

Fig. 1

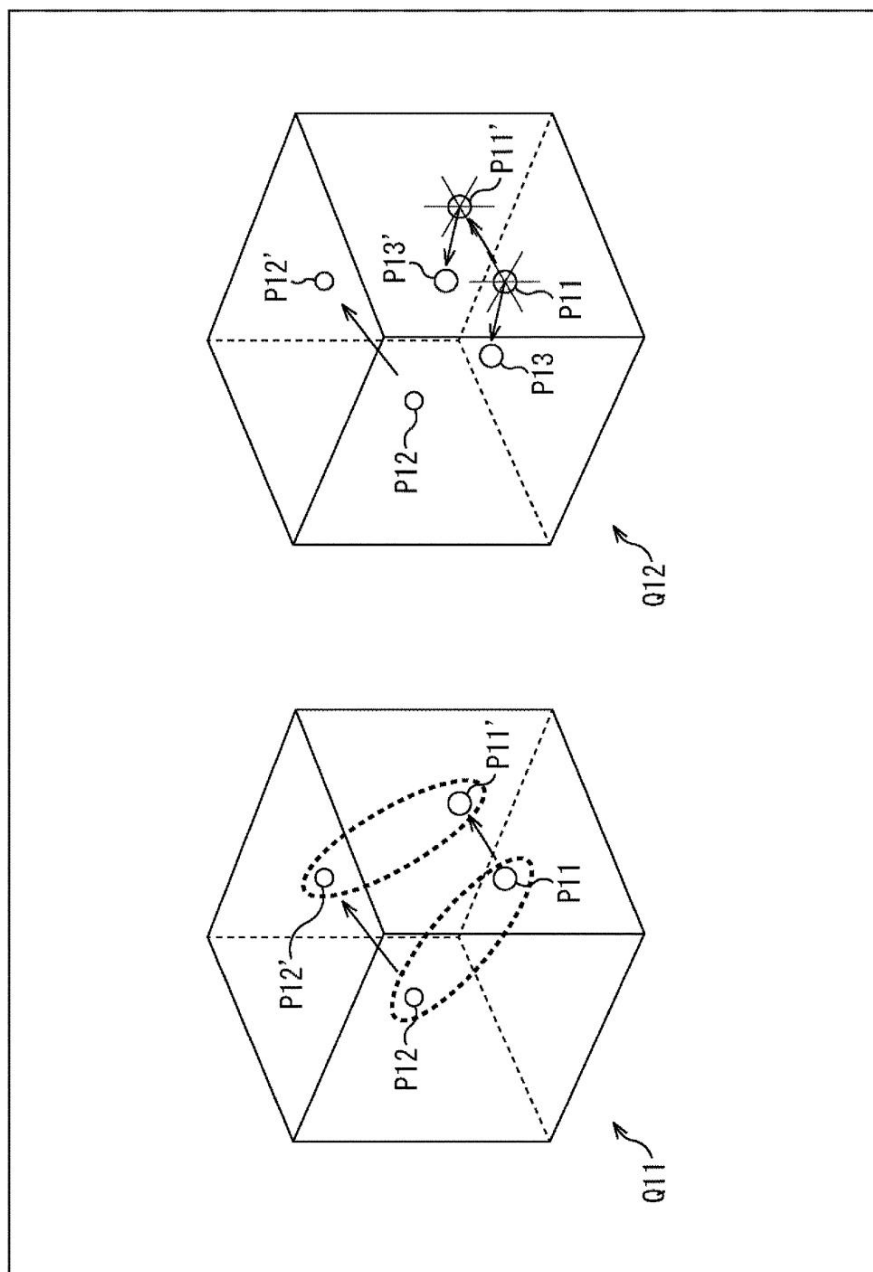


Fig. 2

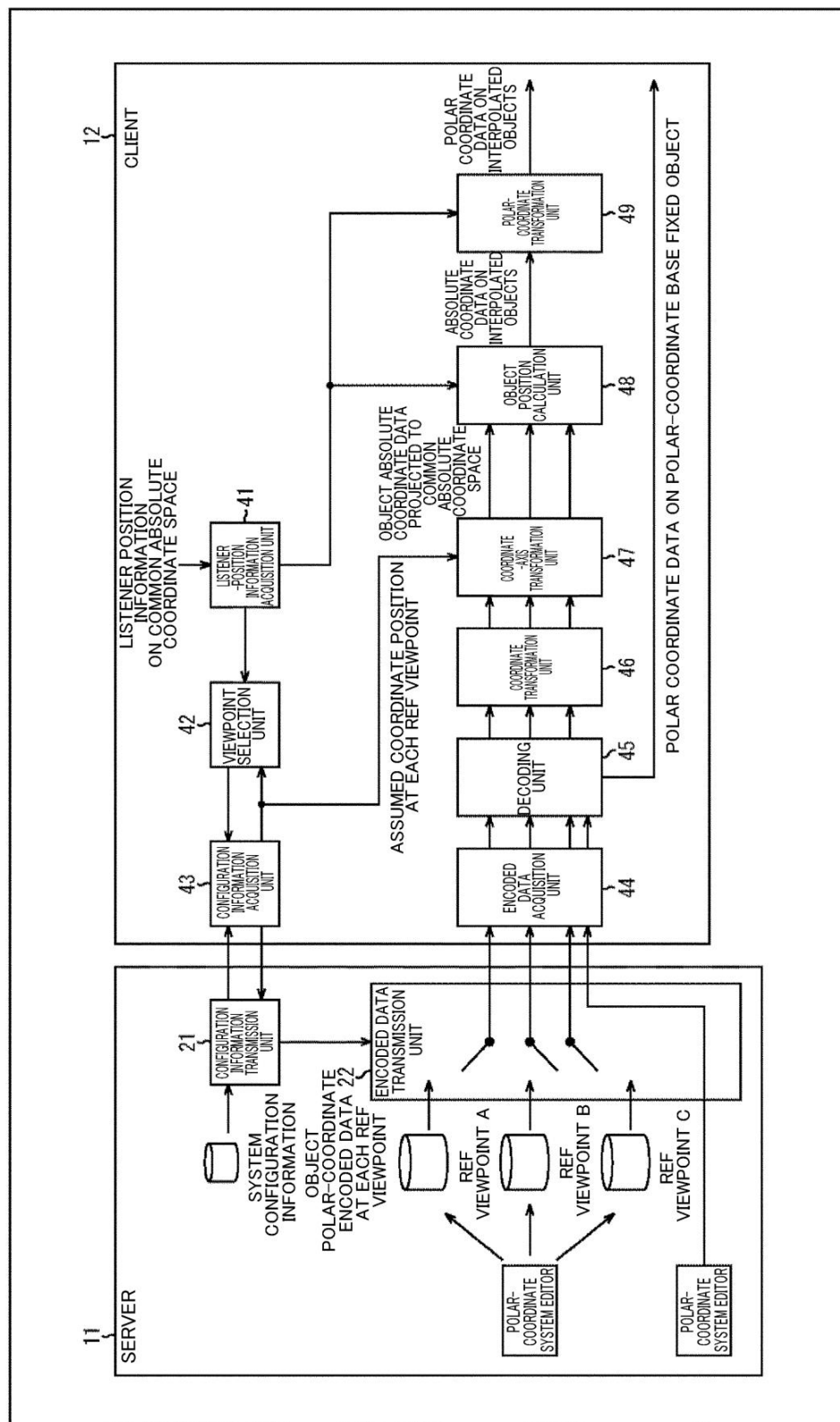


Fig. 3

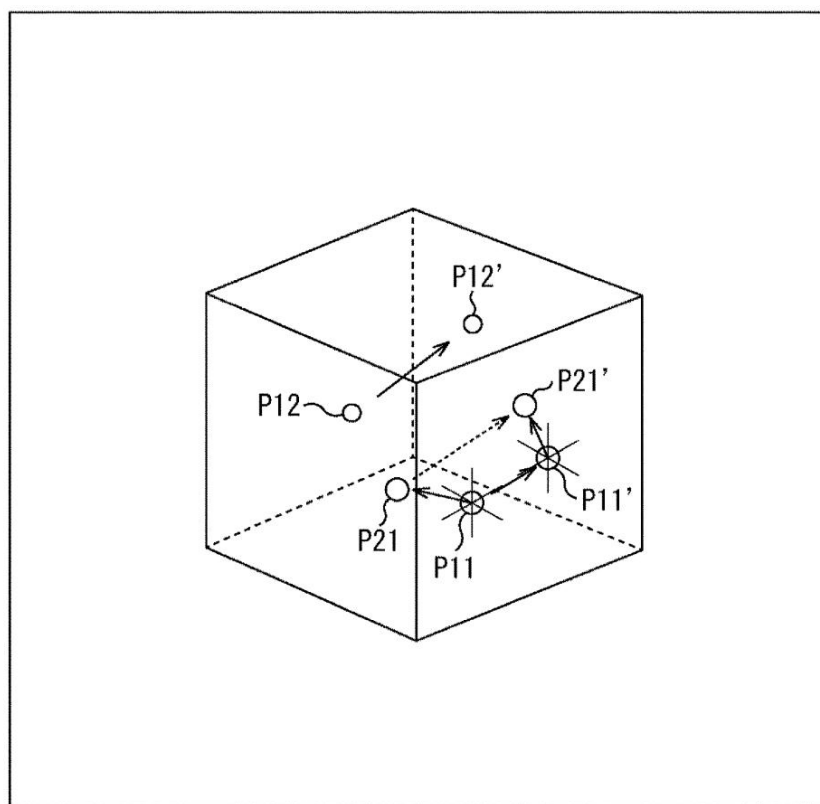


Fig. 4

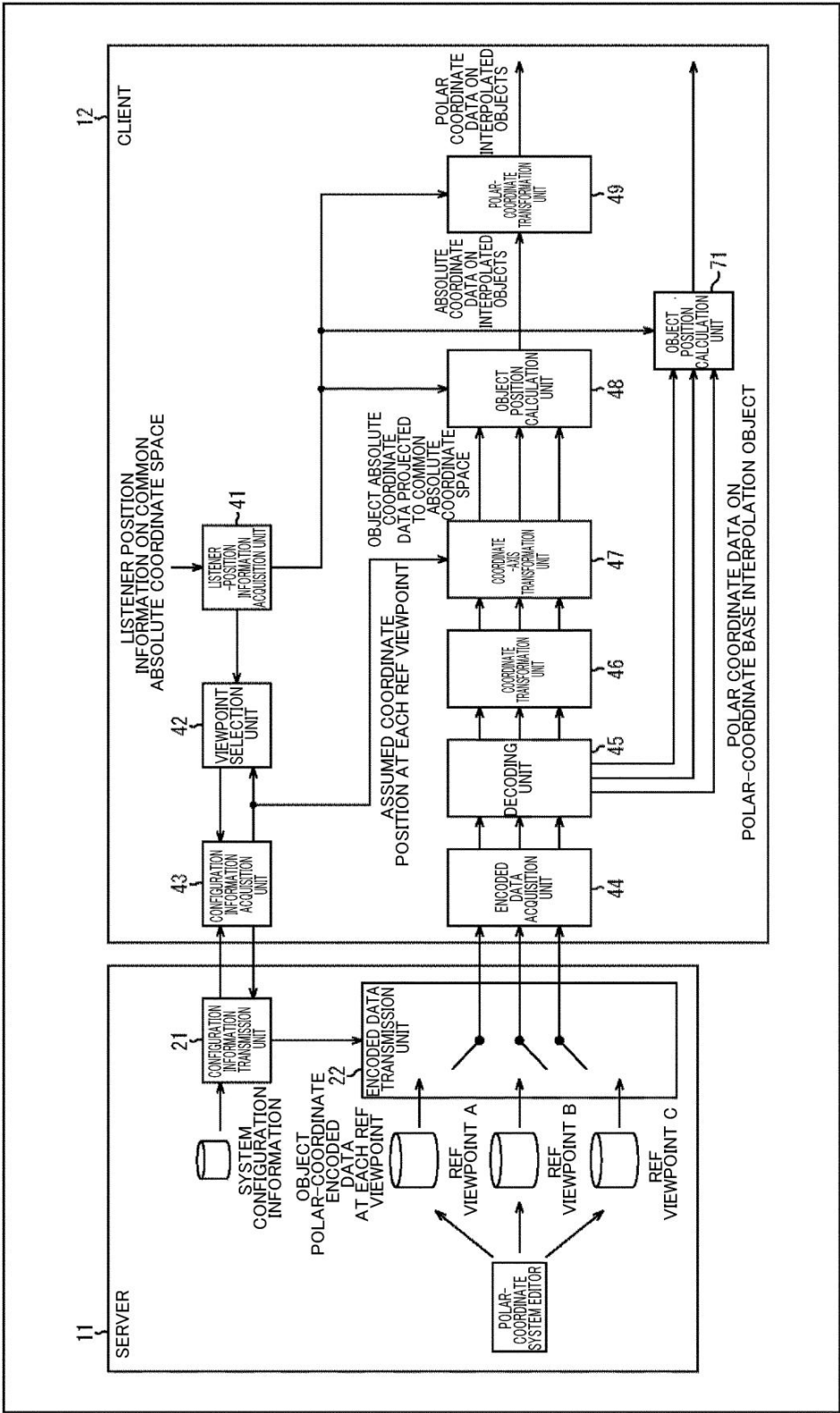


Fig. 5

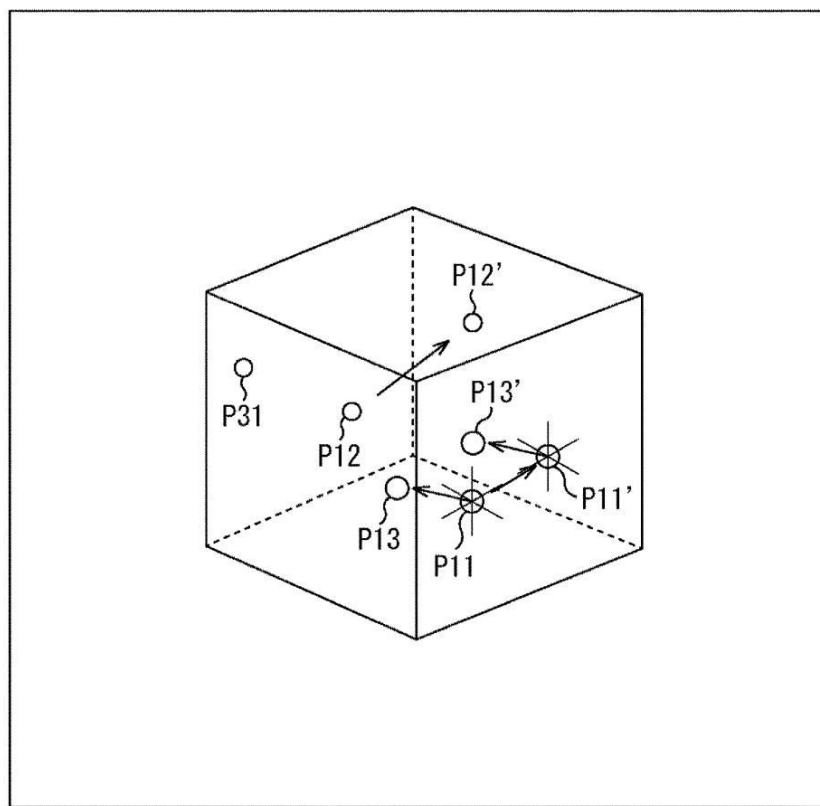


Fig. 6

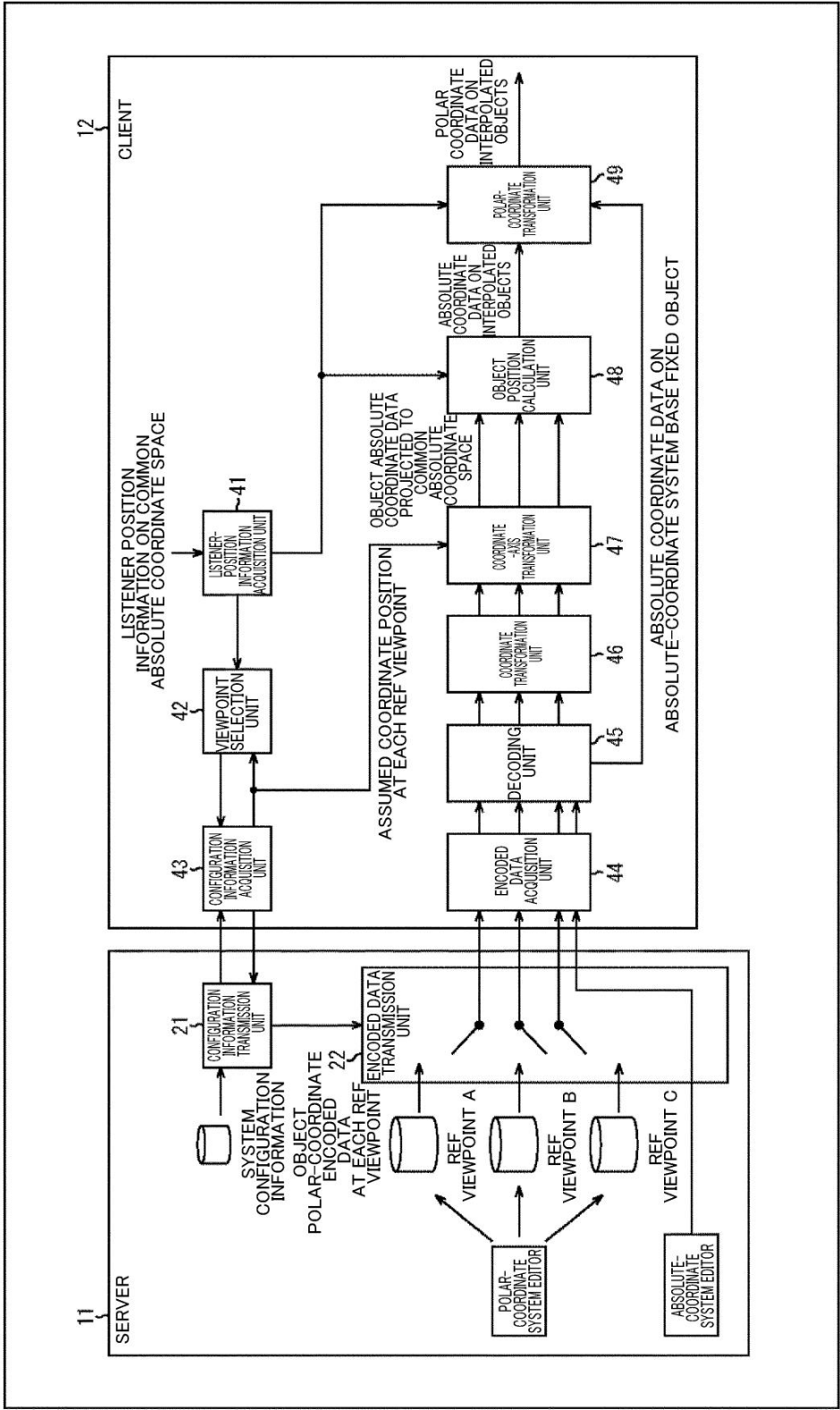


Fig. 7

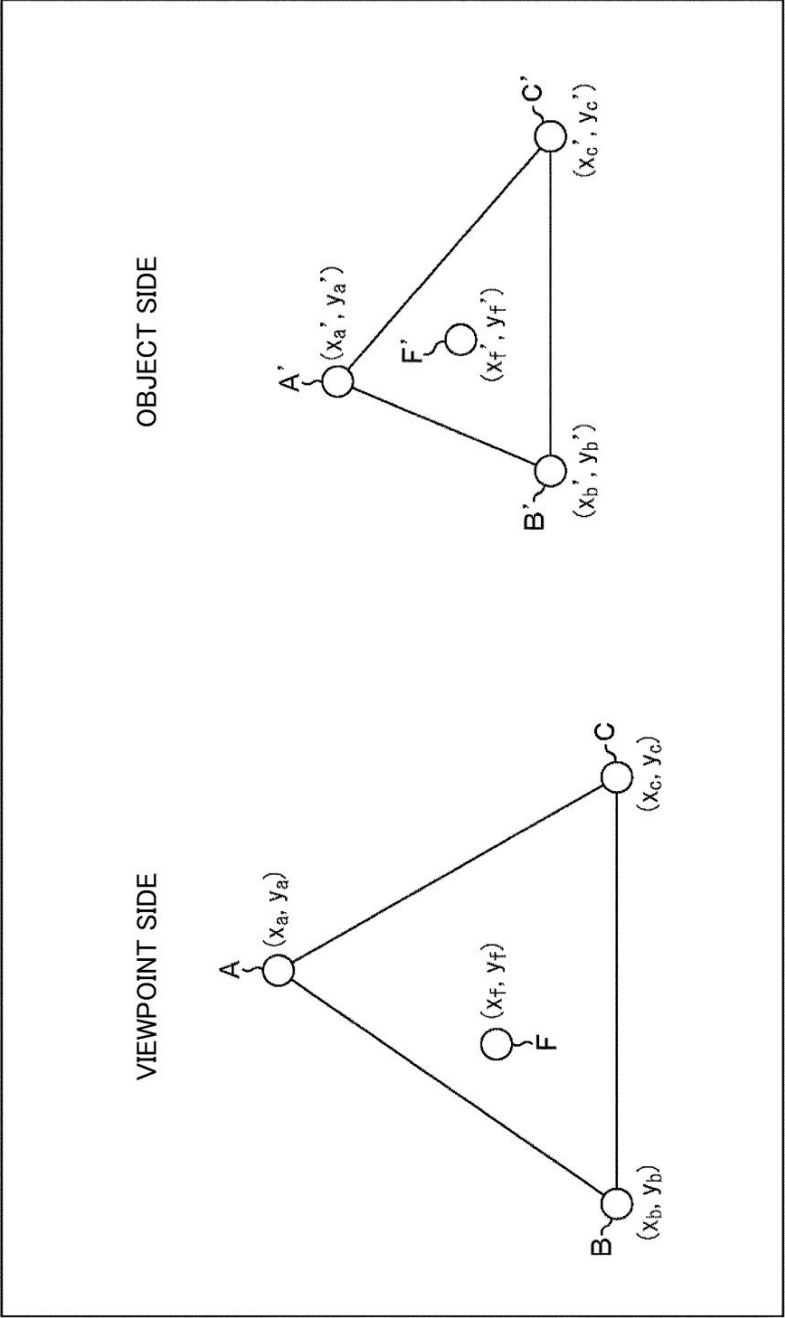


Fig. 8

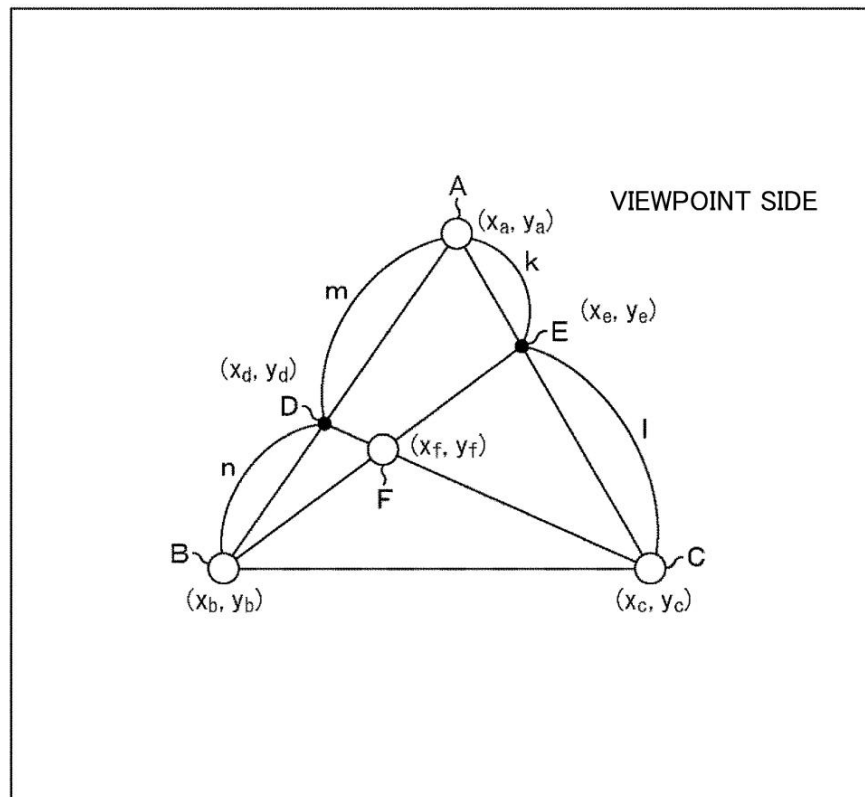


Fig. 9

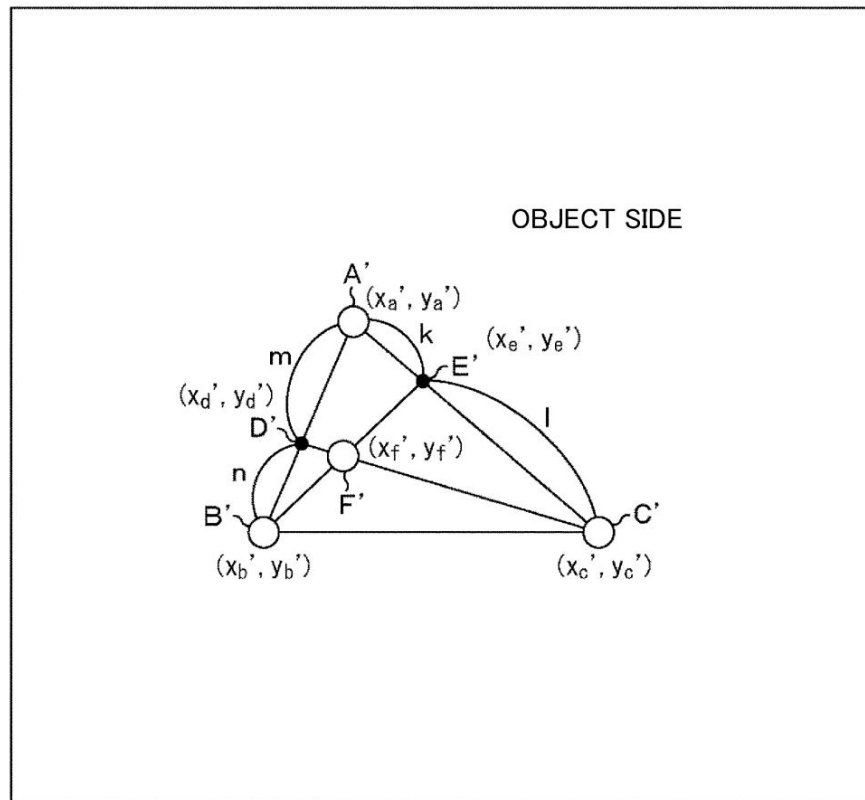


Fig. 10

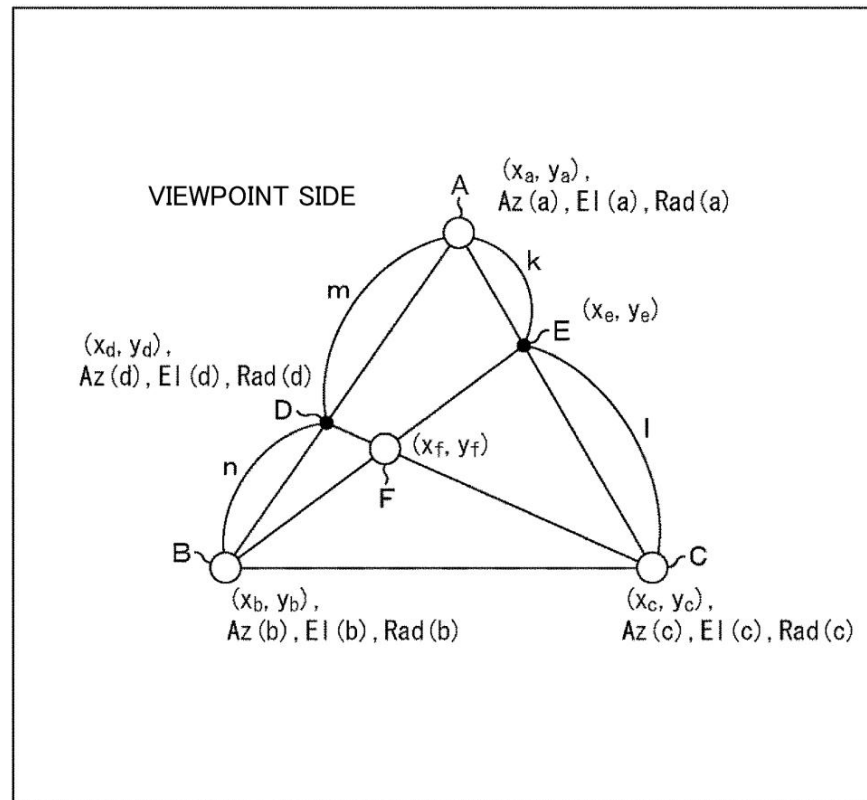


Fig. 11

syntax	bit	Comment
<code>fv_astructure_info_polar() {</code>		
