US010523914B1

(12) **United States Patent**
Phillips et al.

(10) **Patent No.: US 10,523,914 B1**
(45) **Date of Patent: Dec. 31, 2019**

(54) **SYSTEM AND METHOD FOR PROVIDING MULTIPLE 360° IMMERSIVE VIDEO SESSIONS IN A NETWORK**

(71) Applicant: **TELEFONAKTIEBOLAGET L M ERICSSON (PUBL)**, Stockholm (SE)

(72) Inventors: **Chris Phillips**, Hartwell, GA (US); **Robert Hammond Forsman**, Sugar Hill, GA (US); **Sarel Cohen**, Tel-Aviv (IL)

(73) Assignee: **Telefonaktiebolaget LM Ericsson (publ)**, Stockholm (SE)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **16/046,629**

(22) Filed: **Jul. 26, 2018**

(51) **Int. Cl.**

H04N 13/161	(2018.01)
H04N 7/15	(2006.01)
H04N 13/156	(2018.01)
H04N 19/597	(2014.01)
H04L 29/06	(2006.01)

(Continued)

(52) **U.S. Cl.**
CPC ***H04N 13/161*** (2018.05); ***H04L 65/601*** (2013.01); ***H04L 65/80*** (2013.01); ***H04N 7/15*** (2013.01); ***H04N 13/156*** (2018.05); ***H04N 19/11*** (2014.11); ***H04N 19/154*** (2014.11); ***H04N 19/172*** (2014.11); ***H04N 19/597*** (2014.11)

(58) **Field of Classification Search**
USPC 348/14.03
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,975,283 B2 * 7/2011 Bedingfield, Sr. H04N 7/17318 725/12
8,767,014 B2 7/2014 Vaught et al.
(Continued)

FOREIGN PATENT DOCUMENTS

GB	2554877	4/2018
GB	2557416 A	6/2018

(Continued)

OTHER PUBLICATIONS

USPTO, Notice of Allowance, U.S. Appl. No. 16/046,576, dated Apr. 30, 2019, 8 pgs.

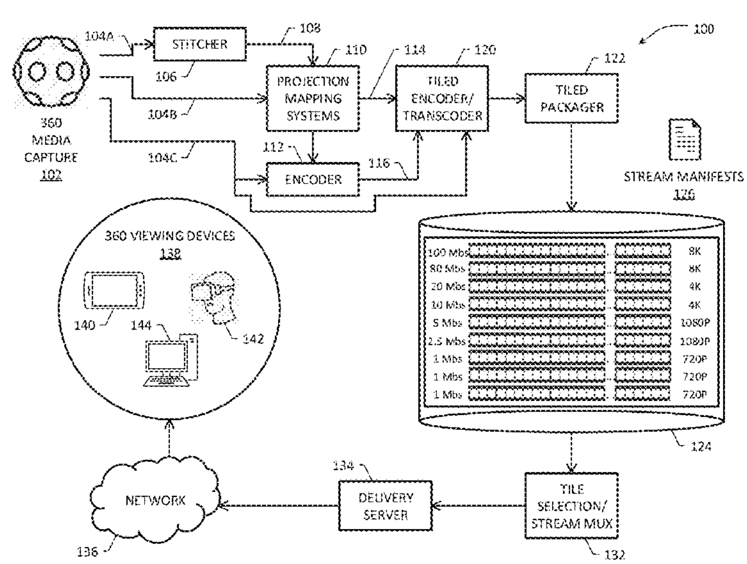
(Continued)

Primary Examiner — Amal S Zenati
(74) *Attorney, Agent, or Firm* — The Danamraj Law Group, P.C.

(57) **ABSTRACT**

A system and method for providing multisession 360° immersive video services in a network portion having a dedicated bandwidth capacity. In one embodiment, a bandwidth optimization method is operative to determine respective bandwidth allocations for the video sessions responsive to receiving an indication of the dedicated bandwidth capacity, a plurality of weights corresponding to the video sessions and a plurality of bitrate caps configured for the video sessions. Respective bandwidth allocations may be provided to a tile selection process configured to select tiles for a corresponding video session on a frame-by-frame basis from a plurality of tile-encoded bitrate representations of a media input stream associated with the video session, wherein each tile-encoded bitrate representation is provided with a separate video quality.

24 Claims, 50 Drawing Sheets



(51) **Int. Cl.**

H04N 19/154	(2014.01)
H04N 19/11	(2014.01)
H04N 19/172	(2014.01)

(56) **References Cited**

U.S. PATENT DOCUMENTS

9,392,212	B1	7/2016	Ross	
9,918,136	B2	3/2018	Cole et al.	
2015/0212576	A1	7/2015	Ambrus et al.	
2015/0288732	A1*	10/2015	Phillips	H04L 65/60 709/219
2015/0346812	A1	12/2015	Cole et al.	
2015/0382057	A1	12/2015	Huang	
2016/0277772	A1*	9/2016	Campbell	H04N 21/21805
2016/0353146	A1*	12/2016	Weaver	H04N 21/2662
2017/0139578	A1	5/2017	Dickerson et al.	
2017/0339416	A1	11/2017	Hendry et al.	
2017/0374375	A1	12/2017	Makar et al.	
2017/0374411	A1*	12/2017	Lederer	H04N 21/8456
2018/0070113	A1	3/2018	Phillips et al.	
2018/0070119	A1	3/2018	Phillips et al.	
2018/0160160	A1	6/2018	Swaminathan et al.	
2019/0110856	A1	4/2019	Barral	

FOREIGN PATENT DOCUMENTS

JP	WO2016129549	8/2016
WO	WO2018047063 A1	3/2018

OTHER PUBLICATIONS

5G Americas, "Network Slicing for 5G Networks & Services", 5G Americas White Paper—Network Slicing for 5G and Beyond, Nov. 2016, 35 pgs.

Bassbouss et al., "High Quality 360° Video Rendering and Streaming," Fraunhofer Institute for Open Communication Systems FOKUS, Nov. 2016, 4 pgs.

Corbillon et al., "Viewport-Adaptive Navigable 360-Degree Video Delivery," 2017 IEEE International Conference on Communications (ICC), May 21, 2017, 7 pgs.

Fallon et al., "Using the COMPA Autonomous Architecture for Mobile Network Security", 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), May 2017, 8 pgs.

GSMA, "An Introduction to Network Slicing", 2017, 20 pgs.

Kuzyakov et al., "Next-generation video encoding techniques for 360 video and VR," Engineering Blog: Facebook Code—Facebook, Jan. 21, 2016, 7 pgs. <https://code.facebook.com/posts/1126354007399553/next-generation-video-encoding-techniques-for-360-video-and-vr/>.

Misra et al., "An Overview of Tiles in HEVC," IEEE Journal of Selected Topics in Signal Processing, Dec. 2013, pp. 969-970, vol. 7, No. 6.

Rune et al., "Architecture evolution for automation and network programmability", Ericsson Review, Nov. 28, 2014, 11 pgs.

Sanchez et al., "Compressed domain video processing for tile based panoramic streaming using HEVC," 2015 IEEE International Conference on Image Processing (ICIP), Sep. 27, 2015, 5 pgs.

Sanchez et al., "Low Complexity Cloud-video-Mixing Using HEVC," 11th Annual IEEE CCNC—Multimedia Networking, Services and Applications, Jan. 10, 2014, pp. 213-218, Las Vegas, Nevada, USA.

Skupin et al., "HEVC Tile Based Streaming to Head Mounted Displays," 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), Jan. 8, 2017, pp. 613-615.

Sullivan et al., "Overview of the High Efficiency Video Coding(HEVC) Standard", IEEE Transactions on Circuits and Systems for Video Technology, Dec. 2012, 20 pgs., vol. 22, No. 12.

Valero et al., "Eye-tracking Based Video Streaming System with Fast Quality Switching," Master's Thesis at KTH Information and Communication Technology, Jul. 2017, Stockholm, Sweden, 89 pgs.

Van Brandenburg, "CDN Optimization for VR Streaming," Proceedings of the IBC 2017 Conference, Amsterdam, The Netherlands, Sep. 2017, 9 pgs.

Advertisement Overlay Insertion for HEVC based services, Fraunhofer Heinrich Hertz Institute, accessed Nov. 10, 2017. <https://www.hhi.fraunhofer.de/en/departments/vca/research-groups/multimedia-commuications/research-topics/advertisement-overlay-insertion-for-hevc-based-services.html>.

PCT, International Search Report, International Application No. PCT/EP2019/061840, Jun. 18, 2019, 16 pgs.

ISA/SE, International Search Report, PCT/SE2019/050559, Sep. 18, 2019, 4 pgs.

ISA/SE, Written Opinion of the International Search Authority, PCT/SE2019/050559, Sep. 18, 2019, 8 pgs.

ISA/SE, International Search Report, PCT/SE2019/050552, dated Sep. 20, 2019, 5 pgs.

ISA/SE, Written Opinion of the International Search Authority, PCT/SE2019/050552, dated Sep. 20, 2019, 9 pgs.

Guo et al., Viewpoint Switching in Multiview Video Streaming, 2005 IEEE International Symposium on Circuits and Systems (ISCAS), May 23-26, 2005, pp. 3471-3474, IEEE, Japan.

Ling et al., Adaptive Multi-Path Video Streaming, Proceedings of the Eighth IEEE International Symposium on Multimedia (ISM'06), Dec. 11-13, 2006, IEEE Computer Society, 8 pgs.

Lungaro et al., Gaze-aware streaming solutions for the next generation of mobile VR experiences, IEEE Transactions on Visualization and Computer Graphics, Jan. 25, 2018, pp. 1535-1544, vol. 24, IEEE.

Ozcinar et al., Dynamic bitrate allocation of interactive real-time streamed multi-view video with view-switch prediction, Signal, Image and Video Processing, Mar. 27, 2017, pp. 1279-1285, vol. 11, Springer-Verlag, United Kingdom.

Zhu et al., Inter-view-predicted redundant pictures for viewpoint switching in multiview video streaming, 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, Mar. 14-19, 2010, pp. 746-749, IEEE, USA.

* cited by examiner

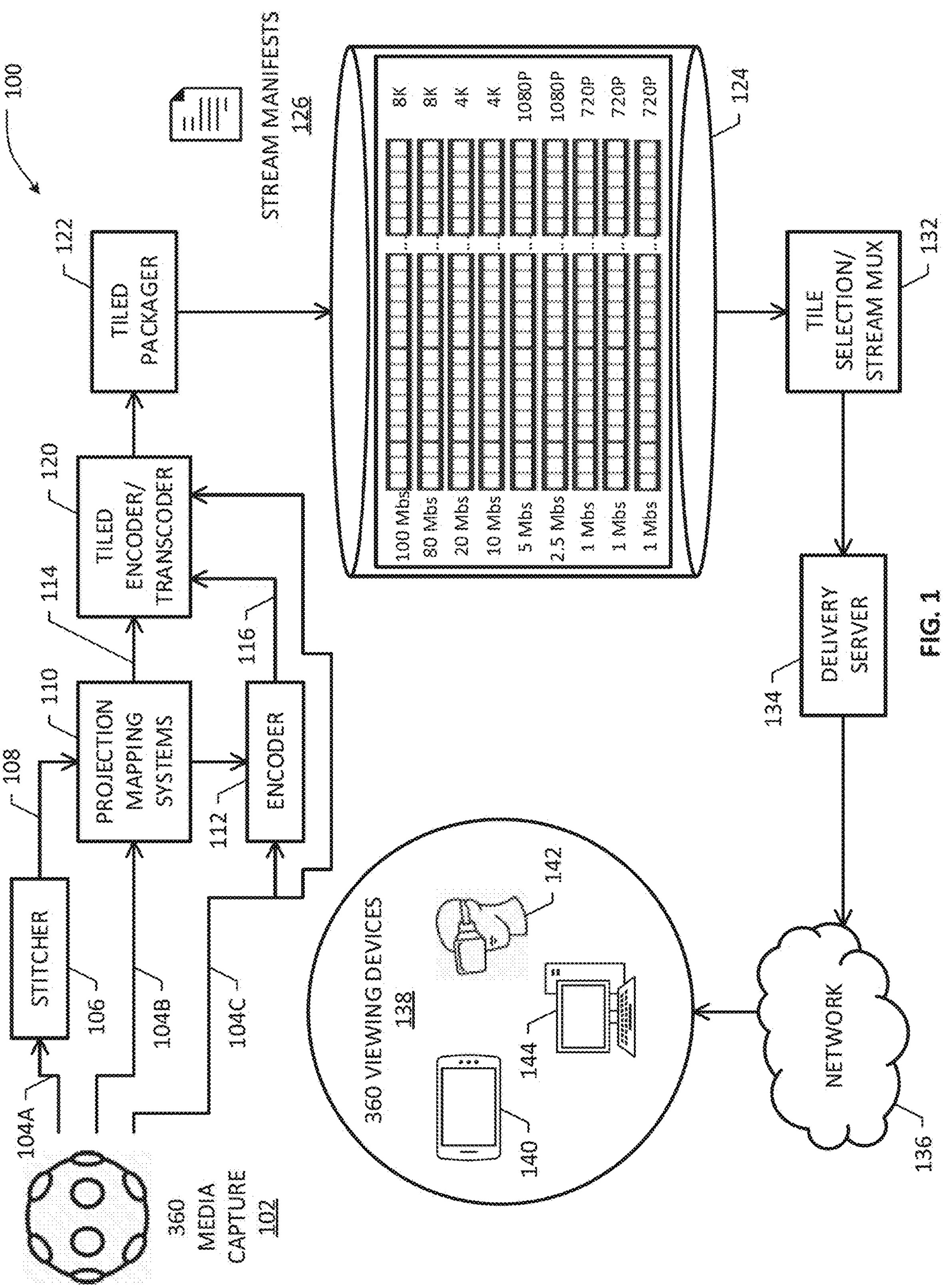

100
360 MEDIA CAPTURE 102
104A
104B
104C
STITCHER
106
108
PROJECTION MAPPING SYSTEMS
110
ENCODER
112
114
116
TILED ENCODER/ TRANSCODER
120
TILED PACKAGER
122
STREAM MANIFESTS 126
124
100 Mbs
80 Mbs
20 Mbs
10 Mbs
5 Mbs
2.5 Mbs
1 Mbs
1 Mbs
1 Mbs
8K
8K
4K
4K
1080P
1080P
720P
720P
720P
TILE SELECTION/ STREAM MUX
132
DELIVERY SERVER
134
NETWORK
136
360 VIEWING DEVICES 138
140
142
144

FIG. 1

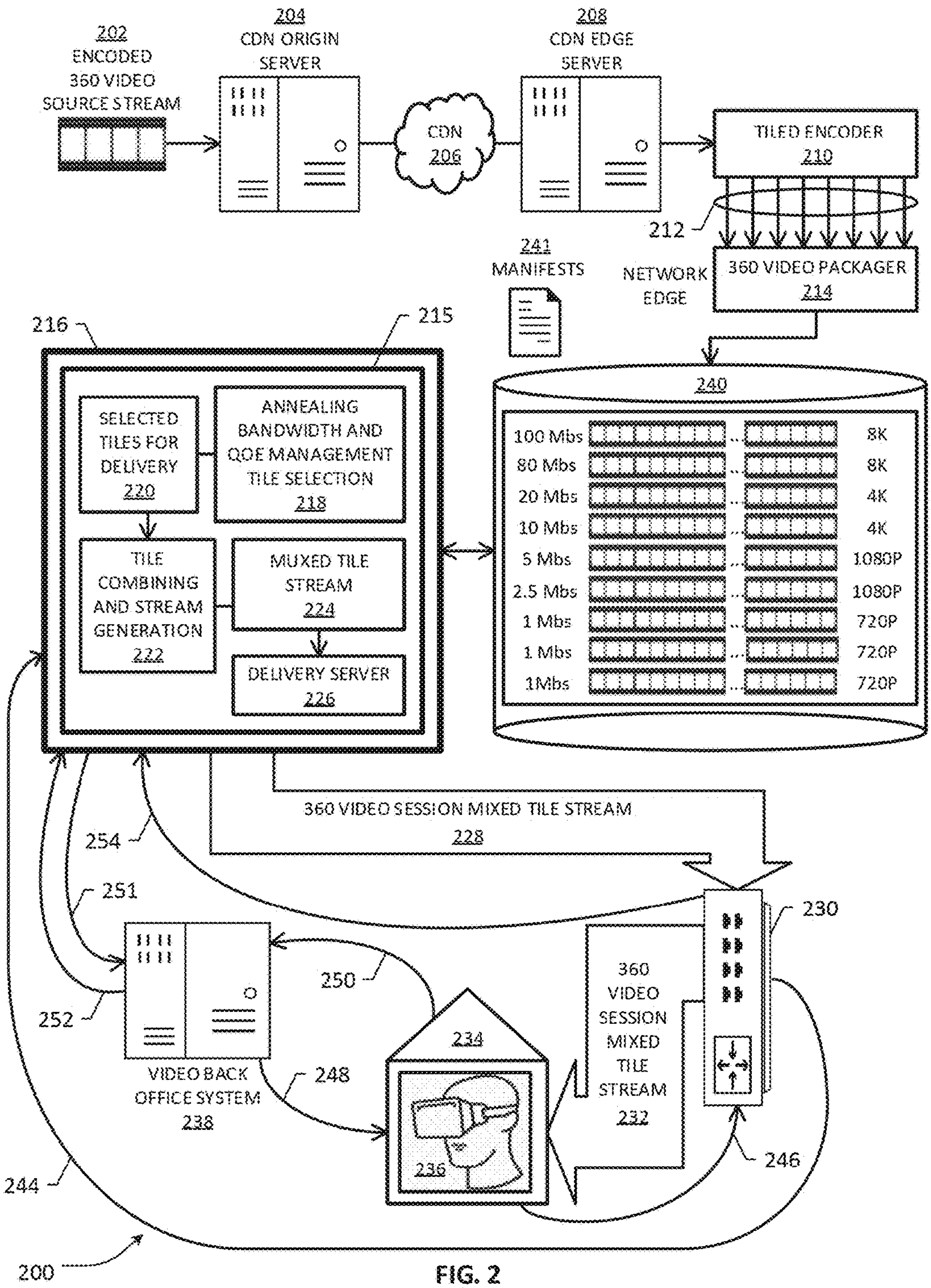

202
ENCODED
360 VIDEO
SOURCE STREAM
204
CDN ORIGIN
SERVER
CDN
206
208
CDN EDGE
SERVER
TILED ENCODER
210
212
360 VIDEO PACKAGER
214
NETWORK
EDGE
241
MANIFESTS
216
215
SELECTED
TILES FOR
DELIVERY
220
ANNEALING
BANDWIDTH AND
QOE MANAGEMENT
TILE SELECTION
218
TILE
COMBINING
AND STREAM
GENERATION
222
MUXED TILE
STREAM
224
DELIVERY SERVER
226
240
100 Mbs 8K
80 Mbs 8K
20 Mbs 4K
10 Mbs 4K
5 Mbs 1080P
2.5 Mbs 1080P
1 Mbs 720P
1 Mbs 720P
1Mbs 720P
360 VIDEO SESSION MIXED TILE STREAM
228
254
251
230
250
252
VIDEO BACK
OFFICE SYSTEM
238
234
236
248
360
VIDEO
SESSION
MIXED
TILE
STREAM
232
246
244
200

FIG. 2

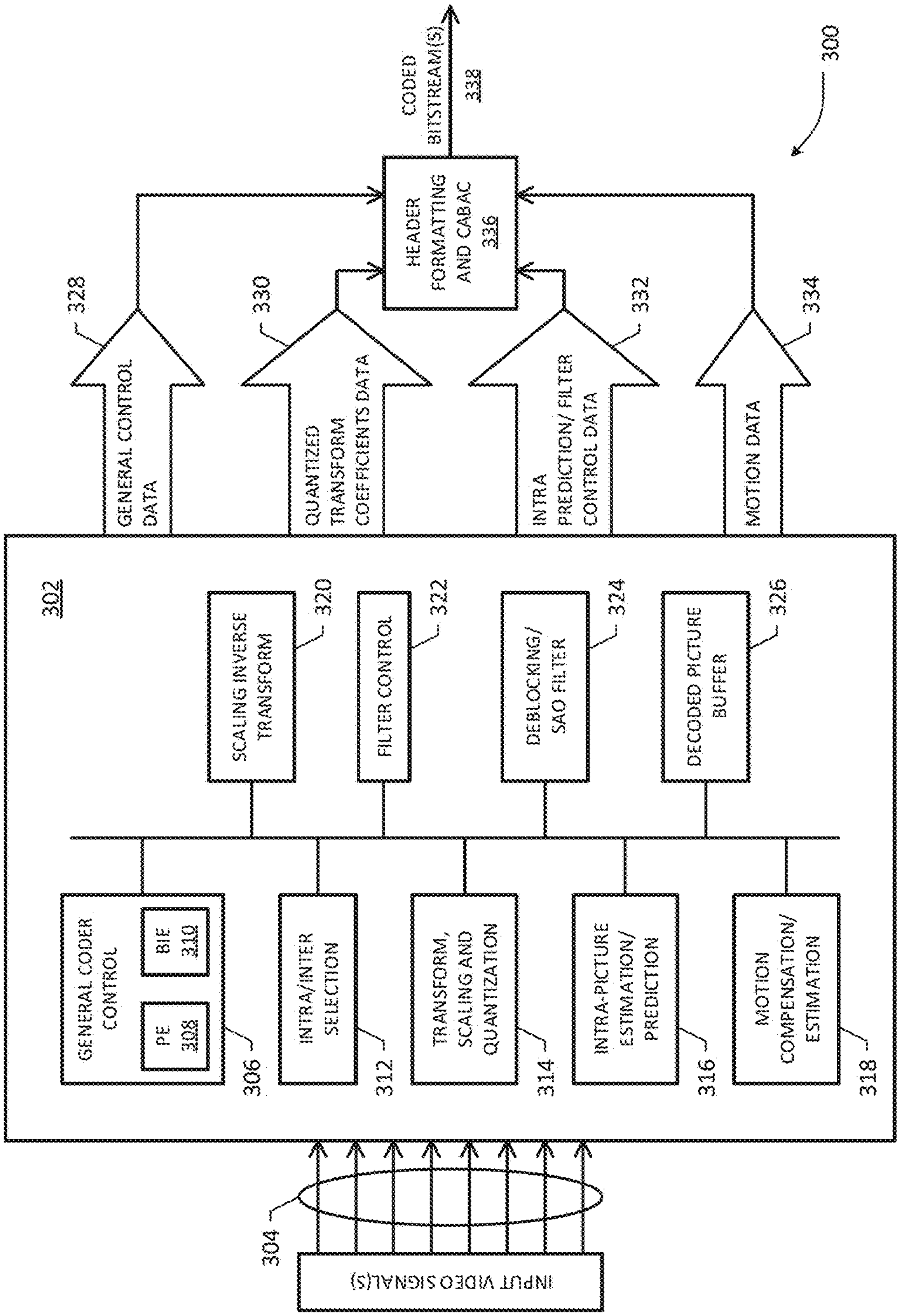

CODED BITSTREAM(S) 338
300
HEADER FORMATTING AND CABAC 336
328
330
332
334
GENERAL CONTROL DATA
QUANTIZED TRANSFORM COEFFICIENTS DATA
INTRA PREDICTION/ FILTER CONTROL DATA
MOTION DATA
302
SCALING INVERSE TRANSFORM
320
FILTER CONTROL
322
DEBLOCKING/ SAO FILTER
324
DECODED PICTURE BUFFER
326
GENERAL CODER CONTROL
PE 308
BIE 310
306
INTRA/INTER SELECTION
312
TRANSFORM, SCALING AND QUANTIZATION
314
INTRA-PICTURE ESTIMATION/ PREDICTION
316
MOTION COMPENSATION/ ESTIMATION
318
304
INPUT VIDEO SIGNAL(S)

FIG. 3

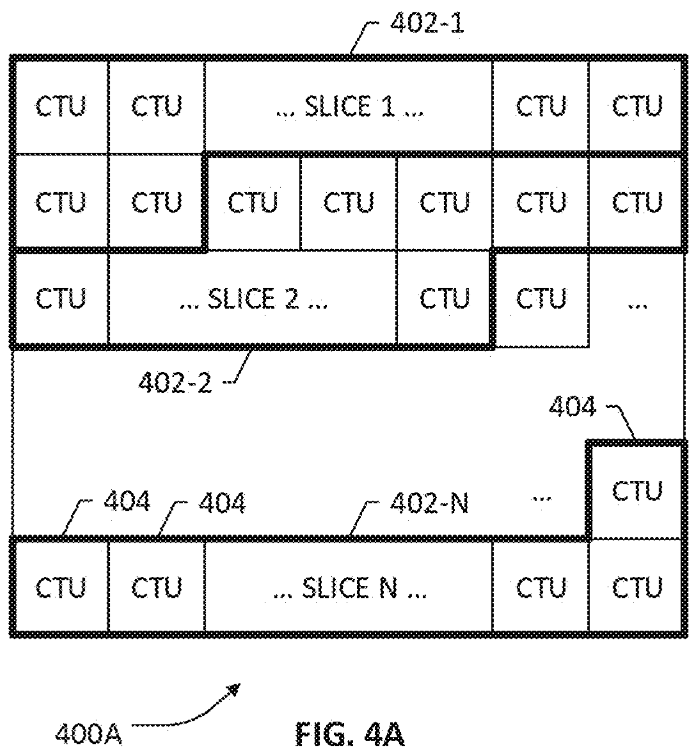

402-1
CTU
CTU
... SLICE 1 ...
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
CTU
... SLICE 2 ...
CTU
CTU
...
402-2
404
404
404
402-N
...
CTU
CTU
CTU
... SLICE N ...
CTU
CTU
400A

FIG. 4A

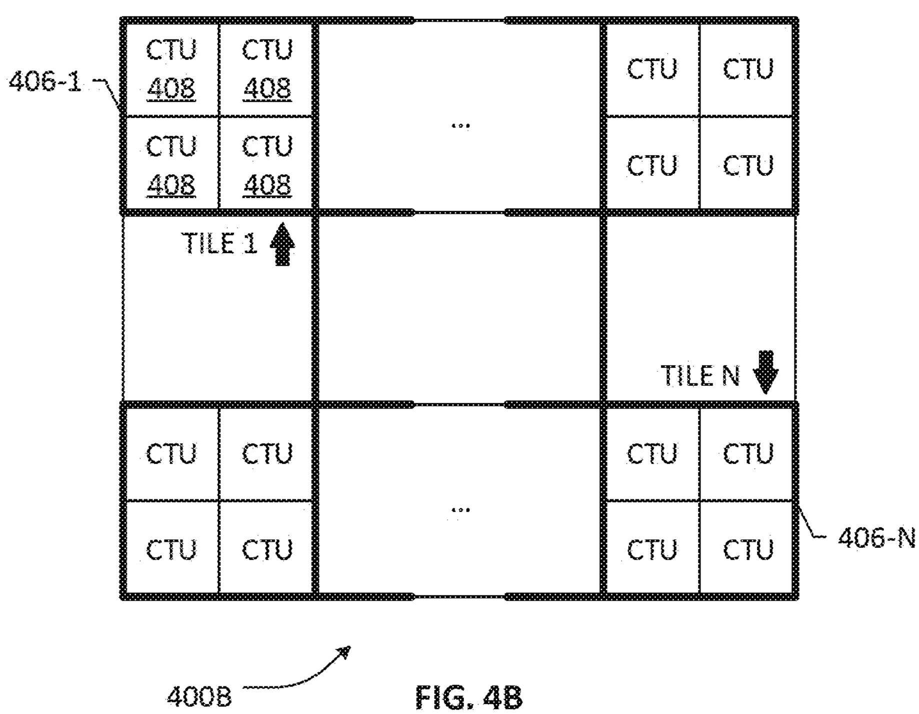

406-1
CTU 408
CTU 408
CTU 408
CTU 408
...
CTU
CTU
CTU
CTU
TILE 1
TILE N
CTU
CTU
CTU
CTU
...
CTU
CTU
CTU
CTU
406-N
400B

FIG. 4B

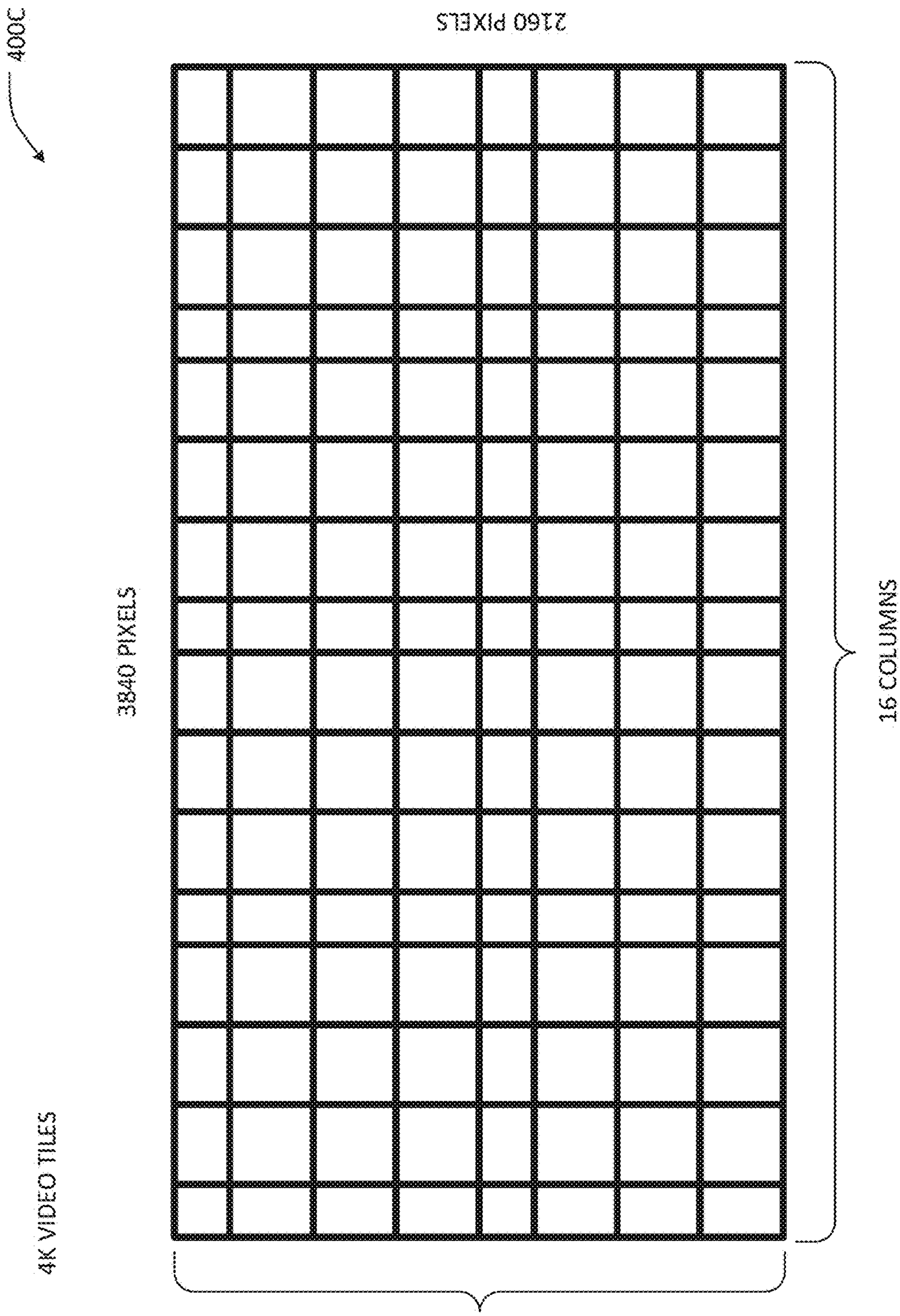

400C
2160 PIXELS
3840 PIXELS
16 COLUMNS
8 ROWS
4K VIDEO TILES

FIG. 4C

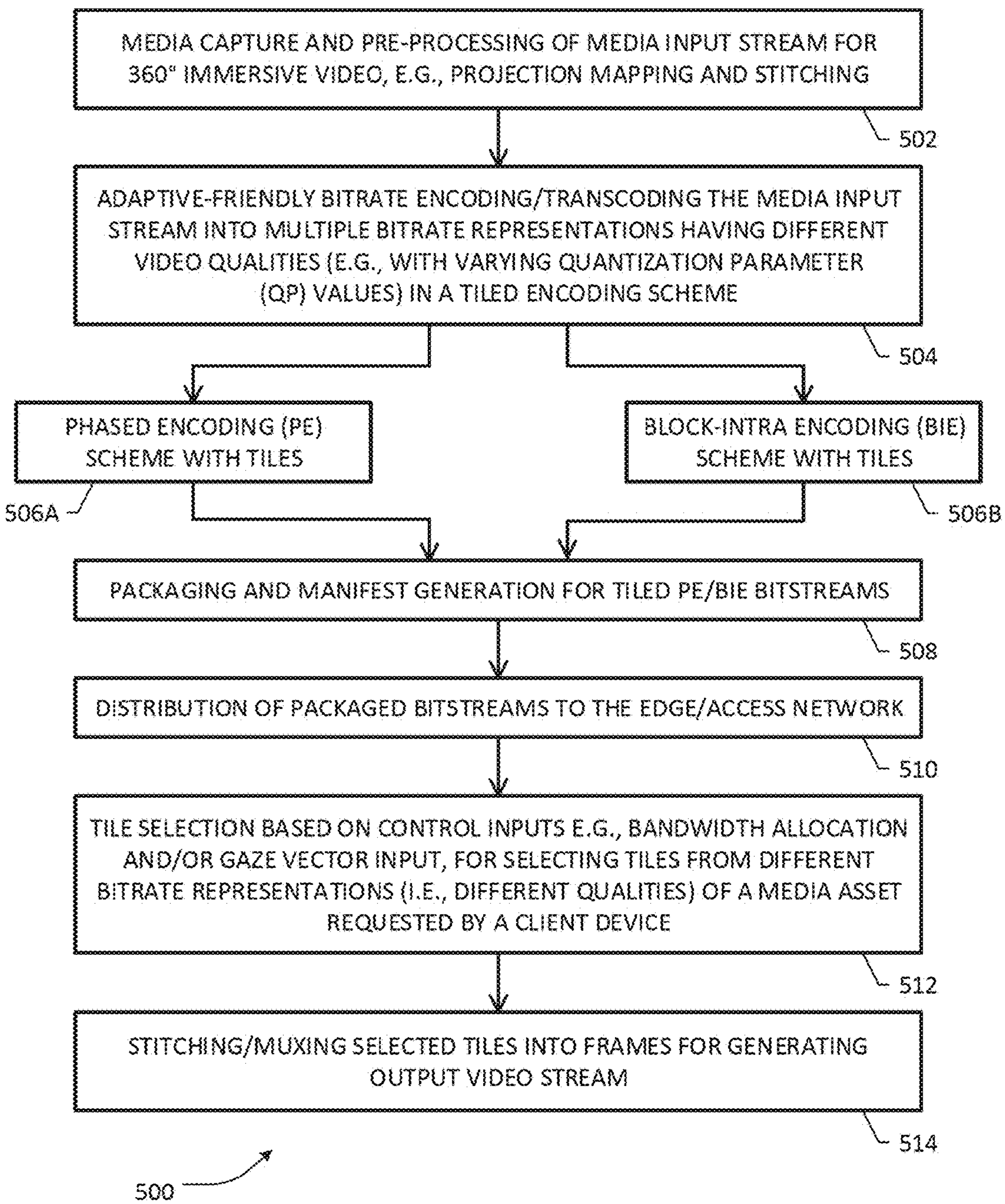

MEDIA CAPTURE AND PRE-PROCESSING OF MEDIA INPUT STREAM FOR 360° IMMERSIVE VIDEO, E.G., PROJECTION MAPPING AND STITCHING
502
ADAPTIVE-FRIENDLY BITRATE ENCODING/TRANSCODING THE MEDIA INPUT STREAM INTO MULTIPLE BITRATE REPRESENTATIONS HAVING DIFFERENT VIDEO QUALITIES (E.G., WITH VARYING QUANTIZATION PARAMETER (QP) VALUES) IN A TILED ENCODING SCHEME
504
PHASED ENCODING (PE) SCHEME WITH TILES
506A
BLOCK-INTRA ENCODING (BIE) SCHEME WITH TILES
506B
PACKAGING AND MANIFEST GENERATION FOR TILED PE/BIE BITSTREAMS
508
DISTRIBUTION OF PACKAGED BITSTREAMS TO THE EDGE/ACCESS NETWORK
510
TILE SELECTION BASED ON CONTROL INPUTS E.G., BANDWIDTH ALLOCATION AND/OR GAZE VECTOR INPUT, FOR SELECTING TILES FROM DIFFERENT BITRATE REPRESENTATIONS (I.E., DIFFERENT QUALITIES) OF A MEDIA ASSET REQUESTED BY A CLIENT DEVICE
512
STITCHING/MUXING SELECTED TILES INTO FRAMES FOR GENERATING OUTPUT VIDEO STREAM
514
500

FIG. 5

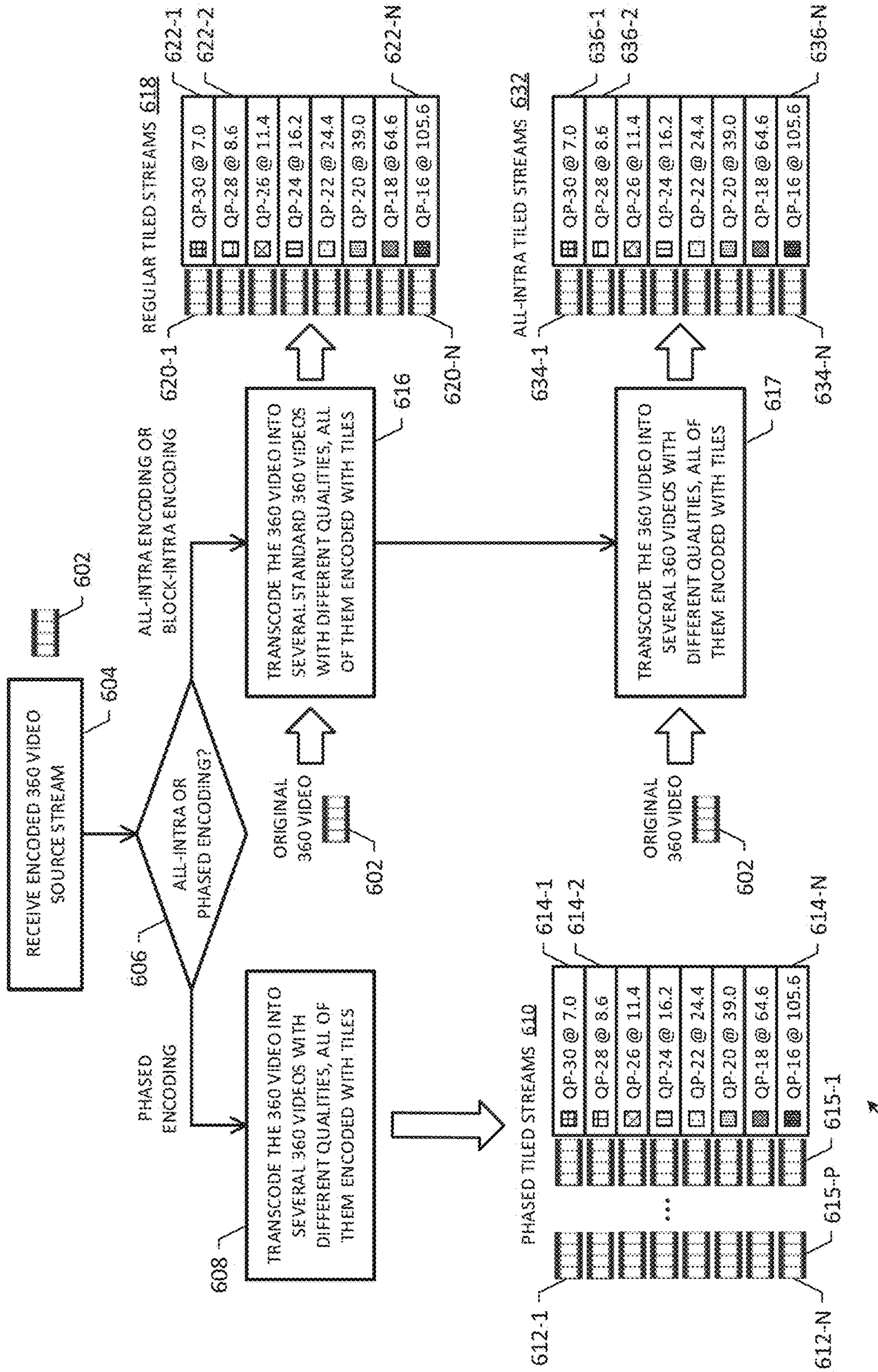

RECEIVE ENCODED 360 VIDEO SOURCE STREAM
604
602
ALL-INTRA OR PHASED ENCODING?
606
PHASED ENCODING
ALL-INTRA ENCODING OR BLOCK-INTRA ENCODING
TRANSCODE THE 360 VIDEO INTO SEVERAL 360 VIDEOS WITH DIFFERENT QUALITIES, ALL OF THEM ENCODED WITH TILES
608
TRANSCODE THE 360 VIDEO INTO SEVERAL STANDARD 360 VIDEOS WITH DIFFERENT QUALITIES, ALL OF THEM ENCODED WITH TILES
616
TRANSCODE THE 360 VIDEO INTO SEVERAL 360 VIDEOS WITH DIFFERENT QUALITIES, ALL OF THEM ENCODED WITH TILES
617
ORIGINAL 360 VIDEO
602
ORIGINAL 360 VIDEO
602
REGULAR TILED STREAMS 618
620-1
620-N
622-1
622-2
622-N
ALL-INTRA TILED STREAMS 632
634-1
634-N
636-1
636-2
636-N
PHASED TILED STREAMS 610
612-1
612-N
614-1
614-2
614-N
615-1
615-P
QP-30 @ 7.0
QP-28 @ 8.6
QP-26 @ 11.4
QP-24 @ 16.2
QP-22 @ 24.4
QP-20 @ 39.0
QP-18 @ 64.6
QP-16 @ 105.6
600

