

FIG. 1

[illegible]

FIG. 3

21 {
aligned(8) class SubPictureCompositionBox extends TrackGroupTypeBox('spco') {
 SubPictureRegionBox();
 RegionWisePackingBox(); // optional
 // optional boxes
}

FIG. 4

22 {

```
aligned(8) class SubPictureRegionBox extends FullBox(' sprg', 0, 0) {  
    unsigned int(16) track_x;  
    unsigned int(16) track_y;  
    unsigned int(16) track_width;  
    unsigned int(16) track_height;  
    unsigned int(16) composition_width;  
    unsigned int(16) composition_height;  
}
```

FIG. 5

23 {

track_x	HORIZONTAL POSITION, ON ENTIRE PICTURE, OF sub-picture STORED IN track
track_y	VERTICAL POSITION, ON ENTIRE PICTURE, OF sub-picture STORED IN track
track_width	WIDTH OF sub-picture STORED IN track
track_height	HEIGHT OF sub-picture STORED IN track
composition_width	WIDTH OF ENTIRE PICTURE
composition_height	HEIGHT OF ENTIRE PICTURE

FIG. 6

24 {

```
aligned(8) class RegionWisePackingBox extends FullBox('rwpk', 0, 0) {  
    RegionWisePackingStruct();  
}
```

FIG. 7

```
25 {
    aligned(8) class RegionWisePackingStruct() {
        unsigned int(1) constituent_picture_matching_flag;
        bit(7) reserved = 0;
        unsigned int(8) num_regions;
        unsigned int(32) proj_picture_width;
        unsigned int(32) proj_picture_height;
        unsigned int(16) packed_picture_width;
        unsigned int(16) packed_picture_height;
        for (i = 0; i < num_regions; i++) {
            bit(3) reserved = 0;
            unsigned int(1) guard_band_flag[i];
            unsigned int(4) packing_type[i];
            if (packing_type[i] == 0) {
                RectRegionPacking(i);
                if (guard_band_flag[i])
                    GuardBand(i);
            }
        }
    }
}
```

FIG. 8

26

constituent_picture_matching_flag	FLAG INDICATING WHETHER OR NOT SAME region-wise packing IS APPLIED TO Left view AND Right view IN CASE WHERE PICTURE IS STEREOSCOPIC. 0: MONOSCOPIC OR DIFFERENT packings ARE APPLIED TO Left view AND Right view. 1: SAME packing IS APPLIED TO Left view AND Right view.
num_regions	NUMBER OF packed regions
proj_picture_width	WIDTH OF projected picture
proj_picture_height	HEIGHT OF projected picture
packed_picture_width	WIDTH OF packed picture
packed_picture_height	HEIGHT OF packed picture
guard_band_flag[i]	0: NO GUARD BAND IS PRESENT IN packed region 1: GUARD BAND IS PRESENT IN packed region
packing_type[i]	SHAPE OF paced region. 0: RECTANGLE Others: reserved
GuardBand(i)	GUARD BAND INFORMATION RELATED TO PERIPHERY OF REGION

FIG. 9

27
{

```
aligned(8) class RectRegionPacking(i) {  
    unsigned int(32) proj_reg_width[i];  
    unsigned int(32) proj_reg_height[i];  
    unsigned int(32) proj_reg_top[i];  
    unsigned int(32) proj_reg_left[i];  
    unsigned int(3) transform_type[i];  
    bit(5) reserved = 0;  
    unsigned int(16) packed_reg_width[i];  
    unsigned int(16) packed_reg_height[i];  
    unsigned int(16) packed_reg_top[i];  
    unsigned int(16) packed_reg_left[i];  
}
```

FIG. 10

28 {

proj_reg_width[i]	WIDTH OF projected region CORRESPONDING TO APPLICATION SOURCE OF region-wise packing
proj_reg_height[i]	HEIGHT OF projected region CORRESPONDING TO APPLICATION SOURCE OF region-wise packing
proj_reg_top[i]	VERTICAL POSITION OF projected region CORRESPONDING TO APPLICATION SOURCE OF region-wise packing
proj_reg_left[i]	HORIZONTAL POSITION OF projected region CORRESPONDING TO APPLICATION SOURCE OF region-wise packing
transform_type[i]	INDICATES ROTATION OR MIRRORING OF packed region.
packed_reg_width[i]	WIDTH OF packed region REARRANGED BY region-wise packing
packed_reg_height[i]	HEIGHT OF packed region REARRANGED BY region-wise packing
packed_reg_top[i]	VERTICAL POSITION OF packed region REARRANGED BY region-wise packing
packed_reg_left[i]	HORIZONTAL POSITION OF packed region REARRANGED BY region-wise packing

