FIG. 1

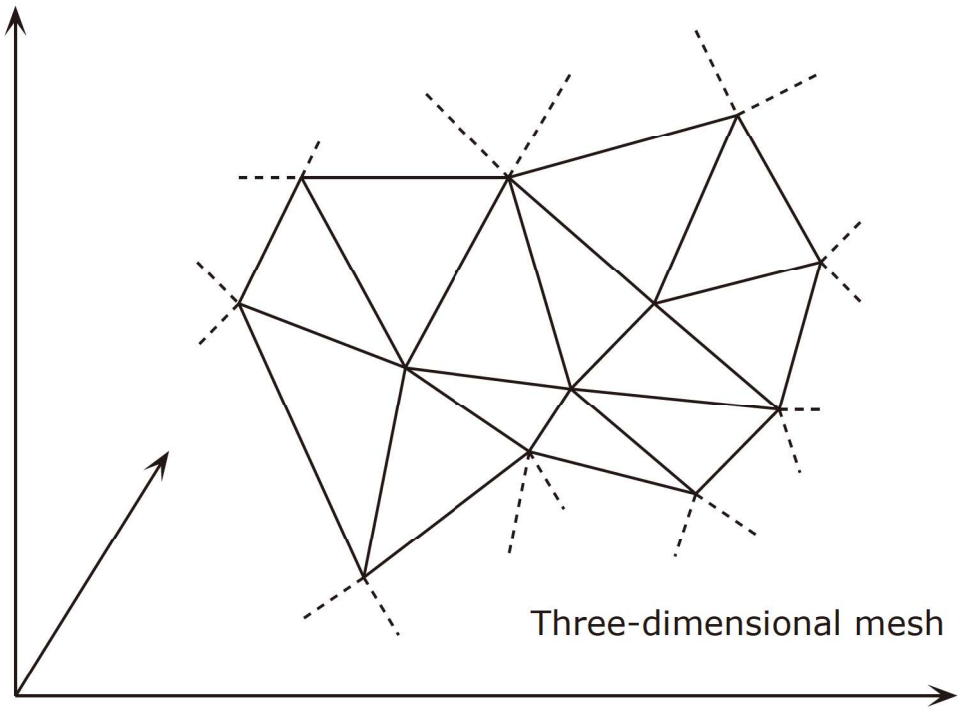


FIG. 2

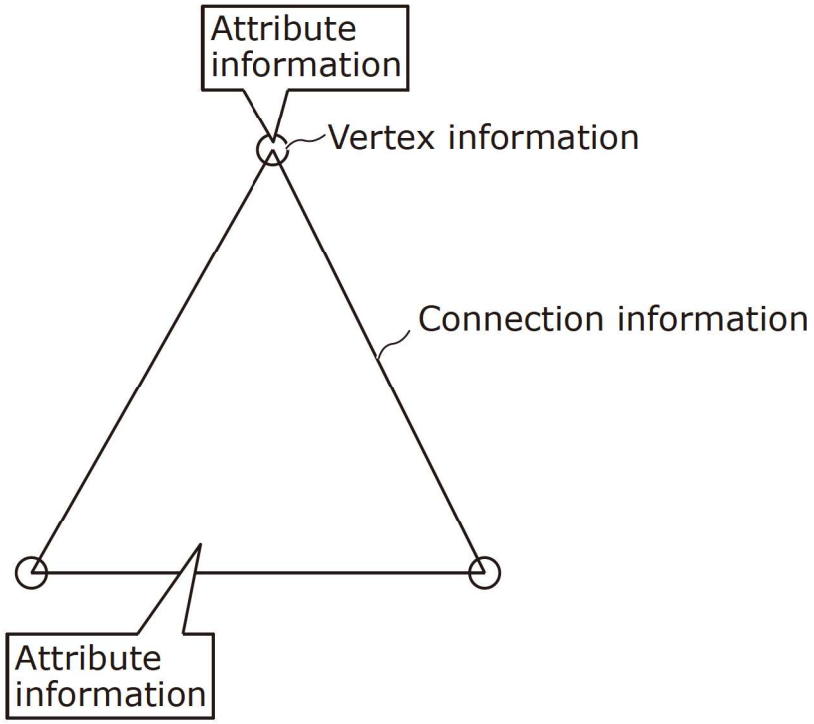

Attribute information
Vertex information
Connection information
Attribute information

FIG. 3

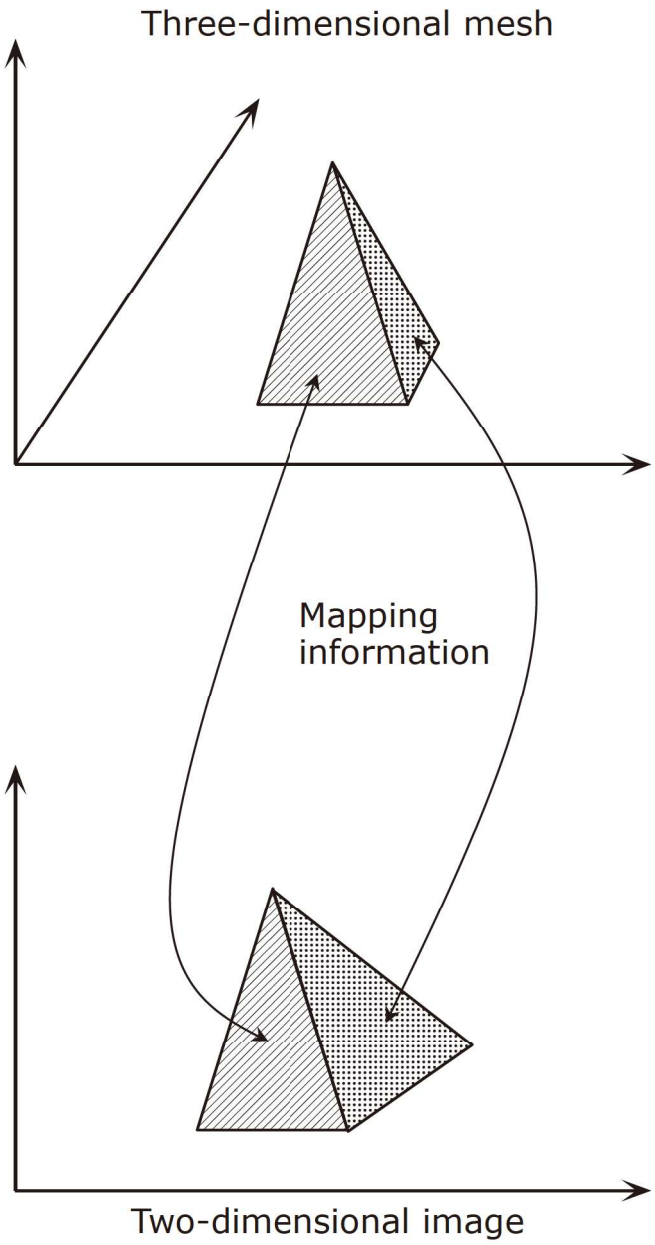

Three-dimensional mesh
Mapping
information
Two-dimensional image

FIG. 4

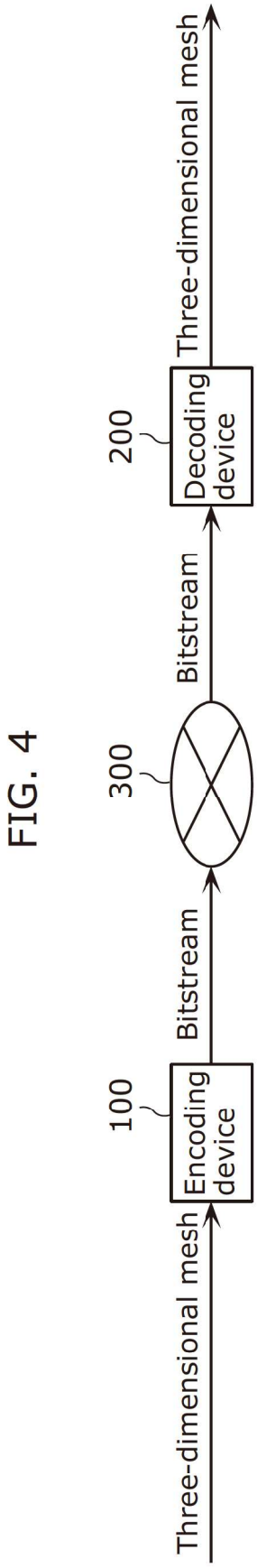

Three-dimensional mesh
100
Encoding device
Bitstream
300
Bitstream
200
Decoding device
Three-dimensional mesh

FIG. 5

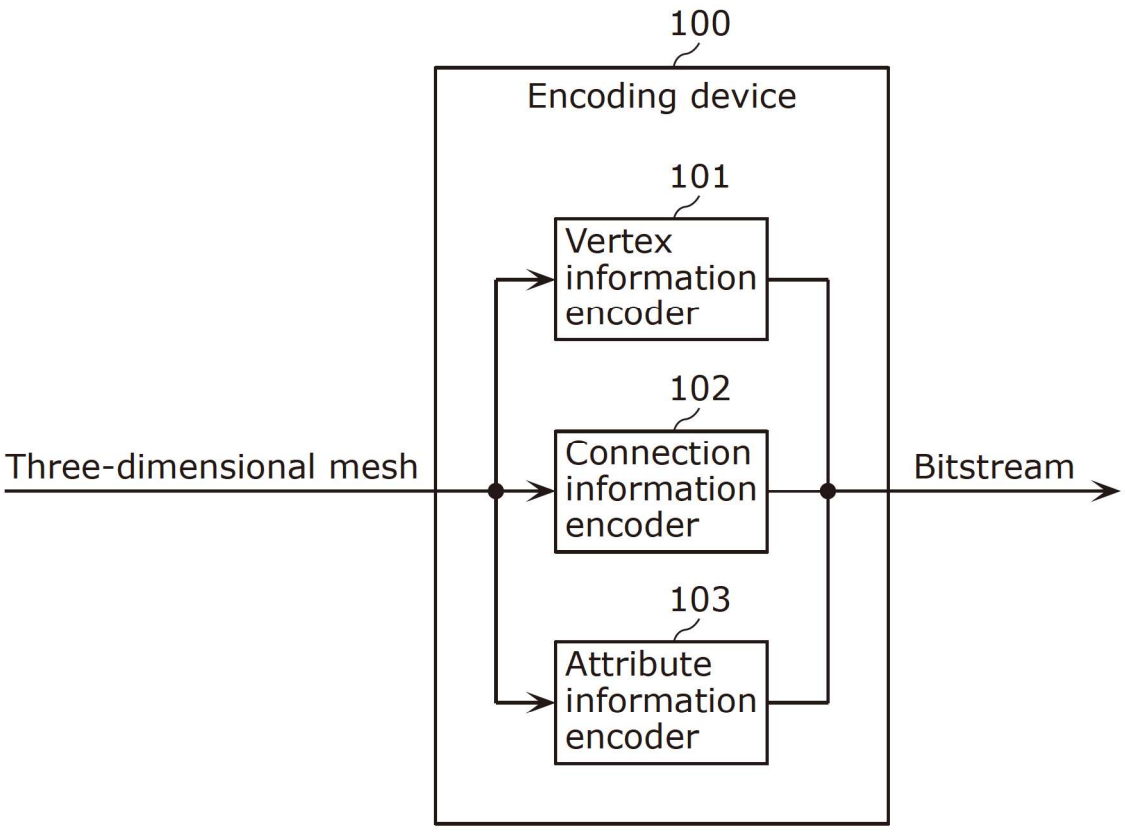

100
Encoding device
101
Vertex information encoder
102
Connection information encoder
103
Attribute information encoder
Three-dimensional mesh
Bitstream