FIG. 6

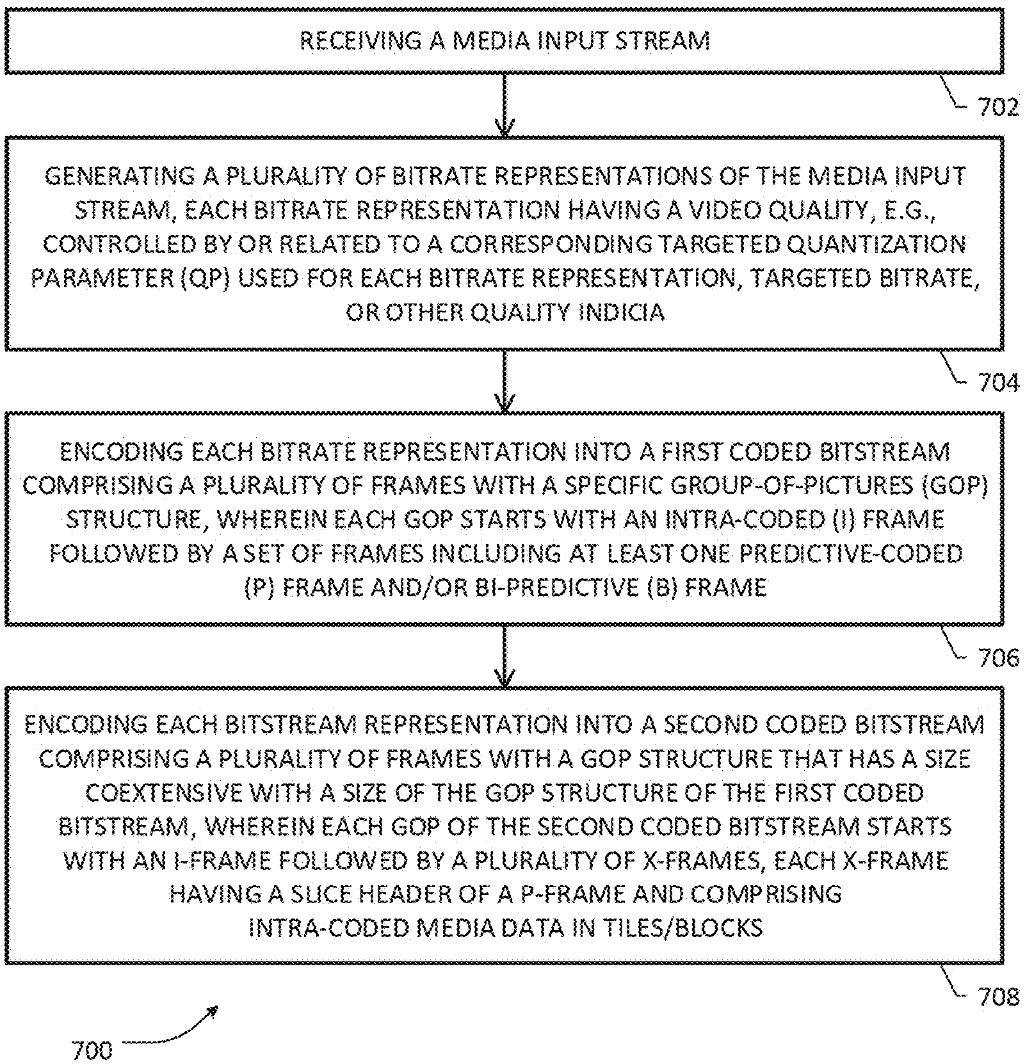

RECEIVING A MEDIA INPUT STREAM
702
GENERATING A PLURALITY OF BITRATE REPRESENTATIONS OF THE MEDIA INPUT STREAM, EACH BITRATE REPRESENTATION HAVING A VIDEO QUALITY, E.G., CONTROLLED BY OR RELATED TO A CORRESPONDING TARGETED QUANTIZATION PARAMETER (QP) USED FOR EACH BITRATE REPRESENTATION, TARGETED BITRATE, OR OTHER QUALITY INDICIA
704
ENCODING EACH BITRATE REPRESENTATION INTO A FIRST CODED BITSTREAM COMPRISING A PLURALITY OF FRAMES WITH A SPECIFIC GROUP-OF-PICTURES (GOP) STRUCTURE, WHEREIN EACH GOP STARTS WITH AN INTRA-CODED (I) FRAME FOLLOWED BY A SET OF FRAMES INCLUDING AT LEAST ONE PREDICTIVE-CODED (P) FRAME AND/OR BI-PREDICTIVE (B) FRAME
706
ENCODING EACH BITSTREAM REPRESENTATION INTO A SECOND CODED BITSTREAM COMPRISING A PLURALITY OF FRAMES WITH A GOP STRUCTURE THAT HAS A SIZE COEXTENSIVE WITH A SIZE OF THE GOP STRUCTURE OF THE FIRST CODED BITSTREAM, WHEREIN EACH GOP OF THE SECOND CODED BITSTREAM STARTS WITH AN I-FRAME FOLLOWED BY A PLURALITY OF X-FRAMES, EACH X-FRAME HAVING A SLICE HEADER OF A P-FRAME AND COMPRISING INTRA-CODED MEDIA DATA IN TILES/BLOCKS
708
700

FIG. 7

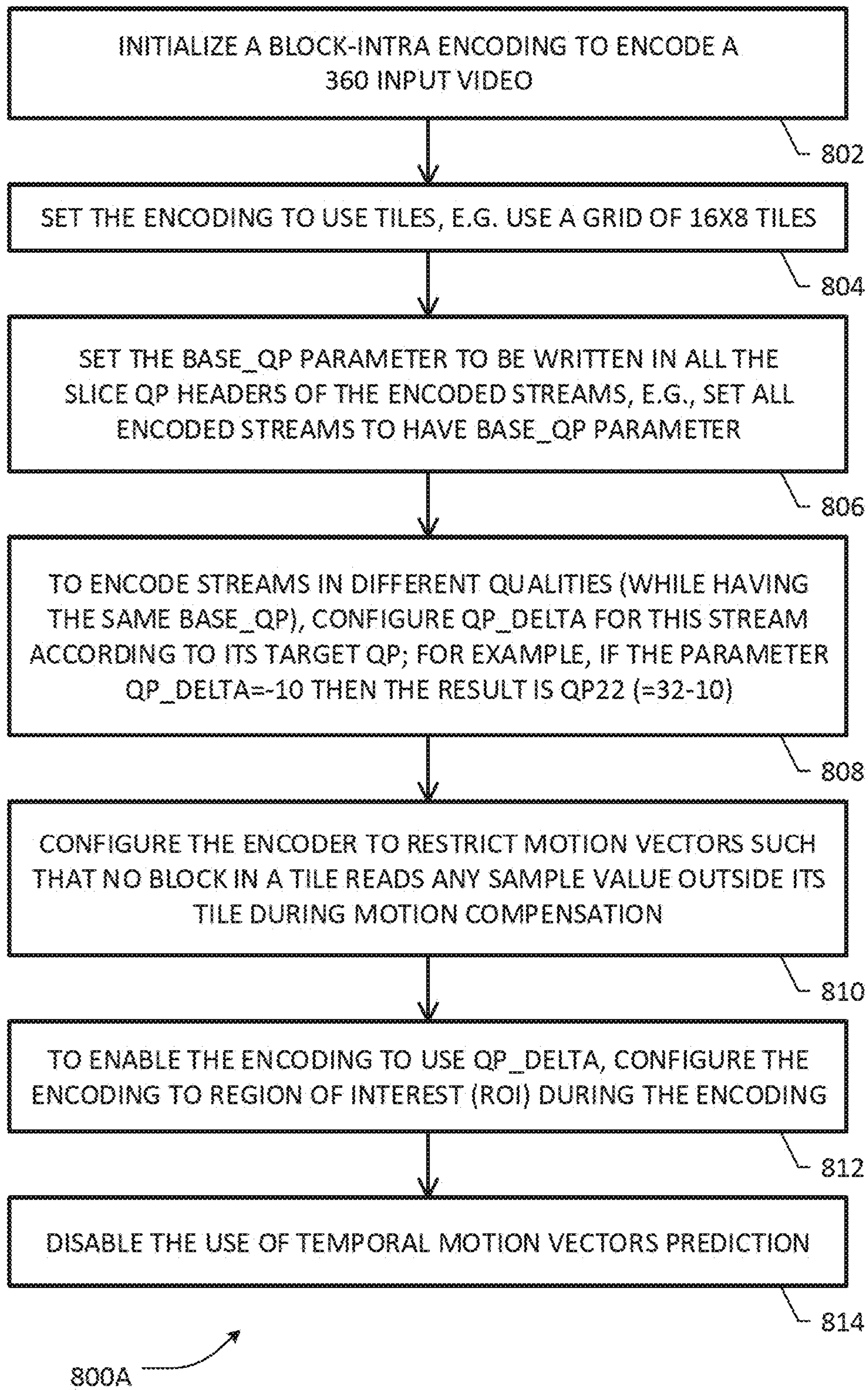

INITIALIZE A BLOCK-INTRA ENCODING TO ENCODE A 360 INPUT VIDEO
802
SET THE ENCODING TO USE TILES, E.G. USE A GRID OF 16X8 TILES
804
SET THE BASE_QP PARAMETER TO BE WRITTEN IN ALL THE SLICE QP HEADERS OF THE ENCODED STREAMS, E.G., SET ALL ENCODED STREAMS TO HAVE BASE_QP PARAMETER
806
TO ENCODE STREAMS IN DIFFERENT QUALITIES (WHILE HAVING THE SAME BASE_QP), CONFIGURE QP_DELTA FOR THIS STREAM ACCORDING TO ITS TARGET QP; FOR EXAMPLE, IF THE PARAMETER QP_DELTA=-10 THEN THE RESULT IS QP22 (=32-10)
808
CONFIGURE THE ENCODER TO RESTRICT MOTION VECTORS SUCH THAT NO BLOCK IN A TILE READS ANY SAMPLE VALUE OUTSIDE ITS TILE DURING MOTION COMPENSATION
810
TO ENABLE THE ENCODING TO USE QP_DELTA, CONFIGURE THE ENCODING TO REGION OF INTEREST (ROI) DURING THE ENCODING
812
DISABLE THE USE OF TEMPORAL MOTION VECTORS PREDICTION
814
800A

FIG. 8A

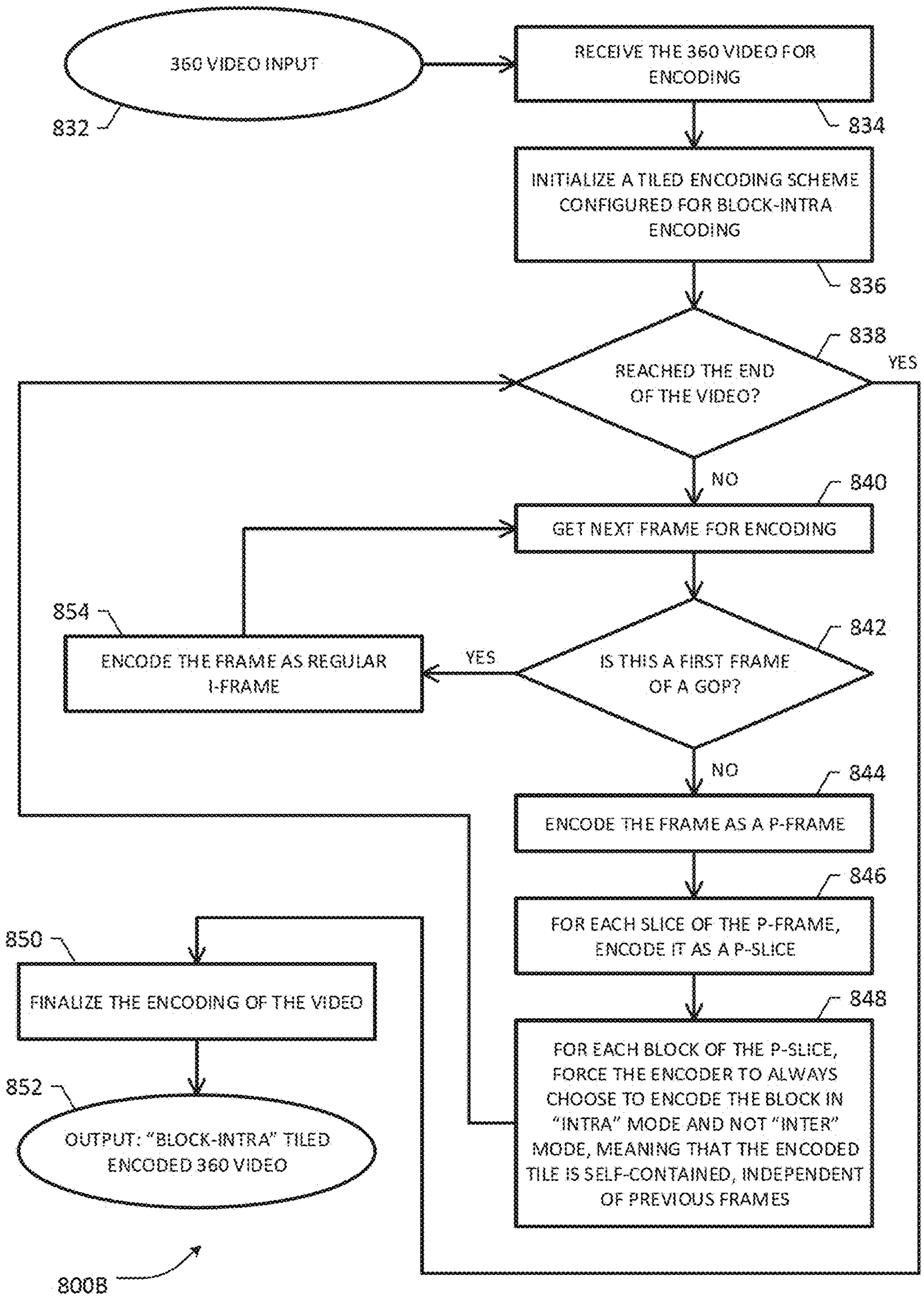

360 VIDEO INPUT
832
RECEIVE THE 360 VIDEO FOR ENCODING
834
INITIALIZE A TILED ENCODING SCHEME CONFIGURED FOR BLOCK-INTRA ENCODING
836
REACHED THE END OF THE VIDEO?
838
YES
NO
GET NEXT FRAME FOR ENCODING
840
854
ENCODE THE FRAME AS REGULAR I-FRAME
YES
IS THIS A FIRST FRAME OF A GOP?
842
NO
ENCODE THE FRAME AS A P-FRAME
844
FOR EACH SLICE OF THE P-FRAME, ENCODE IT AS A P-SLICE
846
850
FINALIZE THE ENCODING OF THE VIDEO
848
FOR EACH BLOCK OF THE P-SLICE, FORCE THE ENCODER TO ALWAYS CHOOSE TO ENCODE THE BLOCK IN "INTRA" MODE AND NOT "INTER" MODE, MEANING THAT THE ENCODED TILE IS SELF-CONTAINED, INDEPENDENT OF PREVIOUS FRAMES
852
OUTPUT: "BLOCK-INTRA" TILED ENCODED 360 VIDEO
800B

FIG. 8B

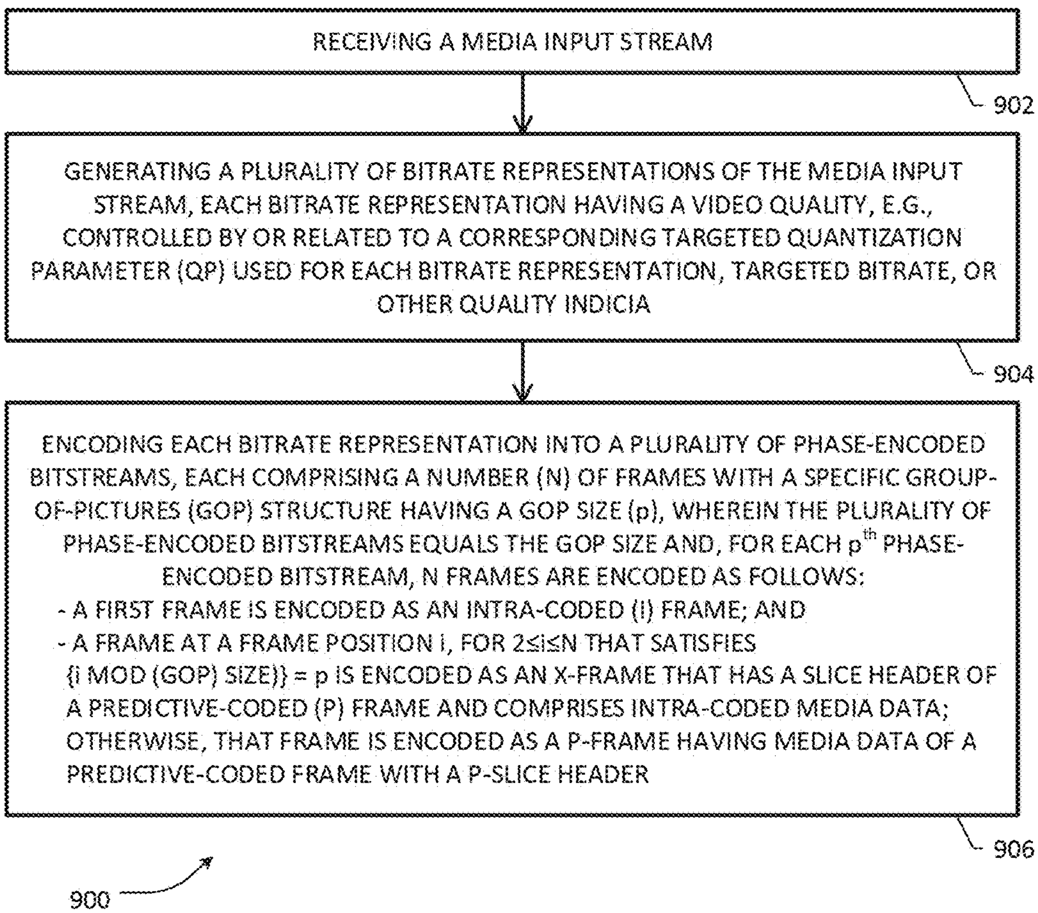

RECEIVING A MEDIA INPUT STREAM
902
GENERATING A PLURALITY OF BITRATE REPRESENTATIONS OF THE MEDIA INPUT STREAM, EACH BITRATE REPRESENTATION HAVING A VIDEO QUALITY, E.G., CONTROLLED BY OR RELATED TO A CORRESPONDING TARGETED QUANTIZATION PARAMETER (QP) USED FOR EACH BITRATE REPRESENTATION, TARGETED BITRATE, OR OTHER QUALITY INDICIA
904
ENCODING EACH BITRATE REPRESENTATION INTO A PLURALITY OF PHASE-ENCODED BITSTREAMS, EACH COMPRISING A NUMBER (N) OF FRAMES WITH A SPECIFIC GROUP-OF-PICTURES (GOP) STRUCTURE HAVING A GOP SIZE (p), WHEREIN THE PLURALITY OF PHASE-ENCODED BITSTREAMS EQUALS THE GOP SIZE AND, FOR EACH p^{th} PHASE-ENCODED BITSTREAM, N FRAMES ARE ENCODED AS FOLLOWS:
- A FIRST FRAME IS ENCODED AS AN INTRA-CODED (I) FRAME; AND
- A FRAME AT A FRAME POSITION i, FOR $2 \leq i \leq N$ THAT SATISFIES {i MOD (GOP) SIZE)} = p IS ENCODED AS AN X-FRAME THAT HAS A SLICE HEADER OF A PREDICTIVE-CODED (P) FRAME AND COMPRISES INTRA-CODED MEDIA DATA; OTHERWISE, THAT FRAME IS ENCODED AS A P-FRAME HAVING MEDIA DATA OF A PREDICTIVE-CODED FRAME WITH A P-SLICE HEADER
906
900

FIG. 9

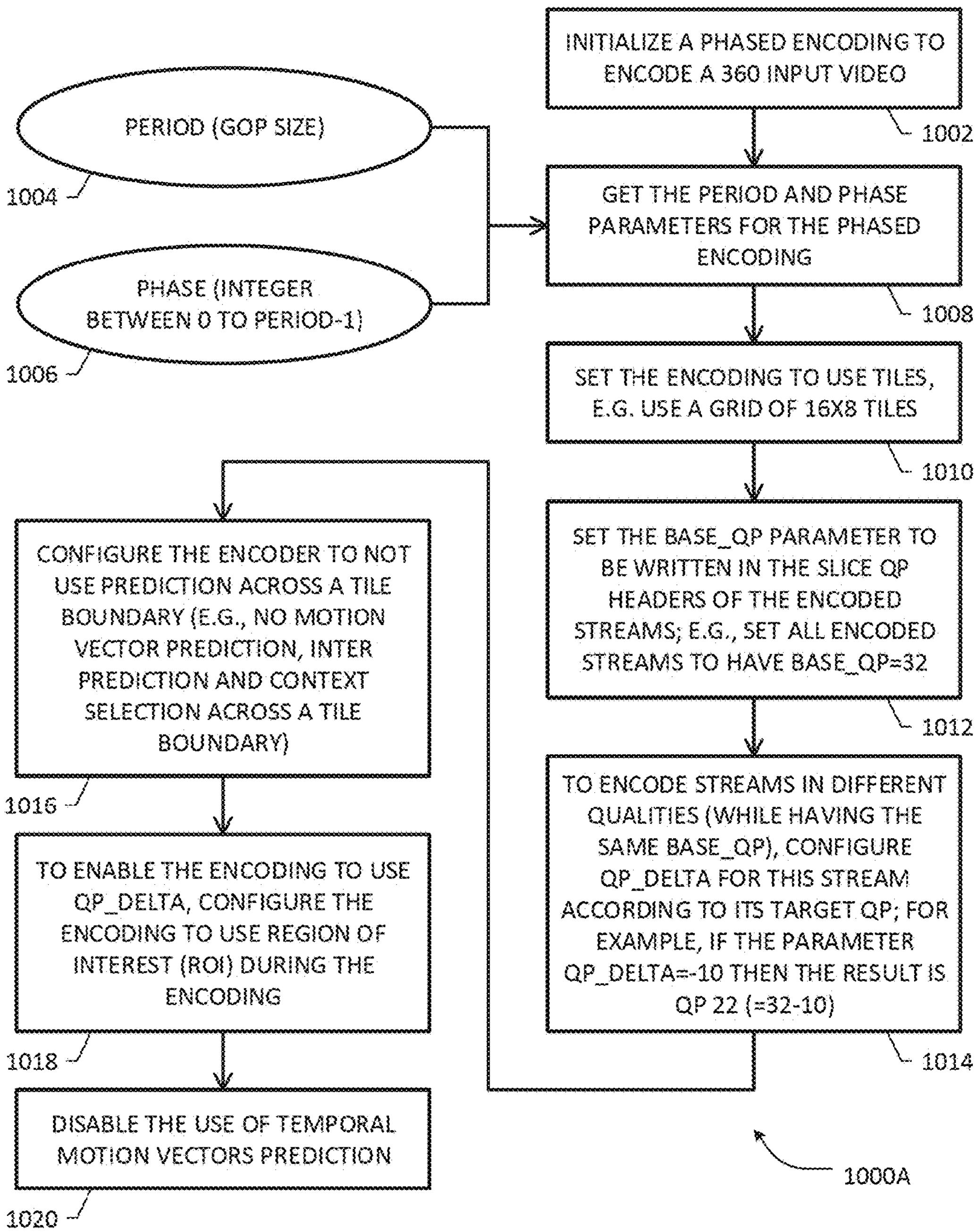

INITIALIZE A PHASED ENCODING TO ENCODE A 360 INPUT VIDEO
1002
PERIOD (GOP SIZE)
1004
PHASE (INTEGER BETWEEN 0 TO PERIOD-1)
1006
GET THE PERIOD AND PHASE PARAMETERS FOR THE PHASED ENCODING
1008
SET THE ENCODING TO USE TILES, E.G. USE A GRID OF 16X8 TILES
1010
SET THE BASE_QP PARAMETER TO BE WRITTEN IN THE SLICE QP HEADERS OF THE ENCODED STREAMS; E.G., SET ALL ENCODED STREAMS TO HAVE BASE_QP=32
1012
TO ENCODE STREAMS IN DIFFERENT QUALITIES (WHILE HAVING THE SAME BASE_QP), CONFIGURE QP_DELTA FOR THIS STREAM ACCORDING TO ITS TARGET QP; FOR EXAMPLE, IF THE PARAMETER QP_DELTA=-10 THEN THE RESULT IS QP 22 (=32-10)
1014
CONFIGURE THE ENCODER TO NOT USE PREDICTION ACROSS A TILE BOUNDARY (E.G., NO MOTION VECTOR PREDICTION, INTER PREDICTION AND CONTEXT SELECTION ACROSS A TILE BOUNDARY)
1016
TO ENABLE THE ENCODING TO USE QP_DELTA, CONFIGURE THE ENCODING TO USE REGION OF INTEREST (ROI) DURING THE ENCODING
1018
DISABLE THE USE OF TEMPORAL MOTION VECTORS PREDICTION
1020
1000A

FIG. 10A

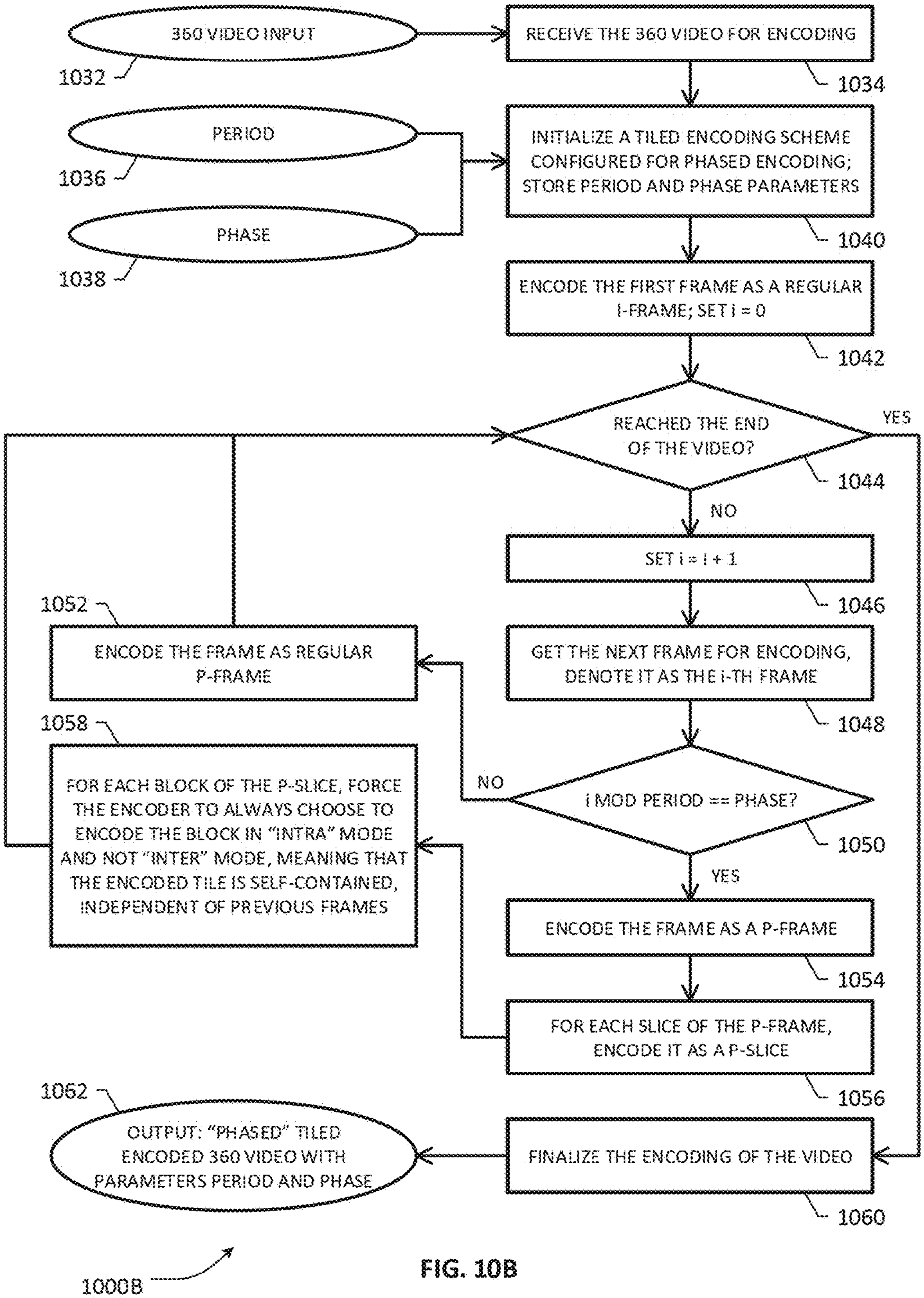

360 VIDEO INPUT
1032
RECEIVE THE 360 VIDEO FOR ENCODING
1034
PERIOD
1036
PHASE
1038
INITIALIZE A TILED ENCODING SCHEME CONFIGURED FOR PHASED ENCODING; STORE PERIOD AND PHASE PARAMETERS
1040
ENCODE THE FIRST FRAME AS A REGULAR I-FRAME; SET i = 0
1042
REACHED THE END OF THE VIDEO?
1044
YES
NO
SET i = i + 1
1046
GET THE NEXT FRAME FOR ENCODING, DENOTE IT AS THE i-TH FRAME
1048
1052
ENCODE THE FRAME AS REGULAR P-FRAME
1058
FOR EACH BLOCK OF THE P-SLICE, FORCE THE ENCODER TO ALWAYS CHOOSE TO ENCODE THE BLOCK IN "INTRA" MODE AND NOT "INTER" MODE, MEANING THAT THE ENCODED TILE IS SELF-CONTAINED, INDEPENDENT OF PREVIOUS FRAMES
NO
i MOD PERIOD == PHASE?
1050
YES
ENCODE THE FRAME AS A P-FRAME
1054
FOR EACH SLICE OF THE P-FRAME, ENCODE IT AS A P-SLICE
1056
1062
OUTPUT: "PHASED" TILED ENCODED 360 VIDEO WITH PARAMETERS PERIOD AND PHASE
FINALIZE THE ENCODING OF THE VIDEO
1060
1000B

FIG. 10B

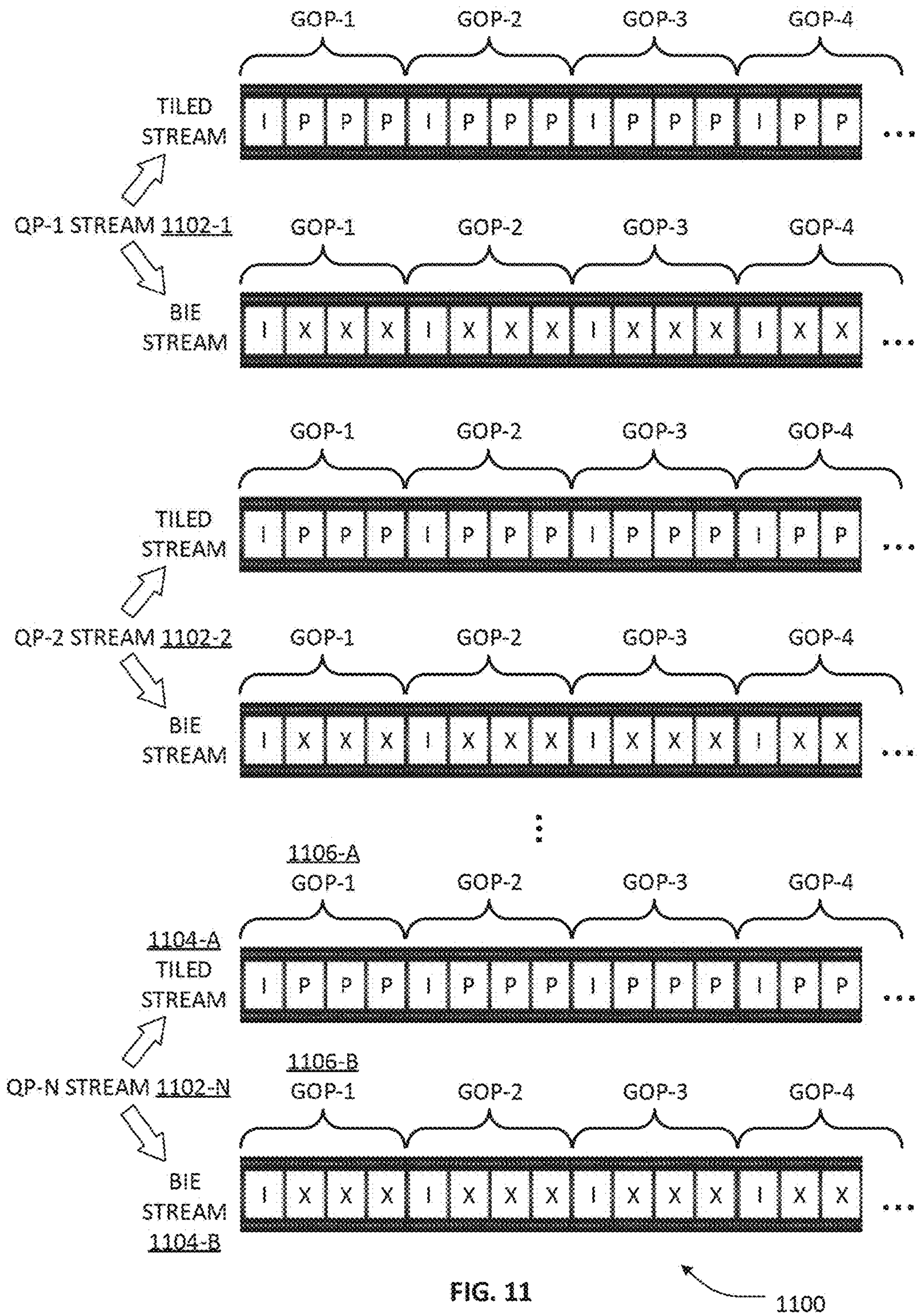

GOP-1
GOP-2
GOP-3
GOP-4
TILED STREAM
I P P P I P P P I P P P I P P ...
QP-1 STREAM 1102-1
BIE STREAM
I X X X I X X X I X X X I X X ...
QP-2 STREAM 1102-2
1106-A
1104-A
QP-N STREAM 1102-N
1106-B
1104-B
1100

FIG. 11

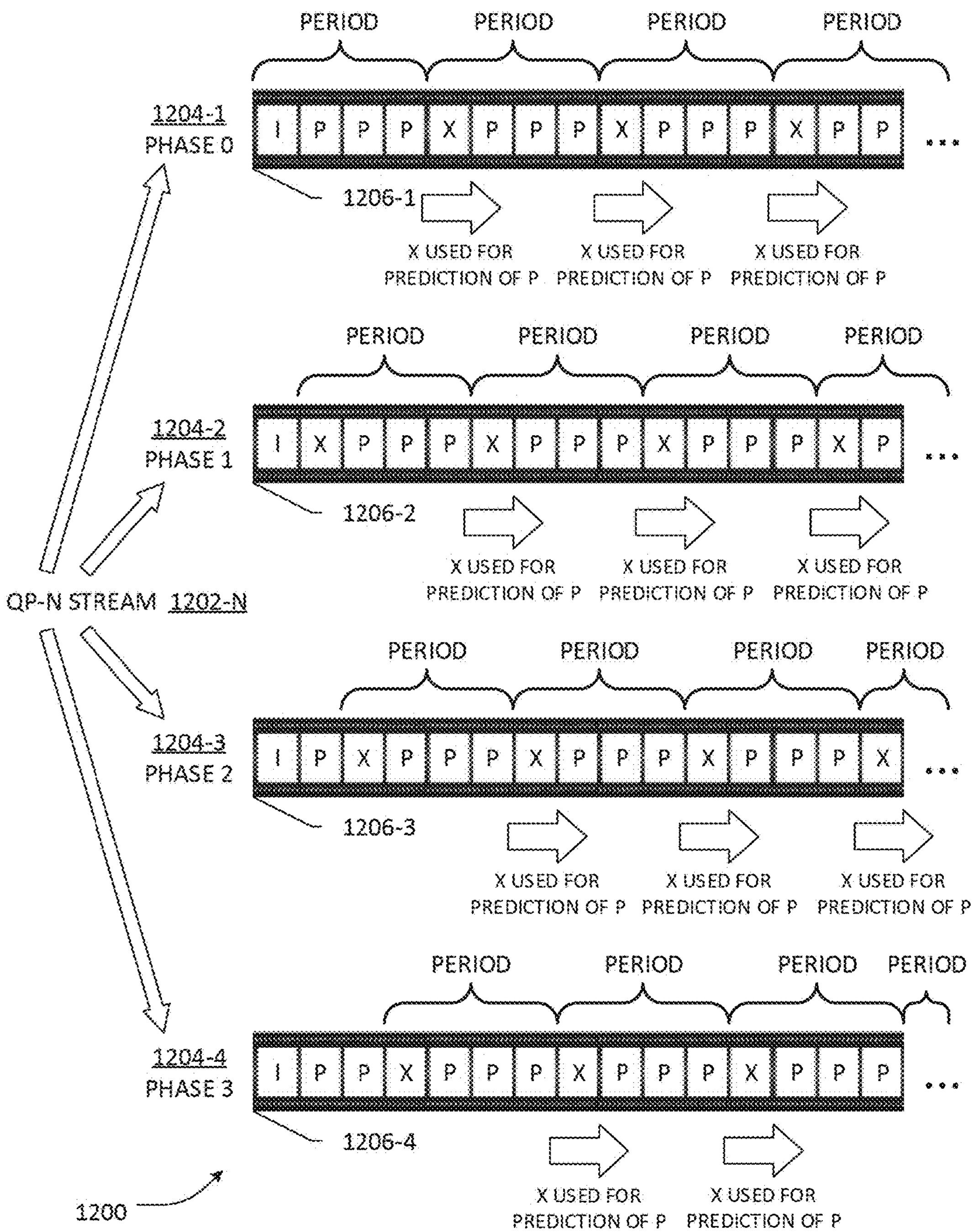

PERIOD
PERIOD
PERIOD
PERIOD
1204-1
PHASE 0
I P P P X P P P X P P P X P P ...
1206-1
X USED FOR PREDICTION OF P
X USED FOR PREDICTION OF P
X USED FOR PREDICTION OF P
1204-2
PHASE 1
I X P P P X P P P X P P P X P ...
1206-2
QP-N STREAM 1202-N
1204-3
PHASE 2
I P X P P P X P P P X P P P X ...
1206-3
1204-4
PHASE 3
I P P X P P P X P P P X P P P ...
1206-4
1200

FIG. 12

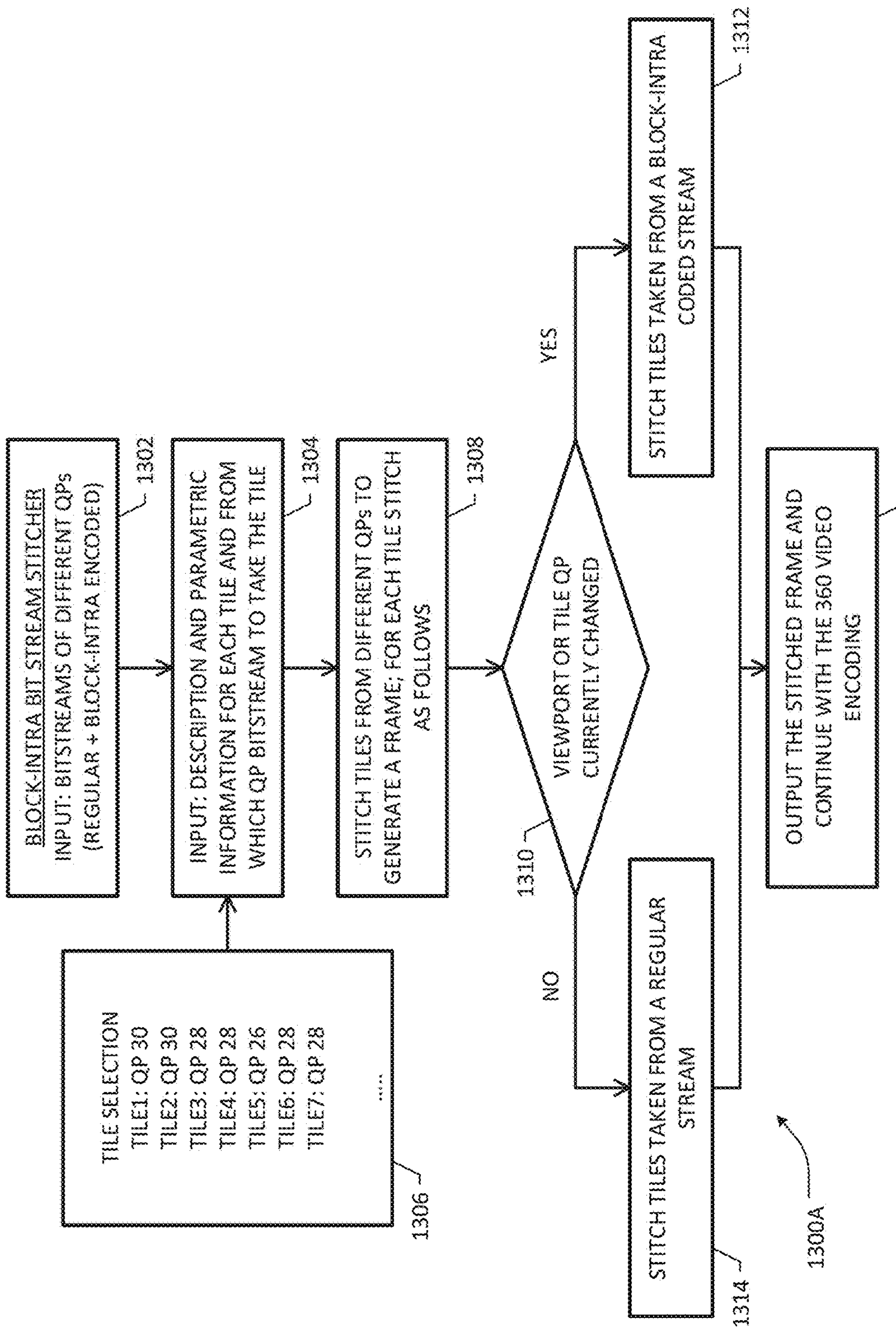

BLOCK-INTRA BIT STREAM STITCHER
INPUT: BITSTREAMS OF DIFFERENT QPs (REGULAR + BLOCK-INTRA ENCODED)
1302
INPUT: DESCRIPTION AND PARAMETRIC INFORMATION FOR EACH TILE AND FROM WHICH QP BITSTREAM TO TAKE THE TILE
1304
TILE SELECTION
TILE1: QP 30
TILE2: QP 30
TILE3: QP 28
TILE4: QP 28
TILE5: QP 26
TILE6: QP 28
TILE7: QP 28
.....
1306
STITCH TILES FROM DIFFERENT QPs TO GENERATE A FRAME; FOR EACH TILE STITCH AS FOLLOWS
1308
VIEWPORT OR TILE QP CURRENTLY CHANGED
1310
YES
NO
STITCH TILES TAKEN FROM A BLOCK-INTRA CODED STREAM
1312
STITCH TILES TAKEN FROM A REGULAR STREAM
1314
OUTPUT THE STITCHED FRAME AND CONTINUE WITH THE 360 VIDEO ENCODING
1316
1300A

