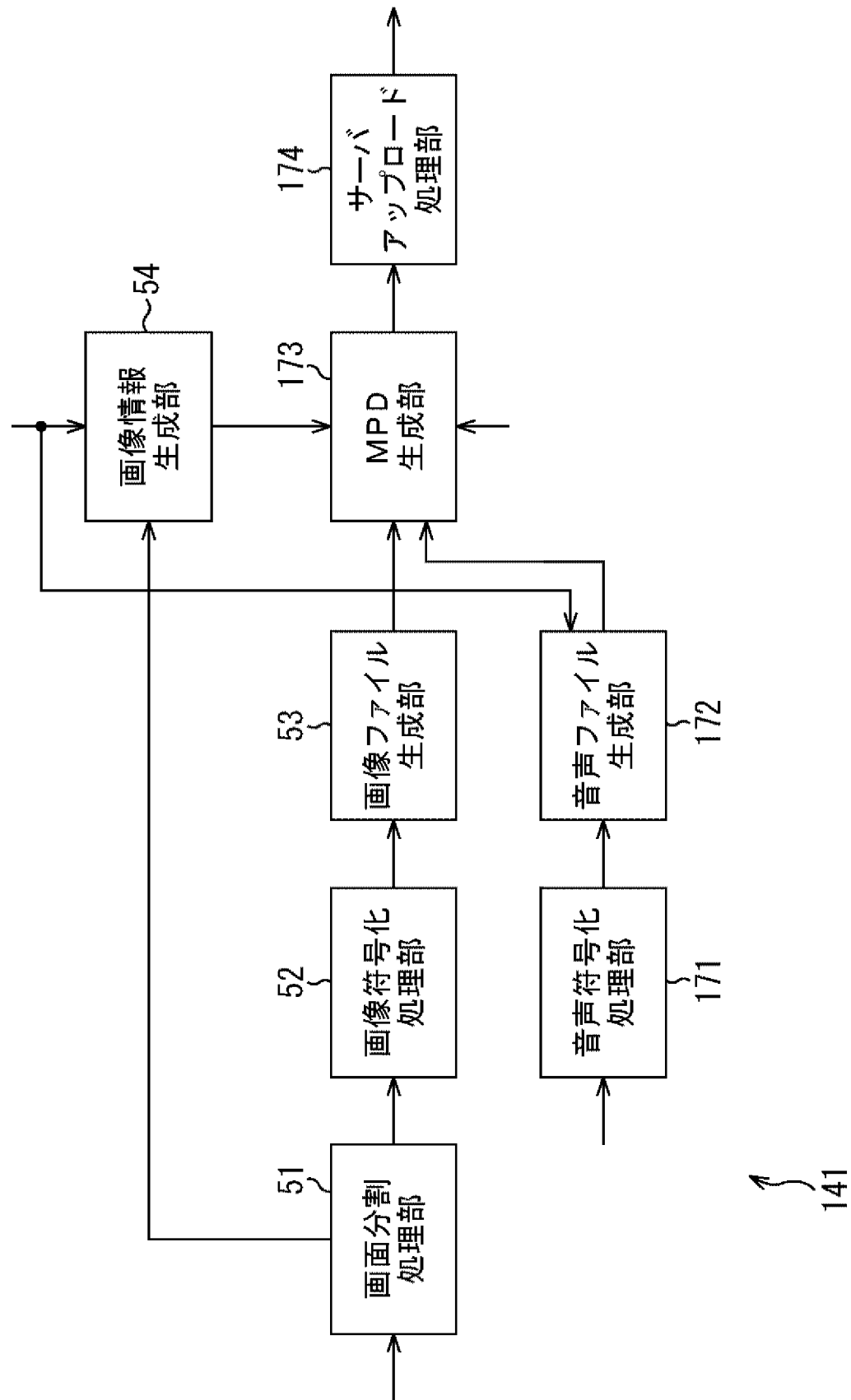
[図44]

図44



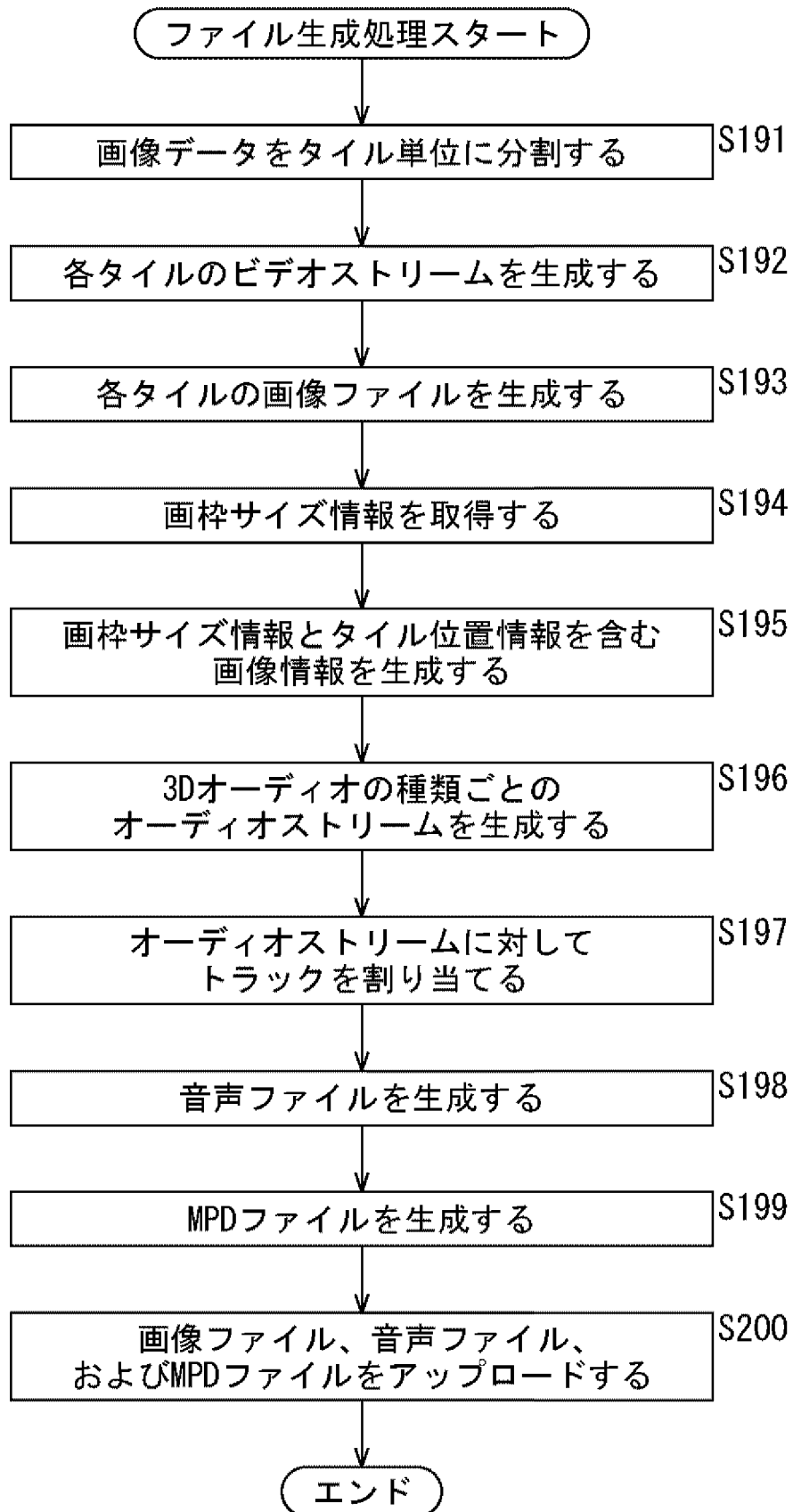
[図45]

図45



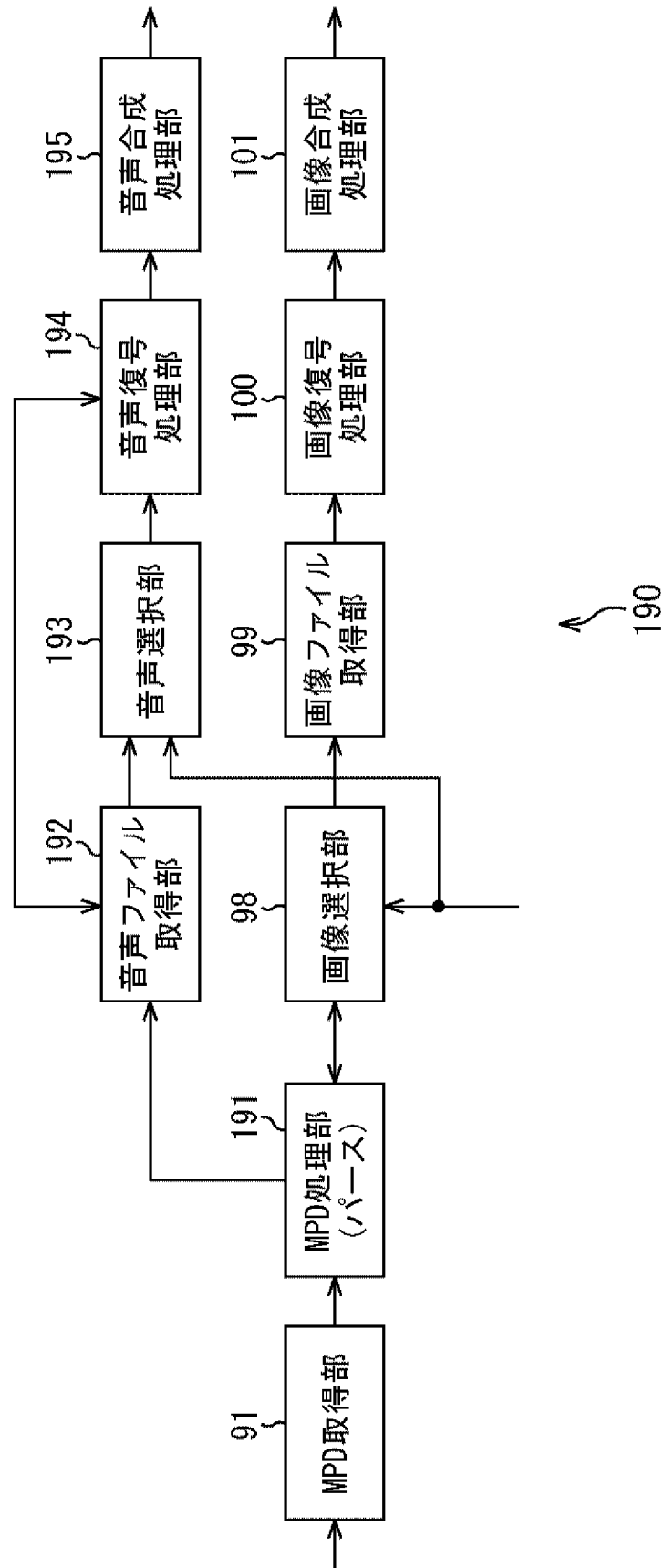
[図46]

図46



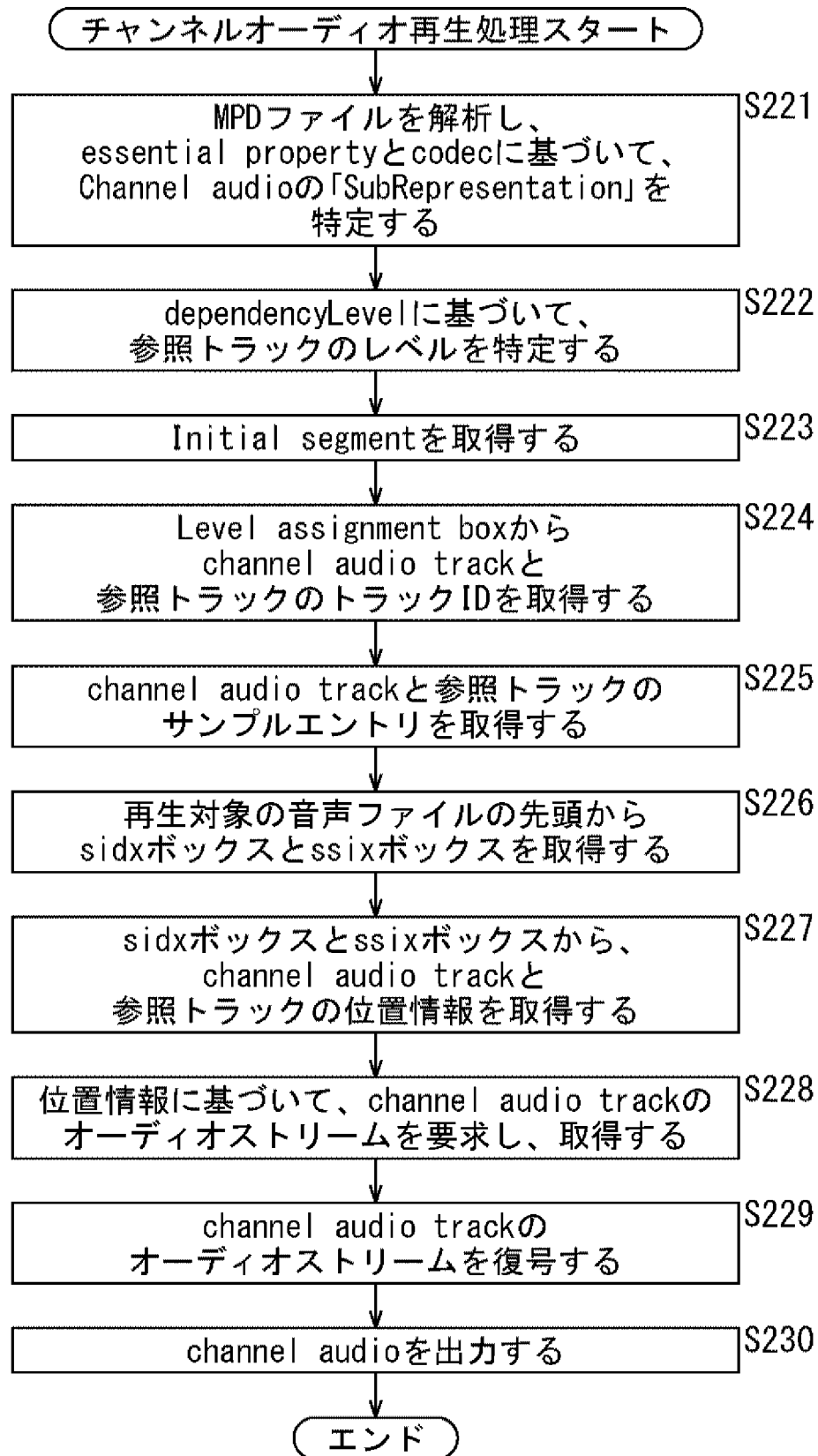
[図47]

図47



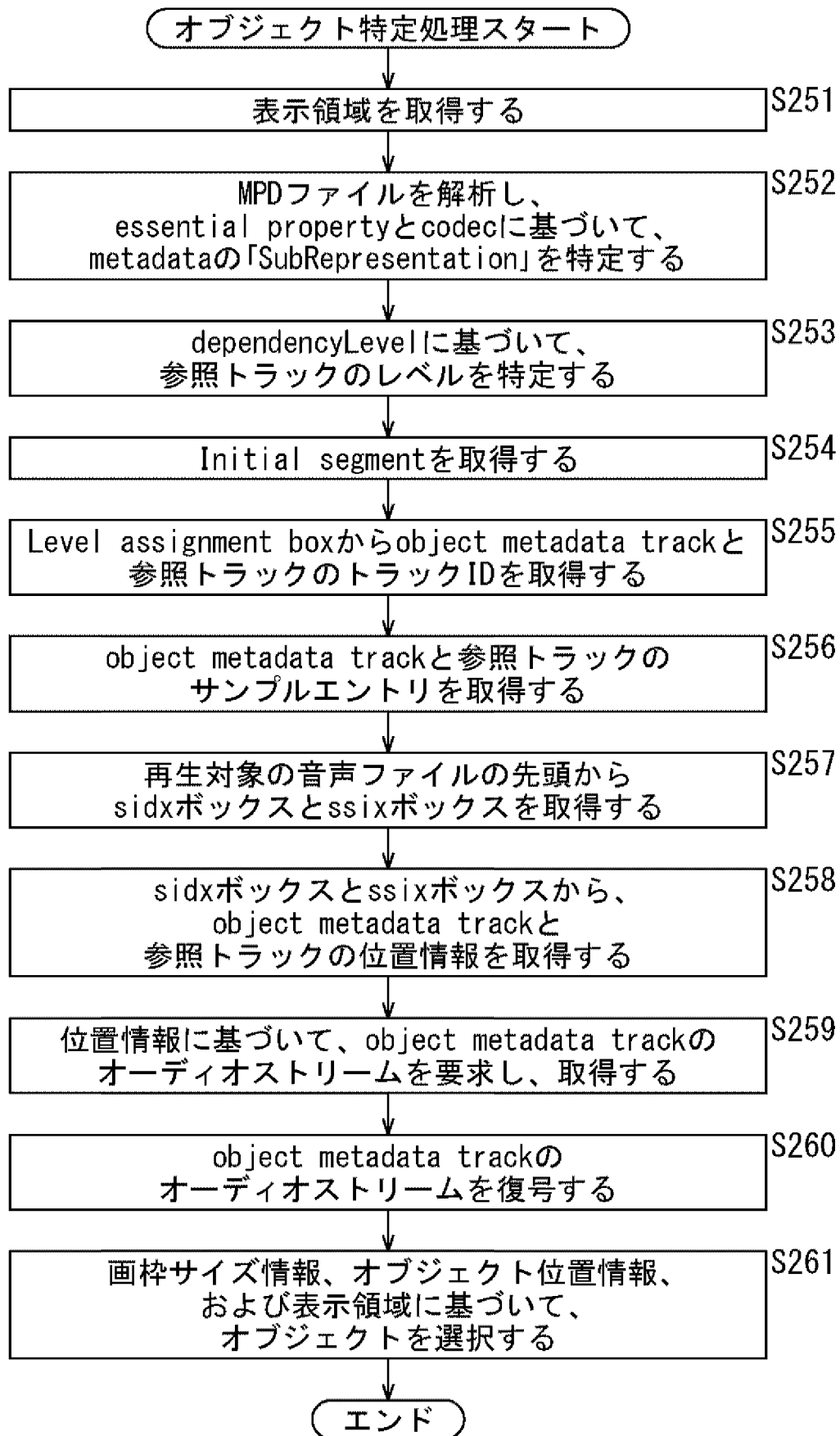
[図48]