FIG. 6

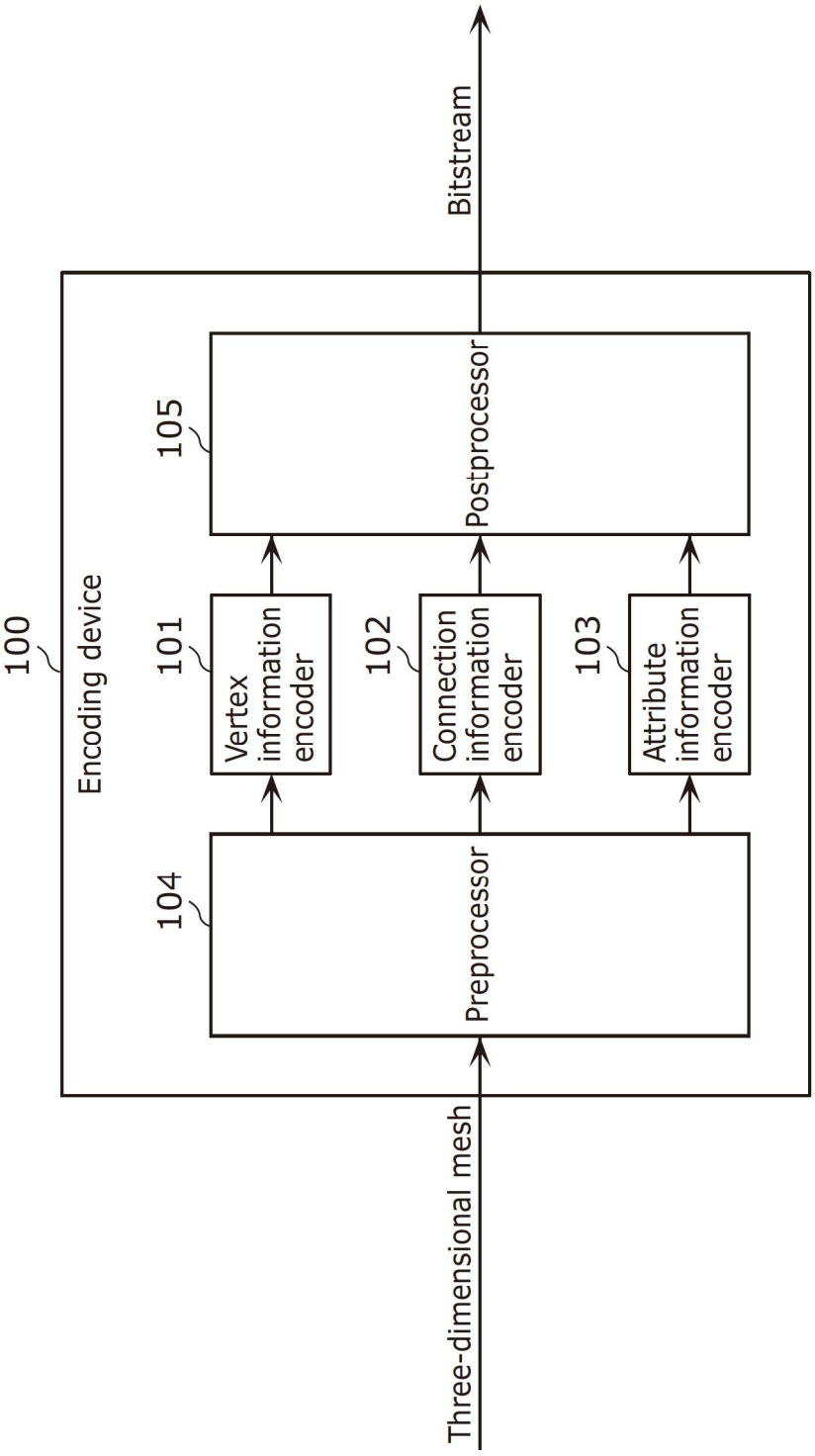

Three-dimensional mesh
100
Encoding device
104
Preprocessor
101
Vertex information encoder
102
Connection information encoder
103
Attribute information encoder
105
Postprocessor
Bitstream

FIG. 7

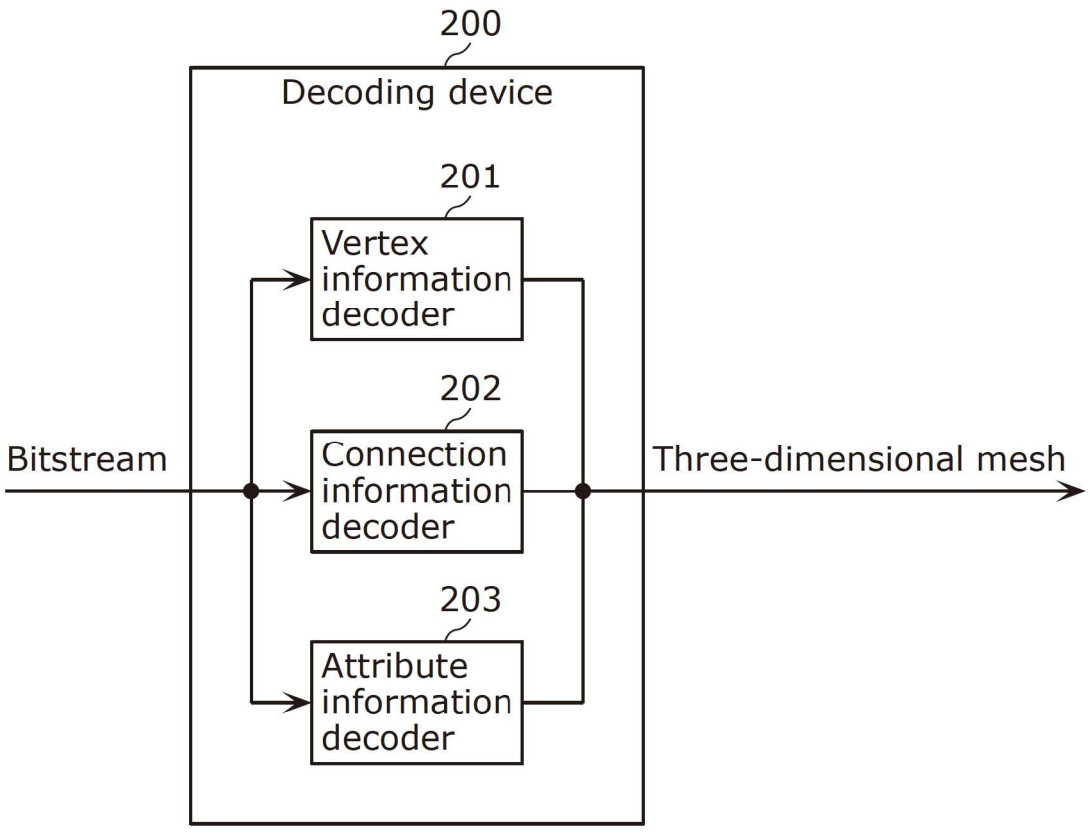

200
Decoding device
201
Vertex information decoder
202
Connection information decoder
203
Attribute information decoder
Bitstream
Three-dimensional mesh

FIG. 8

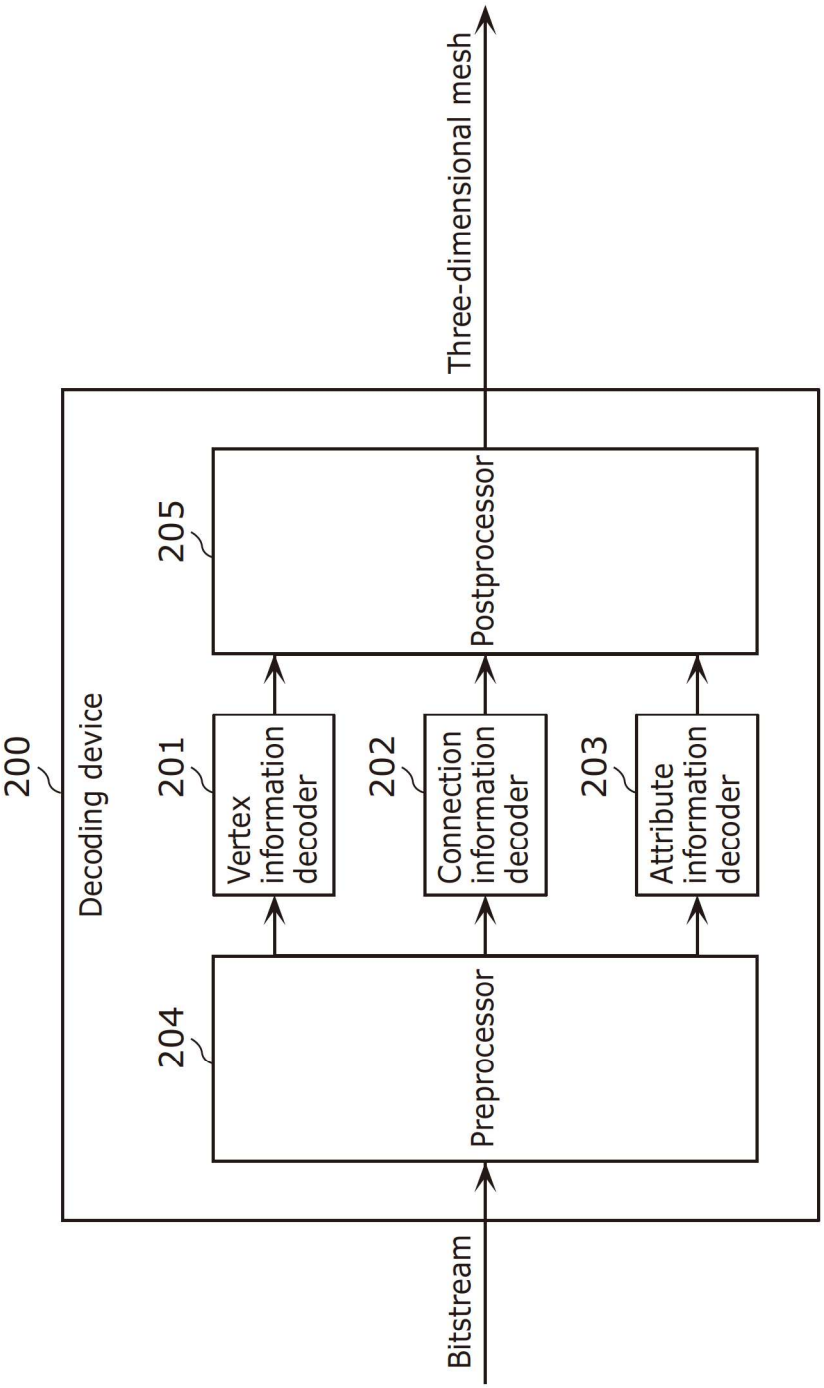

Bitstream
Decoding device
200
204
Preprocessor
201
Vertex information decoder
202
Connection information decoder
203
Attribute information decoder
205
Postprocessor
Three-dimensional mesh

