FIG. 1

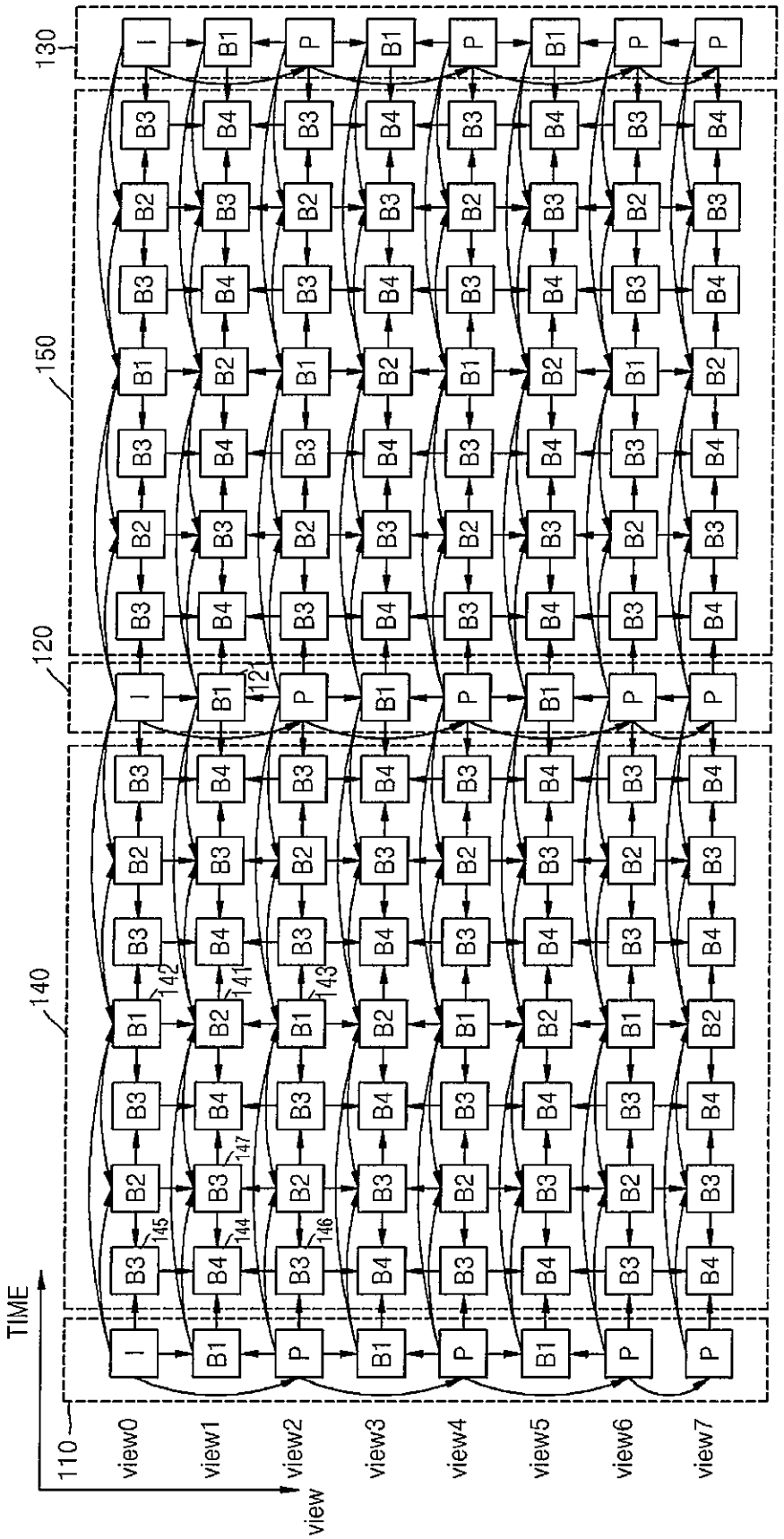

TIME
view
110
130
150
120
140
view0
view1
view2
view3
view4
view5
view6
view7
I
P
B1
B2
B3
B4
142
141
143
145
144
146
147

