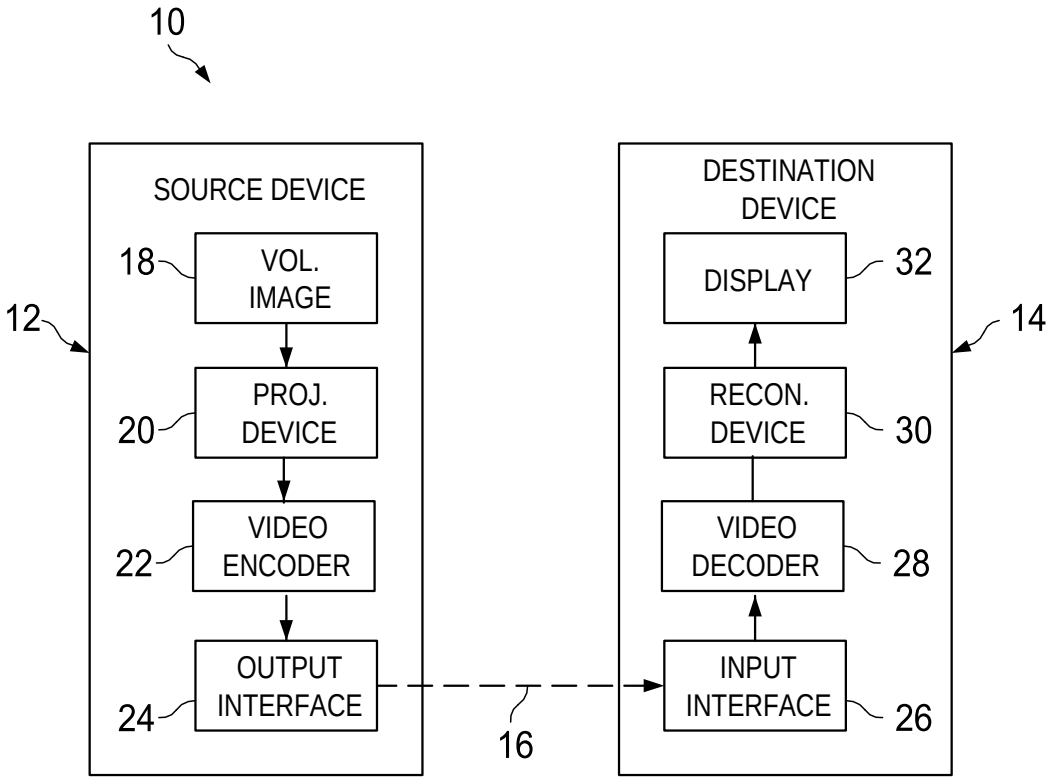

10
SOURCE DEVICE
18
VOL. IMAGE
12
PROJ. DEVICE
20
VIDEO ENCODER
22
OUTPUT INTERFACE
24
16
DESTINATION DEVICE
DISPLAY
32
14
RECON. DEVICE
30
VIDEO DECODER
28
INPUT INTERFACE
26

FIG. 1

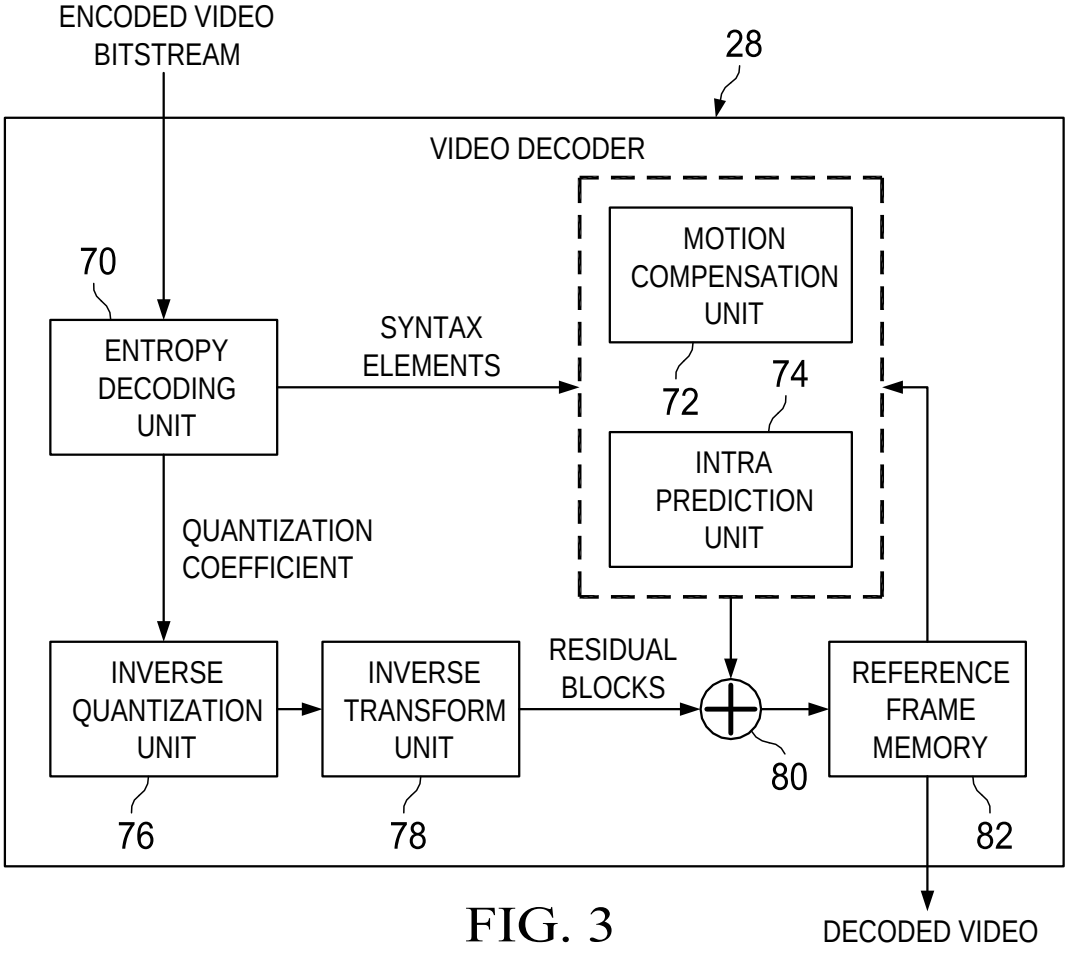

ENCODED VIDEO BITSTREAM
28
VIDEO DECODER
70
ENTROPY DECODING UNIT
SYNTAX ELEMENTS
MOTION COMPENSATION UNIT
72
74
INTRA PREDICTION UNIT
QUANTIZATION COEFFICIENT
INVERSE QUANTIZATION UNIT
INVERSE TRANSFORM UNIT
RESIDUAL BLOCKS
80
REFERENCE FRAME MEMORY
76
78
82
DECODED VIDEO

FIG. 3

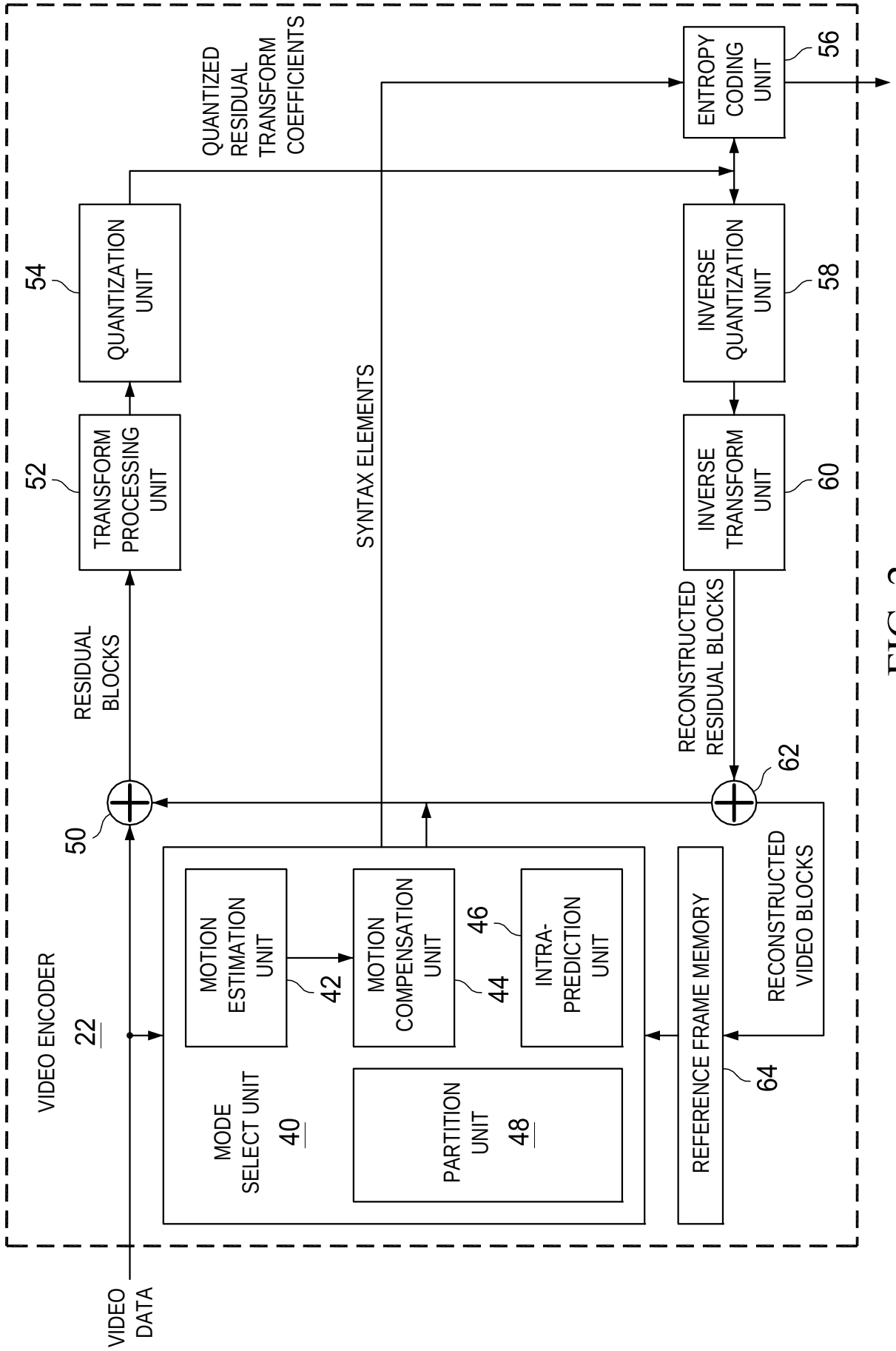

VIDEO ENCODER
22
VIDEO DATA
MODE SELECT UNIT
40
MOTION ESTIMATION UNIT
42
MOTION COMPENSATION UNIT
44
INTRA-PREDICTION UNIT
46
PARTITION UNIT
48
50
RESIDUAL BLOCKS
52
TRANSFORM PROCESSING UNIT
54
QUANTIZATION UNIT
QUANTIZED RESIDUAL TRANSFORM COEFFICIENTS
SYNTAX ELEMENTS
56
ENTROPY CODING UNIT
58
INVERSE QUANTIZATION UNIT
60
INVERSE TRANSFORM UNIT
RECONSTRUCTED RESIDUAL BLOCKS
62
RECONSTRUCTED VIDEO BLOCKS
64
REFERENCE FRAME MEMORY