FIG. 9

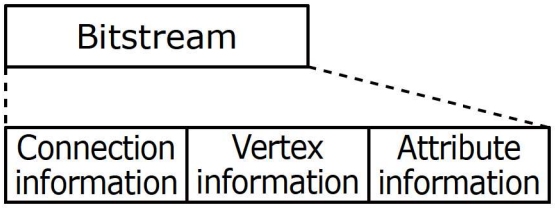


FIG. 10

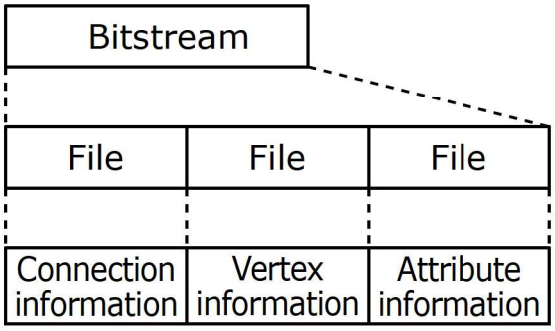


FIG. 11

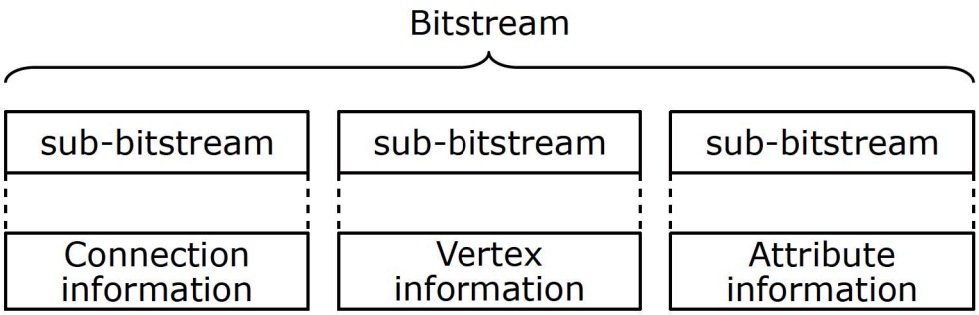


FIG. 12

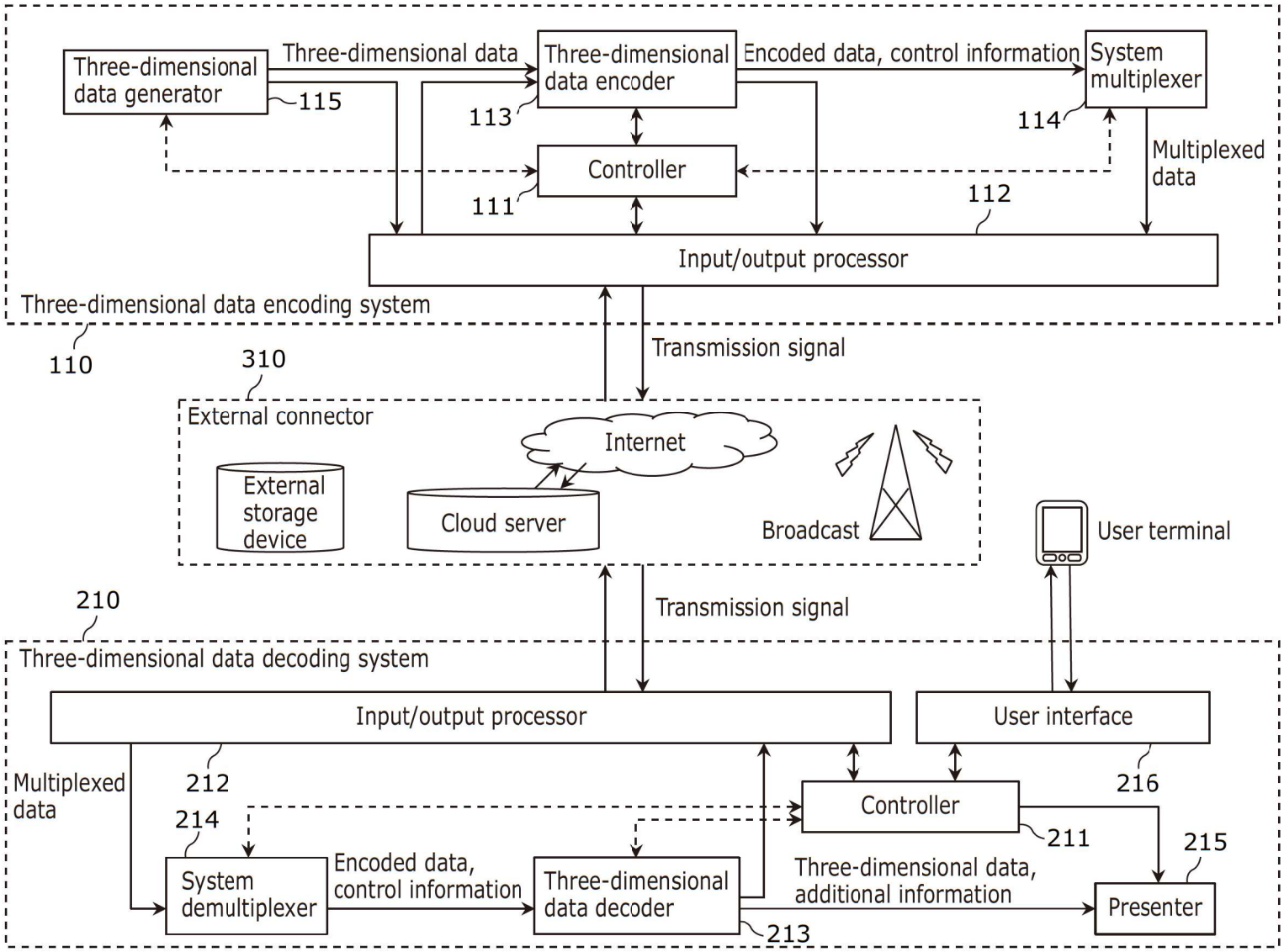

Three-dimensional data generator
115
Three-dimensional data
Three-dimensional data encoder
113
Encoded data, control information
System multiplexer
114
Multiplexed data
Controller
111
112
Input/output processor
Three-dimensional data encoding system
110
310
Transmission signal
External connector
Internet
External storage device
Cloud server
Broadcast
User terminal
210
Transmission signal
Three-dimensional data decoding system
Input/output processor
User interface
Multiplexed data
212
214
216
Controller
211
215
System demultiplexer
Encoded data, control information
Three-dimensional data decoder
Three-dimensional data, additional information
Presenter
213

FIG. 13

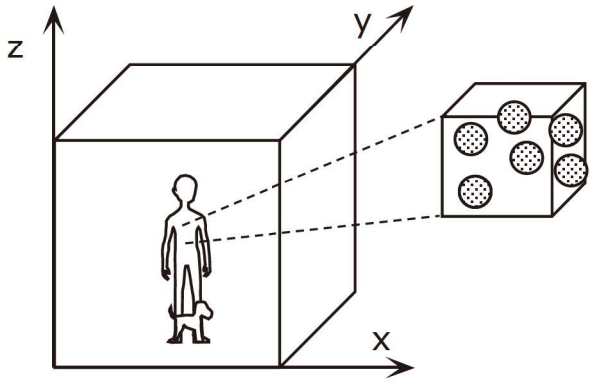


FIG. 14

	Position information	Attribute information
n = 1	$G(1) = (x_1, y_1, z_1)$	$A(1) = (R_1, G_1, B_1)$
n = 2	$G(2) = (x_2, y_2, z_2)$	$A(2) = (R_2, G_2, B_2)$
⋮	⋮	⋮
n = N	$G(N) = (x_N, y_N, z_N)$	$A(N) = (R_N, G_N, B_N)$

FIG. 15

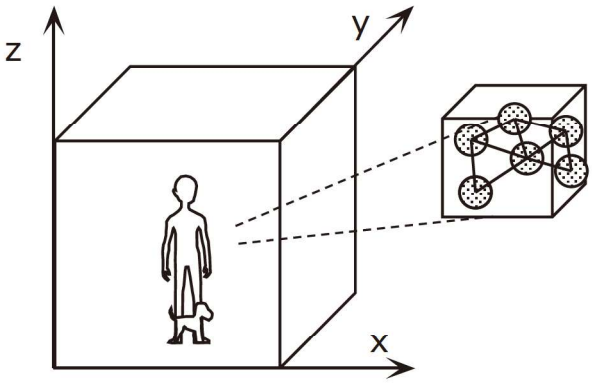


FIG. 16

		Position information	Attribute information
Vertex information	n = 1	G(1) = (x_1, y_1, z_1)	A1(1) = (R_1, G_1, B_1)
	n = 2	G(2) = (x_2, y_2, z_2)	A1(2) = (R_2, G_2, B_2)
	⋮	⋮	⋮
	n = N	G(N) = (x_N, y_N, z_N)	A1(N) = (R_N, G_N, B_N)
Attribute information	m = 1	A2(1) = (U_1, V_1)	
	⋮	⋮	
	m = M	A2(M) = (U_M, V_M)	
Connection information	k = 1	n[1, 3, 4]	m[2, 4, 6]
	⋮	⋮	⋮
	k = K	n[1, N-1, N]	m[10, 20, M]

FIG. 17

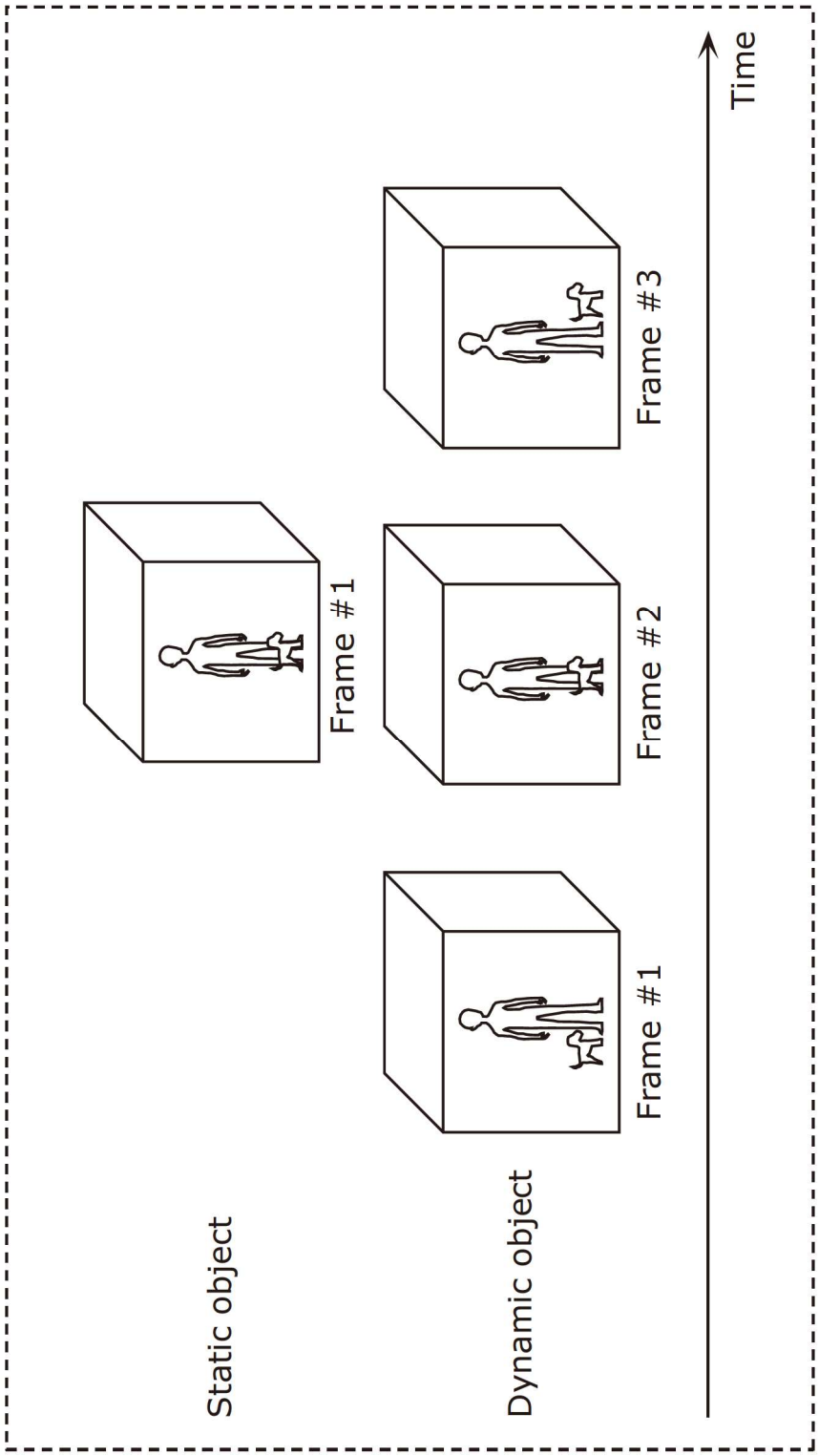

Time
Frame #3
Frame #2
Frame #1
Frame #1
Static object
Dynamic object

FIG. 18

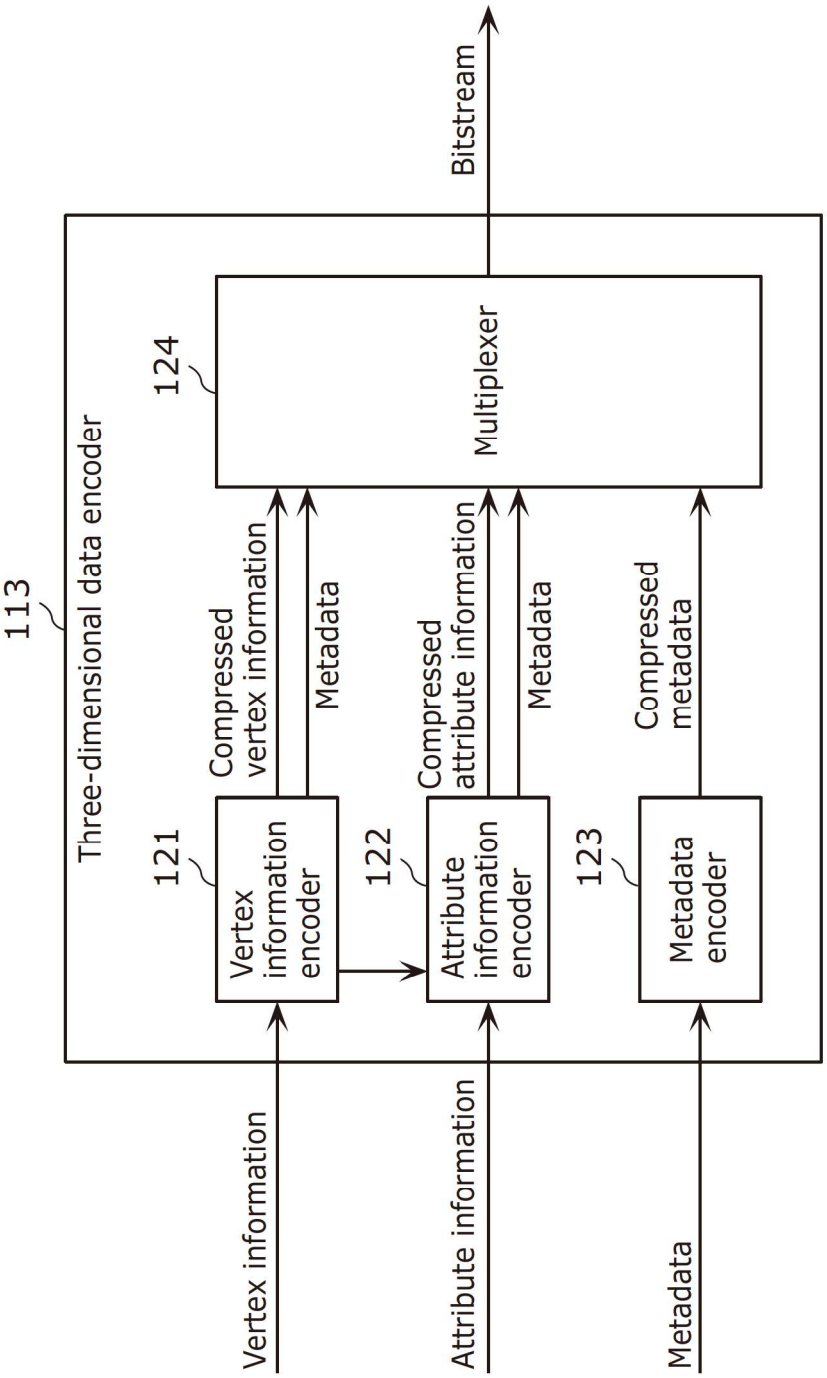

Vertex information
Attribute information
Metadata
Three-dimensional data encoder
113
121
Vertex information encoder
122
Attribute information encoder
123
Metadata encoder
Compressed vertex information
Metadata
Compressed attribute information
Metadata
Compressed metadata
124
Multiplexer
Bitstream