FIG. 11

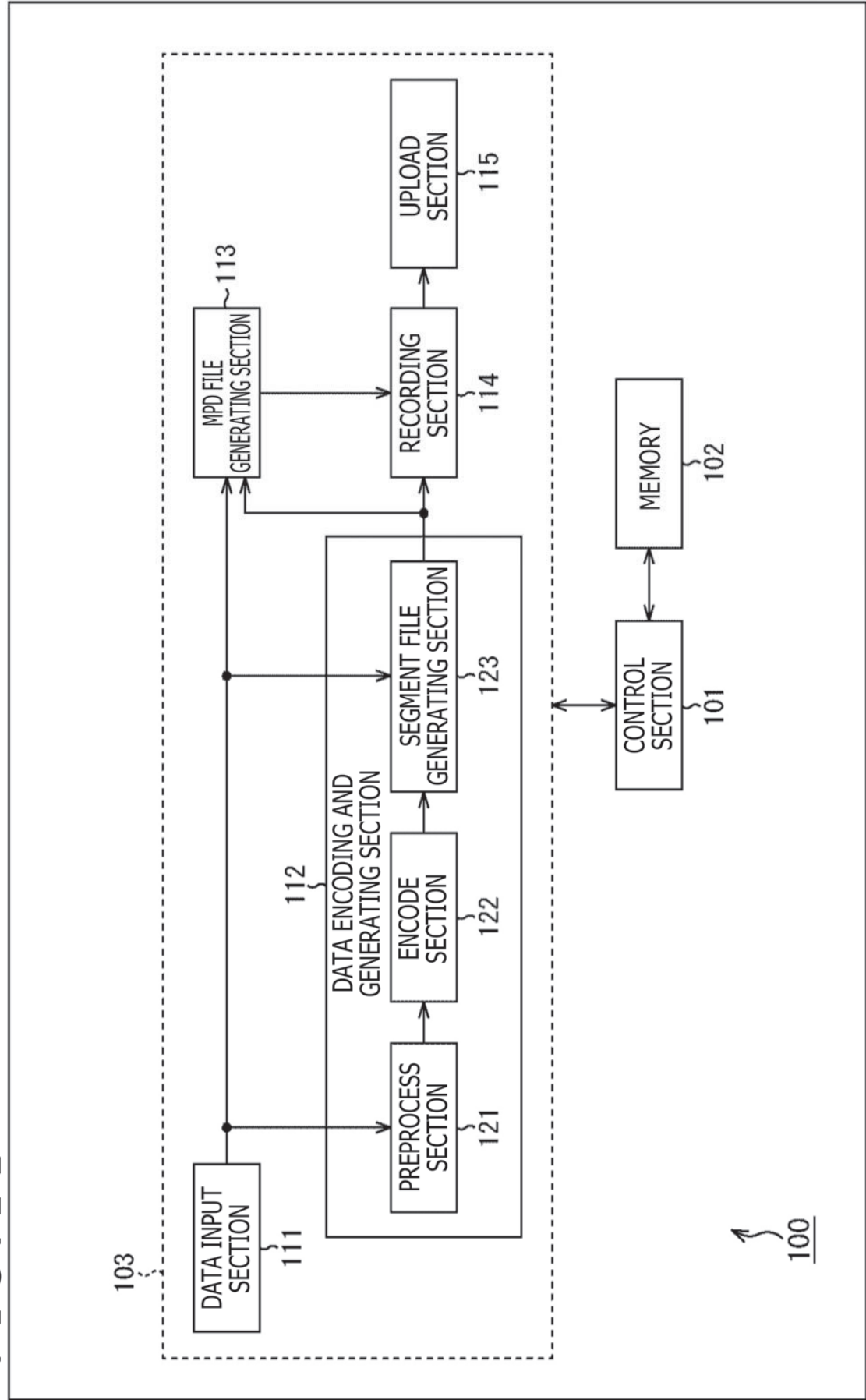


FIG. 12

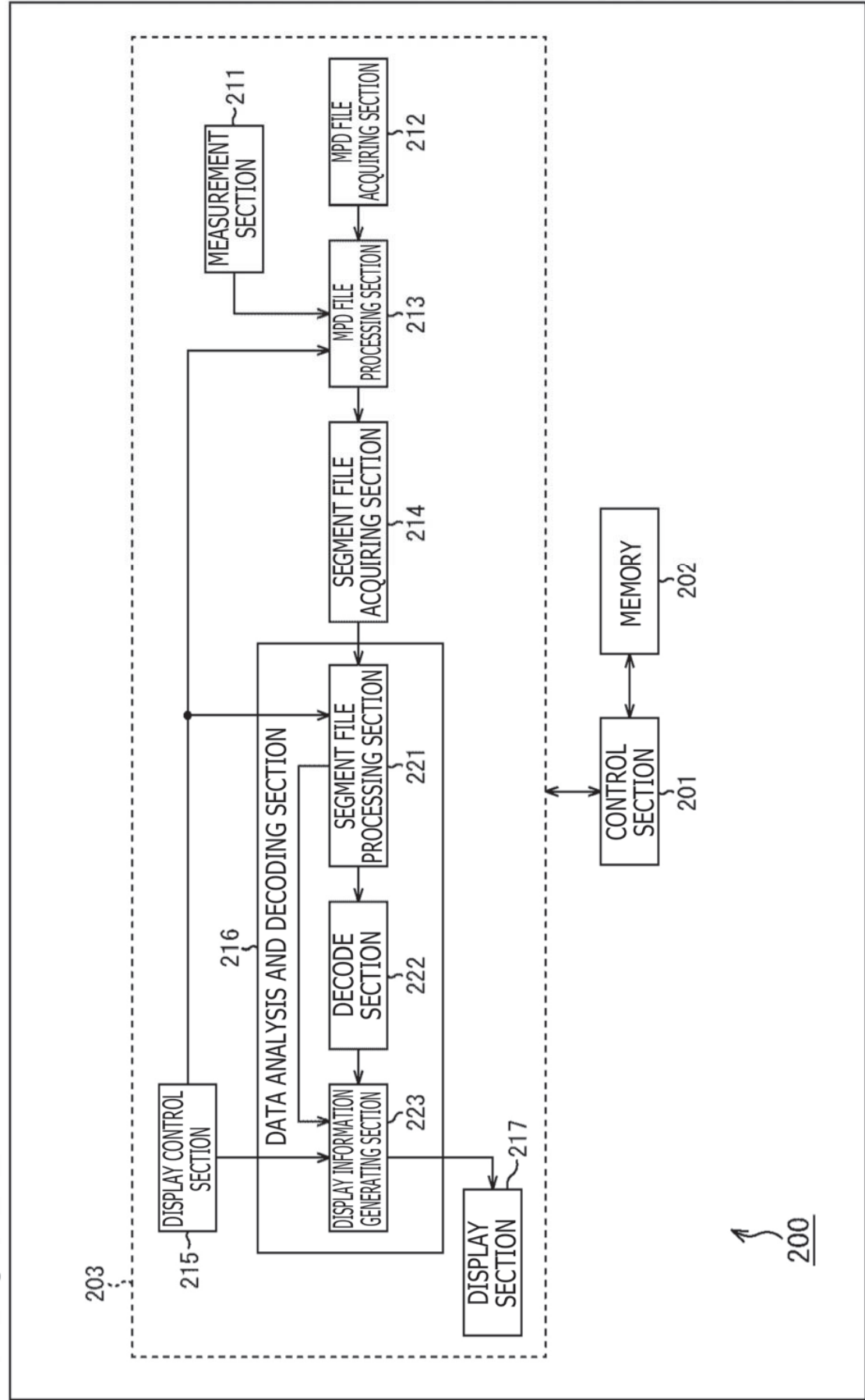


FIG. 13

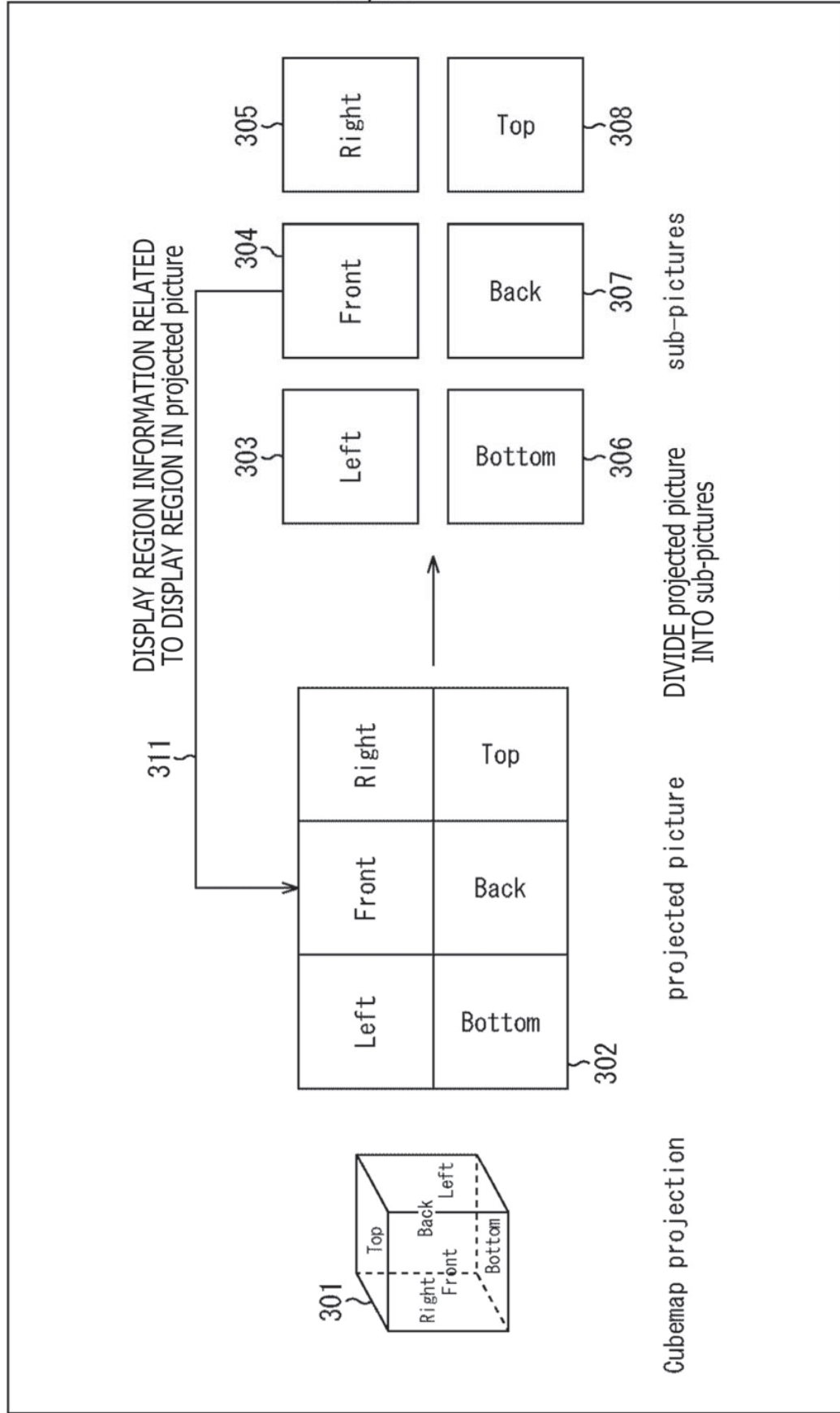


FIG. 14

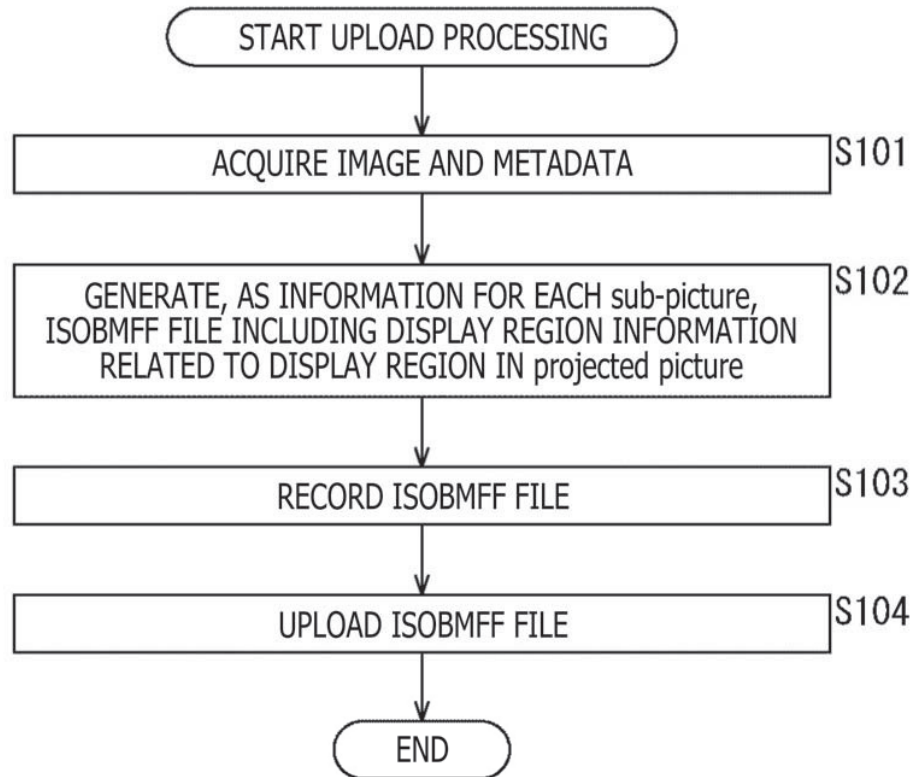


FIG. 15

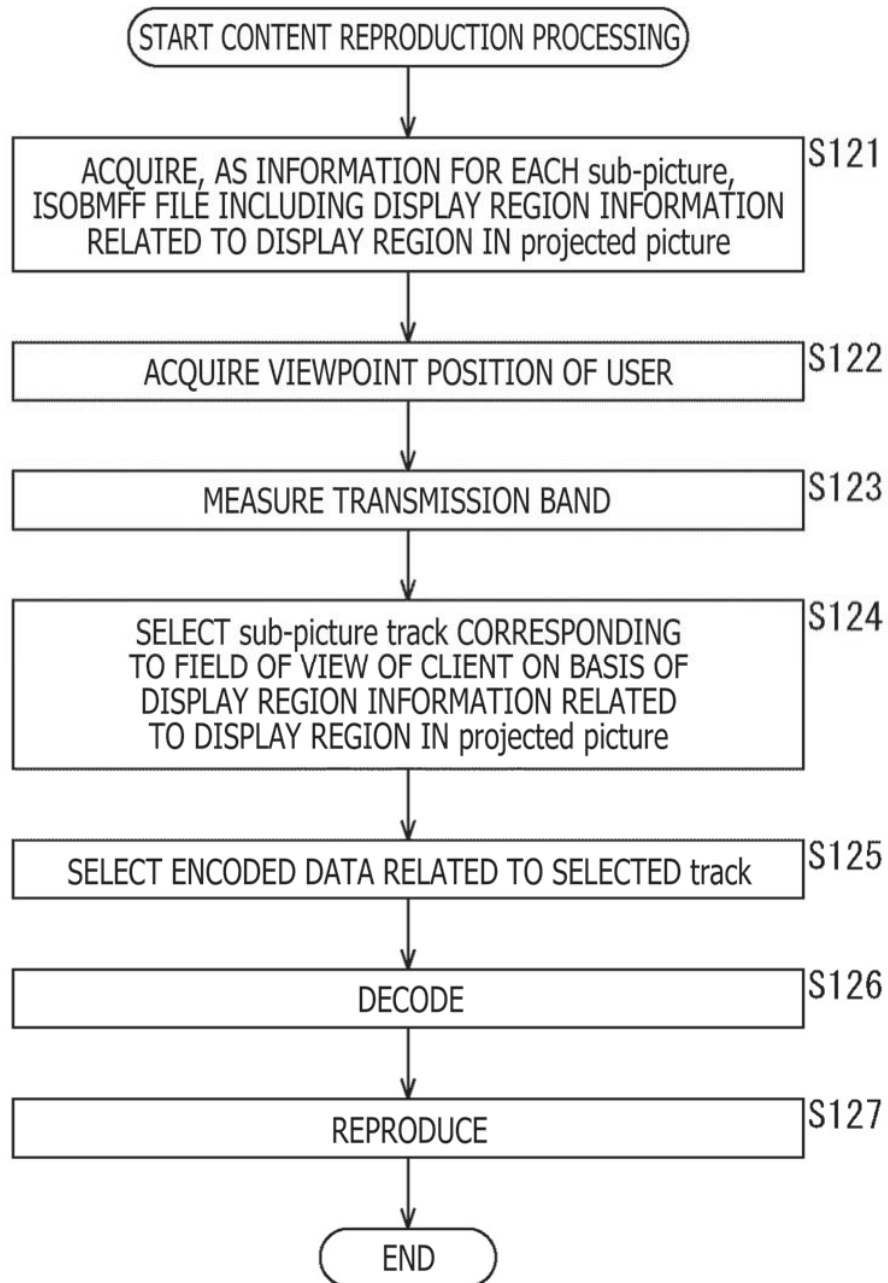


FIG. 16

331
{

```
aligned(8) class 2DCoverageInformationBox() extends FullBox('2dco', 0, 0) {  
    unsigned int(32) proj_picture_width;  
    unsigned int(32) proj_picture_height;  
    unsigned int(16) proj_reg_width;  
    unsigned int(16) proj_reg_height;  
    unsigned int(16) proj_reg_top;  
    unsigned int(16) proj_reg_left;  
}
```

FIG. 17

332

proj_picture_width	WIDTH OF projected picture.
proj_picture_height	HEIGHT OF projected picture.
proj_reg_width	WIDTH OF REGION ON projected picture TO WHICH picture IN track CORRESPONDS.
proj_reg_height	HEIGHT OF REGION ON projected picture TO WHICH picture IN track CORRESPONDS.
proj_reg_top	VERTICAL COORDINATE OF REGION ON projected picture TO WHICH picture IN track CORRESPONDS.
proj_reg_left	REGION HORIZONTAL COORDINATE ON projected picture TO WHICH picture IN track CORRESPONDS.

FIG. 18

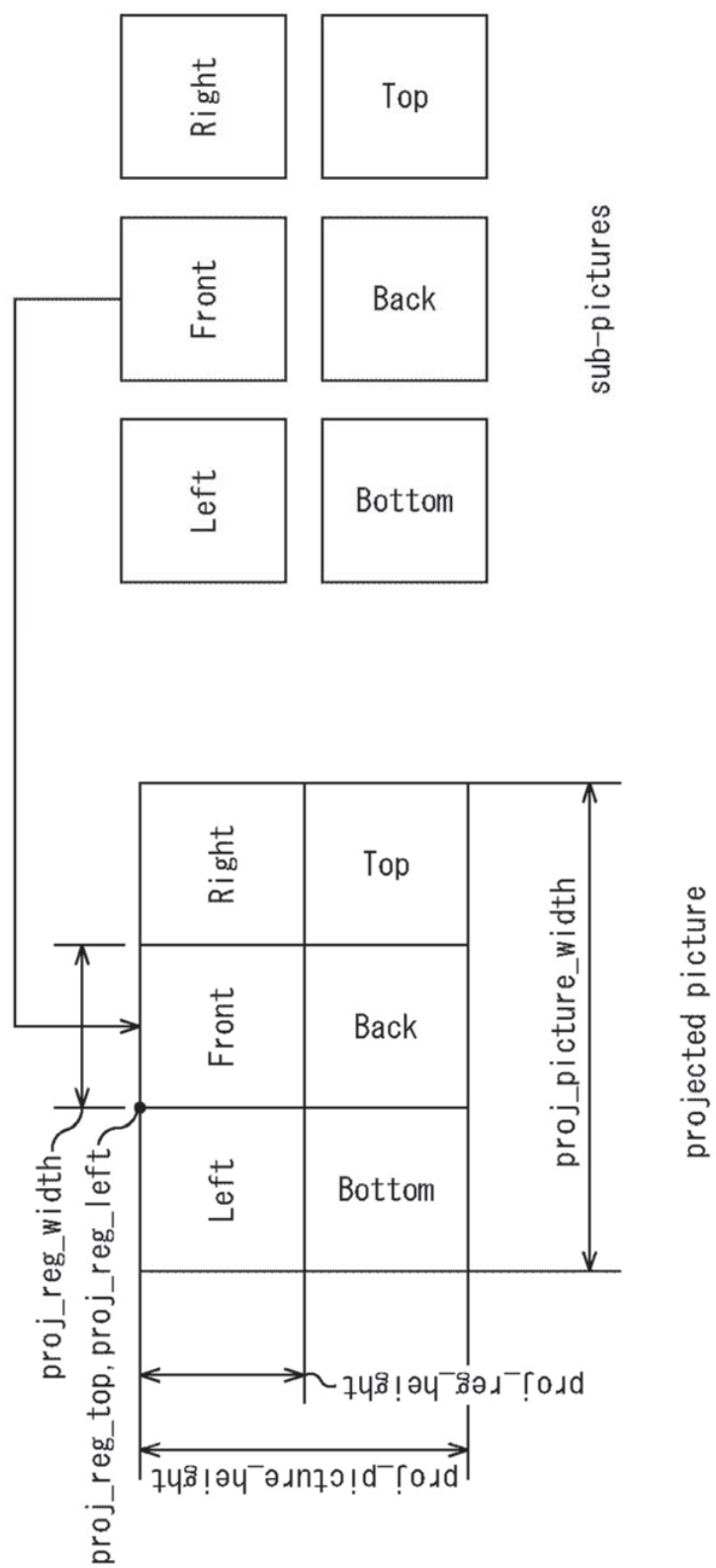


FIG. 19

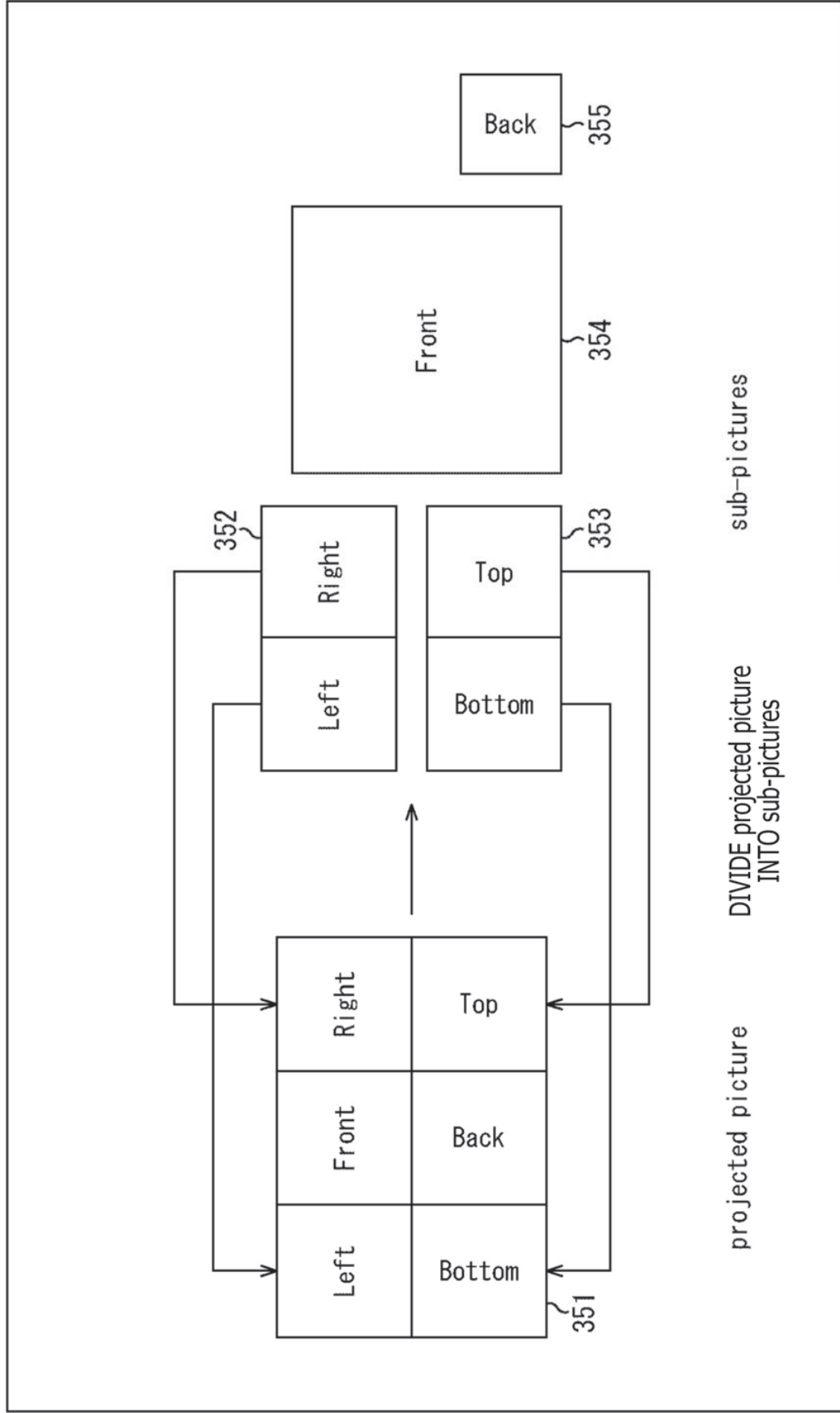


FIG. 20

```
371 {  
    aligned(8) class 2DCoverageInformationBox() extends FullBox('2dco', 0, 0) {  
        unsigned int(32) proj_picture_width;  
        unsigned int(32) proj_picture_height;  
        unsigned int(8) num_regions;  
        for(i=0; i<num_regions; i++) {  
            unsigned int(16) proj_reg_width[i];  
            unsigned int(16) proj_reg_height[i];  
            unsigned int(16) proj_reg_top[i];  
            unsigned int(16) proj_reg_left[i];  
        }  
    }  
}
```