FIG. 2

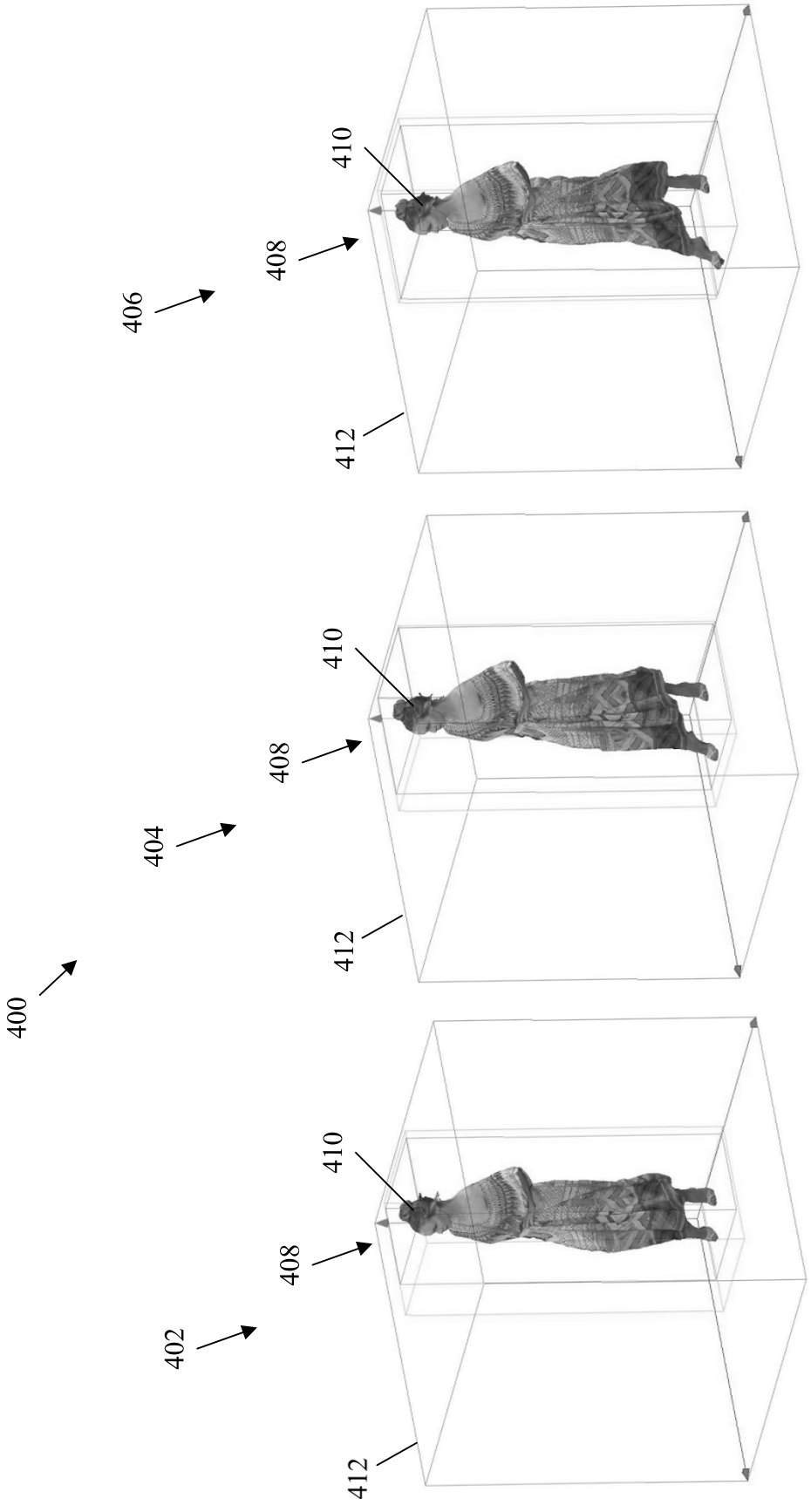


FIG. 4

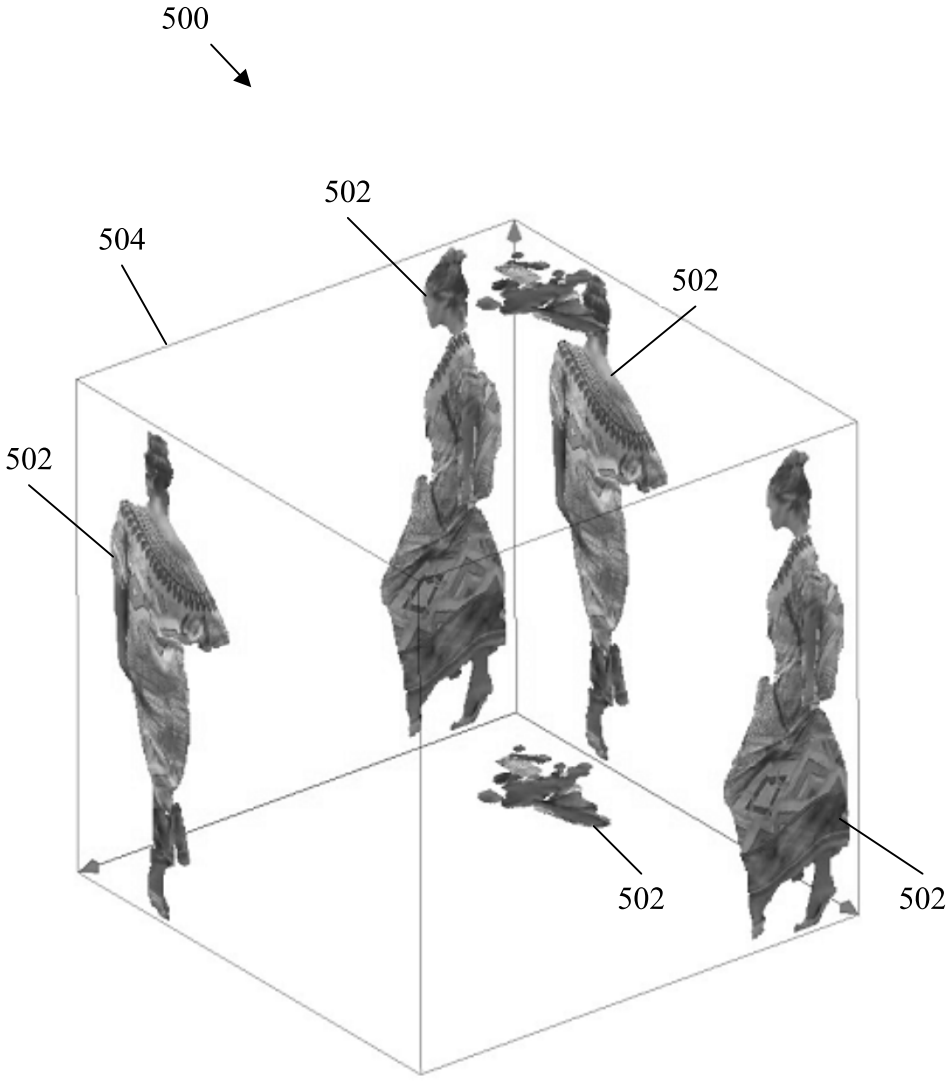


FIG. 5

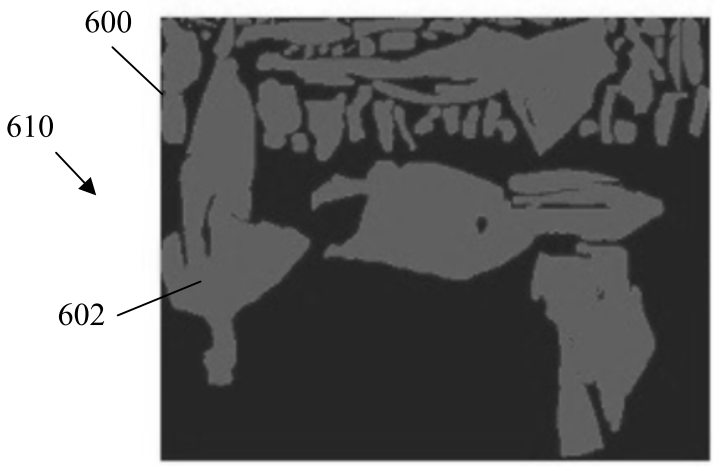


FIG. 6A

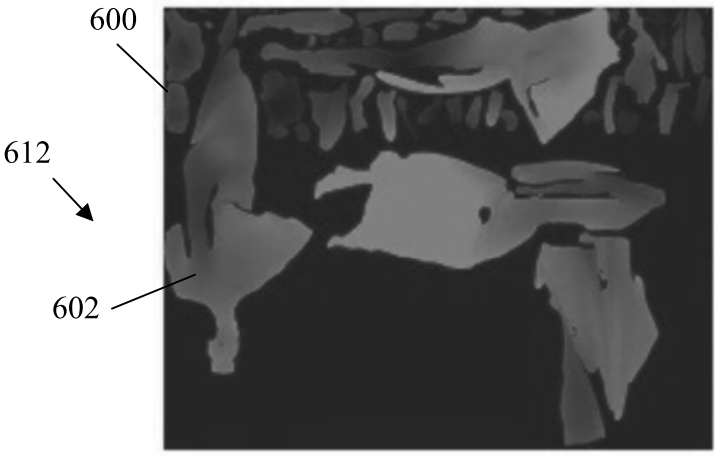


FIG. 6B

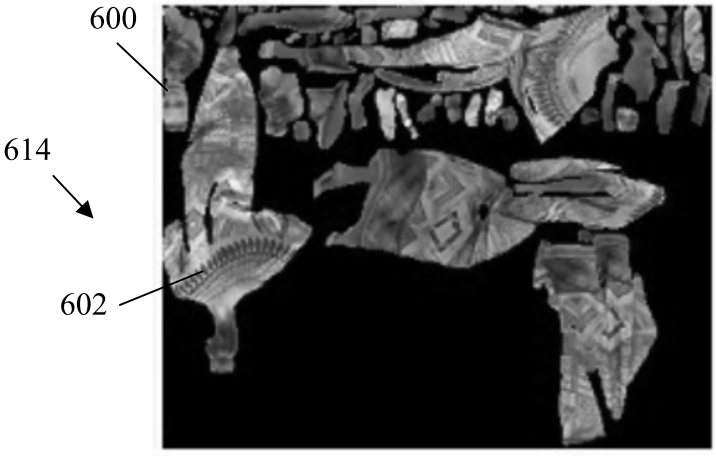


FIG. 6C

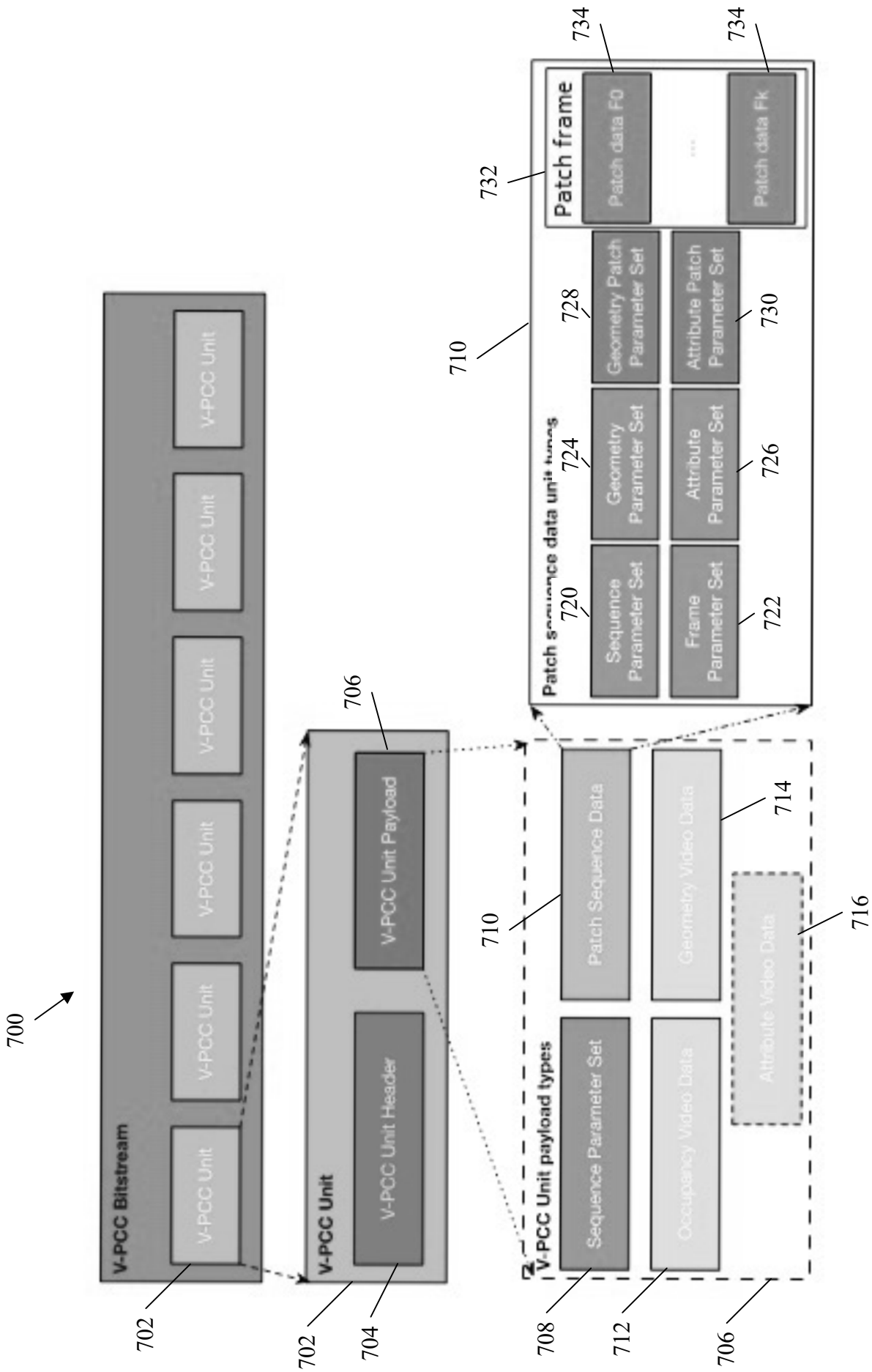

700
V-PCC Bitstream
V-PCC Unit
V-PCC Unit
V-PCC Unit
V-PCC Unit
V-PCC Unit
V-PCC Unit
V-PCC Unit Header
V-PCC Unit Payload
V-PCC Unit payload types
Sequence Parameter Set
Occupancy Video Data
Patch Sequence Data
Geometry Video Data
Attribute Video Data
Patch sequence data unit types
Sequence Parameter Set
Frame Parameter Set
Geometry Parameter Set
Attribute Parameter Set
Geometry Patch Parameter Set
Attribute Patch Parameter Set
Patch frame
Patch data F0
Patch data Fk
702
702
704
706
708
710
710
712
714
716
720
722
724
726
728
730
732
734
734