FIG. 19

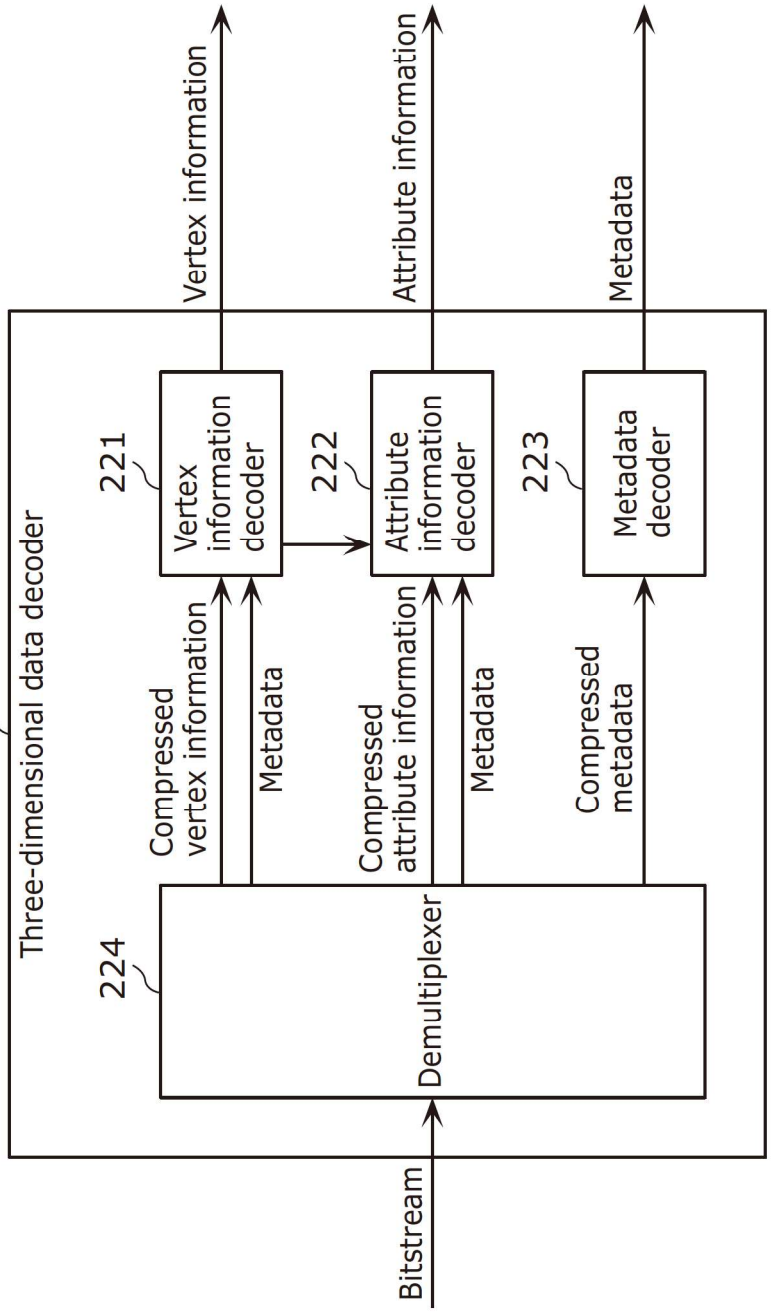

213
Three-dimensional data decoder
221
Vertex information decoder
222
Attribute information decoder
223
Metadata decoder
224
Demultiplexer
Compressed vertex information
Metadata
Compressed attribute information
Metadata
Compressed metadata
Vertex information
Attribute information
Metadata
Bitstream

FIG. 20

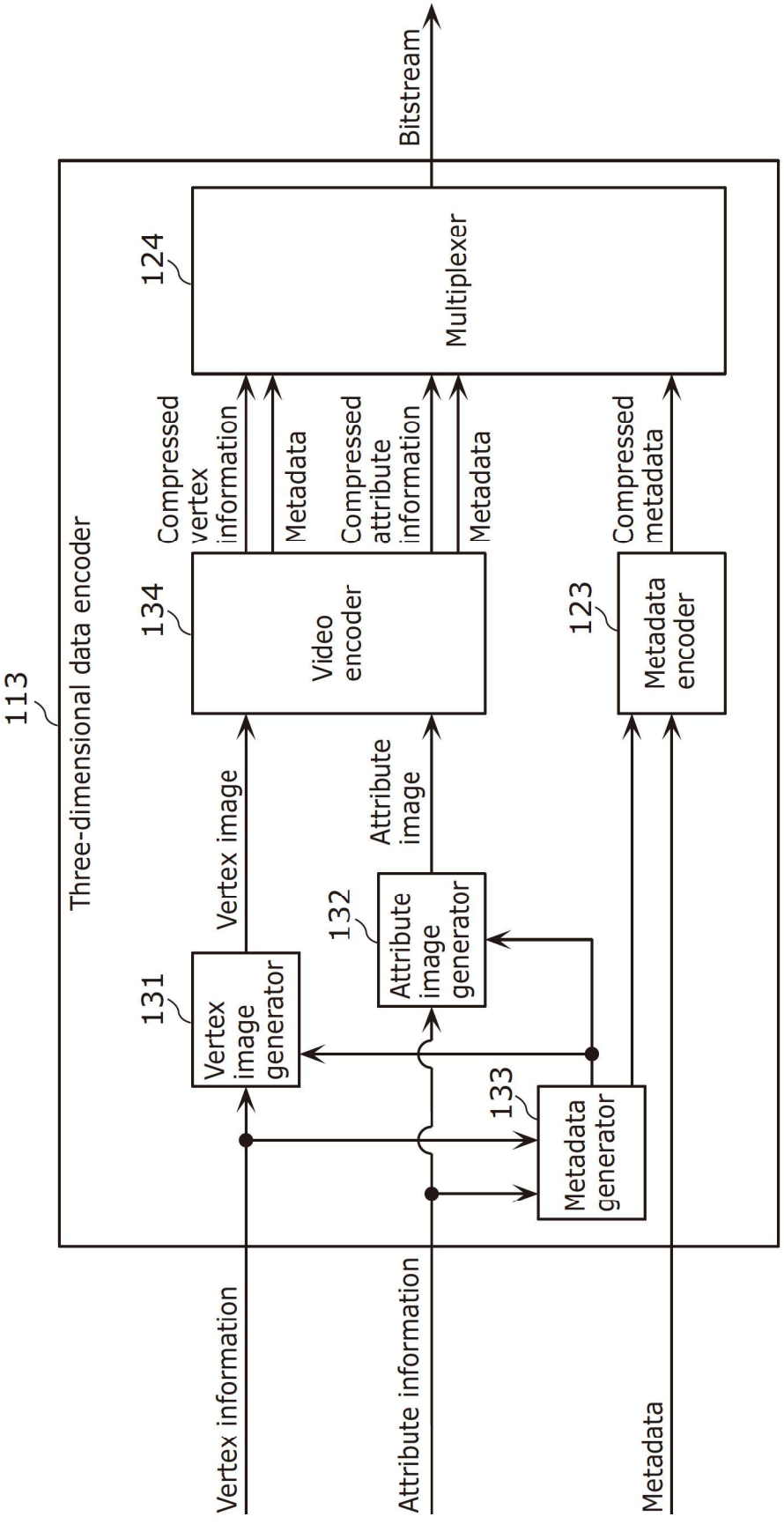

Bitstream
124
Multiplexer
Compressed vertex information
Metadata
Compressed attribute information
Metadata
Compressed metadata
134
Video encoder
123
Metadata encoder
113
Three-dimensional data encoder
Vertex image
Attribute image
132
Attribute image generator
131
Vertex image generator
133
Metadata generator
Vertex information
Attribute information
Metadata

FIG. 21

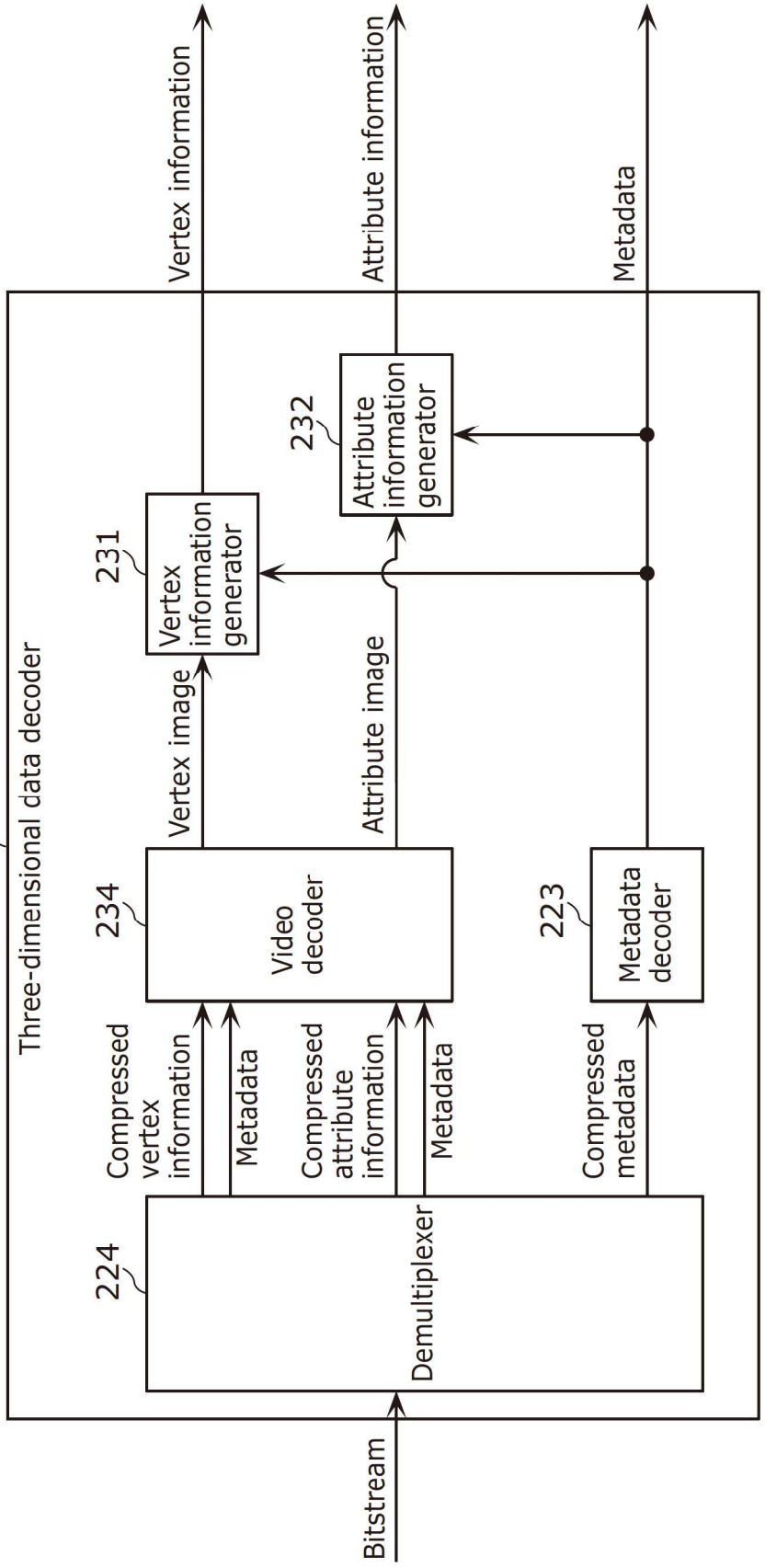

213
Three-dimensional data decoder
Bitstream
224
Demultiplexer
Compressed vertex information
Metadata
Compressed attribute information
Metadata
Compressed metadata
234
Video decoder
Vertex image
Attribute image
231
Vertex information generator
232
Attribute information generator
223
Metadata decoder
Vertex information
Attribute information
Metadata