FIG. 21

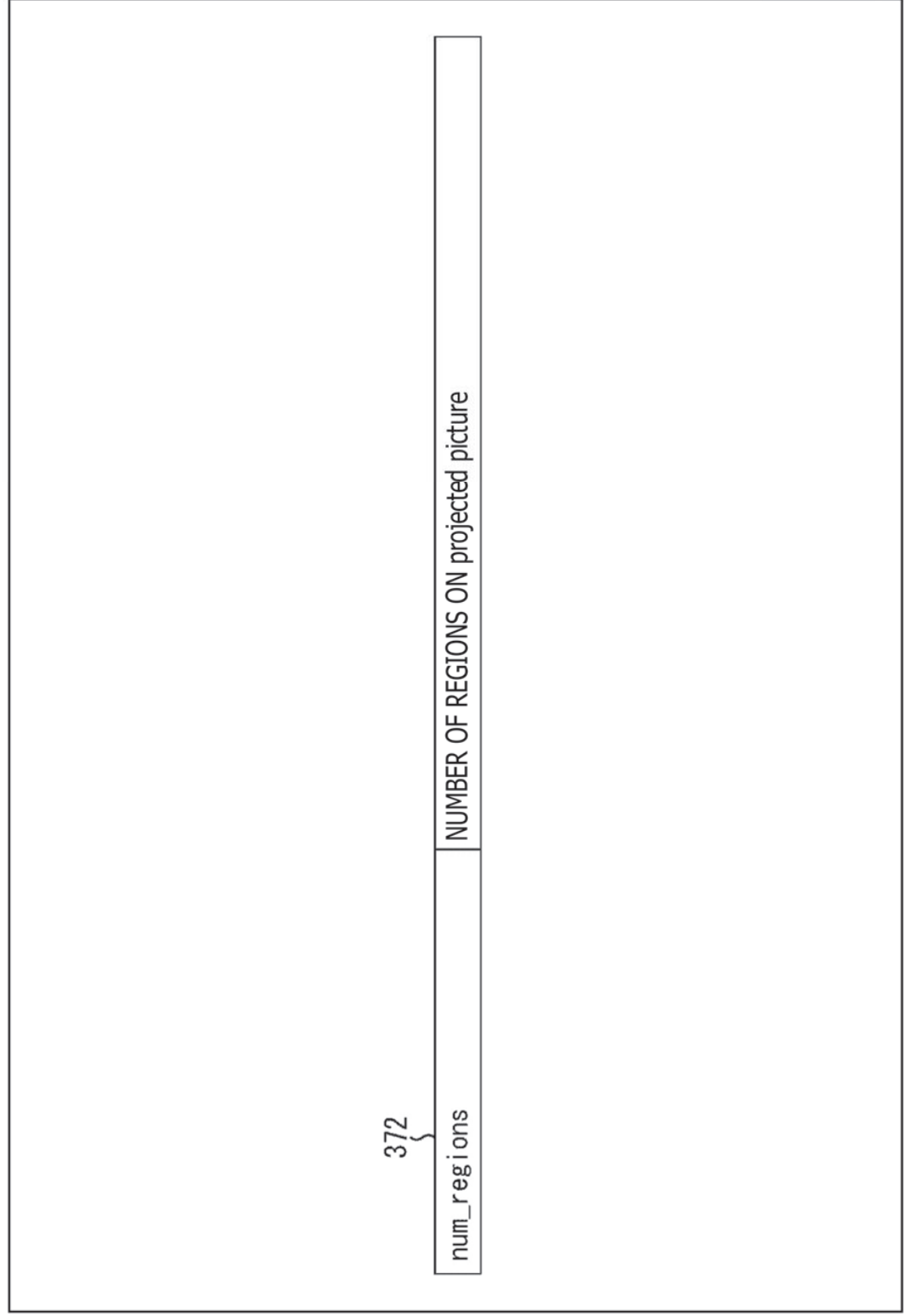


FIG. 22

```

373 aligned(8) class RegionWisePackingStruct() {
    unsigned int(1) constituent_picture_matching_flag;
    unsigned int(1) 2D_coverage_flag;
    bit(6) reserved = 0;
    unsigned int(8) num_regions;
    unsigned int(32) proj_picture_width;
    unsigned int(32) proj_picture_height;
    if(2D_coverage_flag == 0) {
        unsigned int(16) packed_picture_width;
        unsigned int(16) packed_picture_height;
    }
    for (i = 0; i < num_regions; i++) {
        if(2D_coverage_flag == 1)
            RectProjectedRegion(i);
        else {
            bit(3) reserved = 0;
            unsigned int(1) guard_band_flag[i];
            unsigned int(4) packing_type[i];
            if (packing_type[i] == 0) {
                RectRegionPacking(i);
                if (guard_band_flag[i])
                    GuardBand(i);
            }
        }
    }
}

```

FIG. 23

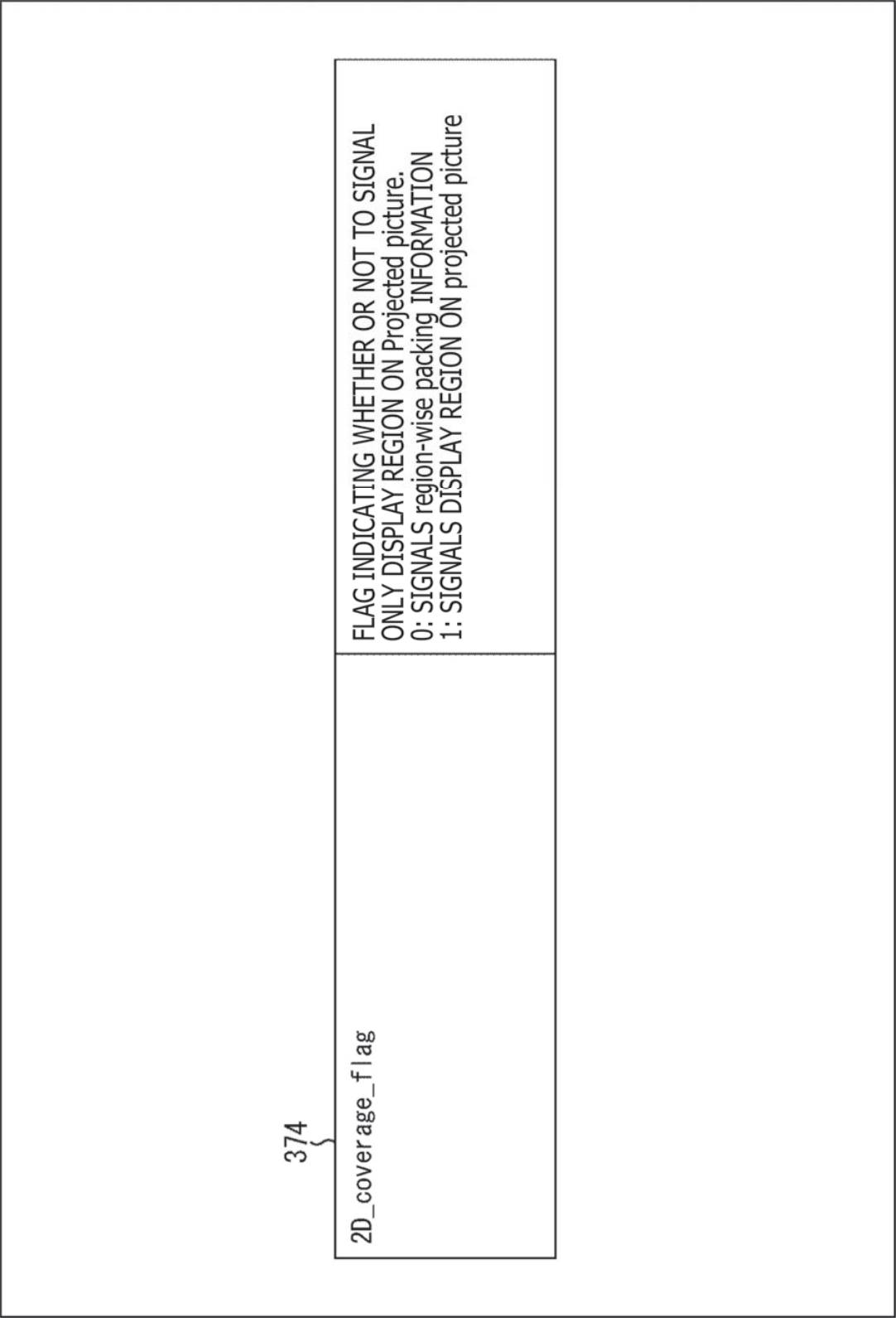


FIG. 24

375 {

```
aligned(8) class RectProjectedRegion(i) {  
    unsigned int(32) proj_reg_width[i];  
    unsigned int(32) proj_reg_height[i];  
    unsigned int(32) proj_reg_top[i];  
    unsigned int(32) proj_reg_left[i];  
}
```

FIG. 25

376
{

proj_reg_width	WIDTH OF REGION ON projected picture TO WHICH picture IN track CORRESPONDS.
proj_reg_height	HEIGHT OF REGION ON projected picture TO WHICH picture IN track CORRESPONDS.
proj_reg_top	VERTICAL COORDINATE OF REGION ON projected picture TO WHICH picture IN track CORRESPONDS.
proj_reg_left	REGION HORIZONTAL COORDINATE ON projected picture TO WHICH picture IN track CORRESPONDS.

FIG. 26

```

377 aligned(8) class RegionWisePackingStruct() {
    unsigned int(1) constituent_picture_matching_flag;
    unsigned int(1) 2D_coverage_flag;
    bit(6) reserved = 0;
    unsigned int(8) num_regions;
    unsigned int(32) proj_picture_width;
    unsigned int(32) proj_picture_height;
    if(2D_coverage_flag == 0) {
        unsigned int(16) packed_picture_width;
        unsigned int(16) packed_picture_height;
    }
    for (i = 0; i < num_regions; i++) {
        bit(3) reserved = 0;
        unsigned int(1) guard_band_flag[i];
        unsigned int(4) packing_type[i];
        if (packing_type[i] == 0) {
            RectRegionPacking(i, 2D_coverage_flag);
            if (guard_band_flag[i])
                GuardBand(i);
        }
    }
}

```

FIG. 27

```
378 {
    aligned(8) class RectRegionPacking(i, proj_region_flag) {
        unsigned int(32) proj_reg_width[i];
        unsigned int(32) proj_reg_height[i];
        unsigned int(32) proj_reg_top[i];
        unsigned int(32) proj_reg_left[i];
        if (proj_region_flag == 0) {
            unsigned int(3) transform_type[i];
            bit(5) reserved = 0;
            unsigned int(16) packed_reg_width[i];
            unsigned int(16) packed_reg_height[i];
            unsigned int(16) packed_reg_top[i];
            unsigned int(16) packed_reg_left[i];
        }
    }
}
```

FIG. 28

```
379 {  
    aligned(8) class CoverageInformationBox extends FullBox('covi', 0, 0) {  
        unsigned int(1) 2D_coverage_flag;  
        bit(7) reserved = 0;  
        if(2D_coverage_flag==0)  
            ContentCoverageStruct();  
        else  
            2DContentCoverageStruct();  
    }  
}
```

FIG. 29

380 {

2D_coverage_flag	SIGNAL TYPE OF DISPLAY REGION INFORMATION. 0: SIGNAL SPHERICAL-SURFACE DISPLAY REGION INFORMATION 1: SIGNAL DISPLAY REGION ON projected picture
ContentCoverageStruct ()	SIGNAL SPHERICAL-SURFACE DISPLAY REGION IN track
2DContentCoverageStruct ()	SIGNAL DISPLAY REGION ON projected picture IN track. FIELDS IN 2D Content Coverage Struct ARE SIMILAR TO FIELDS IN 2D Coverage Information Box IN FIG. 8.

FIG. 30

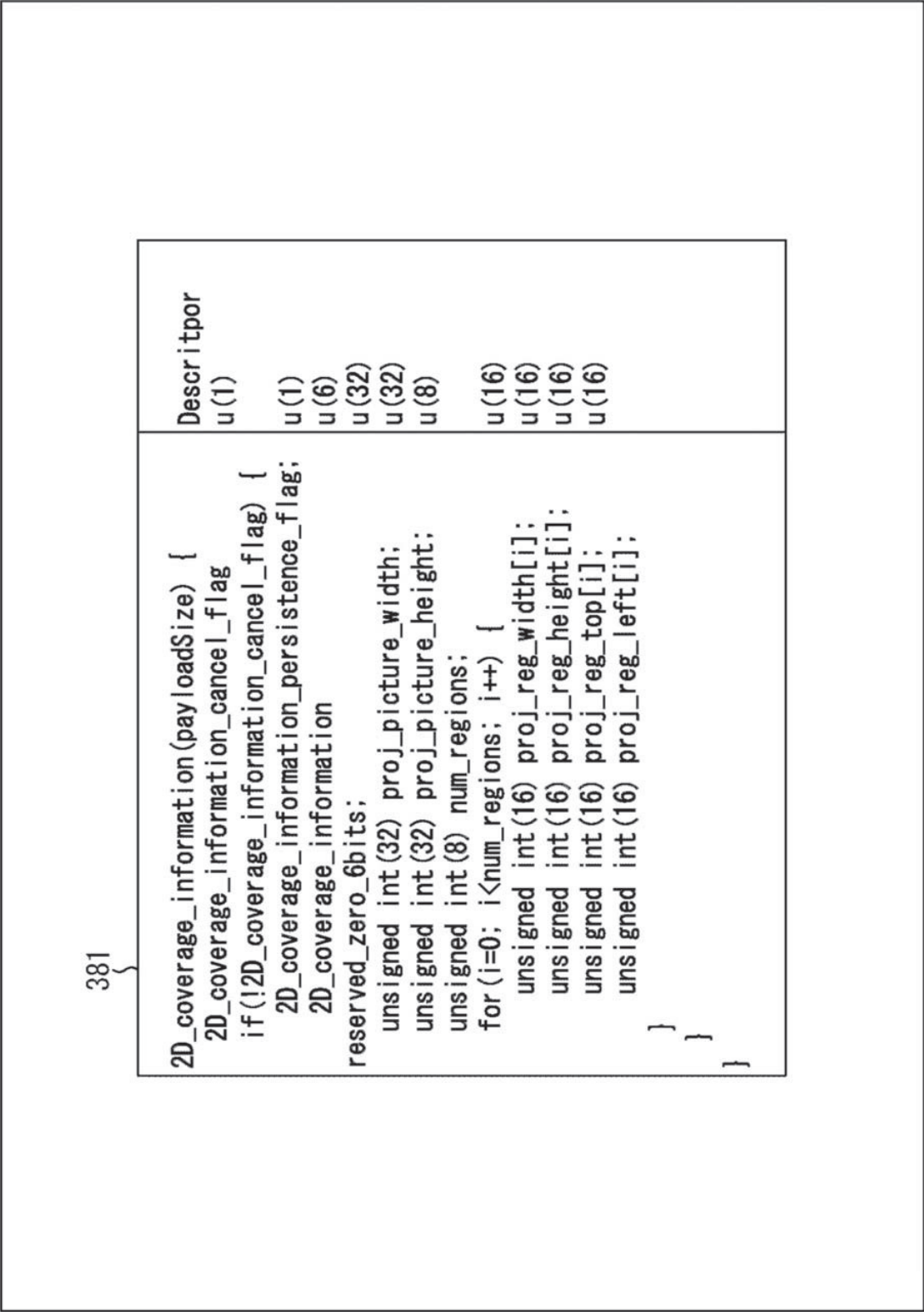


FIG. 31

2D_coverage_information_cancel_flag	FOR FLAG = 1, PERSISTENT APPLICATION OF SEI PRECEDING IN ORDER OF OUTPUT IS CANCELLED. FOR FLAG = 0, 2D coverage INFORMATION IS SIGNALLED.
2D_coverage_information_persistence_flag	FOR FLAG = 0, SEI INFORMATION IS APPLIED ONLY TO PICTURE INCLUDING SEI. FOR FLAG = 1, APPLICATION OF SEI IS PERSISTENT UNTIL NEW coded video sequence IS STARTED OR END OF STREAM IS REACHED.
2D_coverage_information_reserved_zero_6bits	FILLED WITH 0.
proj_picture_width	WIDTH OF projected picture.
proj_picture_height	HEIGHT OF projected picture.
num_regions	NUMBER OF REGIONS ON projected picture.
proj_reg_width	WIDTH OF REGION ON projected picture TO WHICH STREAM CORRESPONDS.
proj_reg_height	HEIGHT OF REGION ON projected picture TO WHICH STREAM CORRESPONDS.
proj_reg_top	VERTICAL COORDINATE OF REGION ON projected picture TO WHICH STREAM CORRESPONDS.
proj_reg_left	REGION HORIZONTAL COORDINATE ON projected picture TO WHICH STREAM CORRESPONDS.