<code> NumOfObj;</code>	INT32	THE NUMBER OF OBJECTS
<code> NumOfRefViewPoint;</code>	INT32	THE NUMBER OF REFERENCE VIEWPOINTS
<code> for (i = 0; i < NumOfRefViewPoint; i++) {</code>		
<code> RefViewZ[i];</code>	FLOAT32	ASSUMED ABSOLUTE COORDINATE POSITION (X AXIS) AT REFERENCE VIEWPOINT FLOAT32 VALUE
<code> RefViewY[i];</code>	FLOAT32	ASSUMED ABSOLUTE COORDINATE POSITION (Y AXIS) AT REFERENCE VIEWPOINT FLOAT32 VALUE
<code> RefViewX[i];</code>	FLOAT32	ASSUMED ABSOLUTE COORDINATE POSITION (Z AXIS) AT REFERENCE VIEWPOINT FLOAT32 VALUE
<code> RefYaw[i];</code>	FLOAT32	ASSUMED FACE ORIENTATION OF LISTENER AT EACH REFERENCE VIEWPOINT POSITION (YAW DIRECTION)
<code> }</code>		
<code> for (i = 0; i < NumOfObj; i++) {</code>		
<code> ObjectOverlapMode[i];</code>	INT32	REPRODUCTION MODE IN WHICH LISTENER AND OBJECT OVERLAP EACH OTHER
<code> if (ObjectOverlapMode[i] == 2) {</code>		
<code> ProhibitRadius</code>	FLOAT32	NORMALIZED VALUE OF DISTANCE FROM OBJECT TO LISTENER IF SPACE IS NORMALIZED BY 1.0
<code> }</code>		
<code> }</code>		
<code> InterpolateMode</code>	INT32 (CHAR)	INTERPOLATION MODE PERMITTED FOR CLIENT (8-BIT BIT MASK)