FIG. 22

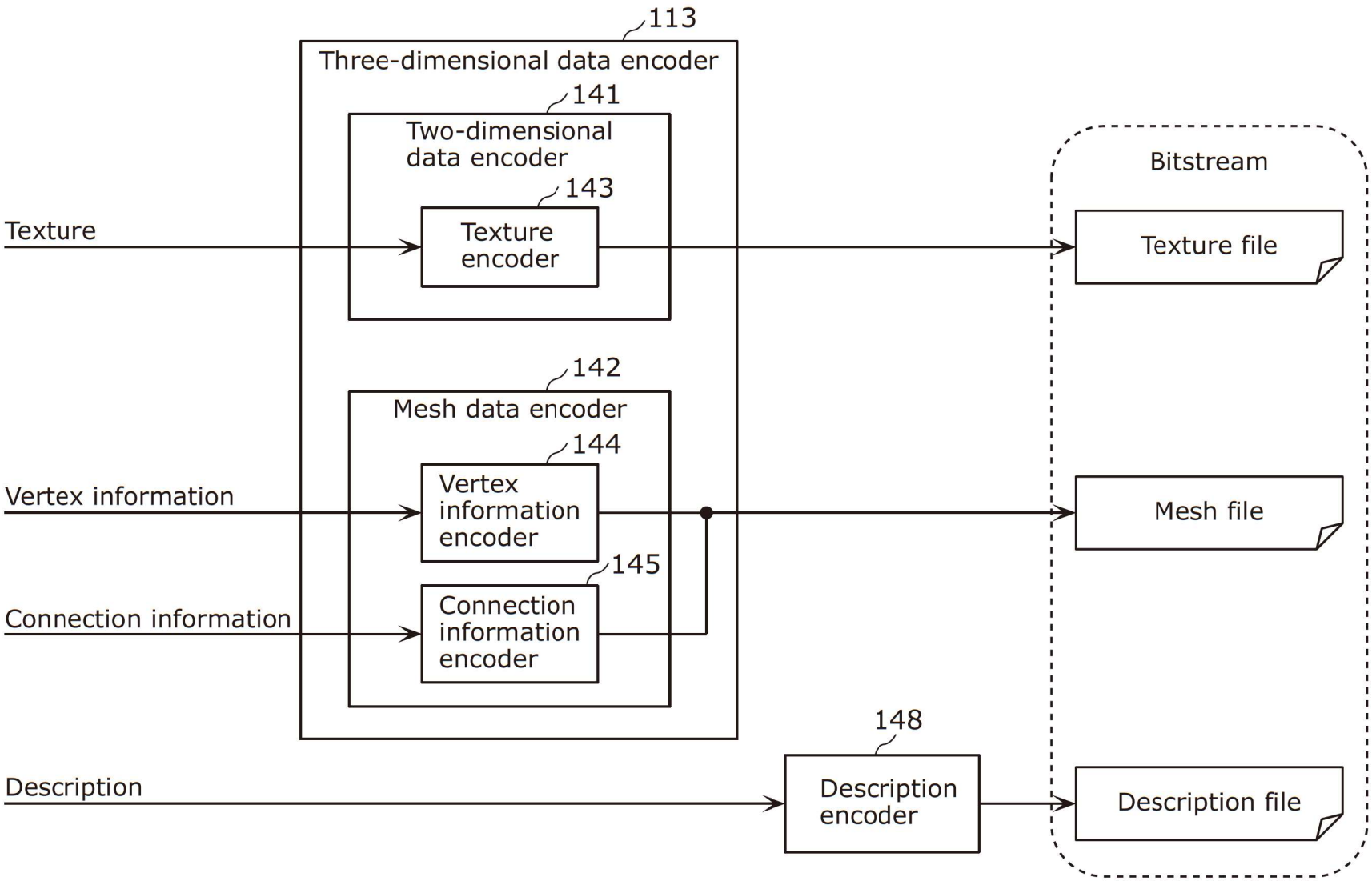

113
Three-dimensional data encoder
141
Two-dimensional data encoder
143
Texture encoder
142
Mesh data encoder
144
Vertex information encoder
145
Connection information encoder
148
Description encoder
Texture
Vertex information
Connection information
Description
Bitstream
Texture file
Mesh file
Description file

FIG. 23

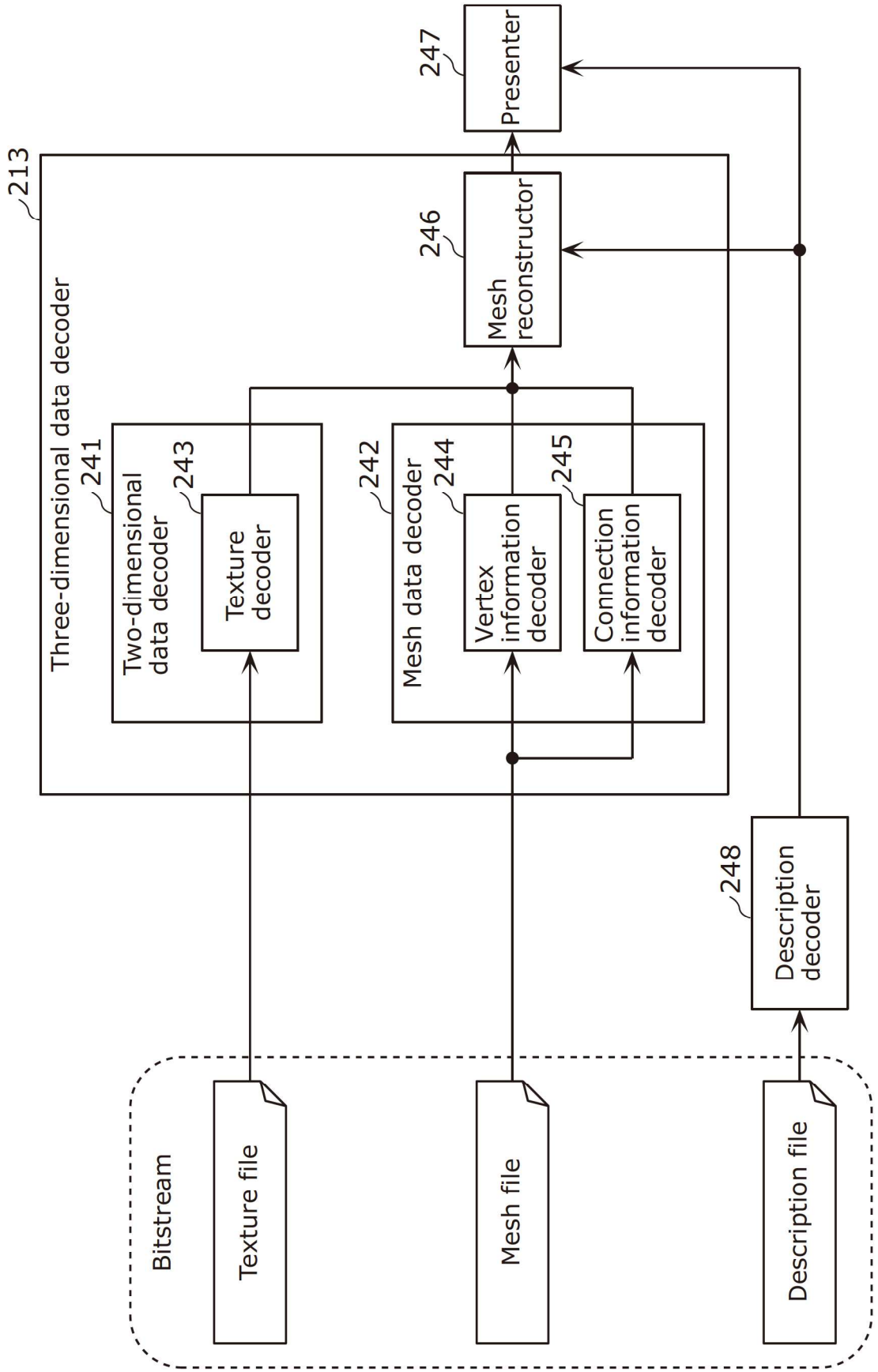

Bitstream
Texture file
Mesh file
Description file
213
Three-dimensional data decoder
241
Two-dimensional data decoder
243
Texture decoder
242
Mesh data decoder
244
Vertex information decoder
245
Connection information decoder
246
Mesh reconstructor
247
Presenter
248
Description decoder

FIG. 24

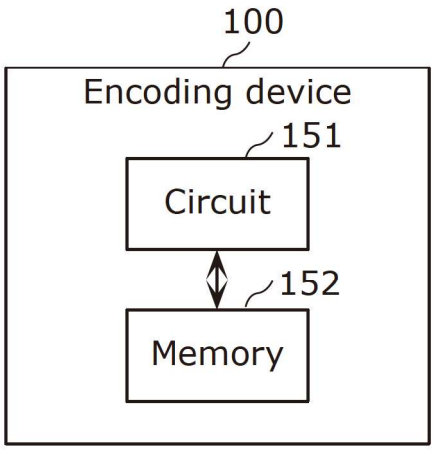

100
Encoding device
151
Circuit
152
Memory

FIG. 25

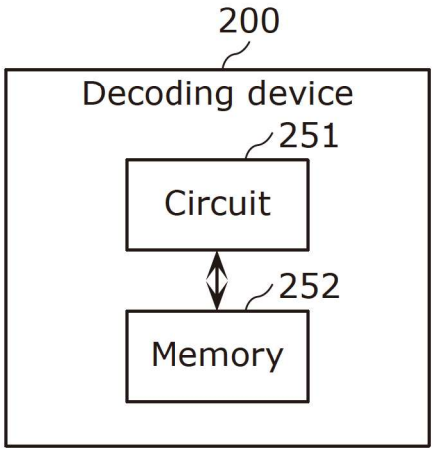

200
Decoding device
251
Circuit
252
Memory

FIG. 26

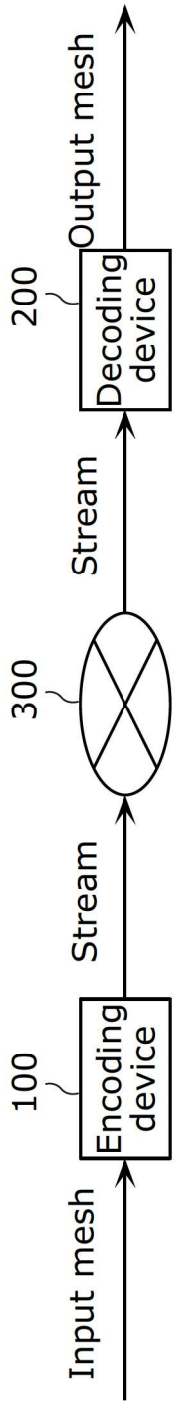

Input mesh
100
Encoding device
Stream
300
Stream
200
Decoding device
Output mesh

FIG. 27

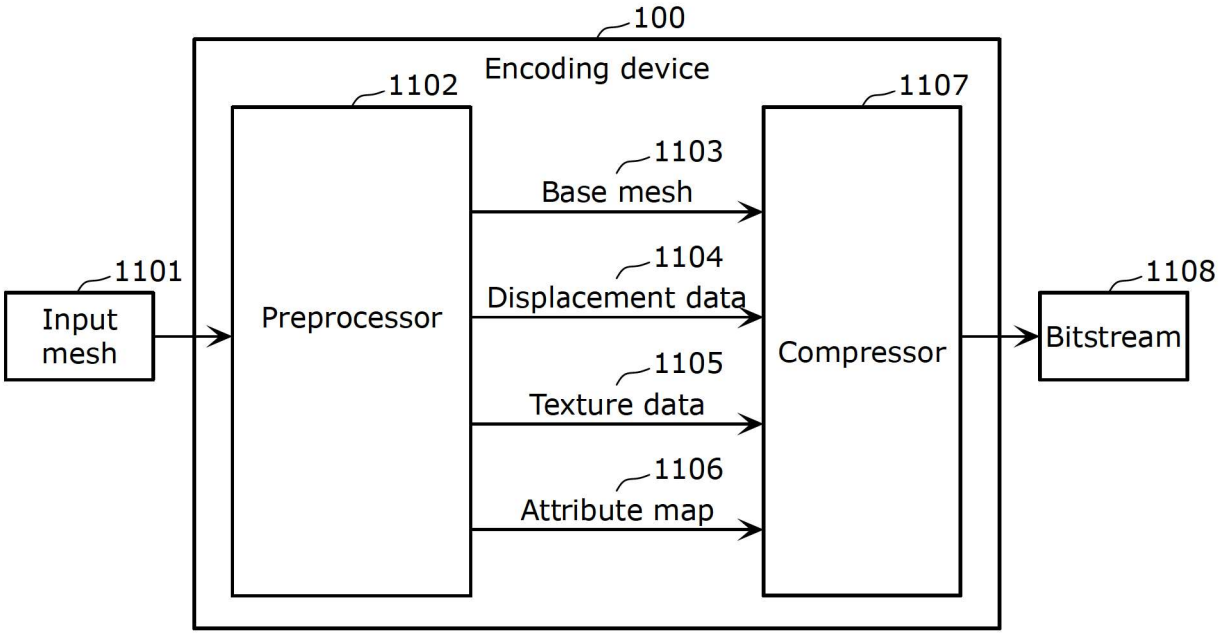

100
Encoding device
1102
Preprocessor
1107
Compressor
1101
Input mesh
1103
Base mesh
1104
Displacement data
1105
Texture data
1106
Attribute map
1108
Bitstream

FIG. 28

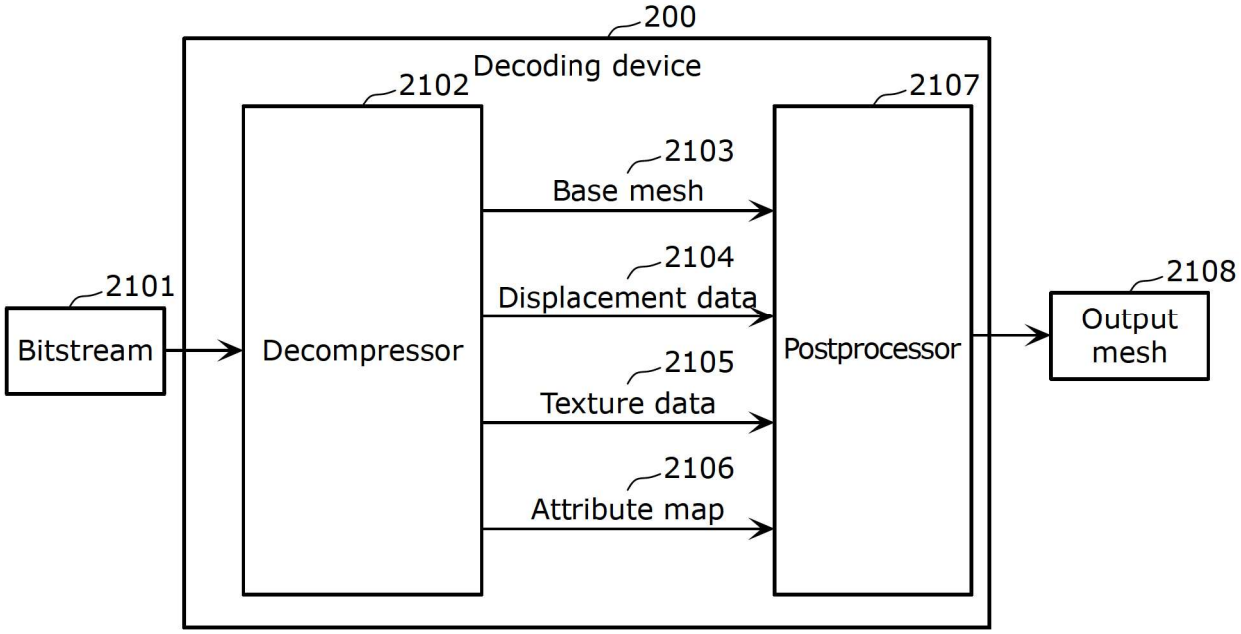

200
Decoding device
2102
Decompressor
2107
Postprocessor
2101
Bitstream
2103
Base mesh
2104
Displacement data
2105
Texture data
2106
Attribute map
2108
Output mesh