FIG. 32

```
383 {  
    class 2DCoverageInformationSampleEntry() extends  
    MetaDataSampleEntry(2dco) {  
        unsigned int(32) proj_picture_width;  
        unsigned int(32) proj_picture_height;  
        Box[] other_boxes; // optional  
    }  
}
```

FIG. 33

384

{

```
aligned(8) 2DCoverageInformationSample() {  
    unsigned int(8) num_regions;  
    for (i=0; i<num_regions; i++) {  
        unsigned int(16) proj_reg_width[i];  
        unsigned int(16) proj_reg_height[i];  
        unsigned int(16) proj_reg_top[i];  
        unsigned int(16) proj_reg_left[i];  
    }  
}
```

FIG. 34

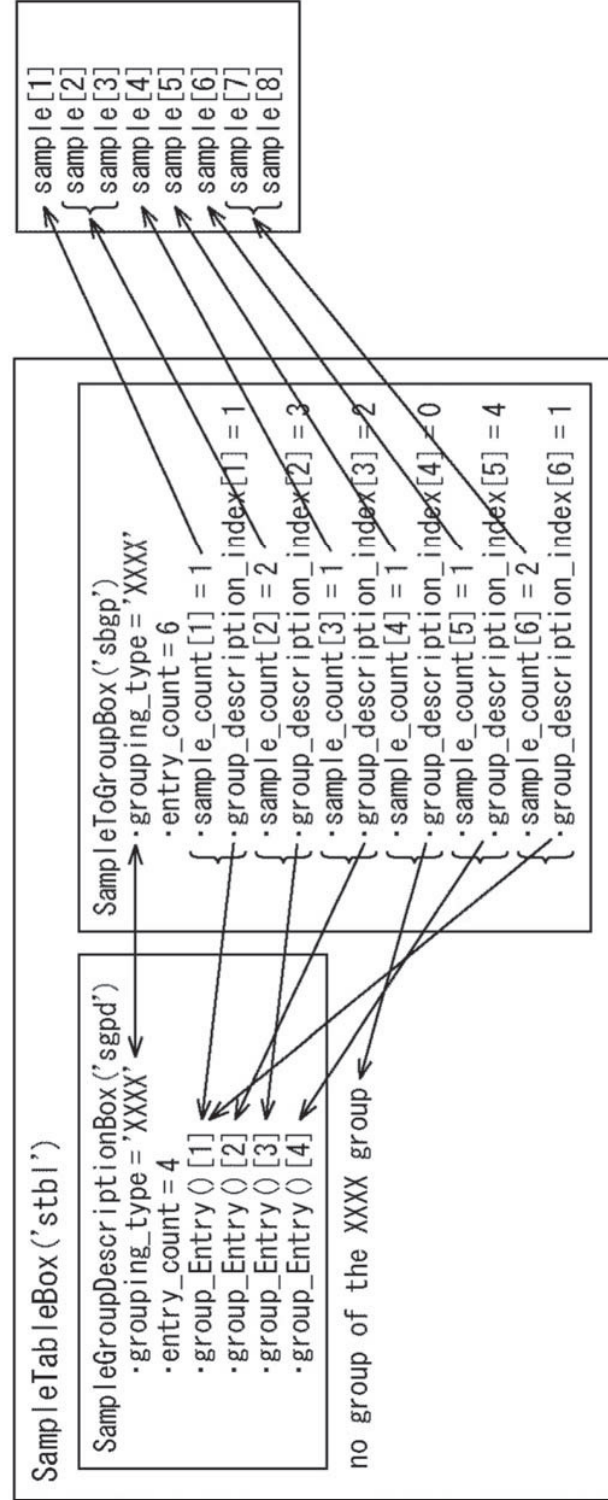


FIG. 35

```
391 {  
    class 2DCoverageInformationSampleGroupEntry extends VisualSampleGroupEntry ('2cgp') {  
        unsigned int(32) proj_picture_width;  
        unsigned int(32) proj_picture_height;  
        unsigned int(8) num_regions;  
        for(i=0; i<num_regions; i++) {  
            unsigned int(16) proj_reg_width[i];  
            unsigned int(16) proj_reg_height[i];  
            unsigned int(16) proj_reg_top[i];  
            unsigned int(16) proj_reg_left[i];  
        }  
    }  
}
```

FIG. 36

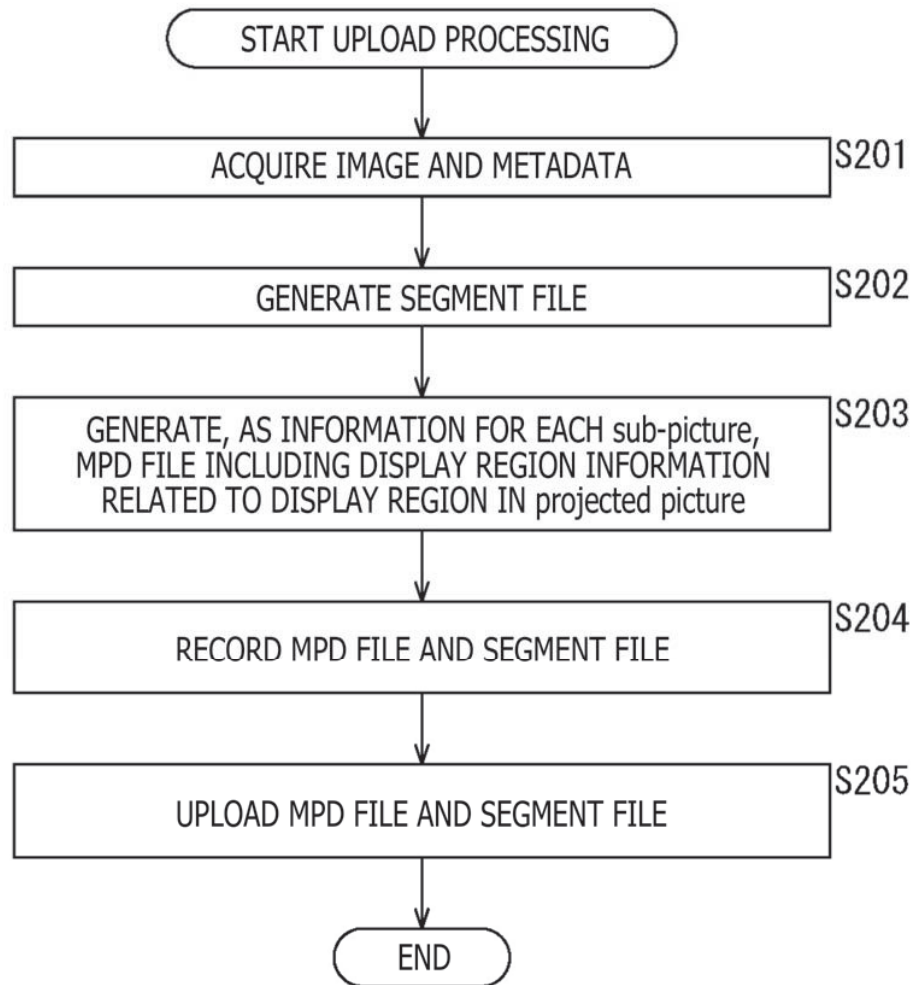


FIG. 37

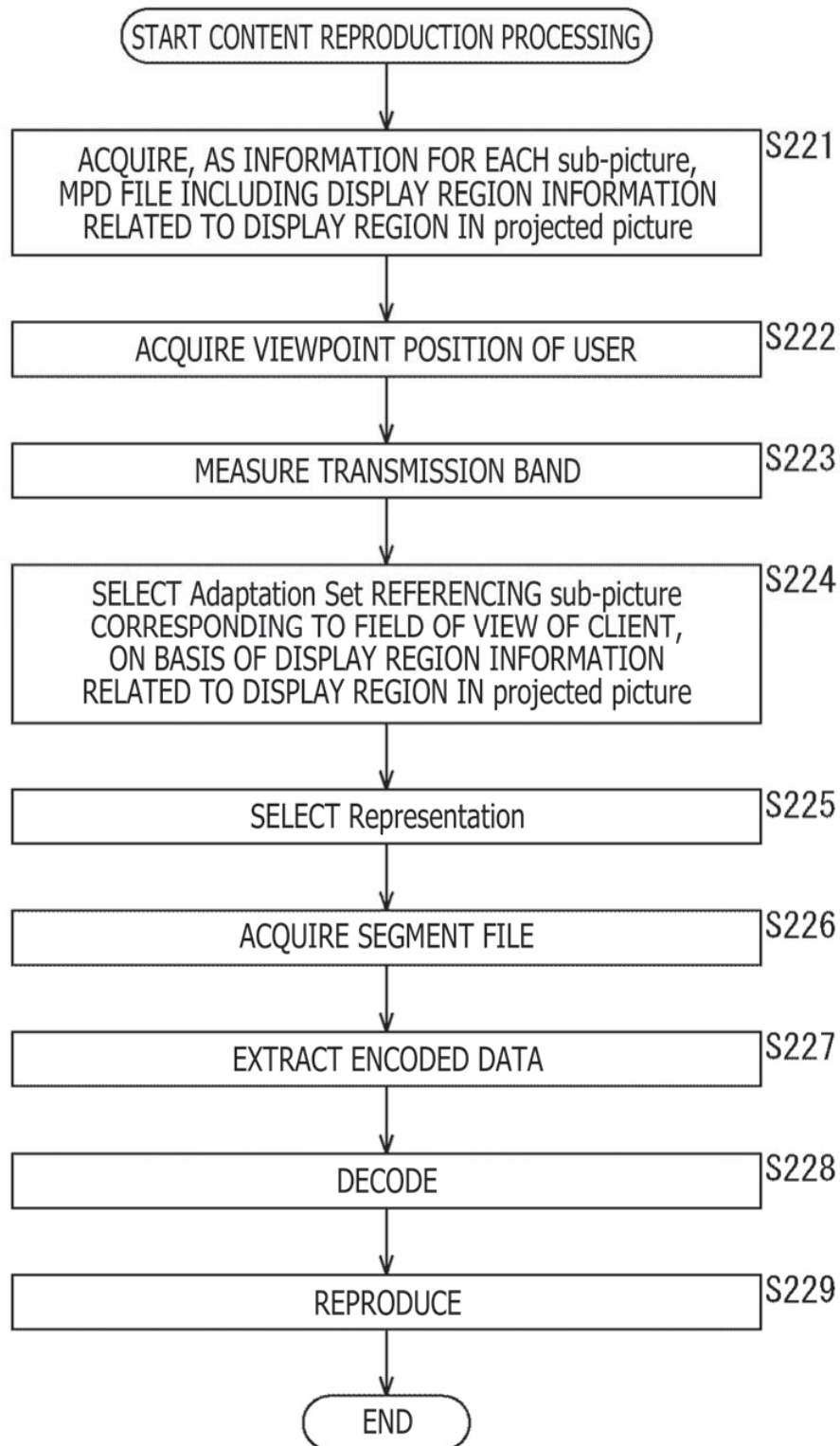


FIG. 38

411
}

ATTRIBUTE VALUE	DATA TYPE	DESCRIPTION
omaf:@proj_picture_width	xs:unsignedInt	WIDTH OF projected picture
omaf:@proj_picture_height	xs:unsignedInt	HEIGHT OF projected picture
omaf:@proj_reg_width	xs:unsignedInt	WIDTH OF REGION ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS.
omaf:@proj_reg_height	xs:unsignedInt	HEIGHT OF REGION ON projected picture TO WHICH PICTURE REFERENCED BY Adaptation Set CORRESPONDS.
omaf:@proj_reg_top	xs:unsignedInt	VERTICAL COORDINATE OF REGION ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS.
omaf:@proj_reg_left	xs:unsignedInt	REGION HORIZONTAL COORDINATE ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS.

FIG. 39

412

ELEMENT AND ATTRIBUTE VALUE	DATA TYPE	DESCRIPTION
twoDCoverage	omaf:twoDCoverageInfoType	CONTAINER ELEMENT
twoDCoverage@proj_picture_width	xs:unsignedInt	WIDTH OF projected picture
twoDCoverage@proj_picture_height	xs:unsignedInt	HEIGHT OF projected picture
twoDCoverage.twoDCoverageInfo	omaf:twoDCoverageInfoType	ELEMENT INDICATING REGION INFORMATION RELATED TO REGION ON projected picture PLURALITY OF SIGNALS CAN BE PROVIDED
twoDCoverage.twoDCoverageInfo@proj_reg_width	xs:unsignedInt	WIDTH OF REGION ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS
twoDCoverage.twoDCoverageInfo@proj_reg_height	xs:unsignedInt	HEIGHT OF REGION ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS
twoDCoverage.twoDCoverageInfo@proj_reg_top	xs:unsignedInt	VERTICAL COORDINATE OF REGION ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS
twoDCoverage.twoDCoverageInfo@proj_reg_left	xs:unsignedInt	REGION HORIZONTAL COORDINATE ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS

FIG. 40

413

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:mpeg:mpeg1:omaf:2017"
  xmlns:omaf="urn:mpeg:mpeg1:omaf:2017"
  elementFormDefault="qualified">
  <xs:element name="twoDCoverage" type="omaf:twoDCoverageType"/>
  <xs:complexType name="twoDCoverageType">
    <xs:sequence>
      <xs:element name="twoDCoverageInfo" type="omaf:twoDCoverageInfoType" minOccurs="1" maxOccurs="255"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="proj_picture_width" type="xs:unsignedInt" use="required"/>
    <xs:attribute name="proj_picture_height" type="xs:unsignedInt" use="required"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="twoDCoverageInfoType">
    <xs:attribute name="proj_reg_width" type="xs:unsignedInt" use="required"/>
    <xs:attribute name="proj_reg_height" type="xs:unsignedInt" use="required"/>
    <xs:attribute name="proj_reg_top" type="xs:unsignedInt" use="required"/>
    <xs:attribute name="proj_reg_left" type="xs:unsignedInt" use="required"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
</xs:schema>

```

FIG. 41

ATTRIBUTE VALUE	DATA TYPE	DESCRIPTION
omaf:@packing_type	omaf:OptionalListOfUnsignedByte	INDICATES PACKING TYPE OF region-wise packing 0: PACKING OF RECTANGULAR REGION Others: reserved
omaf:@proj_picture_width	xs:unsignedInt	WIDTH OF projected picture
omaf:@proj_picture_height	xs:unsignedInt	HEIGHT OF projected picture
omaf:@proj_reg_width	xs:unsignedInt	WIDTH OF REGION ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS
omaf:@proj_reg_height	xs:unsignedInt	HEIGHT OF REGION ON Projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS
omaf:@proj_reg_top	xs:unsignedInt	VERTICAL COORDINATE OF REGION ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS
omaf:@proj_reg_left	xs:unsignedInt	REGION HORIZONTAL COORDINATE ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS

414

FIG. 42

415

ATTRIBUTE VALUE	DATA TYPE	DESCRIPTION
omaf: @packing_type	omaf:OptionalListOfUnsignedByte	INDICATES PACKING TYPE OF region-wise packing 0: PACKING OF RECTANGULAR REGION Others: reserved
twoDCoverage	omaf:twoDCoverageType	CONTAINER ELEMENT
twoDCoverage@proj_picture_width	xs:unsignedInt	WIDTH OF projected picture
twoDCoverage@proj_picture_height	xs:unsignedInt	HEIGHT OF projected picture
twoDCoverage.twoDCoverageInfo	omaf:twoDCoverageInfoType	ELEMENT INDICATING REGION INFORMATION RELATED TO REGION ON projected picture
twoDCoverage.twoDCoverageInfo@proj_reg_width	xs:unsignedInt	WIDTH OF REGION ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS
twoDCoverage.twoDCoverageInfo@proj_reg_height	xs:unsignedInt	HEIGHT OF REGION ON Projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS
twoDCoverage.twoDCoverageInfo@proj_reg_top	xs:unsignedInt	VERTICAL COORDINATE OF REGION ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS
twoDCoverage.twoDCoverageInfo@proj_reg_left	xs:unsignedInt	REGION HORIZONTAL COORDINATE ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS

FIG. 43

416

ATTRIBUTE VALUE	DATA TYPE	DESCRIPTION
cc	omaf:CCType	CONTAINER ELEMENT
cc@2D_coverage_flag	xs:boolean	INDICATES WHETHER DISPLAY REGION IS DEFINED ON SPHERICAL SURFACE OR ON projected picture 0: ON SPHERICAL SURFACE 1: ON projected picture
cc.sphericalCoverage	omaf:sphericalCoverageType	CONTAINER ELEMENT IN SPHERICAL-SURFACE DISPLAY REGION INFORMATION PRESENT ONLY WHEN cc@2D_coverage_flag = 0
cc.sphericalCoverage @shape_type	xs:unsignedByte	INDICATES SHAPE OF SPHERICAL-SURFACE REGION 0: REGION ENCLOSED BY 4 great circles 1: REGION ENCLOSED BY TWO azimuth circles AND TWO elevation angles others: reserved
cc.sphericalCoverage @view_idc_presence_flag	xs:boolean	FLAG INDICATING WHETHER OR NOT view_idc ATTRIBUTE IS PRESENT 0: view_idc ATTRIBUTE IS NOT PRESENT 1: view_idc ATTRIBUTE IS PRESENT
cc.sphericalCoverage @default_view_idc	omaf:ViewType	INDICATES view COMMON TO ALL REGIONS 0: monoscopic view 1: left view 2: right view 3: stereoscopic view THIS ATTRIBUTE IS INEVITABLY PRESENT WHEN cc@view_idc_presence_flag = 0 THIS ATTRIBUTE IS INHIBITED FROM BEING PRESENT WHEN cc@view_idc_presence_flag = 1
cc.sphericalCoverage.coverageInfo	omaf:coverageInfoType	ELEMENT INDICATING SPHERICAL-SURFACE REGION INFORMATION PLURALITY OF SIGNALS CAN BE PROVIDED
cc.sphericalCoverage.coverageInfo@view_idc	omaf:ViewType	INDICATES DIFFERENT views FOR RESPECTIVE REGIONS 0: monoscopic view 1: left view 2: right view 3: stereoscopic view THIS ATTRIBUTE IS INHIBITED FROM BEING PRESENT WHEN cc@view_idc_presence_flag = 0 THIS ATTRIBUTE IS INEVITABLY PRESENT WHEN cc@view_idc_presence_flag = 1

FIG. 44

417

cc. sphericalCoverage.coverageInfo@center_azimuth	omaf:Range1	AZIMUTH ANGLE OF SPHERICAL-SURFACE DISPLAY REGION CENTER
cc. sphericalCoverage.coverageInfo@center_elevation	omaf:Range2	ELEVATION ANGLE OF SPHERICAL-SURFACE DISPLAY REGION CENTER
cc. sphericalCoverage.coverageInfo@center_tilt	omaf:Range1	TILT ANGLE OF SPHERICAL-SURFACE DISPLAY REGION CENTER
cc. sphericalCoverage.coverageInfo@azimuth_range	omaf:Range	AZIMUTH RANGE OF SPHERICAL-SURFACE DISPLAY REGION
cc. sphericalCoverage.coverageInfo@elevation_range	omaf:Range	ELEVATION RANGE OF SPHERICAL-SURFACE DISPLAY REGION
cc. twoDCoverage	omaf:twoDCoverageType	CONTAINER ELEMENT IN DISPLAY REGION INFORMATION ON projected picture PRESENT ONLY WHEN cc@2D_coverage_flag = 1
cc. twoDCoverage@proj_picture_width	xs:unsignedInt	WIDTH OF projected picture
cc. twoDCoverage@proj_picture_height	xs:unsignedInt	HEIGHT OF projected picture
cc. twoDCoverage.twoDCoverageInfo	omaf:twoDCoverageInfoType	ELEMENT INDICATING REGION INFORMATION RELATED TO REGION ON projected picture PLURALITY OF SIGNALS CAN BE PROVIDED
cc. twoDCoverage.twoDCoverageInfo@proj_reg_width	xs:unsignedInt	WIDTH OF REGION ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS
cc. twoDCoverage.twoDCoverageInfo@proj_reg_height	xs:unsignedInt	HEIGHT OF REGION ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS
cc. twoDCoverage.twoDCoverageInfo@proj_reg_top	xs:unsignedInt	VERTICAL COORDINATE OF REGION ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS
cc. twoDCoverage.twoDCoverageInfo@proj_reg_left	xs:unsignedInt	HORIZONTAL COORDINATE OF REGION ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS

FIG. 45

418

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:mpeg:mpegI:omaf:2017"
  xmlns:omaf="urn:mpeg:mpegI:omaf:2017"
  elementFormDefault="qualified">
  <xs:element name="cc" type="CCType"/>
  <xs:complexType name="CCType">
    <xs:sequence>
      <xs:element name="sphericalCoverage" type="omaf:sphericalCoverageType" minOccurs="0" maxOccurs="1"/>
      <xs:element name="twoDCoverage" type="omaf:twoDCoverageType" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:attribute name="2D_coverage_flag" type="xs:boolean" use="optional" default="0"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="omaf:sphericalCoverageType">
  <xs:sequence>
    <xs:element name="coverageInfo" type="omaf:coverageInfoType" minOccurs="1" maxOccurs="255"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="shape_type" type="xs:unsignedByte" use="optional" default="0"/>
  <xs:attribute name="view_idc_presence_flag" type="xs:boolean" use="optional"/>
  <xs:attribute name="default_view_idc" type="omaf:ViewType" use="optional"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

FIG. 46

~419

```

<xs:complexType name="coverageInfoType">
  <xs:attribute name="view_idc" type="omaf:ViewType" use="optional" default="0"/>
  <xs:attribute name="center_azimuth" type="omaf:Range1" use="optional" default="0"/>
  <xs:attribute name="center_elevation" type="omaf:Range2" use="optional" default="0"/>
  <xs:attribute name="center_tilt" type="omaf:Range1" use="optional" default="0"/>
  <xs:attribute name="azimuth_range" type="omaf:Range" use="optional" default="23592960"/>
  <xs:attribute name="elevation_range" type="omaf:VRange" use="optional" default="11796480"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="twoDCoverageType">
  <xs:sequence>
    <xs:element name="twoDCoverageInfo" type="omaf:twoDCoverageInfoType" minOccurs="1" maxOccurs="255"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="proj_picture_width" type="xs:unsignedInt" use="required"/>
  <xs:attribute name="proj_picture_height" type="xs:unsignedInt" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="twoDCoverageInfoType">
  <xs:attribute name="proj_reg_width" type="xs:unsignedInt" use="required"/>
  <xs:attribute name="proj_reg_height" type="xs:unsignedInt" use="required"/>
  <xs:attribute name="proj_reg_top" type="xs:unsignedInt" use="required"/>
  <xs:attribute name="proj_reg_left" type="xs:unsignedInt" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

FIG. 47

420

```

<xs:simpleType name="Range1">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="-11796480"/>
    <xs:maxInclusive value="11796479"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Range2">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="-5898240"/>
    <xs:maxInclusive value="5898240"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="HRange">
  <xs:restriction base="xs:unsignedInt">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="23592960"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="VRange">
  <xs:restriction base="xs:unsignedInt">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="11796480"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ViewType">
  <xs:restriction base="xs:unsignedByte">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="3"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

FIG. 48

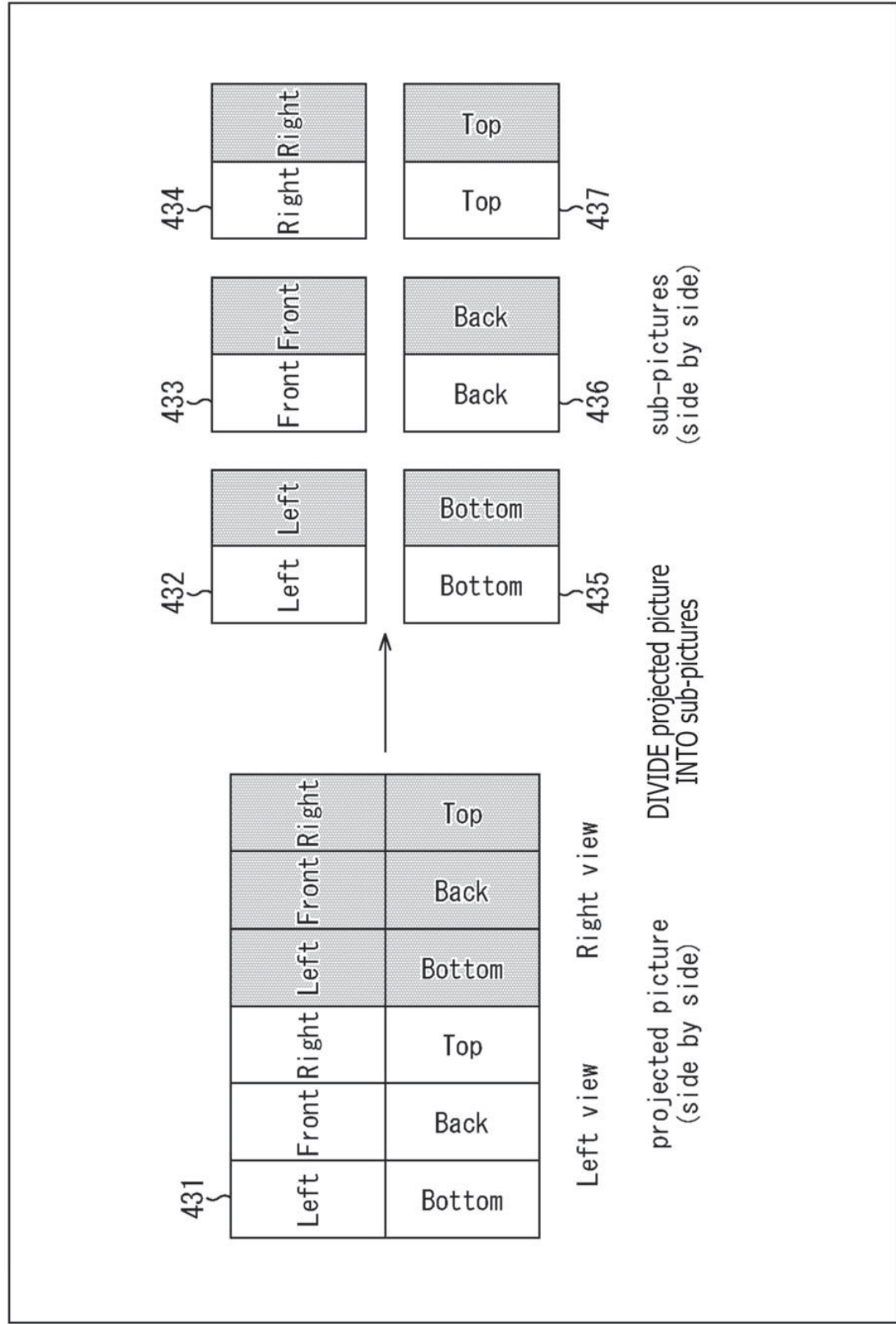


FIG. 49

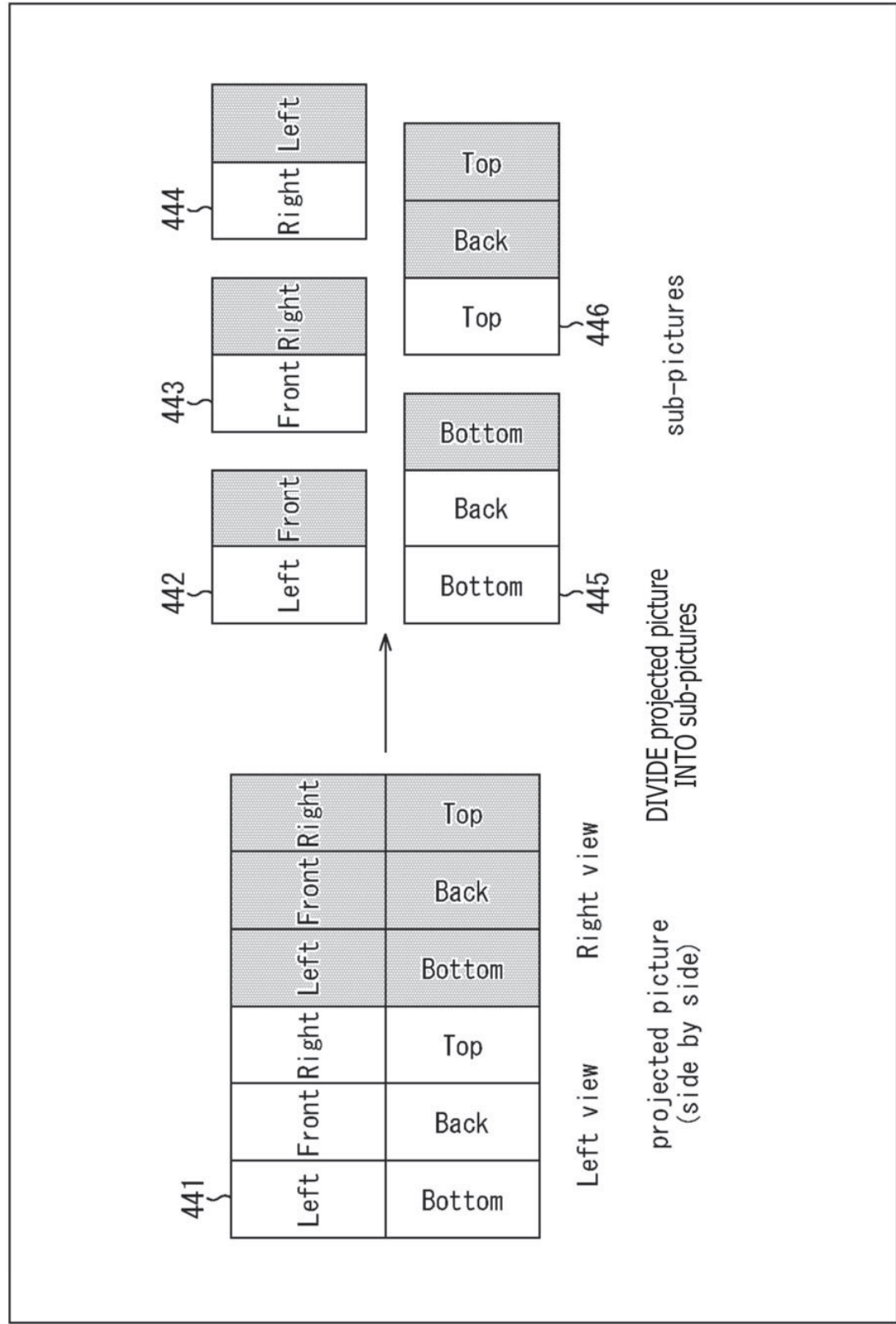


FIG. 50

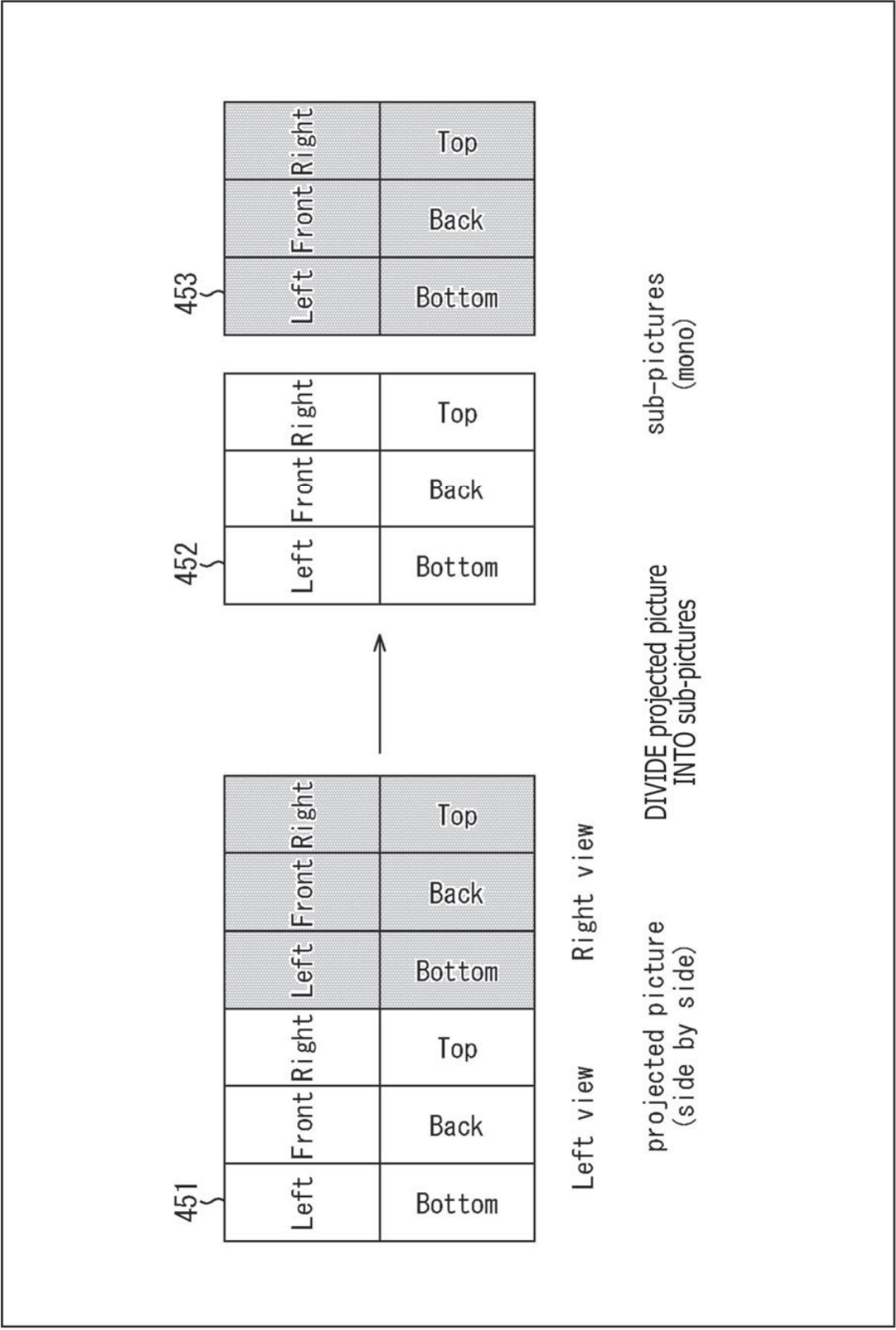


FIG. 51

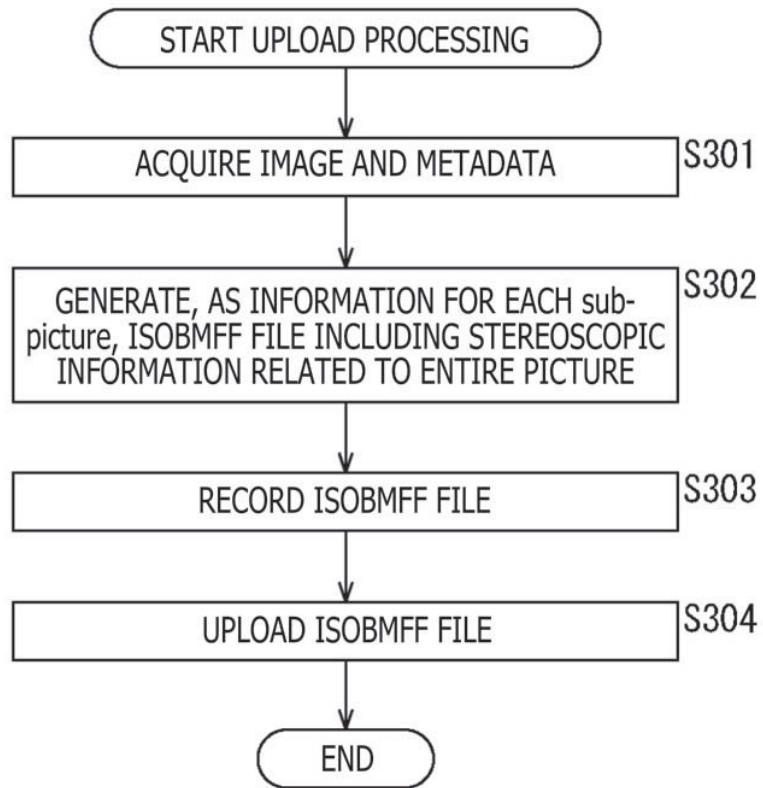


FIG. 52

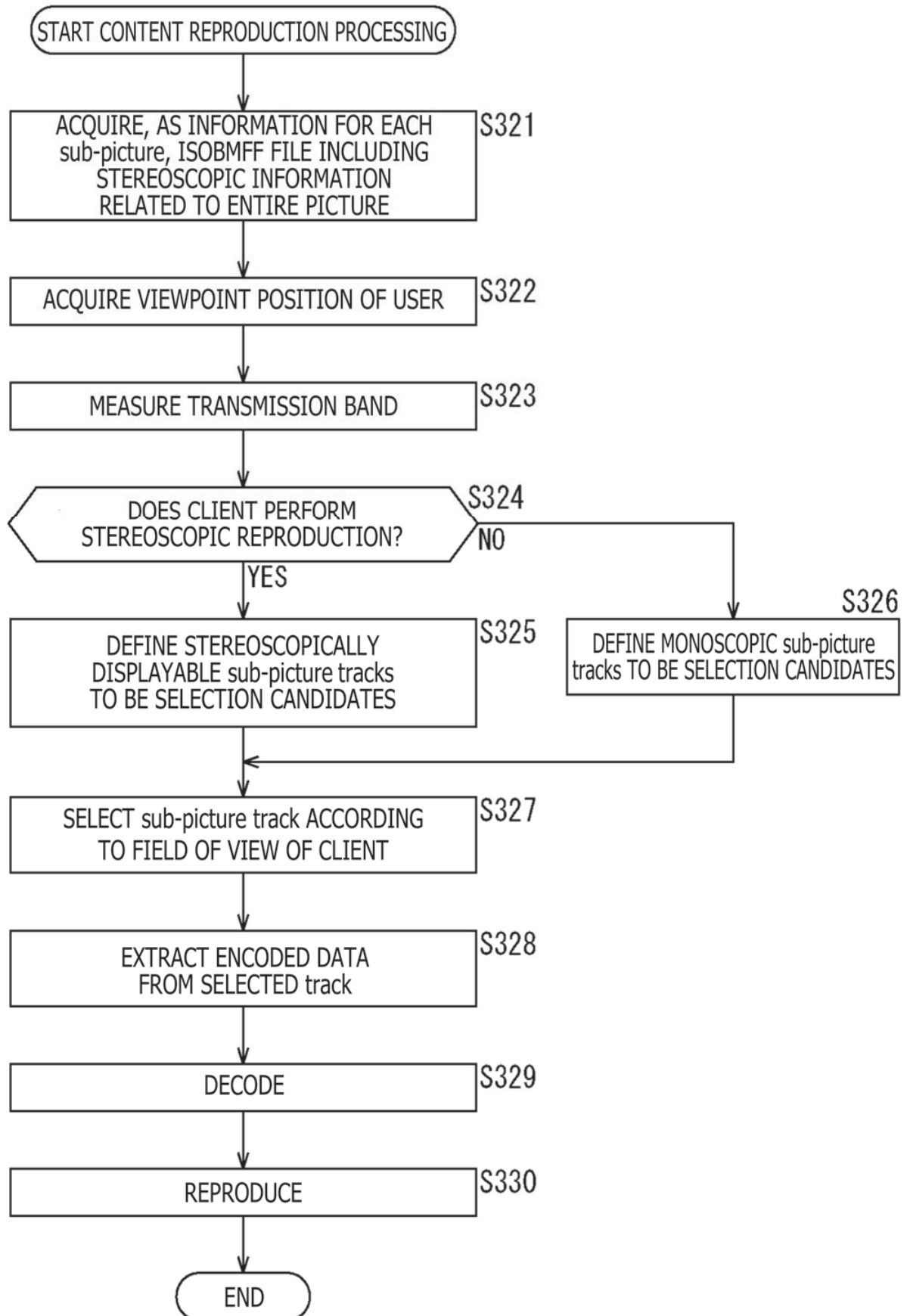


FIG. 53

461
{