FIG. 7

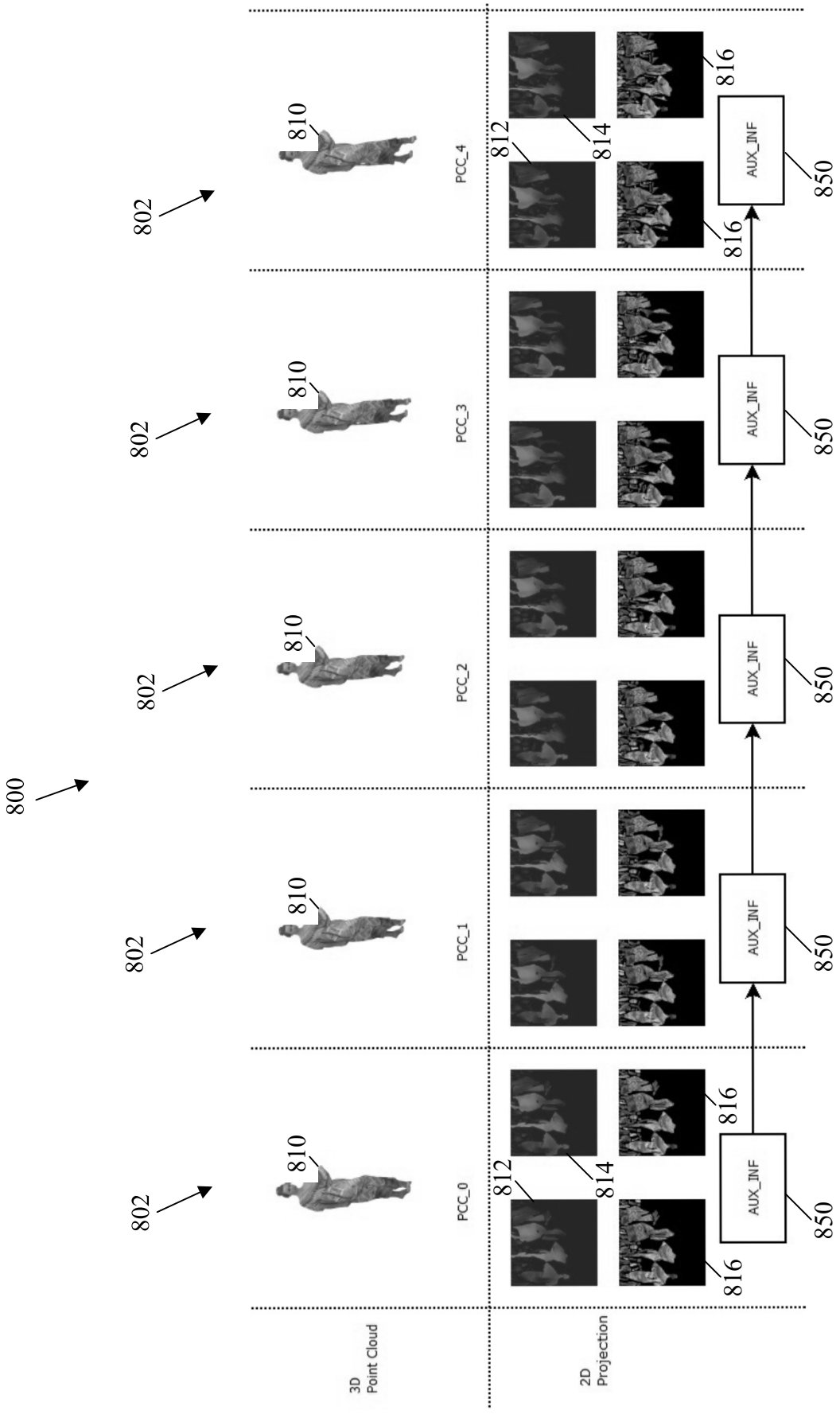

800
802
802
802
802
802
810
810
810
810
810
PCC_0
PCC_1
PCC_2
PCC_3
PCC_4
812
814
816
816
812
814
816
816
AUX_INF
AUX_INF
AUX_INF
AUX_INF
850
850
850
850
850
3D
Point Cloud
2D
Projection