FIG. 29

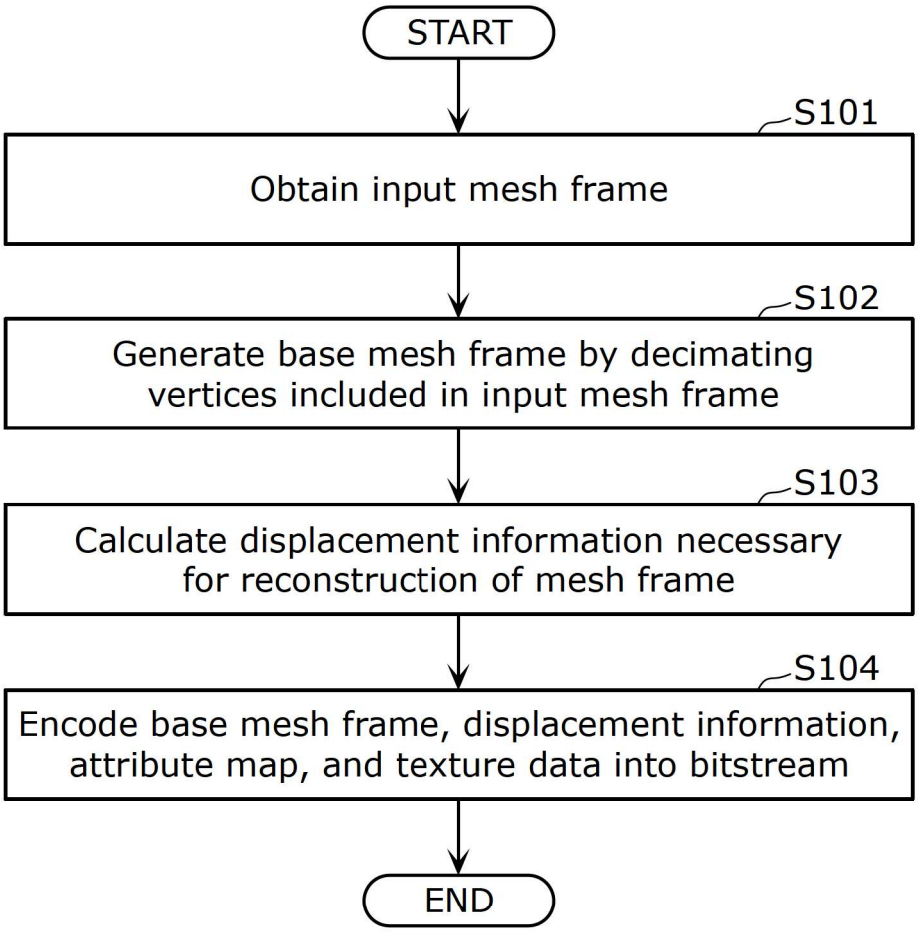

START
S101
Obtain input mesh frame
S102
Generate base mesh frame by decimating vertices included in input mesh frame
S103
Calculate displacement information necessary for reconstruction of mesh frame
S104
Encode base mesh frame, displacement information, attribute map, and texture data into bitstream
END

FIG. 30

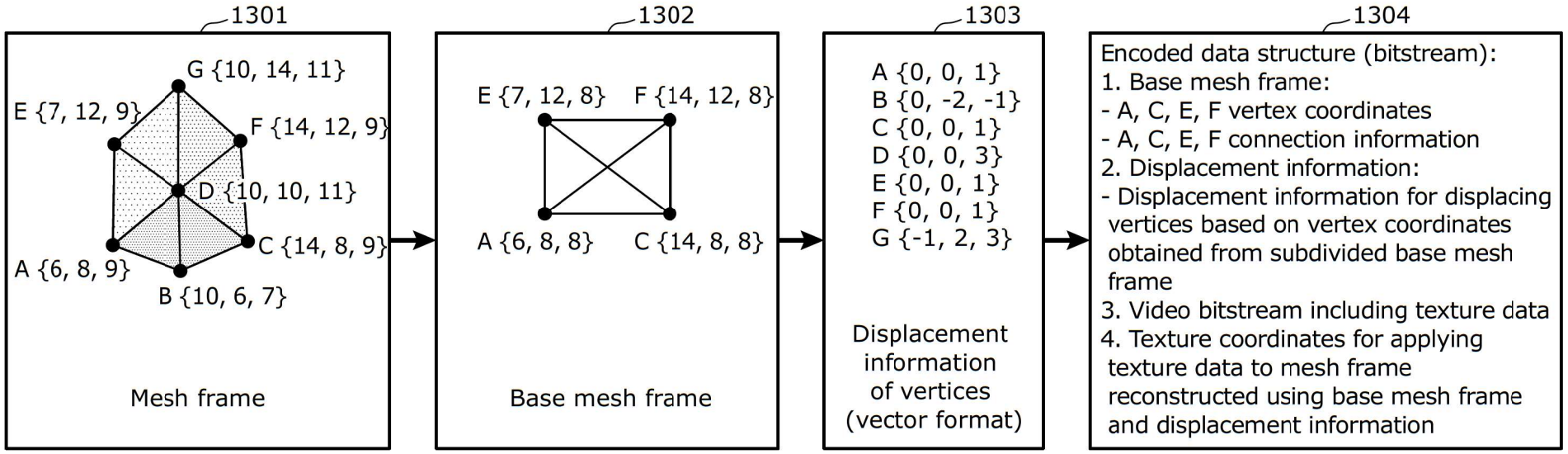

1301
G {10, 14, 11}
E {7, 12, 9}
F {14, 12, 9}
D {10, 10, 11}
C {14, 8, 9}
A {6, 8, 9}
B {10, 6, 7}
Mesh frame
1302
E {7, 12, 8}
F {14, 12, 8}
A {6, 8, 8}
C {14, 8, 8}
Base mesh frame
1303
A {0, 0, 1}
B {0, -2, -1}
C {0, 0, 1}
D {0, 0, 3}
E {0, 0, 1}
F {0, 0, 1}
G {-1, 2, 3}
Displacement information of vertices (vector format)
1304
Encoded data structure (bitstream):
1. Base mesh frame:
- A, C, E, F vertex coordinates
- A, C, E, F connection information
2. Displacement information:
- Displacement information for displacing vertices based on vertex coordinates obtained from subdivided base mesh frame
3. Video bitstream including texture data
4. Texture coordinates for applying texture data to mesh frame reconstructed using base mesh frame and displacement information

FIG. 31

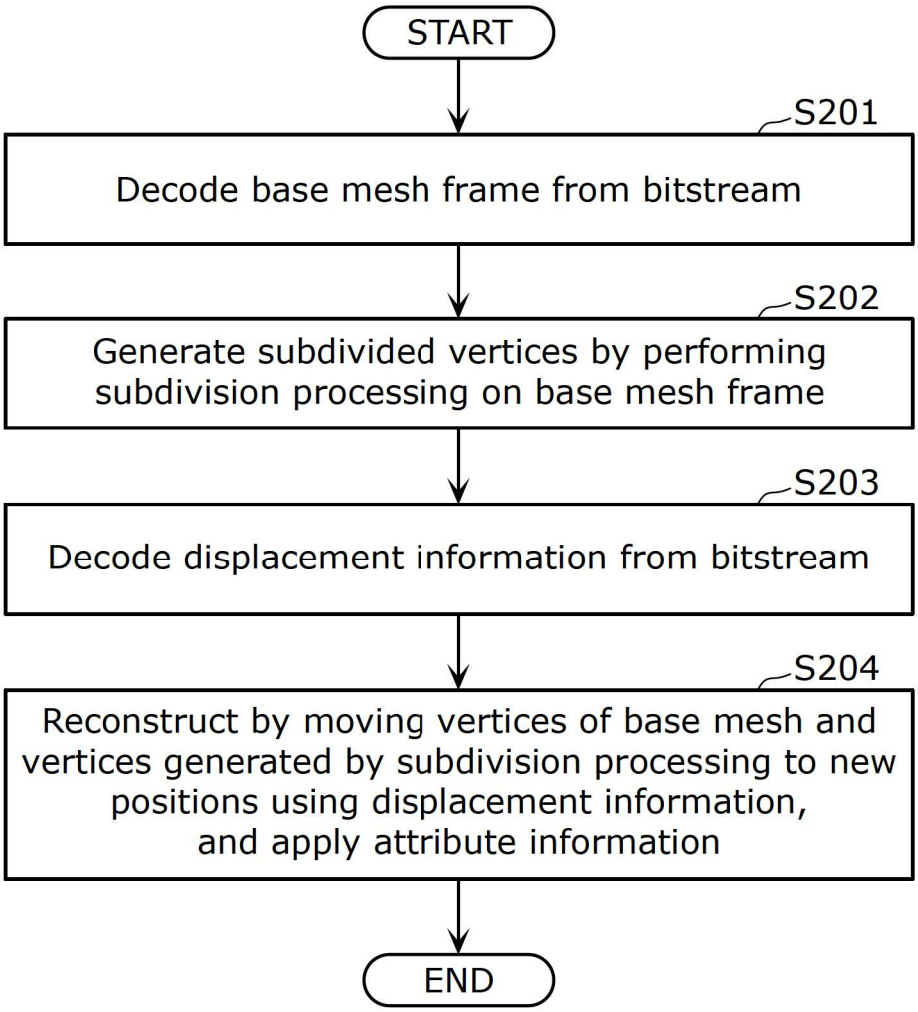

START
S201
Decode base mesh frame from bitstream
S202
Generate subdivided vertices by performing subdivision processing on base mesh frame
S203
Decode displacement information from bitstream
S204
Reconstruct by moving vertices of base mesh and vertices generated by subdivision processing to new positions using displacement information, and apply attribute information
END

FIG. 32

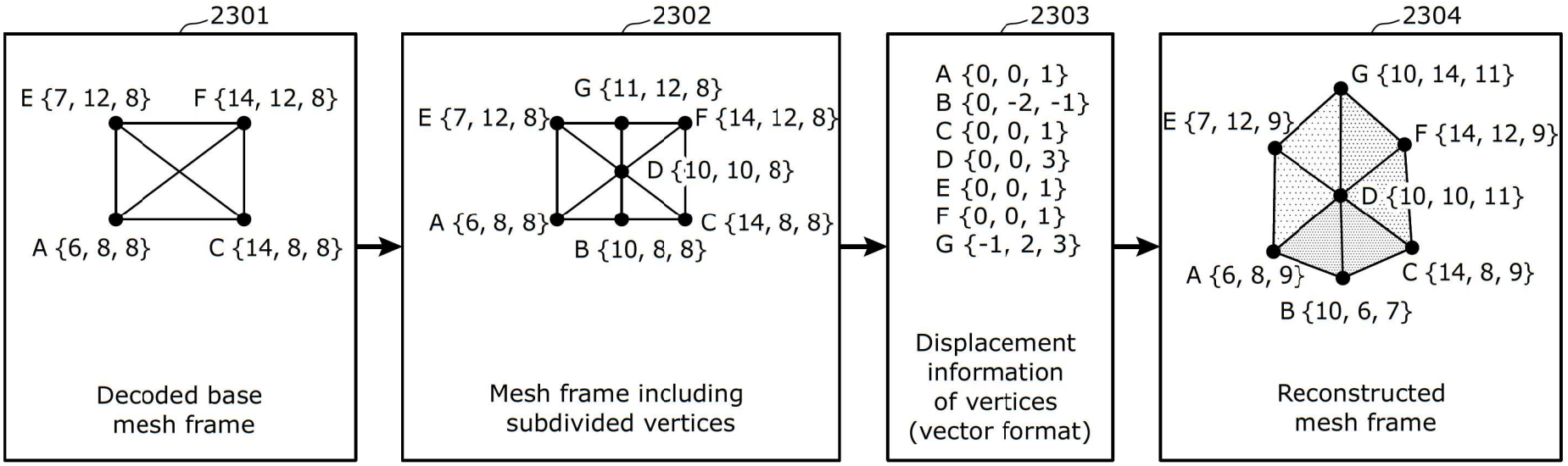

2301
E {7, 12, 8}
F {14, 12, 8}
A {6, 8, 8}
C {14, 8, 8}
Decoded base mesh frame
2302
G {11, 12, 8}
E {7, 12, 8}
F {14, 12, 8}
D {10, 10, 8}
A {6, 8, 8}
C {14, 8, 8}
B {10, 8, 8}
Mesh frame including subdivided vertices
2303
A {0, 0, 1}
B {0, -2, -1}
C {0, 0, 1}
D {0, 0, 3}
E {0, 0, 1}
F {0, 0, 1}
G {-1, 2, 3}
Displacement information of vertices (vector format)
2304
G {10, 14, 11}
E {7, 12, 9}
F {14, 12, 9}
D {10, 10, 11}
A {6, 8, 9}
C {14, 8, 9}
B {10, 6, 7}
Reconstructed mesh frame