```
aligned(8) class OriginalStereoVideoBox extends extends FullBox('ostv', version = 0, 0)
{
    template unsigned int(30) reserved = 0;
    unsigned int(2) single_view_allowed;
    unsigned int(32) stereo_scheme;
    unsigned int(32) length;
    unsigned int(8)[length] stereo_indication_type;
    Box[] any_box; // optional
}
```

FIG. 54

single_view_allowed	0: CONTENT IS INTENDED TO BE DISPLAYED ONLY BY STEREOSCOPIC DISPLAY (single_view_allowed&1) = 1: CONTENT IS ALLOWED TO BE DISPLAYED IN right view ON MONOSCOPIC DISPLAY (single_view_allowed&2) = 2: CONTENT IS ALLOWED TO BE DISPLAYED IN left view ON MONOSCOPIC DISPLAY
stereo_scheme	1: FRAME PACKING METHOD COMPLIES WITH Frame packing arrangement SEI OF ISO/IEC 14496-10 2: FRAME PACKING METHOD COMPLIES WITH Annex. L OF ISO/IEC 13818-2 3: FRAME PACKING METHOD COMPLIES WITH frame/service compatible AND 2D/3D Mixed service OF ISO/IEC 23000-11
length	BYTE LENGTH OF stereo_indication_type
stereo_indication_type	FRAME PACKING METHOD IN ACCORDANCE WITH stereo_scheme

FIG. 55

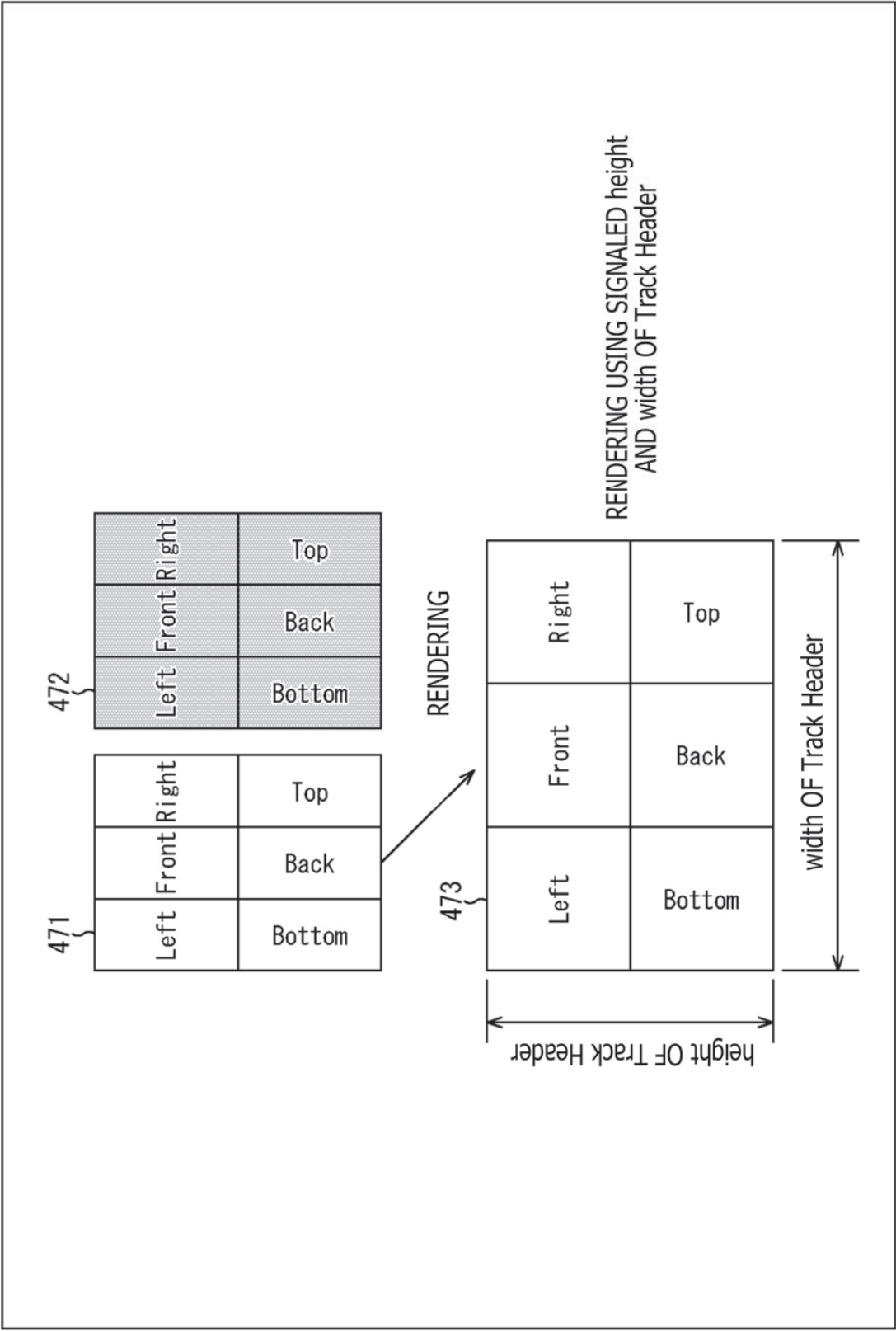


FIG. 56

481
}