FIG. 8

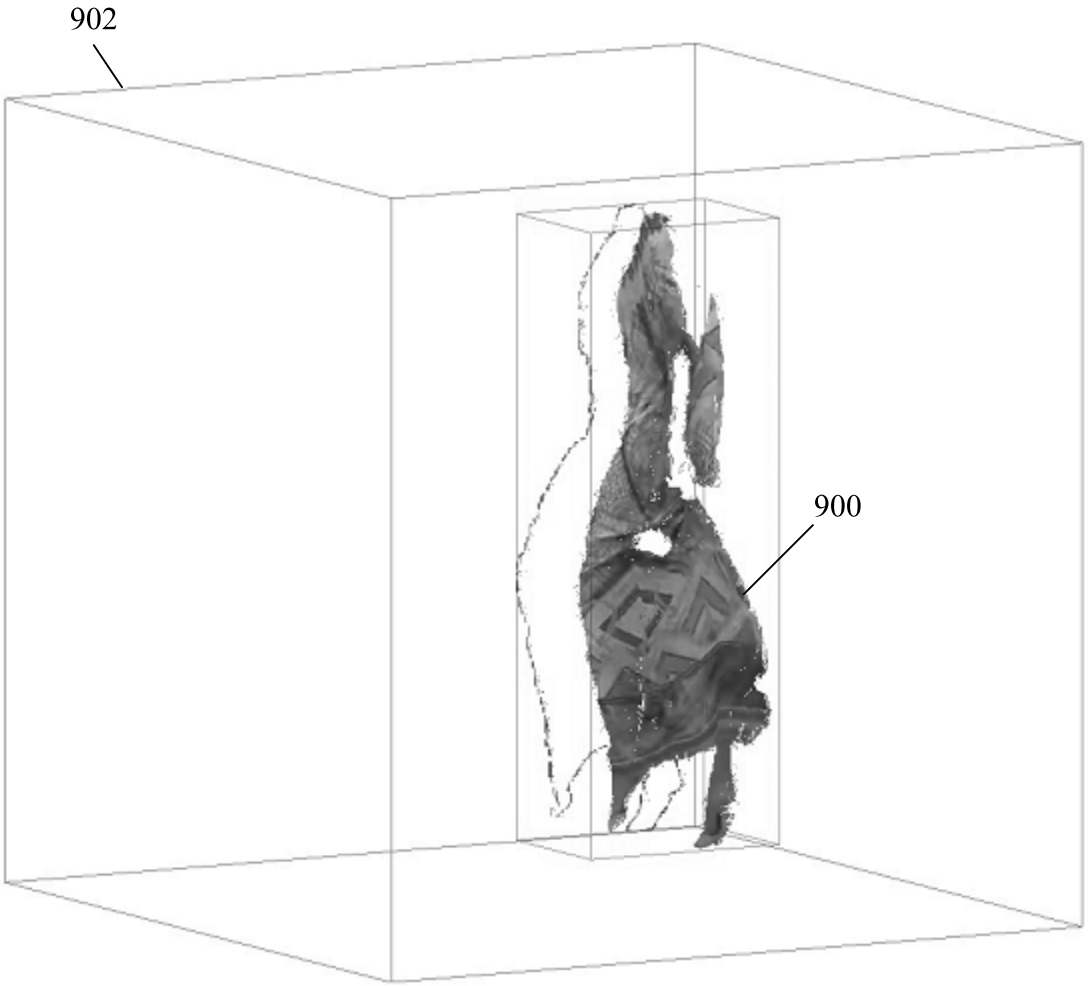


FIG. 9

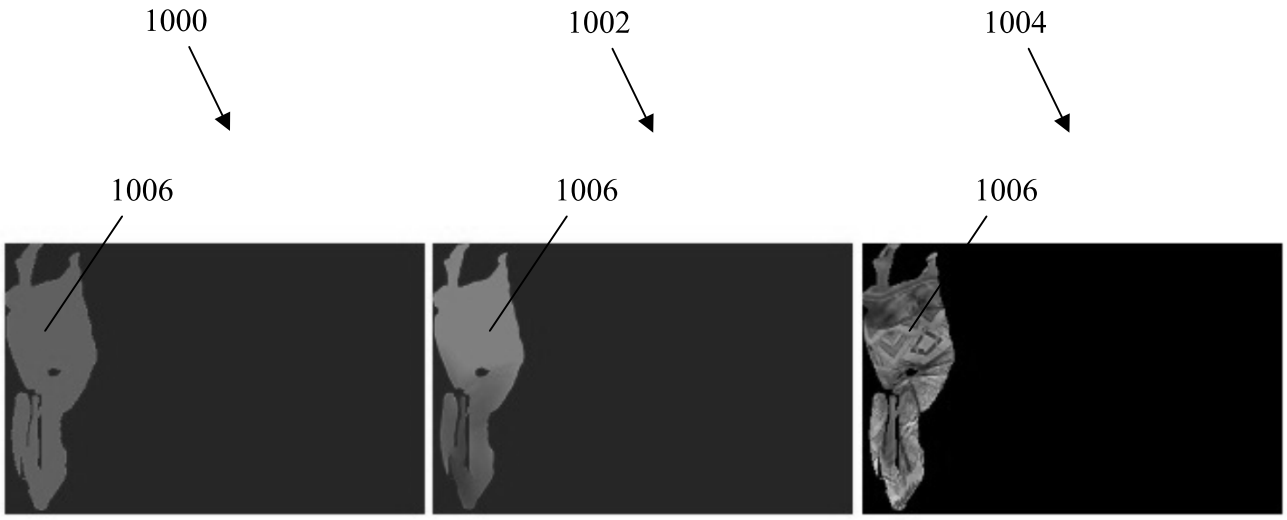


FIG. 10A FIG. 10B FIG. 10C

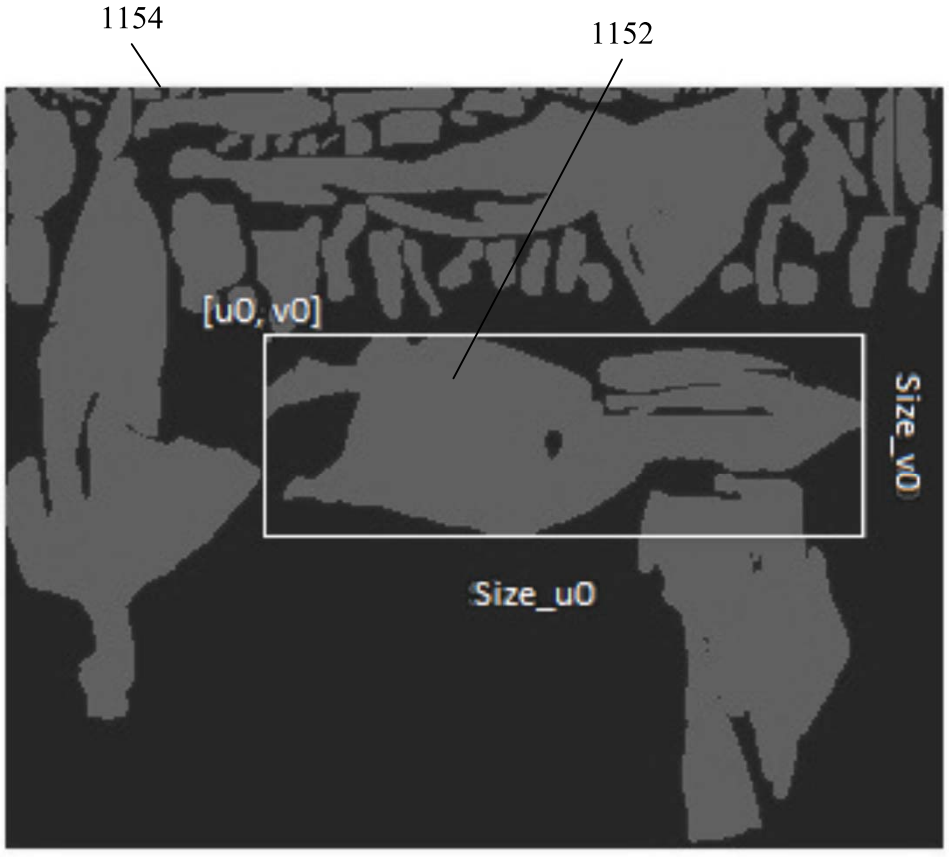


FIG. 11

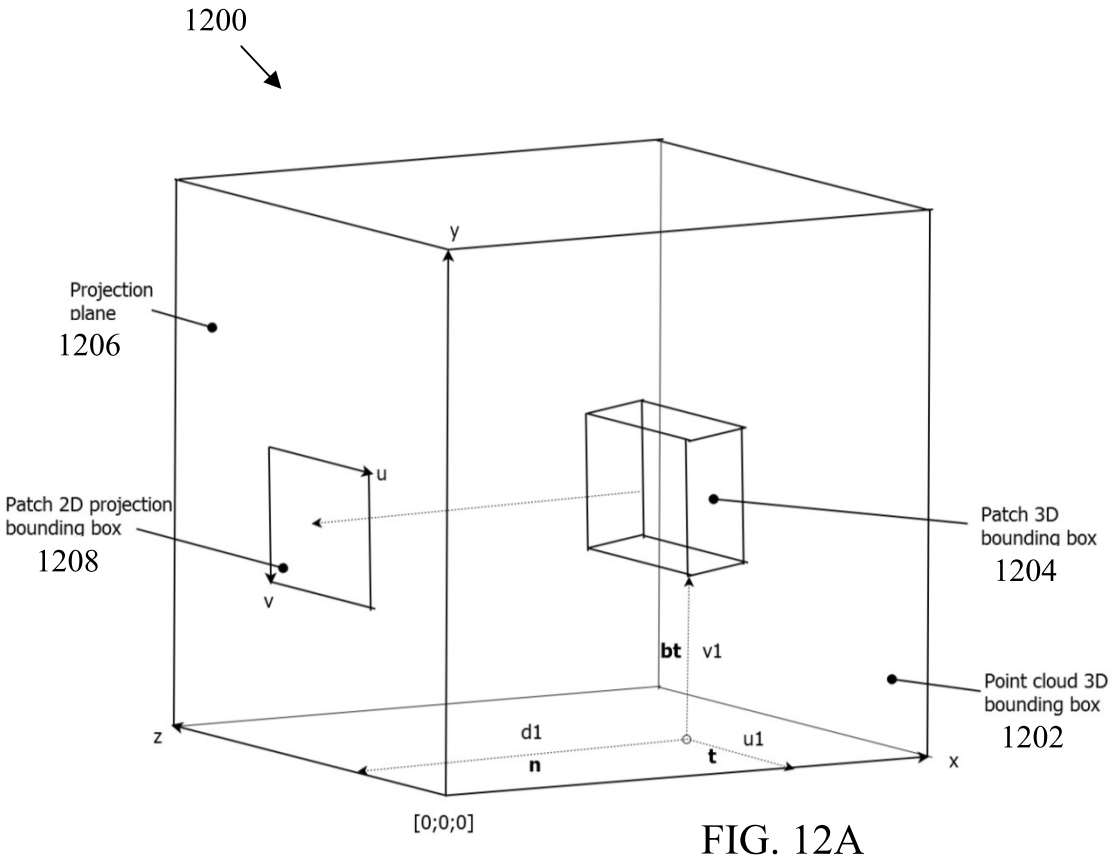


FIG. 12A

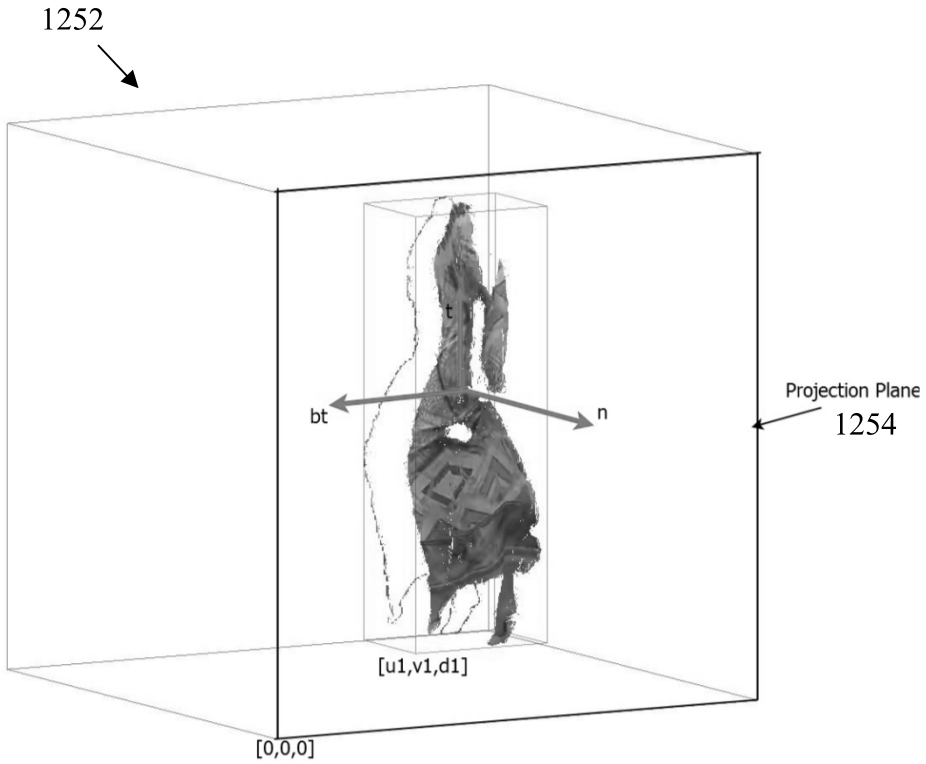


FIG. 12B

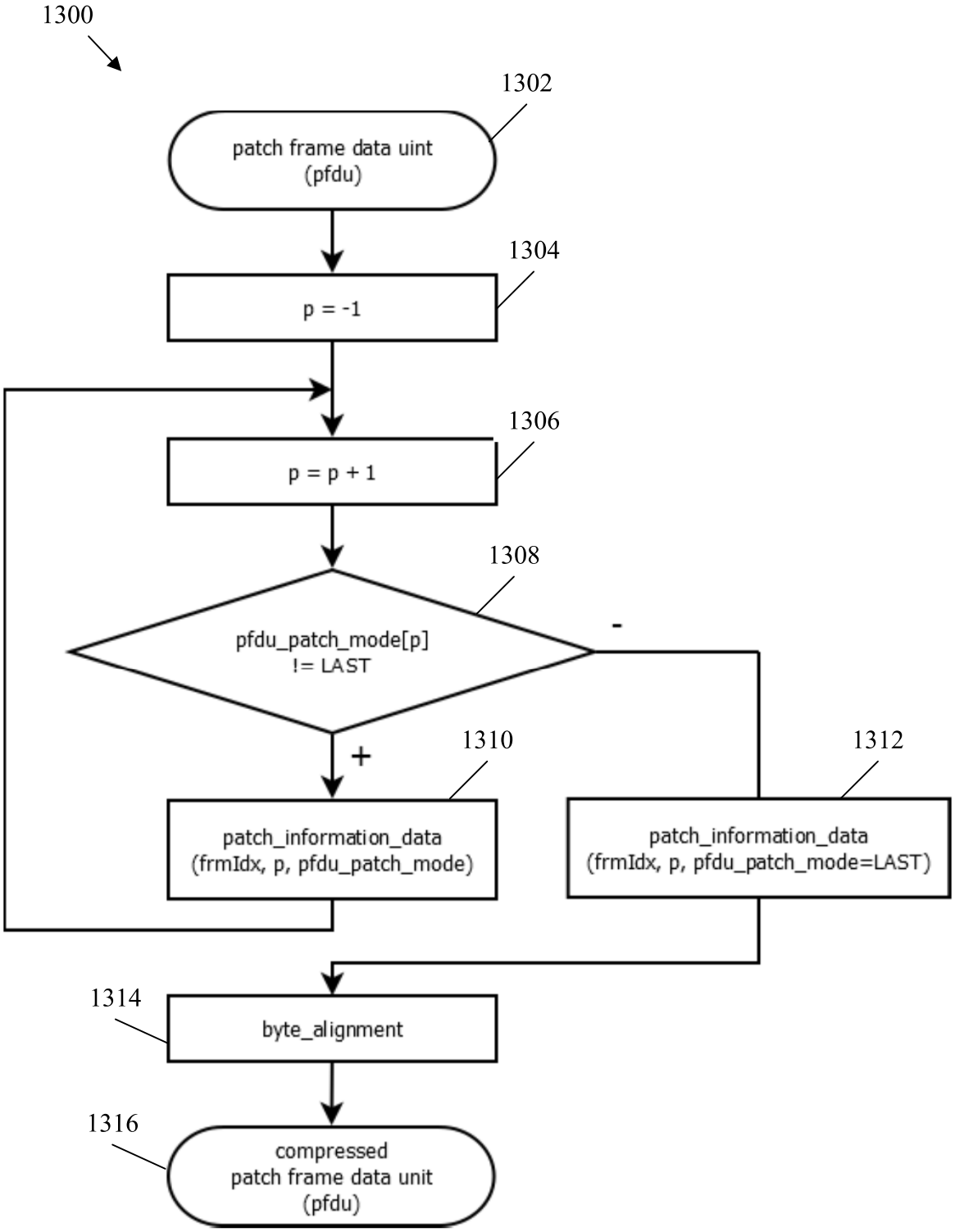

1300
1302
patch frame data uint
(pfdu)
1304
p = -1
1306
p = p + 1
1308
pfdu_patch_mode[p]
!= LAST
-
+
1310
patch_information_data
(frmIdx, p, pfdu_patch_mode)
1312
patch_information_data
(frmIdx, p, pfdu_patch_mode=LAST)
1314
byte_alignment
1316
compressed
patch frame data unit
(pfdu)

FIG. 13

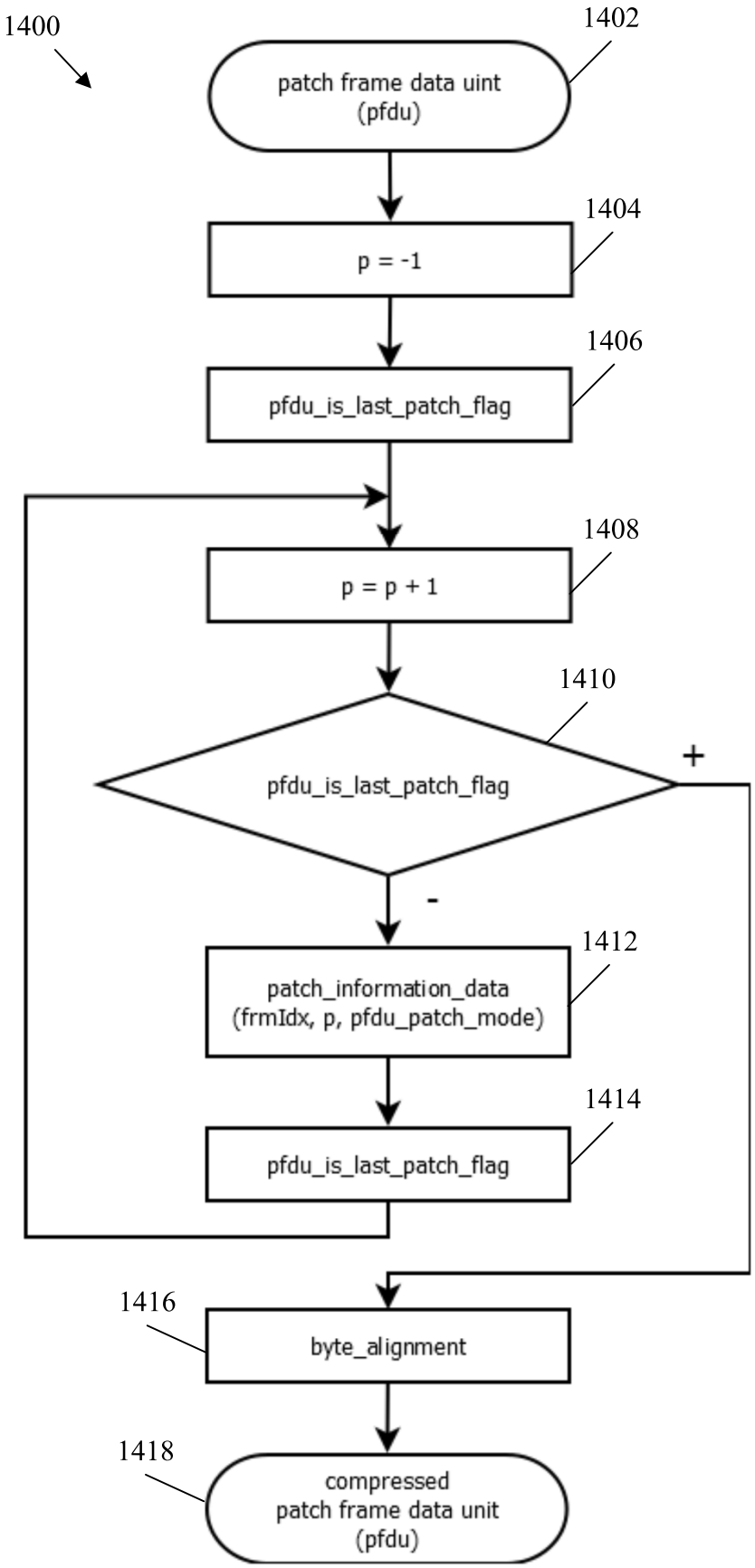

1400
1402
patch frame data uint
(pfdu)
1404
p = -1
1406
pfdu_is_last_patch_flag
1408
p = p + 1
1410
pfdu_is_last_patch_flag
+
-
1412
patch_information_data
(frmIdx, p, pfdu_patch_mode)
1414
pfdu_is_last_patch_flag
1416
byte_alignment
1418
compressed
patch frame data unit
(pfdu)