FIG. 2

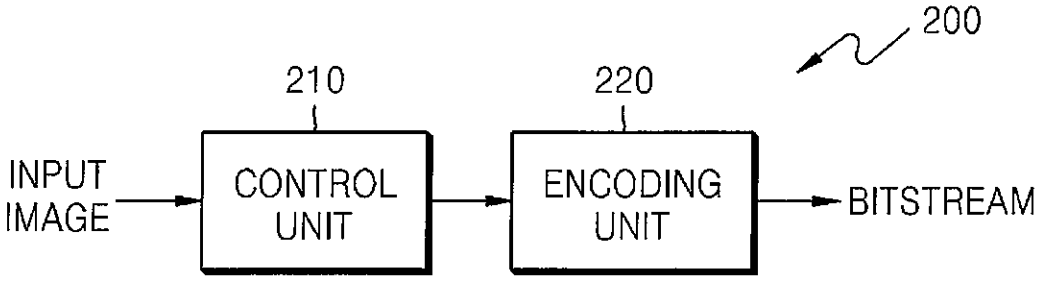

200
210
220
INPUT IMAGE
CONTROL UNIT
ENCODING UNIT
BITSTREAM

FIG. 3

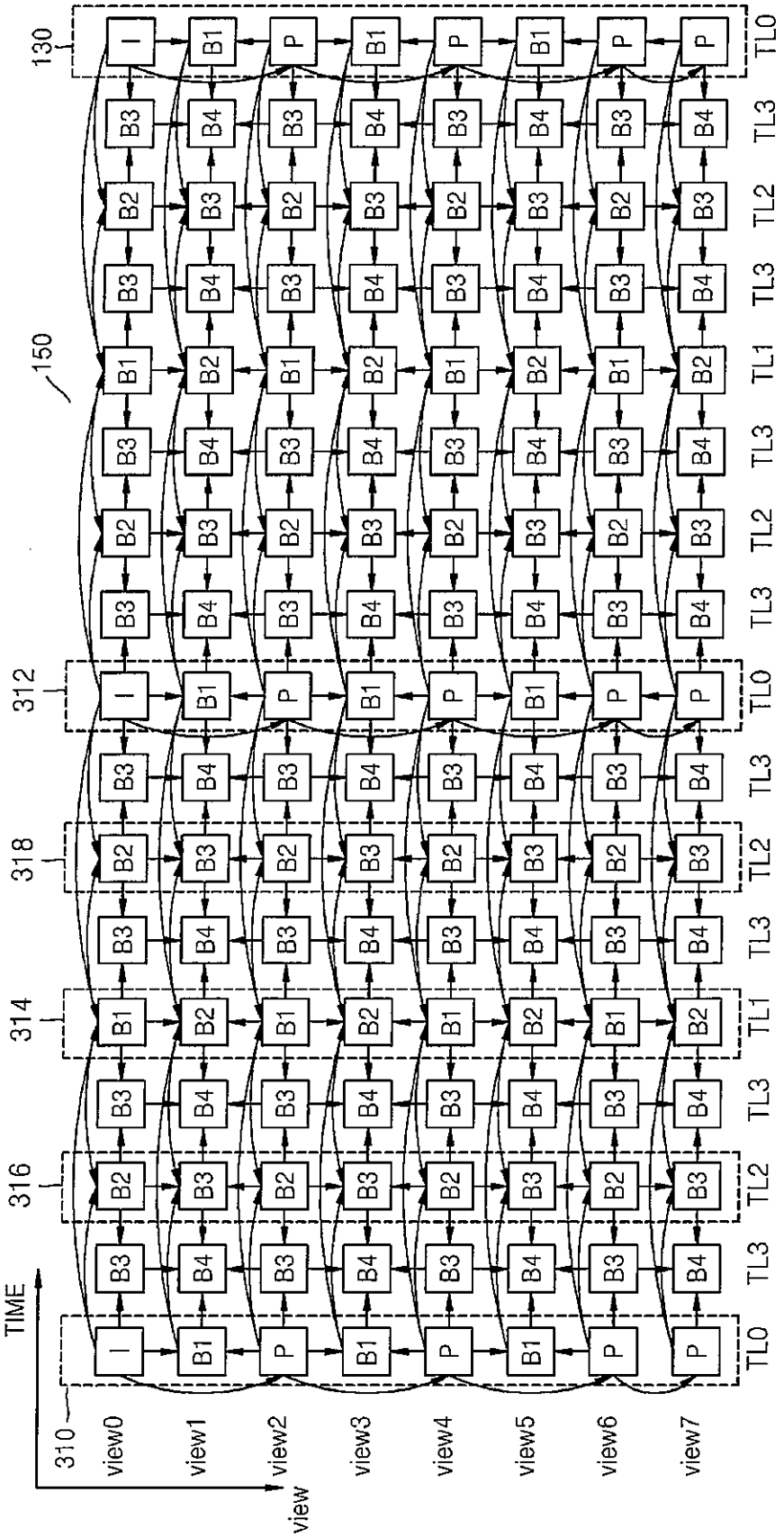

TIME
view
view0
view1
view2
view3
view4
view5
view6
view7
310
316
314
318
312
150
130
I
P
B1
B2
B3
B4
TL0
TL1
TL2
TL3