```
class PixelAspectRatioBox extends Box ('pasp') {  
    unsigned int (32) hSpacing;  
    unsigned int (32) vSpacing;  
}
```

FIG. 57

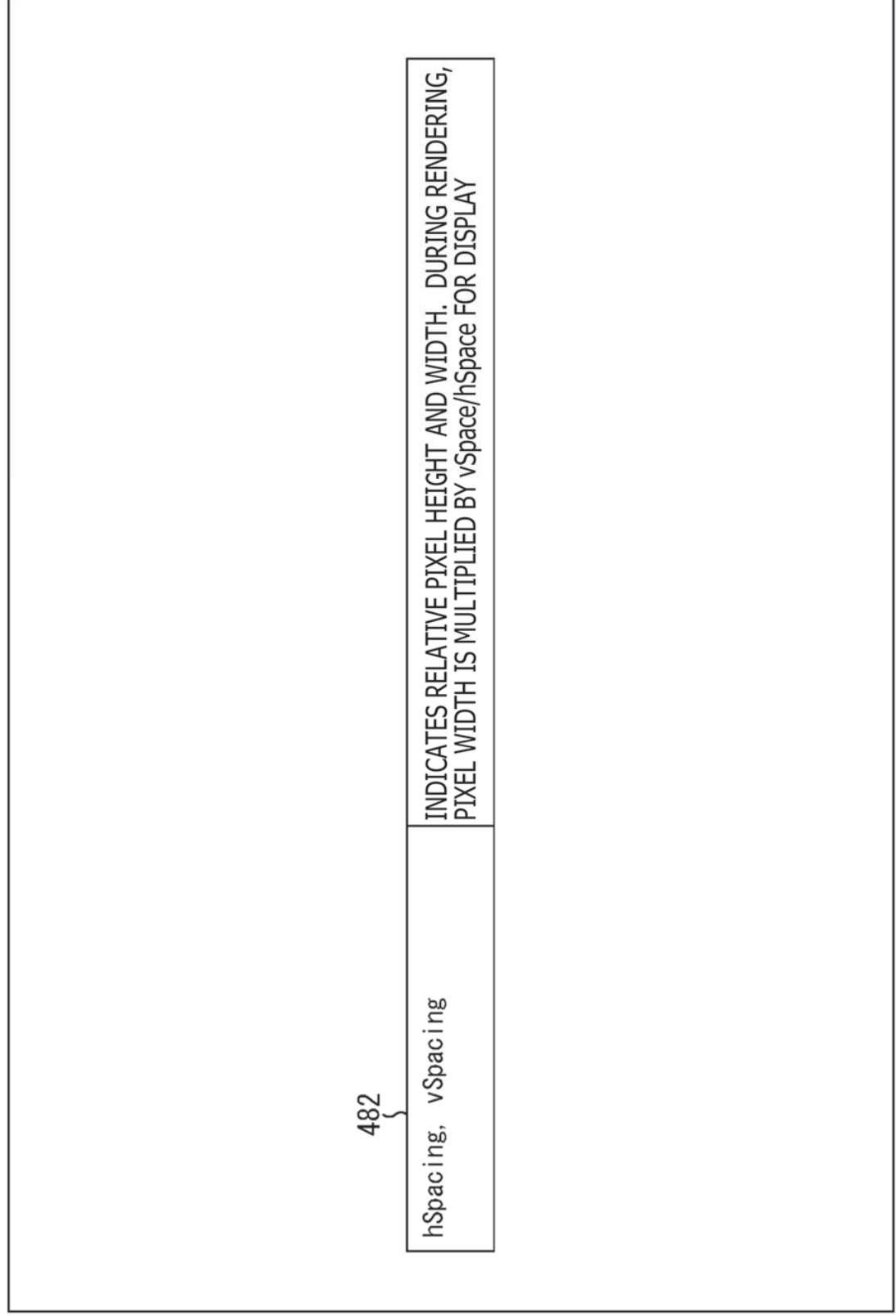


FIG. 58

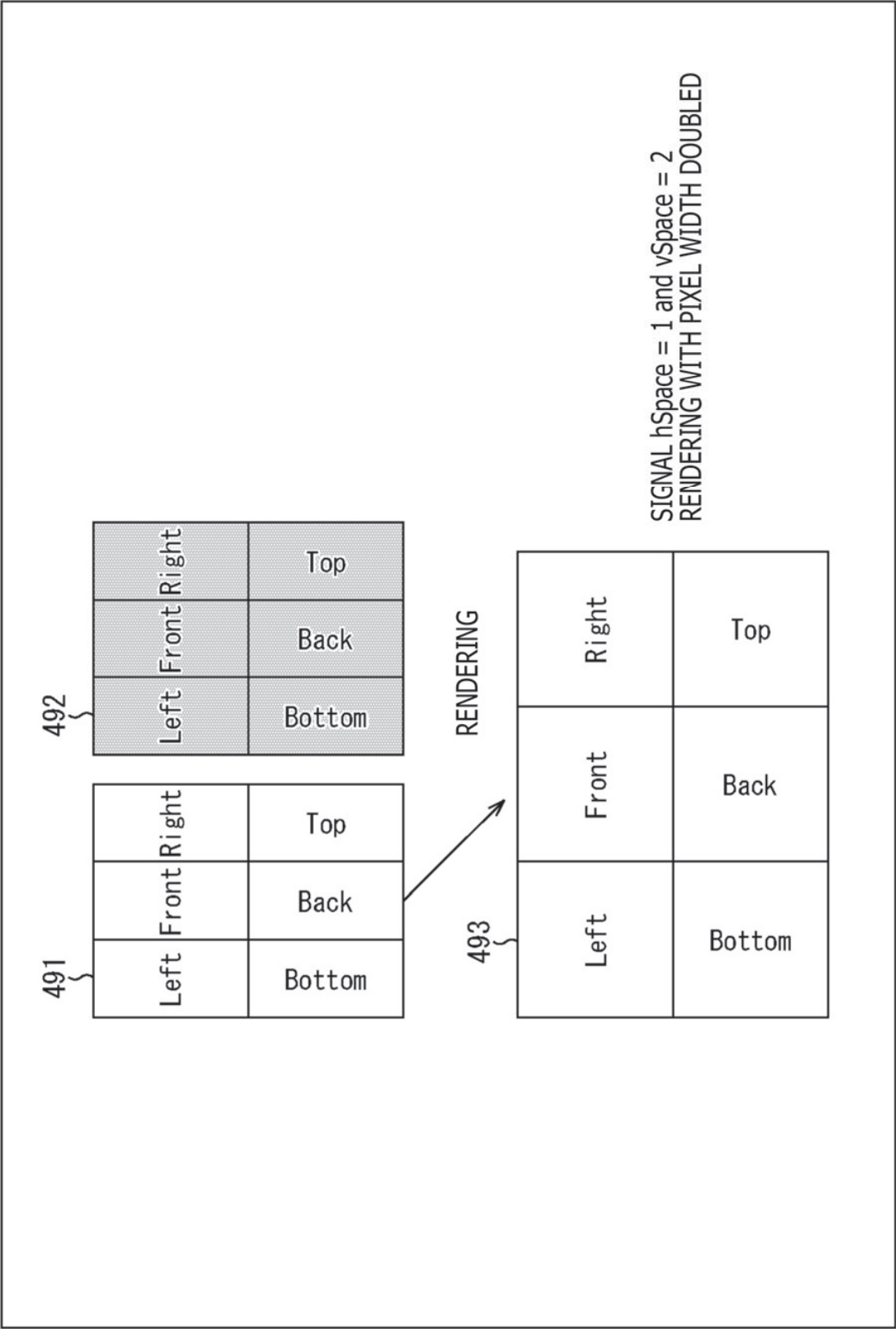


FIG. 59

501 {

```
aligned(8) class OriginalSchemeInformationBox extends Box('osch') {  
    Box scheme_specific_data[];  
}
```

FIG. 60

```
502 {
    aligned(8) class 2DCoverageInformationBox() extends FullBox('2dco', 0, 0) {
        bit(6) reserved = 0;
        unsigned int(2) stereo_presentation_suitable;
        unsigned int(32) proj_picture_width;
        unsigned int(32) proj_picture_height;
        unsigned int(8) num_regions;
        for (i=0; i<num_regions; i++) {
            unsigned int(16) proj_reg_width[i];
            unsigned int(16) proj_reg_height[i];
            unsigned int(16) proj_reg_top[i];
            unsigned int(16) proj_reg_left[i];
        }
    }
}
```

FIG. 61

503

stereo_presentation_suitable	0: PICTURE IN track IS MONOSCOPIC OR INCLUDES L view AND R view BUT IS NOT STEREOSCOPICALLY DISPLAYABLE 1: PICTURE IN track CAN BE STEREOSCOPICALLY DISPLAYED IN PARTIAL REGION 2: PICTURE IN track CAN BE STEREOSCOPICALLY DISPLAYED OVER ENTIRE REGION Others: reserved
------------------------------	--

FIG. 62

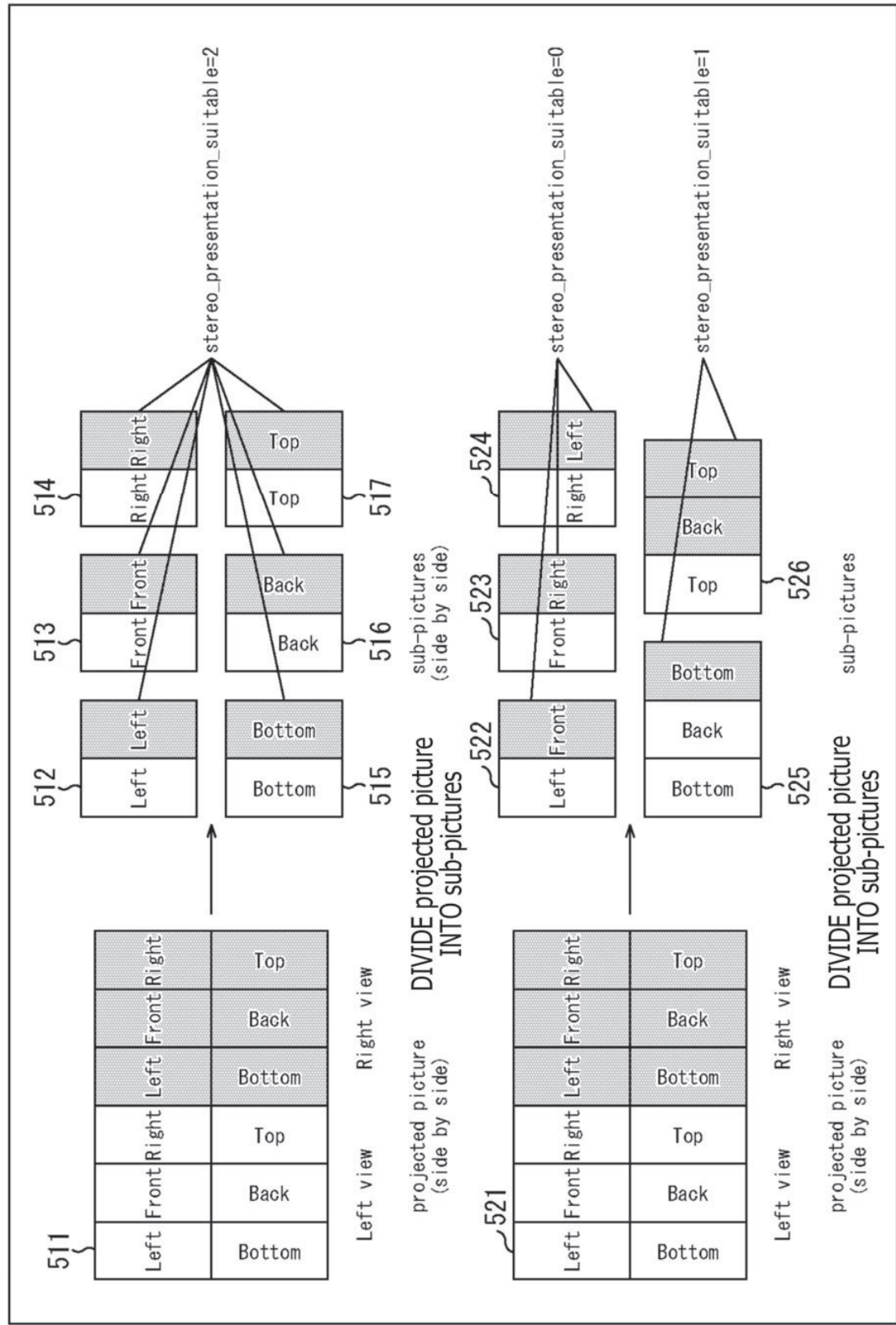


FIG. 63

531 {

```
aligned(8) class TrackStereoVideoBox() extends FullBox('tstv', 0, 0) {  
    bit(7) reserved = 0;  
    unsigned int(1) stereo_presentation_suitable;  
}
```

FIG. 64

532