FIG. 14

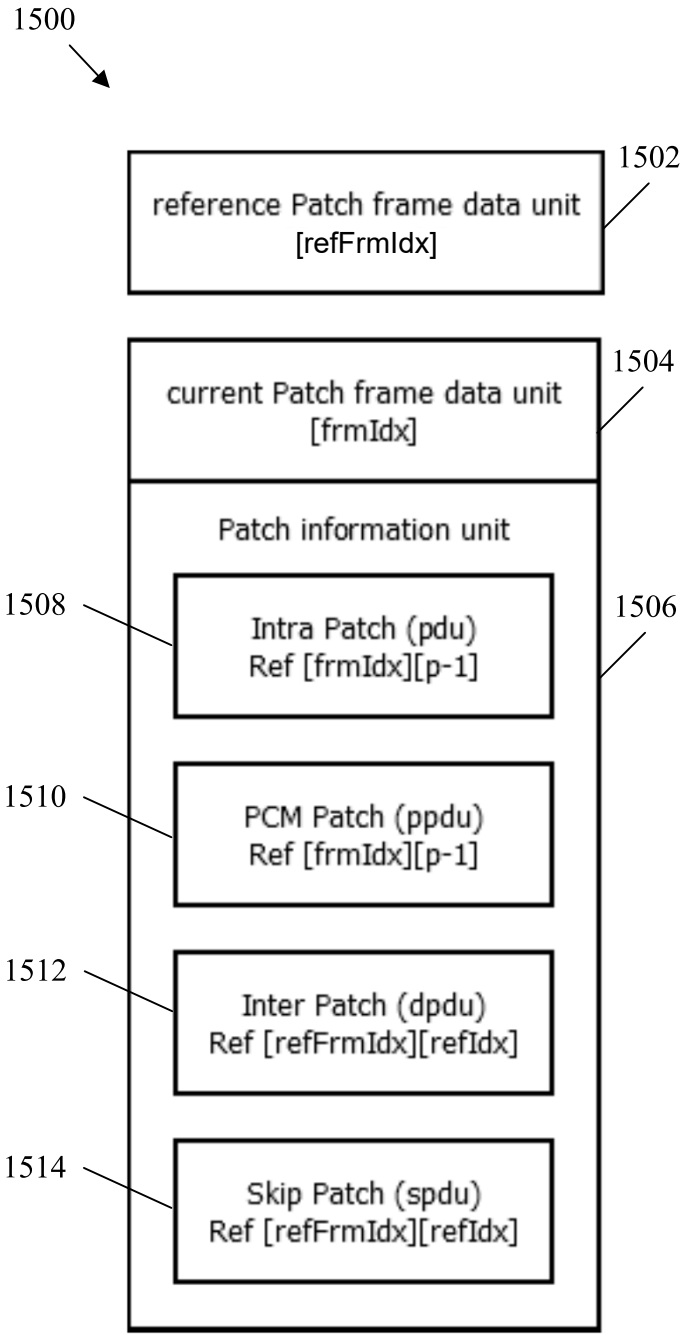

1500
reference Patch frame data unit
[refFrmIdx]
1502
current Patch frame data unit
[frmIdx]
1504
Patch information unit
1508
Intra Patch (pdu)
Ref [frmIdx][p-1]
1506
1510
PCM Patch (ppdu)
Ref [frmIdx][p-1]
1512
Inter Patch (dpdu)
Ref [refFrmIdx][refIdx]
1514
Skip Patch (spdu)
Ref [refFrmIdx][refIdx]

FIG. 15

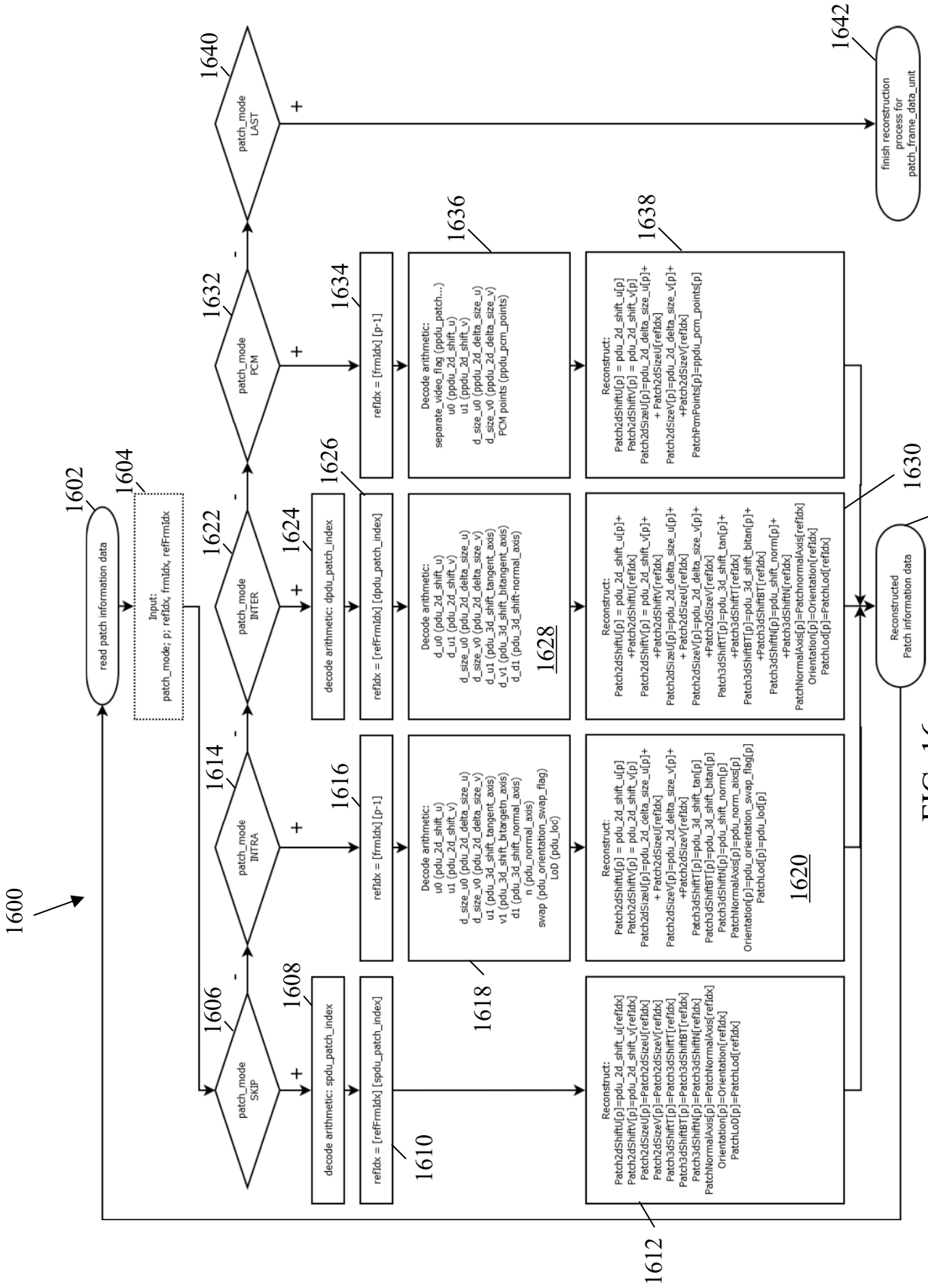

1600
1602
read patch information data
1604
Input:
patch_mode; p; refIdx; frm1dx; refFrm1dx
1606
patch_mode SKIP
1608
decode arithmetic: spdu_patch_index
refIdx = [refFrm1dx] [spdu_patch_index]
1610
1612
Reconstruct:
Patch2dShiftU[p]=pdu_2d_shift_u[refIdx]
Patch2dShiftV[p]=pdu_2d_shift_v[refIdx]
Patch2dSizeU[p]=Patch2dSizeU[refIdx]
Patch2dSizeV[p]=Patch2dSizeV[refIdx]
Patch3dShiftT[p]=Patch3dShiftT[refIdx]
Patch3dShiftBT[p]=Patch3dShiftBT[refIdx]
Patch3dShiftN[p]=Patch3dShiftN[refIdx]
PatchNormalAxis[p]=PatchNormalAxis[refIdx]
Orientation[p]=Orientation[refIdx]
PatchLoD[p]=PatchLod[refIdx]
1614
patch_mode INTRA
1616
refIdx = [frm1dx] [p-1]
1618
Decode arithmetic:
u0 (pdu_2d_shift_u)
u1 (pdu_2d_shift_v)
d_size_u0 (pdu_2d_delta_size_u)
d_size_v0 (pdu_2d_delta_size_v)
u1 (pdu_3d_shift_tangent_axis)
v1 (pdu_3d_shift_bitargetn_axis)
d1 (pdu_3d_shift_normal_axis)
n (pdu_normal_axis)
swap (pdu_orientation_swap_flag)
LoD (pdu_lod)
1620
Reconstruct:
Patch2dShiftU[p] = pdu_2d_shift_u[p]
Patch2dShiftV[p] = pdu_2d_shift_v[p]
Patch2dSizeU[p]=pdu_2d_celta_size_u[p]+
+ Patch2dSizeU[refIdx]
Patch2dSizeV[p]=pdu_2d_delta_size_v[p]+
+Patch2dSizeV[refIdx]
Patch3dShiftT[p]=pdu_3d_shift_tan[p]
Patch3dShiftBT[p]=pdu_3d_shift_bitan[p]
Patch3dShiftN[p]=pdu_shift_norm[p]
PatchNormalAxis[p]=pdu_norm_axis[p]
Orientation[p]=pdu_orientation_swap_flag[p]
PatchLod[p]=pdu_lod[p]
1622
patch_mode INTER
1624
decode arithmetic: dpdu_patch_index
refIdx = [refFrm1dx] [dpdu_patch_index]
1626
Decode arithmetic:
d_u0 (pdu_2d_shift_u)
d_u1 (pdu_2d_shift_v)
d_size_u0 (pdu_2d_delta_size_u)
d_size_v0 (pdu_2d_delta_size_v)
d_u1 (pdu_3d_shift_tangent_axis)
d_v1 (pdu_3d_shift_bitangent_axis)
d_d1 (pdu_3d_shift-normal_axis)
1628
Reconstruct:
Patch2dShiftU[p] = pdu_2d_shift_u[p]+
+Patch2dShiftU[refIdx]
Patch2dShiftV[p] = pdu_2d_shift_v[p]+
+Patch2dShiftV[refIdx]
Patch2dSizeU[p]=pdu_2d_delta_size_u[p]+
+ Patch2dSizeU[refIdx]
Patch2dSizeV[p]=pdu_2d_delta_size_v[p]+
+Patch2dSizeV[refIdx]
Patch3dShiftT[p]=pdu_3d_shift_tan[p]+
+Patch3dShiftT[refIdx]
Patch3dShiftBT[p]=pdu_3d_shift_bitan[p]+
+Patch3dShiftBT[refIdx]
Patch3dShiftN[p]=pdu_shift_norm[p]+
+Patch3dShiftN[refIdx]
PatchNormalAxis[p]=PatchNormalAxis[refIdx]
Orientation[p]=Orientation[refIdx]
PatchLod[p]=PatchLod[refIdx]
1630
1632
patch_mode PCM
1634
refIdx = [frm1dx] [p-1]
1636
Decode arithmetic:
separate_video_flag (ppdu_patch...)
u0 (ppdu_2d_shift_u)
u1 (ppdu_2d_shift_v)
d_size_u0 (ppdu_2d_delta_size_u)
d_size_v0 (ppdu_2d_delta_size_v)
PCM points (ppdu_pcm_points)
1638
Reconstruct:
Patch2dShiftU[p] = pdu_2d_shift_u[p]
Patch2dShiftV[p] = pdu_2d_shift_v[p]
Patch2dSizeU[p]=pdu_2d_delta_size_u[p]+
+ Patch2dSizeU[refIdx]
Patch2dSizeV[p]=pdu_2d_delta_size_v[p]+
+Patch2dSizeV[refIdx]
PatchPomPoints[p]=ppdu_pcm_points[p]
1640
patch_mode LAST
1642
finish reconstruction process for patch_frame_data_unit
1650
Reconstructed Patch information data