FIG. 4A

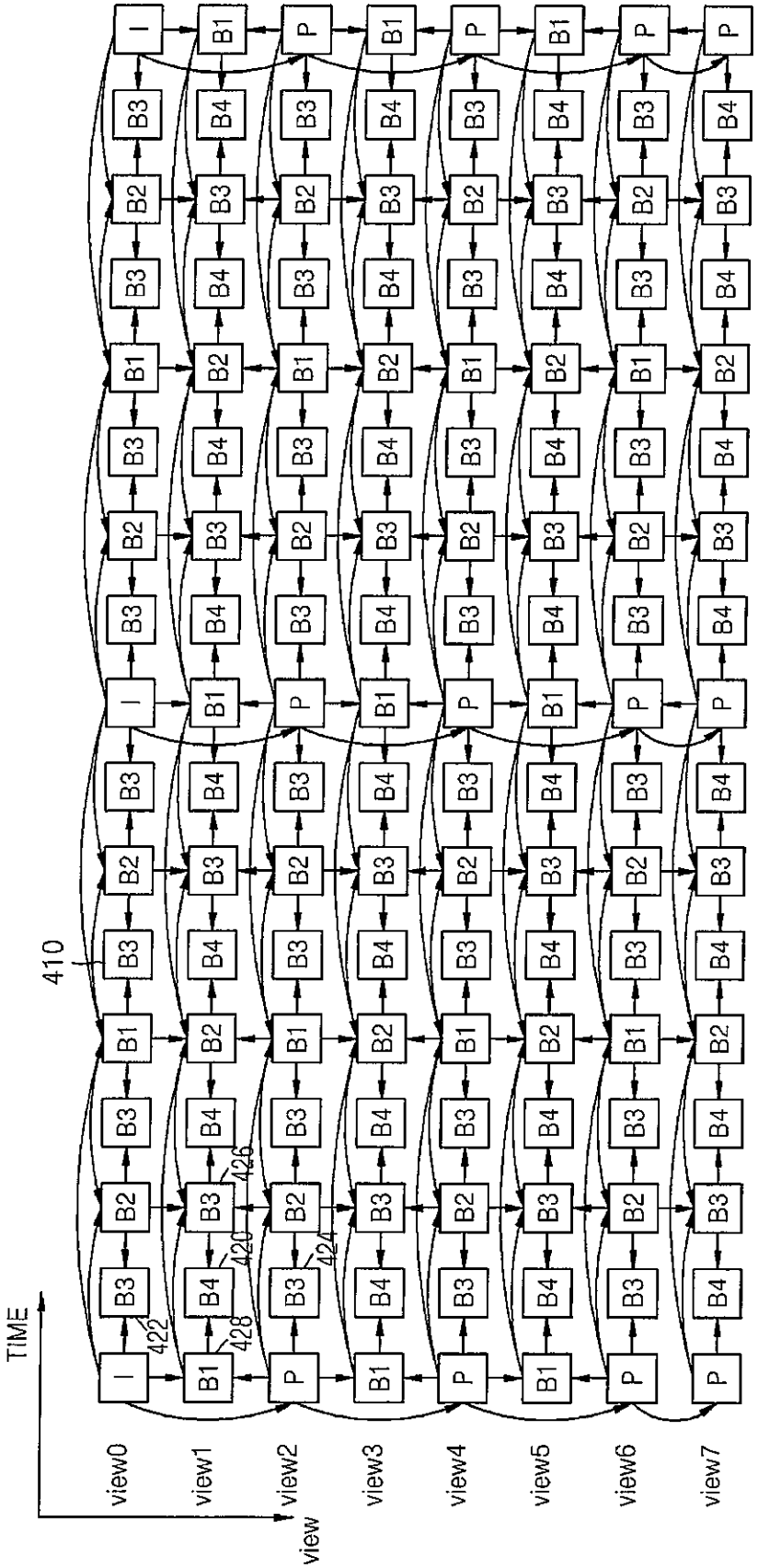

410
422
420
424
426
428
TIME
view
view0
view1
view2
view3
view4
view5
view6
view7
I
P
B1
B2
B3
B4