図48



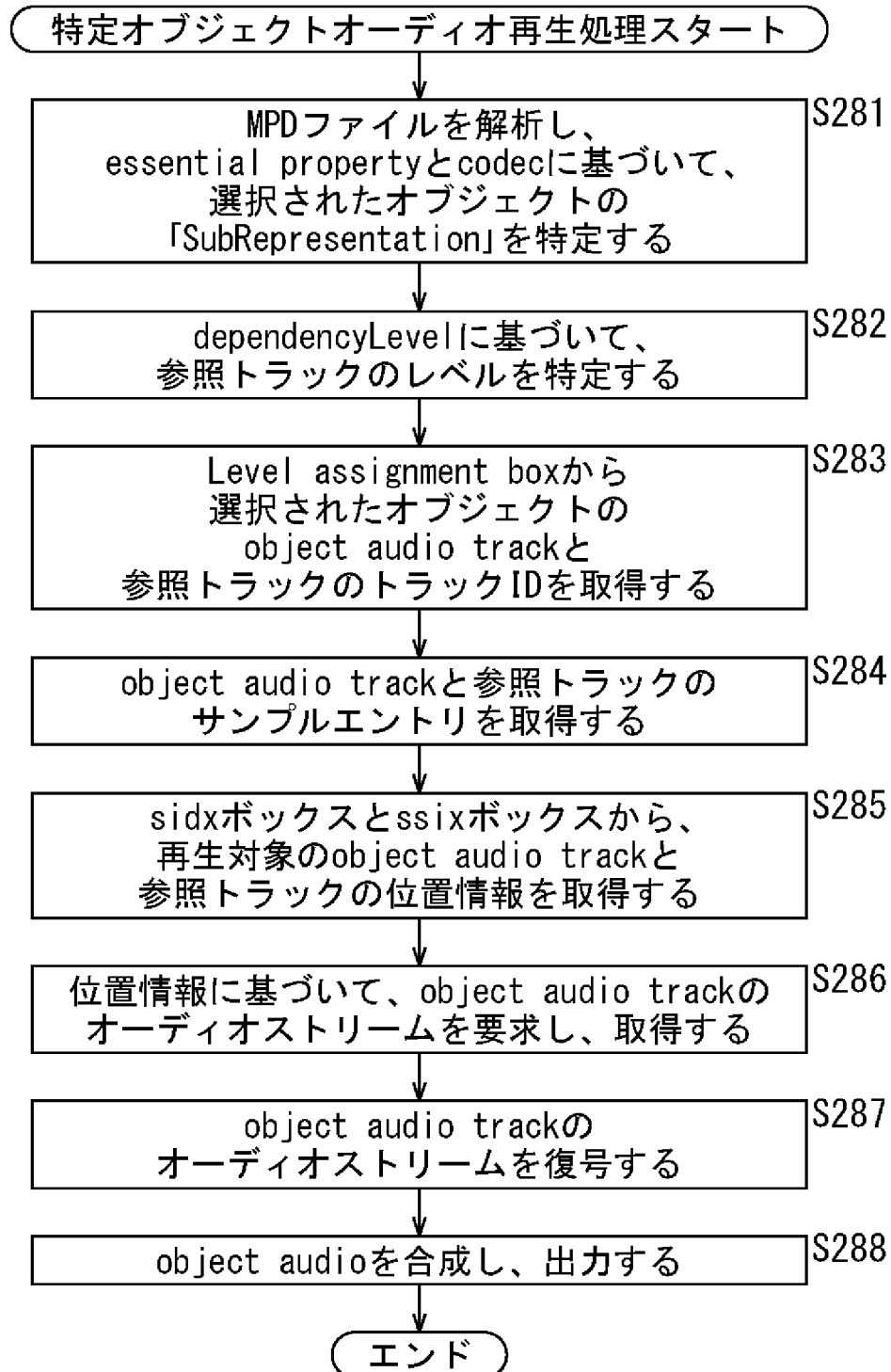
[図49]

図49



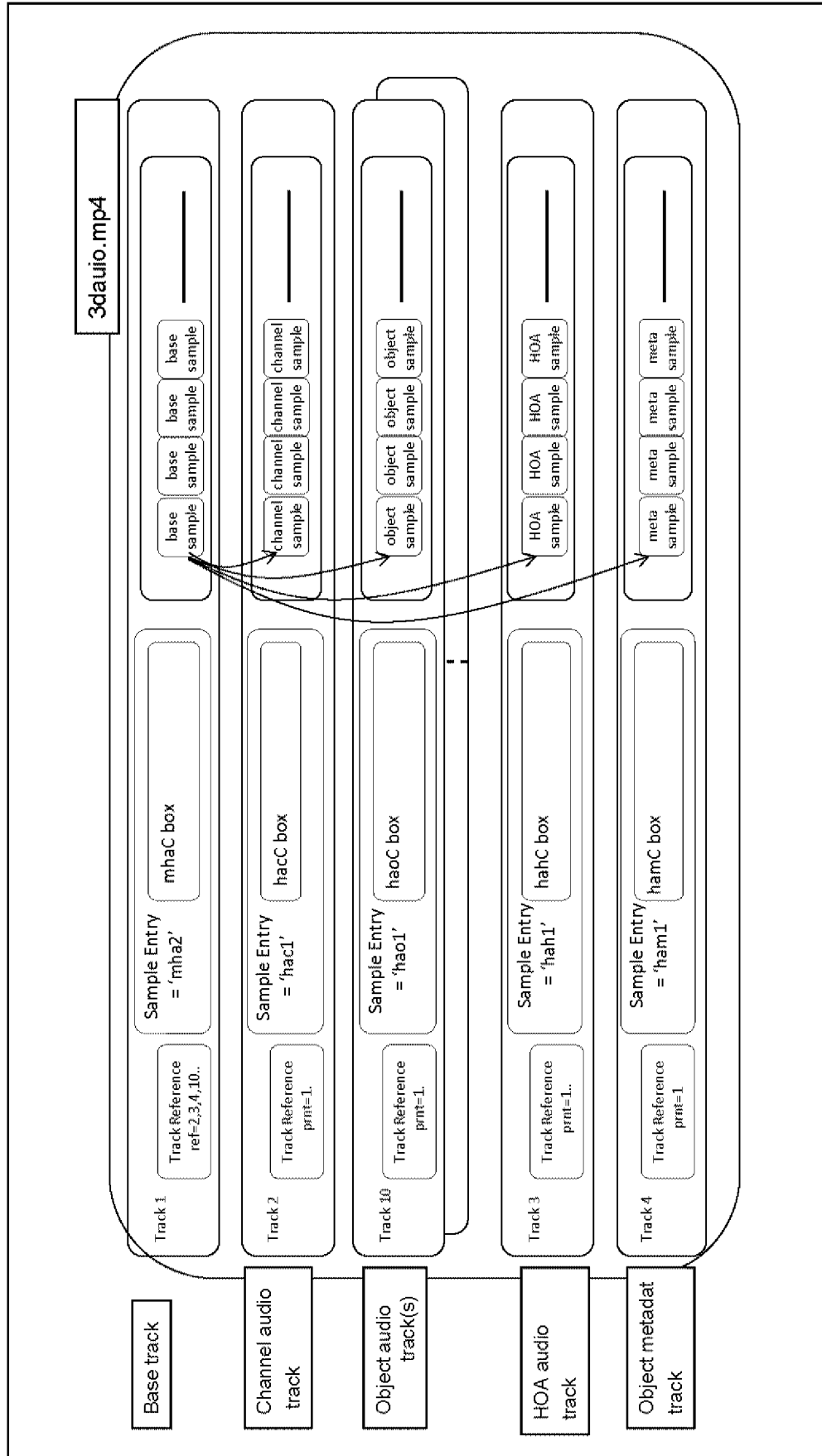
[図50]

図50



[図51]

図51



[図52]

図52

```

class MHASampleEntry() extends AudioSampleEntry('mha2') {
    MHAConfigurationBox config;
    MPEG4BitRateBox();           // optional
    MPEG4ExtensionDescriptorsBox (); // optional
}

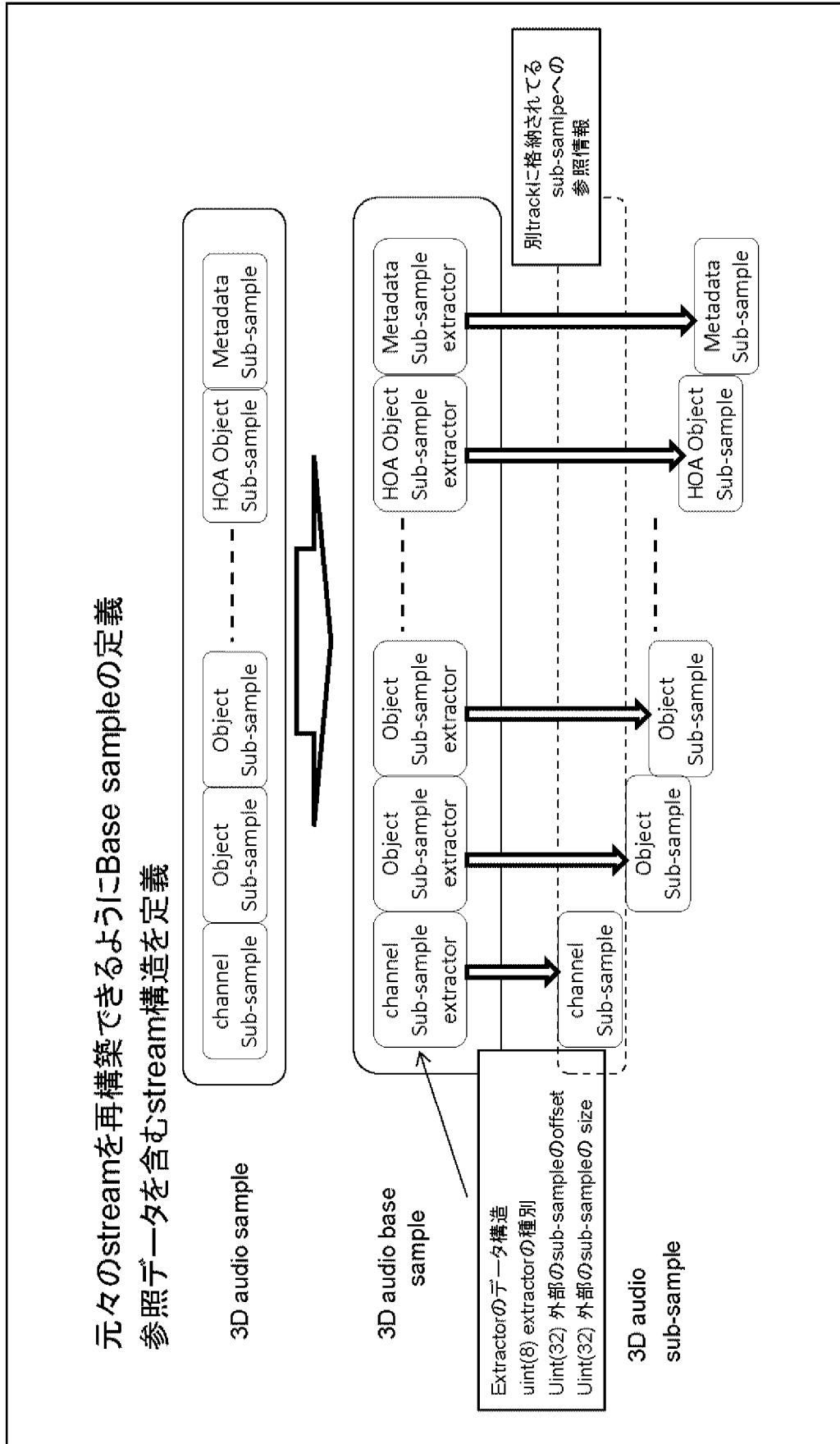
class MHAConfigurationBox() extends Box('mhaC') {
    MHADecoderConfigurationRecord() MHAConfig;
}

aligned(8) class MHADecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    unsigned int(8) MPEGHAudioProfile;
    unsigned int(8) MPEGHAudioLevel;
    unsigned int(8) referenceChannelLayout;
    unsigned int(16) mpeg3daConfigLength;
    bit(8*mpeg3daConfigLength) mpeg3daConfig;
}

```

[図53]