FIG. 16

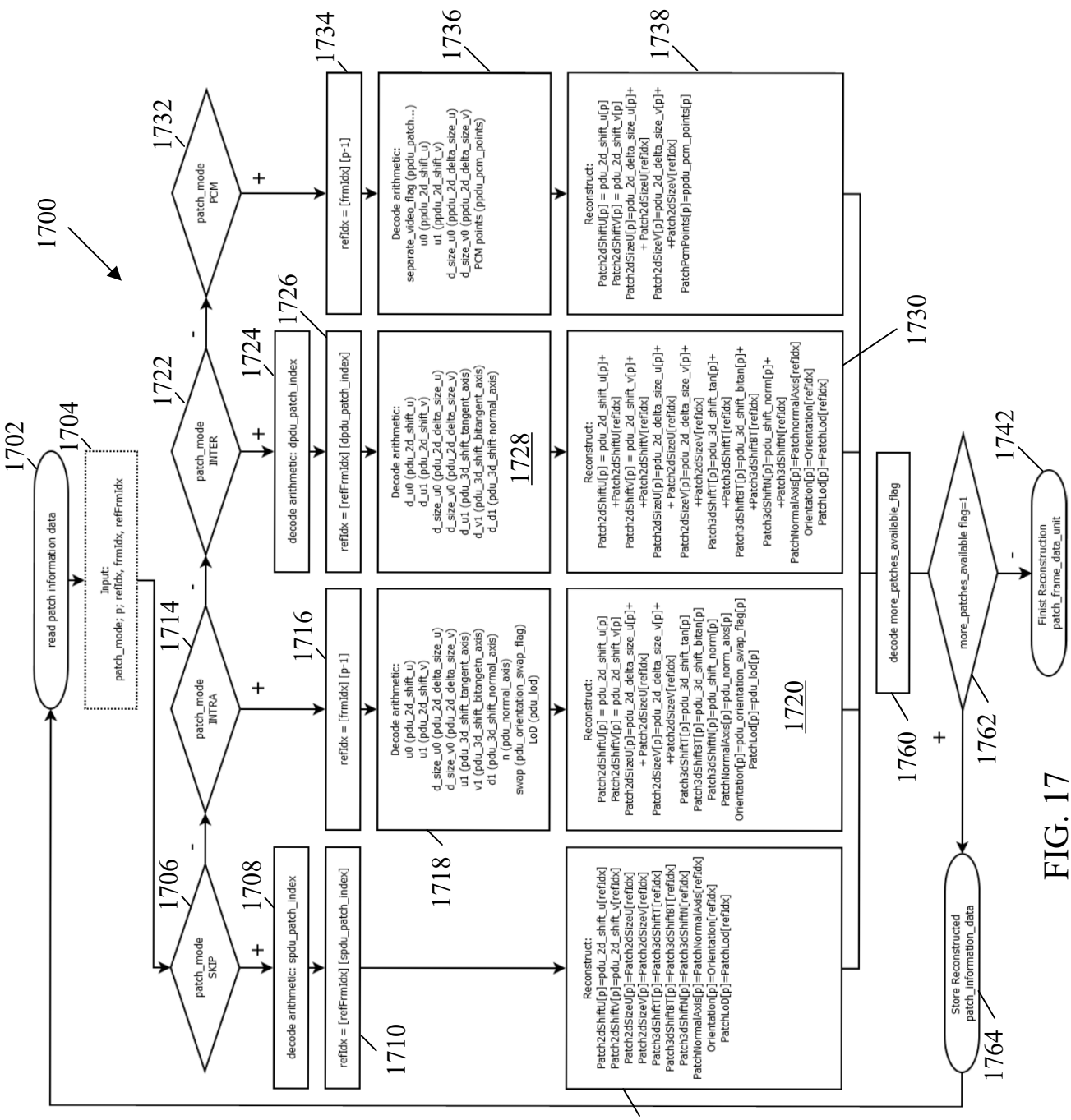

1700
1702
read patch information data
1704
Input:
patch_mode; p; refIdx, frmIdx, refFrmIdx
1706
patch_mode SKIP
1708
decode arithmetic: spdu_patch_index
refIdx = [refFrmIdx] [spdu_patch_index]
1710
1712
Reconstruct:
Patch2dShiftU[p]=pdu_2d_shift_u[refIdx]
Patch2dShiftV[p] = pdu_2d_shift_v[refIdx]
Patch2dSizeU[p]=Patch2dSizeU[refIdx]
Patch2dSizeV[p]=Patch2dSizeV[refIdx]
Patch3dShiftT[p]=Patch3dShiftT[refIdx]
Patch3dShiftBT[p]=Patch3dShiftBT[refIdx]
Patch3dShiftN[p]=Patch3dShiftN[refIdx]
PatchNormalAxis[p]=PatchNormalAxis[refIdx]
Orientation[p]=Orientation[refIdx]
PatchLoD[p]=PatchLod[refIdx]
1714
patch_mode INTRA
1716
refIdx = [frmIdx] [p-1]
1718
Decode arithmetic:
u0 (pdu_2d_shift_u)
u1 (pdu_2d_shift_v)
d_size_u0 (pdu_2d_delta_size_u)
d_size_v0 (pdu_2d_delta_size_v)
u1 (pdu_3d_shift_tangent_axis)
v1 (pdu_3d_shift_bitangetn_axis)
d1 (pdu_3d_shift_normal_axis)
n (pdu_normal_axis)
swap (pdu_orientation_swap_flag)
LoD (pdu_lod)
1720
Reconstruct:
Patch2dShiftU[p] = pdu_2d_shift_u[p]
Patch2dShiftV[p] = pdu_2d_shift_v[p]
Patch2dSizeU[p]=pdu_2d_delta_size_u[p]+
+ Patch2dSizeU[refIdx]
Patch2dSizeV[p]=pdu_2d_delta_size_v[p]+
+Patch2dSizeV[refIdx]
Patch3dShiftT[p]=pdu_3d_shift_tan[p]
Patch3dShiftBT[p]=pdu_3d_shift_bitan[p]
Patch3dShiftN[p]=pdu_shift_norm[p]
PatchNormalAxis[p]=pdu_norm_aixs[p]
Orientation[p]=pdu_orientation_swap_flag[p]
PatchLoD[p]=pdu_lod[p]
1722
patch_mode INTER
1724
decode arithmetic: dpdu_patch_index
refIdx = [refFrmIdx] [dpdu_patch_index]
1726
Decode arithmetic:
d_u0 (pdu_2d_shift_u)
d_u1 (pdu_2d_shift_v)
d_size_u0 (pdu_2d_delta_size_u)
d_size_v0 (pdu_2d_delta_size_v)
d_u1 (pdu_3d_shift_tangent_axis)
d_v1 (pdu_3d_shift_bitangent_axis)
d_d1 (pdu_3d_shift-normal_axis)
1728
Reconstruct:
Patch2dShiftU[p] = pdu_2d_shift_u[p]+
+Patch2dShiftU[refIdx]
Patch2dShiftV[p] = pdu_2d_shift_v[p]+
+Patch2dShiftV[refIdx]
Patch2dSizeU[p]=pdu_2d_delta_size_u[p]+
+ Patch2dSizeU[refIdx]
Patch2dSizeV[p]=pdu_2d_delta_size_v[p]+
+Patch2dSizeV[refIdx]
Patch3dShiftT[p]=pdu_3d_shift_tan[p]+
+Patch3dShiftT[refIdx]
Patch3dShiftBT[p]=pdu_3d_shift_bitan[p]+
+Patch3dShiftBT[refIdx]
Patch3dShiftN[p]=pdu_shift_norm[p]+
+Patch3dShiftN[refIdx]
PatchNormalAxis[p]=PatchnormalAxis[refIdx]
Orientation[p]=Orientation[refIdx]
PatchLod[p]=PatchLod[refIdx]
1730
1732
patch_mode PCM
1734
refIdx = [frmIdx] [p-1]
1736
Decode arithmetic:
separate_video_flag (ppdu_patch...)
u0 (ppdu_2d_shift_u)
u1 (ppdu_2d_shift_v)
d_size_u0 (ppdu_2d_delta_size_u)
d_size_v0 (ppdu_2d_delta_size_v)
PCM points (ppdu_pcm_points)
1738
Reconstruct:
Patch2dShiftU[p] = pdu_2d_shift_u[p]
Patch2dShiftV[p] = pdu_2d_shift_v[p]
Patch2dSizeU[p]=pdu_2d_delta_size_u[p]+
+ Patch2dSizeU[refIdx]
Patch2dSizeV[p]=pdu_2d_delta_size_v[p]+
+Patch2dSizeV[refIdx]
PatchPcmPoints[p]=ppdu_pcm_points[p]
1740
decode more_patches_available_flag
1742
more_patches_available flag=1
1760
1762
Finist Reconstruction
patch_frame_data_unit
Store Reconstructed
patch_information_data
1764

FIG. 17

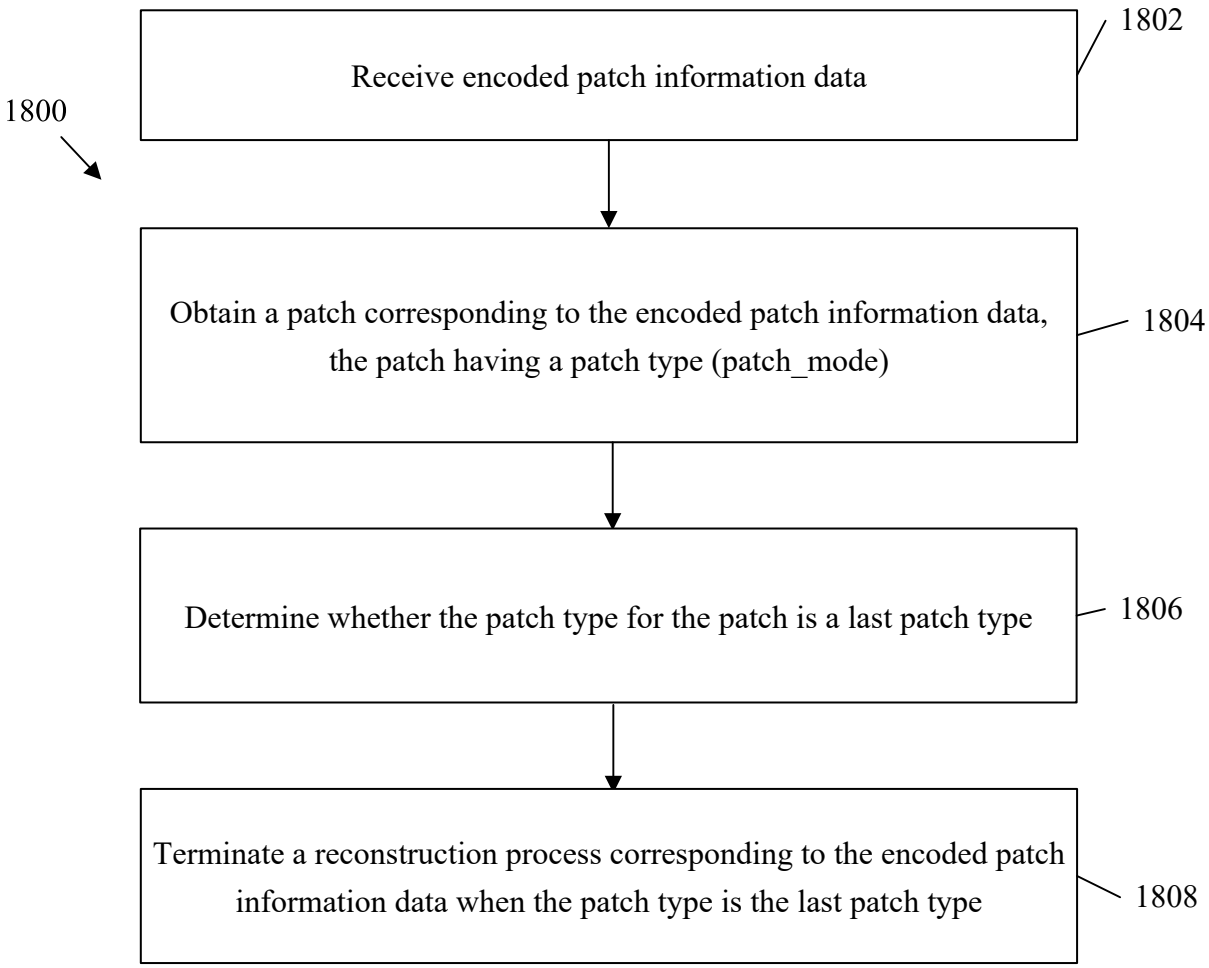

1800
Receive encoded patch information data
1802
Obtain a patch corresponding to the encoded patch information data, the patch having a patch type (patch_mode)
1804
Determine whether the patch type for the patch is a last patch type
1806
Terminate a reconstruction process corresponding to the encoded patch information data when the patch type is the last patch type
1808