```

aligned(8) class 2DCoverageInformationBox() extends FullBox('2dco', 0, 0) {
    bit(7) reserved = 0;
    unsigned int(1) view_idc_presence_flag;
    if (view_idc_presence_flag == 0) {
        unsigned int(2) default_view_idc;
        bit(6) reserved = 0;
    }
    unsigned int(32) proj_picture_width;
    unsigned int(32) proj_picture_height;
    unsigned int(8) num_regions;
    for(i=0; i<num_regions; i++) {
        if (view_idc_presence_flag == 1) {
            unsigned int(2) view_idc;
            bit(6) reserved = 0;
        }
        unsigned int(16) proj_reg_width[i];
        unsigned int(16) proj_reg_height[i];
        unsigned int(16) proj_reg_top[i];
        unsigned int(16) proj_reg_left[i];
    }
}

```

FIG. 65

533 {	
view_idc_presence_flag	INDICATES WHETHER OR NOT SEPARATE view_idc IS PRESENT IN EACH REGION. 0: SEPARATE view_idc IS NOT PRESENT IN EACH REGION. 1: SEPARATE view_idc IS PRESENT IN EACH REGION.
default_view_idc	INDICATES view COMMON TO ALL REGIONS. 0: monoscopic view 1: left view 2: right view 3: stereoscopic view
view_idc	INDICATES DIFFERENT views FOR RESPECTIVE REGIONS. 0: monoscopic view 1: left view 2: right view 3: stereoscopic view IN CASE WHERE THIS FIELD IS NOT PRESENT, default_view_idc INDICATES DIFFERENT view FOR EACH REGION.

FIG. 66

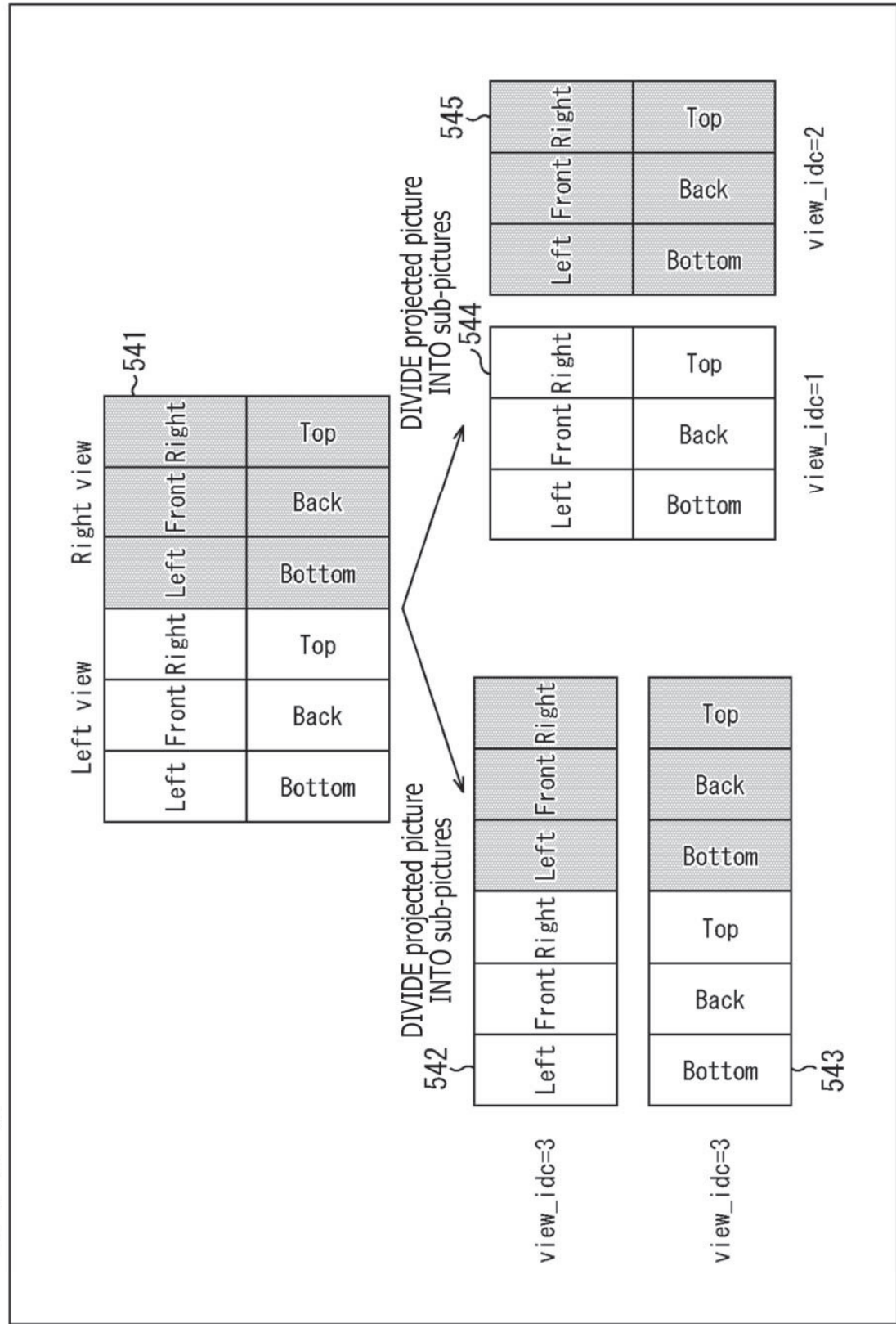


FIG. 67

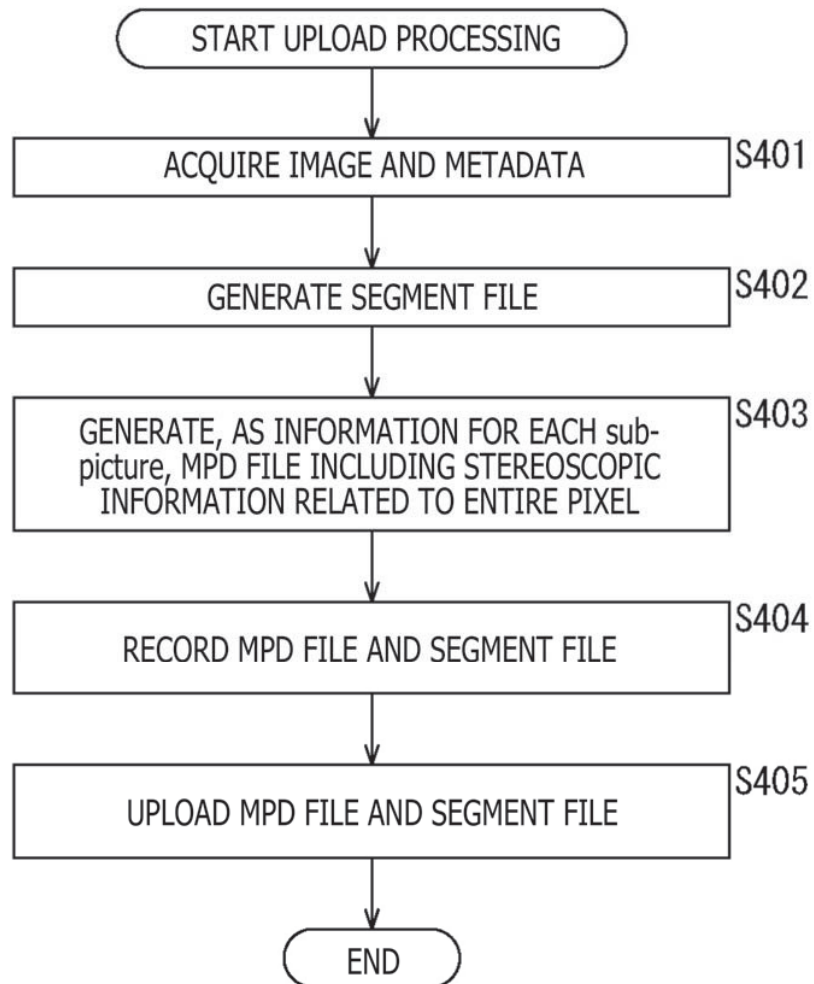


FIG. 68

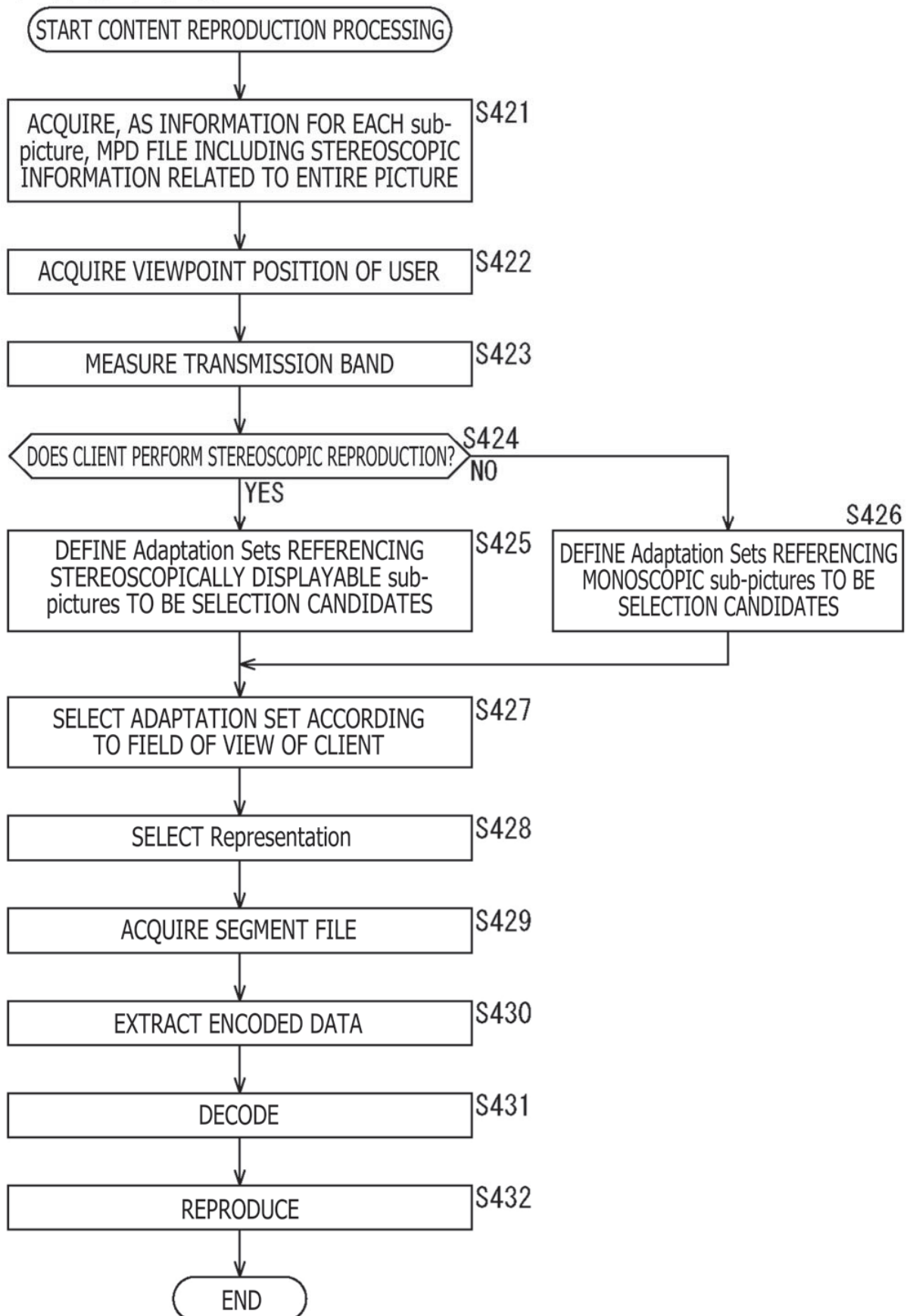


FIG. 69

551

ATTRIBUTE VALUE	DATA TYPE	DESCRIPTION
twoDCoverage	omaf:twoDCoverageType	CONTAINER ELEMENT
twoDCoverage@stereo_presentation_suitable	omaf:StereoPresentationType	INDICATES WHETHER Adaptation Sets IS STEREOSCOPICALLY DISPLAYABLE 0: PICTURE REFERENCED BY Adaptation Sets IS NOT MONOSCOPICALLY OR STEREOSCOPICALLY DISPLAYABLE 1: PICTURE REFERENCED BY Adaptation Sets IS STEREOSCOPICALLY DISPLAYABLE IN PARTIAL REGION 2: PICTURE REFERENCED BY Adaptation Sets IS STEREOSCOPICALLY DISPLAYABLE OVER ENTIRE REGION
twoDCoverage@view_idc_presence_flag	xs:boolean	INDICATES WHETHER OR NOT SEPARATE view_idc IS PRESENT IN EACH REGION 0: SEPARATE view_idc IS NOT PRESENT IN EACH REGION 1: SEPARATE view_idc IS PRESENT IN EACH REGION
twoDCoverage@default_view_idc	omaf:ViewType	INDICATES VIEW COMMON TO ALL REGIONS 0: monoscopic view 1: left view 2: right view 3: stereoscopic view THIS ATTRIBUTE IS INEVITABLY PRESENT WHEN twoDCoverage@view_idc_presence_flag = 0 THIS ATTRIBUTE IS INHIBITED FROM BEING PRESENT WHEN twoDCoverage@view_idc_presence_flag = 1
twoDCoverage@proj_picture_width	xs:unsignedInt	WIDTH OF projected picture
twoDCoverage@proj_picture_height	xs:unsignedInt	HEIGHT OF projected picture
twoDCoverage.twoDCoverageInfo	omaf:twoDCoverageInfoType	ELEMENT INDICATING REGION INFORMATION RELATED TO REGION ON projected picture PLURALITY OF SIGNALS CAN BE PROVIDED

FIG. 70

552
}

twoDCoverage.twoDCoverageInfo@view_idc	omaf:ViewType	INDICATES DIFFERENT views FOR RESPECTIVE REGIONS 0: monoscopic view 1: left view 2: right view 3: stereoscopic view THIS ATTRIBUTE IS INHIBITED FROM BEING PRESENT WHEN twoDCoverage@view_idc_presence_flag = 0 THIS ATTRIBUTE IS INEVITABLY PRESENT WHEN twoDCoverage@view_idc_presence_flag = 1
twoDCoverage.twoDCoverageInfo@proj_reg_width	xs:unsignedInt	WIDTH OF REGION ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS
twoDCoverage.twoDCoverageInfo@proj_reg_height	xs:unsignedInt	HEIGHT OF REGION ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS
twoDCoverage.twoDCoverageInfo@proj_reg_top	xs:unsignedInt	VERTICAL COORDINATE OF REGION ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPOND
twoDCoverage.twoDCoverageInfo@proj_reg_left	xs:unsignedInt	REGION HORIZONTAL COORDINATE ON projected picture TO WHICH picture REFERENCED BY Adaptation Set CORRESPONDS

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:mpeg:mpeg1:omaf:2017"
  xmlns:omaf="urn:mpeg:mpeg1:omaf:2017"
  elementFormDefault="qualified">
  <xs:element name="twoDCoverage" type="omaf:twoDCoverageType"/>
  <xs:complexType name="twoDCoverageType">
    <xs:sequence>
      <xs:element name="twoDCoverageInfo" type="omaf:twoDCoverageInfoType" minOccurs="1" maxOccurs="255"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:attribute name="stereo_presentation_suitable" type="omaf:StereoPresentationType" use="optional"/>
  <xs:attribute name="view_idc_presence_flag" type="xs:boolean" use="optional"/>
  <xs:attribute name="default_view_idc" type="omaf:ViewType" use="optional"/>
  <xs:attribute name="proj_picture_width" type="xs:unsignedInt" use="required"/>
  <xs:attribute name="proj_picture_height" type="xs:unsignedInt" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="twoDCoverageInfoType">
  <xs:attribute name="view_idc" type="omaf:ViewType" use="optional"/>
  <xs:attribute name="proj_reg_width" type="xs:unsignedInt" use="required"/>
  <xs:attribute name="proj_reg_height" type="xs:unsignedInt" use="required"/>
  <xs:attribute name="proj_reg_top" type="xs:unsignedInt" use="required"/>
  <xs:attribute name="proj_reg_left" type="xs:unsignedInt" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:simpleType name="StereoPresentationType">
  <xs:restriction base="xs:unsignedByte">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="2"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ViewType">
  <xs:restriction base="xs:unsignedByte">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="3"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

FIG. 72

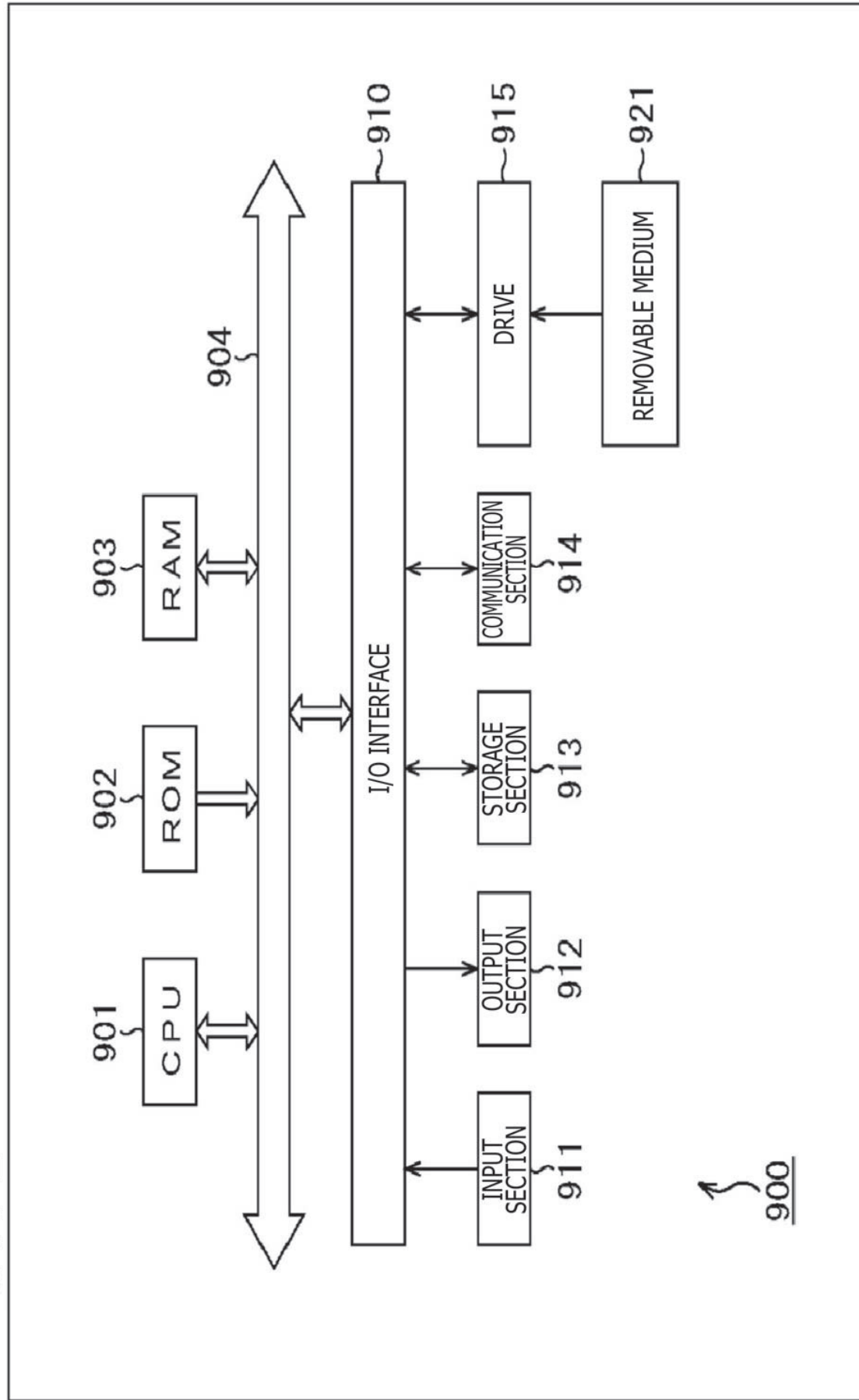


FIG. 73

1001

```
aligned(8) class SubPictureCompositionBox extends TrackGroupTypeBox('spco') {  
    unsigned int(1) identical_to_proj_pic_flag;  
    unsigned int(7) reserved = 0;  
    SubPictureRegionBox();  
    RegionWisePackingBox(); // optional  
    // optional boxes  
}
```

FIG. 74

<SupplementalProperty schemeIdUri="urn:mpeg:mpegI:omaf:2017:pr id">