図53



[図55]
図55

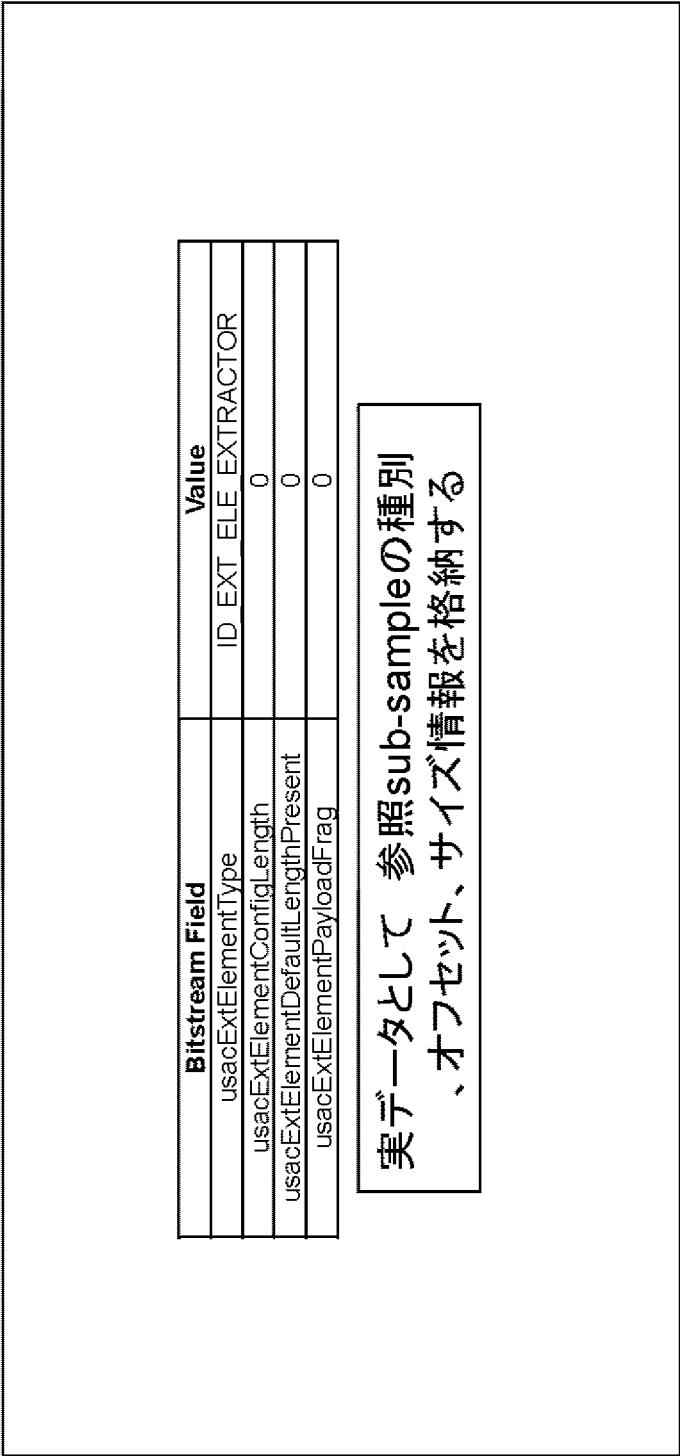
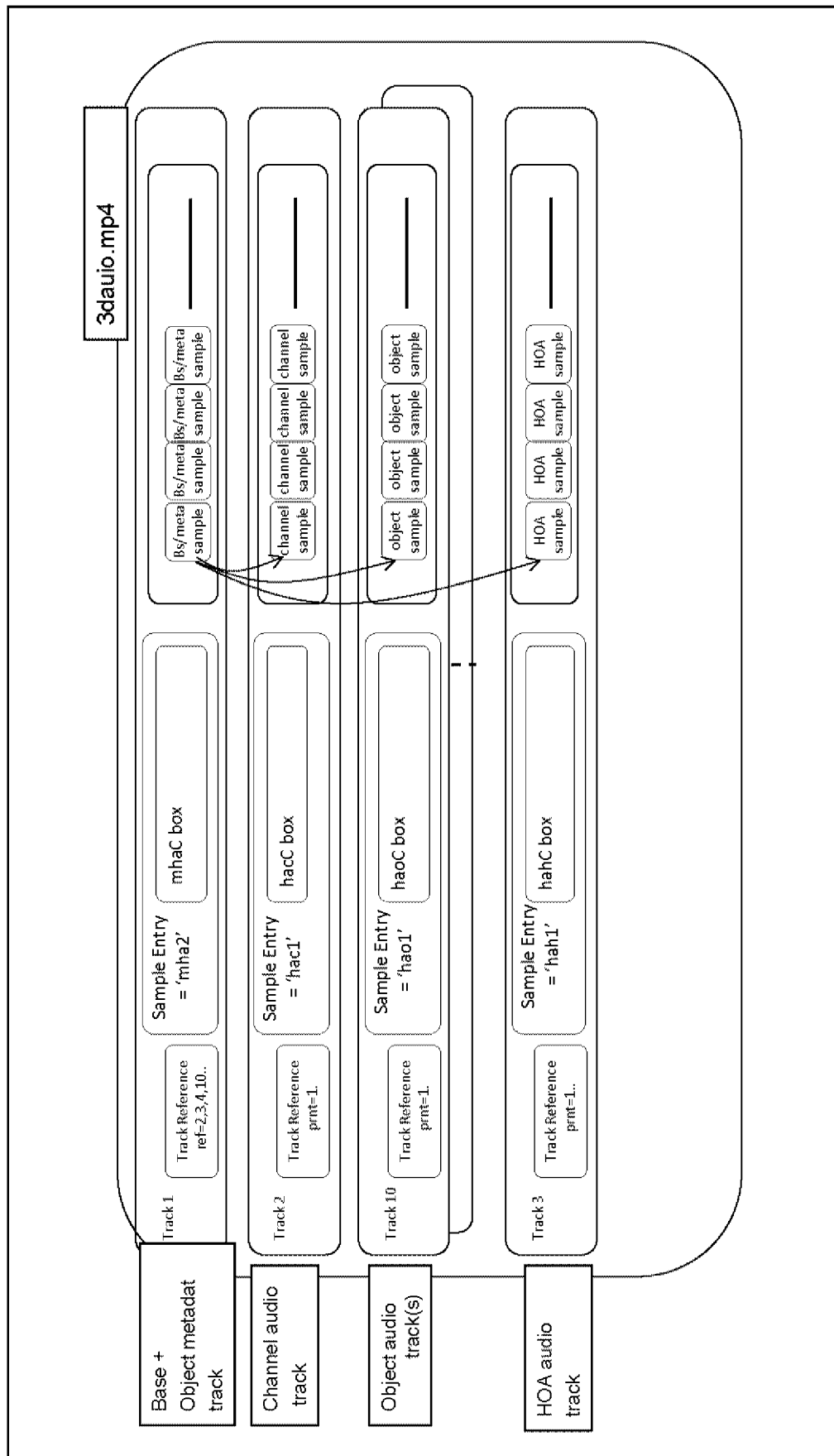
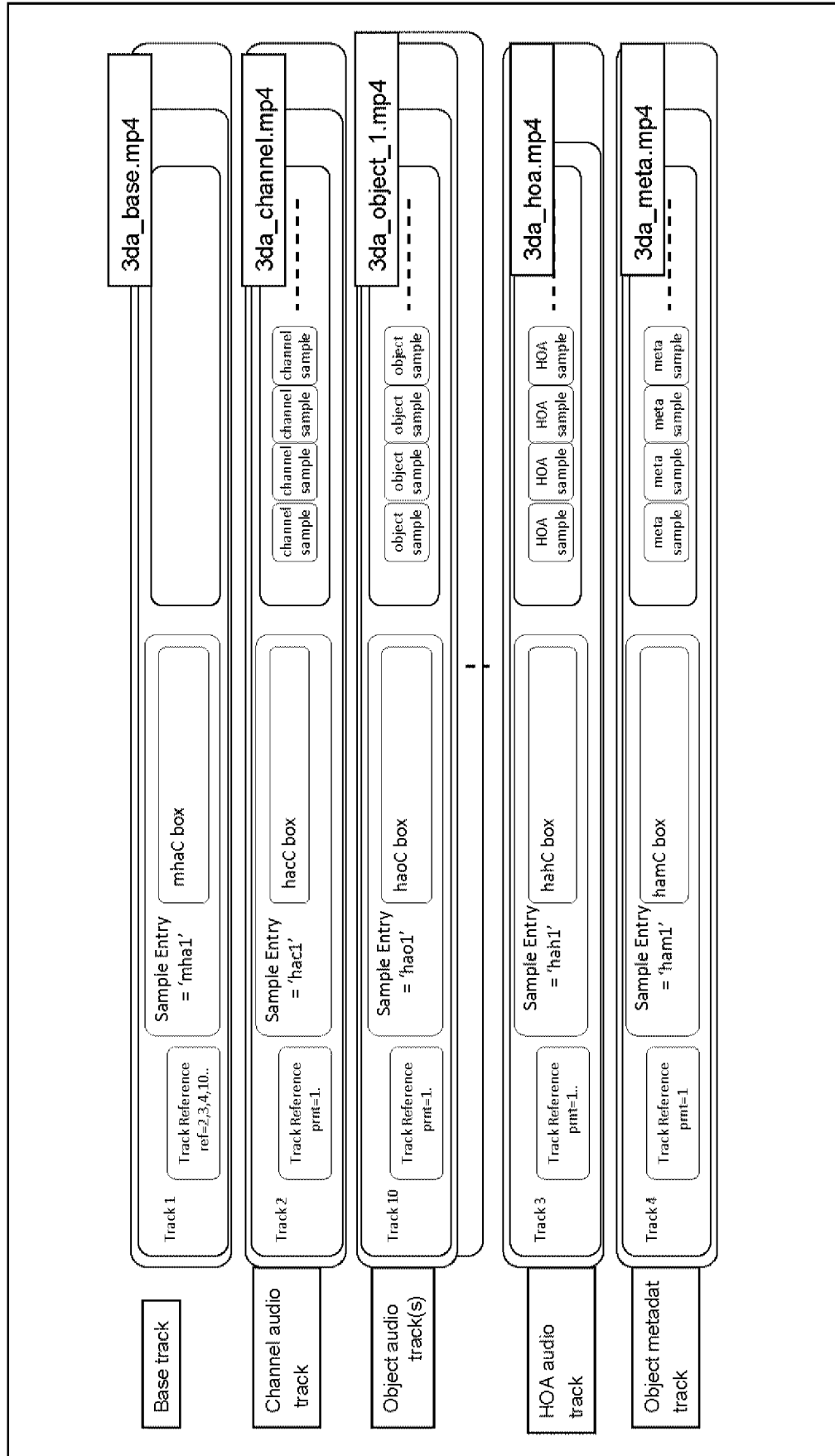


图56



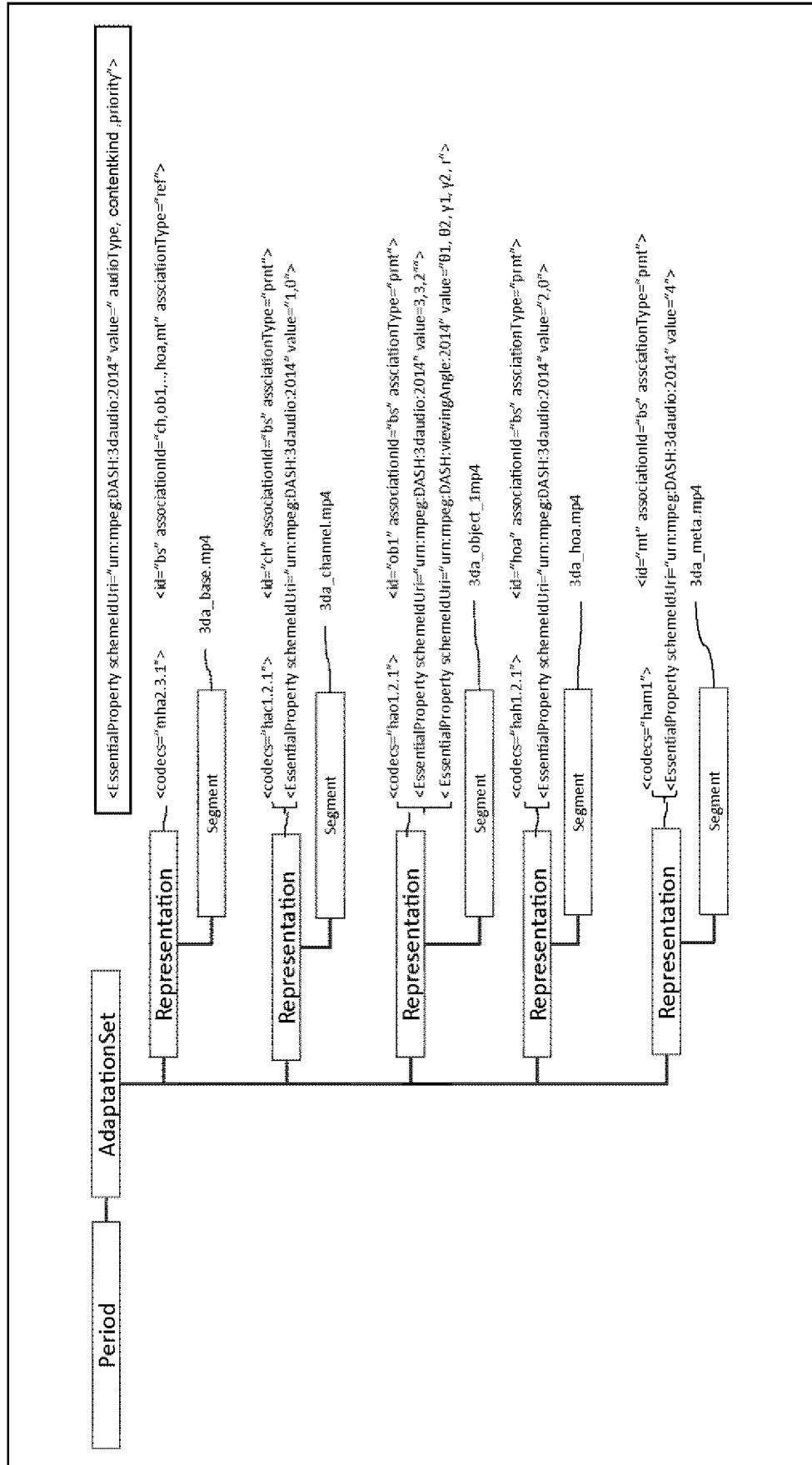
[図57]

図57



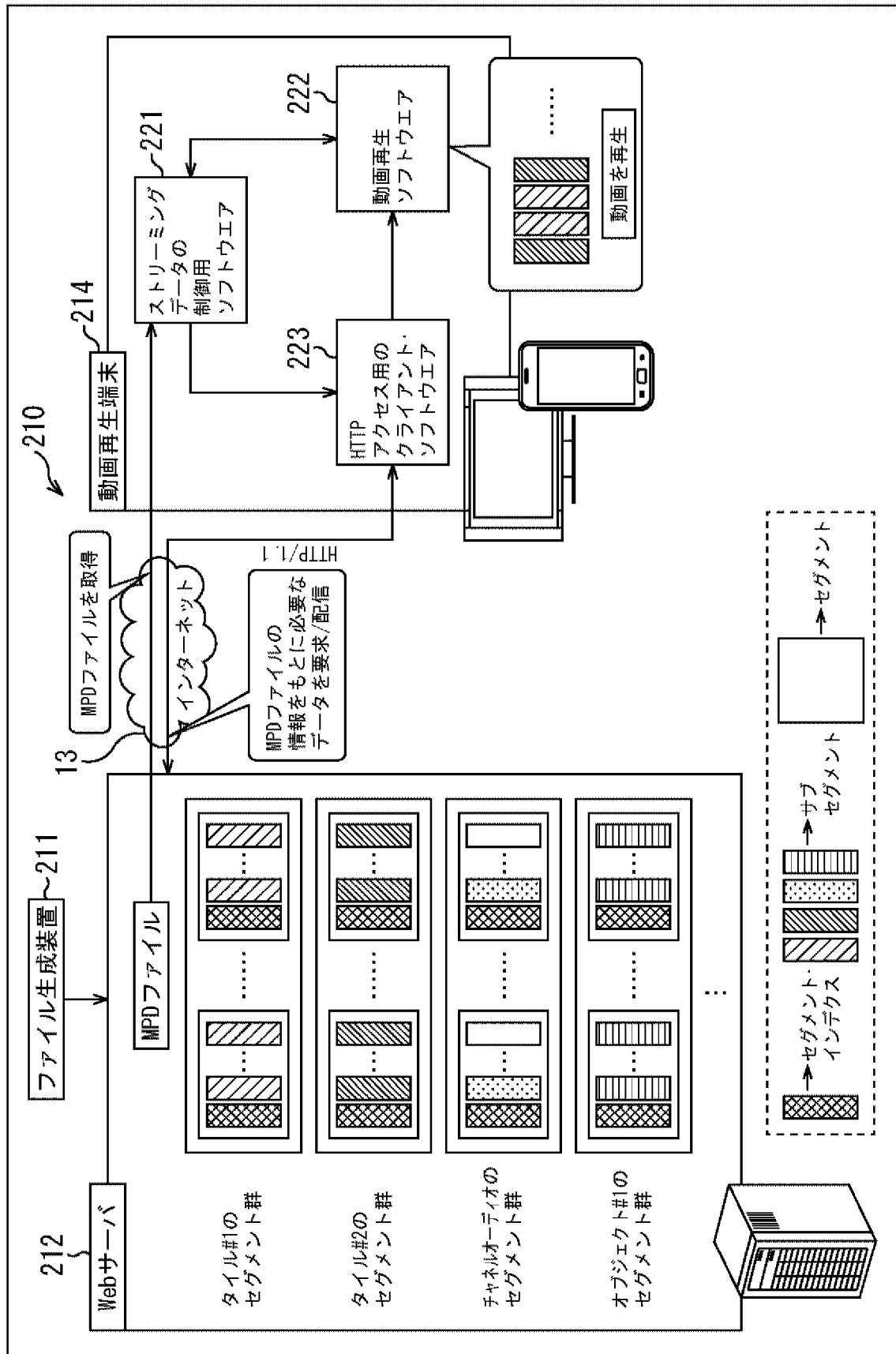
[図58]

図58



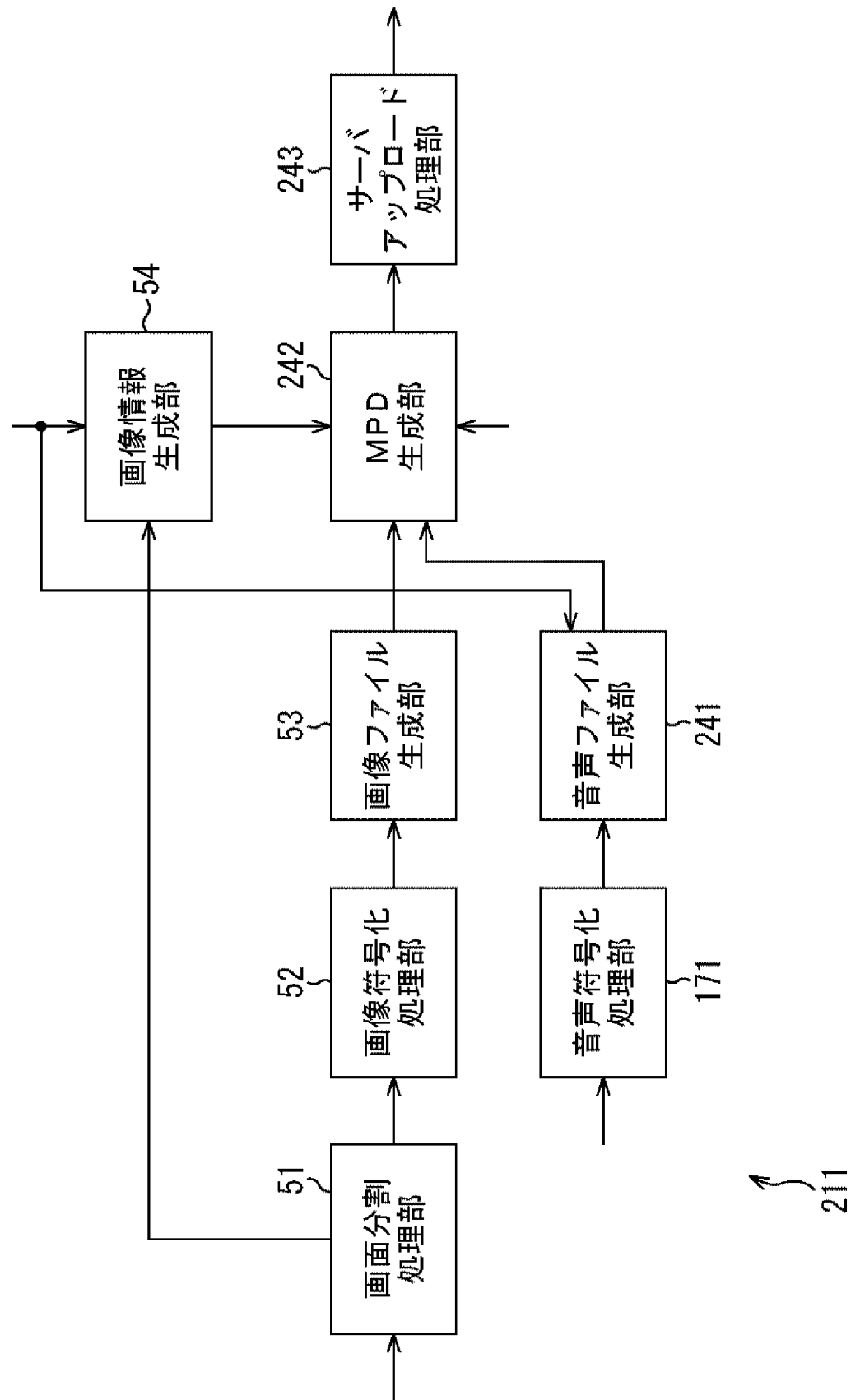
[図59]