FIG. 13A

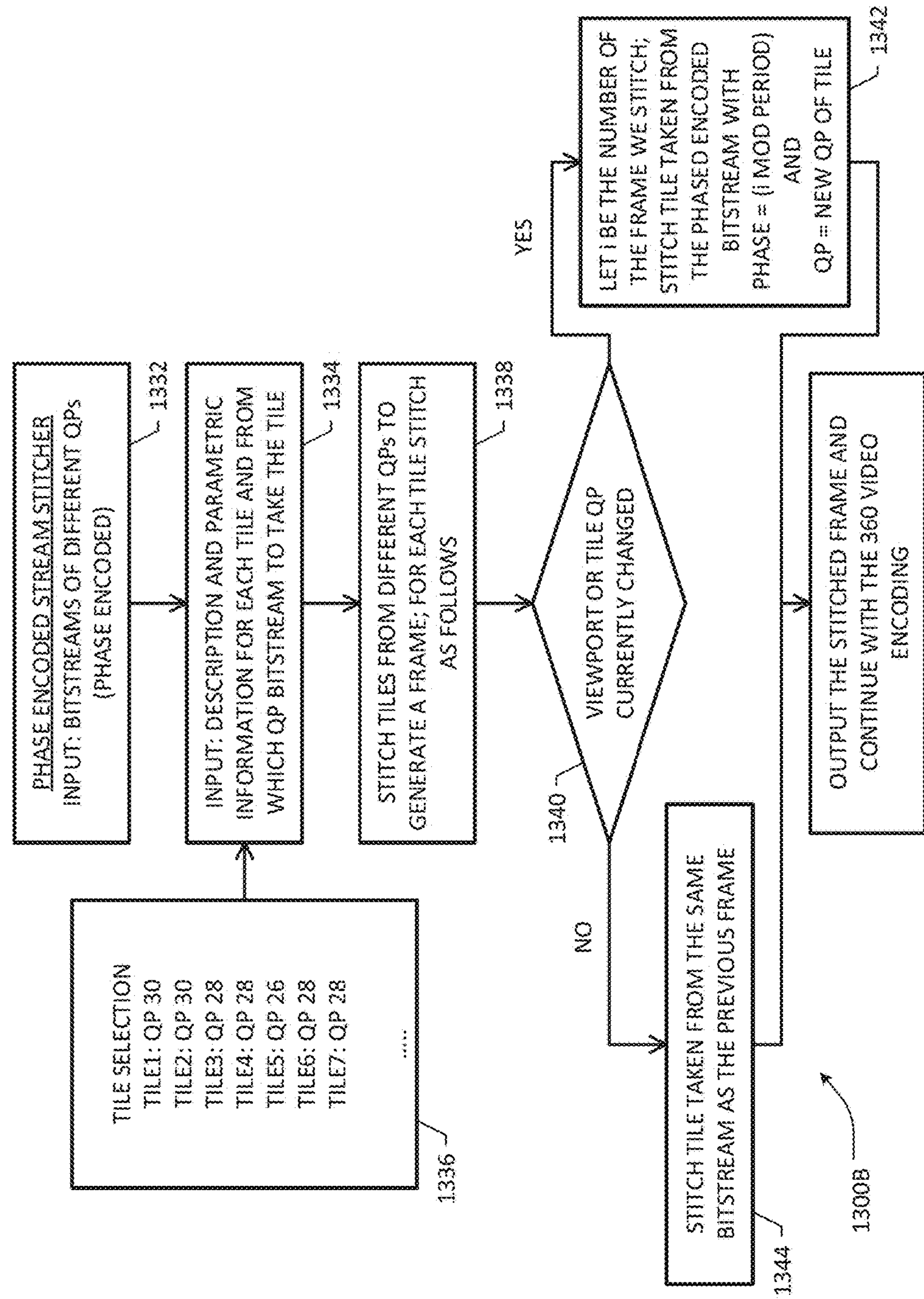

PHASE ENCODED STREAM STITCHER
INPUT: BITSTREAMS OF DIFFERENT QPs (PHASE ENCODED)
1332
INPUT: DESCRIPTION AND PARAMETRIC INFORMATION FOR EACH TILE AND FROM WHICH QP BITSTREAM TO TAKE THE TILE
1334
TILE SELECTION
TILE1: QP 30
TILE2: QP 30
TILE3: QP 28
TILE4: QP 28
TILE5: QP 26
TILE6: QP 28
TILE7: QP 28
.....
1336
STITCH TILES FROM DIFFERENT QPs TO GENERATE A FRAME; FOR EACH TILE STITCH AS FOLLOWS
1338
VIEWPORT OR TILE QP CURRENTLY CHANGED
1340
YES
LET i BE THE NUMBER OF THE FRAME WE STITCH; STITCH TILE TAKEN FROM THE PHASED ENCODED BITSTREAM WITH PHASE = (i MOD PERIOD) AND QP = NEW QP OF TILE
1342
NO
STITCH TILE TAKEN FROM THE SAME BITSTREAM AS THE PREVIOUS FRAME
1344
OUTPUT THE STITCHED FRAME AND CONTINUE WITH THE 360 VIDEO ENCODING
1346
1300B

FIG. 13B

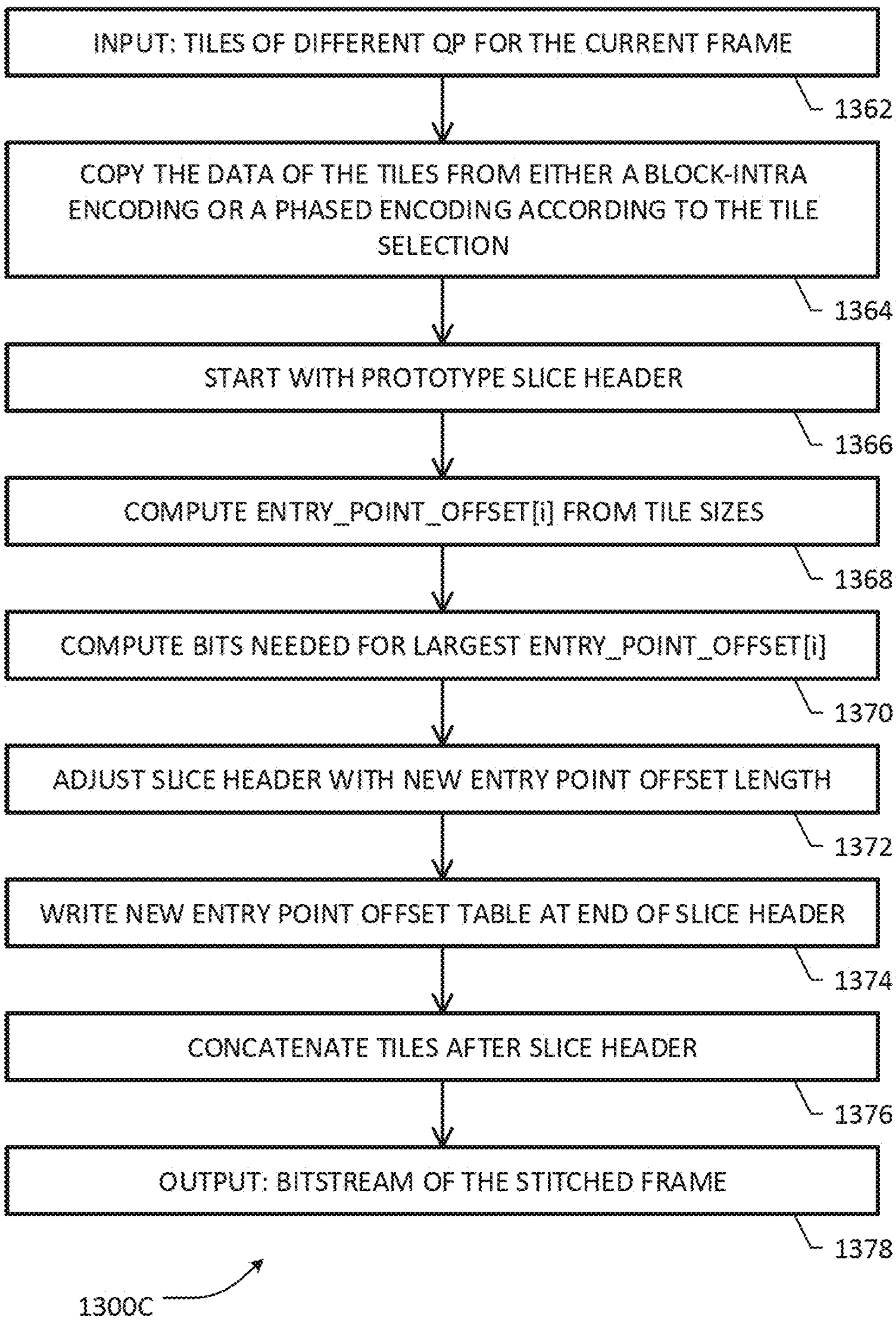

INPUT: TILES OF DIFFERENT QP FOR THE CURRENT FRAME
1362
COPY THE DATA OF THE TILES FROM EITHER A BLOCK-INTRA ENCODING OR A PHASED ENCODING ACCORDING TO THE TILE SELECTION
1364
START WITH PROTOTYPE SLICE HEADER
1366
COMPUTE ENTRY_POINT_OFFSET[i] FROM TILE SIZES
1368
COMPUTE BITS NEEDED FOR LARGEST ENTRY_POINT_OFFSET[i]
1370
ADJUST SLICE HEADER WITH NEW ENTRY POINT OFFSET LENGTH
1372
WRITE NEW ENTRY POINT OFFSET TABLE AT END OF SLICE HEADER
1374
CONCATENATE TILES AFTER SLICE HEADER
1376
OUTPUT: BITSTREAM OF THE STITCHED FRAME
1378
1300C

FIG. 13C

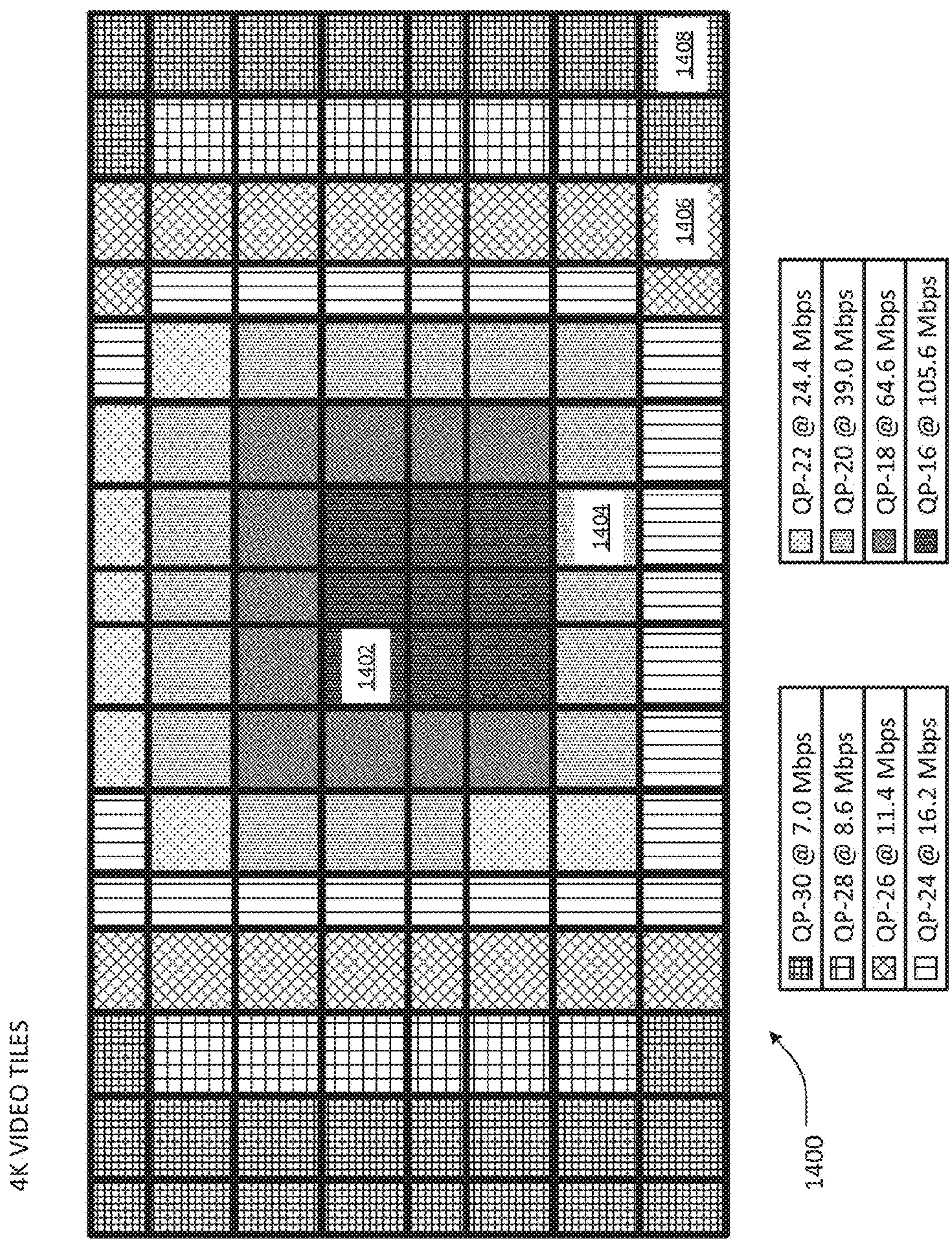

4K VIDEO TILES
1400
1402
1404
1406
1408
QP-30 @ 7.0 Mbps
QP-28 @ 8.6 Mbps
QP-26 @ 11.4 Mbps
QP-24 @ 16.2 Mbps
QP-22 @ 24.4 Mbps
QP-20 @ 39.0 Mbps
QP-18 @ 64.6 Mbps
QP-16 @ 105.6 Mbps

FIG. 14

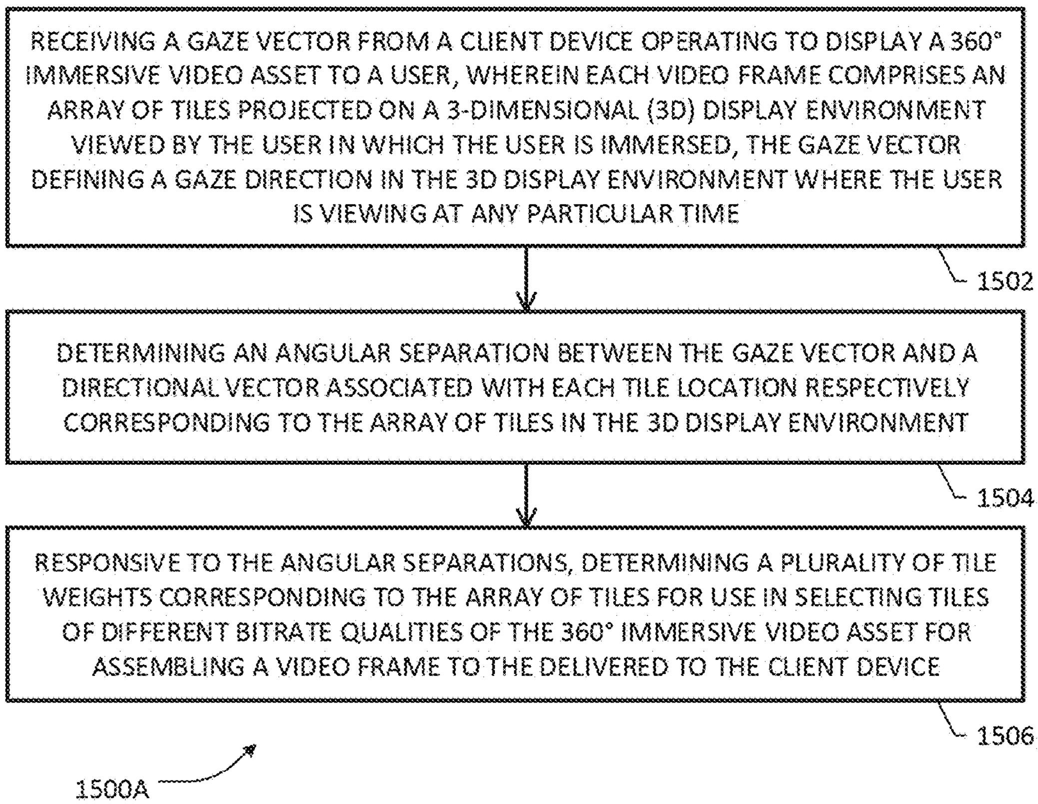


FIG. 15A

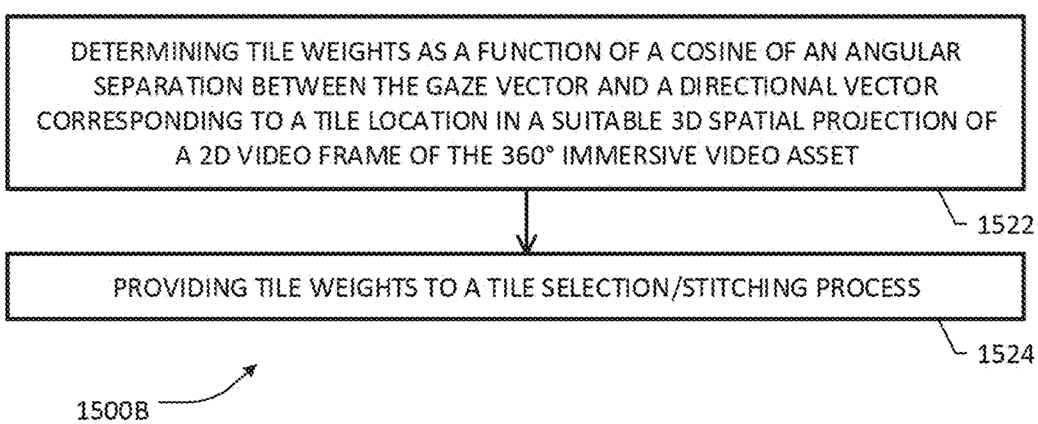


FIG. 15B

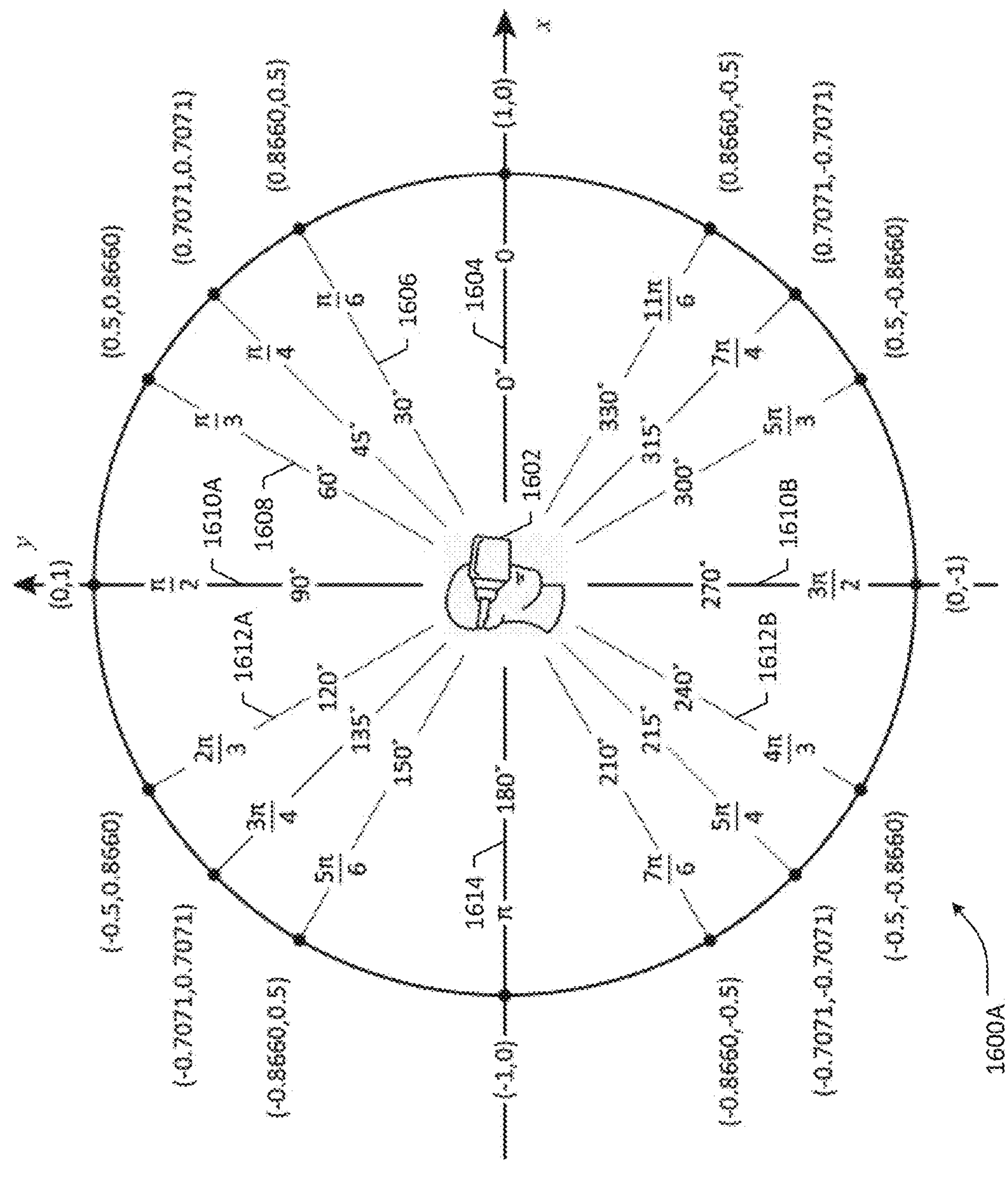

x
y
(1,0)
(0,1)
(-1,0)
(0,-1)
(0.8660,0.5)
(0.7071,0.7071)
(0.5,0.8660)
(-0.5,0.8660)
(-0.7071,0.7071)
(-0.8660,0.5)
(-0.8660,-0.5)
(-0.7071,-0.7071)
(-0.5,-0.8660)
(0.5,-0.8660)
(0.7071,-0.7071)
(0.8660,-0.5)
0
0°
30°
45°
60°
90°
120°
135°
150°
180°
210°
215°
240°
270°
300°
315°
330°
π
1602
1604
1606
1608
1610A
1610B
1612A
1612B
1614
1600A

FIG. 16A

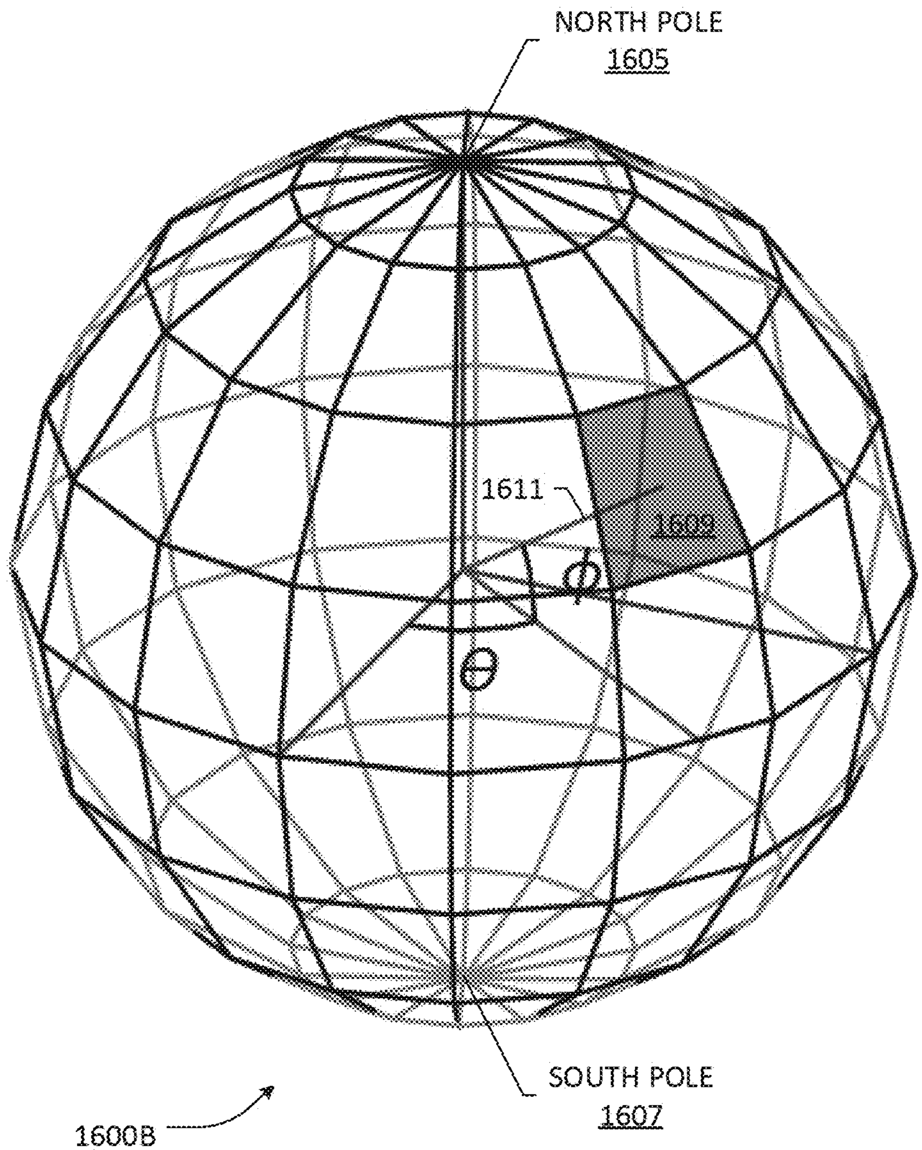


FIG. 16B

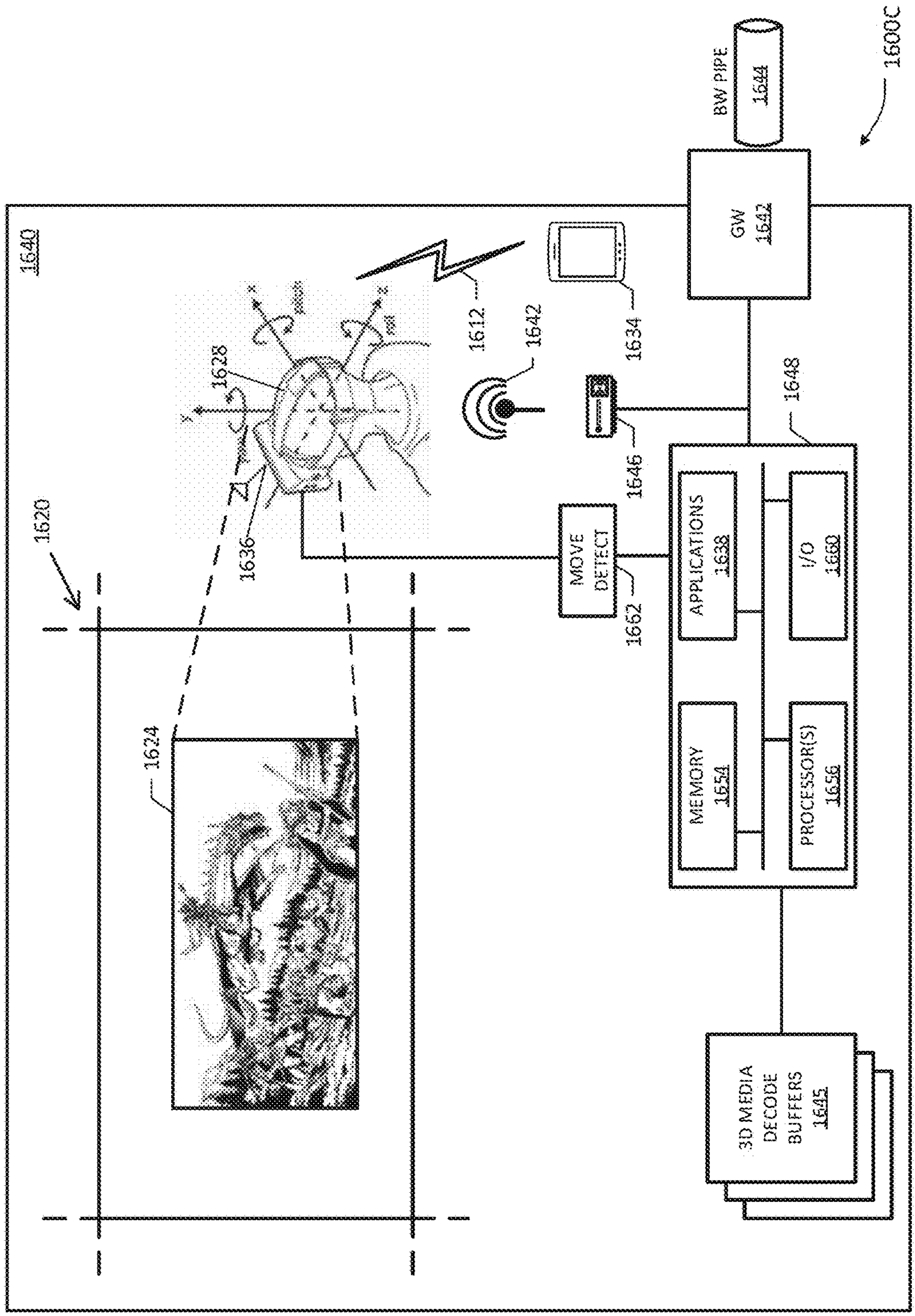

1600C
BW PIPE
1644
GW
1642
1640
1612
1642
1634
1628
1648
1646
1620
1636
MOVE
DETECT
1662
APPLICATIONS
1638
I/O
1660
MEMORY
1654
PROCESSOR(S)
1656
1624
3D MEDIA
DECODE
BUFFERS
1645

FIG. 16C

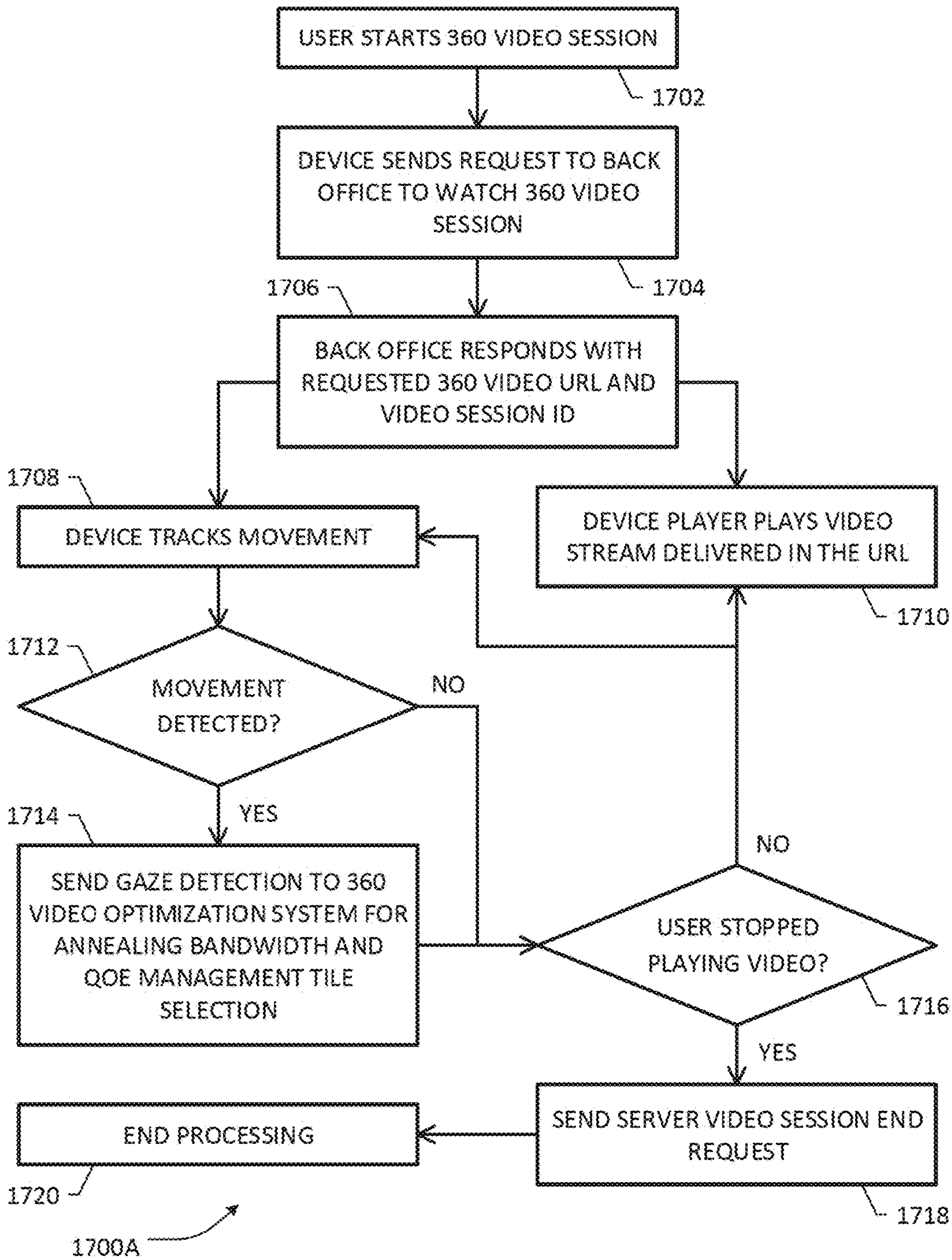

USER STARTS 360 VIDEO SESSION
1702
DEVICE SENDS REQUEST TO BACK OFFICE TO WATCH 360 VIDEO SESSION
1704
1706
BACK OFFICE RESPONDS WITH REQUESTED 360 VIDEO URL AND VIDEO SESSION ID
1708
DEVICE TRACKS MOVEMENT
DEVICE PLAYER PLAYS VIDEO STREAM DELIVERED IN THE URL
1710
1712
MOVEMENT DETECTED?
NO
YES
1714
SEND GAZE DETECTION TO 360 VIDEO OPTIMIZATION SYSTEM FOR ANNEALING BANDWIDTH AND QOE MANAGEMENT TILE SELECTION
NO
USER STOPPED PLAYING VIDEO?
1716
YES
END PROCESSING
SEND SERVER VIDEO SESSION END REQUEST
1720
1718
1700A

FIG. 17A

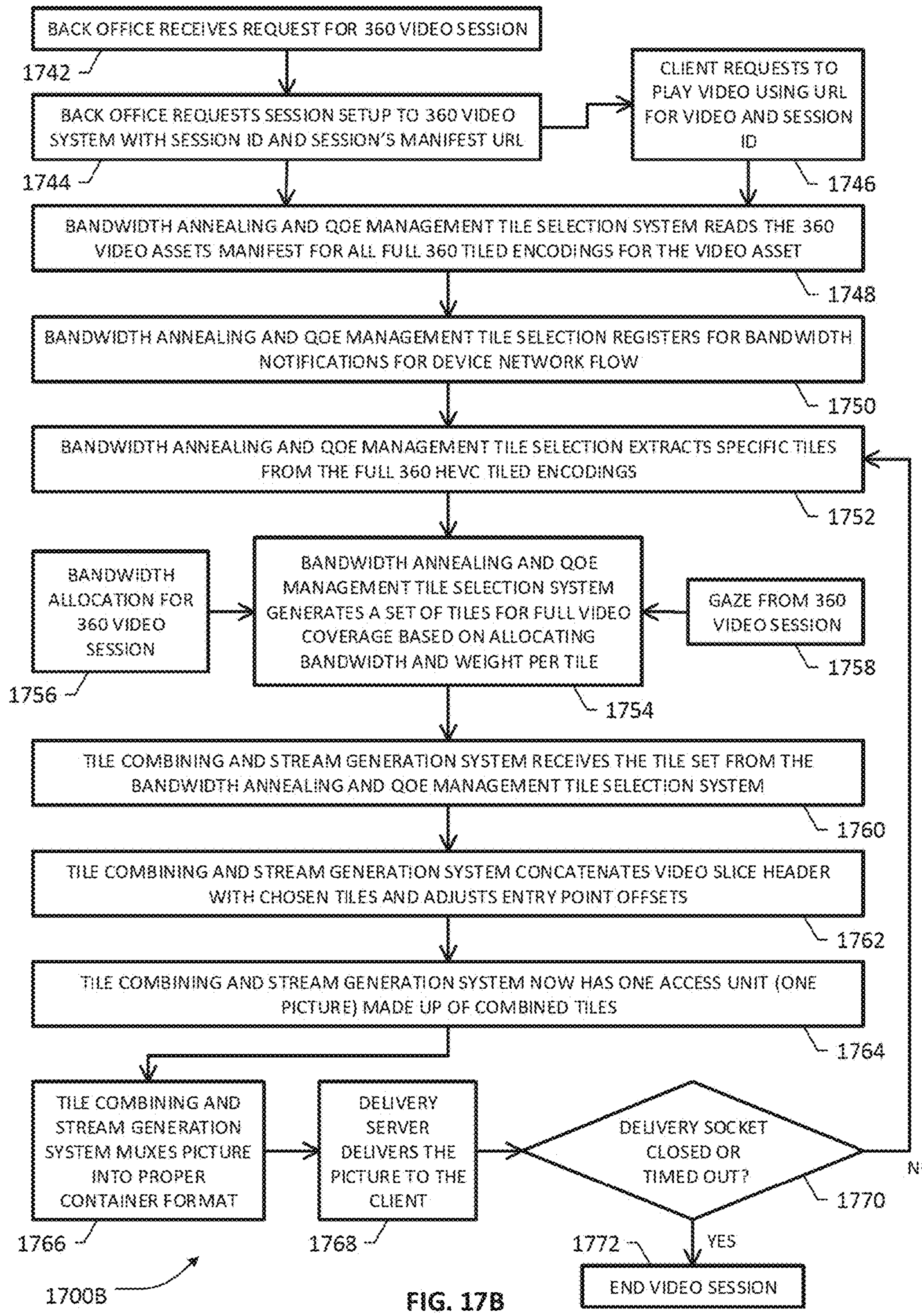

BACK OFFICE RECEIVES REQUEST FOR 360 VIDEO SESSION
1742
BACK OFFICE REQUESTS SESSION SETUP TO 360 VIDEO SYSTEM WITH SESSION ID AND SESSION'S MANIFEST URL
1744
CLIENT REQUESTS TO PLAY VIDEO USING URL FOR VIDEO AND SESSION ID
1746
BANDWIDTH ANNEALING AND QOE MANAGEMENT TILE SELECTION SYSTEM READS THE 360 VIDEO ASSETS MANIFEST FOR ALL FULL 360 TILED ENCODINGS FOR THE VIDEO ASSET
1748
BANDWIDTH ANNEALING AND QOE MANAGEMENT TILE SELECTION REGISTERS FOR BANDWIDTH NOTIFICATIONS FOR DEVICE NETWORK FLOW
1750
BANDWIDTH ANNEALING AND QOE MANAGEMENT TILE SELECTION EXTRACTS SPECIFIC TILES FROM THE FULL 360 HEVC TILED ENCODINGS
1752
BANDWIDTH ALLOCATION FOR 360 VIDEO SESSION
1756
BANDWIDTH ANNEALING AND QOE MANAGEMENT TILE SELECTION SYSTEM GENERATES A SET OF TILES FOR FULL VIDEO COVERAGE BASED ON ALLOCATING BANDWIDTH AND WEIGHT PER TILE
1754
GAZE FROM 360 VIDEO SESSION
1758
TILE COMBINING AND STREAM GENERATION SYSTEM RECEIVES THE TILE SET FROM THE BANDWIDTH ANNEALING AND QOE MANAGEMENT TILE SELECTION SYSTEM
1760
TILE COMBINING AND STREAM GENERATION SYSTEM CONCATENATES VIDEO SLICE HEADER WITH CHOSEN TILES AND ADJUSTS ENTRY POINT OFFSETS
1762
TILE COMBINING AND STREAM GENERATION SYSTEM NOW HAS ONE ACCESS UNIT (ONE PICTURE) MADE UP OF COMBINED TILES
1764
TILE COMBINING AND STREAM GENERATION SYSTEM MUXES PICTURE INTO PROPER CONTAINER FORMAT
1766
DELIVERY SERVER DELIVERS THE PICTURE TO THE CLIENT
1768
DELIVERY SOCKET CLOSED OR TIMED OUT?
NO
1770
YES
1772
END VIDEO SESSION
1700B

FIG. 17B

128 TILES

1802

1.27	1.23	1.21	1.22	1.26	1.31	1.38	1.45	1.51	1.55	1.57	1.56	1.52	1.46	1.39	1.33
0.10	0.09	0.08	0.09	0.10	1.12	1.31	1.51	1.68	1.79	1.84	1.81	1.70	1.54	1.35	1.15
0.07	0.05	0.05	0.05	0.07	0.09	1.19	1.48	1.74	1.91	1.98	1.94	1.78	1.54	1.25	0.10
0.05	0.03	0.02	0.02	0.04	0.07	1.04	1.39	1.69	1.90	1.98	1.92	1.74	1.45	1.11	0.08
0.03	0.01	0.00	0.01	0.03	0.05	0.09	1.24	1.53	1.74	1.82	1.77	1.58	1.30	0.10	0.06
0.03	0.01	0.00	0.01	0.02	0.05	0.08	1.05	1.30	1.47	1.54	1.50	1.34	1.10	0.08	0.05
0.03	0.02	0.02	0.02	0.03	0.05	0.07	0.08	1.02	1.13	1.18	1.15	1.05	0.09	0.07	0.05
0.05	0.04	0.04	0.04	0.05	0.05	0.06	0.07	0.07	0.08	0.08	0.08	0.07	0.07	0.06	0.05