<code> NonInterpolatePolarObjFlag</code>	INT32	FLAG INDICATING PRESENCE OR ABSENCE OF POLAR-COORDINATE BASE FIXED OBJECT 0: ABSENT, 1: PRESENT
<code> if (NonInterpolatePolarObjFlag == 1) {</code>		
<code> NumOfObj_s_NIPO;</code>	INT32	THE NUMBER OF POLAR-COORDINATE BASE FIXED OBJECTS
<code> }</code>		
<code> NonInterpolateCartesianObjFlag</code>	INT32	FLAG INDICATING PRESENCE OR ABSENCE OF ABSOLUTE-COORDINATE BASE FIXED OBJECT 0: ABSENT, 1: PRESENT
<code> if (NonInterpolateCartesianObjFlag == 1) {</code>		
<code> NumOfObj_s_NICO;</code>		THE NUMBER OF ABSOLUTE-COORDINATE BASE FIXED OBJECTS
<code> }</code>		
<code> InterpolatePolarObjFlag</code>	INT32	FLAG INDICATING PRESENCE OR ABSENCE OF POLAR-COORDINATE BASE INTERPOLATION OBJECT 0: ABSENT, 1: PRESENT
<code> if (InterpolatePolarObjFlag == 1) {</code>		
<code> NumOfObj_s_IPO;</code>		
<code> }</code>		
<code> NumOfAncBytes;</code>	INT32	SIZE OF EXTENDED INFORMATION REGION (MAX 255 BYTE)
<code> for (i = 0; i < NumOfAncBytes; i++) {</code>		
<code> AncByteData[i];</code>	INT32 (CHAR)	EXTENDED REGION BYTE DATA
<code> }</code>		
<code>}</code>		