図59



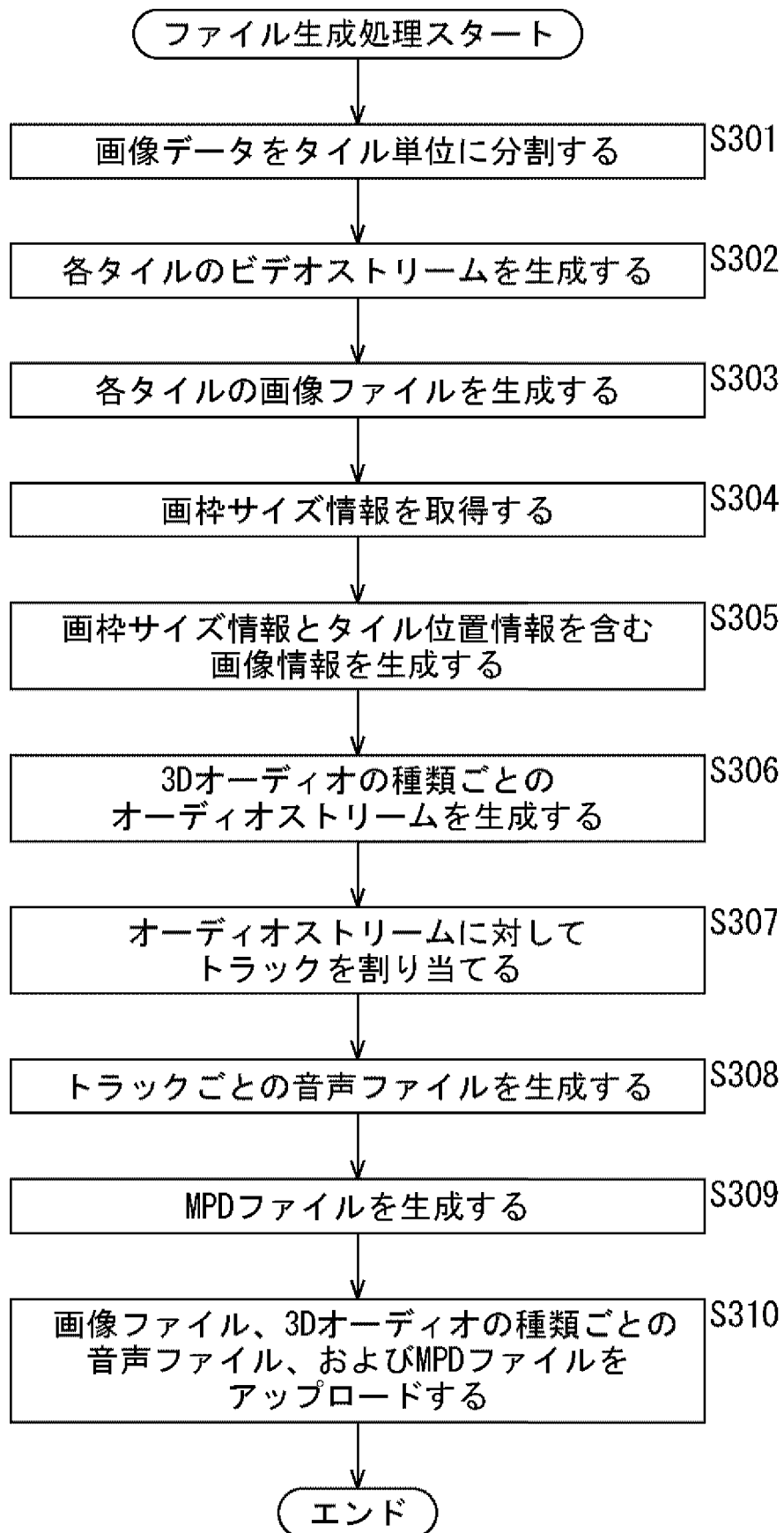
[図60]

図60



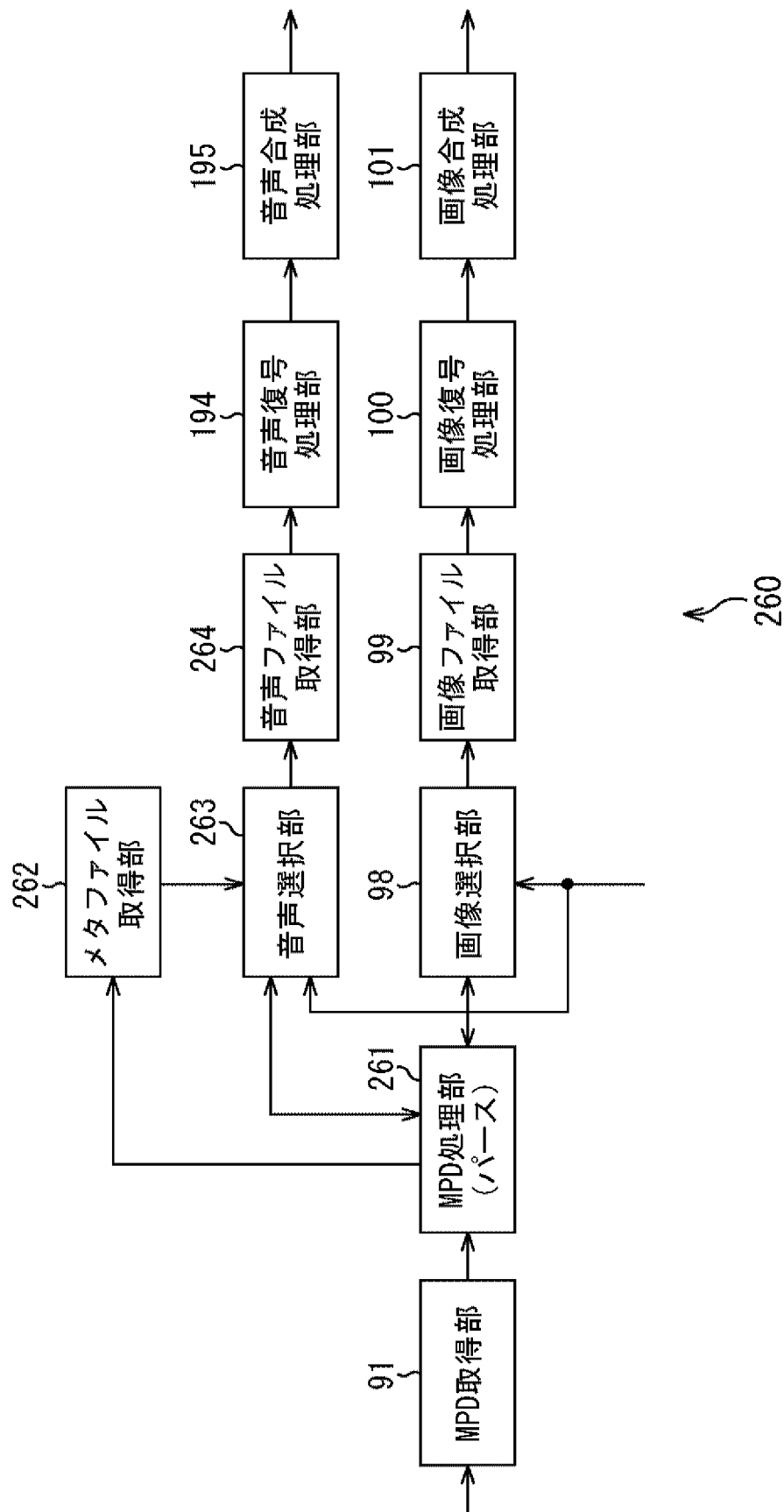
[図61]

図61



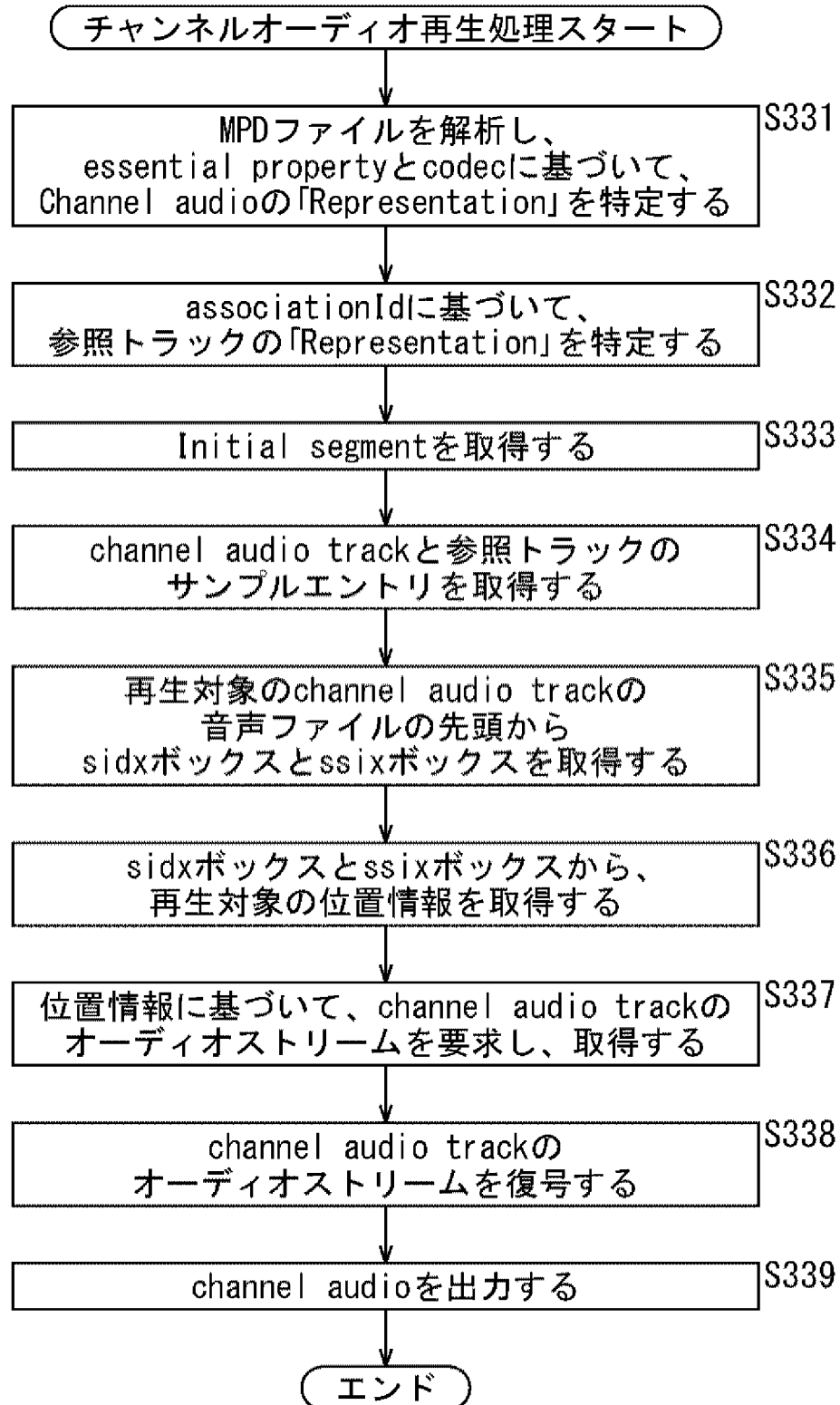
[図62]

図62



[図63]

図63



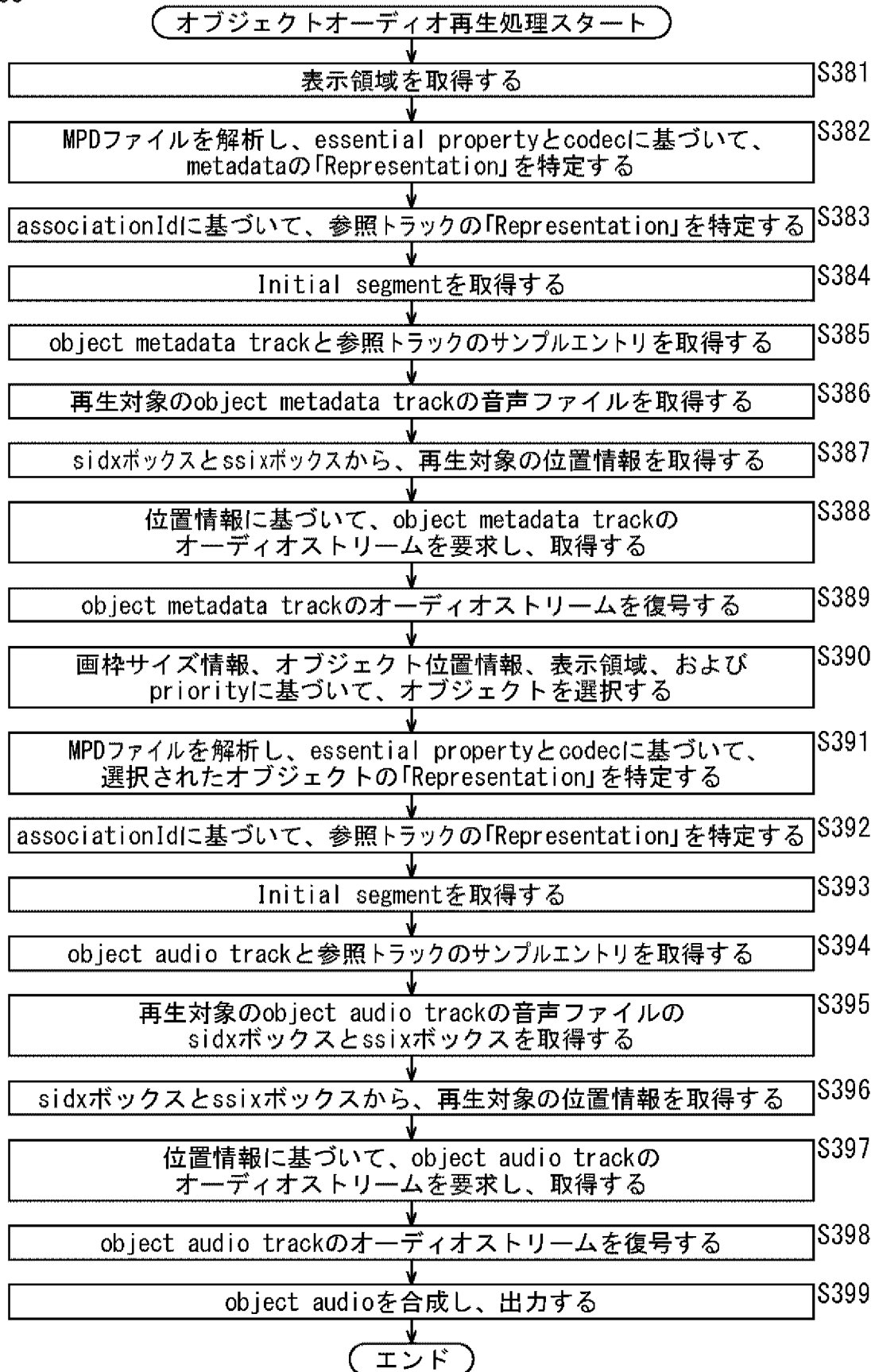
[図64]

図64



[図65]