FIG. 4B

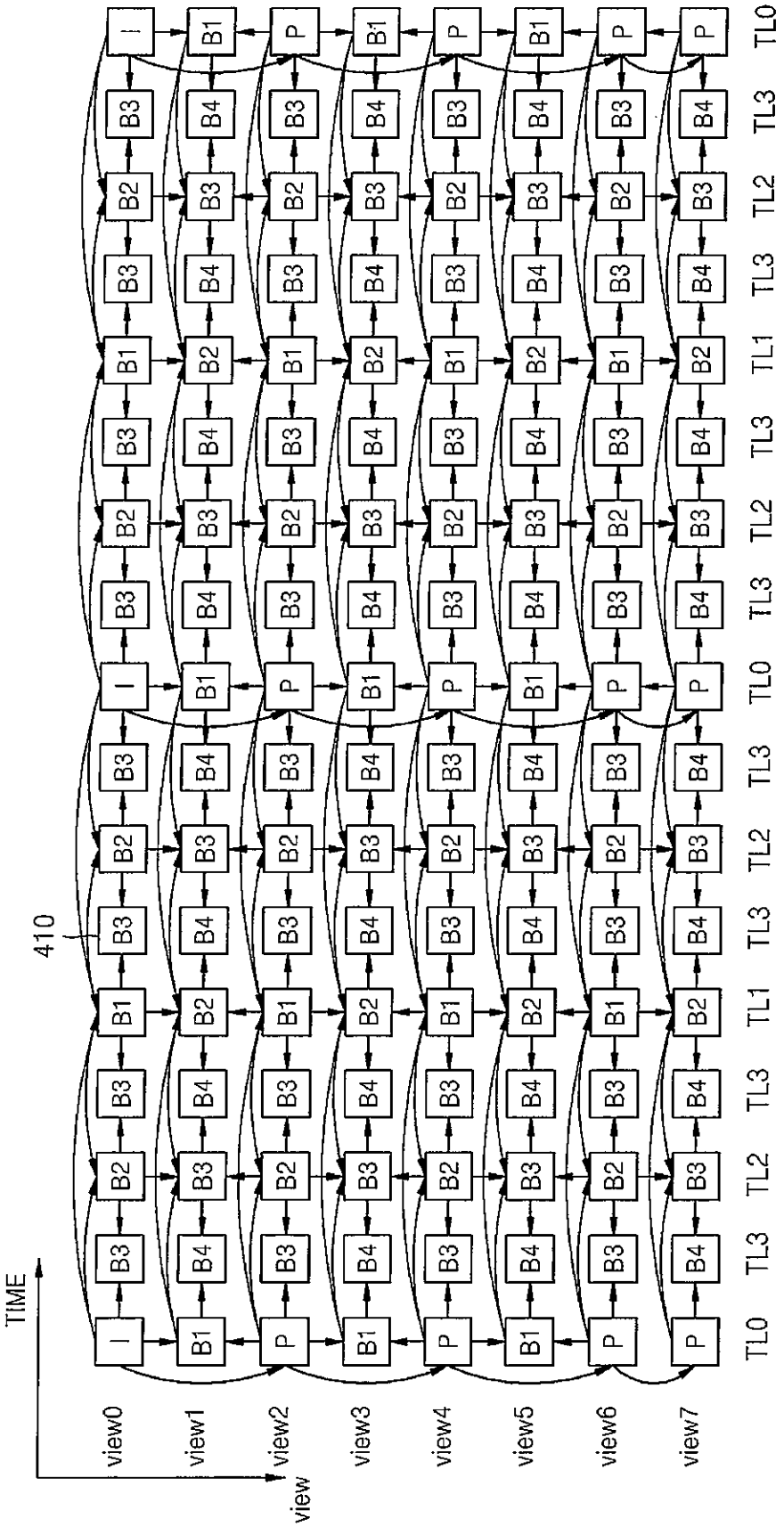

410
TIME
view
view0
view1
view2
view3
view4
view5
view6
view7
I
B1
P
B2
B3
B4
TL0
TL1
TL2
TL3

FIG. 5

```
nal_unit_header_svc_mvc_extension(){
    svc_mvc_flag
    if(!svc_mvc_flag){
    reserved_zero_one_bit
    priority_id
    temporal_level
    dependency_id
    quality_level
    idr_flag
    layer_base_flag
    use_base_prediction_flag
    discardable_flag
    reserved_zero_three_bits
    tl0_frame_idx_present_flag
    fragment_order
    } else{
     nal_ref_idc_view
       temproal_level
       view_level
       achor_pic_flag
       view_id
       idr_flag
     reserved_zero_four_bits
    }
   nalUnitHeaderBytes+=3
}
```

FIG. 6A

```
seq_parameter_set_mvc_extension(){
        num_views_minus_1
        temporal_level_max
        for(i=o; i<=num_views_minus_1; i++)
                view_id[i]
        for(i=0; i<=num_views_minus_1; i++){
                num_anchor_refs_l0[i]
                for(j=0; j<=num_anchor_refs_l0[i]; j++)
                        anchor_ref_l0[i][j]
                num_anchor_refs_l1[i]
                for(j=0; j<=num_anchor_refs_l1[i]; j++)
                        anchor_ref_l1[i][j]
        }
        for(i=0; i<=num_views_minus_1; i++){
                num_non_anchor_refs_l0[i]
                for(j=0; j<=num_non_anchor_refs_l0[i]; j++)
                        non_anchor_ref_l0[i][j]
                num_non_anchor_refs_l1[i]
                for(j=0; j<=num_non_anchor_refs_l1[i]; j++)
                        non_anchor_ref_l1[i][j]
        }