FIG. 33

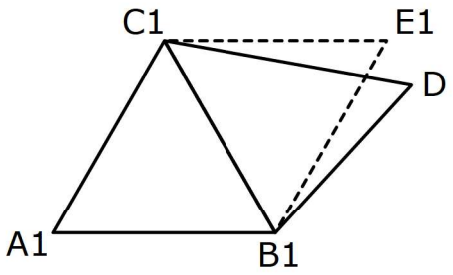


FIG. 34

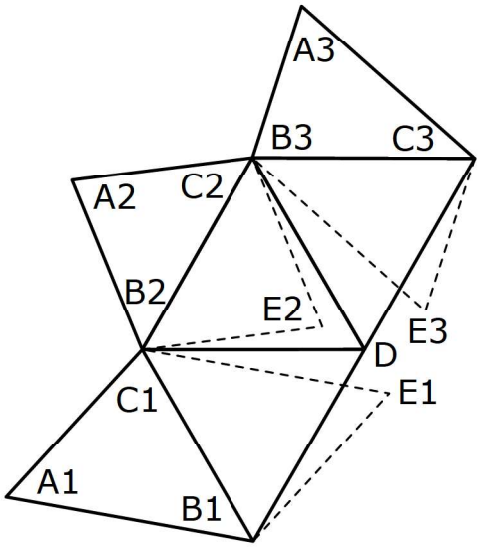


FIG. 35

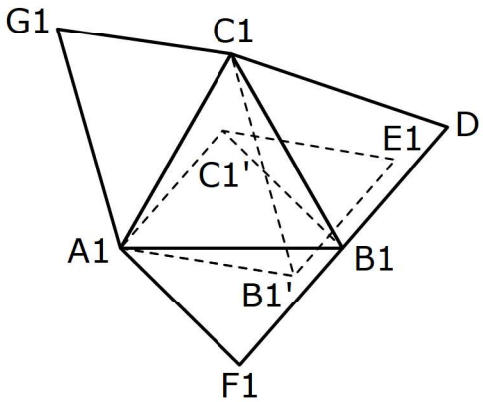


FIG. 36

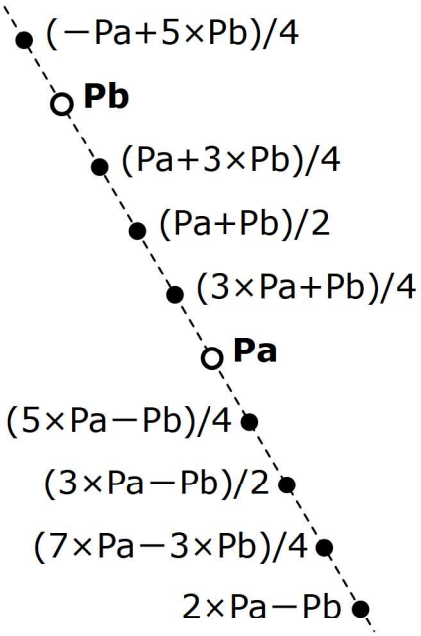

(−Pa+5×Pb)/4
Pb
(Pa+3×Pb)/4
(Pa+Pb)/2
(3×Pa+Pb)/4
Pa
(5×Pa−Pb)/4
(3×Pa−Pb)/2
(7×Pa−3×Pb)/4
2×Pa−Pb

FIG. 37

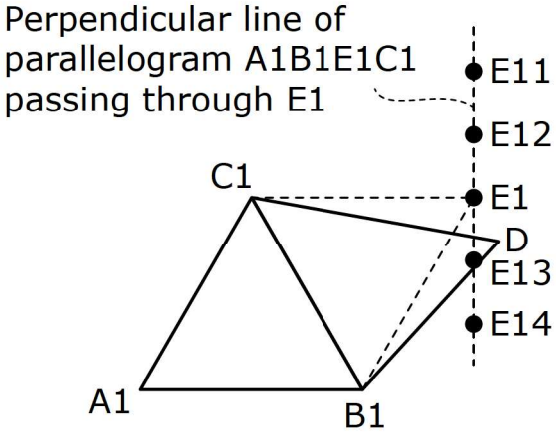

Perpendicular line of
parallelogram A1B1E1C1
passing through E1
E11
E12
C1
E1
D
E13
E14
A1
B1

FIG. 38

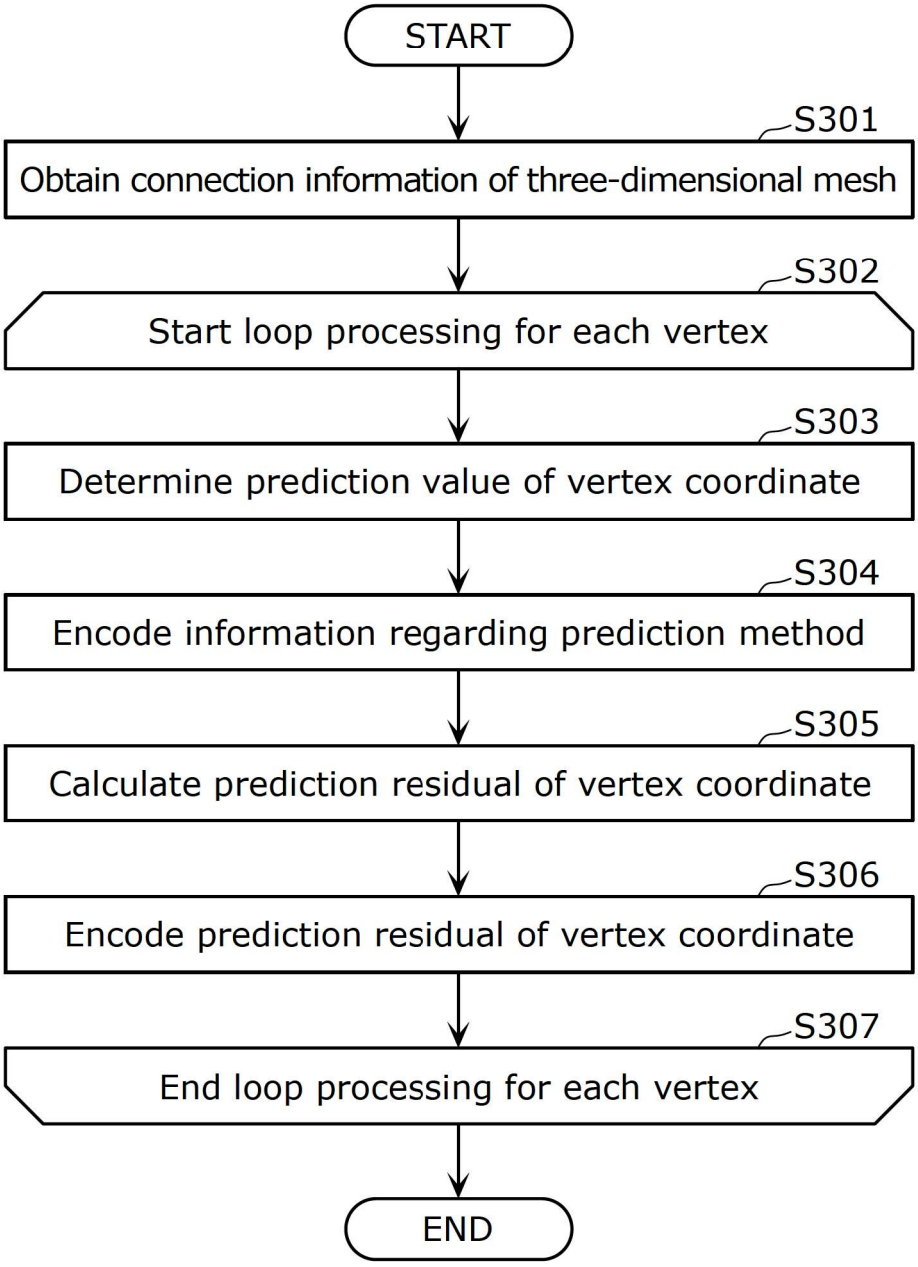

START
S301
Obtain connection information of three-dimensional mesh
S302
Start loop processing for each vertex
S303
Determine prediction value of vertex coordinate
S304
Encode information regarding prediction method
S305
Calculate prediction residual of vertex coordinate
S306
Encode prediction residual of vertex coordinate
S307
End loop processing for each vertex
END

FIG. 39

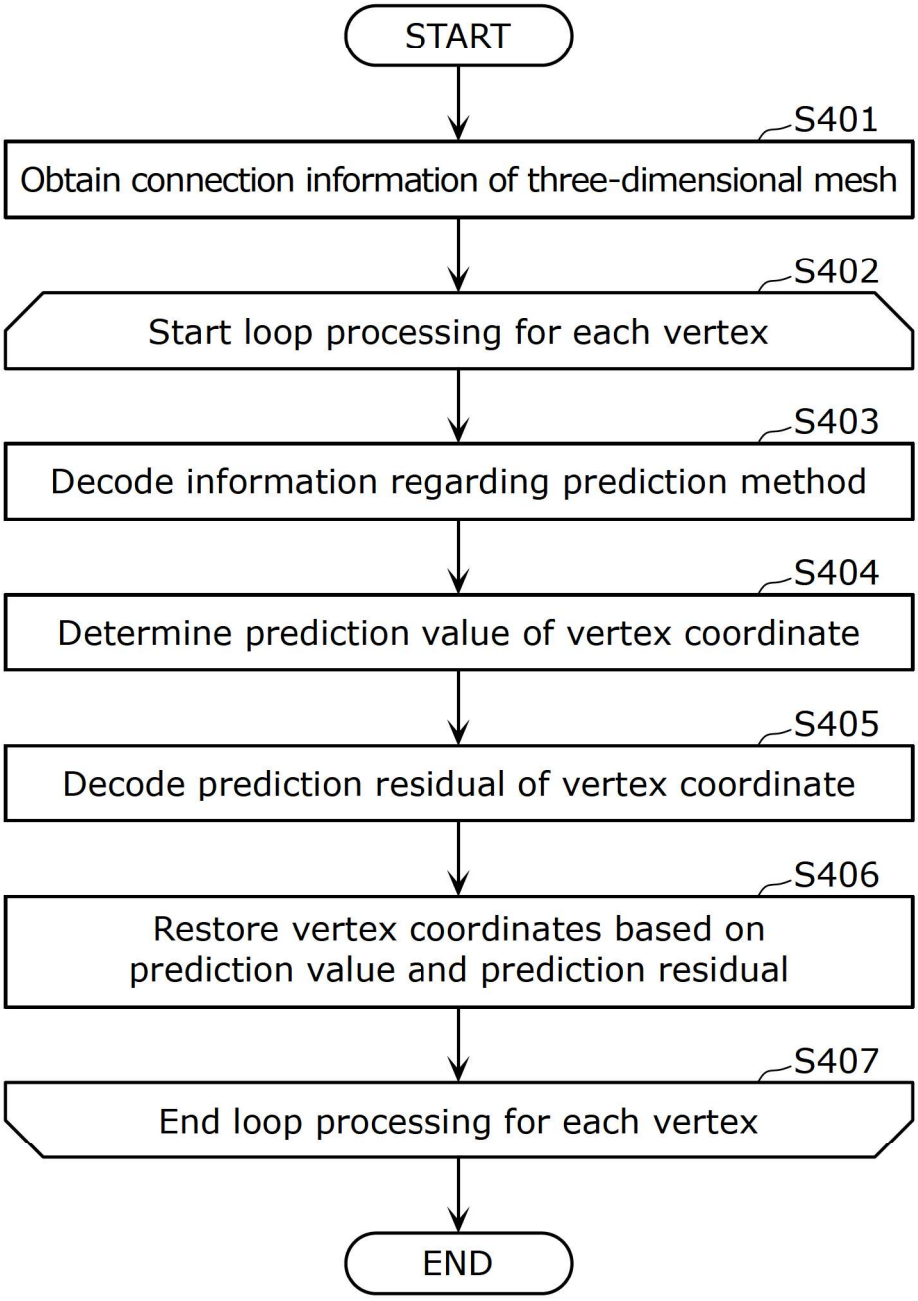

START
S401
Obtain connection information of three-dimensional mesh
S402
Start loop processing for each vertex
S403
Decode information regarding prediction method
S404
Determine prediction value of vertex coordinate
S405
Decode prediction residual of vertex coordinate
S406
Restore vertex coordinates based on prediction value and prediction residual
S407
End loop processing for each vertex
END