1800A

FIG. 18A

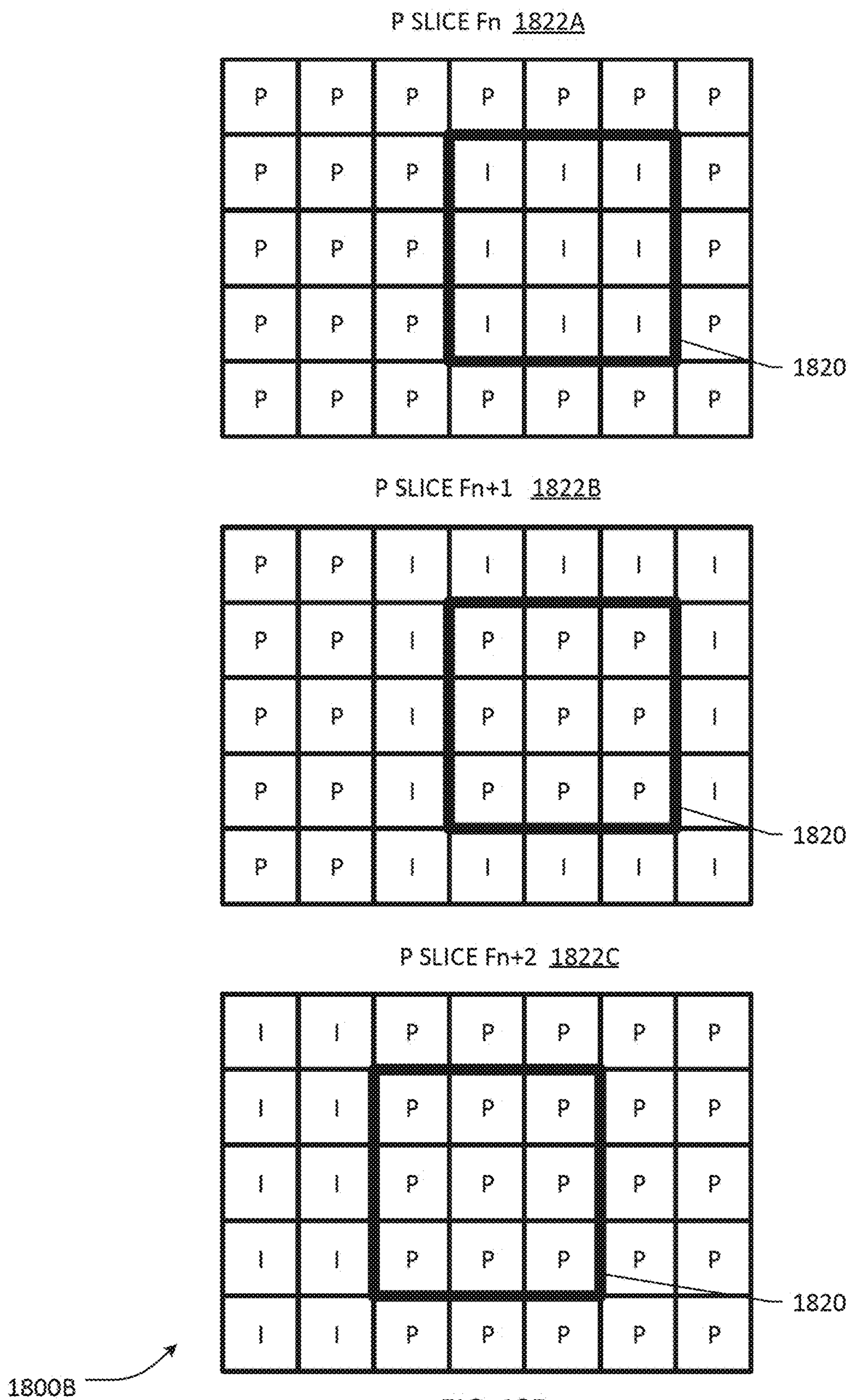

P SLICE Fn 1822A
1820
P SLICE Fn+1 1822B
1820
P SLICE Fn+2 1822C
1820
1800B

FIG. 18B

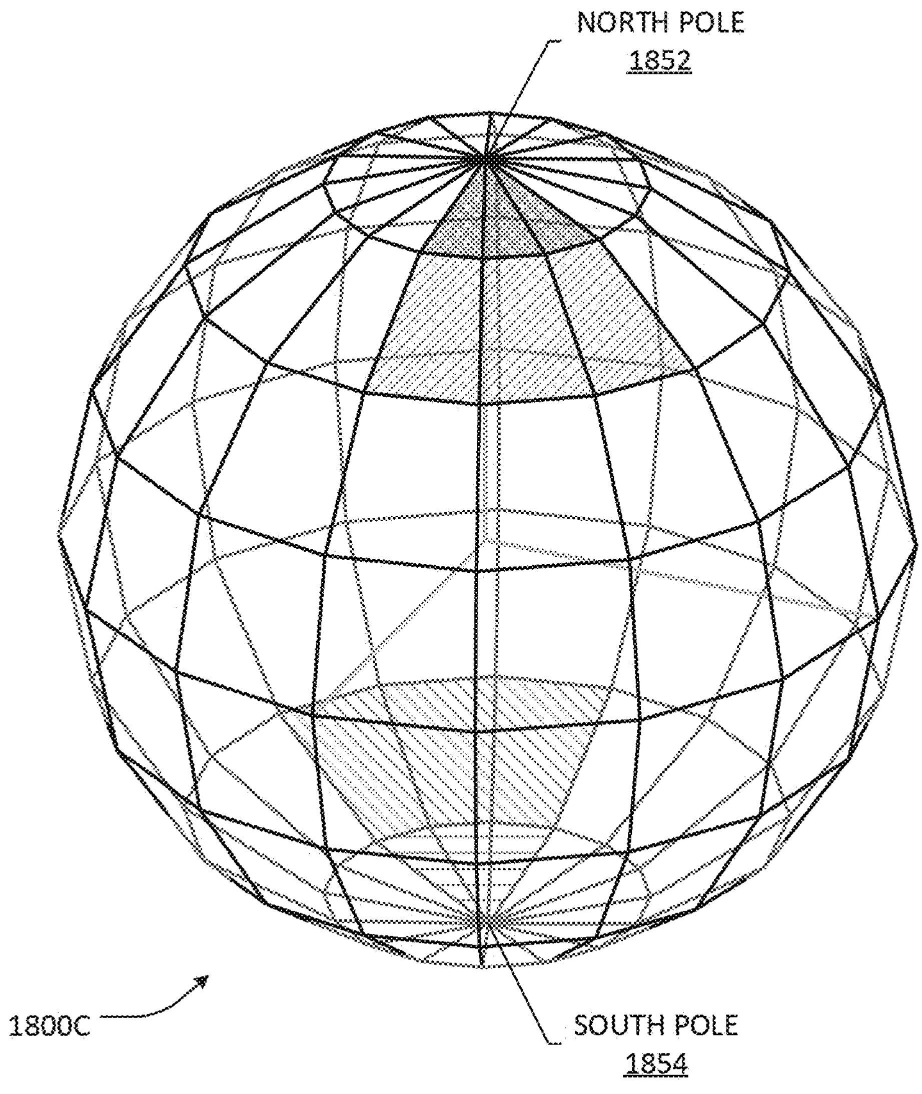


FIG. 18C

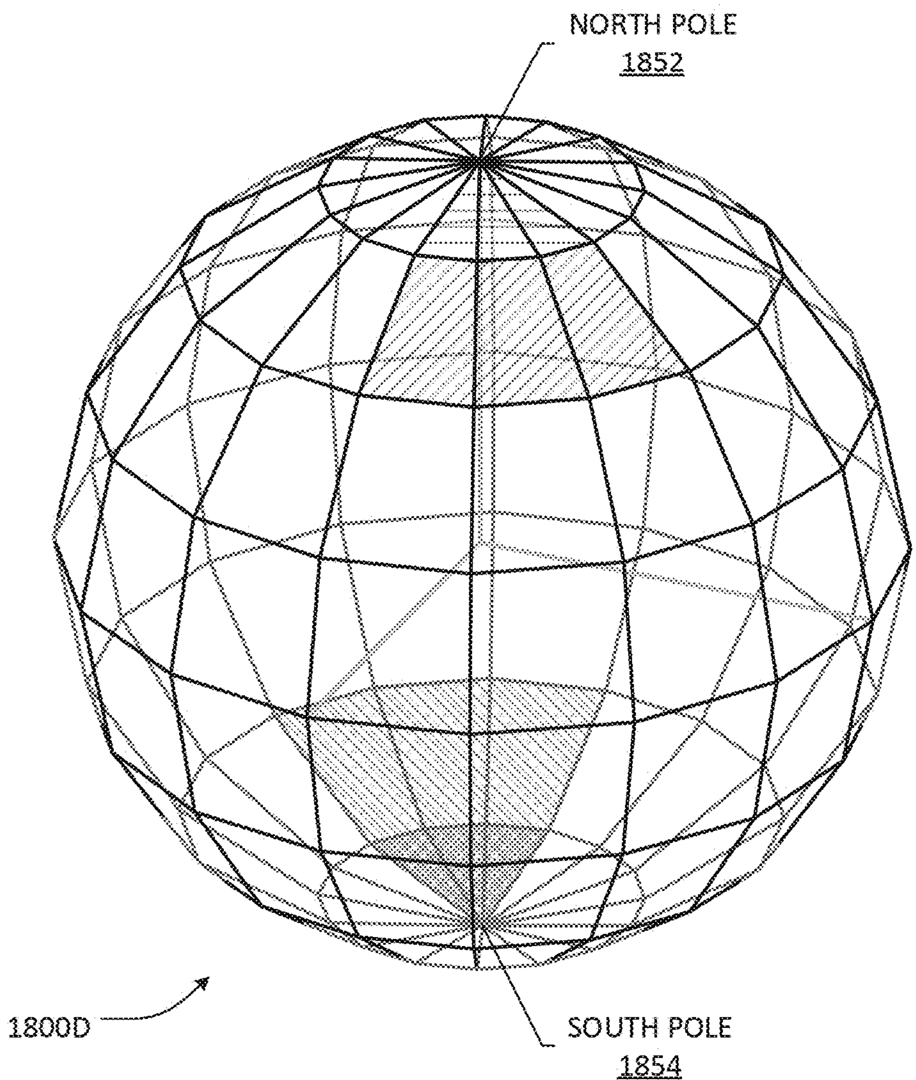


FIG. 18D

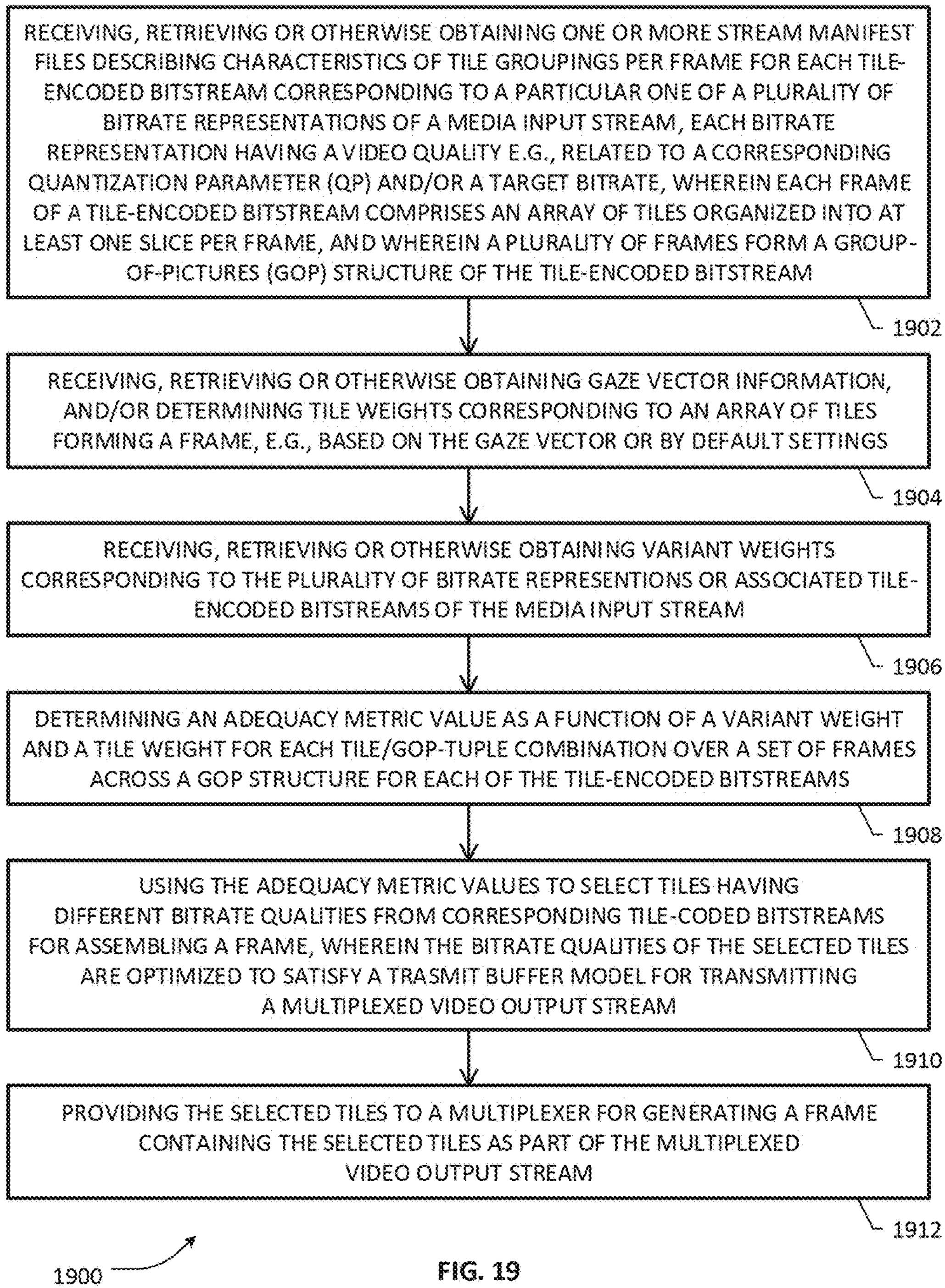

RECEIVING, RETRIEVING OR OTHERWISE OBTAINING ONE OR MORE STREAM MANIFEST FILES DESCRIBING CHARACTERISTICS OF TILE GROUPINGS PER FRAME FOR EACH TILE-ENCODED BITSTREAM CORRESPONDING TO A PARTICULAR ONE OF A PLURALITY OF BITRATE REPRESENTATIONS OF A MEDIA INPUT STREAM, EACH BITRATE REPRESENTATION HAVING A VIDEO QUALITY E.G., RELATED TO A CORRESPONDING QUANTIZATION PARAMETER (QP) AND/OR A TARGET BITRATE, WHEREIN EACH FRAME OF A TILE-ENCODED BITSTREAM COMPRISES AN ARRAY OF TILES ORGANIZED INTO AT LEAST ONE SLICE PER FRAME, AND WHEREIN A PLURALITY OF FRAMES FORM A GROUP-OF-PICTURES (GOP) STRUCTURE OF THE TILE-ENCODED BITSTREAM
1902
RECEIVING, RETRIEVING OR OTHERWISE OBTAINING GAZE VECTOR INFORMATION, AND/OR DETERMINING TILE WEIGHTS CORRESPONDING TO AN ARRAY OF TILES FORMING A FRAME, E.G., BASED ON THE GAZE VECTOR OR BY DEFAULT SETTINGS
1904
RECEIVING, RETRIEVING OR OTHERWISE OBTAINING VARIANT WEIGHTS CORRESPONDING TO THE PLURALITY OF BITRATE REPRESENTIONS OR ASSOCIATED TILE-ENCODED BITSTREAMS OF THE MEDIA INPUT STREAM
1906
DETERMINING AN ADEQUACY METRIC VALUE AS A FUNCTION OF A VARIANT WEIGHT AND A TILE WEIGHT FOR EACH TILE/GOP-TUPLE COMBINATION OVER A SET OF FRAMES ACROSS A GOP STRUCTURE FOR EACH OF THE TILE-ENCODED BITSTREAMS
1908
USING THE ADEQUACY METRIC VALUES TO SELECT TILES HAVING DIFFERENT BITRATE QUALITIES FROM CORRESPONDING TILE-CODED BITSTREAMS FOR ASSEMBLING A FRAME, WHEREIN THE BITRATE QUALITIES OF THE SELECTED TILES ARE OPTIMIZED TO SATISFY A TRASMIT BUFFER MODEL FOR TRANSMITTING A MULTIPLEXED VIDEO OUTPUT STREAM
1910
PROVIDING THE SELECTED TILES TO A MULTIPLEXER FOR GENERATING A FRAME CONTAINING THE SELECTED TILES AS PART OF THE MULTIPLEXED VIDEO OUTPUT STREAM
1912
1900

FIG. 19

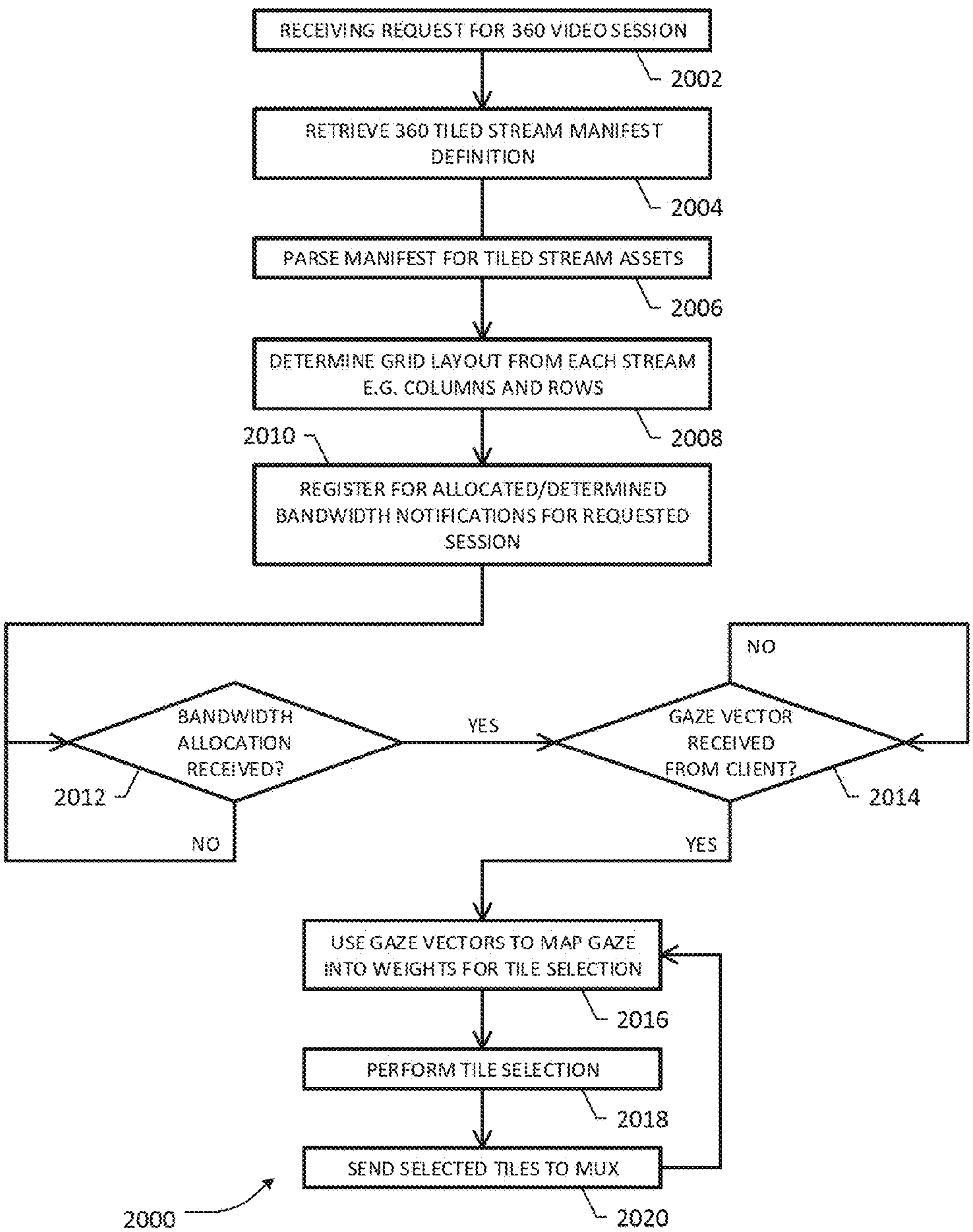

RECEIVING REQUEST FOR 360 VIDEO SESSION
2002
RETRIEVE 360 TILED STREAM MANIFEST DEFINITION
2004
PARSE MANIFEST FOR TILED STREAM ASSETS
2006
DETERMINE GRID LAYOUT FROM EACH STREAM E.G. COLUMNS AND ROWS
2008
2010
REGISTER FOR ALLOCATED/DETERMINED BANDWIDTH NOTIFICATIONS FOR REQUESTED SESSION
BANDWIDTH ALLOCATION RECEIVED?
2012
NO
YES
GAZE VECTOR RECEIVED FROM CLIENT?
NO
2014
YES
USE GAZE VECTORS TO MAP GAZE INTO WEIGHTS FOR TILE SELECTION
2016
PERFORM TILE SELECTION
2018
SEND SELECTED TILES TO MUX
2020
2000

FIG. 20

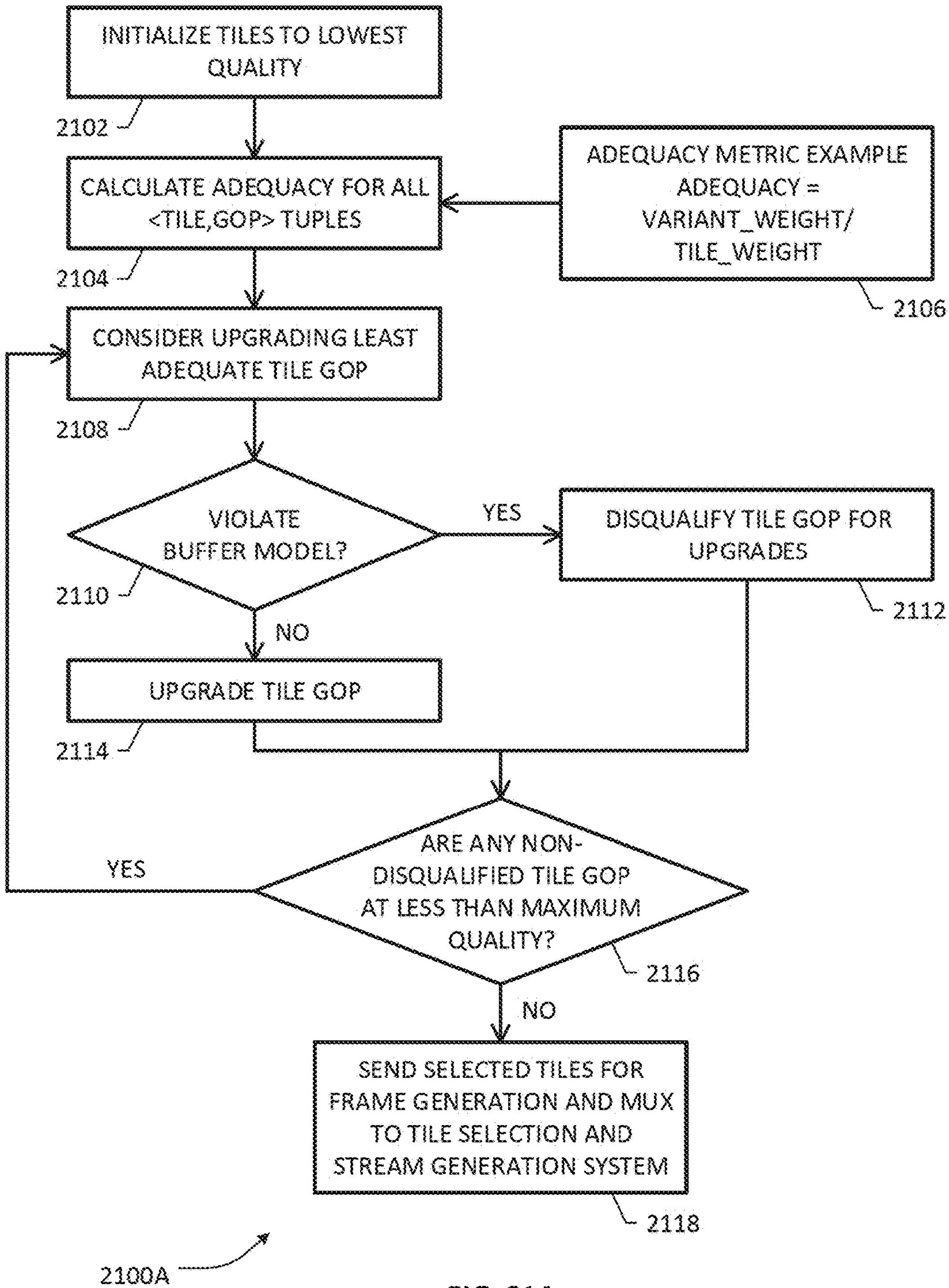

INITIALIZE TILES TO LOWEST QUALITY
2102
CALCULATE ADEQUACY FOR ALL <TILE,GOP> TUPLES
2104
ADEQUACY METRIC EXAMPLE ADEQUACY = VARIANT_WEIGHT/ TILE_WEIGHT
2106
CONSIDER UPGRADING LEAST ADEQUATE TILE GOP
2108
VIOLATE BUFFER MODEL?
2110
YES
DISQUALIFY TILE GOP FOR UPGRADES
2112
NO
UPGRADE TILE GOP
2114
YES
ARE ANY NON-DISQUALIFIED TILE GOP AT LESS THAN MAXIMUM QUALITY?
2116
NO
SEND SELECTED TILES FOR FRAME GENERATION AND MUX TO TILE SELECTION AND STREAM GENERATION SYSTEM
2118
2100A

FIG. 21A

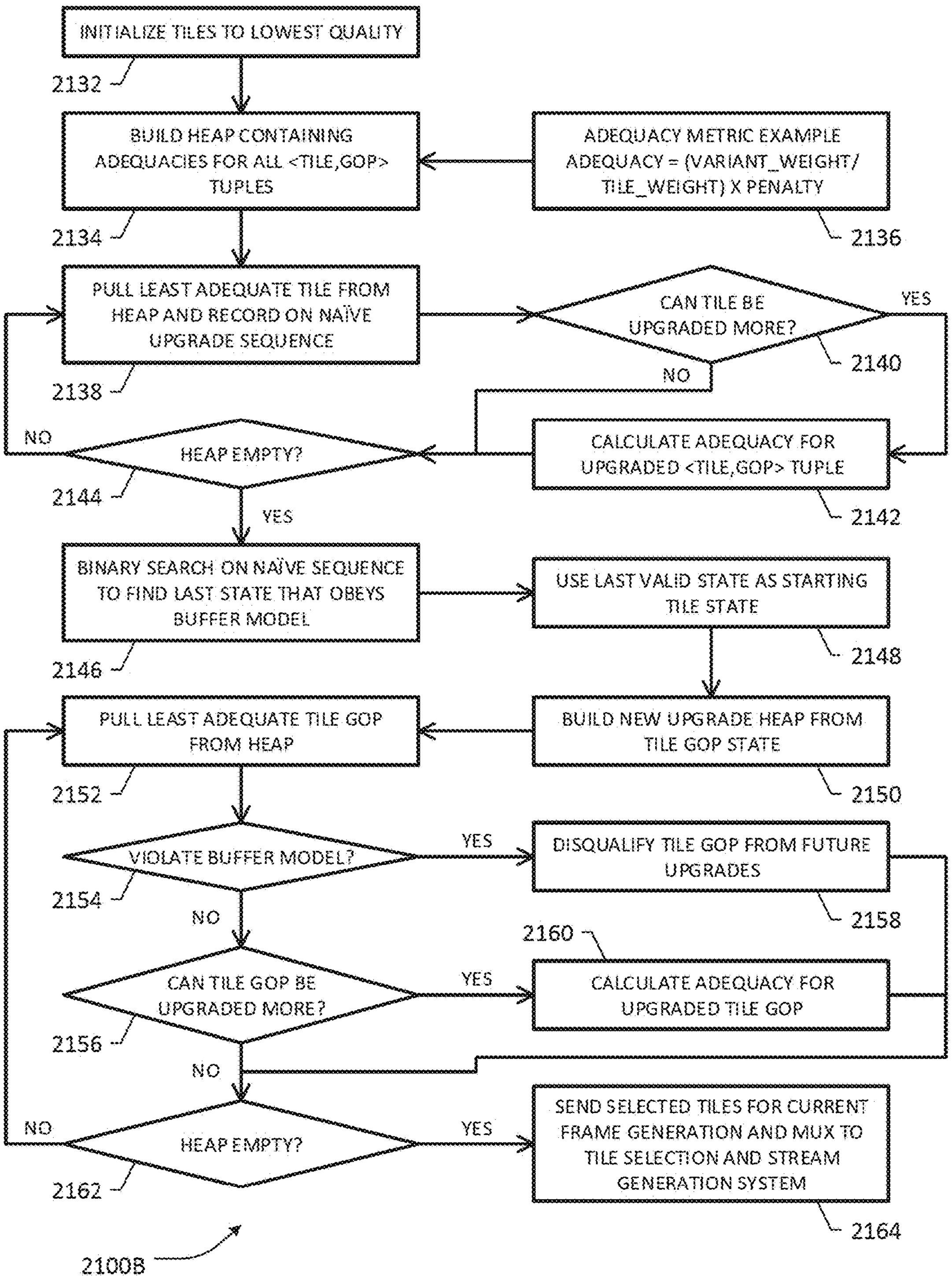

INITIALIZE TILES TO LOWEST QUALITY
2132
BUILD HEAP CONTAINING ADEQUACIES FOR ALL <TILE,GOP> TUPLES
ADEQUACY METRIC EXAMPLE ADEQUACY = (VARIANT_WEIGHT/ TILE_WEIGHT) X PENALTY
2134
2136
PULL LEAST ADEQUATE TILE FROM HEAP AND RECORD ON NAÏVE UPGRADE SEQUENCE
CAN TILE BE UPGRADED MORE?
YES
2140
NO
2138
NO
HEAP EMPTY?
CALCULATE ADEQUACY FOR UPGRADED <TILE,GOP> TUPLE
2144
YES
2142
BINARY SEARCH ON NAÏVE SEQUENCE TO FIND LAST STATE THAT OBEYS BUFFER MODEL
USE LAST VALID STATE AS STARTING TILE STATE
2148
2146
PULL LEAST ADEQUATE TILE GOP FROM HEAP
BUILD NEW UPGRADE HEAP FROM TILE GOP STATE
2152
2150
VIOLATE BUFFER MODEL?
YES
DISQUALIFY TILE GOP FROM FUTURE UPGRADES
2154
NO
2158
2160
CAN TILE GOP BE UPGRADED MORE?
YES
CALCULATE ADEQUACY FOR UPGRADED TILE GOP
2156
NO
NO
HEAP EMPTY?
YES
SEND SELECTED TILES FOR CURRENT FRAME GENERATION AND MUX TO TILE SELECTION AND STREAM GENERATION SYSTEM
2162
2164
2100B

FIG. 21B

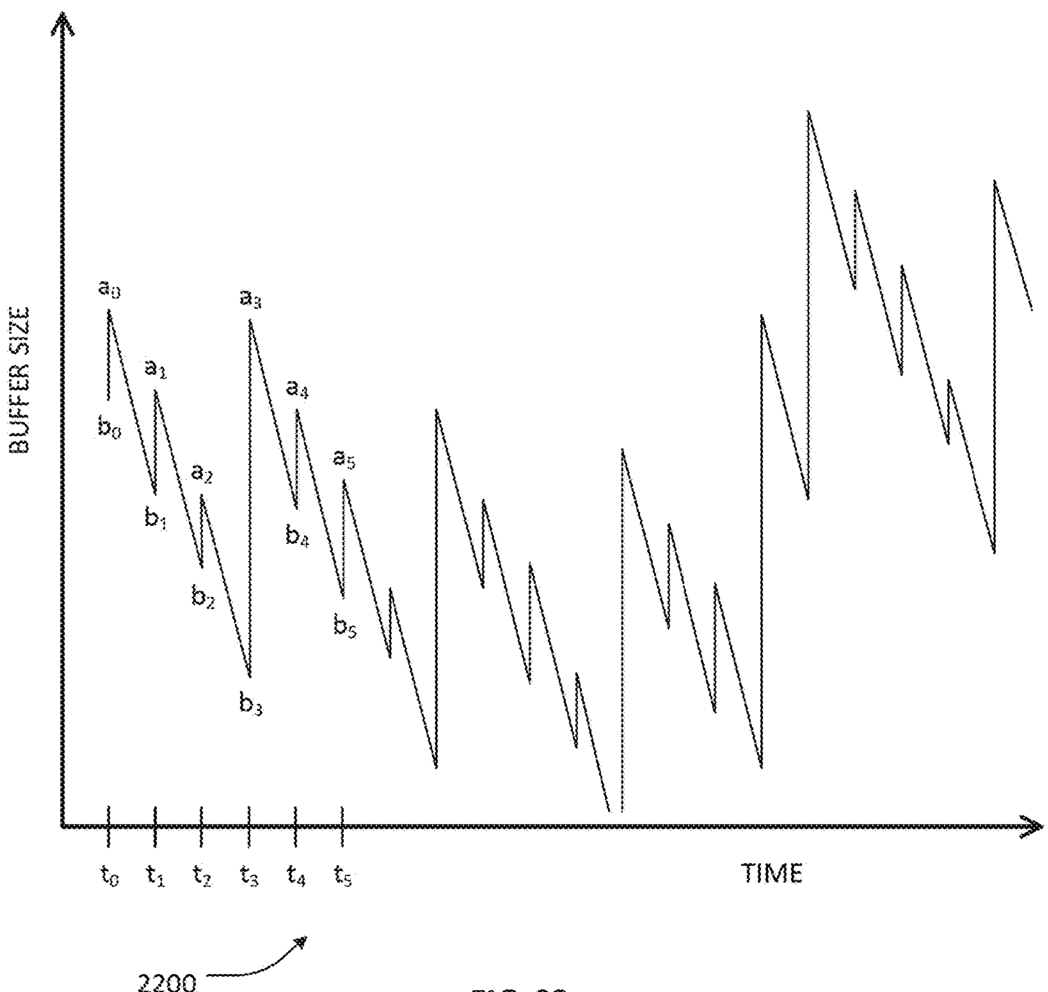

BUFFER SIZE
a0
a1
a2
a3
a4
a5
b0
b1
b2
b3
b4
b5
t0
t1
t2
t3
t4
t5
TIME
2200

FIG. 22

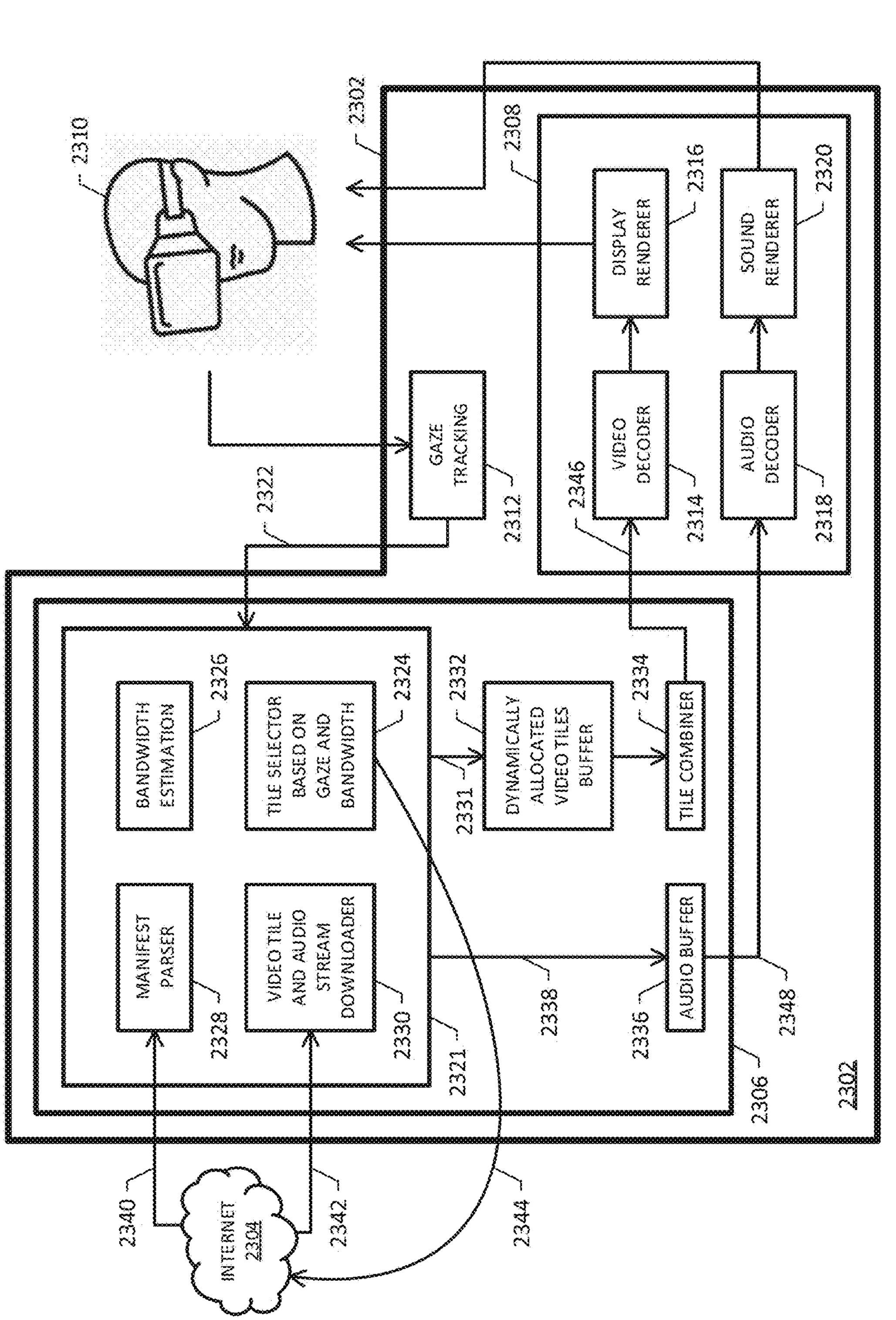

2300
2302
2310
2308
2316
2320
DISPLAY RENDERER
SOUND RENDERER
GAZE TRACKING
VIDEO DECODER
AUDIO DECODER
2322
2346
2312
2314
2318
2326
2324
2332
2334
BANDWIDTH ESTIMATION
TILE SELECTOR BASED ON GAZE AND BANDWIDTH
DYNAMICALLY ALLOCATED VIDEO TILES BUFFER
TILE COMBINER
2331
MANIFEST PARSER
VIDEO TILE AND AUDIO STREAM DOWNLOADER
AUDIO BUFFER
2328
2330
2321
2338
2336
2348
2306
2302
2340
2342
2344
INTERNET 2304