図65

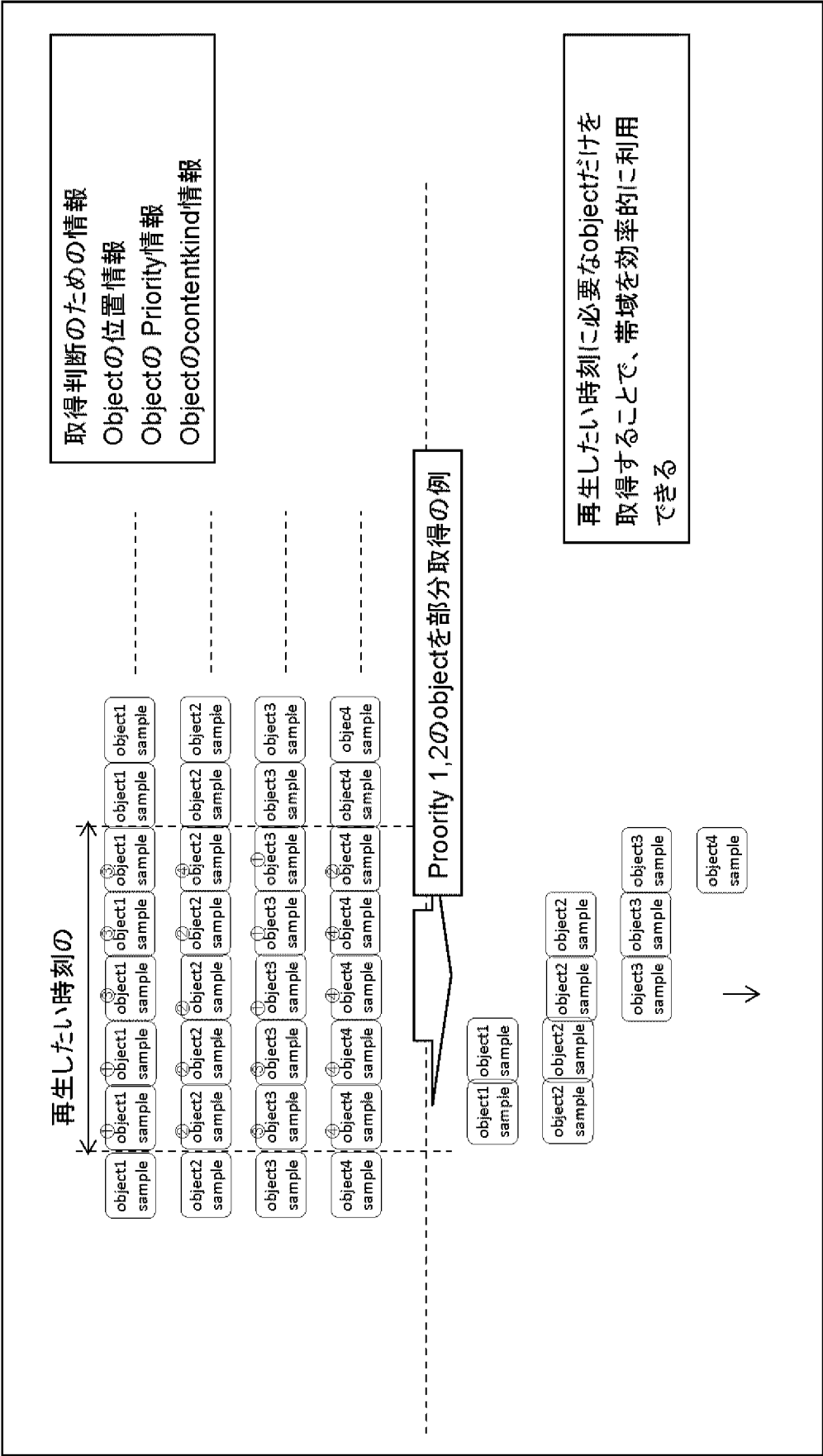


[図66]

図66

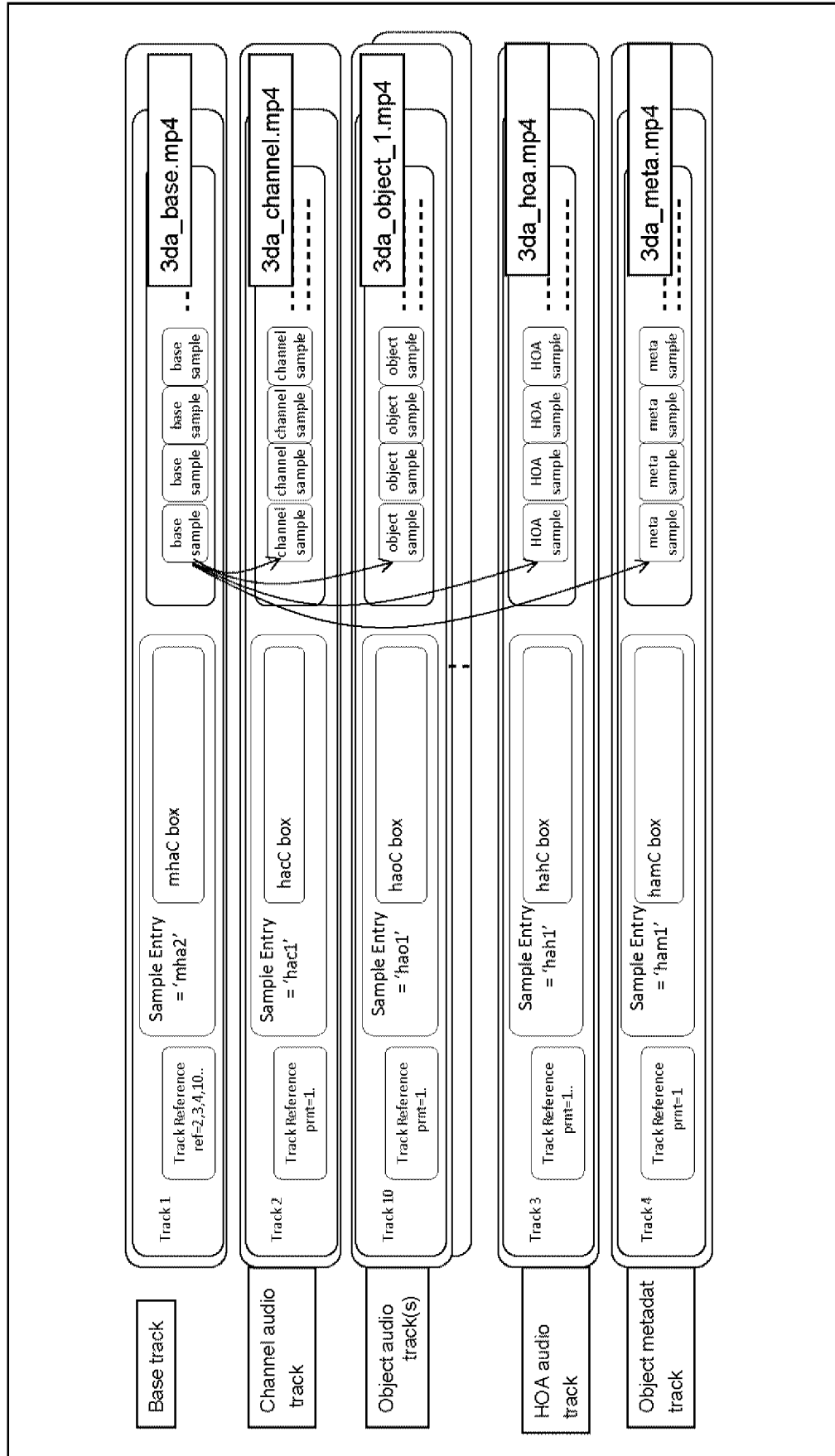


[図67]
図67



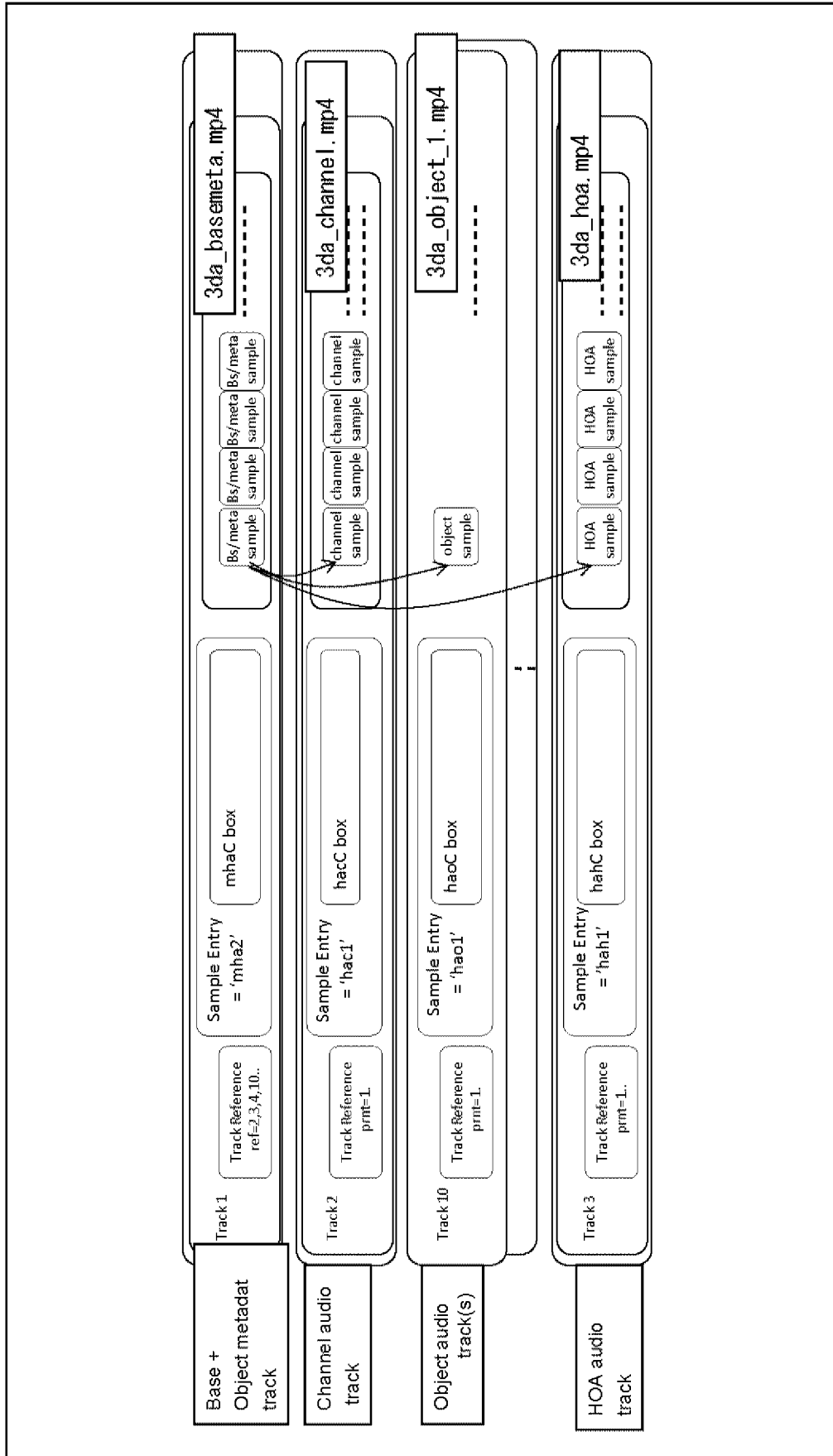
[図68]

図68

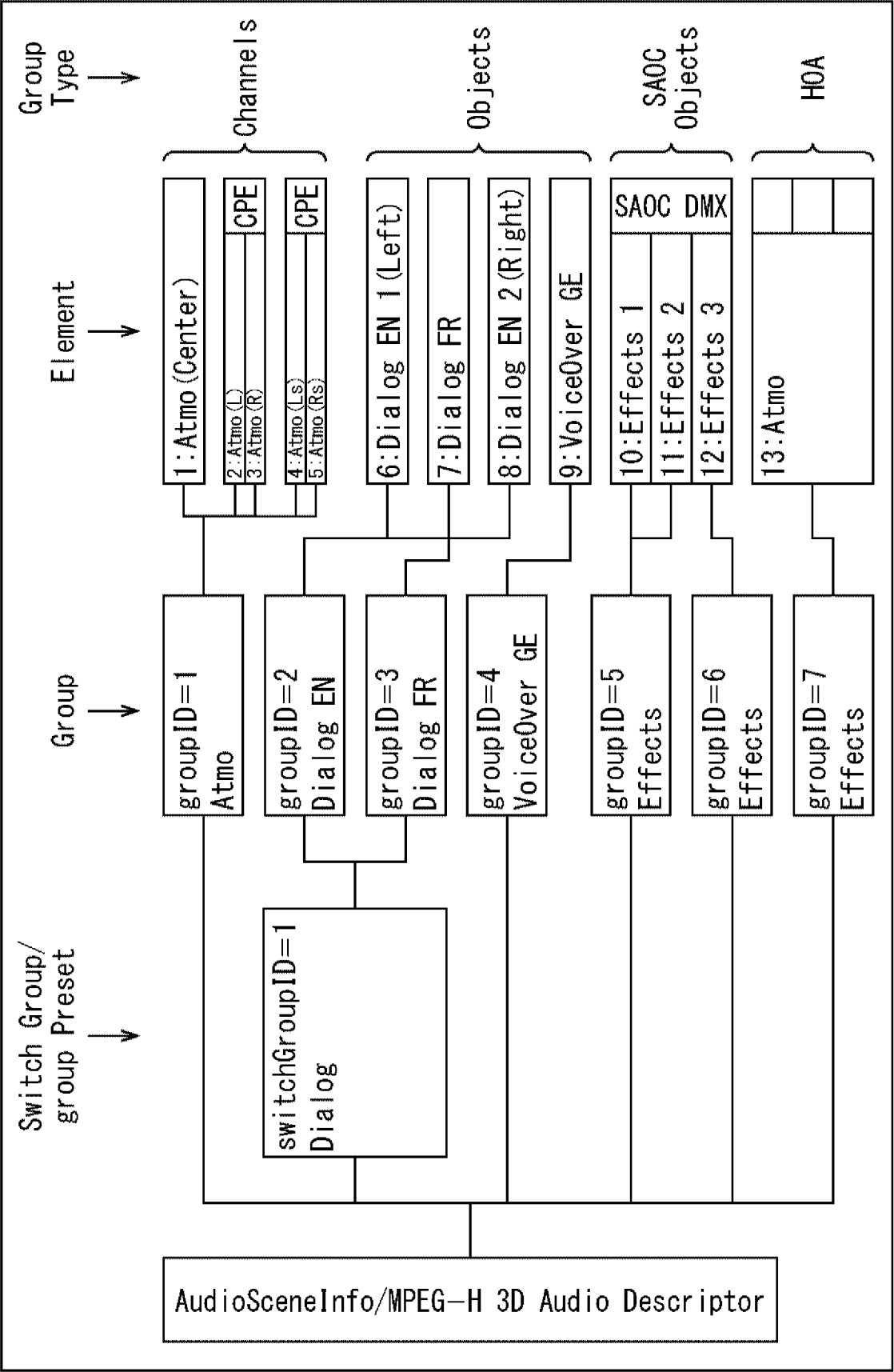


[図69]