FIG. 40

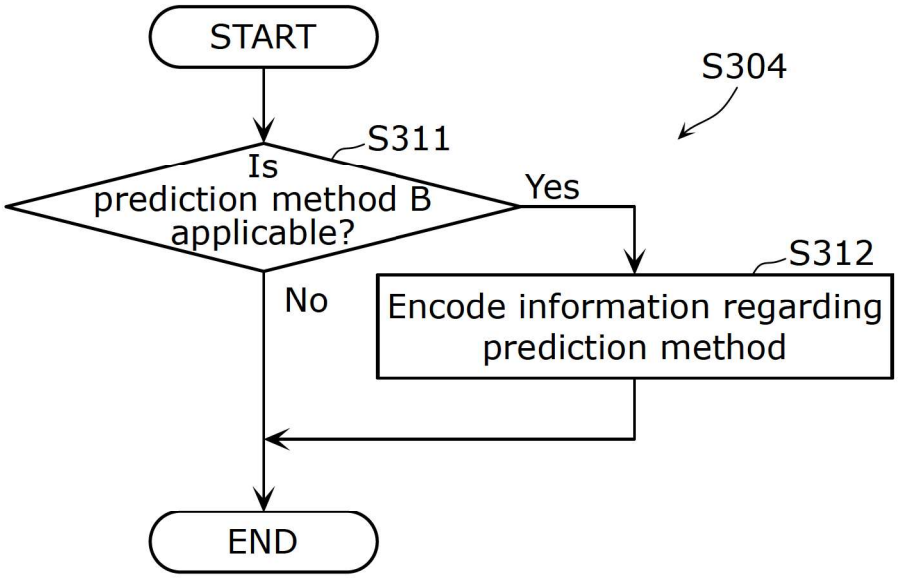

START
S304
S311
Is prediction method B applicable?
Yes
S312
Encode information regarding prediction method
No
END

FIG. 41

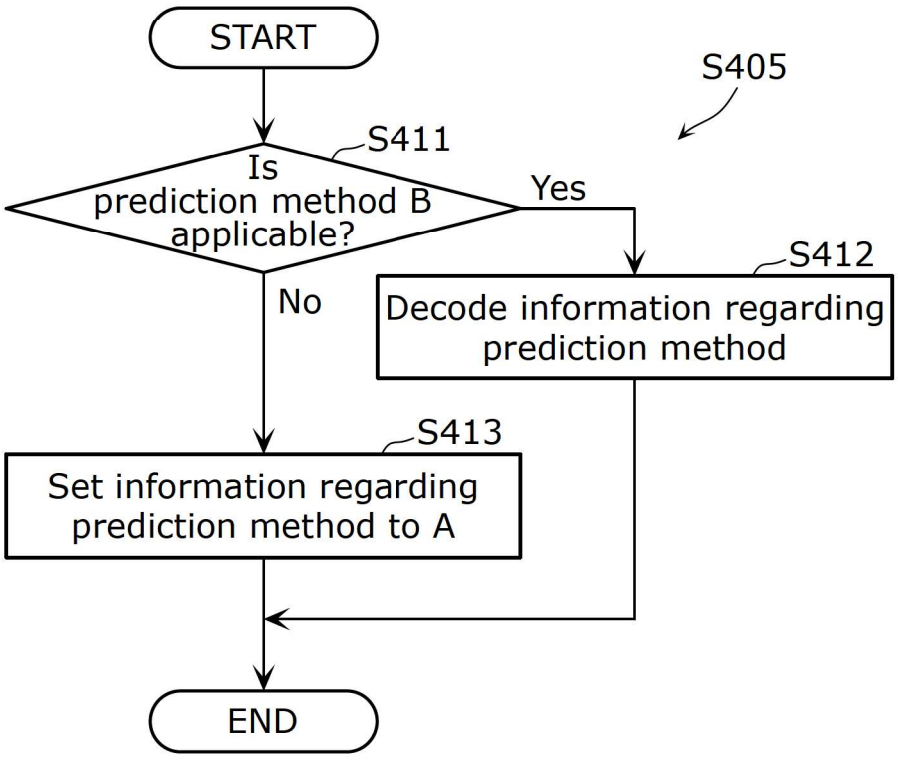

START
S405
S411
Is prediction method B applicable?
Yes
S412
Decode information regarding prediction method
No
S413
Set information regarding prediction method to A
END

FIG. 42

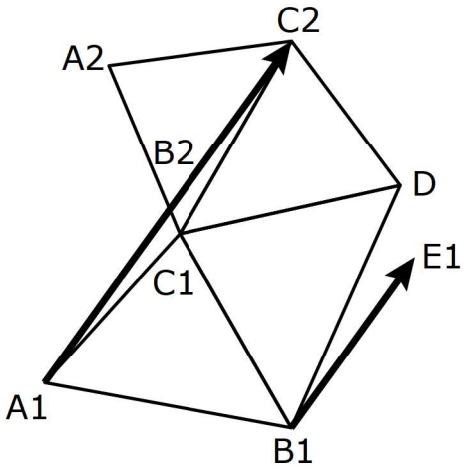


FIG. 43

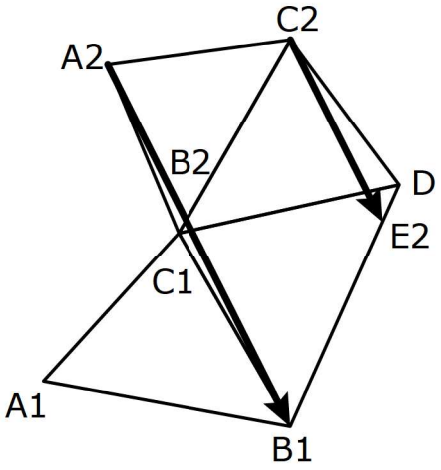


FIG. 44

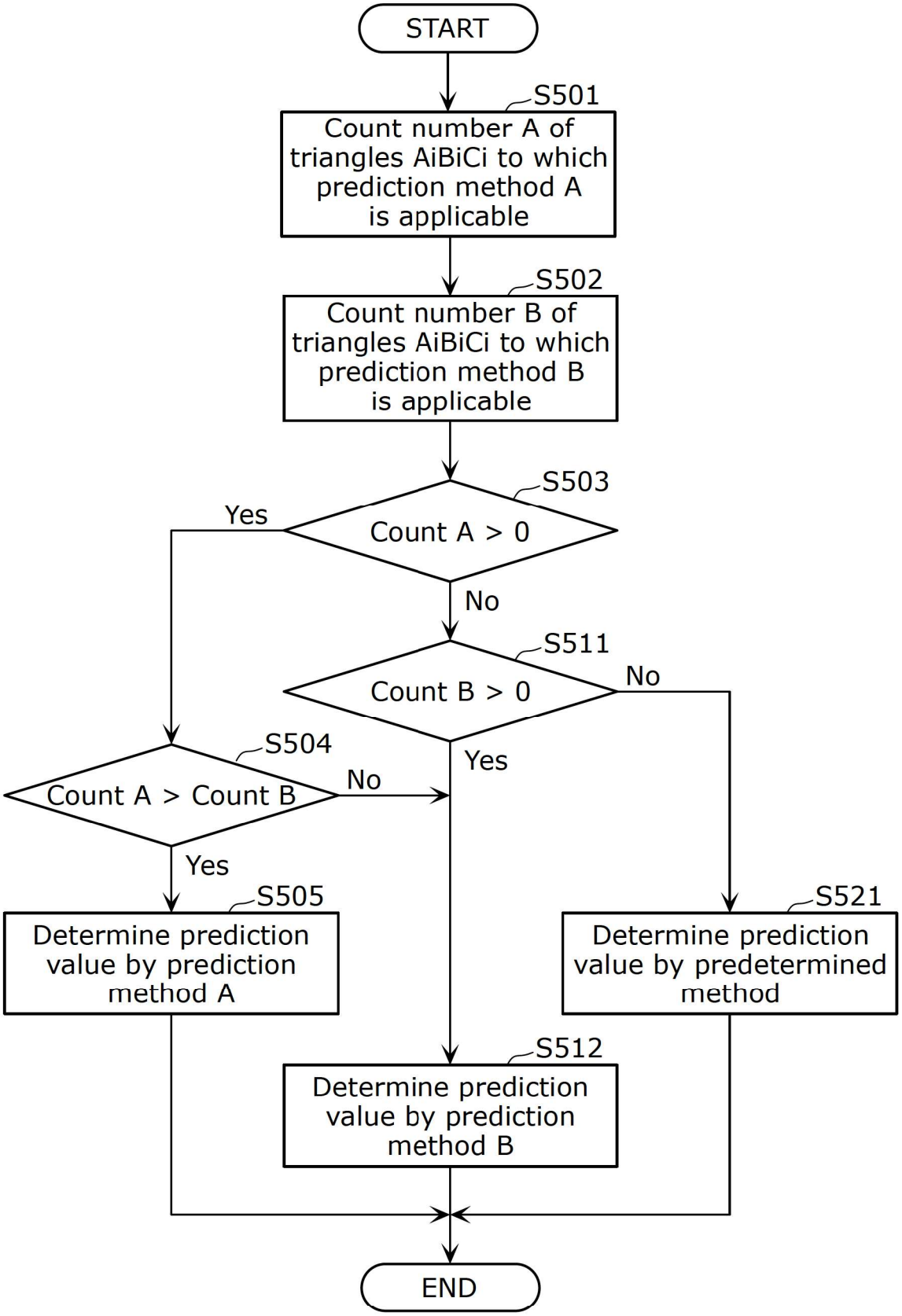

START
S501
Count number A of triangles AiBiCi to which prediction method A is applicable
S502
Count number B of triangles AiBiCi to which prediction method B is applicable
S503
Count A > 0
Yes
No
S511
Count B > 0
No
Yes
S504
Count A > Count B
No
Yes
S505
Determine prediction value by prediction method A
S521
Determine prediction value by predetermined method
S512
Determine prediction value by prediction method B
END

FIG. 45

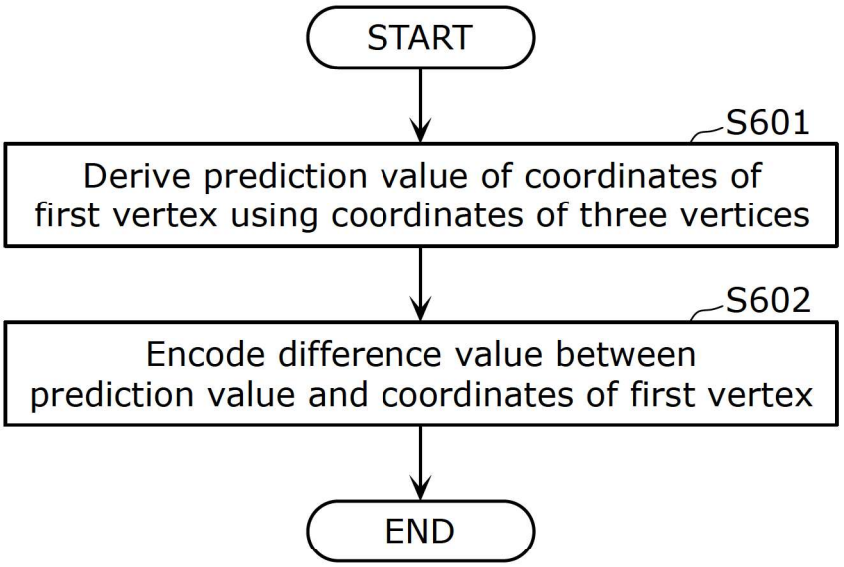
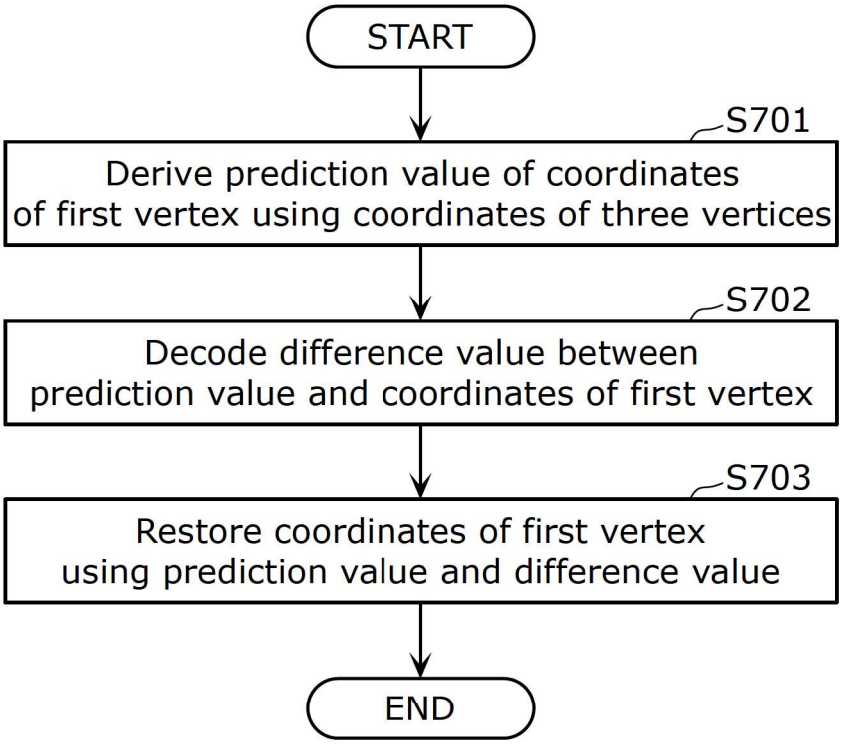

START
S601
Derive prediction value of coordinates of first vertex using coordinates of three vertices
S602
Encode difference value between prediction value and coordinates of first vertex
END

FIG. 46


START
S701
Derive prediction value of coordinates of first vertex using coordinates of three vertices
S702
Decode difference value between prediction value and coordinates of first vertex
S703
Restore coordinates of first vertex using prediction value and difference value
END