        for(i=1; i<=temporal_level_max; i++)
                use_interview_pred_temporal_level[i];
}
```

FIG. 6B

```
seq_parameter_set_mvc_extension(){
        num_views_minus_1
        temporal_level_max
        for(i=o; i<=num_views_minus_1; i++)
                view_id[i]
        for(i=0; i<=num_views_minus_1; i++){
                num_anchor_refs_l0[i]
                for(j=0; j<=num_anchor_refs_l0[i]; j++)
                        anchor_ref_l0[i][j]
                num_anchor_refs_l1[i]
                for(j=0; j<=num_anchor_refs_l1[i]; j++)
                        anchor_ref_l1[i][j]
        }
        for(i=0; i<=num_views_minus_1; i++){
                num_non_anchor_refs_l0[i]
                for(j=0; j<=num_non_anchor_refs_l0[i]; j++)
                        non_anchor_ref_l0[i][j]
                num_non_anchor_refs_l1[i]
                for(j=0; j<=num_non_anchor_refs_l1[i]; j++)
                        non_anchor_ref_l1[i][j]
                for(j=1; j<=temporal_level_max; j++)
                        use_interview_pred_temporal_level[i][j];
        }

}
```

FIG. 6C

```
Use_interview_prediction_temporal_level(payloadSize){
  temporal_level_max
    for(i=1; i<=temporal_level_max; i++) {
      use_interview_prediction_temporal_level[i][j];
    }
}
```

FIG. 6D

```
Use_interview_prediction_temporal_level(payloadSize){
(
  num_view_minus_1
  temporal_level_max
   for(i=0; i<= num_view_minus1;i++) {
    for(j=0; j<=temporal_level_max; j++) {
      use_interview_prediction_temporal_level[i][j];
    }
  }
}
```