Fig. 12

syntax	bit	Comment
object_metadata_fva_polar {		
fva_structure_info_polar {		CONFIGURATION INFORMATION
for (i = 0; i < NumOfRefViewPoint; i++) {		
object_metadata {		TRANSMIT ABSOLUTE-COORDINATE BASE INTERPOLATION OBJECTS (AAI-COMPLIANT OBJECTS) OBJECT_METADATA() AS MANY AS REF VIEWPOINTS
}		
if (NonInterpolatePolarObjFlag == 1) {		
object_metadata_nonintpPolar {		POLAR-COORDINATE BASE FIXED OBJECT
}		
if (NonInterpolateCartesianObjFlag == 1) {		
object_metadata_nonintpCarte {		ABSOLUTE-COORDINATE BASE FIXED OBJECT
}		
if (InterpolatePolarObjFlag == 1) {		
for (i = 0; i < NumOfRefViewPoint; i++) {		
object_metadata_intpPolar {		TRANSMIT POLAR-COORDINATE BASE INTERPOLATION OBJECTS OBJECT_METADATA() AS MANY AS REF VIEWPOINTS
}		
}		
}		

Fig. 13

syntax	bit	Comment
object_metadata_fva_polar {		
fva_structure_info_polar_present	INT32	CONFIGURATION INFORMATION PRESENCE FLAG
if (fva_structure_info_polar_present == 1) {		
fva_structure_info_polar {		CONFIGURATION INFORMATION
}		
for (i=0; i < 3; i++) {		
object_metadata {		TRANSMIT ABSOLUTE-COORDINATE BASE INTERPOLATION OBJECTS (AAI-COMPLIANT OBJECTS) OBJECT_METADATA() AS MANY AS REF VIEWPOINTS
}		
if (NonInterpolatePolarObjFlag == 1) {		
object_metadata_nonintpPolar {		POLAR-COORDINATE BASE FIXED OBJECT
}		
if (NonInterpolateCartesianObjFlag == 1) {		
object_metadata_nonintpCarte {		ABSOLUTE-COORDINATE BASE FIXED OBJECT
}		
if (InterpolatePolarObjFlag == 1) {		
for (i=0; i < 3; i++) {		
object_metadata_intpPolar {		TRANSMIT POLAR-COORDINATE BASE INTERPOLATION OBJECTS OBJECT_METADATA() AS MANY AS REF VIEWPOINTS
}		
}		
}		

Fig. 14

syntax	bit	Comment
<code>object_metadata_nonintpolar () {</code>		
<code>for (i=0; i < NumOfObs_NIP0; i++) {</code>		
<code>PosAzi[i];</code>	F loat32	AZIMUTH: ANGLE IN HORIZONTAL DIRECTION
<code>PosEle[i];</code>	F loat32	ELEVATION: ANGLE IN VERTICAL DIRECTION
<code>PosRad[i];</code>	F loat32	RADIUS: RADIUS
<code>Gain[i]</code>	F loat32	GAIN: GAIN AMOUNT
<code>}</code>		
<code>}</code>		

Fig. 15

syntax	bit	Comment
object_metadata_intpPolar () {		
for (i = 0; i < NumOfObjs_IP0; i++) {		
PosAzi[i];	F oat32	AZIMUTH: ANGLE IN HORIZONTAL DIRECTION
PosEle[i];	F oat32	ELEVATION: ANGLE IN VERTICAL DIRECTION
PosRad[i];	F oat32	RADIUS: RADIUS
Gain[i]	F oat32	GAIN: GAIN AMOUNT
}		
}		

Fig. 16

syntax	bit	Comment
object_metadata_nonintpCarte() {		
for (i = 0; i < NumOf0bjs_NICO; i++) {		
PosX[i];	F oat32	X-COORDINATE POSITION INFORMATION
PosY[i];	F oat32	Y-COORDINATE POSITION INFORMATION
PosZ[i];	F oat32	Z-COORDINATE POSITION INFORMATION
Gain[i]	F oat32	GAIN: GAIN AMOUNT
}		
}		