FIG. 18

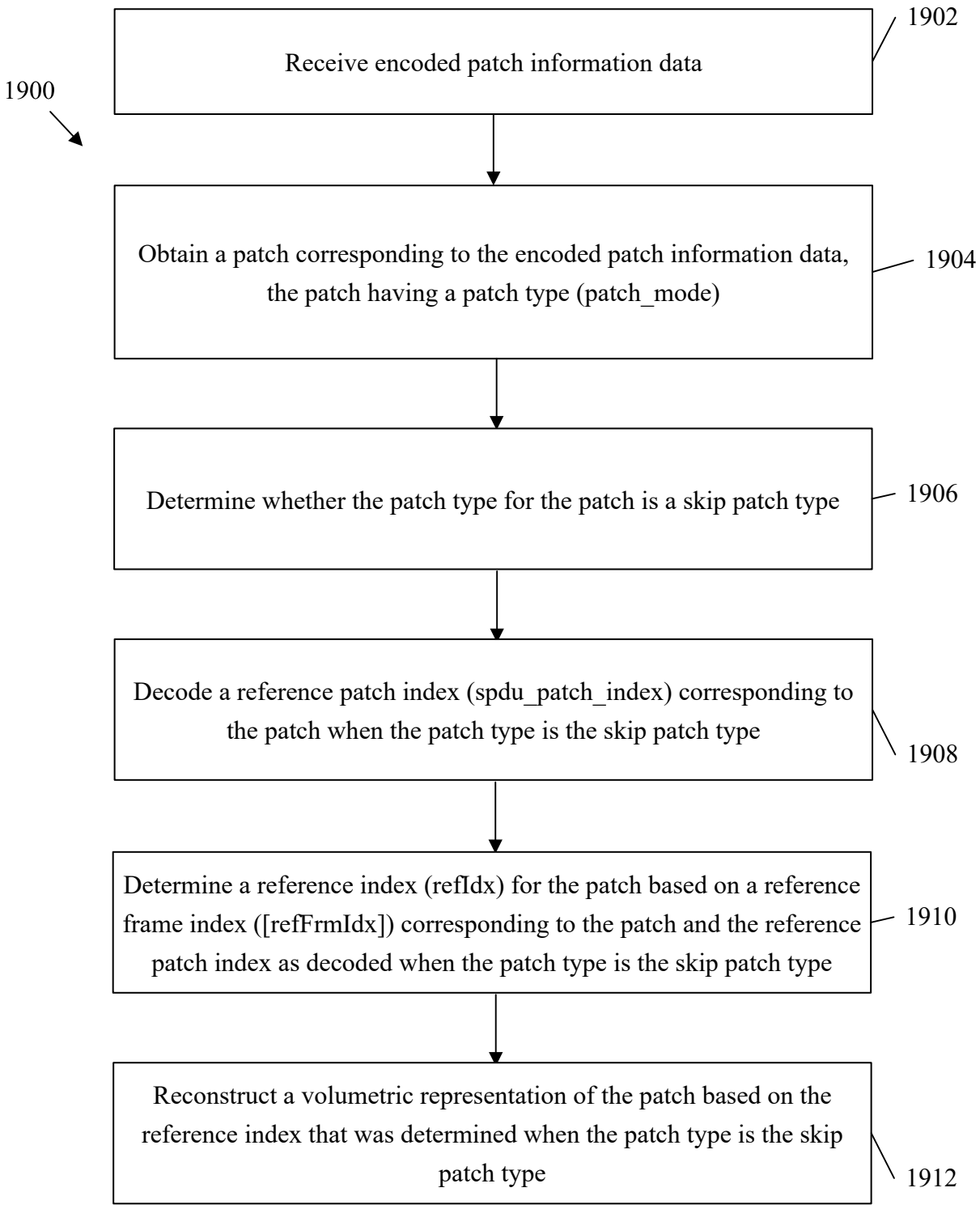

1900
Receive encoded patch information data
1902
Obtain a patch corresponding to the encoded patch information data, the patch having a patch type (patch_mode)
1904
Determine whether the patch type for the patch is a skip patch type
1906
Decode a reference patch index (spdu_patch_index) corresponding to the patch when the patch type is the skip patch type
1908
Determine a reference index (refIdx) for the patch based on a reference frame index ([refFrmIdx]) corresponding to the patch and the reference patch index as decoded when the patch type is the skip patch type
1910
Reconstruct a volumetric representation of the patch based on the reference index that was determined when the patch type is the skip patch type
1912

FIG. 19

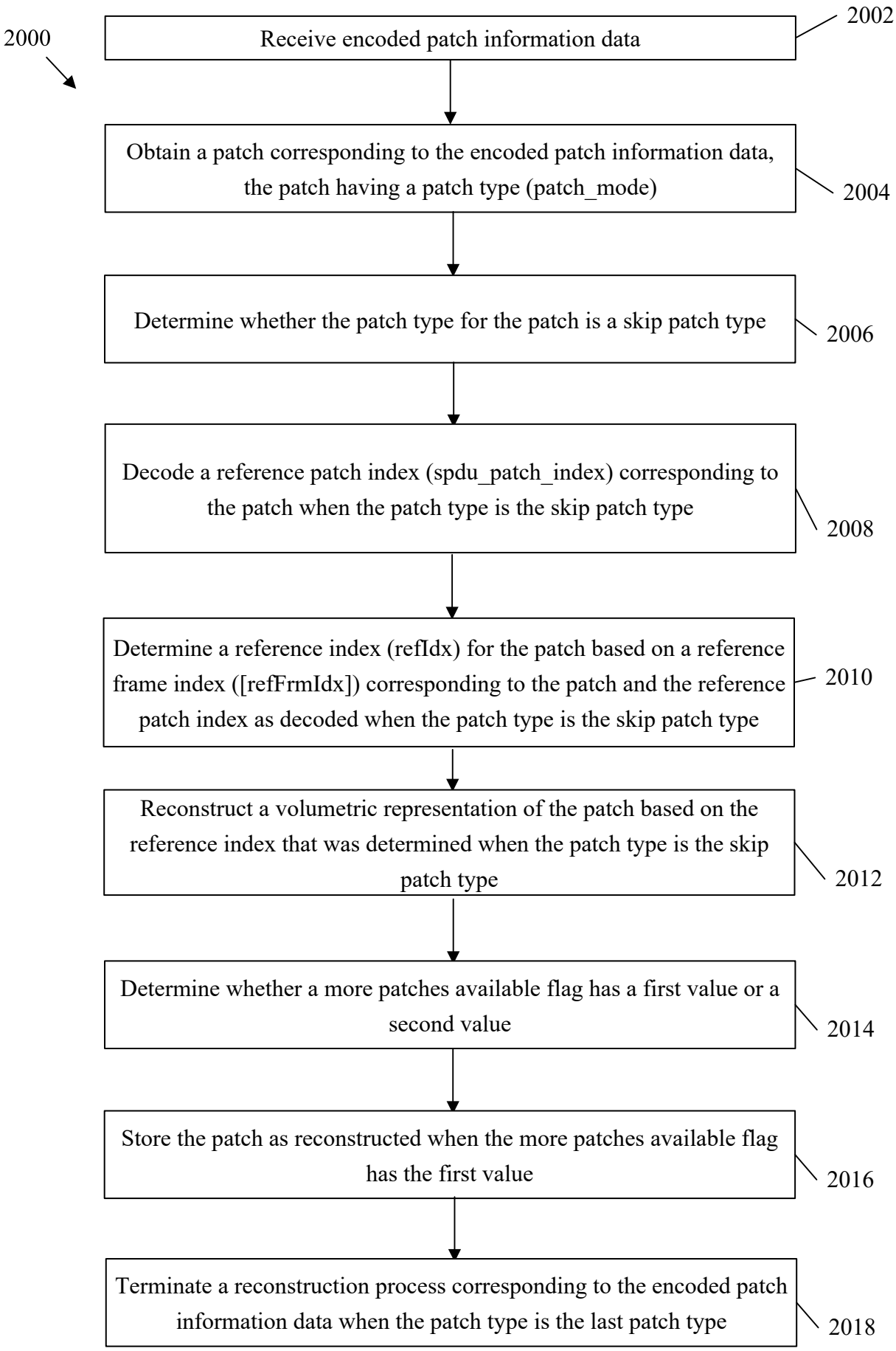

2000
Receive encoded patch information data
2002
Obtain a patch corresponding to the encoded patch information data, the patch having a patch type (patch_mode)
2004
Determine whether the patch type for the patch is a skip patch type
2006
Decode a reference patch index (spdu_patch_index) corresponding to the patch when the patch type is the skip patch type
2008
Determine a reference index (refIdx) for the patch based on a reference frame index ([refFrmIdx]) corresponding to the patch and the reference patch index as decoded when the patch type is the skip patch type
2010
Reconstruct a volumetric representation of the patch based on the reference index that was determined when the patch type is the skip patch type
2012
Determine whether a more patches available flag has a first value or a second value
2014
Store the patch as reconstructed when the more patches available flag has the first value
2016
Terminate a reconstruction process corresponding to the encoded patch information data when the patch type is the last patch type
2018

FIG. 20

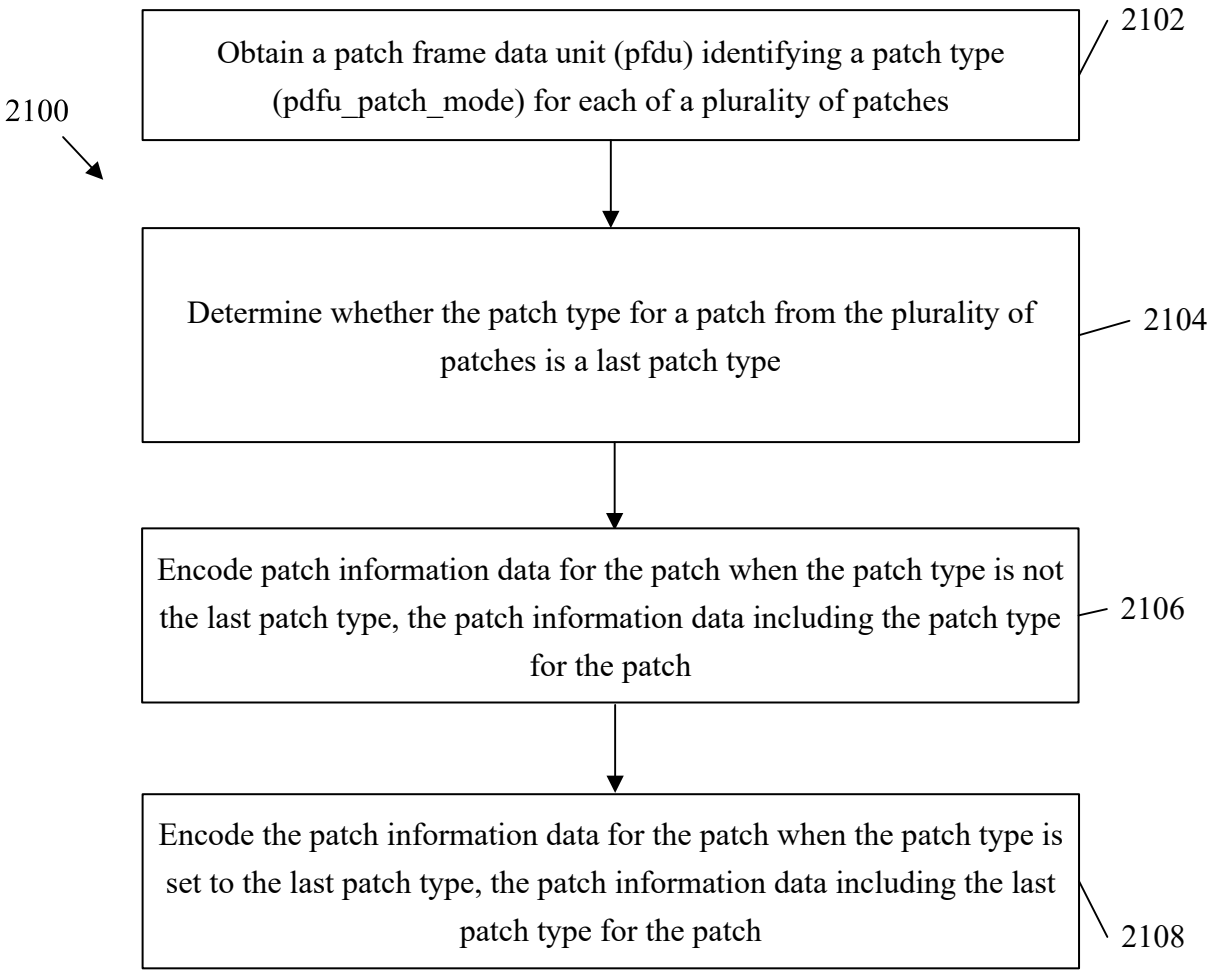

2100
Obtain a patch frame data unit (pfdu) identifying a patch type (pdfu_patch_mode) for each of a plurality of patches
2102
Determine whether the patch type for a patch from the plurality of patches is a last patch type
2104
Encode patch information data for the patch when the patch type is not the last patch type, the patch information data including the patch type for the patch
2106
Encode the patch information data for the patch when the patch type is set to the last patch type, the patch information data including the last patch type for the patch
2108