FIG. 7

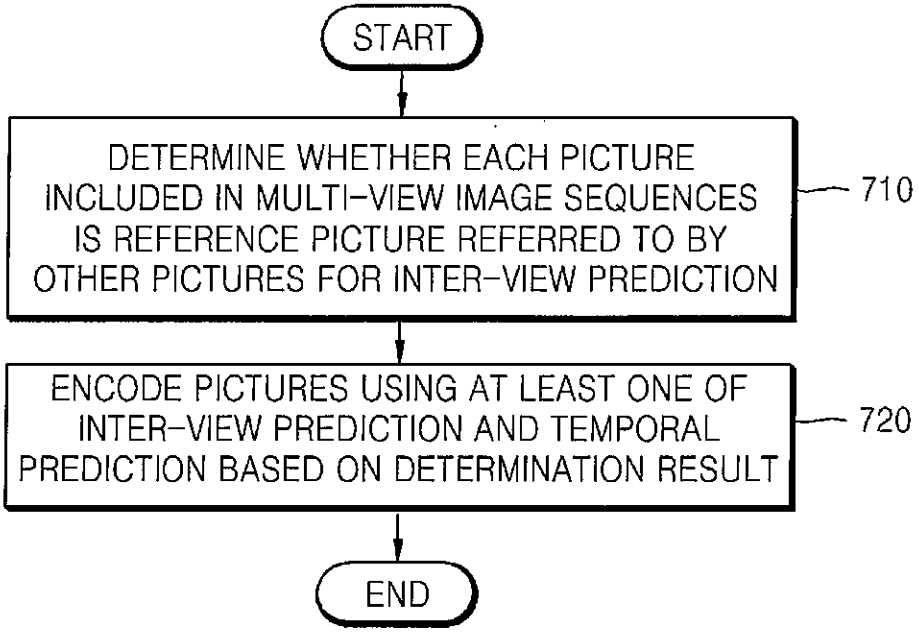

START
DETERMINE WHETHER EACH PICTURE INCLUDED IN MULTI-VIEW IMAGE SEQUENCES IS REFERENCE PICTURE REFERRED TO BY OTHER PICTURES FOR INTER-VIEW PREDICTION
710
ENCODE PICTURES USING AT LEAST ONE OF INTER-VIEW PREDICTION AND TEMPORAL PREDICTION BASED ON DETERMINATION RESULT
720
END

FIG. 8

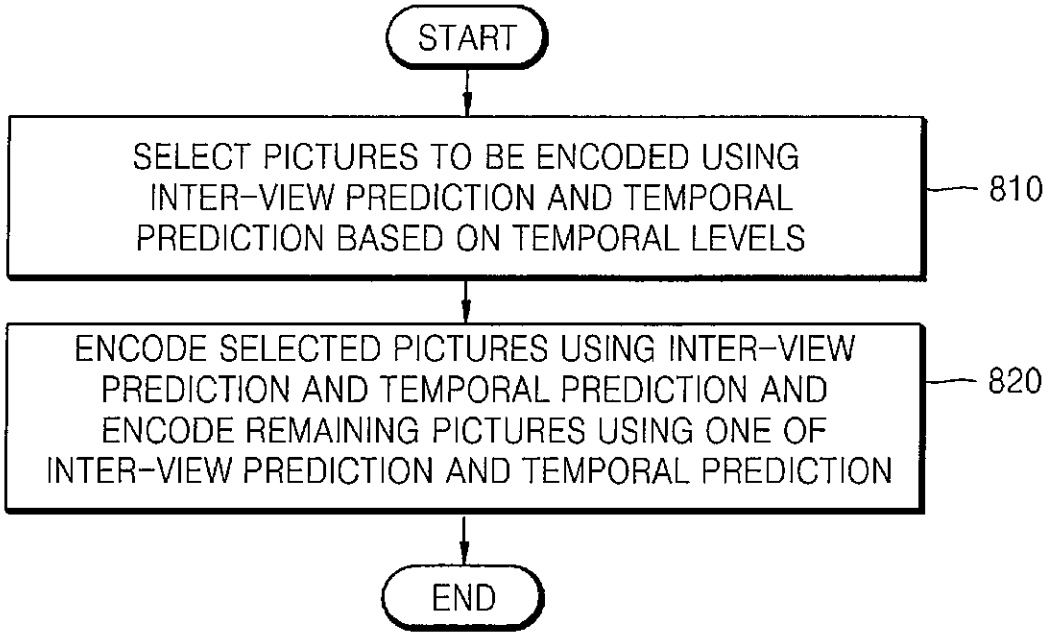

START
SELECT PICTURES TO BE ENCODED USING INTER-VIEW PREDICTION AND TEMPORAL PREDICTION BASED ON TEMPORAL LEVELS
810
ENCODE SELECTED PICTURES USING INTER-VIEW PREDICTION AND TEMPORAL PREDICTION AND ENCODE REMAINING PICTURES USING ONE OF INTER-VIEW PREDICTION AND TEMPORAL PREDICTION
820
END