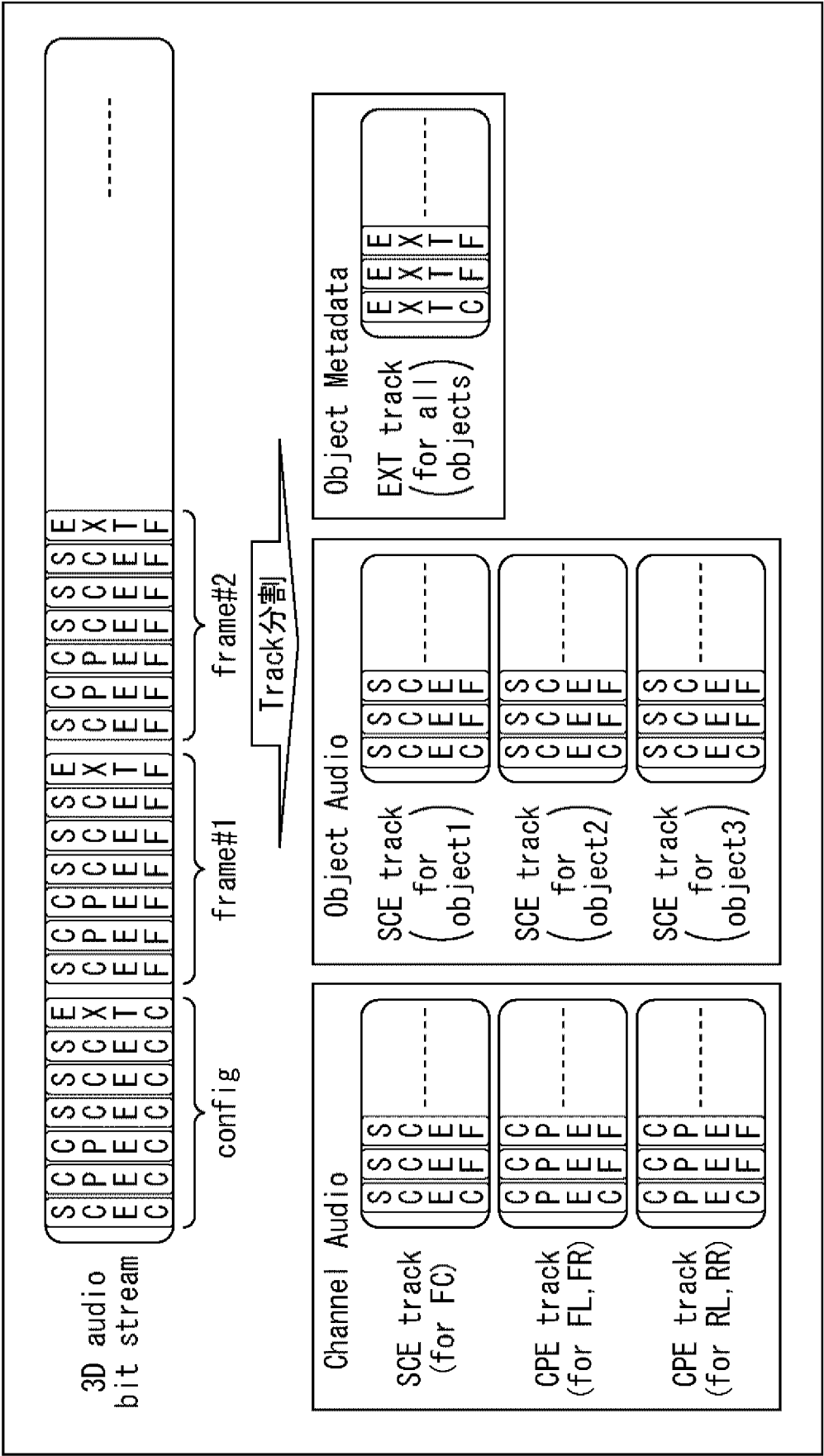
図69



[図70]
図70

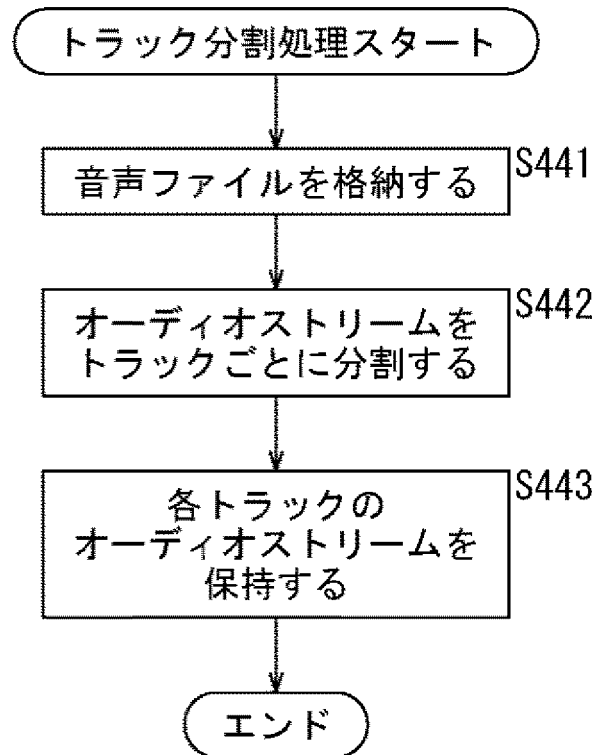


[図71]
図71

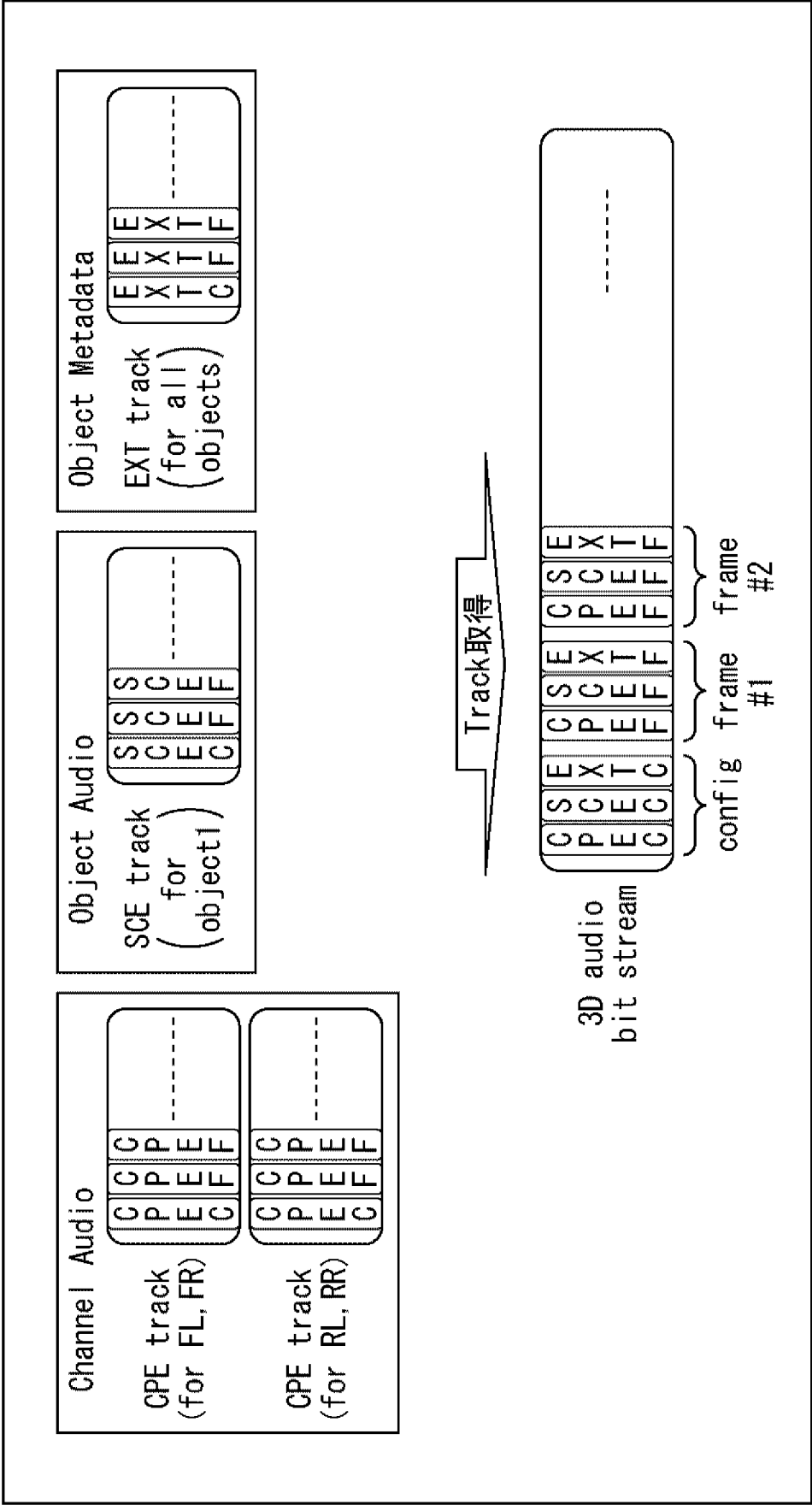


[図72]

図72

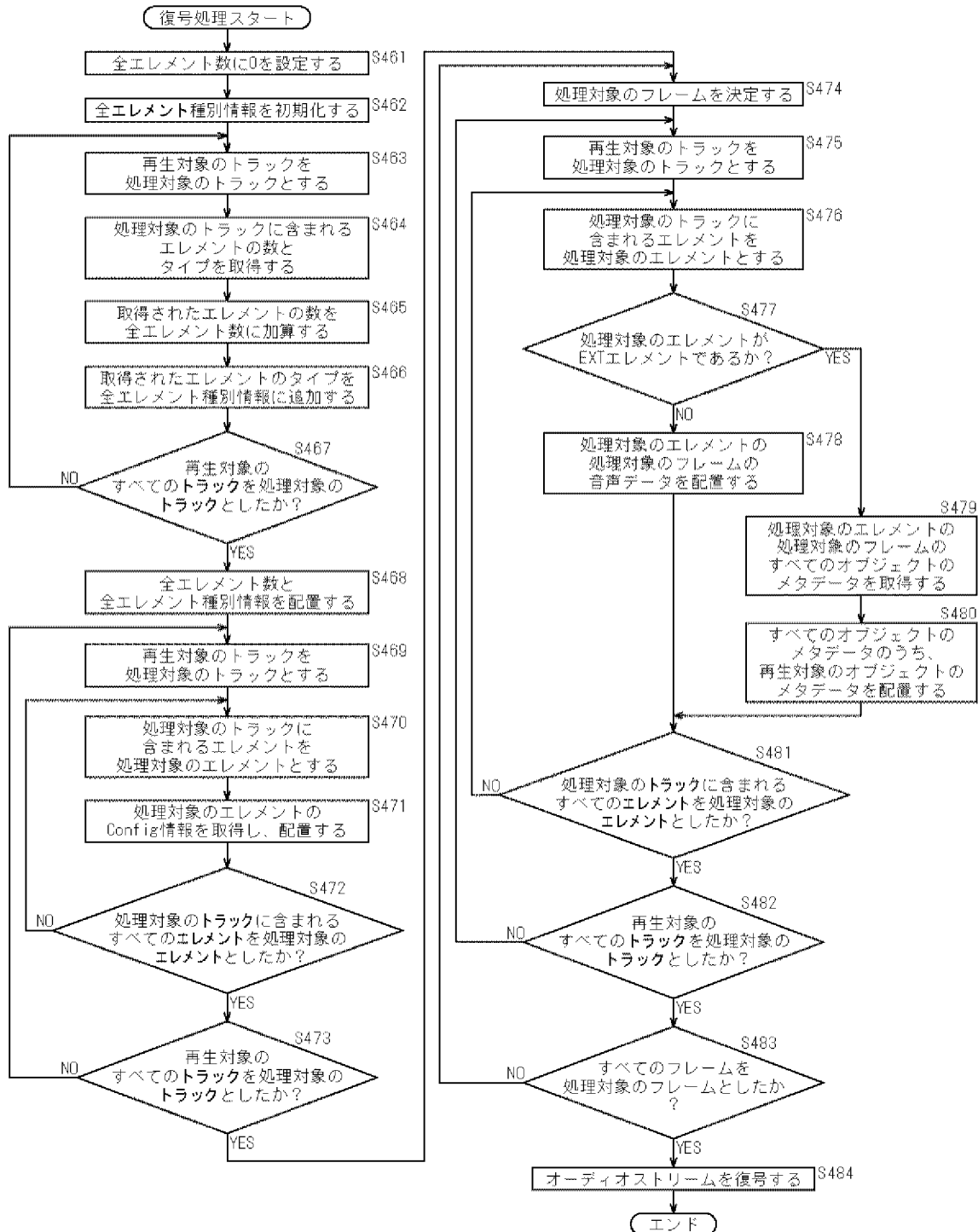


[図73]
図73



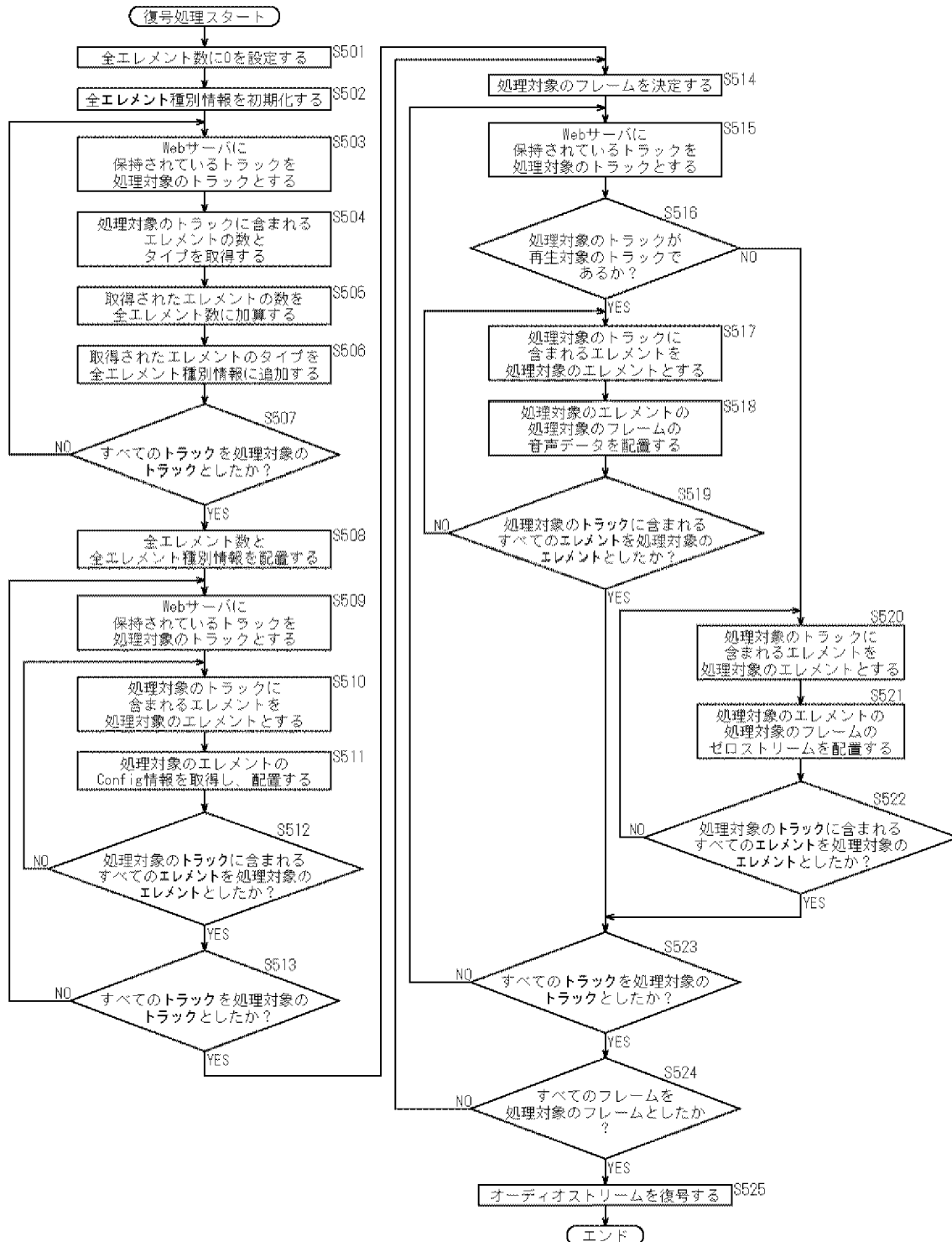
[図74]

図74

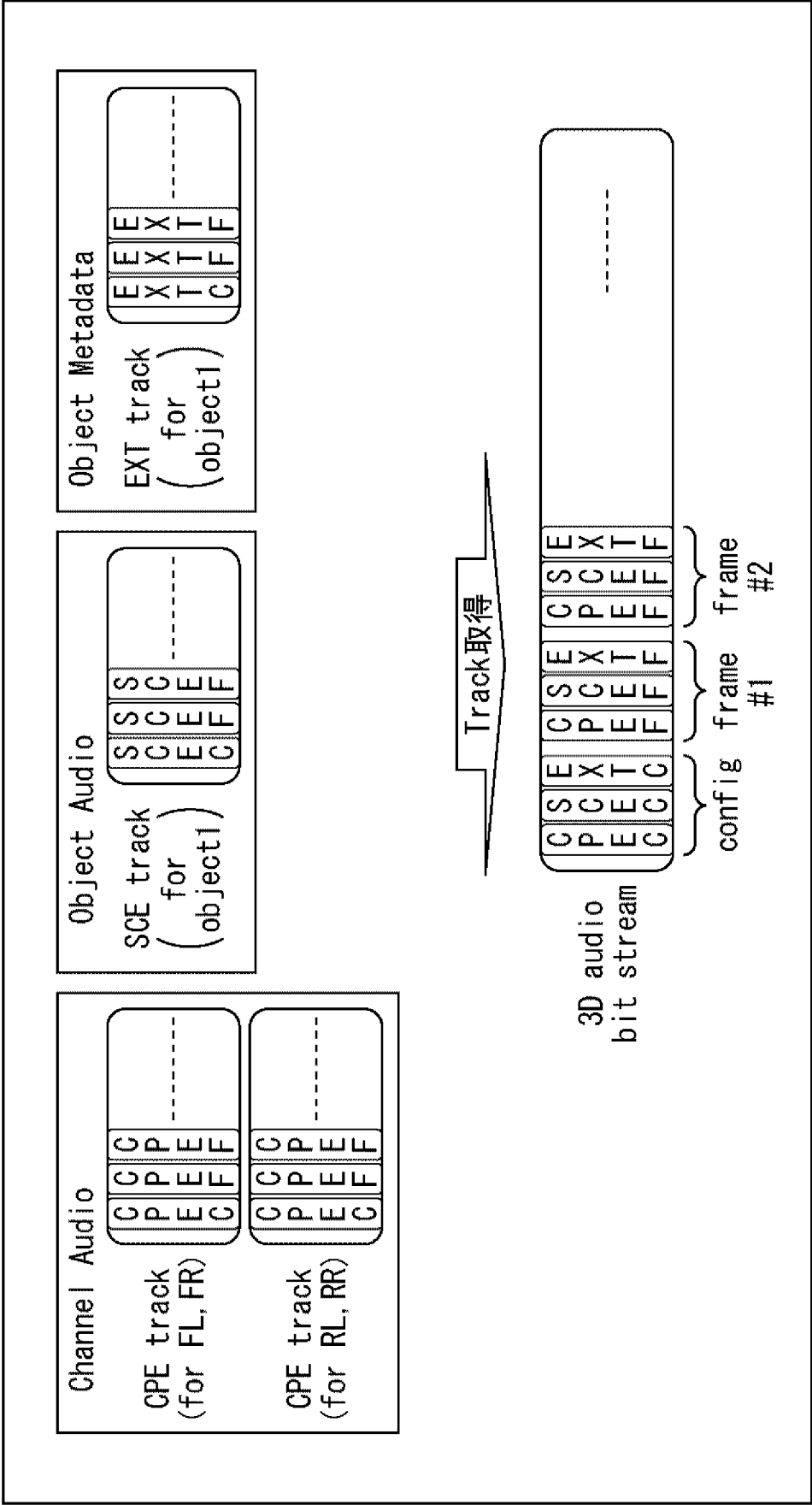


[図76]