FIG. 23

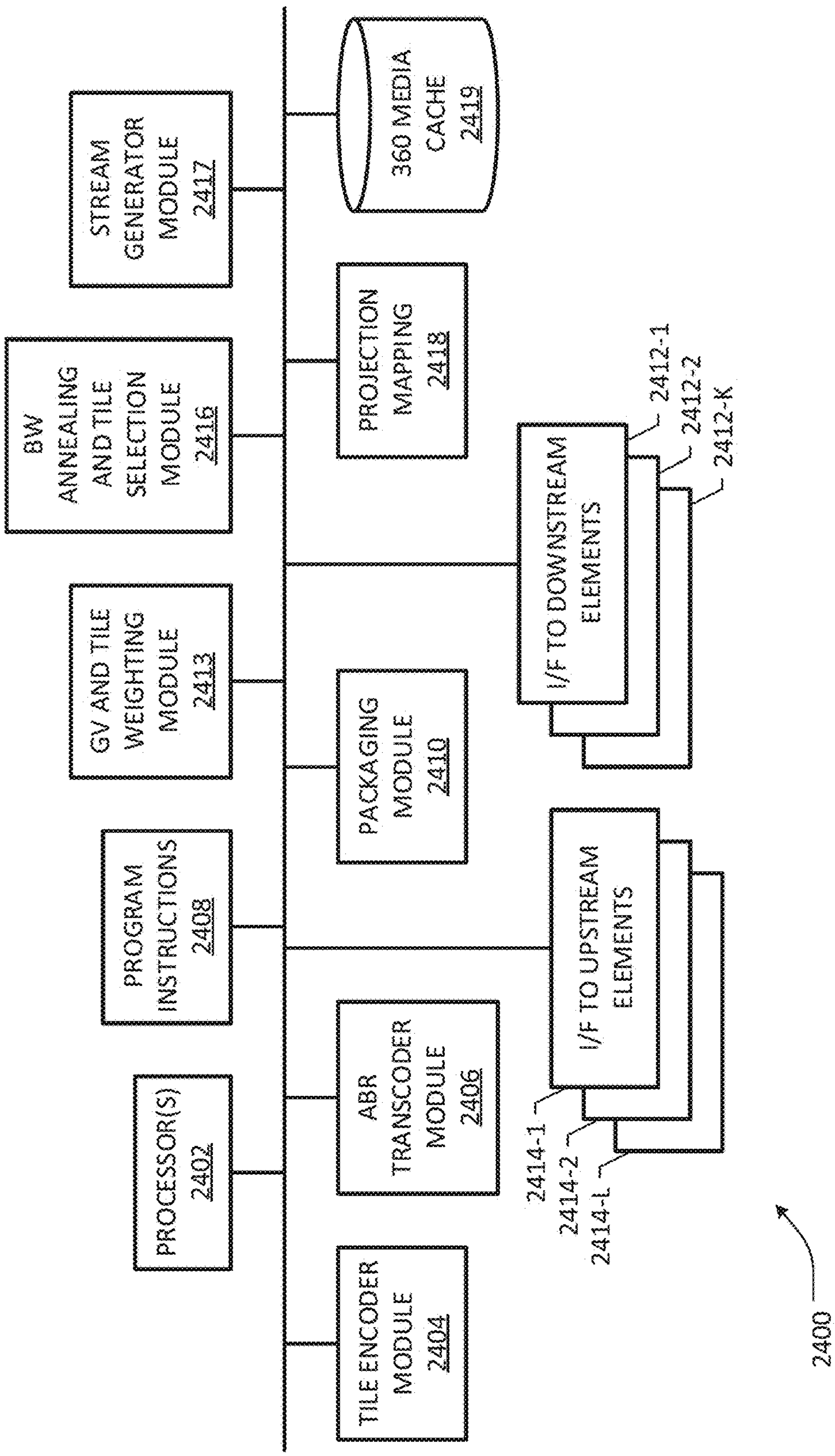
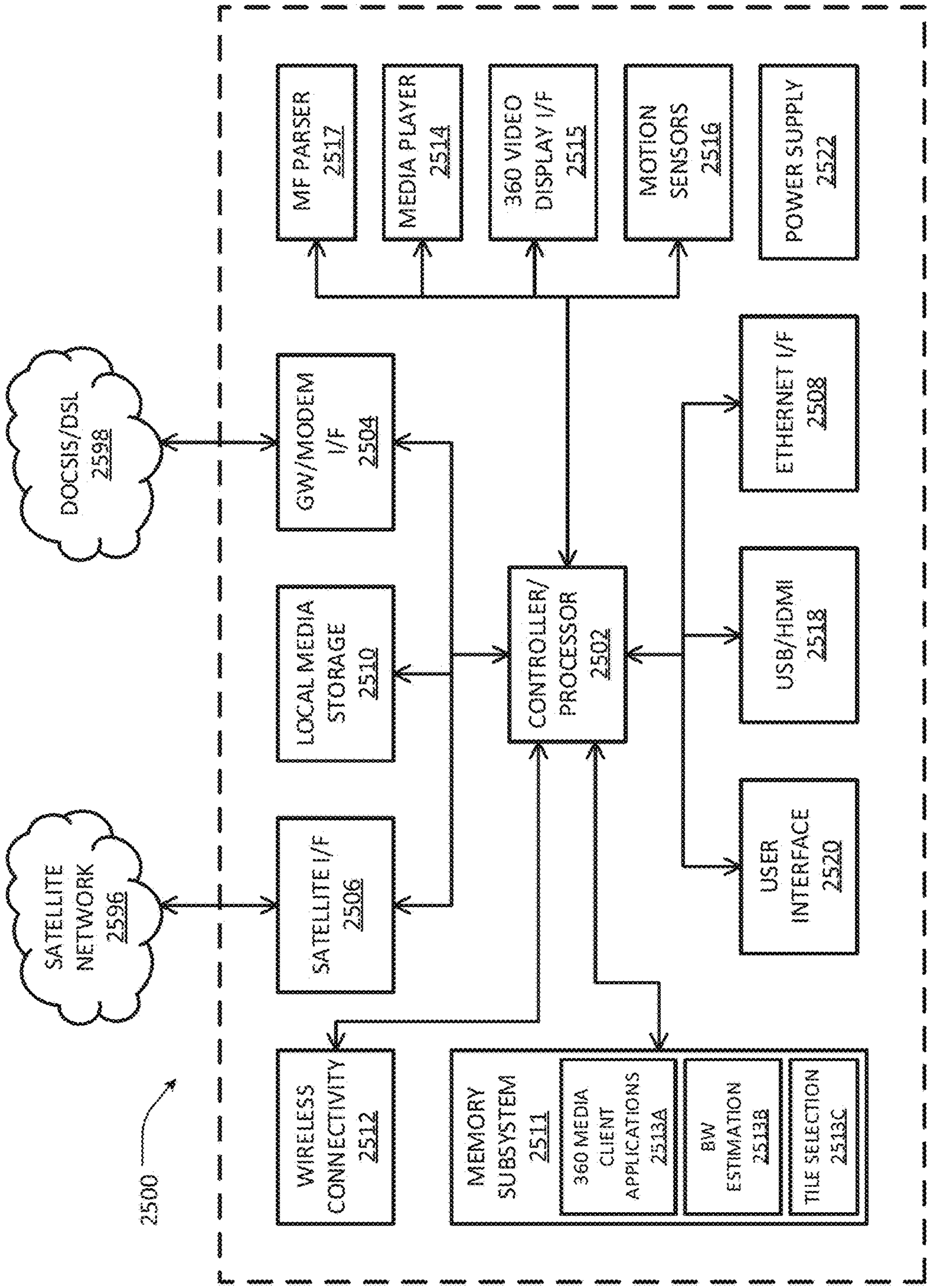

STREAM GENERATOR MODULE 2417
BW ANNEALING AND TILE SELECTION MODULE 2416
GV AND TILE WEIGHTING MODULE 2413
PROGRAM INSTRUCTIONS 2408
PROCESSOR(S) 2402
360 MEDIA CACHE 2419
PROJECTION MAPPING 2418
PACKAGING MODULE 2410
ABR TRANSCODER MODULE 2406
TILE ENCODER MODULE 2404
I/F TO DOWNSTREAM ELEMENTS
2412-1
2412-2
2412-K
I/F TO UPSTREAM ELEMENTS
2414-1
2414-2
2414-L
2400

FIG. 24



2500
SATELLITE NETWORK 2596
DOCSIS/DSL 2598
WIRELESS CONNECTIVITY 2512
SATELLITE I/F 2506
LOCAL MEDIA STORAGE 2510
GW/MODEM I/F 2504
MF PARSER 2517
MEDIA PLAYER 2514
360 VIDEO DISPLAY I/F 2515
MOTION SENSORS 2516
POWER SUPPLY 2522
CONTROLLER/ PROCESSOR 2502
MEMORY SUBSYSTEM 2511
360 MEDIA CLIENT APPLICATIONS 2513A
BW ESTIMATION 2513B
TILE SELECTION 2513C
USER INTERFACE 2520
USB/HDMI 2518
ETHERNET I/F 2508

FIG. 25

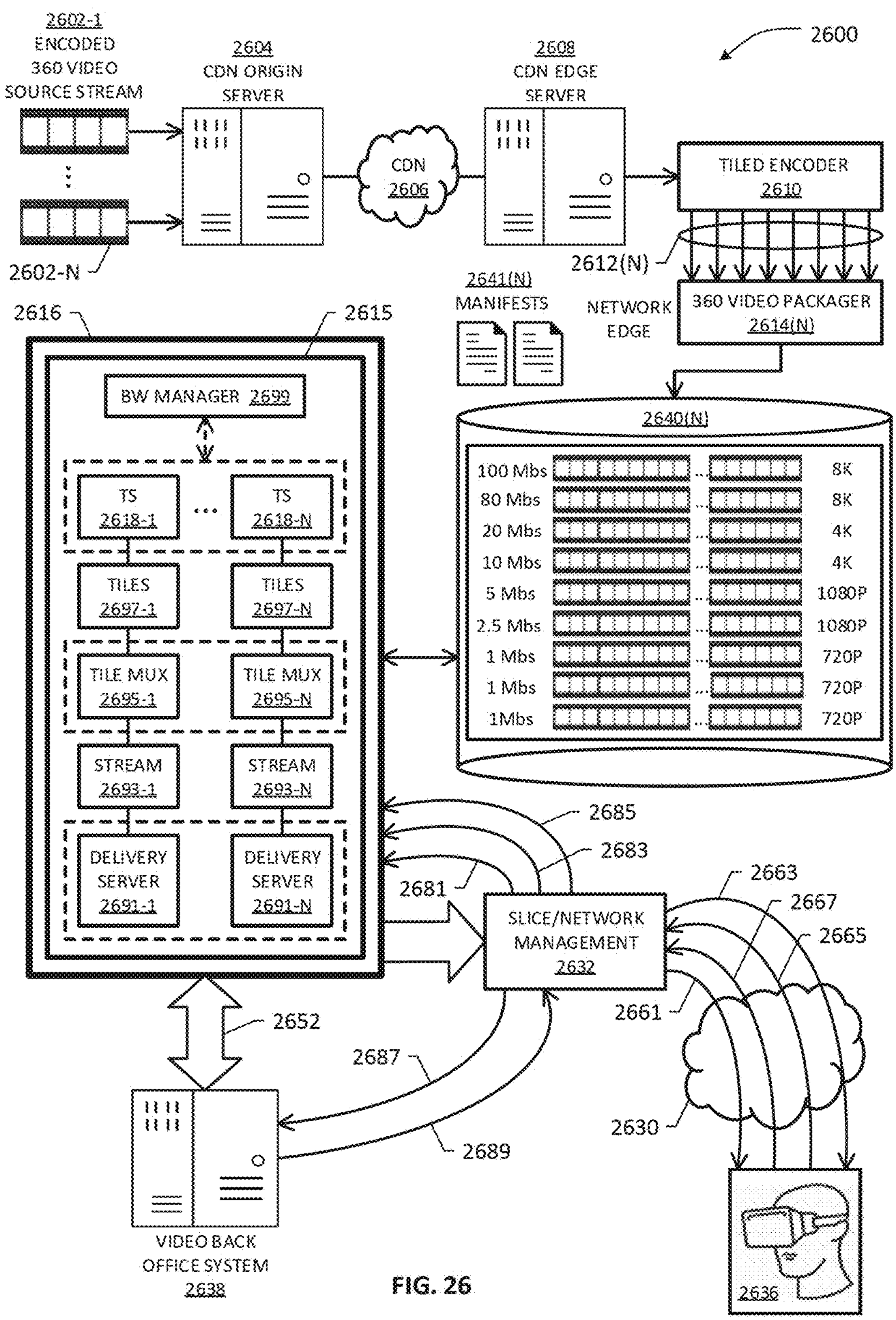

2600
2602-1
ENCODED 360 VIDEO SOURCE STREAM
2602-N
2604
CDN ORIGIN SERVER
CDN
2606
2608
CDN EDGE SERVER
TILED ENCODER
2610
2612(N)
360 VIDEO PACKAGER
2614(N)
NETWORK EDGE
2641(N)
MANIFESTS
2616
2615
BW MANAGER 2699
TS
2618-1
TS
2618-N
TILES
2697-1
TILES
2697-N
TILE MUX
2695-1
TILE MUX
2695-N
STREAM
2693-1
STREAM
2693-N
DELIVERY SERVER
2691-1
DELIVERY SERVER
2691-N
2640(N)
100 Mbs 8K
80 Mbs 8K
20 Mbs 4K
10 Mbs 4K
5 Mbs 1080P
2.5 Mbs 1080P
1 Mbs 720P
1 Mbs 720P
1Mbs 720P
2685
2683
2681
2663
2667
2665
SLICE/NETWORK MANAGEMENT
2632
2661
2652
2687
2689
2630
VIDEO BACK OFFICE SYSTEM
2638
2636

FIG. 26

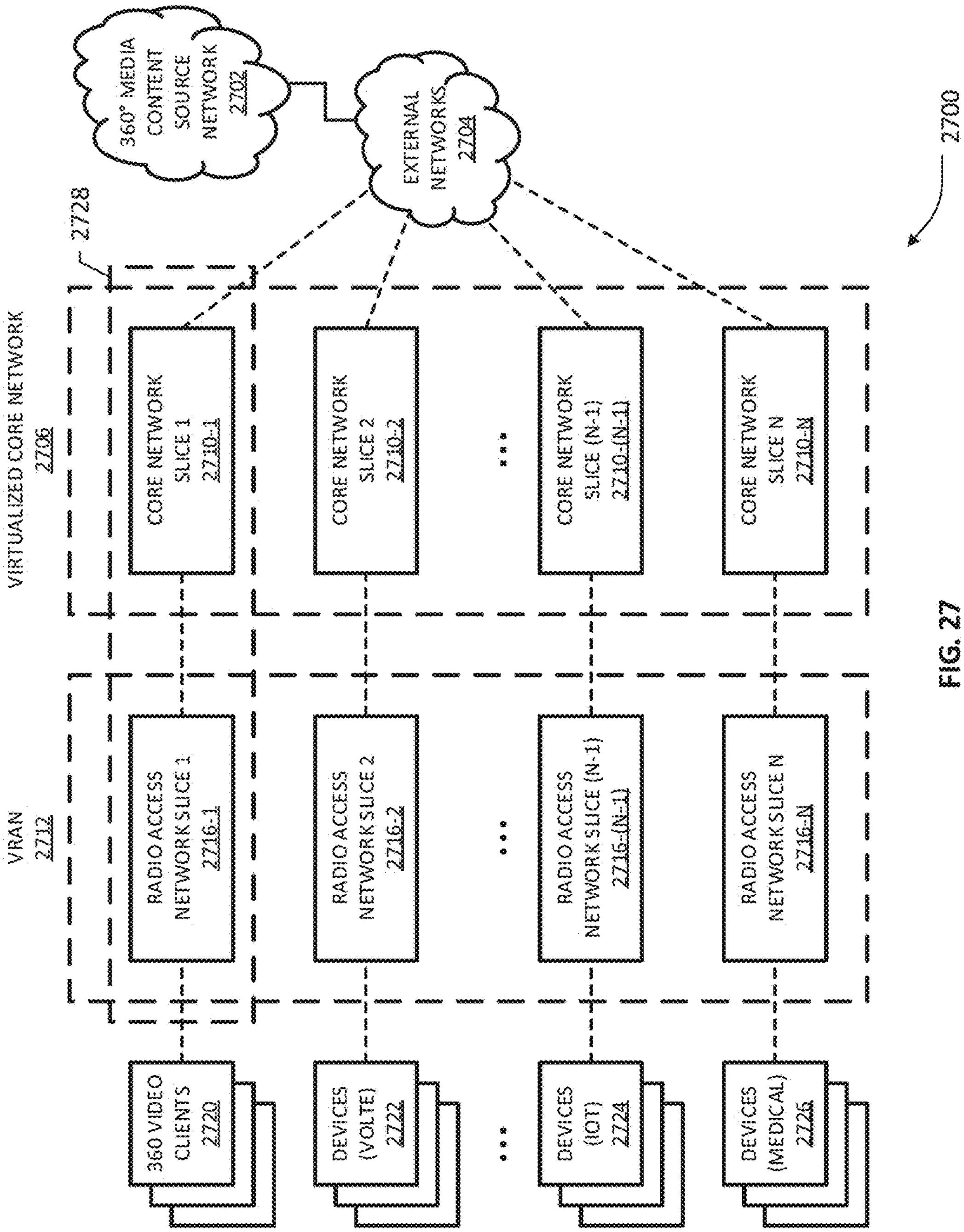

360° MEDIA CONTENT SOURCE NETWORK 2702
EXTERNAL NETWORKS 2704
2728
VIRTUALIZED CORE NETWORK 2706
CORE NETWORK SLICE 1 2710-1
CORE NETWORK SLICE 2 2710-2
CORE NETWORK SLICE (N-1) 2710-(N-1)
CORE NETWORK SLICE N 2710-N
VRAN 2712
RADIO ACCESS NETWORK SLICE 1 2716-1
RADIO ACCESS NETWORK SLICE 2 2716-2
RADIO ACCESS NETWORK SLICE (N-1) 2716-(N-1)
RADIO ACCESS NETWORK SLICE N 2716-N
360 VIDEO CLIENTS 2720
DEVICES (VOLTE) 2722
DEVICES (IOT) 2724
DEVICES (MEDICAL) 2726
2700

FIG. 27

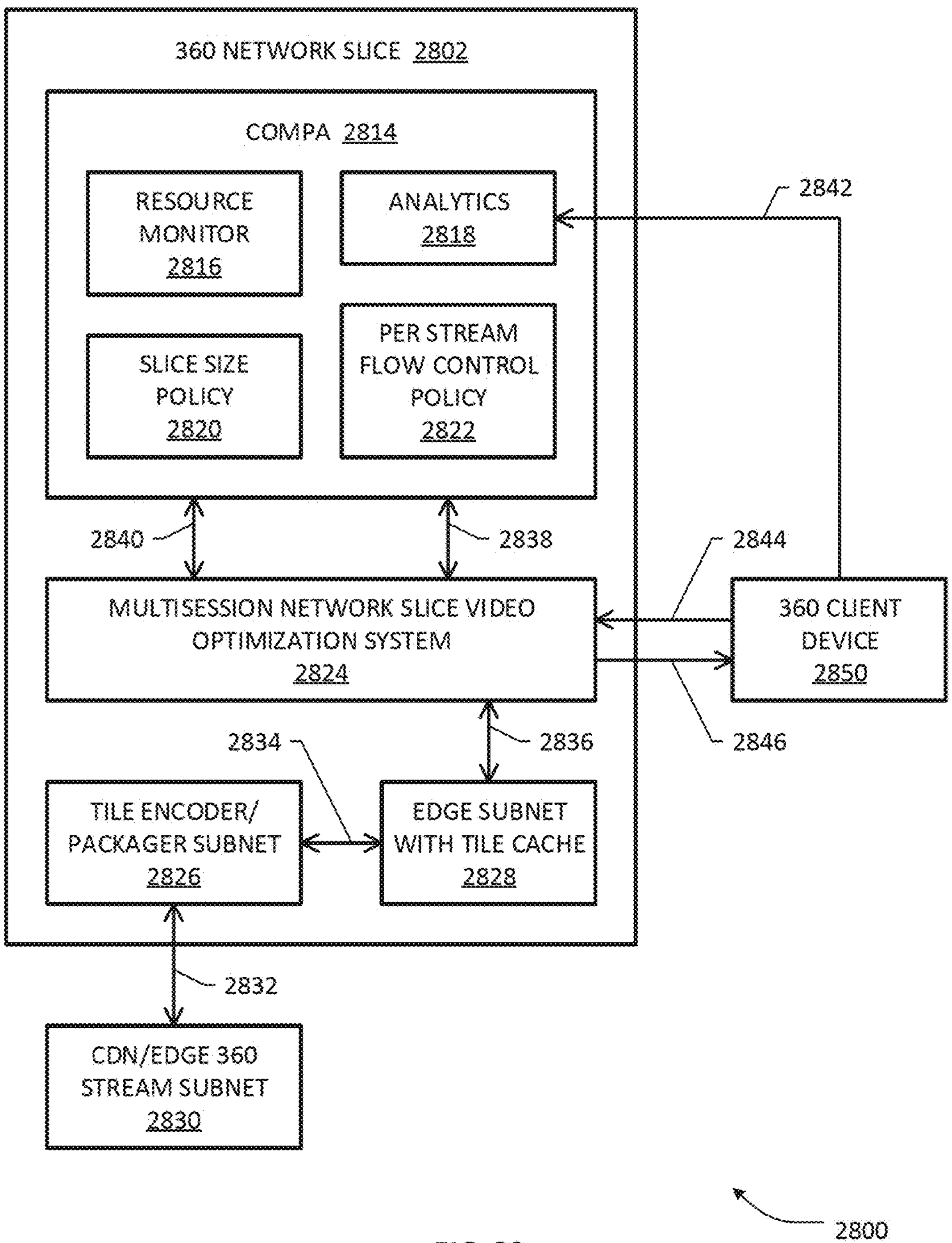
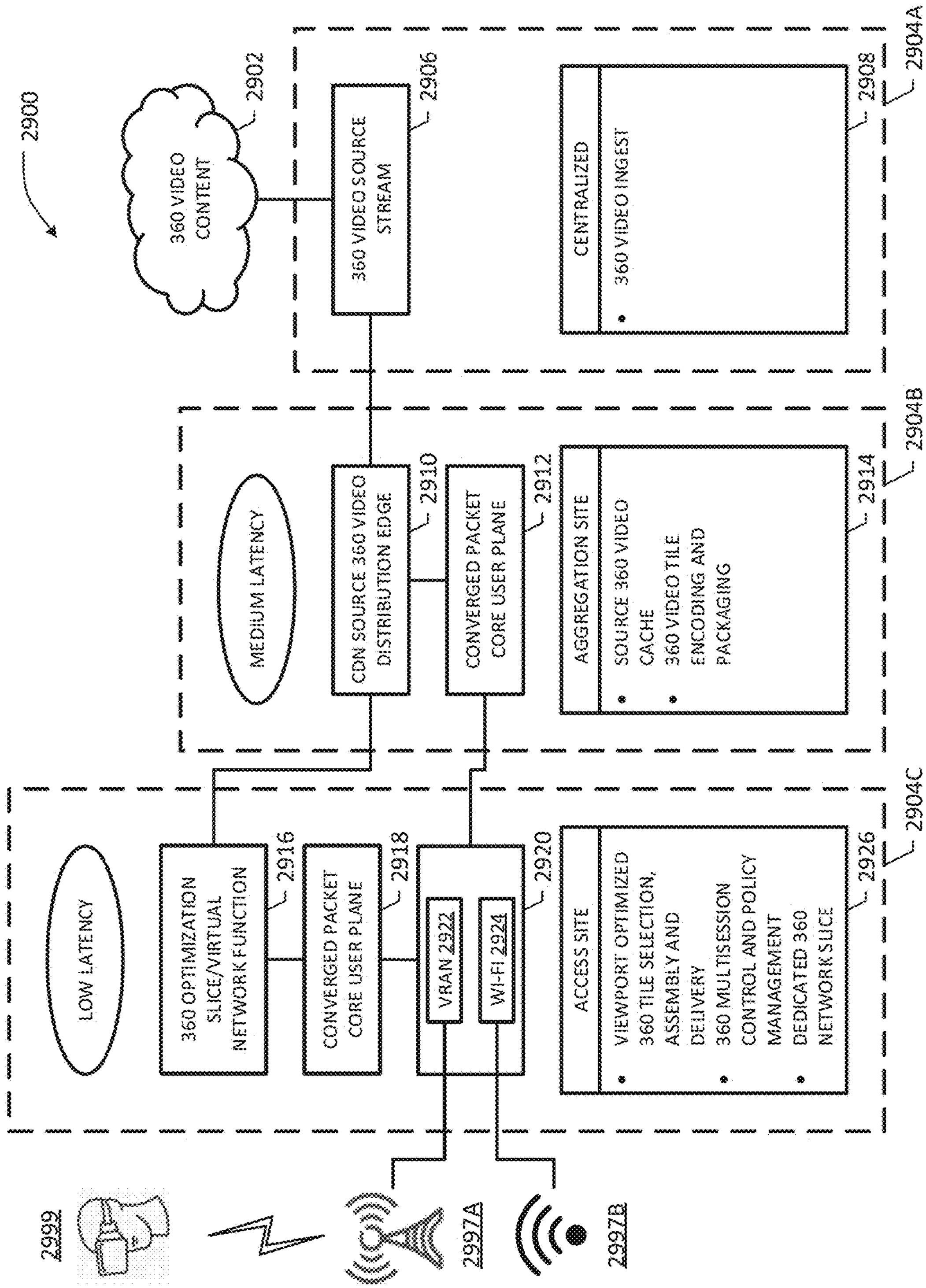

360 NETWORK SLICE 2802
COMPA 2814
RESOURCE MONITOR 2816
ANALYTICS 2818
SLICE SIZE POLICY 2820
PER STREAM FLOW CONTROL POLICY 2822
2842
2840
2838
2844
MULTISESSION NETWORK SLICE VIDEO OPTIMIZATION SYSTEM 2824
360 CLIENT DEVICE 2850
2846
2834
2836
TILE ENCODER/ PACKAGER SUBNET 2826
EDGE SUBNET WITH TILE CACHE 2828
2832
CDN/EDGE 360 STREAM SUBNET 2830
2800

FIG. 28



2900
360 VIDEO CONTENT
2902
360 VIDEO SOURCE STREAM
2906
CENTRALIZED
360 VIDEO INGEST
2908
2904A
MEDIUM LATENCY
CDN SOURCE 360 VIDEO DISTRIBUTION EDGE
2910
CONVERGED PACKET CORE USER PLANE
2912
AGGREGATION SITE
SOURCE 360 VIDEO CACHE
360 VIDEO TILE ENCODING AND PACKAGING
2914
2904B
LOW LATENCY
360 OPTIMIZATION SLICE/VIRTUAL NETWORK FUNCTION
2916
CONVERGED PACKET CORE USER PLANE
2918
VRAN 2922
WI-FI 2924
2920
ACCESS SITE
VIEWPORT OPTIMIZED 360 TILE SELECTION, ASSEMBLY AND DELIVERY
360 MULTISESSION CONTROL AND POLICY MANAGEMENT
DEDICATED 360 NETWORK SLICE
2926
2904C
2999
2997A
2997B

FIG.29

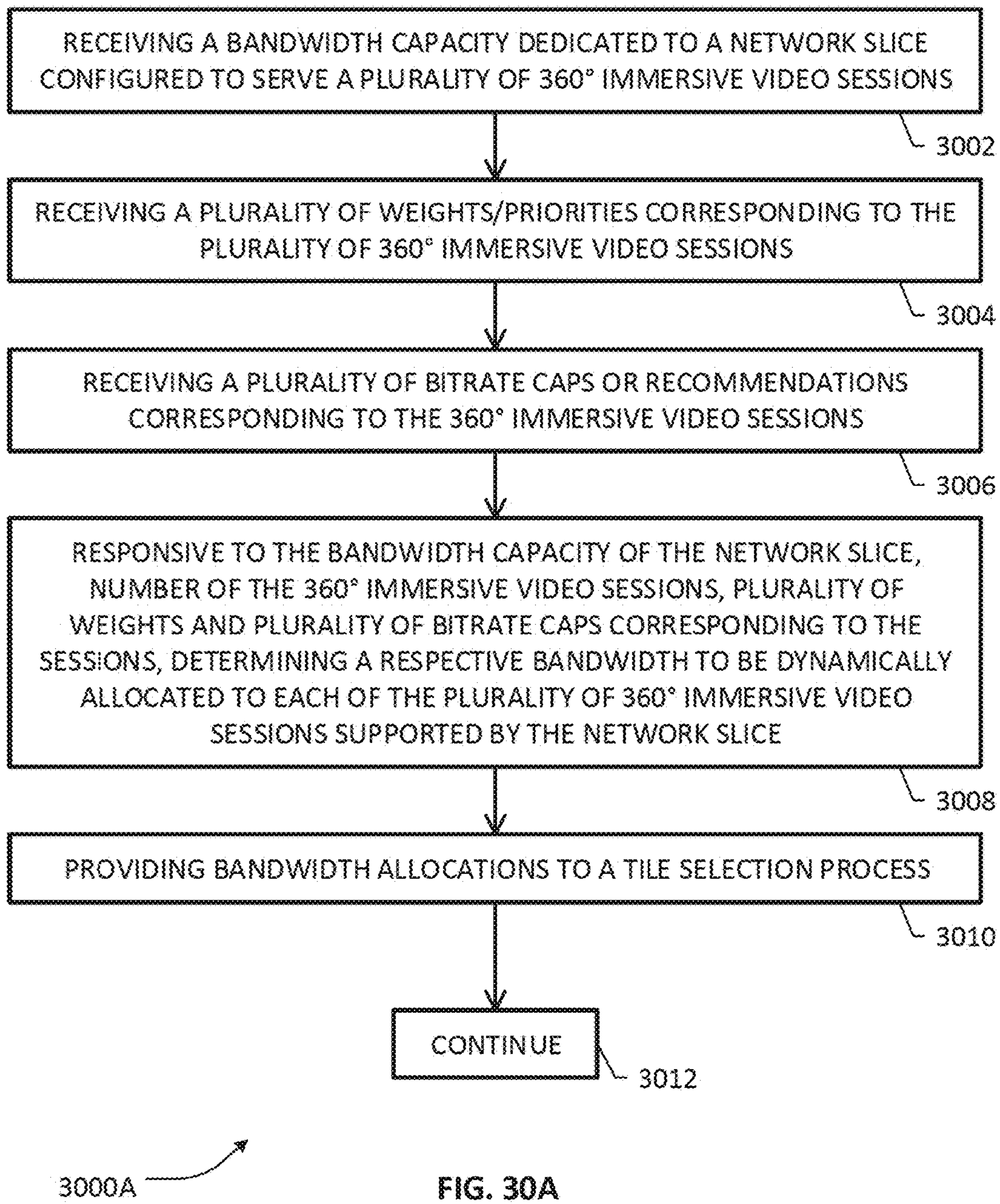

RECEIVING A BANDWIDTH CAPACITY DEDICATED TO A NETWORK SLICE CONFIGURED TO SERVE A PLURALITY OF 360° IMMERSIVE VIDEO SESSIONS
3002
RECEIVING A PLURALITY OF WEIGHTS/PRIORITIES CORRESPONDING TO THE PLURALITY OF 360° IMMERSIVE VIDEO SESSIONS
3004
RECEIVING A PLURALITY OF BITRATE CAPS OR RECOMMENDATIONS CORRESPONDING TO THE 360° IMMERSIVE VIDEO SESSIONS
3006
RESPONSIVE TO THE BANDWIDTH CAPACITY OF THE NETWORK SLICE, NUMBER OF THE 360° IMMERSIVE VIDEO SESSIONS, PLURALITY OF WEIGHTS AND PLURALITY OF BITRATE CAPS CORRESPONDING TO THE SESSIONS, DETERMINING A RESPECTIVE BANDWIDTH TO BE DYNAMICALLY ALLOCATED TO EACH OF THE PLURALITY OF 360° IMMERSIVE VIDEO SESSIONS SUPPORTED BY THE NETWORK SLICE
3008
PROVIDING BANDWIDTH ALLOCATIONS TO A TILE SELECTION PROCESS
3010
CONTINUE
3012
3000A

FIG. 30A

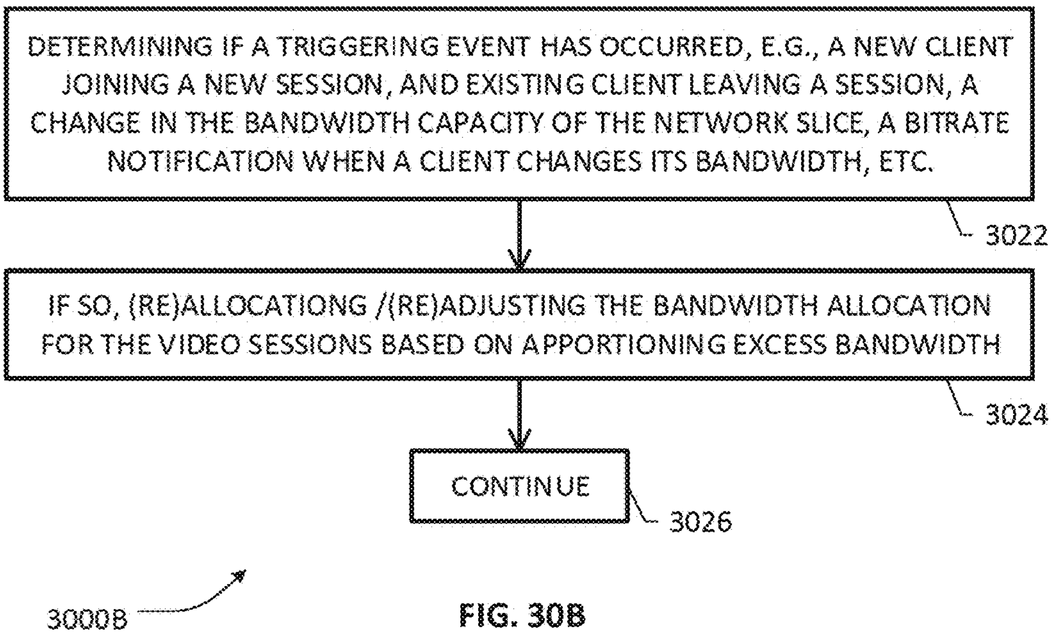


FIG. 30B

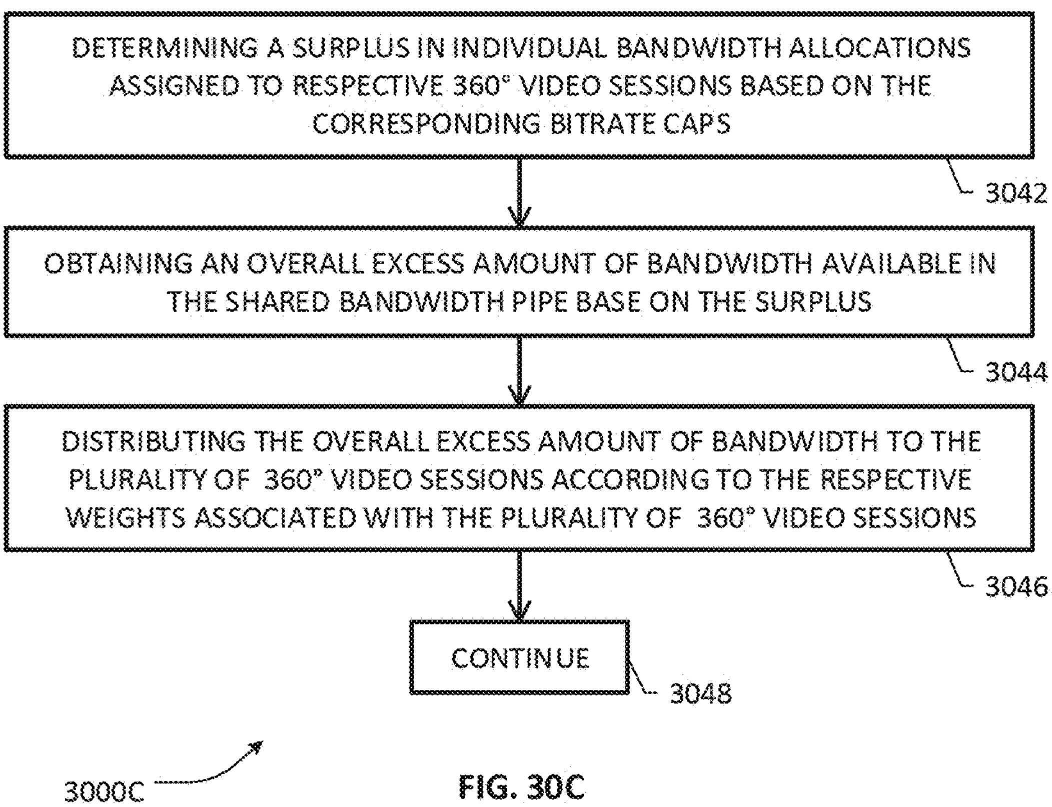


FIG. 30C

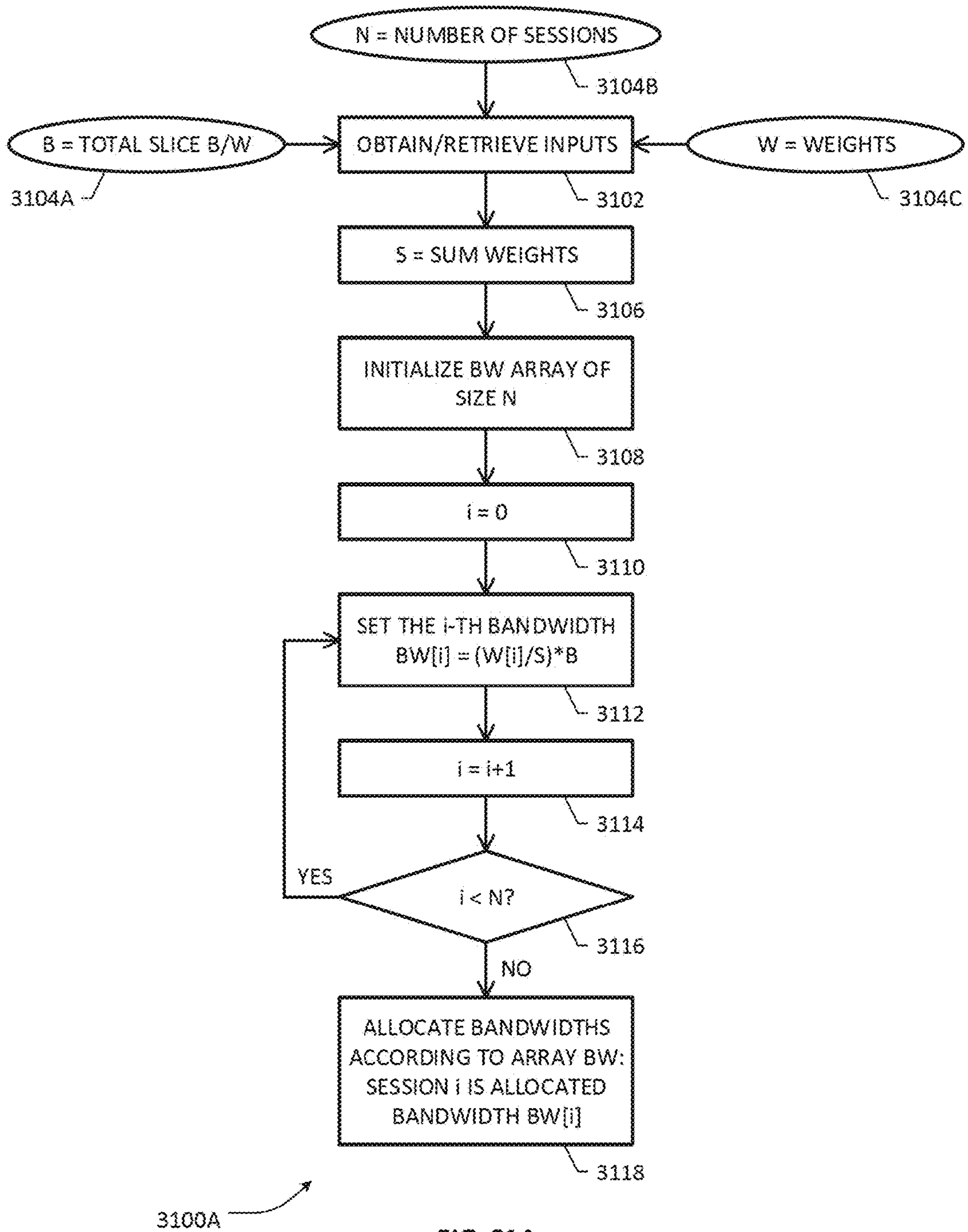

N = NUMBER OF SESSIONS
3104B
B = TOTAL SLICE B/W
3104A
OBTAIN/RETRIEVE INPUTS
3102
W = WEIGHTS
3104C
S = SUM WEIGHTS
3106
INITIALIZE BW ARRAY OF SIZE N
3108
i = 0
3110
SET THE i-TH BANDWIDTH BW[i] = (W[i]/S)*B
3112
i = i+1
3114
YES
i < N?
3116
NO
ALLOCATE BANDWIDTHS ACCORDING TO ARRAY BW: SESSION i IS ALLOCATED BANDWIDTH BW[i]
3118
3100A