Fig. 17

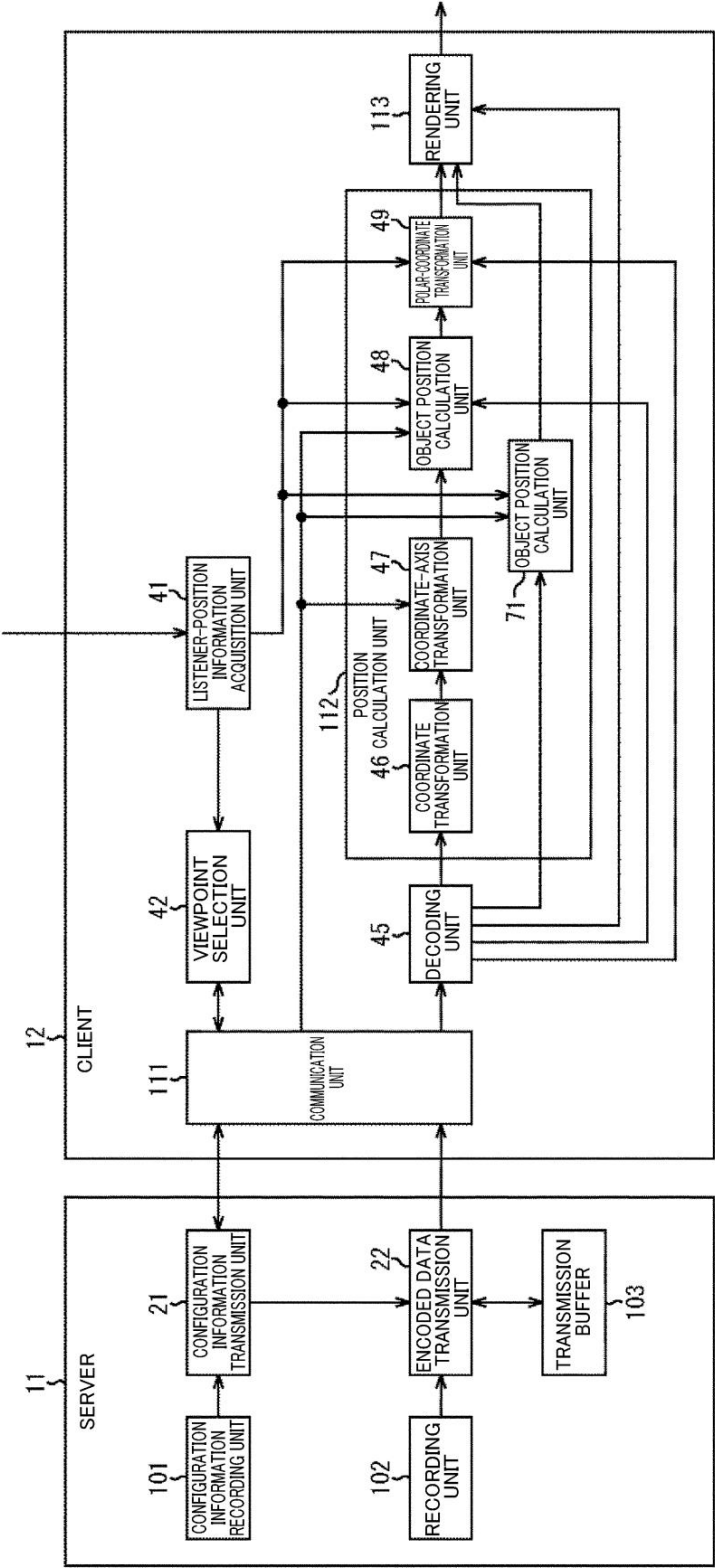


Fig. 18

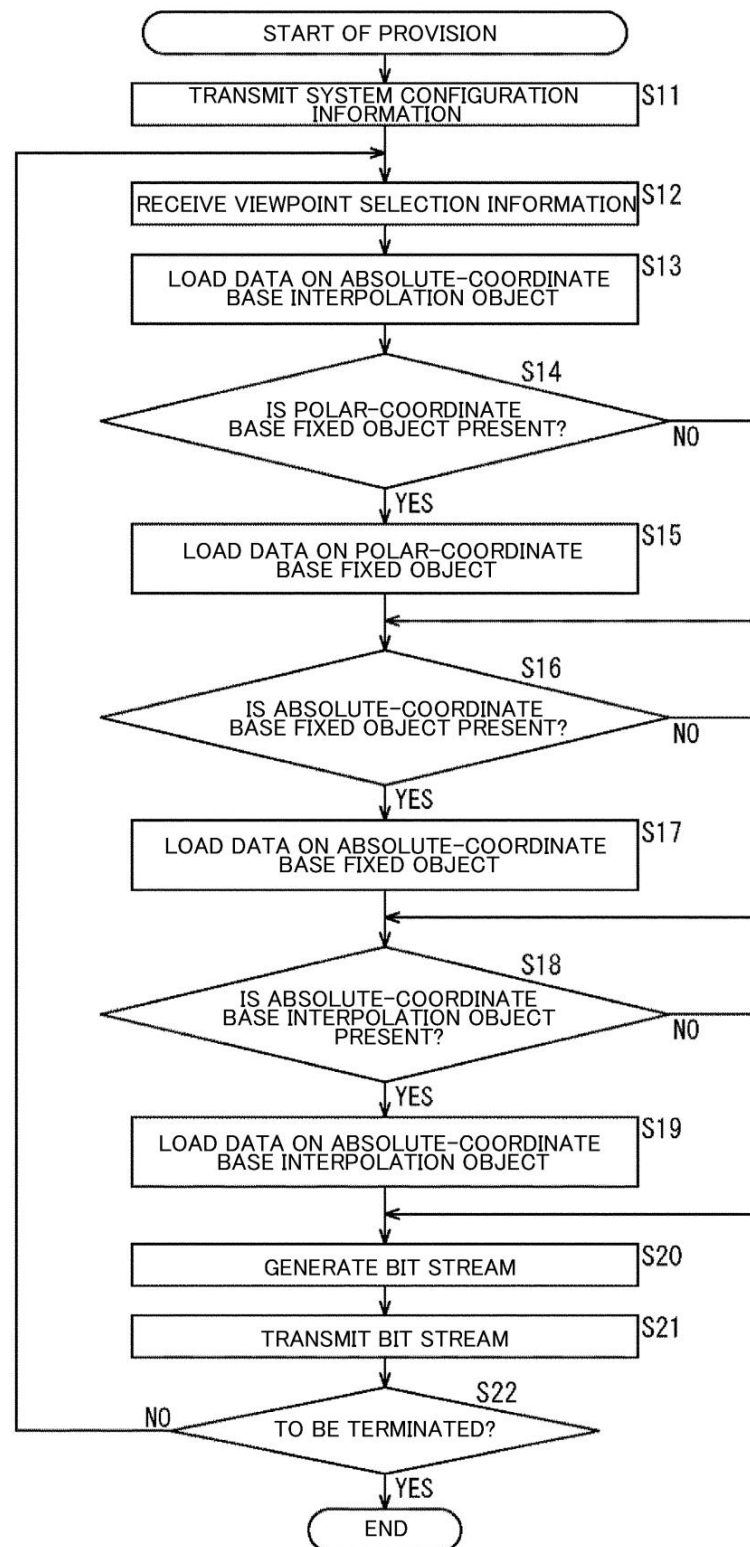