図76

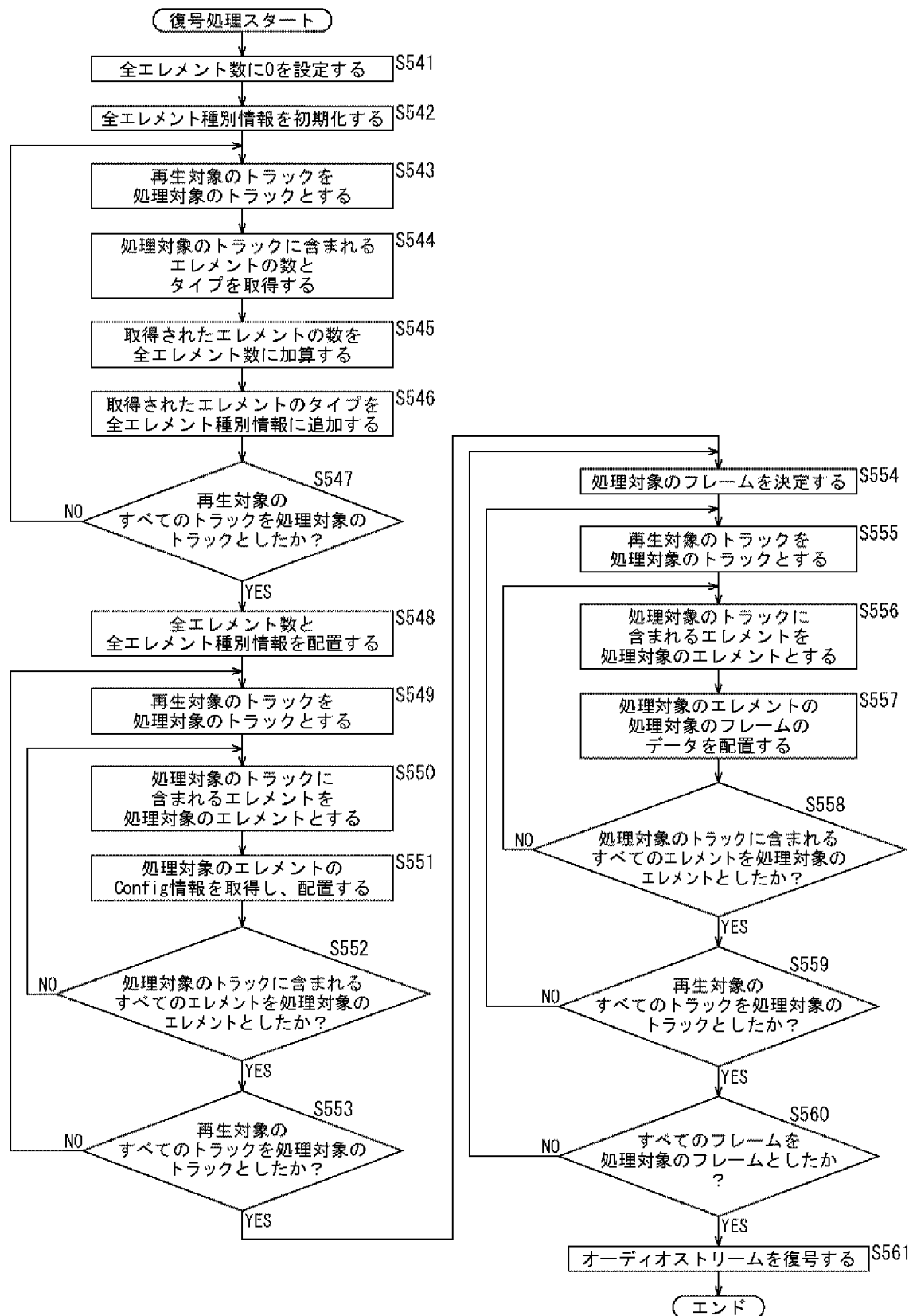


[図78]
図78



[図79]

図79



[80]
80

Syntax	No. of bits	Mnemonic
<pre>mpeg3daDecoderConfig() { numElements = escapedValue(4,8,16) + 1; elementLengthPresent 1 for (elemIdx=0; elemIdx<numElements; ++elemIdx) { usacElementType[elemIdx] switch (usacElementType[elemIdx]) { case ID_USAC_SCE: mpeg3daSingleChannelElementConfig(sbrRatioIndex); break; case ID_USAC_CPE: mpeg3daChannelPairElementConfig(sbrRatioIndex); break; case ID_USAC_LFE: mpeg3daLfeElementConfig(); break; case ID_USAC_EXT: mpeg3daExtElementConfig(); break; } } }</pre>		uimbsf
NOTE: mpeg3daSingleChannelElementConfig(), mpeg3daChannelPairElementConfig(), mpeg3daLfeElementConfig() and mpeg3daExtElementConfig() signaled at position elemIdx refer to the corresponding elements in mpeg3daFrame() at the respective position elemIdx.		

[図81]

図81

Syntax	No. of bits	Mnemonic
<pre> mpegh3daExtElementConfig() { usacExtElementType = escapedValue(4, 8, 16); usacExtElementConfigLength = escapedValue(4, 8, 16); if (usacExtElementDefaultLengthPresent) { usacExtElementDefaultLength = escapedValue(8, 16, 0) + 1; } else { usacExtElementDefaultLength = 0; } usacExtElementPayloadFrag; switch (usacExtElementType) { case ID_EXT_ELE_FILL: break; case ID_EXT_ELE_MPEGS: SpatialSpecificConfig(); break; case ID_EXT_ELE_SAOC: SAOCSpecificConfig(); break; case ID_EXT_ELE_SAOC_3D: SAOC3DSpecificConfig(); break; case ID_EXT_ELE_AUDIOPREROLL: break; case ID_EXT_ELE_UNI_DRC: mpegH3daUniDrcConfig(); break; case ID_EXT_ELE_OBJ_METADATA: ObjectMetadataConfig(); break; case ID_EXT_ELE_HOA: HOAConfig(); break; case ID_EXT_ELE_EXTRACTOR: ExtractorConfig(); break; default: while (usacExtElementConfigLength-- > 0) { tmp; } break; } } </pre>	1 1 NOTE 8	uimbsf uimbsf uimbsf

[図82]
図82

Syntax	No. of bits	Mnemonic
<pre>ExtractorConfig() { usacElementType Extractor; switch (usacElementType[elemIdx]) { case ID_USAC_SCE: break; case ID_USAC_CPE: break; case ID_USAC_LFE: break; case ID_USAC_EXT: usacExtElementTypeExtractor = escapedValue(4, 8, 16); break; } configLength; configOffset; }</pre>	2	uimbsf

[83]
 83

usacExtElementType	The concatenated usacExtElementSegmentData represents:
ID_EXT_ELE_FIL	Series of fill_byte
ID_EXT_ELE_MPEGS	SpatialFrame()
ID_EXT_ELE_SAOC	SaocFrame()
ID_EXT_ELE_AUDIOPREROLL	AudioPreRoll()
ID_EXT_ELE_UNI_DRC	uniDrcGain() as defined in ISO/IEC 23003-4
ID_EXT_ELE_OBJ_METADATA	object_metadata()
ID_EXT_ELE_SAOC_3D	Saoc3DFrame()
ID_EXT_ELE_HOA	HOAFrame()
ID_EXT_ELE_EXTRACTOR	ExtractorMetadata()
unknown	unknown data. The data block shall be discarded.

[図84]
図84

Syntax	No. of bits	Mnemonic
ExtractorMetadata() { elementLength; elementOffset; }	16 64	uimsbf uimsbf

[図85]

図85

Syntax	No. of bits	Mnemonic
<pre> mpegh3daDecoderConfig() { numElements = escapedValue(4, 8, 16) + 1; elementLengthPresent flagExtractor for (elemIdx=0; elemIdx<numElements; ++elemIdx) { usacElementType[elemIdx] switch (usacElementType[elemIdx]) { case ID_USAC_SCE: if (!flagExtractor) { mpegh3daSingleChannelElementConfig(sbrRatioIndex); } break; case ID_USAC_CPE: if (!flagExtractor) { mpegh3daChannelPairElementConfig(sbrRatioIndex); } break; case ID_USAC_LFE: if (!flagExtractor) { mpegh3daLfeElementConfig(); } break; case ID_USAC_EXT: if (!flagExtractor) { usacExtElementType = escapedValue(4, 8, 16); } else { mpegh3daExtElementConfig(); } break; } config length config offset } } </pre>	<pre> 1 1 2 </pre>	<pre> uimsbf </pre>

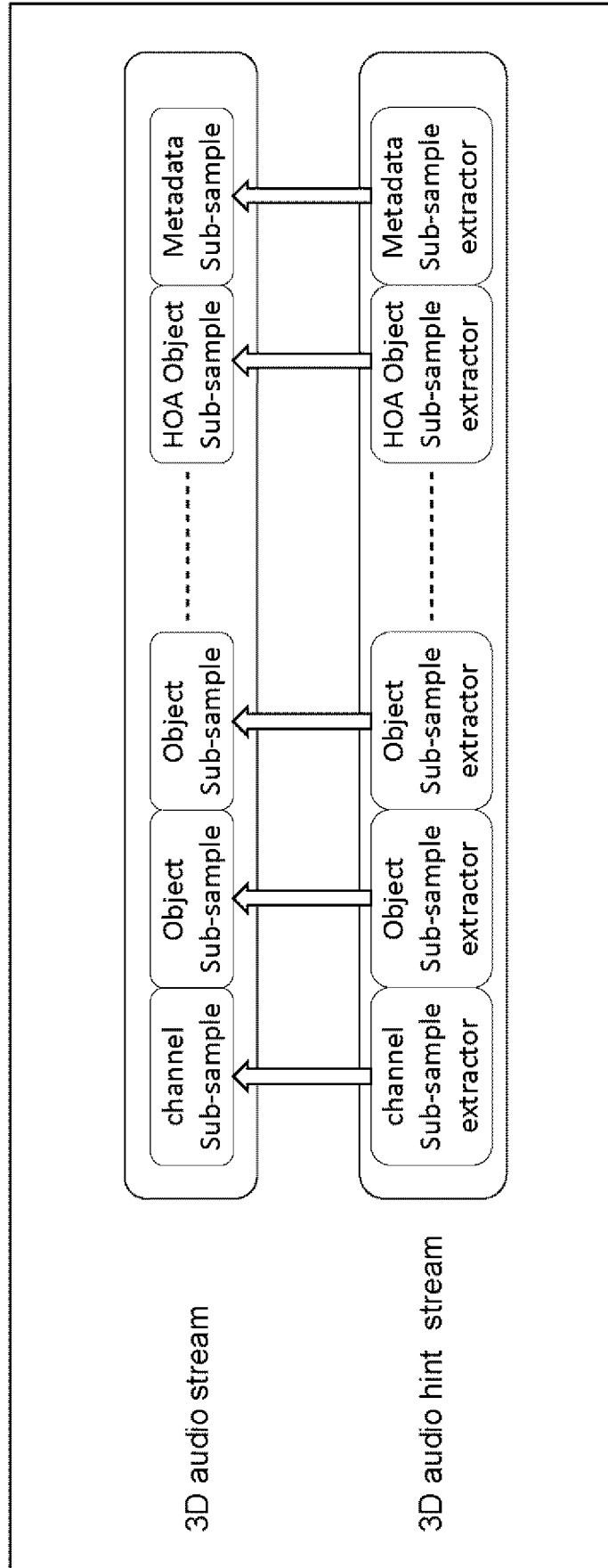
[図86]

図86

Syntax	No. of bits	Mnemonic
<pre> mpegh3daFrame() { usacIndependencyFlag; for (elemIdx=0; elemIdx<numElements; ++elemIdx) { if ((usacElementType[elemIdx] != ID_USAC_EXT) && (elementLengthPresent == 1)) { elementLength } switch (usacElementType[elemIdx]) { case ID_USAC_SCE: if (flagExtractor) { element Offset } else { mpeg3daSingleChannelElement(usacIndependencyFlag); } break; case ID_USAC_CPE: if (flagExtractor) { element Offset } else { mpeg3daChannelPairElement(usacIndependencyFlag); } break; case ID_USAC_LFE: if (flagExtractor) { element Offset } else { mpeg3daLfeElement(usacIndependencyFlag); } break; case ID_USAC_EXT: if (flagExtractor) { element Length element Offset } else { mpeg3daExtElement(usacIndependencyFlag); } break; } } } </pre>	1 16 64 64 64 64 16 64	uimbsf uimbsf uimbsf elementLength, NOTE 1 uimbsf elementLength, NOTE 1 uimbsf elementLength, NOTE 1 uimbsf uimbsf
NOTE 1: If present, elementLength represents the length of the corresponding element it refers to in number of bits.		

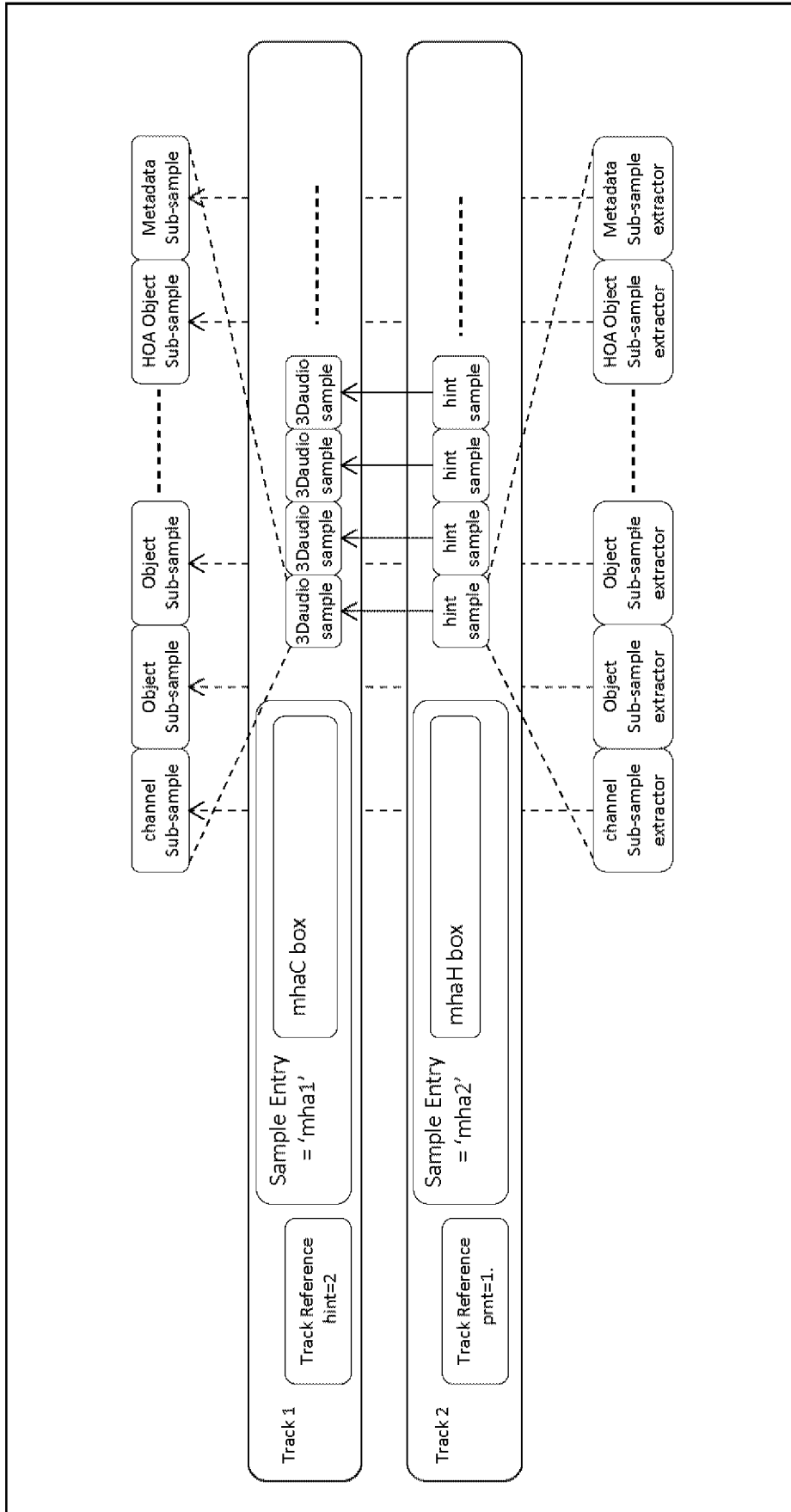
[図87]