FIG. 31A

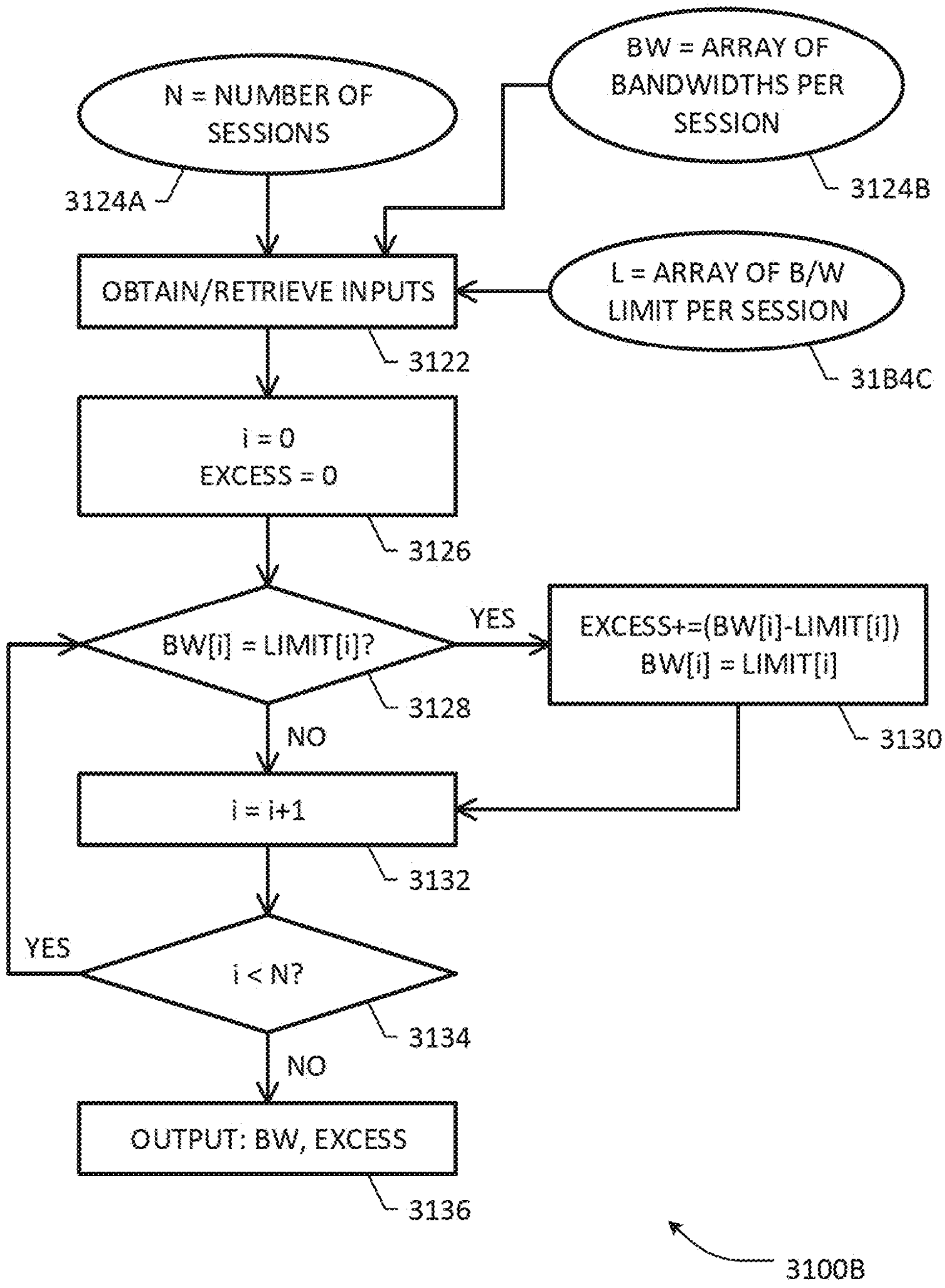
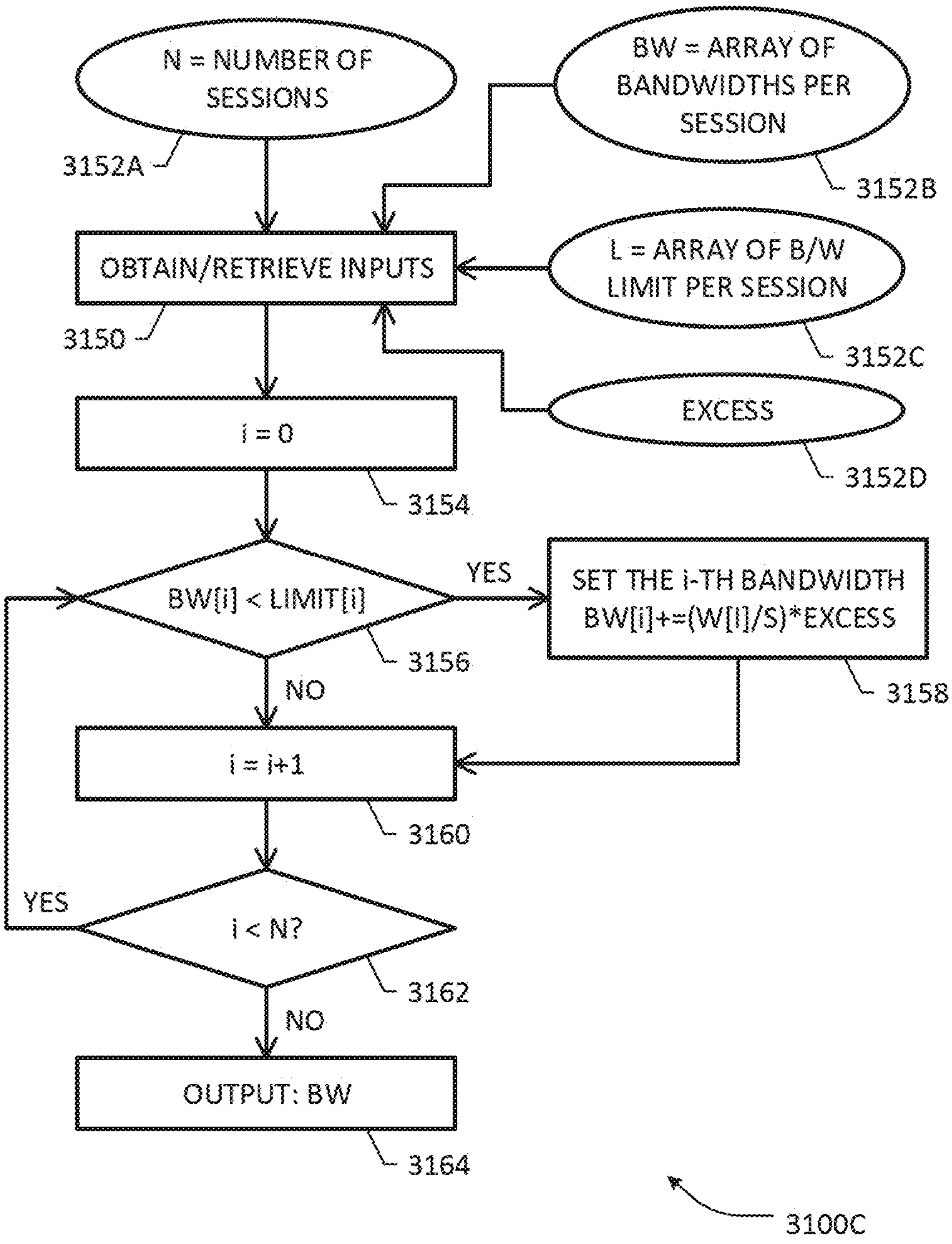
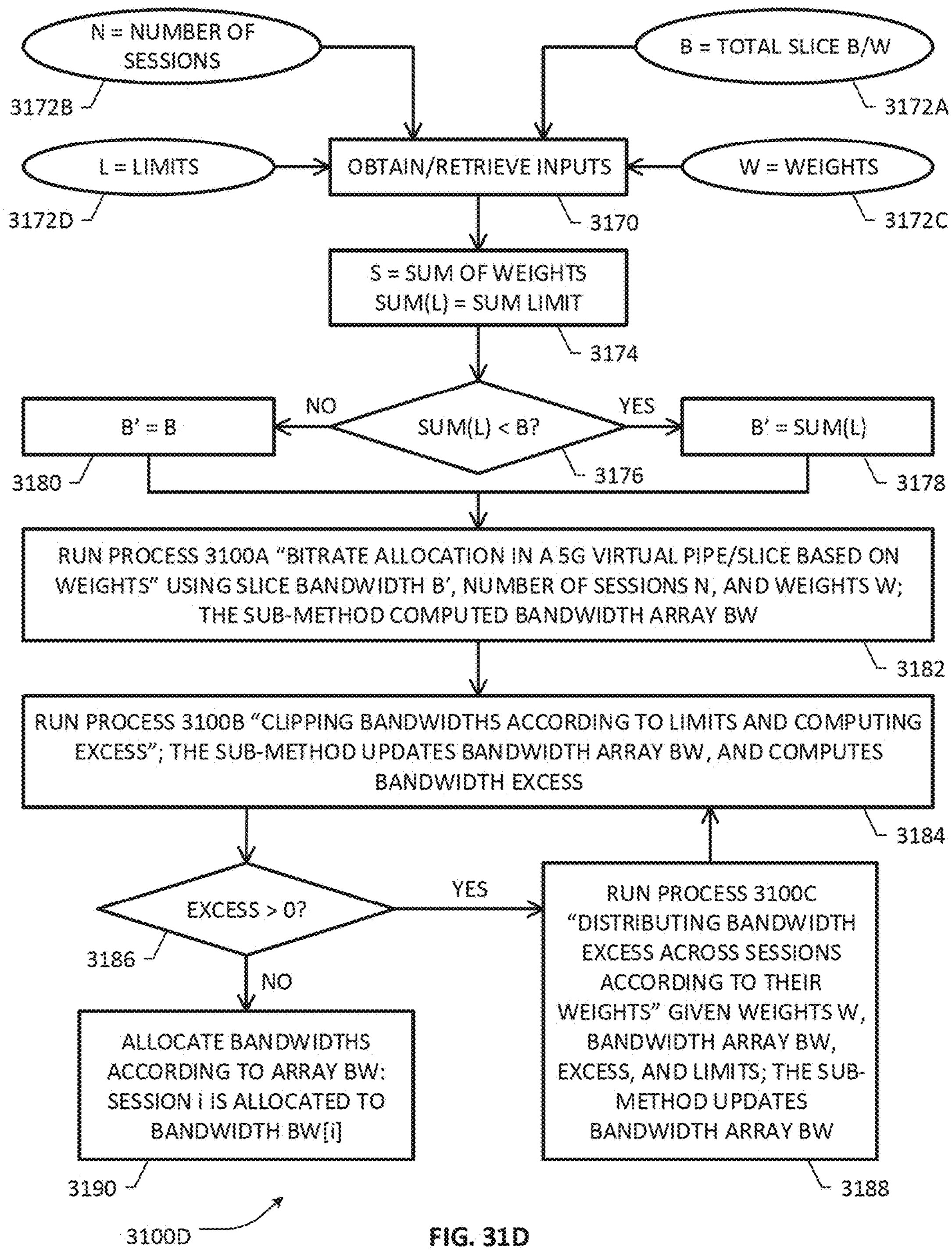

N = NUMBER OF SESSIONS
3124A
BW = ARRAY OF BANDWIDTHS PER SESSION
3124B
OBTAIN/RETRIEVE INPUTS
3122
L = ARRAY OF B/W LIMIT PER SESSION
31B4C
i = 0
EXCESS = 0
3126
BW[i] = LIMIT[i]?
3128
YES
EXCESS+=(BW[i]-LIMIT[i])
BW[i] = LIMIT[i]
3130
NO
i = i+1
3132
YES
i < N?
3134
NO
OUTPUT: BW, EXCESS
3136
3100B

FIG. 31B



N = NUMBER OF SESSIONS
3152A
BW = ARRAY OF BANDWIDTHS PER SESSION
3152B
OBTAIN/RETRIEVE INPUTS
L = ARRAY OF B/W LIMIT PER SESSION
3150
3152C
EXCESS
i = 0
3152D
3154
YES
BW[i] < LIMIT[i]
SET THE i-TH BANDWIDTH BW[i]+=(W[I]/S)*EXCESS
3156
3158
NO
i = i+1
3160
YES
i < N?
3162
NO
OUTPUT: BW
3164
3100C

FIG. 31C



N = NUMBER OF SESSIONS
3172B
B = TOTAL SLICE B/W
3172A
L = LIMITS
3172D
OBTAIN/RETRIEVE INPUTS
3170
W = WEIGHTS
3172C
S = SUM OF WEIGHTS
SUM(L) = SUM LIMIT
3174
NO
YES
B' = B
3180
SUM(L) < B?
3176
B' = SUM(L)
3178
RUN PROCESS 3100A "BITRATE ALLOCATION IN A 5G VIRTUAL PIPE/SLICE BASED ON WEIGHTS" USING SLICE BANDWIDTH B', NUMBER OF SESSIONS N, AND WEIGHTS W; THE SUB-METHOD COMPUTED BANDWIDTH ARRAY BW
3182
RUN PROCESS 3100B "CLIPPING BANDWIDTHS ACCORDING TO LIMITS AND COMPUTING EXCESS"; THE SUB-METHOD UPDATES BANDWIDTH ARRAY BW, AND COMPUTES BANDWIDTH EXCESS
3184
EXCESS > 0?
3186
YES
NO
RUN PROCESS 3100C "DISTRIBUTING BANDWIDTH EXCESS ACROSS SESSIONS ACCORDING TO THEIR WEIGHTS" GIVEN WEIGHTS W, BANDWIDTH ARRAY BW, EXCESS, AND LIMITS; THE SUB-METHOD UPDATES BANDWIDTH ARRAY BW
3188
ALLOCATE BANDWIDTHS ACCORDING TO ARRAY BW: SESSION i IS ALLOCATED TO BANDWIDTH BW[i]
3190
3100D

FIG. 31D

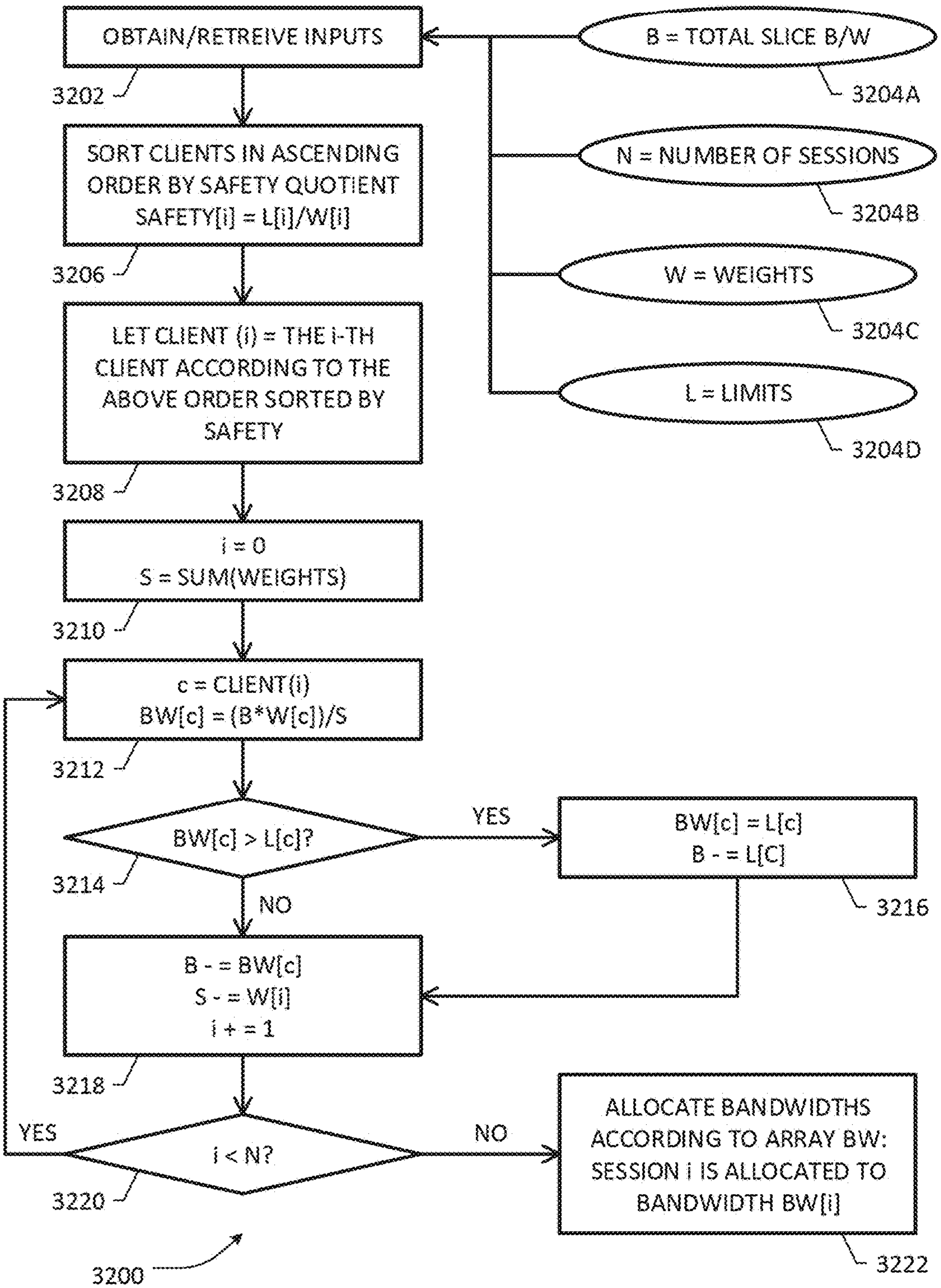

OBTAIN/RETREIVE INPUTS
3202
B = TOTAL SLICE B/W
3204A
N = NUMBER OF SESSIONS
3204B
W = WEIGHTS
3204C
L = LIMITS
3204D
SORT CLIENTS IN ASCENDING ORDER BY SAFETY QUOTIENT SAFETY[i] = L[i]/W[i]
3206
LET CLIENT (i) = THE i-TH CLIENT ACCORDING TO THE ABOVE ORDER SORTED BY SAFETY
3208
i = 0
S = SUM(WEIGHTS)
3210
c = CLIENT(i)
BW[c] = (B*W[c])/S
3212
BW[c] > L[c]?
3214
YES
BW[c] = L[c]
B - = L[C]
3216
NO
B - = BW[c]
S - = W[i]
i + = 1
3218
i < N?
3220
YES
NO
ALLOCATE BANDWIDTHS ACCORDING TO ARRAY BW: SESSION i IS ALLOCATED TO BANDWIDTH BW[i]
3222
3200

FIG. 32

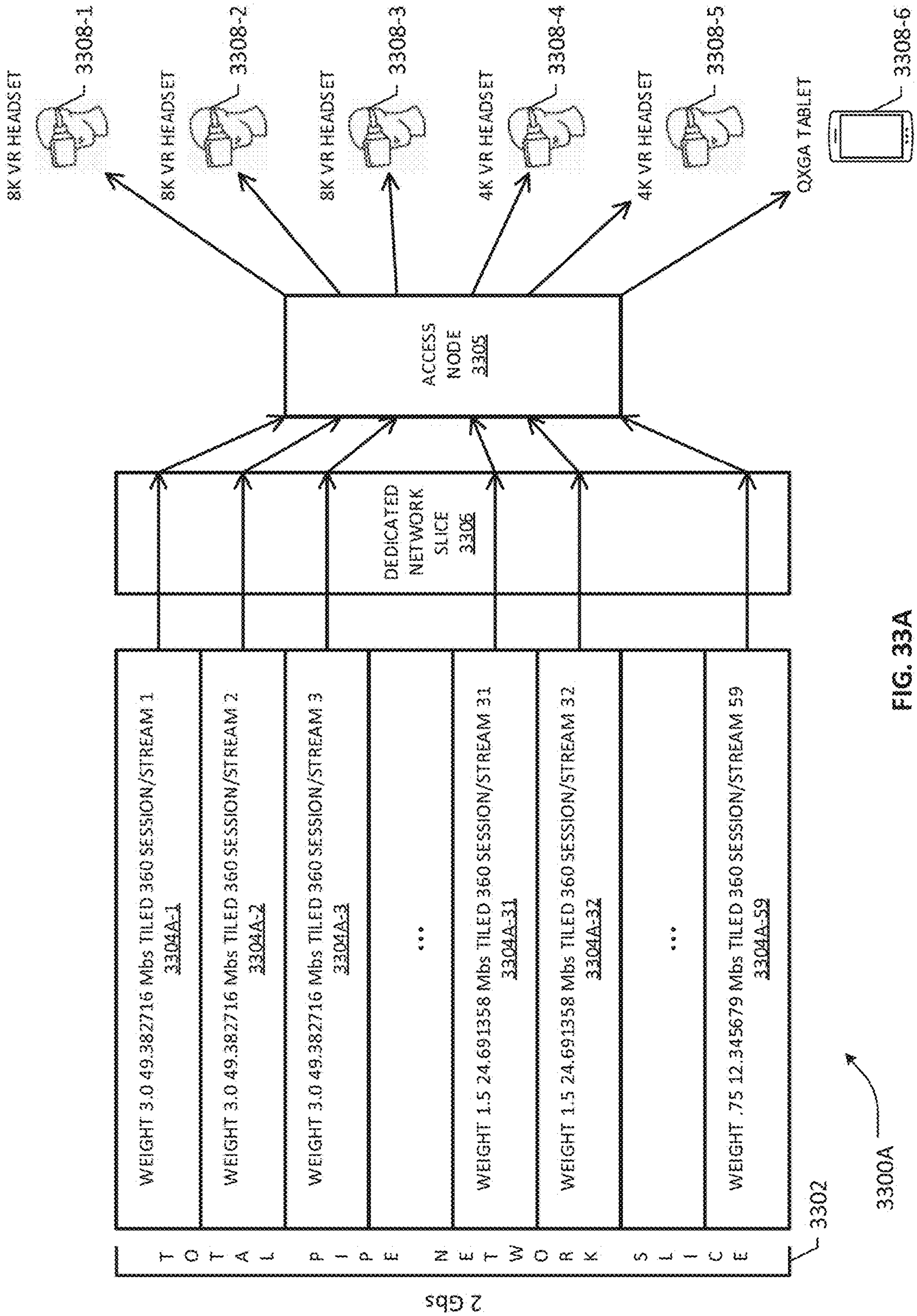

8K VR HEADSET
3308-1
8K VR HEADSET
3308-2
8K VR HEADSET
3308-3
4K VR HEADSET
3308-4
4K VR HEADSET
3308-5
QXGA TABLET
3308-6
ACCESS NODE 3305
DEDICATED NETWORK SLICE 3306
WEIGHT 3.0 49.382716 Mbs TILED 360 SESSION/STREAM 1 3304A-1
WEIGHT 3.0 49.382716 Mbs TILED 360 SESSION/STREAM 2 3304A-2
WEIGHT 3.0 49.382716 Mbs TILED 360 SESSION/STREAM 3 3304A-3
...
WEIGHT 1.5 24.691358 Mbs TILED 360 SESSION/STREAM 31 3304A-31
WEIGHT 1.5 24.691358 Mbs TILED 360 SESSION/STREAM 32 3304A-32
...
WEIGHT .75 12.345679 Mbs TILED 360 SESSION/STREAM 59 3304A-59
TOTAL PIPE NETWORK SLICE
3302
2 Gbs
3300A

FIG. 33A

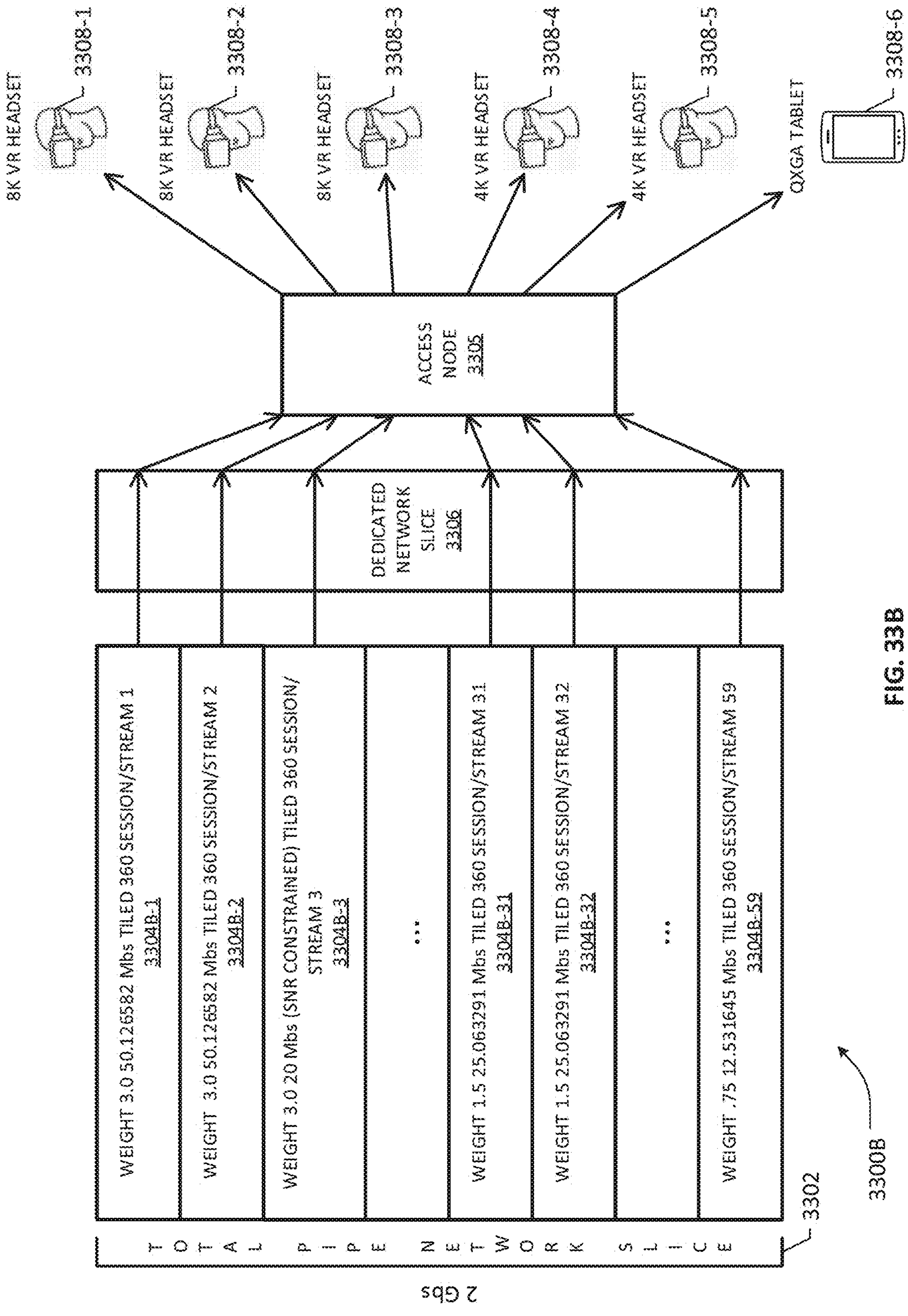

8K VR HEADSET
3308-1
8K VR HEADSET
3308-2
8K VR HEADSET
3308-3
4K VR HEADSET
3308-4
4K VR HEADSET
3308-5
QXGA TABLET
3308-6
ACCESS NODE 3305
DEDICATED NETWORK SLICE 3306
WEIGHT 3.0 50.126582 Mbs TILED 360 SESSION/STREAM 1 3304B-1
WEIGHT 3.0 50.126582 Mbs TILED 360 SESSION/STREAM 2 3304B-2
WEIGHT 3.0 20 Mbs (SNR CONSTRAINED) TILED 360 SESSION/ STREAM 3 3304B-3
...
WEIGHT 1.5 25.063291 Mbs TILED 360 SESSION/STREAM 31 3304B-31
WEIGHT 1.5 25.063291 Mbs TILED 360 SESSION/STREAM 32 3304B-32
...
WEIGHT .75 12.531645 Mbs TILED 360 SESSION/STREAM 59 3304B-59
TOTAL PIPE NETWORK SLICE
2 Gbs
3302
3300B

FIG. 33B

SYSTEM AND METHOD FOR PROVIDING MULTIPLE 360° IMMERSIVE VIDEO SESSIONS IN A NETWORK

TECHNICAL FIELD

The present disclosure generally relates to communication networks. More particularly, and not by way of any limitation, the present disclosure is directed to a system and method for providing multiple 360° immersive video sessions in a network.

BACKGROUND

The introduction of virtual reality has brought new applications to the forefront in addition to improving several existing technologies. One improvement over existing technologies can be seen in the case of 360° immersive video, also variously referred to as panoramic video, 360-degree video or 360 video, and the like.

360-degree video offers the user with an immersive "being there" experience. The increased immersion of virtual reality can easily be applied to video, providing superior user experience over the traditional video that is projected on flat surfaces. The popularity of navigable 360-degree video systems has also grown with the advent of omnidirectional capturing systems and interactive displaying systems, such as head-mounted displays (HMDs) or headsets. However, content providers have been contending with bandwidth constrained network environments to deliver 360-degree video content in an efficient way in order to ensure a satisfactory viewing experience because 360-degree video assets are ultra high resolution spherical videos, which contain an omnidirectional view of the scenes requiring enormous amounts of data.

Current 360 video headsets are 2K-resolution display devices, covering 1K per eye. In order to achieve the best quality in the headset, a typical network requires sending an 8K 360 video stream to the device. It is known that video compression allows efficient utilization of bandwidth in a media streaming network by reducing the number of bits to represent a picture. Whereas advances in video compression technologies continue to grow apace, several lacunae remain in the field of 360 video delivery and display with respect to efficiently managing bandwidth in today's network architectures, especially where multiple 360-degree video sessions are to be effectuated in a dedicated bandwidth environment, requiring further innovation as will be set forth hereinbelow.

SUMMARY

The present patent disclosure is broadly directed to systems, methods, apparatuses, devices, and associated non-transitory computer-readable media and network architecture for effectuating optimized 360° immersive video viewing experiences including, inter alia, where multisession 360° immersive video services are provided in a network portion having a dedicated or reserved bandwidth capacity. In one aspect, certain embodiments are directed to optimized encoding schemes that may be implemented in an arrangement involving encoding of source video streams into tile-encoded streams having different qualities. In a further aspect, certain embodiments are directed to utilizing user gaze vector information in determining tile weights based on the location of tiles with respect to a user's viewport. In still further aspects, example embodiments are directed to tile selection and bandwidth annealing schemes where bandwidth-optimized tiles are selected responsive to gaze vector information that may be stitched into a multiplexed coded video sequence for providing an enhanced viewing experience.

In one example embodiment, a bandwidth optimization method in a network environment configured to support multiple 360° video sessions is disclosed. The claimed method comprises, inter alia, receiving or otherwise obtaining an indication with respect to a bandwidth capacity dedicated to a network portion configured for supporting a plurality of 360° video sessions in a shared/reserved bandwidth pipe. In one variation, the network portion having a dedicated bandwidth may comprise a network slice. In another variation, the network portion having a dedicated bandwidth may comprise a non-sliced network infrastructure configured to support a dedicated bandwidth pipe for effectuating multisession 360° video services. Each video session may be configured to serve a corresponding client device for playing an immersive video asset, wherein each video frame comprises an array of tiles projected on a 3-dimensional (3D) display environment viewed by a user operating the client device. A plurality of weights corresponding to the plurality of 360° video sessions are received or otherwise obtained, wherein each weight is operative to indicate a priority value associated with a respective video session. A plurality of bitrate caps corresponding to the plurality of 360° video sessions are received or otherwise obtained, each bitrate cap indicating a bandwidth limit associated with a respective video session. Responsive to the bandwidth capacity indication, total number of the plurality of 360° video sessions and corresponding pluralities of weights and bitrate caps, a respective bandwidth allocation is determined for each 360° video session without violating the bitrate cap (e.g., a minimum acceptable quality) of the 360° video session. Thereafter, the respective bandwidth allocations may be provided to a tile selection process operative to select tiles for a corresponding video session on a frame-by-frame basis from a plurality of tile-encoded bitrate representations of a media input stream associated with the video session, wherein each tile-encoded bitrate representation is provided with a separate video quality.

In one variation, for each session, bitrates of at least a portion of the selected tiles for a video session are optimized based on gaze vector information received from the client device associated with the video session. In one variation, the bitrate caps need not necessarily comprise a strict limit; rather, they may be configured as suggestions or recommendations, with a degree of variability. In one variation, the plurality of tile-encoded bitrate representations for a particular session of the plurality of 360° video sessions are generated as a plurality of phase-encoded bitstreams based on at least one of High Efficiency Video Coding (HEVC) H.265 compression, Alliance for Open Media (AOMedia) Video 1 (AV1) compression and H.266/Versatile Video Coding (VVC) compression. In another variation, at least a portion of the tile-encoded bitrate representations corresponding to a particular session of the plurality of 360° video sessions are generated as block-intra-encoded bitstreams based on at least one of HEVC/H.265 compression, AV1 compression and H.266/VVC compression.

In another variation, an embodiment of a bandwidth optimization method may further comprise: determining a surplus in individual bandwidth allocations assigned to respective 360° video sessions based on the corresponding bitrate caps; obtaining an overall excess amount of bandwidth available in the shared bandwidth pipe based on the

surplus; and distributing the overall excess amount of bandwidth to the plurality of 360° video sessions according to the respective weights associated with the plurality of 360° video sessions.

In a still further variation, the gaze vector information associated with a video session of the plurality of 360° video sessions may be obtained by tracking an orientation of the user's headset associated with the client device for displaying the particular immersive video asset. In another variation, the gaze vector information may be obtained by tracking a movement of the user's eyeballs with respect to different portions of the 3D display environment while the particular immersive video asset is being displayed. Regardless of how the gaze vectors are obtained, they may comprise, without limitation, normalized/non-normalized Cartesian coordinate vectors, normalized/non-normalized spherical coordinate vectors, or vectors defined in a suitable 3D geometrical coordinate system, and the like.

In another aspect, an embodiment of a video server system operative in association with a multisession 360-degree immersive video streaming network is disclosed, which comprises one or more processors and one or more persistent memory modules having program instructions thereon that are configured to perform, when executed by the processor(s) of the system, one or more embodiments of the bandwidth allocation/optimization processes set forth in the present patent application.

In a still further aspect, an embodiment of a client device operative in a 360-degree immersive video environment is disclosed, which comprises one or more processors, a media player having user controls, and one or more persistent memory modules having program instructions thereon that are configured to perform when executed by the processors an embodiment of a device-based video playback method as set forth hereinbelow.

In an example implementation, a client device may be configured to operate with various types of coded bitstreams having different qualities that may be generated based on at least one on at least one of HEVC/H.265 compression, AV1 compression and H.266/VVC compression.

In still further aspects, one or more embodiments of a non-transitory tangible computer-readable medium or distributed media containing computer-executable program instructions or code portions stored thereon are disclosed for performing one or more embodiments of the methods of the present invention when executed by a processor entity of a network node, apparatus, system, network element, subscriber device, and the like, mutatis mutandis. Further features of the various embodiments are as claimed in the dependent claims.

Example embodiments disclosed herein provide several benefits in a multisession immersive media consumption environment including but not limited to the ability to optimize video quality even in a bandwidth-constrained network. As gaze vector information corresponding to each session is provided to a tile selection mechanism operating in association with a bitrate-capped session pipe, viewport-optimized selection of tiles with highest bitrate quality provides for a superior viewing experience in more realistic viewing scenarios where the users' gaze may be constantly changing. Embodiments are also particularly advantageous in a sliced network architecture since an example implementation allows for the ability to add policy management on 360° immersive video flows based on content, device, subscriber policies, etc., for dynamically (re)allocating the bandwidth within the slice so as to maximize capacity utilization.

In additional and/or related aspects, tiled video frames of a 360° immersive video asset may be advantageously assembled with a subset of tiles optimized for higher quality viewports based on gaze vector information and allocated bandwidth. Because the frames are selectively viewport-optimized, transport of high quality multiplexed streams is possible even in bandwidth-constrained environments without sacrificing the viewing experience. Example embodiments may be advantageously configured such that the highest quality tiles will always be delivered in the direct view, with controlled degrading qualities across multiple areas farther from the direct field of vision with the lowest quality in the area that is in the diametrically opposite direction of where the user is looking. Accordingly, when a stream is delivered to the device, the user always gets the highest video QoE in the area that they are directly looking at. Further, when the user moves their head, mid-GOP switching facilitated by some example embodiments allows receiving high quality tiles as quickly as possible with minimal latency. With the tiles encoded for gradual refresh, when a user changes their field of vision, example embodiments can further reduce the latency of the video as the size of the video buffer may be minimized by sending several high quality tiles in the initial upgrade of the next frame to deliver. Over the course of the next several frames, an example embodiment gradually increases the quality of the remaining tiles until the quality of tiles is reached based on the current field of vision and allowed bandwidth.

Additional benefits and advantages of the embodiments will be apparent in view of the following description and accompanying Figures.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the present disclosure are illustrated by way of example, and not by way of limitation, in the Figures of the accompanying drawings in which like references indicate similar elements. It should be noted that different references to "an" or "one" embodiment in this disclosure are not necessarily to the same embodiment, and such references may mean at least one. Further, when a particular feature, structure, or characteristic is described in connection with an embodiment, it is submitted that it is within the knowledge of one skilled in the art to effect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described.

The accompanying drawings are incorporated into and form a part of the specification to illustrate one or more exemplary embodiments of the present disclosure. Various advantages and features of the disclosure will be understood from the following Detailed Description taken in connection with the appended claims and with reference to the attached drawing Figures in which:

FIG. **1** depicts a generalized example network environment wherein one or more embodiments of the present invention may be practiced for providing 360° immersive video over a variety of network configurations;

FIG. **2** depicts an example network architecture comprising a portion of the environment shown in FIG. **1** for facilitating optimized tile encoding of 360° immersive video according to an example embodiment;

FIG. **3** depicts a block diagram of an example tile encoder that may be provided as part of a media preparation and/or processing system configured to operate in an arrangement of the network architecture of FIG. **2**;

FIGS. 4A-4C illustrate example video frames containing one or more slices and/or tiles per each frame in an example encoder arrangement;

FIG. 5 is a flowchart illustrative of various blocks, steps and/or acts of a media preparation/processing method that may be (re)combined in one or more arrangements, with or without blocks, steps and/or acts of additional flowcharts of the present disclosure, for facilitating optimized 360° immersive video according to one or more embodiments of the present invention;

FIG. 6 is illustrative of various blocks, steps and/or acts of an example encoding arrangement involving either a Phased Encoding (PE) scheme or a Block-Intra Encoding (BIE) scheme that may be implemented as part of the example media preparation/processing method of FIG. 5 according to one or more embodiments of the present invention;

FIG. 7 is a flowchart illustrative of a BIE scheme according to an example embodiment of the present invention;

FIG. 8A is a flowchart illustrative of a process for configuring a BIE scheme in a tiled encoding arrangement according to an example embodiment of the present invention;

FIG. 8B is a flowchart illustrative of additional blocks, steps and/or acts in an example BIE scheme according to an example embodiment of the present invention;

FIG. 9 is a flowchart illustrative of a PE scheme according to an example embodiment of the present invention;

FIG. 10A is a flowchart illustrative of a process for configuring a PE scheme in a tiled encoding arrangement according to an example embodiment of the present invention;

FIG. 10B is a flowchart illustrative of additional blocks, steps and/or acts in an example PE scheme according to an example embodiment of the present invention;

FIG. 11 depicts a plurality of coded bitstreams having different qualities generated by a BIE-based tiled encoder system in an example embodiment;

FIG. 12 depicts a plurality of coded bitstreams having different phases for a particular bitrate representation generated by a PE-based tiled encoder system in an example embodiment;

FIG. 13A is illustrative of various blocks, steps and/or acts of an example tile stitching scheme involving BIE-based tiled streams according to an embodiment of the present invention;

FIG. 13B is illustrative of various blocks, steps and/or acts of an example tile stitching scheme involving PE-based tiled streams according to an embodiment of the present invention;

FIG. 13C is a flowchart illustrative of additional blocks, steps and/or acts with respect to an example tile stitching scheme according to an embodiment of the present invention;

FIG. 14 is illustrative of a 360° video frame comprising tiles selected from coded bitstreams having different qualities or QPs in accordance with an example embodiment of the present invention;

FIGS. 15A and 15B are flowcharts illustrative of various blocks, steps and/or acts of a method that may be (re) combined in one or more arrangements, with or without blocks, steps and/or acts of additional flowcharts of the present disclosure, for facilitating optimized tile selection based on weights associated with user gaze in a 360° immersive video viewing environment according to one or more embodiments of the present invention;

FIGS. 16A and 16B are illustrative of example geometrical arrangements for facilitating determination of angular separation between a user's gaze direction and tile positions in a tile encoded frame;

FIG. 16C is illustrative of an example 360° immersive video viewing environment for purposes of one or more embodiments of the present invention;

FIG. 17A is a flowchart illustrative of additional blocks, steps and/or acts with respect to an example 360° immersive video optimization process according to an example embodiment of the present invention;

FIG. 17B is a flowchart illustrative of additional blocks, steps and/or acts with respect to further aspects of an example 360° immersive video optimization process according to an example embodiment of the present invention;

FIG. 18A depicts an example video frame having tile locations with different weights determined in accordance with an embodiment of the present invention;

FIG. 18B depicts an example device buffer with frames of differently-coded viewport tiles;

FIGS. 18C and 18D illustrate 3D viewing spaces where tile qualities are distributed based on user gaze direction;

FIG. 19 is a flowchart illustrative of various blocks, steps and/or acts of a tile selection and bandwidth annealing process that may be (re)combined in one or more arrangements of a media preparation/processing method, with or without blocks, steps and/or acts of additional flowcharts of the present disclosure, according to one or more embodiments of the present invention;

FIG. 20 is a flowchart illustrative of additional blocks, steps and/or acts with respect to an example tile selection and bandwidth annealing process according to an embodiment of the present invention;

FIGS. 21A and 21B are flowcharts illustrative of additional blocks, steps and/or acts with respect to further aspects of a tile selection and bandwidth annealing process according to an example embodiment of the present invention;

FIG. 22 is illustrative of a transmit buffer model configuration for use a tile selection and bandwidth annealing arrangement according to an example embodiment of the present invention;

FIG. 23 depicts an arrangement where a UE device may be configured to perform certain aspects of 360° immersive video optimization for purposes of an embodiment of the present patent disclosure;

FIG. 24 depicts a block diagram of an apparatus that may be (re)configured and/or (re)arranged as a platform, node or element to effectuate one or more aspects of 360° immersive video processing, preparation and optimization according to an embodiment of the present invention;

FIG. 25 depicts is a block diagram of an example UE device with additional details for purposes of an embodiment of the present patent disclosure;

FIG. 26 depicts an example network environment for providing multiple 360° immersive video sessions according to an embodiment of the present invention;

FIG. 27 depicts an example network slice architecture configured to support multisession 360° immersive video services according to an embodiment of the present invention;

FIG. 28 depicts an example network slice arrangement operative with a multisession network slice video optimization system in accordance with an embodiment;

FIG. 29 depicts an example 360° immersive video network layout with different latencies according to an embodiment of the present invention;

FIGS. **30**A-**30**C depict flowcharts illustrative of various blocks, steps and/or acts relative to bandwidth optimization in a network slice configured to support multiple 360° immersive video sessions according to one or more embodiments of the present invention;

FIGS. **31**A-**31**D depict flowcharts illustrative of various blocks, steps and/or acts relative to additional details in a bandwidth optimization scheme for supporting multiple 360° immersive video sessions according to one or more embodiments of the present invention;

FIG. **32** depicts a flowchart illustrative of various blocks, steps and/or acts relative to bandwidth optimization in a network slice configured to support multiple 360° immersive video sessions according to yet another example embodiment of the present invention; and

FIGS. **33**A and **33**B depict example virtual pipe and weighted bandwidth allocation scenarios according to an embodiment of the present invention.

DETAILED DESCRIPTION

In the description herein for embodiments of the present invention, numerous specific details are provided, such as examples of components and/or methods, to provide a thorough understanding of embodiments of the present invention. One skilled in the relevant art will recognize, however, that an embodiment of the invention can be practiced without one or more of the specific details, or with other apparatus, systems, assemblies, methods, components, materials, parts, and/or the like. In other instances, well-known structures, materials, or operations are not specifically shown or described in detail to avoid obscuring aspects of embodiments of the present invention. Accordingly, it will be appreciated by one skilled in the art that the embodiments of the present disclosure may be practiced without such specific components. It should be further recognized that those of ordinary skill in the art, with the aid of the Detailed Description set forth herein and taking reference to the accompanying drawings, will be able to make and use one or more embodiments without undue experimentation.

Additionally, terms such as "coupled" and "connected," along with their derivatives, may be used in the following description, claims, or both. It should be understood that these terms are not necessarily intended as synonyms for each other. "Coupled" may be used to indicate that two or more elements, which may or may not be in direct physical or electrical contact with each other, co-operate or interact with each other. "Connected" may be used to indicate the establishment of communication, i.e., a communicative relationship, between two or more elements that are coupled with each other. Further, in one or more example embodiments set forth herein, generally speaking, an element, component or module may be configured to perform a function if the element may be programmed for performing or otherwise structurally arranged to perform that function.