FIG. 21

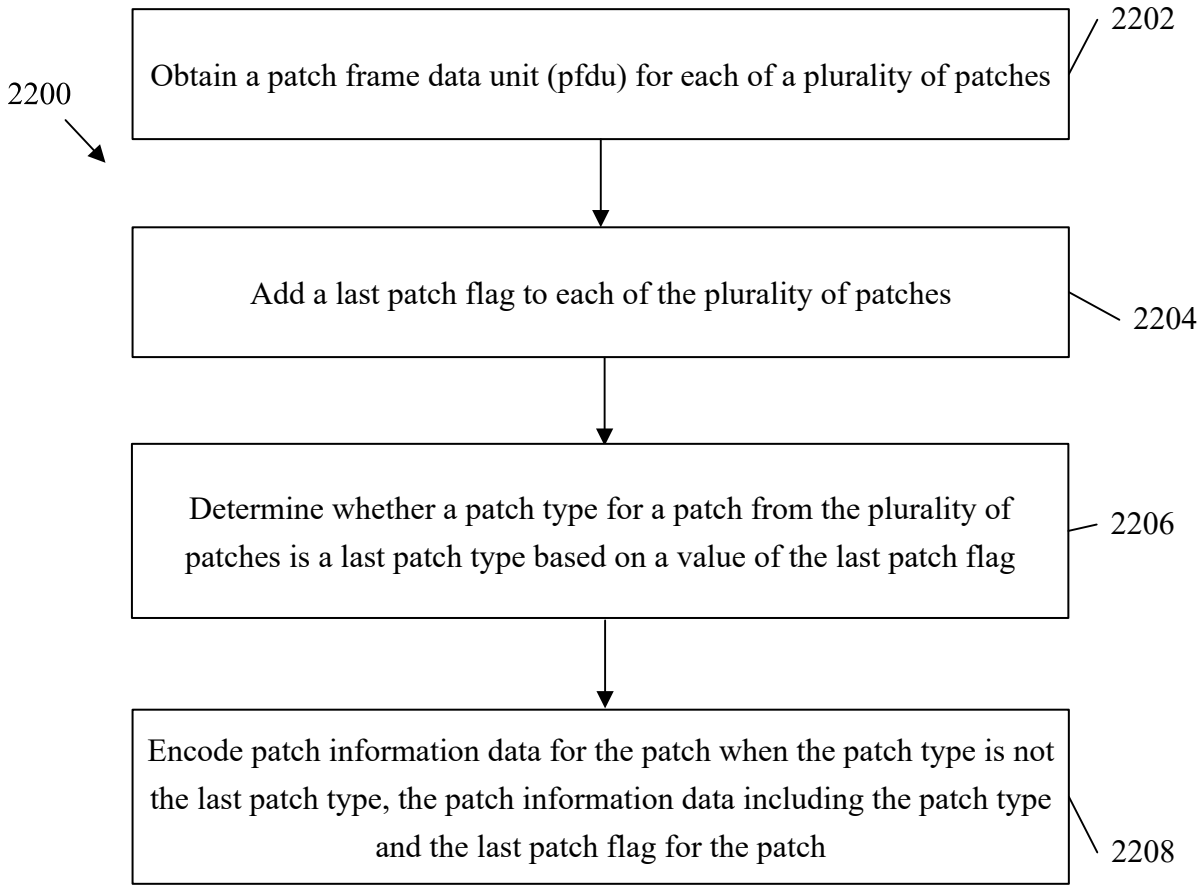

2200
Obtain a patch frame data unit (pfdu) for each of a plurality of patches
2202
Add a last patch flag to each of the plurality of patches
2204
Determine whether a patch type for a patch from the plurality of patches is a last patch type based on a value of the last patch flag
2206
Encode patch information data for the patch when the patch type is not the last patch type, the patch information data including the patch type and the last patch flag for the patch
2208

FIG. 22

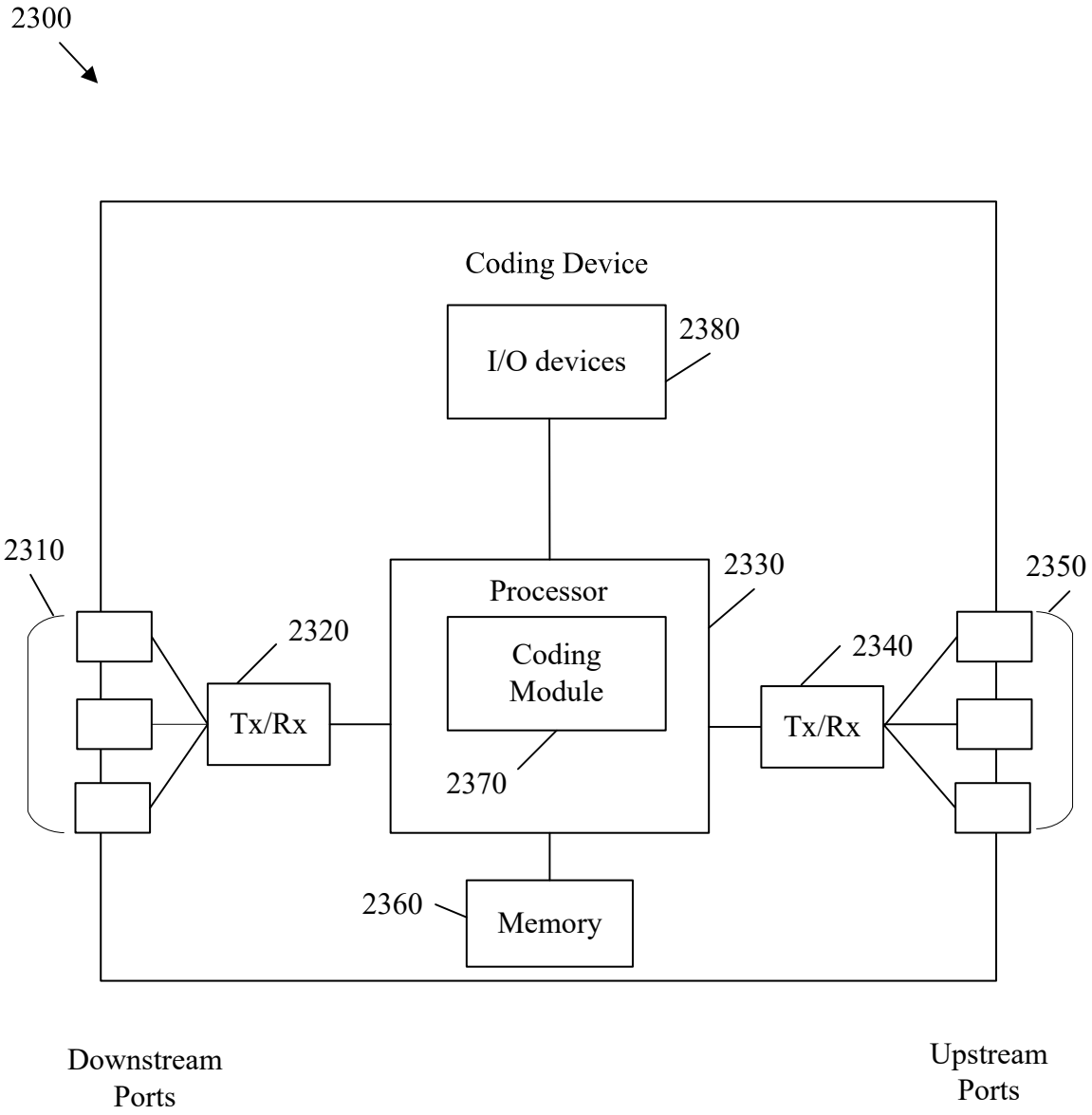

2300
Coding Device
I/O devices
2380
2310
Processor
2330
2350
2320
Coding
Module
2340
Tx/Rx
Tx/Rx
2370
2360
Memory
Downstream
Ports
Upstream
Ports

FIG. 23

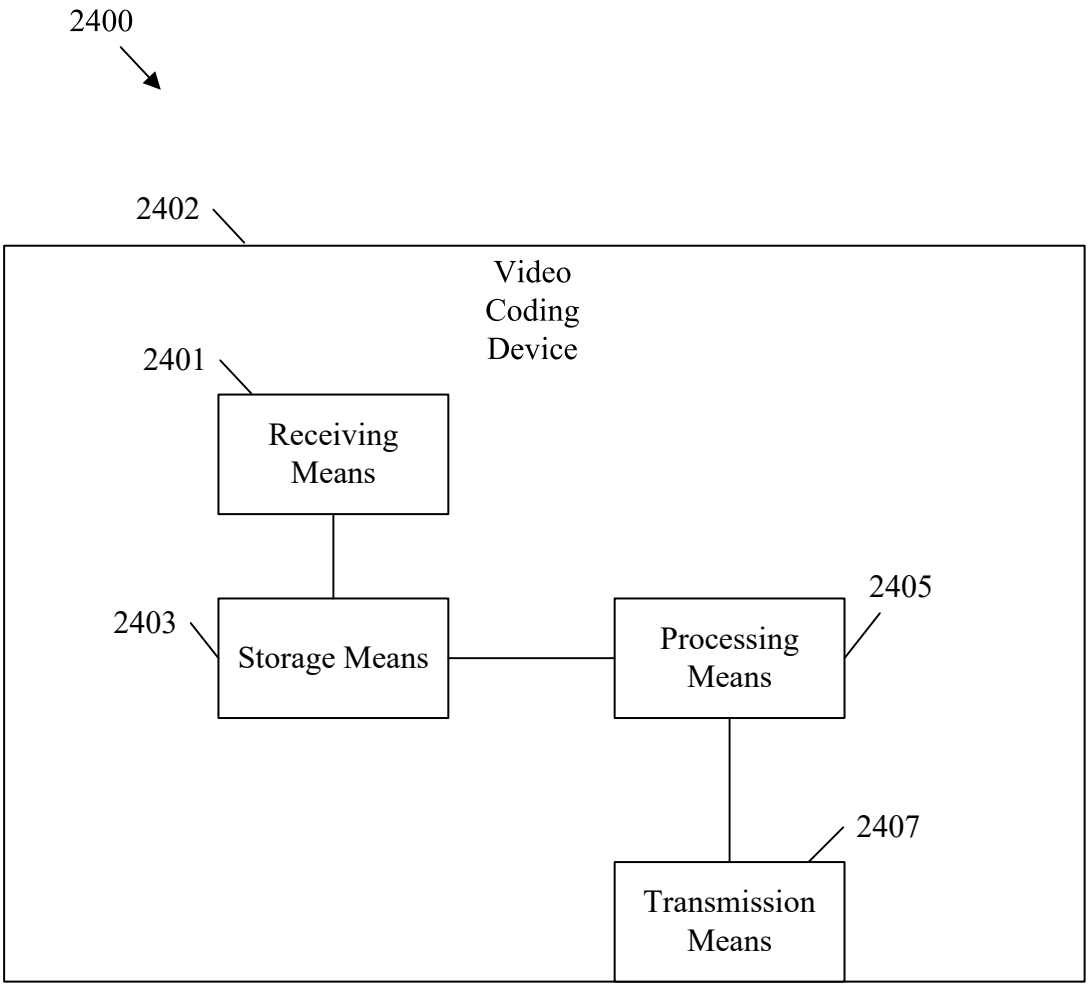

2400
2402
Video
Coding
Device
2401
Receiving
Means
2403
Storage Means
2405
Processing
Means
2407
Transmission
Means

FIG. 24