図87



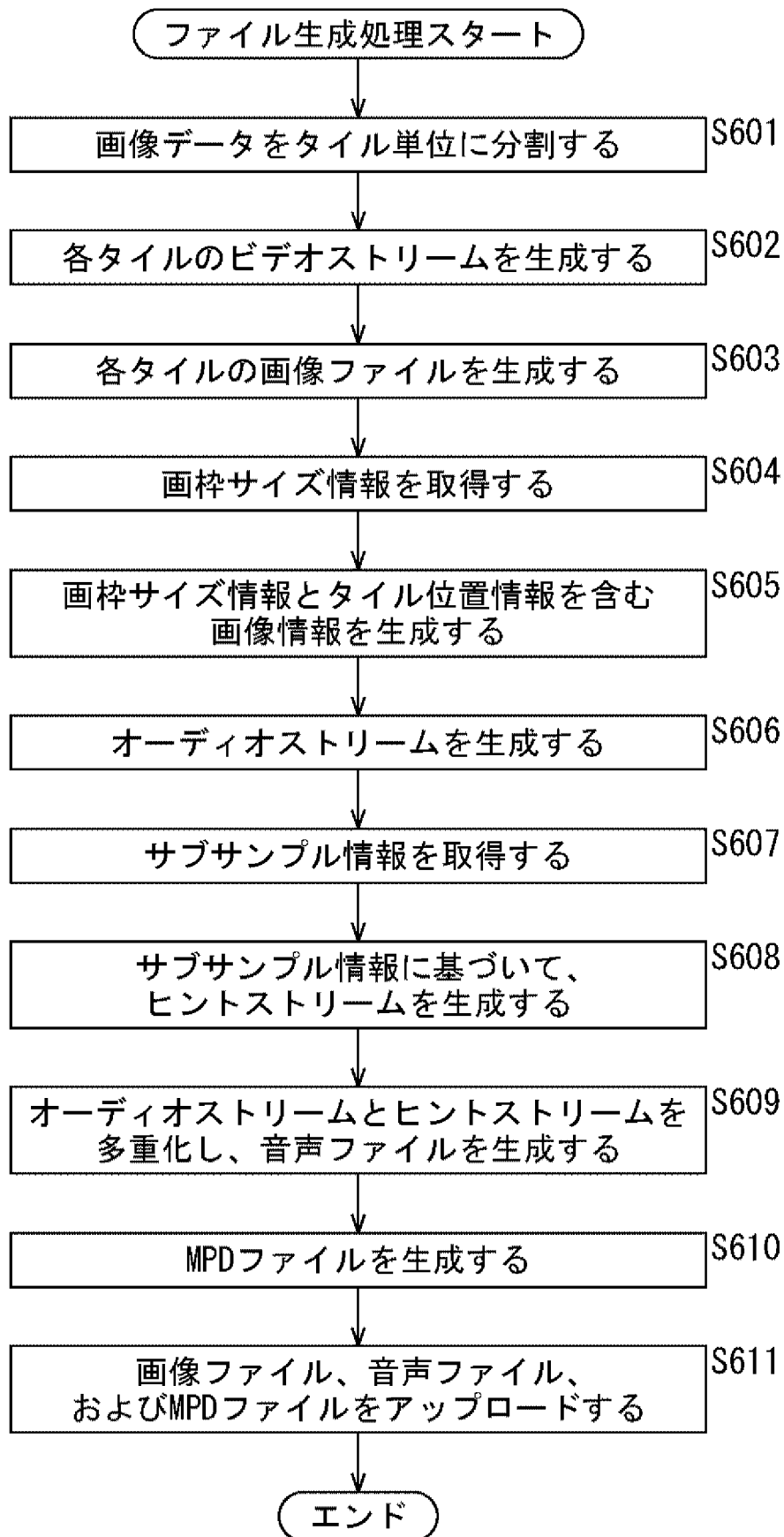
[図88]

図88



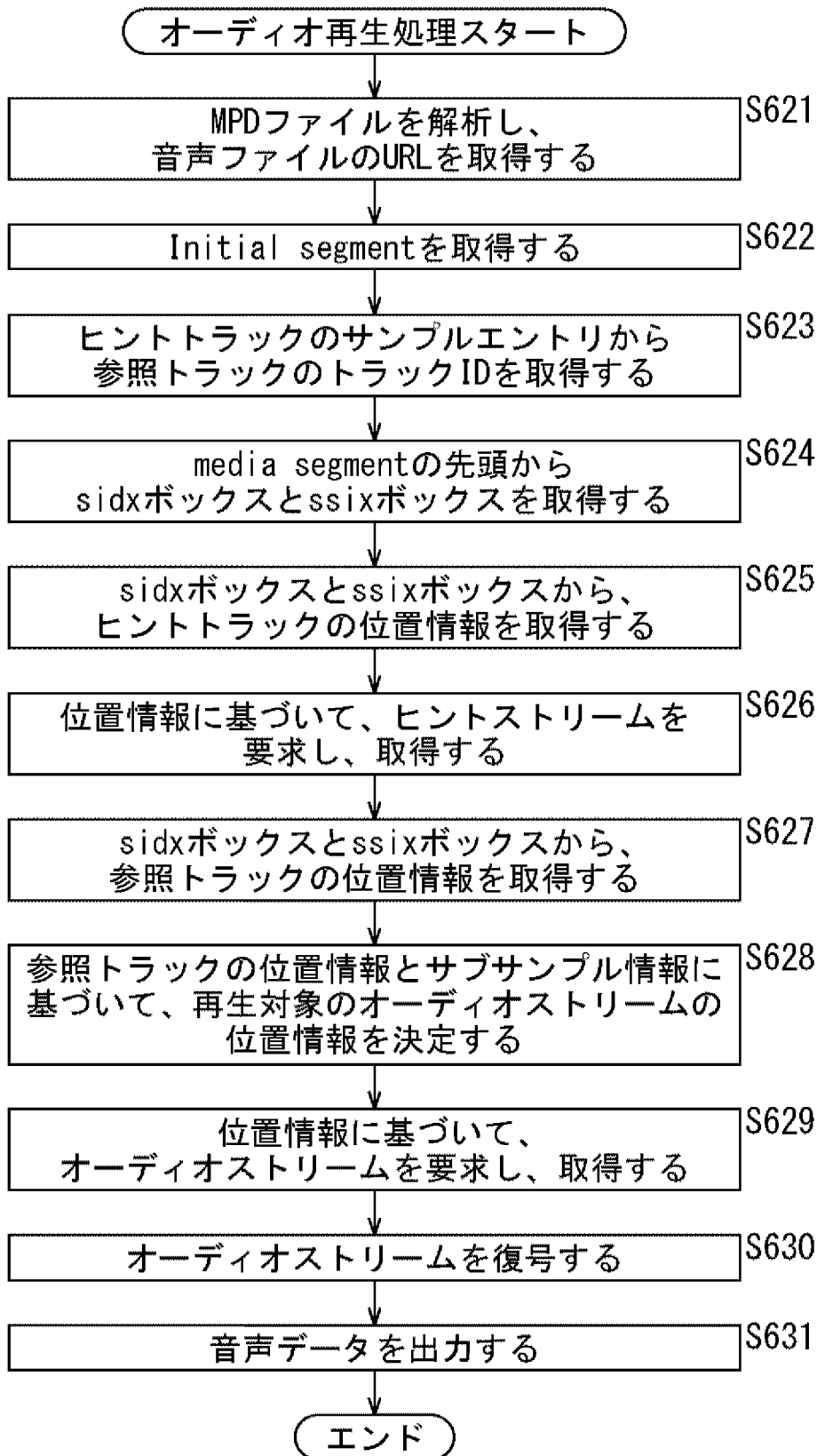
[図89]

図89



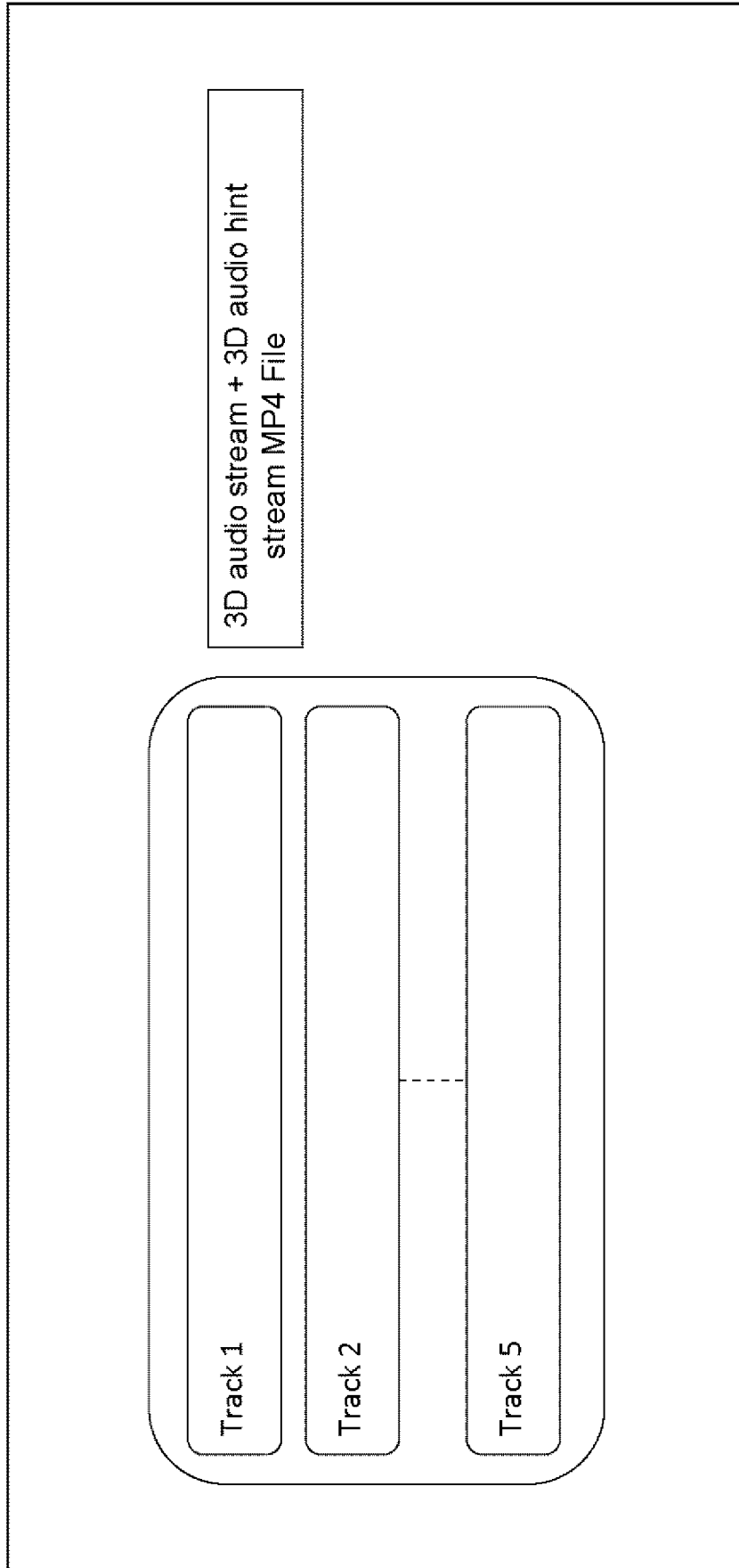
[図90]

図90



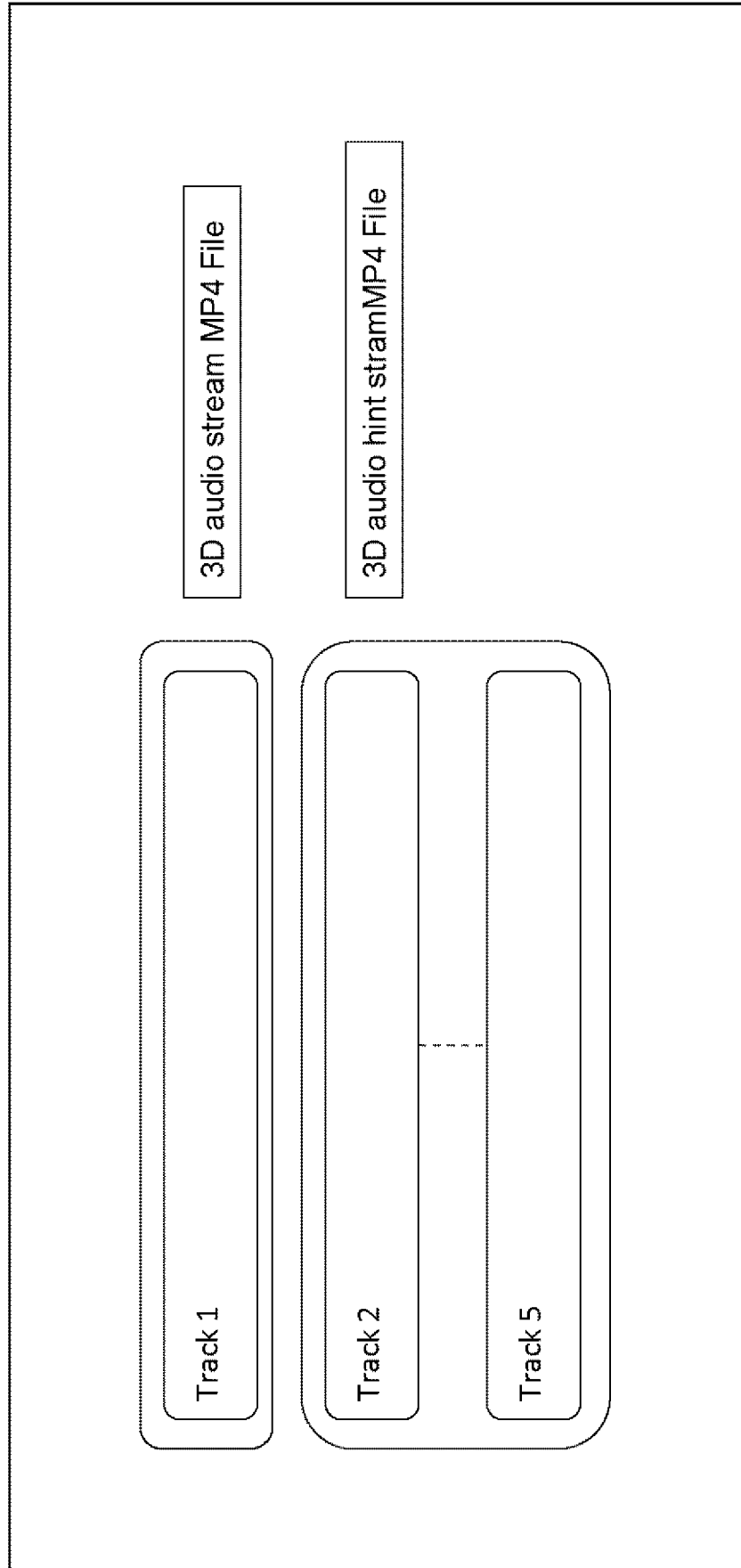
[図92]

図92



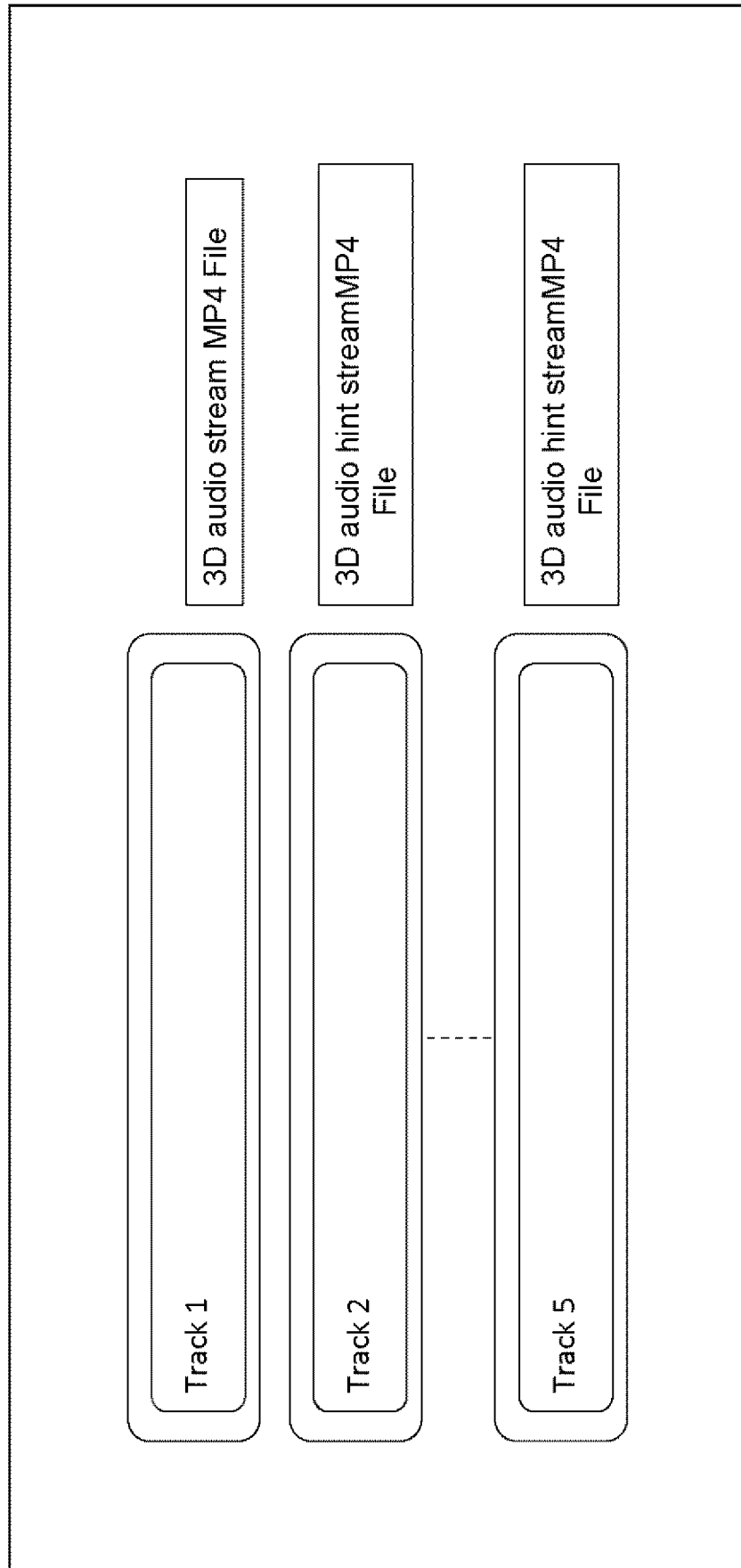
[図93]

図93



[図94]

図94



[図95]

図95