FIG. 9

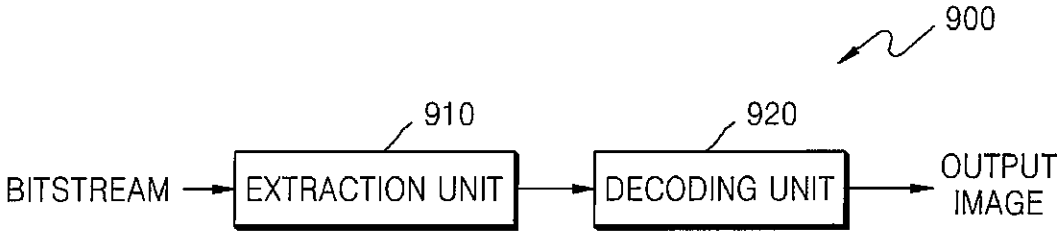
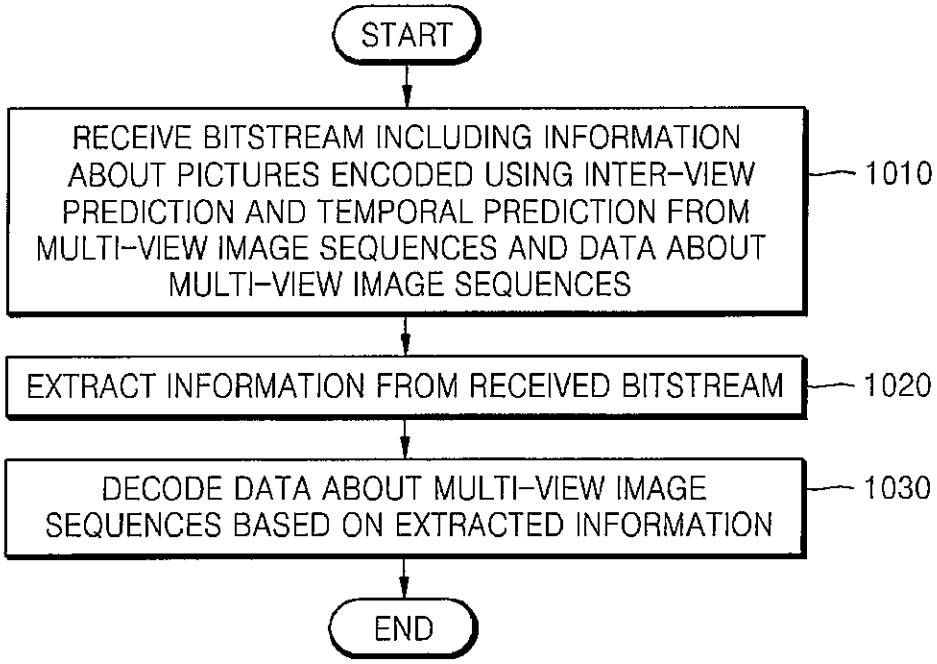

900
910
920
BITSTREAM
EXTRACTION UNIT
DECODING UNIT
OUTPUT IMAGE

FIG. 10


START
RECEIVE BITSTREAM INCLUDING INFORMATION ABOUT PICTURES ENCODED USING INTER-VIEW PREDICTION AND TEMPORAL PREDICTION FROM MULTI-VIEW IMAGE SEQUENCES AND DATA ABOUT MULTI-VIEW IMAGE SEQUENCES
1010
EXTRACT INFORMATION FROM RECEIVED BITSTREAM
1020
DECODE DATA ABOUT MULTI-VIEW IMAGE SEQUENCES BASED ON EXTRACTED INFORMATION
1030
END