As used herein, a network element, node or subsystem may be comprised of one or more pieces of service network equipment, including hardware and software that communicatively interconnects other equipment on a network (e.g., other network elements, end stations, etc.), and is adapted to host one or more applications or services, either in a virtualized/non-virtualized environment, with respect to a plurality of subscribers and associated user equipment (UE) nodes that are operative to receive/consume content in a media distribution network where media content assets may be distributed and delivered using stream-based or file-based mechanisms. As such, some network elements may be disposed in a wireless radio network environment whereas other network elements may be disposed in a public packet-switched network infrastructure, including or otherwise involving suitable content delivery network (CDN) infrastructure that may comprise public, private, or mixed CDNs. Further, suitable network elements including one or more embodiments set forth herein may involve terrestrial and/or satellite broadband delivery infrastructures, e.g., a Digital Subscriber Line (DSL) network architecture, a Data Over Cable Service Interface Specification (DOCSIS)-compliant Cable Modem Termination System (CMTS) architecture, switched digital video (SDV) network architecture, a Hybrid Fiber-Coaxial (HFC) network architecture, a suitable satellite access network architecture or a broadband wireless access network architecture over cellular and/or WiFi connectivity. Accordingly, some network elements may comprise "multiple services network elements" that provide support for multiple network-based functions (e.g., 360° immersive A/V media preparation, delivery policy management, session control, QoS policy enforcement, bandwidth scheduling management, content provider priority policy management, streaming policy management, and the like), in addition to providing support for multiple application services (e.g., data and multimedia applications including 360° immersive video assets (also referred to as 360-degree video assets or simply 360 video assets) in varying qualities or definitions). Example subscriber end stations or client devices may comprise various devices, tethered or untethered, that may consume or deliver media content assets using streaming and/or file-based downloading technologies, which may involve some type of rate adaptation in certain embodiments. Illustrative client devices or UE devices may therefore include any device configured to execute, inter alia, one or more client applications for receiving, recording, storing, and/or decoding/rendering 360 video content, live media and/or static/on-demand media, which may comprise Virtual Reality (VR) media, Augmented Reality (AR) media, Mixed Reality (MR) media, from one or more content providers, e.g., via a broadband access network, using HTTP, HTTPS, RTP, and the like. Accordingly, such client devices may include Next Generation IP-based STBs, networked TVs, personal/digital video recorders (PVR/DVRs), networked media projectors, portable laptops, netbooks, palm tops, tablets, smartphones, multimedia/video phones, mobile/wireless user equipment, portable media players, portable gaming systems or consoles (such as the WHO, Play Station 3®, etc.) operating in concert with 3D display devices and the like, which may access or consume 360-degree content/services provided via a suitable media distribution network wherein a bandwidth and Quality of Experience (QoE) scheme may be provided in accordance with to one or more embodiments set forth herein.

One or more embodiments of the present patent disclosure may be implemented using different combinations of software, firmware, and/or hardware. Thus, one or more of the techniques shown in the Figures (e.g., flowcharts) may be implemented using code and data stored and executed on one or more electronic devices or nodes (e.g., a subscriber client device or end station, a network element, etc.). Such electronic devices may store and communicate (internally and/or with other electronic devices over a network) code and data using computer-readable media, such as non-transitory computer-readable storage media (e.g., magnetic disks, optical disks, random access memory, read-only memory, flash memory devices, phase-change memory, etc.), transitory computer-readable transmission media (e.g., electrical, optical, acoustical or other form of propagated signals—such as carrier waves, infrared signals, digital signals), etc. In addition, such network elements may typically include a set of one or more processors coupled to one or more other components, such as one or more storage devices (e.g., non-transitory machine-readable storage media) as well as storage database(s), user input/output devices (e.g., a keyboard, a touch screen, a pointing device, and/or a display), and network connections for effectuating signaling and/or bearer media transmission. The coupling of the set of processors and other components may be typically through one or more buses and bridges (also termed as bus controllers), arranged in any known (e.g., symmetric/shared multiprocessing) or heretofore unknown architectures. Thus, the storage device or component of a given electronic device or network element may be configured to store code and/or data for execution on one or more processors of that element, node or electronic device for purposes of implementing one or more techniques of the present disclosure.

Referring now to the drawings and more particularly to FIG. **1**, depicted therein is a generalized example network environment **100** where one or more embodiments of the present invention may be practiced for providing immersive video distributed over a variety of configurations for consumption by one or more viewing devices. An example video source/capture system **102** is illustrative of any arrangement configured to record, generate, read, decode, provide, or otherwise obtain media that is renderable for 360° viewing in myriad client device environments, which may include tethered or untethered devices, standalone pieces of equipment, subscriber premises equipment, gaming equipment, and/or equipment operating in paired combination(s) with 3D display devices, etc., operating with a variety of access/connection technologies, as noted elsewhere in the present patent application. By way of illustration, computers/displays **144**, which may be associated with head-mounted displays (HMDs) or headsets **142**, which may in turn also be associated with portable devices such as tablets, smartphones, phablets, gaming devices, etc., collectively shown as devices **140**, and the like, generally shown as client devices **138**, may be configured to decode and render various types of 360° video content that may be encoded and bandwidth-optimized according to the teachings of the present invention as will be set forth in additional detail further below. In one embodiment, example 360° immersive video source/capture system **102** may comprise one or more high-definition cameras (e.g., 4K, 8K, etc.), including omnidirectional or panoramic cameras, etc. or a video storage that may be configured to provide source video streams in a number of ways. Depending on the configuration and level of integration with respect to video preprocessing, output streams from example 360° immersive video source/capture 102 may be provided as streams compatible with one or more interfaces, High Definition Multimedia Interface (HDMI), Serial Digital Interface (SDI), High Definition SDI (HD-SDI), or other formats, which may comprise unstitched or stitched streams, with or without projection-mapping, and with or without source video encoding. For example, unstitched source streams without projection mapping **104**A may be provided to a video stitcher **106** that combines streams covering overlapping angles into a stitched stream **108**. In another embodiment, video source steams may comprise stitched HDMI/SDI/HD-DSI streams **104**B. Also, there may be other processing of captured video that may involve les correction. Where the streams are not projection-mapped, a projection mapping system **110** is operative to generate a projection-mapped steam **114** from stitched streams **10413**/**108** using a suitable map projection scheme, e.g., a spherical image projection including, without limitation, equirectangular projection, Cube Map projection, Equi-Angular Cubemap (EAC) projection, Pyramid projection, Fish-Eye projection, etc. In a still further embodiment, video streams may comprise stitched and projection-mapped streams **104**C that may be provided to a source video encoding module **112** operative to effectuate one or more encoding or compression schemes depending on implementation, e.g., including, without limitation, H.264 or Advanced Video Coding (MPEG-4 AVC), High Efficiency Video Coding (HEVC) or H.265 (MPEG-H Part 2), H.262 (MPEG-2), H.264 (MPEG-4, Part 2), Alliance for Open Media (AOMedia) Video 1 (AV1), H.266, Versatile Video Coding (WC), Future Video Coding (FVC), etc., where some of the schemes may or may not include tile encoding and/or may or may not adaptive bitrate (ABR) transcoding. In one arrangement, projection-mapped streams from the projection mapping system **110** may also be provided to the encoder system **112** for effectuating appropriate video compression. Depending on the configuration and the level of integration with respect to preprocessing in media preparation, a tiled encoder/transcoder **120** is advantageously provided in accordance with the teachings of the present invention to process uncompressed video streams received from the projection mapping system **110** (video streams **114**), compressed video streams received from the encoder system **112** (video streams **116**), or video streams **104**C from the video source/capture system **102**. As will be set forth in further detail below, tiled encoder/transcoder **120**, whose functionality may be integrated with the encoder system **112** and/or the projection mapping system **110** in some embodiments, is operative to generate encoded streams of multiple bitrate representations of an input video stream corresponding to a 360° immersive video asset or program, wherein each bitrate representation having a certain video quality level may be encoded to contain frames with appropriately modified tile, frame and/or slice data to facilitate bandwidth-optimized 360° video distribution. A tiled packager **122** is operative to package the encoded streams from encoder/transcoder **120** for storage **124** and provide associated manifest files **126** describing tile groupings, tile locations, media types and related characteristics of the encoded streams. As will be further set forth below, a tile selection and stream generation system **132** is operative to select appropriate tiles responsive to control inputs and generate a multiplexed video output stream that may be delivered by a delivery server **134** associated with an access network **136** serving the viewing devices **138**. In an example implementation, delivery of the multiplexed video streams to end users may be effectuated based on a number of protocols, e.g., HTTP/S, chunked HTTP/S, RTP/RTCP, etc., over a variety of network infrastructures, as noted elsewhere in the present patent application.

Skilled artisans will recognize that the foregoing generalized example network environment **100** may be implemented in a hierarchical network architecture, with various aspects of media capture and preparation, including, e.g., source stream stitching, projection mapping, source media compression, tiled/ABR encoding/transcoding, packaging, etc., as well as distributing/uploading and edge node processes taking place in different network portions disposed at different hierarchical levels, involving one or more operators, content delivery networks (CDNs), edge networks, and the like. Further, in some implementations, at least some of the foregoing apparatuses and processes may be cloud-based. In some arrangements, a CDN can be a large distributed system of servers deployed in multiple data centers connected to the Internet or other public/private communications network. A CDN can be a managed or unmanaged network, and can also be a federation of managed or unmanaged networks.

An example embodiment of a media server/source system operatively associated within the foregoing example network environment may therefore be configured, e.g., as a global headend, to accept media content from live sources and/or static file sources, e.g., online content providers such as Hulu®, Netflix®, YouTube®, or Amazon® Prime, as well as VOD catalog or content providers or studios such as, e.g., Disney, Warner, Sony, etc. Media content from live sources may comprise live programming captured relative to any type of event, e.g., sporting/entertainment/gaming events, concerts, live TV shows, live news broadcasting sources, such as, for instance, national broadcasters (e.g., NBC, ABC, etc.) as well as cable broadcaster channels like Time Warner channels of CNN, ESPN, CNBC, etc., and local broadcasters, etc., including any secondary media insertions such as advertisement media channels.

Without limitation, an example network architecture **200** (which may form a portion of the environment shown in FIG. **1**) is depicted in FIG. **2** for facilitating optimized tile encoding of immersive video according to an embodiment of the present invention. A media input stream **202** is illustrative of a video stream corresponding to a 360° video asset that may be suitably stitched, projection-mapped and/or encoded as set forth in FIG. **1**, which may be distributed, uploaded or otherwise provided to a CDN origin server **204** associated with an operator content delivery network **206**. Broadly, media input stream **202** may comprise a stream corresponding to least one of live TV content, IPTV content, time-shifted (TS) TV content, place-shifted (PS) TV content, gaming content, Video on Demand (VOD) content, VR/AR/MR content, networked digital video recorder (nDVR) content, and the like, or any content that is (pre)processed for 360-degree viewing experience. A CDN edge server **208** coupled to CDN **206** may be configured to receive the uploaded media stream(s) **202** corresponding to respective video assets, which may be stored in suitable database(s) (not specifically shown). A tiled encoder **210**, which may be configured to operate in compliance with a standard codec scheme (e.g., HEVC, AV1, etc.) is operative to generate a plurality of tiled adaptive bitrate streams **212** where each stream may comprise tiles of a specific resolution, bitrate, and pixel sizes (depending on aspect ratios). By way of illustration, steams **212** may comprise one or more 32K streams (30730 horizontal pixels×17280 vertical pixels), 16K streams (15360 horizontal pixels×8640 vertical pixels), one or more 8K streams (7680 horizontal pixels×4320 vertical pixels), one or more 4K streams (3840 horizontal pixels×2160 vertical pixels), one or more HD streams (1920 horizontal pixels×1080 vertical pixels), one or more 720p streams (1280 horizontal pixels×720 vertical pixels), etc., wherein higher resolution streams may be encoded at higher bitrate ranges while lower resolution streams may be encoded at lower bitrate ranges. For instance, 32K streams may be encoded in the range of 800-1000 Mbits/s (or Mbps), 16K streams may be encoded in the range of 200-300 Mbps, 8K streams may be encoded in the range of 80 to 100 Mbps, and so on to 720p streams encoded in the range of 1.2 to 3 Mbps. Further, tiled adaptive bitrate streams **212**, also referred to as tile-encoded bitstreams, may comprise frames having a suitable number of tiles per frame, e.g., 128 tiles for 4K, depending on the scheme being employed.

In one arrangement, tiled encoder **210** may be configured to generate tiled-encoded bitstreams as a plurality of phase-encoded streams for each bitrate representation of the media input stream **202**, wherein each phase-encoded stream for a particular bitrate representation is provided with a specialized frame at a particular location in the Group-of-Pictures (GOP) structure of the stream depending on the phase as will be set forth in additional detail further below. This scheme of encoding may be referred to as Phased Encoding (PE) scheme with respect to certain embodiments of present invention. In another arrangement, tiled encoder **210** may be configured to generate a pair of tiled-encoded bitstreams, e.g., a first and a second tile-encoded bitstream, for each bitrate representation of the media input stream **202**, wherein a first encoded bitstream may comprise a regular or standard tile-coded bitstream generated according to a known or heretofore unknown coding scheme and a second encoded bitstream may be coded such that a specialized frame is provided at each location in a GOP structure, as will be set forth in additional further below. This scheme of encoding may be referred to as Block-Intra Encoding (BIE) or All-Intra Encoding (AIE) scheme with respect to certain embodiments of the present invention.

Regardless of whether PE-coding scheme or BIE-coding scheme is used, a packager **214** is operative to package the tile-encoded bitstreams **212** and generate suitable manifest files describing characteristics of tile groupings per frame for each tile-encoded bitstream, e.g., tile location, slice header information, various types of metadata including picture timing, color space information, video parametric information, etc., which may be stored at a suitable packaged media storage facility **240**, along with suitable stream manifests **241**. A network edge node **216** including a video optimization system **215** comprising a plurality of modules or subsystems is operative in association with a video back office system **238** for effectuating a 360° immersive video session with a premises device **236** of subscriber premises **234** that is served by a managed bandwidth pipe **232** effectuated via a suitable access network (e.g., a DSL/DOCSIS network portion having suitable infrastructure that may include, e.g., routers, DSLAM/CMTS elements, etc., or suitable 3G/4G/5G radio access network elements, including fixed wireless infrastructure in certain implementations, and the like), generally represented by node or element **230**.

In one arrangement, video optimization system **215** may comprise a tile selection subsystem **218** that is operative responsive to bandwidth annealing and QoE management policies, as well as user gaze vector information, inter alia, to provide tiles **220** selected from different video quality bitstreams to a tile combining and stream generation subsystem **222**. Multiplexed video frames with tiles from different bitstreams **224** may be provided to a delivery service **226** for facilitating the transmission of muxed tile stream **228** to the downstream infrastructure **230**. Broadly, when a user request **250** for a 360° immersive video session is generated, it is processed by the video back office system **238** and forwarded to the video optimization system **215** via a message **252** for obtaining a session ID and associated location information for the requested 360° media. Responsive to a response message **251** from the video optimization system **215**, the video back office system **238** is operative to provide a response **248** including appropriate URL information for the media and a session ID to the requesting device **236**. User gaze information (which may be a default setting initially) and associated session ID information may be provided to the infrastructure element **230** as a message **246**, which may be propagated to the video optimization system **215** as message **254**. Also, the infrastructure element **230** is operative to provide a dynamic bandwidth allocation message **254** that includes the session ID information to the video optimization system **215** in a related or separate process. As noted previously, tile selection subsystem **218** may be configured to operate in response to control messages relative to bandwidth allocation, user gaze vector information, or both, for selecting tiles having different video qualities, which may be combined or stitched into frames in order to generate a muxed tile-encoded video output stream. In one arrangement, the tile combining and stream generation subsystem **222** may be provided as part of the video optimization system **215** during video stream delivery. In another arrangement, the tile stitching may be effectuated during playout on the client side (e.g., at the client device **236** or some other premises equipment associated therewith) rather than on the server side. In this arrangement, a client-side stitching functionality is operative to receive the selected tiles and perform the necessary stitching in order to generate a stitched stream to be decoded and rendered. Various embodiments relative to the foregoing processes, subsystems and components will be set forth in further detail in the following sections.

FIG. **3** depicts a block diagram of an example tile encoder **300** that may be provided as part of a media preparation and/or processing system configured to operate within an arrangement of the network architecture of FIG. **2**. Without limitation, example tile encoder **300** will be set forth below that may be configured to effectuate either a PE coding scheme or a BIE coding scheme for generating multi-bitrate video streams having different qualities with respect to each media asset while being compliant with known or heretofore unknown standard codec schemes, such as, e.g., H.265, H.266, VVC, AV1, etc., that are compatible with tile encoding. Broadly, in one embodiment, a specialized frame (or, somewhat synonymously, a picture) is generated that is encoded as a predictive-coded (P) picture or frame (i.e., having a header identifying it as a P-frame) but only contains coding blocks or units that are encoded as intra-coded blocks or units (i.e., I-blocks). In another embodiment, a specialized frame may comprise a frame identified as a bi-predictive (B) frame but contains only I-blocks. For purposes of the present patent application, these specialized frames are referred to as "block-intra" frames or "X" frames, where media image data of all the blocks are forced to be coded as intra-coded (i.e., no temporal estimation or prediction).

For purposes of example embodiments herein, a GOP structure is a group of successive pictures in a coded video stream, which specifies the order in which intra- and inter-frames are arranged. Each coded video stream comprises successive GOPs, from which the visible frames may be generated. Generally, a GOP structure may contain the following picture types: (1) I-picture or I-frame (intra coded picture)—a picture that is coded independently of all other pictures. Each GOP begins (in decoding order) with this type of picture. (2) P-picture or P-frame (predictive coded picture)—contains motion-compensated difference information relative to previously decoded pictures. In older designs such as MEPG-1, H.262/MPEG-2 and H.263, each P picture can only reference one picture, and that picture must precede the P picture in display order as well as in decoding order and must be an I or P picture. These constraints do not apply in the newer standards such as, e.g., H.264/MPEG-4 AVC, H.265/HEVC, etc. (3) B-picture or B-frame (bi-predictive coded picture or bidirectionally predictive coded picture)—which contains difference information from the preceding and following I- or P-frame within a GOP, and contains motion-compensated difference information relative to previously decoded pictures. In older designs such as MPEG-1 and H.262/MPEG-2, each B-picture can only reference two pictures, the one which precedes the B picture in display order and the one which follows, and all referenced pictures must be I or P pictures. These constraints do not apply in the newer standards such as, e.g., H.264/MPEG-4 AVC, H.265/HEVC, etc. (4) D-picture or D-frame (DC direct coded picture)—serves as a fast-access representation of a picture for loss robustness or fast-forward in certain types of video (e.g., MPEG-1 video).

In general, an I-frame indicates the beginning of a GOP. Afterwards several P and B frames may follow. The I-frames contain the full image and do not require any additional information to reconstruct it. Typically, encoders use GOP structures that cause each I-frame to be a "clean random access point," such that decoding can start cleanly on an I-frame and any errors within the GOP structure are corrected after processing a correct I-frame. The GOP structure is often referred by two numbers, for example, M=3, N=12. The first number tells the distance between two anchor frames (I or P). The second one tells the distance between two full images (I-frames), which is the GOP size. For the example M=3, N=12, the GOP structure is {IBBPBBPBBPBBI}. Instead of the M parameter the maximal count of B-frames between two consecutive anchor frames can be used. For example, in a sequence with pattern {IBBBBPBBBBPBBBBI}, the GOP size is equal to 15 (length between two I frames) and distance between two anchor frames (M value) is 5 (length between I and P frames or length between two consecutive P Frames).

While a typical GOP starts with an I-frame, some embodiments herein provide a structure where a GOP may commence with an X-frame instead, in addition to placing the X-frames at specific locations or replacing the P- and/or B-frames in the GOP structure as will be set forth in additional detail further below.

Skilled artisans will recognize that depending on codec implementation, a picture or frame may be partitioned into a number of ways at different levels of granularity, for example, to facilitate, inter alia, coding efficiency, parallel processing, etc. In one arrangement, a frame may be partitioned into a number of coding tree units (CTUs), each containing certain number of luma coding tree blocks (CTBs) and chroma CTBs, which in turn may comprise multiple coding blocks (CBs). A frame may be split into one or more slices, each being a spatially distinct region of a frame that may be encoded separately from any other region in the same frame and identified with a slice header. In general, slices are self-contained and contain a sequence of CTUs that are processed in the order of a raster scan, wherein slices can be coded as I-slices, P-slices, or B-slices similar to I-frames, P-frames, or B-frames, respectively. In one arrangement, slices may be used to effectuate resynchronization to minimize data losses, and may contain a varying number of CTUs per slice depending on the activity in a video scene. FIG. **4**A illustrates an example video frame **400**A containing a plurality of slices **402-1** to **402**-N, where an example slice **402**-N contains a number of CTUs **404**.

In addition to slices, an encoding scheme may also define a number of tiles per frame, which may also be configured to be self-contained and independently decodable rectangular or square regions of a picture, based on vertical and horizontal partitioning to form a grid, in order to facilitate parallel processing at the encode and decode stages. In one variant, the self-contained and independently decodable tiles may use temporal prediction from the co-located tiles of previously encoded pictures or frames. Multiple tiles may share header information by being contained in the same slice, where a tile may comprise a certain number of CTUs. It is not a requirement that each tile include the same number of CTUs. Accordingly, in one arrangement, the tiles of a frame may have different sizes. If a frame contains a single slice, the tiles of the frame will therefore have the same slice header and picture header information. In another arrangement, a frame may include one or more slices, each slice containing one or more tiles, and each tile in turn containing one or more CTUs. FIG. **4**B illustrates an example video frame **400**B containing a plurality of CTUs, organized into a matrix or array of tiles **406-1** to **406**-N, wherein each tile is shown as a square with 4 CTUs **408** in a 2×2 configuration. By way of further illustration, an example 4K video frame **400**C according to HEVC is shown in FIG. **4**C, which may comprise an array of 3840 horizontal pixels by 2160 vertical pixels, that is partitioned into 16 columns and 8 rows, thereby resulting in 128 tiles. As noted earlier, these tiles may not be necessarily sized equally within the frame **400**C.

For purposes of the present patent application, because video frames may be partitioned in numerous ways and at different levels, the terms "coding tree unit", "coding tree block", "coding unit", "macro block", or "block" or terms of similar import will be generally treated as an abstract unit of coding that may applied with respect to a tile, slice and/or frame without limitation to any particular video compression standard or technology.

Returning to FIG. **3**, example tile encoder **300** may be configured with respect to a PE- or BIE-based scheme to generate an X-frame where it is coded as a P- or B-frame having the corresponding header but with individual slices and/or tiles that are intra-coded, i.e., as I-slices and/or I-tiles, comprising only I-blocks. In other words, the X-frames may have header information of a P- or B-frame (or P-slice or B-slice if only one slice per frame is provided), but all the media image data is intra-coded as data of an I-frame. Remaining frames of a video sequence may be encoded normally in accordance with a known or heretofore unknown scheme as noted previously. Accordingly, a general coder control **306** may be configured to select between the PE and BIE schemes **308**, **310** for providing appropriate control signals and/or parameters to remaining components and structures of a frontend portion **302** of the tile encoder so as to force the encoding of the special frames as needed according to a particular implementation of the PE or BIE schemes with respect to one or more input video signals **304**. In general, each picture in a PE scheme is either encoded as a regular I-frame (e.g., for the first picture in a sequence) or as an X-frame for those input pictures that match the phase/period, and as regular P- or B-frames for all other pictures of the video sequence, as will be described in further detail below. With respect to the BIE scheme, a BIE-coded sequence may be provided where X-frames are provided for all P- and B-frames of a GOP structure of the sequence. Accordingly, an intra/inter selection block **312** is configured such that intra-picture estimation/prediction **316** is always active and used for all blocks of a picture. Likewise, motion compensation and estimation **318** may be disabled since all blocks are intra-coded for an X-frame. Remaining blocks comprising transform, scaling and quantization **314**, inverse transform **320**, filter control **322**, deblocking and sample adaptive offset (SAO) filtering **324**, decoded picture buffer **326** may remain unaffected in an example embodiment depending on the tile encoder implementation. General control data **328**, quantized transform coefficients data **330**, intra prediction and filter control data **332** and motion data **334** may be provided to a header formatter and entropy encoder **336** (e.g., a context-adaptive binary arithmetic coding (CABAC) engine) for generating one or more coded bitstreams **338** corresponding to each bitrate representation of the video asset. As noted previously, the coded bitstreams **338** may be provided to a tiled packager (not shown in this FIG. **3**) for packaging and manifest generation to facilitate (pre)provisioning of the assets at appropriate downstream network locations.

FIG. **6** is illustrative of various blocks, steps and/or acts of an example encoding arrangement **600** involving either a PE scheme or a BIE scheme that may be implemented as part of an example media preparation/processing according to an embodiment of the present invention. At block **604**, a video source stream **602** is received, which may be unencoded, encoded, stitched, projection-mapped, or otherwise pre-processed, as previously described. At block **606**, a determination may be made where PE or BIE is selected. A mode selector in a tile encoder system, e.g., tile encoder **300** in FIG. **3**, may be appropriately configured in response. Upon selecting PE, the video source stream **602** may be encoded/transcoded into a plurality of streams with different qualities and/or bitrates, each being encoded with tiles, as set forth at block **608**. Each quality or bitrate stream is phase-encoded to generate a plurality of PE streams **610**. By way of illustration, reference numeral **614-1** refers to quality information relating to a set of phase-encoded streams **612-1** with corresponding phases **615-1** to **615**-P (depending on where an X-frame is placed in GOP structure, where P is the GOP size), all PE streams having a Quantization Parameter (QP) setting of 30 and/or a bitrate of around 7.0 Mbits/s, which may be indicative of a lower end of quality. In similar fashion, reference numeral **614**-N refers to quality information relating to a set of phase-encoded steams **612**-N with corresponding phases **615-1** to **615**-P, all having a QP setting of 16 and/or a bitrate of around 105.6 Mbits/s, which may be indicative of a higher end of quality.

If BIE (also referred to as All-Intra Encoding, as noted elsewhere in the present patent application) is selected, the video source stream **602** may be encoded/transcoded into a plurality of streams with varying qualities and/or bitrates (block **616**). In one example embodiment, each of the streams may be tile-encoded using a standard coding scheme (e.g., HEVC, AV1, etc.) to generate normal or regular tile-encoded streams **618**. Similar to the discussion above with respect to the phased-tiled streams **610**, reference numeral **622-1** refers by way of illustration to quality information relating to a regular tile-encoded stream **620-1** having a QP setting of 30 and/or a bitrate of around 7.0 Mbits/s, which may be indicative of a lower end of quality. Likewise, reference numeral **622**-N refers to quality information relating to a regular tile-encoded steam **620**-N having a QP setting value of 16 and/or a bitrate of around 105.6 Mbits/s, which may be indicative of a higher quality stream.

Additionally, the video source stream **602** is also encoded/transcoded into a plurality of streams with corresponding qualities and/or bitrates (block **617**) where each stream is tile-encoded such that all frames of it's GOP structure are provided as X frames. By way of illustration, reference numeral **632** refers to a plurality of BIE-coded and tiled streams, wherein quality information **636-1** having a QP setting of 30 and/or a bitrate of around 7.0 Mbits/s (also sometimes abbreviated as Mbs or Mb/s) relates to a lower quality BIE-coded tiled stream **634-1** while quality information **636**-N of QP setting of 16 and/or a bitrate of around 105.6 Mbits/s relates to a higher quality BIE-coded tiled stream **634**-N.

Skilled artisans will recognize upon reference hereto that when an encoder is configured with a target QP, the bitrate of an encoded bitstream is somewhat averaged over the course of the bitstream. For instance, if a QP of 10 is targeted in a source encoding scheme, it is possible that a low bitrate may be seen in areas of no motion (e.g., resulting in 4 Mbs). In areas of high motion, it is possible that the bitrate could shoot up to 200 Mbs. Thus, in an example encoding scheme that targets specific QPs as set forth in the foregoing, the bitrates of the output steams could be variable over a range. Accordingly, it should be appreciated that the bitrates shown in association with the QPs of PE or BIE streams in FIG. **6** are generally indicative of average bitrates over a course of time. As will be seen further below, when QPs are targeted in an encoding scheme (with varying bitrates correspondingly), certain embodiments of the present invention relating to tile selection may be configured to select tiles and fit them in accordance with an overall allocated bitrate with respect to a particular 360-degree immersive video session. In an additional or alternative embodiment, an example encoder may be configured to generate coded bitstreams having specific target bitrates instead of target QPs. In such an arrangement, while an output bitstream may maintain a particular bitrate, the QP values may vary, however. An embodiment of tile selection may therefore select tiles based on video qualities that may be controlled by different encoding parameters and settings and fit them accordingly to optimize the allocated bandwidth. For purposes of the present patent application, the terms "quality", "video quality", and terms of similar import with respect to coded bitstreams or bitrate representations may be broadly related to and/or based on QPs, bitrates, other indicia. Thus, embodiments relating to PE/BIE encoding, tile selection, stitching and the like set forth herein based on targeted QPs are also equally applicable to bitstreams having targeted bitrates, mutatis mutandis.

Accordingly, it should be understood by the reader that although certain examples and portions of the description within this disclosure are provided assuming the use of a fixed quantization (QP) value per stream, streams in practice may contain QP values that vary between pictures and within a picture as noted above. An encoder according to an embodiment of the present invention may control its output bitrate by the means of a rate-control or the like, and thereby change the QP value between pictures. An encoder may also encode pictures within one stream using varying QP values to optimize the visual quality of the stream. Within one picture, the QP value may change between blocks using e.g., adaptive quantization mechanisms to optimize the visual quality as known in the art. The use of "QP" in phrases within this disclosure such as e.g., but not limited to, "encoded with that QP", "video of different QP values", "generated videos with different QP values", "stream having a QP value of N", "QP value of the video stream" should be understood as a way of characterizing streams such that a stream associated with a lower QP value is of higher bitrate and higher quality than one associated with a higher QP value, and not that the QP is kept static for each block in a stream.

It should be further appreciated that adaptive bitrate encoding and tile encoding of media assets may be integrated within an apparatus as part of a content preparation system in one example embodiment such that various types of encoding and/or transcoding may take place in different sequences and/or in parallel processes. Further, additional functionalities such as projection-mapping, source stream stitching, packaging, etc., may also be combined or otherwise integrated with the tile-coding/transcoding schemes of the present patent application depending on implementation.

FIG. **5** is a flowchart illustrative of various blocks, steps and/or acts of a method **500** that may be (re)combined in one or more arrangements, with or without blocks, steps and/or acts of additional flowcharts of the present disclosure, for facilitating optimized 360° immersive video according to one or more embodiments of the present invention. At block **502**, various operations relative to media capture and preprocessing of a media input stream for immersive video, e.g., source stream stitching, encoding, projection mapping, etc. may be effectuated. At block **504**, adaptive-friendly bitrate encoding/transcoding of the preprocessed media input stream into multiple bitrate representations or streams having different video qualities (e.g., with varying QP values) may be effectuated in association with a tiled encoding scheme. As noted previously, either a PE-based coding process (block **506**B) or a BIE-based coding process (block **506**A) may be configured to generate coded bitstream output. It should be noted that the processes of blocks **504** and **506**A/B could be executed as single encoding operations such that the adaptive-friendly bitrate encode/transcoding of block **504** is done using either a PE scheme (block **506**A) or a BIE scheme (block **506**B) using a single encode process. Thereafter, the coded bitstreams may be packaged (block **508**) and distributed to appropriate network edge locations (block **510**) for delivery and consumption by clients using suitable end user equipment. When a user request for a particular media asset is received and processed, a tile selection process based on control inputs e.g., transmission conditions, bandwidth allocation and/or gaze vector input, etc., may be effectuated for selecting tiles from different bitrate representations (i.e., different qualities) of the media asset (block **512**). A stream generation process may be effectuated for stitching the selected tiles into frames as an output video stream to be delivered to the requesting client device (block **514**).

Skilled artisans will recognize that at least a portion of the foregoing steps, acts or operations may comprise media preparation and (pre)provisioning with respect to one or more 360° immersive video assets distributed in a network environment or architecture illustrated in FIGS. **1** and **2** described above. Turning to FIG. **7**, additional details regarding a BIE scheme **700** according to an example embodiment of the present invention are set forth. At blocks **702** and **704**, a media input stream relative to a 360° immersive video asset is received and processed to generate multiple bitrate representations having different/separate qualities, e.g., each video quality related to or controlled by a corresponding targeted QP value used for each bitrate representation and/or targeted bitrate, or other indicia of respective quality. Each bitrate representation is coded into a first coded bitstream comprising a plurality of frames with a specific GOP structure, wherein each GOP starts with an I-frame followed by a set of frames including at least one P-frame or B-frame (block **706**). Further, each bitrate representation is encoded into a second coded bitstream comprising a plurality of frames with a GOP structure that has a size coextensive with the size of the GOP structure of the first coded bitstream, wherein each GOP of the second coded bitstream starts with an I-frame followed by a plurality of X-frames, each X-frame being coded with a slice/picture header of a P- or B-frame and comprising intra-coded media image data only (i.e., similar to an I-frame of the GOP), as set forth at block **708**. As noted previously, the first coded bitstream and the second coded bitstream may be encoded as respective tile-encoded streams using any tile-compatible compression scheme, wherein each frame of the tile-encoded bitstream comprises an array of tiles organized into at least one slice per frame, each tile comprising a portion of the media data of the frame formed as a number of coding units, blocks or trees. One skilled in the art will recognize that in one implementation, processes of block **704** and **706** may be performed in a single encode process as noted previously in respect of blocks **504** and **506**A/B of FIG. **5**. For instance, in practice, a single process encoding/transcoding would be desirable to minimize the computational complexity and minimize degradations introduced by tandem or cascaded encoding.

FIG. **11** depicts a plurality of coded bitstreams **1100** having different qualities or QPs generated by a BIE-based tiled encoder system in an example embodiment. Reference numerals **1102-1** to **1102**-N refer to N streams or bitrate representations having corresponding qualities or QPs. A normally encoded tiled stream **1104**A corresponding to a particular bitrate representation, e.g., QP-N **1102**-N, is illustrated with a GOP structure **1106**A having four frames, starting with an I-frame followed by three P-frames. Corresponding BIE-coded stream **1104**B has a GOP structure **1106**B, which is also illustrated with four frames, starting with an I-frame but followed by three X-frames.

FIG. **8**A is a flowchart illustrative of a process **800**A for configuring a BIE scheme in a tiled encoding arrangement according to an example embodiment of the present invention. Without limitation, example process **800**A will be described in reference to configuring an HEVC scheme for performing BIE based on modifying certain parameters, although other schemes may also be applied for purposes herein.

In general, an embodiment of a BIE configuration method may be configured to receive or obtain as input a source video stream for 360° immersive video and a list of output video qualities (e.g., a list of QP values, such as {QP1=16, QP2=18, QP3=20, QP4=22, QP5=24, QP6=26, QP7=28, QP8=30, or other indicia based on targeted bitrates}). Accordingly, without limitation, for every output video quality (e.g., every QP value), two video streams may be encoded—a regular/standard HEVC video with that QP or quality, and a Block-Intra HEVC video with that QP/quality—as noted previously. In order to be able at a later time (e.g., shortly before decoding) to stitch tiles from different qualities into the same bitstream, the encoding phase of an embodiment provides that all the video streams have the same base_qp (defined below), while the actual difference between the videos of different QP values may be effectuated by means of qp_delta (defined below) from the base QP. For example, a setting of base_qp=22 may be configured, wherein the parametric values base_qp=22 and qp_delta=−6 may be used to achieve QP=16. In general, these two parameters relate to setting the quality (QP) of a video stream. Recall that all the generated videos with the different qp values need to have the same base_qp, while different QP values may be achieved by using qp_delta from the base_qp. This requirement may be imposed based on one particular time instance. That is, if pictures in a bitstream are numbered, then any two pictures from two bitstreams that are used as input for stitching with the same numbers must use the same base_qp value in one arrangement. For purposes of the present invention, "base_qp" may be described as follows: the i^{th} frame (for every i=1 to N, where N is the total number of frames in a video sequence) in all the encoded versions or bitrate representations of the same video will have the same slice QP value. In other words, slice QP is the base_qp. Although slice QP may be set as the same value in all the generated streams, it can vary over time. For purposes of the present invention, the parameter delta_qp may be described as follows: by assigning a given qp_delta, the first block in each tile that signals QP is configured to signal the delta_qp (that amount of variance from the base QP). It may be noted that there could be a deblocking mismatch after stitching in some embodiments.

Another parameter that may be defined for purposes of the present invention is ROI (Region of Interest), which determines an area of a frame where the tiles can be independently encoded so that the subset of the bitstream corresponding to the ROI can be easily extracted and reconstituted into another bitstream. As noted above, in order to later stitch videos of different QPs, it is desirable to utilize the functionality of base_qp and delta_qp. This is supported for example when using HEVC ROI encoding functionality in one illustrative implementation. Accordingly, when encoding with ROI in an embodiment, the base_qp parameter for the slice QP headers may be defined, in addition to defining an ROI grid (independently defined from the grid/array of the tiles of a frame) such that the area of the grid in the i^{th} row and j^{th} column in the ROI grid gets its own delta_qp. Generally, this allows an embodiment to assign different delta_qp to different areas of the ROI grid, whereby selective delta_qp values may be used for purposes of the present invention. For example, to achieve a given desired QP (say QP=16), the base_qp may be defined (say base_qp=22) using the regular qp parameter, and then by using the ROI grid, all the targeted areas may be assigned a delta_qp of −6, thus effectively achieving a QP of 16 for all the tiles in the ROI grid.

In one embodiment, the content at different qualities may be encoded using the same base_qp (slice QP) for a particular frame. For each quality of that frame, a specific desired QP may be set, wherein the delta_qp syntax elements may be used so that all blocks (or alternatively, as many blocks as possible or desired) of that frame are encoded with that desired QP. Additional aspects of a BIE configuration scheme based on HEVC may be set forth as follows.

The encoder may be set to use tiled encoding. During setup, this may be effectuated by setting an appropriate flag for tiled-encoding, as well as configuring a specific grid structure of the tile (e.g., as shown in FIG. **4**C). By way of illustration, the encoder may be configured to provide a 16×8 grid structure of tiles, resulting with 128 tiles in each frame, for a 4K video input.

The encoder may be configured to disable temporal motion vectors prediction. Although an example BIE scheme does not use MVs (motion vectors), temporal motion vector prediction (TMVP) settings may need to be identical across streams to enable stitching at a later time. This configuration is optional, in that an embodiment of BIE may be practiced without disabling TMVP.

Also, many other elements of the slice headers may be configured to be identical across streams. For example, elements such as the number of reference pictures to use, the reference picture set, what reference pictures to use for L0, the Picture Parameter Set (PPS) to use, the picture order count, SAO parameters, etc. Further, it is also required that the decoding order is the same for all bitstreams that are to be used as input for bitstream switching. Skilled artisans will recognize upon reference hereto that a variety of slice header elements may be configured accordingly in an example BIE implementation.

Since a slice uses a single PPS id codeword to identify what PPS to use and the PPS references one single Sequence Parameter Set (SPS), all encodings may be done using identical PPS and SPS id values in an example embodiment. Likewise, many syntax elements in the SPSs and PPSs may also be configured to be identical for the multiple encodings. Although not a necessary requirement, an example BIE embodiment may therefore be configured such that the encodings are effectuated using identical SPSs and PPSs. However, it is strictly necessary that some elements in the SPS and PPS are identical in certain arrangements.

Returning to FIG. **8**A, example BIE configuration process **800**A may commence with initializing a mode selector of an encoder to select BIE for encoding an input video stream as set forth hereinabove (block **802**). At block **804**, the encoder may be configured to use tiles in a particular grid or array arrangement for each frame. At block **806**, the base_qp parameter may be written in all slice QP headers of the encoded streams. To encode streams in different qualities (while having the same base_qp), a qp_delta parameter may be configured as described above with respect to each stream based on target QPs (block **808**). For example, to achieve a target QP of 22 for a particular stream, a qp_delta of −10 may be configured where base_qp is 32. At noted before, it is required that all pictures with the same picture number to be used as an input for stitching must use the same base_qp value. Thus, in one embodiment, it is not a necessary requirement to set the same base_qp parameter in all the stream headers. Spatial motion vector prediction may be configured such that it is restricted within the tile only (block **810**). That is, motion vectors are not allowed to cross tile boundaries in an example embodiment (i.e., only intra-tile prediction is allowed). This means that motion vectors are set such that no sample outside the boundaries of any co-located tile is read during motion compensation interpolation of the blocks inside a tile. An ROI grid may be configured for the encoder such that it uses the qp_delta information for encoding a particular stream with respect to a specific region of the frames (block **812**). Further, TMVP may also be disabled (block **814**) in an example BIE configuration process as set forth above.

It should be noted that whereas the foregoing BIE configuration process **800**A uses certain parameters, additional or alternative embodiments may be practiced where a BIE scheme may be configured to utilize other parameters in addition to and/or in lieu of the parameters exemplified in the flowchart of FIG. **8**A.

FIG. **8**B is a flowchart illustrative of additional blocks, steps and/or acts in an example BIE scheme **800**B according to an embodiment of the present invention. In general, during BIE-based tiled coding, an encoder may be configured to effectuate several decisions. During encoding of tiles which are part of a P-frame, the encoder decides whether or not a particular tile should be encoded using any motion vector and depend on the previous frame, or whether it should be encoded in "intra" mode, where the tile is self-contained and is not dependent on any previous frame (i.e., does not use prediction from any previous frame). As noted previously, during encoding of P-frames in BIE, the encoder is forced to encode all blocks using intra modes. At block **834**, a video input **832** is received for encoding. At block **836**, a tiled encoder is configured for implementing a BIE process as set forth above. For each frame of the video input, an iterative process may be implemented to effectuate appropriate coding decisions on a frame-by-frame basis, which commences with a determination as to whether the video sequence has reached its end (block **838**). If the end is not reached (i.e., there remain frames in the video sequence requiring processing), a next frame is obtained (block **840**). If the frame is determined to be a first frame of a GOP structure (block **842**), it is encoded as a regular I-frame (block **854**) and the process flow returns to obtaining a next frame (block **840**). Otherwise, it is encoded as a P-frame (block **844**). For each slice of the P-frame, it is encoded or provided with a P-slice header (block **846**). For each block or any other suitable coding unit of the P-slice, the encoder is configured to encode the image data in intra mode (block **848**). Thereafter, the process flow returns to determine whether all the frames have been processed (block **838**). If so, the encoding of the video sequence is finalized (block **850**), which may be provided as BIE-tiled bitstream to a downstream entity (e.g., a packaging system), as set forth at block **852**. An alternative arrangement, B-frames may be used in lieu of P-frames for generating X-frames as noted elsewhere in the patent application. Accordingly, blocks **844**, **846** and may be suitably modified to support this arrangement.