Fig. 19

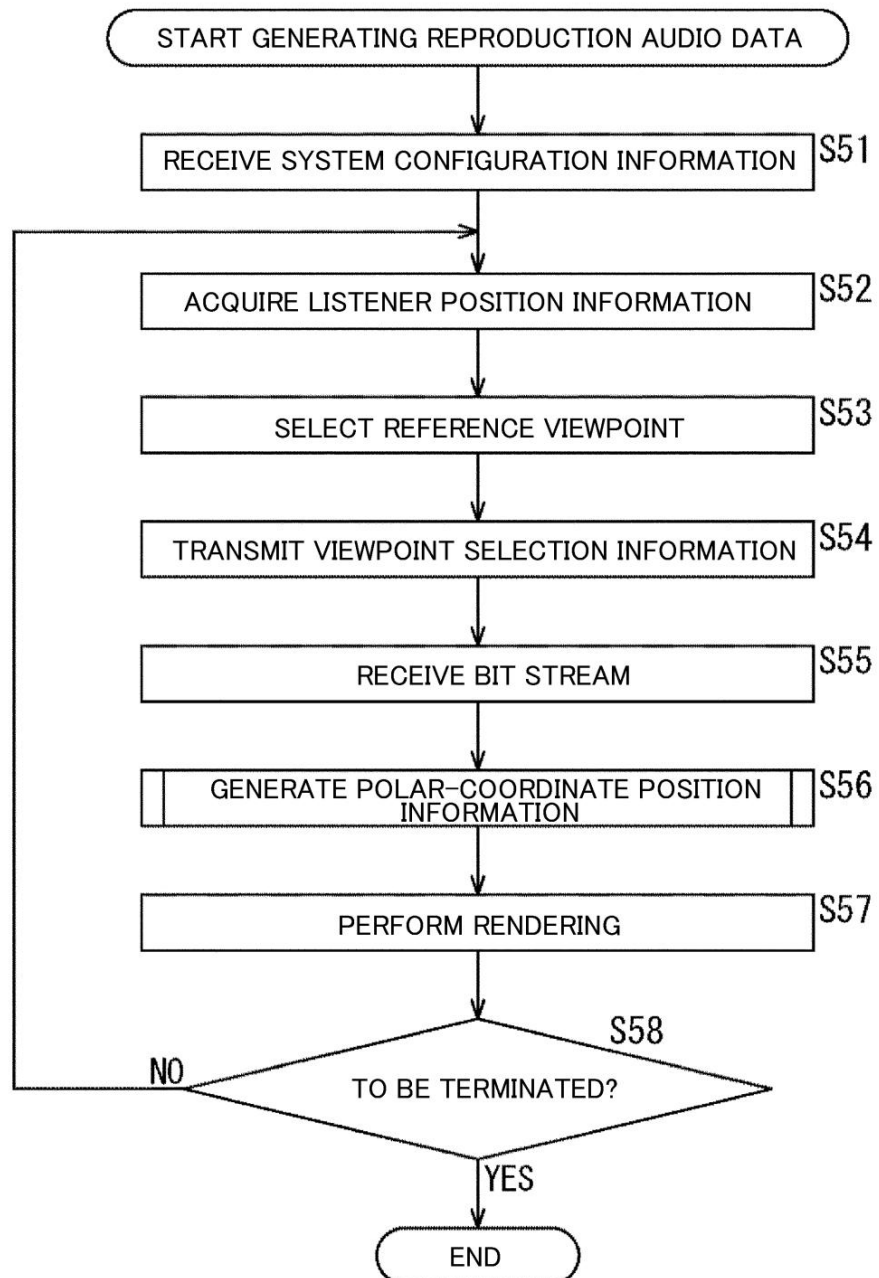


Fig. 20

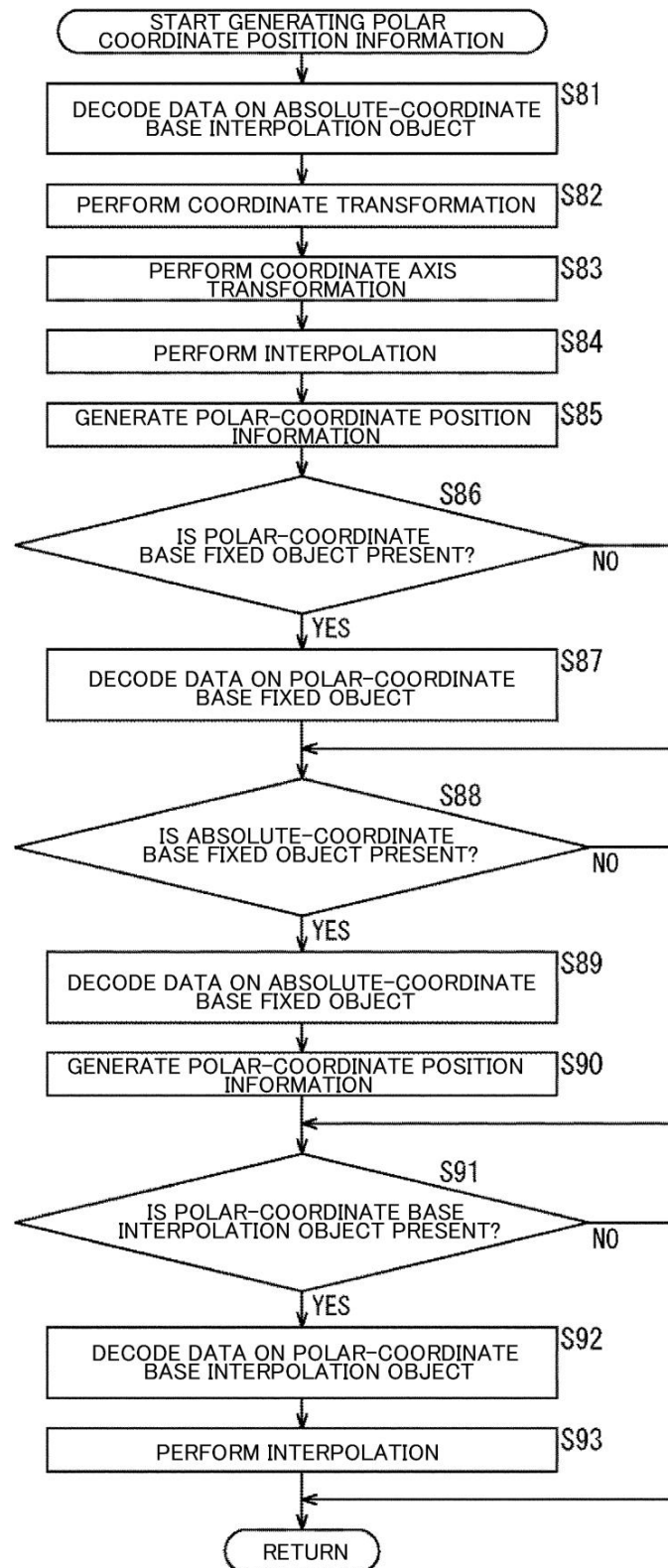


Fig. 21