In a further embodiment of the present invention, X-frames may be used once in each GOP (instead of multiple times as in BIE) based on a PE scheme as noted previously. Essentially, PE-based tiled encoding involves a process and apparatus for generating a stream where all the frames have P-slice headers, except for the first frame which is an I-frame, while periodically there is an X-frame (i.e., BIE-frame or AIE-frame), where all blocks are intra-encoded but the slice headers are of P-slices (or B-slices where B-frames are also encoded in a sequence). In general, all slices of any two pictures that are to be potentially used as inputs to stitching need to have the same slice type, slice qp, as well as a number of other settings in the slice header and PPS. In contrast with the BIE scheme set forth above, where all the frames of a GOP are X-frames except for the first one, an embodiment of a PE scheme is configured to provide X-frames only at select frame locations depending on two parameters: period (which is the size of the GOP, i.e., the number of frames in the GOP) and phase (which is an integer in the range {0 to [period-1]}. Frame locations where the X-frames appear in a PE scheme may be determined as follows. Let N be the total number of frames in a stream. The first frame is encoded as an I-frame. For a frame at i^{th} position, $2 \leq i \leq N$, it is encoded as a regular P-frame if {i Mod (period) ≠ phase}; and otherwise (that is, {i Mod (period)=phase}), the frame is encoded as an X-frame (with P-slice headers and all blocks encoded in intra-mode, independent of previous frames). It should be noted that an example PE scheme may provide as many phase-encoded streams for each quality/bitrate representation of the media input as there are frame locations in a GOP (i.e., GOP size).

By using P- or B-slice headers rather than I-slice headers in X-frames for purposes of the present invention, several advantages may be realized in an exemplary embodiment, including but not limited to facilitating mid-GOP switching in a user viewing environment. Assume the user is watching a 360° immersive video program or content where the directly-gazed field of view (FoV) is in high quality (i.e., lower QP) and the user moves his head in the middle of the GOP. The user now sees a low quality video (higher QP) in their new field of view or viewport. The server can send an I-frame of a high quality (low QP) at the beginning of the next GOP, but this introduces a significant latency, as it would take time until the high quality I-frame of the next GOP for the viewport will be presented. It is desirable to receive or obtain an I-frame of the new field of view that is encoded at high quality as soon as possible while in the middle of the GOP. But it is not feasible to just put an I-frame as is in the middle of the GOP in a conventional immersive video viewing environment. By generating an X-frame (i.e., Block-Intra coded frame or All-Intra coded frame) and transmitting it in the middle of the GOP (at any frame location in a GOP structure, for instance), an embodiment of the present invention is thus effectively able to upgrade the quality of the field of view similar to the effect if an I-frame is presented in the middle of the GOP with high quality tiles. By providing P-slice headers in AI- or BI-encoded frames (i.e., AIE/BIE frames or X-frames), an embodiment of the present invention therefore allows a frame having high quality data in a region of interest (ROI) of FoV to be used in the middle of the GOP.

Further, in a tile encoding scheme where a frame is partitioned into tiles and slices, an embodiment of the present invention involving X-frames enables mixing tiles in a single output compressed frame, where some tiles may use spatial or temporal prediction (i.e., inter-picture prediction) and some tiles may use only spatial prediction (e.g., comprising intra-coded blocks only). The tiles consisting of intra-coded blocks only may originate from an X-frame. In the context of the present patent application, the terms "mixing", "muxing", "stitching", "splicing" or terms of similar import with respect to output stream generation may refer to means and methods to concatenate one compressed tile (e.g., tile A) with another compressed tile (e.g., tile B) to form a part of the bitstream representing a single output frame, where tile A and tile B may originate from separate encodings of the content, which will be set forth in additional detail further below.

One of the advantages of a PE scheme relates to overcoming the issue of drift that may be present in a BIE scheme (i.e., drift elimination or reduction). It should be appreciated that while BIE allows replacement a P-frame of the previous viewport with an X-frame of the new viewport, the following frames are regular P-frames of the new viewport that are encoded with predictions made to previous frames. Thus, when a P-frame is replaced with an X-frame and then a following frame uses this X-frame for prediction instead of the original frames of the regular bitstream, there is a potential for drift where prediction errors may accumulate. On the other hand, in phased encoding, the generated stream uses the X-frame at position=<phase>+i*<period> for the prediction of the following P-frames, and thus the situation where a P-frame uses for prediction a different frame than the frame used during encoding is avoided. Hence, there is no prediction error presented due to predicting from a frame that is different than the frame generated during the encoding, and accordingly, any potential drift due to this type of prediction errors is avoided. However, the PE scheme may require a larger amount of storage since storage of the P-frames that follow the X-frames in the GOP is required.

Further, an embodiment of a PE scheme may be advantageously utilized to facilitate gradual refreshing of frames whereby lower latency is achieved during playout by selecting only a subset of the tiles to upgrade their quality and send their appropriate phased-encoded tiles. While in an embodiment of a BIE scheme, a P-frame is replaced with an X-frame, in a gradual refresh frame annealing scheme the PE-coded streams may be used to replace selected tiles with the corresponding tiles taken from the appropriate PE-coded stream. On the other hand, in another embodiment, a BIE scheme may can also advantageously operate on a tile-by-tile basis. With respect to a PE-based embodiment, accordingly, if period is P and frame number is X, one can obtain the corresponding phase by the following equation: Phase={X Mod P}. Thus, during delivery or playout of a coded video sequence, assume that a certain tile T is selected for upgrade to QP quality q in frame X, then one can replace selected tile (in frame X and following frames until the next upgrade/downgrade of T or viewport change) with the tile T from the stream with phase that satisfies the following relationship: Phase={X Mod P} at QP=q. Thereafter, the co-located tiles in the frames following frame X that belong to the same GOP are replaced by the corresponding co-located tiles from the same PE-encoded stream. It should be appreciated that the advantages of concatenating tiles from different streams when a user changes gaze direction are similar to the scenario set forth above where the user changes his gaze during mid-GOP. Identical slice QPs are used for switching/replacing the tiles because if two input tiles are encoded with different actual QP and were encoded with a single slice per picture, then if the slice QP differs, it would not be possible for the QPs of tiles in the output stream to be correct without low-level rewrite of the stream. Additional details regarding gradual refresh frame annealing and tile selection will be set forth further below in reference to additional embodiments of the present patent application.

A potential disadvantage with respect to PE may be that it requires more storage, since an input stream is encoded in many phases, potentially resulting in as many streams as the GOP size (rather than just two streams as in BIE). This disadvantage may be traded off against the advantage of reduced latency without drift in an example implementation. For fastest quality change response, the number of phases may be set equal to the size of the GOP, i.e., the period P, but an example embodiment may provide a trade-off of using fewer phases and consuming less storage while the latency of the quality upgrade may be longer, since tile upgrades will only be done on the next phase.

FIG. **9** is a flowchart illustrative of a PE scheme **900** according to an example embodiment of the present invention. At block **902**, a media input stream corresponding to a 360° immersive video asset may be received. As before, a plurality of bitrate representations of the media input stream may be generated, each bitrate representation having a separate video quality, e.g., related to or controlled by a corresponding targeted QP value used for the bitrate representation and/or targeted bitrate, or other indicia of respective quality (block **904**). Each bitrate representation controlled by a corresponding QP is encoded into a plurality of phase-encoded bitstreams, each phase-encoded bitstream that belongs to a particular bitrate representation comprising a number (N) of frames with a specific GOP structure having a GOP size (p), wherein the number of the plurality of phase-encoded bitstreams equals the GOP size. In one arrangement, the GOP size, i.e., p>1. For each p^{th} phase-encoded bitstream, N frames are encoded as follows: (i) at least a first frame is encoded as an intra-coded (I) frame; and (ii) a frame at a frame position i, for 2≤i≤N, that satisfies the relationship {i Mod (GOP size)}=p is encoded as an X-frame having a slice header of a P-frame and comprising blocks of only intra-coded media image data only (i.e., similar to an I-frame). Otherwise, that frame is encoded as a regular P-frame having media data of a predictive-coded frame with a P-slice header (block **906**). In some arrangement, the P-frames may also contain intra-coded data. Where B-frames are also encoded in an embodiment, an X-frame in lieu of a regular B-frame may be provided similar to the foregoing process. As noted previously with respect to FIGS. **5** and **7**, operations set forth at blocks **904** and **906** may be combined to be executed in a single encode process for the sake of computational efficiency in one example embodiment.

In an additional or alternative embodiment of a PE scheme, a phase-encoded bitstream may have a frame other than an I-frame as a first frame of the coded video sequence, which may be achieved by appropriate settings in an encoder in accordance with the teachings herein. For example, the first frame can be an X-frame (or some other non-I frame). All other frames of the coded sequence may contain predicted frames (P/B-frames) and X-frames at suitable locations based on phase.

FIG. **12** depicts a plurality of coded bitstreams **1200** having different phases for a particular bitrate representation generated by a PE-based tiled encoder system in an example embodiment. By way of illustration, a QP-N stream **1202**-N having a QP value of 22 is encoded or otherwise provided as four phase-encoded streams **1204-1** to **1204-4** because of the use of a GOP size of four frames in this example. For each PE stream **1204-1** to **1204-4**, the first frame is encoded as an I-frame **1206-1** to **1206-4**. The rest of the frames in each PE stream are encoded as either P- or X-frames based on the phase-position relationship set forth above.

Turning to FIG. **10**A, depicted therein is a flowchart illustrative of a process **1000**A for configuring a PE scheme in a tiled encoding arrangement according to an example embodiment of the present invention. At block **1002**, an encoder may be initialized for selecting a PE scheme with respect to a media input stream corresponding to a 360° immersive video asset. At block **1008**, period and phase parameters are obtained or otherwise configured, where period is equal to the GOP size (block **1004**) and a phase is equal to or less than the GOP size (block **1006**). At block **1010**, the encoder may be set to use tile encoding to generate tiles in a particular grid/array arrangement per frame. Similar to a BIE configuration process set forth previously, a base_qp parameter may be written in slice QP headers of the encoded streams (block **1012**). At noted before, it is required that all pictures with the same picture number to be used as an input for stitching must use the same base_qp value. Thus, it is not a necessary requirement in an example embodiment to set the same base_qp parameter in all the stream headers. To facilitate the encoded streams in different qualities (while having the same base_qp), a qp_delta parameter may be configured as described above with respect to each stream based on target QPs (block **1014**). As before in an example BIE configuration process, a qp_delta of −10 may be configured where base_qp is 32 to achieve a target QP of 22 for a particular stream. Spatial motion vector prediction may be configured such that it is restricted within the tile only (block **1016**). That is, motion vectors are not allowed to cross the tile boundaries in an example embodiment (i.e., only intra-tile prediction is allowed and no inter prediction or context selection across a tile boundary is allowed). This means that motion vectors are set such that no sample outside the boundaries of any co-located tile is read during motion compensation interpolation of the blocks inside a tile. An ROI grid may be configured for the encoder such that it uses the qp_delta information for encoding a particular stream with respect to a specific region of the frames (block **1018**). Further, TMVP may also be disabled (block **1020**) in an example PE configuration process as noted above.

It should be noted that an example PE configuration process is roughly similar to a BIE configuration process in one embodiment, which may be performed for every "phased" stream depending on the GOP size. Further, analogous to a BIE configuration process **800**A that uses certain parameters, additional or alternative embodiments of a PE configuration process may involve other parameters in addition to and/or in lieu of the parameters exemplified in the flowchart of FIG. **10**A.

FIG. **10**B is a flowchart illustrative of blocks, steps and/or acts in an example PE implementation according to an embodiment of the present invention. In general, an encoder may be configured to effectuate several decisions during PE-based tiled coding to generate an X-frame only at specific frame locations of each phase-encoded stream. At block **1034**, a video input **1032** is received for encoding. At block **1040**, a tiled encoder is configured for implementing a PE process based on a period (block **1036**) and phase (block **1038**) as set forth above. For each stream, the first frame is encoded as an I-frame (block **1042**). Thereafter, an iterative process may be implemented to effectuate appropriate coding decisions on a frame-by-frame basis, which commences with a determination as to whether the video sequence has reached its end (block **1044**). If the end is not reached (i.e., there remain frames in the video sequence requiring processing), a frame index (i) is incremented (block **1046**) and a next frame is obtained and denoted as an i^{th} frame (block **1048**). A determination is made if the modular relationship {i Mod (period)=phase} is satisfied. If so, the frame is encoded as an X-frame as set forth at blocks **1054**, **1056** and **1058**. Otherwise, it is encoded as a regular P-frame (block **1052**). Thereafter the process flow returns to determine if all the frames of the video stream have been processed (block **1044**). If so, the process flow proceeds to finalize encoding the video stream (block **1060**), which may be provided as PE-tiled bitstream to a downstream entity (e.g., a packaging system), as set forth at block **1062**.

As noted previously, a PE-based tiled encoding scheme facilitates a gradual refresh annealing process during 360° video delivery, which will be set forth in further detail below. An embodiment of phased encoding may also be used during the playout where a stitcher executing on the server side or on the client side may be used to combine tiles of different qualities. So, at every frame of the video being played, each tile has a current quality, which may correspond to the QP value, target bitrate or other indicia of the video stream the tile was taken from. When bandwidth is sufficiently large or when the user moves his head and the viewport changes, it would be desirable to upgrade the quality (e.g., lower the QP) of some tiles (the tiles on the new viewport for example). Furthermore, to reduce latency by means of reducing the usage of the buffer on the decoder side, an embodiment of the present invention provides that the entire viewport may not be upgraded at once, but rather upgrade it gradually by means of gradual refresh, only upgrading a few tiles in every frame, keeping the decoder buffer small and thus reducing latency. As will be described in additional detail further below, an example bandwidth annealing apparatus may be configured to effectuate a process for determining which tile to upgrade at every moment based on the bandwidth, the viewport and/or current buffer utilization. Further, such a process may also be configured to determine a quality level (i.e., which QP) to which a tile should be upgraded.

For example, assume that during playout, a tile selection apparatus (described in detail further below) determines to upgrade in the i^{th} frame, tile T to quality QP=q. This determination may be provided as a control input to a tile/frame stitcher module, which retrieves, receives or otherwise obtains tile T from the i^{th} frame of the video stream that was encoded with quality QP=base_qp+delta_qp=q using phased encoding, where the phase is determined by the modular relationship: {phase=i Mod (period)}. Then, until the next time the tile selection process decides to change the quality of this tile, tile T is taken from the same stream (i.e., the phased encoded stream with quality QP=q and with the same phase). Accordingly, it will be appreciated that an additional advantage of the PE scheme beyond the ability to perform a gradual refresh of tiles during upgrades is better video quality. Overall, phased encoding gives a better QoE than a BIE scheme where X-frames are substituted without phases, which can result in drift, and result in lower peak signal-to-noise (PSNR) values, thereby resulting in a lower QoE stream for the remainder of the GOP. As noted previously, a potential drawback of phased encoding is the requirement of multiple streams that can result in significant encode processing overhead and storage space.

Example embodiments regarding how to stitch tile-encoded bitstreams using either PE or BIE schemes will be set forth below. As noted previously, tile-stitching embodiments may be implemented at a server during stream delivery phase or on the client side for playout. In general, example embodiments for stitching tiles involve utilizing bitstreams of different qualities (e.g., based on different QPs, targeted bitrates, or other indicia) as well as ensuring that there is compatibility with respect to various pieces of parametric data relating to video pictures, e.g., Video Parameter Set (VPS), Sequence Parameter Set (SPS), Picture Parameter Set (PPS), Supplemental Enhancement Information (SEI), etc., among the bitstreams from which the tiles may be selected. In general, the tile structure should preferably be constant over time to facilitate stitching, which in turn is related to tile-encoding processes performed by an encoder of the present invention. A bitstream stitcher module is operative in response to an input comprising a list of tiles from different tile-encoded streams, which may be combined to generate a new output bitstream, where tiles closer to the viewport have a higher quality compared with tiles further away from the viewport. Further, an example embodiment to perform the tile combination and stream muxing in accordance with the teachings of the present invention may be configured such that output stream generation still remains compliant within known codec standards such as MPEG HEVC/ITU-T/ISO23008 part 2/H.265 specifications as well as emerging standards such as AV1, H.266, VVC, and the like.

For stitching BIE-coded streams, tiles from the regular streams may be used by default for splicing (e.g., until some control input is provided based on user's gaze or bandwidth allocation). The only instances where tiles from the BIE-coded streams are taken is when either the viewport changed (thereby requiring the X-frames which are frames with P-slice headers that can fit in the middle of the GOP but the tiles are intra encoded so the new viewport can be presented) or when a bandwidth annealing process determines to upgrade the quality of the tile (in which case the Block-Intra frame with the P-slice headers contains the tile with the upgraded higher quality).

FIG. **13**A is illustrative of various blocks, steps and/or acts of an example tile stitching scheme **1300**A involving BIE-coded tiled streams according to some example embodiments of the present invention. At block **1302**, a BIE bitstream stitcher receives input bitstreams of different QPs, a first set comprising regular tile-coded streams and a second set comprising BIE tile-coded streams. As noted above, the streams in example embodiments are motion constrained and have the same base_qp for each frame N as the base QP in any other frame N. A tile selection module provides a list of tiles with different QPs (block **1306**), which forms part of overall input regarding the description and parametric information for each tile and the particular QP bitstream from which the tile is to be retrieved or obtained (block **1304**). A tile stitching process may be effectuated on a tile-by-tile basis, as set forth in block **1308**. If the viewport and/or the tile QP has/have changed (block **1310**), the tile is taken from a BIE-coded stream having the appropriate QP and stitched into the frame (block **1312**). Otherwise, the tile is taken from a regular tile-encoded stream and stitched accordingly (block **1314**). After all the tiles are stitched in a frame (in a predetermined grid array), the stitched frame having different qualities of tiles may be provided as output (block **1316**). If additional video frames remain for processing (e.g., encoding and stitching), the process flow may continue.

By way of illustration, consider a block-intra stream stitching scenario in which there are at least three streams: (1) a regular stream of lower quality (e.g. QP setting of 30); (2) a regular stream of higher quality (e.g. QP setting of 22); and (3) a BIE (all-Intra) stream of higher quality. Broadly, when the viewport changes, the quality of some tiles may be increased. That is done in block **1312**, which means that, e.g., a tile at position A that in previous pictures was taken from stream (1) is now taken from stream (3). In the next picture, the tile at position A should be taken from stream (2) if the tile is still within the viewport. If the tile is no longer within the viewport, the position A tile could be taken from stream (1). More particularly, it may be further dependent upon gaze vector information. In other words, it is not just if the tile at position A is in the viewport or not; rather, it is where the tile is located in a gaze-to-weight determination scheme used for tile selection (described in detail further below). Thus, it should be understood that tiles within the viewport depending on where they are located may be upgraded or downgraded based on how far the tile are from the direct line of sight in an example embodiment of the present invention.

In similar fashion, an example tile stitching scheme **1300**B involving PE-based tiled streams is illustrated in FIG. **13**B. A PE bitstream stitcher is operative to receive input bitstreams of different QPs, each encoded into a plurality of phase-encoded bitstreams (block **1332**). A tile selection module provides a list of tiles with different QPs (block **1336**), which forms part of overall input regarding the description and parametric information for each tile and the particular QP bitstream from which the tile is to be retrieved or obtained (block **1334**). A tile stitching process similar to BIE tile stitching may be effectuated on a tile-by-tile basis, as set forth in block **1338**. If the viewport and/or the tile QP has/have changed such that the quality of a current tile is required to change (block **1340**), the tile is taken from a PE-coded stream having the appropriate QP based on a phase-frame modular relationship and stitched into the frame (block **1342**). For example, if the QP of tile at frame I changes to QP=q, the tile from the stream whose phase equals {i Mod (period)} and QP=q is taken and stitched at appropriate location of a tile grid. Otherwise, the tile is taken from the same bitstream from which it was taken in the previous frame and stitched accordingly (block **1344**). After all the tiles are stitched in a frame (in a predetermined grid array), the stitched frame having different qualities of tiles may be provided as output (block **1346**). Further, if additional video frames remain for processing (e.g., encoding and stitching), the process flow may continue.

Regardless of whether tiles from BIE-coded bitstreams or PE-coded bitstreams are stitched, an example embodiment of stitching may involve taking tiles from different streams having compatible slice headers in addition to other parametric information as set forth previously. In general, slice type (i.e., I/P/B-slice), the slice QP and other fields or parameters that may affect the CABAC decoding process may be monitored to ensure compatibility and compliance. Further, some embodiments, such as example embodiments set forth in FIGS. **13A**/**13B**, may require that inter prediction is done using only the previously decoded picture.

Turning to FIG. **13C**, shown therein is a flowchart illustrative of additional blocks, steps and/or acts with respect to an example tile stitching/splicing scheme according to an example embodiment of the present invention. At block **1362**, tiles of different QPs for the current frame (to be stitched) are obtained as input. The data of the tiles (either from the BIE streams or from the PE streams) selected based on a tile selection process is copied in a memory (block **1364**). At block **1366**, the splicing process commences with a prototype slice header that may include a header field, an offset field, etc. (block **1368**). For a tile index (i), an entry_point_offset[i] may be determined from the tile sizes (block **1368**). Bits needed for the largest value of entry_point_offset[i] is determined (block **1370**). The slice header may be adjusted with a new Entry Point Offset (EPO) length based on the largest offset value of all the tile indices as determined previously (block **1372**). At block **1374**, the EPO field is written into the slice header. Thereafter, the tiles are concatenated together after the slice header (block **1376**), thereby generating an output bitstream of the stitched frame (block **1378**).

Skilled artisans will recognize that in order to splice tiles they need to be retrieved from specific source bitstreams responsive to a tile selection process. To facilitate efficient retrieval, an embodiment of splicing may involve providing a memory-mapped tile pointer cache that allows a quicker referencing of parsed files corresponding to tiles, wherein a file format is optimized to be memory mapped instead of being parsed into RAM. Set forth below is an example file format for purposes of an exemplary splicing embodiment:

```
file format:
u(16) magiclen;
s(magiclen) magic_string;
loop { // each iteration is a group
  u(32) nrec;
  for (i=0; i<nrec; i++) {
    // per frame
    u(64) rec_end_relative[i];
  }
  // rec_end_relative[i] is relative to the offset of the file represented by
this comment.
  // rec_end_abs[i] = this_offset + rec_end_relative[i]
  for (i=0; i<nrec; i++) {
  // this iteration is per frame
  // prefixes are NAL units that appear before the Slice in the access unit
    u(32) n_prefixes;
    for (j=0; j<prefixes, j++) {
      u(64) prefix_start_abs[i][j];
      u(64) prefix_len[i][j];
    }
    // the slice
    u(64) slice_start_abs[i];
    u(64) slice_len[i];
    u(32) n_tiles,
    for (j=0; j<n_tiles, j++) {
      u(64) tile_start_abs[i][j];
      u(64) tile_len[i][j];
    }
// suffixes are NAL units that appear after the Slice in the access unit
    u(32) n_suffixes,
    for (j=0; j<n_suffixes; j++) {
      u(64) suffix_start_abs[i][j];
```


-continued

```
      u(64) suffix_len[i][j];
    }
  }
}
magic_string is "TPTCACHE1 [ii[ ][il[ ]llil[ ]il[ ]]]"
```

Referring to FIG. **14**, shown therein is an example 360° video frame **1400** comprising tiles selected and spliced from coded bitstreams having different qualities or QPs in accordance with an embodiment of the present invention. By way of illustration, video frame **1400** is formed from 128 tiles (16 columns by 8 rows) of a 4K video input, shown in unwrapped format (i.e., not projected in a 3D spherical space), wherein a field **1402** which may correspond to an ROI of the frame **1400** based on the viewport or gaze vector location. In accordance with the teachings herein, ROI **1402** may be formed from splicing high quality tiles (i.e., tiles selected from coded bitstreams having low QPs, e.g., QP-16 at 105.6 Mbps, and concatenated in a stitching process). Regions or fields disposed proximate/adjacent to ROI **1402** may have medium quality tiles (e.g., field **1404**). On the other hand, fields or regions distally disposed from ROI **1402**, e.g., those farther away from the viewport, may be formed from lower quality tiles, as exemplified by regions **1406** and **1408**.

To facilitate gaze-based tile selection control, additional embodiments of the present invention involve monitoring where a user is viewing in a 360° immersive video program (i.e., the user's viewport) and determining appropriate tile weights based on the user's gaze. In general, a gaze vector (GV) may be returned by the user/client device defining a gaze direction in a 3D immersive space displaying 360° video, e.g., where the headset is pointed. In further embodiments, the user's eyeball movement may be tracked for similar purposes. As will be seen below, the tiles of a tiled frame also have direction vectors (which are not dependent on the user's gaze) based on how the frame is mapped in a 3D display environment. A dot product (also referred to as a scalar product or inner product) of tile vector and gaze vector can be calculated to determine the angular separation between the gaze direction and the direction of the middle of any tile of a frame, which may be provided to a weighting function module for determining corresponding tile weights.

FIGS. **15A** and **15B** are flowcharts illustrative of various blocks, steps and/or acts of a gaze control scheme that may be (re)combined in one or more arrangements, with or without blocks, steps and/or acts of additional flowcharts of the present disclosure, for facilitating optimized tile selection according to one or more embodiments of the present invention. Process **1500A** involves receiving a gaze vector from a client device operating to display a 360° immersive video asset to a user, wherein each video frame comprises an array of tiles projected on a 3-dimensional (3D) display environment viewed by the user in which the user is immersed, the gaze vector defining a gaze direction in the 3D display environment where the user is viewing at any particular time (block **1502**). In one embodiment, gaze vector information may comprise (x,y,z) information in a 3D Cartesian coordinate system that may be associated with the display environment. In another embodiment, gaze vector information may comprise (ρ,θ,φ) information in a 3D spherical coordinate system based on an equirectangular projection mapping. In another embodiment, a 3D gaze vector may be normalized (to obtain a unit-length directional vector). Skilled artisans will therefore recognize that GV information may be provided in a number of ways depending on geometrical modeling, projection mapping, computational methodology, etc., used in a particular implementation. At block **1504**, a determination may be made as to what the angular separation is between the gaze vector and a directional vector associated with each tile location respectively corresponding to the array of tiles in the 3D display environment, which again may be dependent on the particular geometrical modeling, projection mapping, computational methodology, and the like. At block **1506**, responsive to the angular separations, a plurality of tile weights are determined corresponding to the array of tiles for use in selecting tiles of different bitrate qualities (QPs) of the 360° immersive video asset for assembling a video frame to be delivered to the client device. In general, tiles (or, more broadly, tile positions or locations) close to or within an arbitrary angular distance from the gaze vector may be assigned higher values, whereas tiles that are directly opposite to the gaze vector (i.e., 180° or π radians) may be assigned lowest weight values, with the remaining tiles in between (in both horizontal and vertical directions) receiving varying weight values between the maximum and minimum values according to any suitable mathematical relationship (e.g., linear, quadratic, etc.).

Process **1500**B sets forth additional details with respect to effectuating gaze-based control in an example embodiment. At block **1522**, tile weights may be determined as a function of a cosine of an angular separation between the gaze vector and the directional vector corresponding to a tile location in a suitable 3D spatial projection of a 2D video frame of the 360° immersive video asset. At block **1524**, the tile weights may be provided as an input along with a dynamic bandwidth allocation input to a tile selection and bandwidth annealing process, which is further described elsewhere in the present patent application.

In one example embodiment, depending on where the tile is located in relation to the gaze vector, a determination is made how much bandwidth is allocated to that tile location corresponding to the weight. Where the gaze vector $\vec{a}$ and tile directional vector $\vec{b}$ are denoted by vectors, their dot product may be determined as follows:

$$\vec{a}\cdot\vec{b}=|a|\cdot|b|\cos\theta$$

Upon normalization, i.e., if [â]=gaze/|gaze|, then |a|=1. Likewise, by assigning [$\hat{b}$]=tile_direction/|tile_direction|, |b|=1. Accordingly, by normalizing, the foregoing relationship simplifies to:

$$\hat{a}\cdot\hat{b}=\cos\theta$$

Rather than mapping cos(θ) back to θ to determine a weight, an embodiment of the present invention involves defining a mathematical function to map from cos(θ) to a weight as follows: x=cos(θ), and f(x)={x+1} if x≥0 and f(x)=[α{x+1}] if x<0, where α=a scaling factor, e.g., 0.1. Thus, if the angular separation between a gaze vector and tile directional vector is 0°, cos(θ)=1 and f(x)=2. Likewise, for a tile that is 60° or 300° away from the gaze vector, cos(θ)=0.5 and the corresponding f(x) value is 1.5. In an equirectangular projection of a 3D frame, the angle exactly opposite to where the user is looking is 180°, which yields cos(θ)=−1.0, thereby obtaining a weight f(x) value of 0 regardless of the scaling factor. Accordingly, an example embodiment may provide a suitable scaling factor based on how smoothly or quickly tile qualities may vary in relation to a gaze direction within a frame.

FIG. **16**A illustrates an example unit-circular geometrical arrangement **1600**A for facilitating determination of angular separation between a user's gaze direction and tile positions. A user location **1602** is set as the center of a unit-circular cross section of a 3D spherical space. By referencing the user's gaze along a first referential axis (e.g., X-axis) **1604**, different angular displacements for the tile locations may be determined as set forth above. By way of illustration, reference numerals **1606** and **1608** refer to two tile directional vectors that are 30° and 60° away from the gaze direction **1604**. In general, tile locations approaching near ±90° or thereabouts (e.g., reference numerals **1610**A/**1610**B) with respect to the gaze direction **1604** connote a user's mid-to-far peripheral vision and a weighted scaling factor may be utilized such that tiles in such regions and beyond may be allocated a faster reduction in bandwidth (i.e., lesser quality). At a directional vector location **1614**, the tiles are ±180° away from the gaze direction **1604**.

In an example embodiment, instead of actual angular displacements, cosine values corresponding to different locations may be provided in reference to the gaze direction. For instance, if a tile direction vector is 90° or 270° from the gaze vector, x=0.0 may be fed to the weighting function, which yields a weight of 1.0. Likewise, for a tile direction vector is 330° away, x=0.866 is provided to the weighting function, thereby yielding a weight value of 1.866. As a further example, if the tile direction vector is 120° deg away, x=−0.5, is provided to the weighting function, thereby yielding a weight value of 0.05 (assuming α=0.1), which is the same if the tile direction were 240° away from the gaze vector).

Further, both gaze vector information and tile direction vector information may be converted to appropriate tile coordinate information relative to the tile grid used in tile encoding during media preparation for facilitating identification of tiles by rows and columns, which may be input along with the weight information to a tile selection and bandwidth annealing process. One skilled in the art will recognize that the determination of tile coordinate information is dependent on the projection mapping used in an example embodiment. FIG. **16**B illustrates an equirectangular projection mapping scheme resulting in a spherical display environment **1600**B where the tiles form the surface. One example implementation provides placing a north pole **1605** in the direction of {0,1,0} and a south pole **1607** in the opposite direction, whereas the left and right edges of a tiled frame are in the direction of {0,0,1} and the center of the image (i.e., the tiled frame) is in the direction of {0,0,−1}. In an example implementation involving uniform tile sizes, an embodiment of the present invention provides an apparatus and method for determining the location of a tile **1609** which has a directional vector **1611**, which may be configured to compute t_x (the column index of the tile) and t_y (the row index of the tile) for a given grid arrangement of n_x (the number of tile columns) and n_y (the number of tile rows) as follows, where θ is the polar angle and φ is the azimuthal angle of the spherical coordinate system:

$$\theta=\{[t_x+½]/n_x\}2\pi$$

$$\varphi=\lfloor ½-\{t_y+½\}/n_y\rfloor\pi$$

y=sin φ
r=cos φ
z=r*cos θ
x=r*sin θ

Where the encoding has non-uniform tile sizes, the foregoing equations may be modified based on, e.g., pixel areas of individual tiles, etc. By way of illustration, using (i) as the tile index for the left edge of tile column i, (j) as the tile index for the top edge of the tile row j, w is the number of pixel columns, and h is the number of pixel rows, an embodiment of the present invention may be configured to determine the following wherein both x_i and y_j involve a "floor" operator to round out (i.e., the fractional part is removed) with respect to using an example coding unit or block size (e.g., 64 pixels):

$$x_i \left\lfloor \frac{iw}{64\, n_x} \right\rfloor 64$$

$$\theta = \frac{x_i + x_{i+1}}{2w} 2\pi$$

$$y_j = \left\lfloor \frac{jh}{64\, n_y} 64 \right\rfloor$$

$$\phi = \frac{y_j + y_{j+1}}{2h} 2\pi$$

FIG. **16**C is illustrative of an example 360° immersive video viewing environment **1600**C for purposes of one or more embodiments of the present invention. A premises node or gateway (GW) **1642** associated with subscriber premises **1640** is served by a delivery pipe **1644** for providing immersive media content. In one arrangement, such immersive media content may be presented in a 3D-panoramic virtual space viewed in a suitable headset worn by the subscriber/user. An example UE may comprise a CPE **1646** served by GW **1642**, such as a gaming console, laptop, or a smartphone, for example, that executes one or more gaming or media applications to provide suitable signals to one or more devices such as a display device **1636** mounted to or on a user's head **1628**. Additional examples of such devices may comprise visors, goggles, wired/wireless headgear or helmets, masks, etc. that can display or effectuate an immersive viewing space surrounding the user. In an example display device arrangement, there maybe additional instrumentation such as a gyroscope, accelerometer and a magnetometer, etc., to facilitate head tracking, i.e., when the user **1628** moves her head, the field of view around the simulated space may move accordingly, along with a portion of the space being gazed at (i.e., the viewport) by the user. Thus, in a head-tracking headset, the cone of view or field of view as well as the user's viewport moves around as the user looks up, down and moves side to side or angles her head. An example system may include the so-called 6DoF (six degrees of freedom) arrangement that can plot the user's head in terms of X-, Y- and Z-axes to measure head movements, also known as pitch, yaw and roll, which may be used for tracking the user's point of view within the simulated 3D panoramic viewing space.

By way of illustration, CPE **1646** may be embodied as a platform **1648** including one or more processors **1656**, volatile and nonvolatile/persistent memory **1654**, input/output (I/O) interfaces **1660** (e.g., touch screens, gaming controllers, hand-tracking gloves, etc.), as well as one or more 360-degree media/gaming applications **1638** that can effectuate a 3D virtual viewing space or "screen" **1620** for the user **1628** wearing head-mounted display (HMD) **1636**. In one example arrangement, HMD **1636** may be wirelessly coupled to CPE **1646** via wireless interface **1642**. A plurality of decoder buffers **1645** may be provided as part of an example CPE platform **1646/1648** corresponding to one or more 360° immersive video content channels available to the user **1628**.

Additional 3D-media-capable CPE **1634** (e.g., a tablet, phablet or smartphone, etc.) may also be separately or optionally provided. Example CPE apparatus **1646/1634** operating together or separately in conjunction with HMD **1636** may be operative to effectuate 3D virtual viewing space **1620** that is an immersive environment in which the user **1628** can move her point of view in full 360° in one of a vertical plane, a horizontal plane, or both planes, defined in the 3D environment, wherein the viewport **1624** changes accordingly. In an additional or alternative arrangement, CPE apparatus **1646/1634** operating in conjunction with HMD **1636** may be operative to effectuate a 3D virtual viewing space **1620** that may be partially immersive in that it is less than 360° along any one of the axes.

A movement and gaze detection module **1662** is operative to detect a movement in a point of view or gaze direction of the user/subscriber **1628** with respect to the 3D virtual viewing space **1620** and provide a suitable gaze vector output to a serving node as the subscriber **1628** shifts her gaze within the viewing space **1620**. In one embodiment, a tile weighting module may be configured to operate at a 360° video optimization node (e.g., node **216** in FIG. **2**) to determine appropriate tile weights based on the gaze vector information. In another embodiment, tile weighting may be performed locally at example apparatus **1646/1634** and/or at HMD **1636**.

FIG. **17**A is a flowchart illustrative of additional blocks, steps and/or acts with respect to an example 360° immersive video optimization process according to an example embodiment of the present invention. In particular, process **1700**A exemplifies client-side processing with respect to gaze/movement detection in one implementation. At block **1702**, a user commences a 360° video session, whereupon a client device sends a request to a back office node (e.g., node **238** in FIG. **2**) with respect to a requested 360° video asset (block **1704**). At block **1706**, the back office node responds with a URL for the requested asset and provides a video session ID to the client. Responsive thereto, the client device commences receiving the encoded video asset via streaming from the location identified in the URL, which a device player of the client device decodes and renders in a 3D immersive environment (block **1710**). Also, the client device may commence monitoring or tracking head/ocular movement of the user operating the client device in connection with the ongoing 360° video session (block **1708**). Responsive to detecting that a movement is detected (block **1712**), gaze vector information with respect to a current viewport is provided to a 360° video optimization node (e.g., node **216** in FIG. **2**), which utilizes the gaze vector information in combination with other pieces of information in a bandwidth annealing and tile selection process (block **1714**). In one embodiment, gaze vector information may be generated until the user has stopped playing the video and/or no head/ocular movement is detected (e.g., over a period of time), as illustrated in an iterative loop involving decision blocks **1712** and **1716**. In one embodiment, gaze vectors may be generated at a predetermined frequency (e.g., 40 times per second). As will be seen below, not all gaze vectors may be utilized in an example bandwidth annealing and tile selection process, which may be configured to be triggered only when there is a need for tile quality modification, e.g., upgrading or downgrading. When the user stops playing the video asset, appropriate session termination request/message may be generated to the delivery sever (block **1718**), whereupon the process flow may terminate (block **1720**).

Set forth below is a list of gaze vectors provided by a client device in an example implementation over a configurable time window:

0.3203731,	0.1810199,	0.9298348
0.3201844,	0.1811305,	0.9298784
0.3201652,	0.1811581,	0.9298795
0.3201838,	0.1811286,	0.9298789
0.3201413,	0.1811444,	0.9298905
−0.02325181,	−0.6079658,	0.7936227
−0.01977778,	−0.6028962,	0.7975745
−0.01794935,	−0.6024268,	0.7979723
−0.01342396,	−0.6015137,	0.7987497
−0.01229509,	−0.6009697,	0.7991772
−0.0120346,	−0.5997405,	0.8001041
−0.01373066,	−0.6005607,	0.7994613
−0.01506477,	−0.5993657,	0.8003336
−0.01094525,	−0.5975212,	0.8017784
−0.009084027,	−0.5964078,	0.8026301
−0.008858532,	−0.5953203,	0.8034396
−0.00746176,	−0.5926894,	0.8053966
−0.007450074,	−0.5930958,	0.8050975
−0.01072073,	−0.5926897,	0.8053595
−0.01269324,	−0.5921446,	0.8057318
−0.01323339,	−0.5883871,	0.8084711
−0.01338883,	−0.586729,	0.8096727
−0.01282388,	−0.5847392,	0.81112
−0.01634659,	−0.5839438,	0.8116295
−0.02636183,	−0.5821166,	0.8126778
−0.02774585,	−0.5801842,	0.8140126
−0.0245801,	−0.5784537,	0.8153448
−0.02183155,	−0.5797198,	0.8145235
−0.02022467,	−0.5769228,	0.8165482
−0.9961338,	0.007874234,	0.0874956
−0.9719607,	−0.02848928,	0.2334113
−0.9855442,	−0.0176625,	0.1684957
−0.9825167,	−0.0296559,	0.1837972
−0.9824995,	−0.03729712,	0.182493
−0.982159,	−0.03973407,	0.1838061
−0.9689301,	−0.02837855,	0.2457015
−0.8717358,	−0.01528142,	0.4897378
−0.4374043,	−0.01084228,	0.8991996
0.2052692,	0.0161775,	0.9785718
0.6165089,	−0.005071477,	0.7873316
0.7826833,	−0.01918624,	0.6221244
0.778906,	−0.0795427,	0.6220759
0.7230754,	−0.0673095,	0.6874819
0.6768191,	−0.06240646,	0.7334994
0.5633906,	−0.0747445,	0.8228027

In a non-normalized format, example GVs in a Cartesian coordinate system may comprise (x,y,z) values such as [3,5,1]; [10,4,1], etc. In a normalized spherical coordinate system, the GV values may comprise sets of angles such as, e.g., (59.04°, 80.27°), where r=radius has been normalized out, 8=polar inclination and φ=azimuth angle. Regardless of the format, whereas the gaze vector information may be provided or otherwise obtained at configurable frequencies, time periods, etc., not all gaze vectors may need to be utilized in a tile weight determination process. For example, tile weights may be determined and utilized only in response to triggering a tile selection and bandwidth annealing process, as noted previously with respect to certain embodiments. Accordingly, unused gaze vector information may be periodically discarded in such embodiments.

FIG. **17**B is a flowchart illustrative of additional blocks, steps and/or acts with respect to further aspects of an example 360° immersive video optimization process according to an example embodiment of the present invention. In particular, process **1700**B illustrates server-side processing with respect to tile weight determination based on gaze/movement detection and utilization of tile weights in bandwidth annealing and tile selection, inter alia, in an example implementation. At block **1742**, a video back office node receives a user request for commencing a session, whereupon a session setup request may be generated to a 360° video optimization system (block **1744**). Responsive to obtaining appropriate pieces of information, e.g., session ID, session's manifest URL, etc., the back office provides the requisite information to a client device for starting the requested video asset (block **1746**). A bandwidth annealing and QoE management module with tile selection functionality (also referred to as BWA-TS module in some embodiments) is operative to obtain, retrieve, read and/or process the manifest associated with the requested video asset in all encoding representations (block **1748**). At block **1750**, BWA-TS module may also be configured to receive dynamic bandwidth notifications from the delivery network infrastructure (e.g., DSLAM/CMTS in an example embodiment) with respect to the client device's video session. At block **1752**, BWA-TS module is operative to extract specific tiles from the tiled encoding streams or representations. At block **1754**, BWA-TS module is operative to receive control inputs (blocks **1756**, **1758**) regarding bandwidth allocation for the 360° immersive video session as well as any gaze vector information. As noted previously, if the gaze vector input is not available initially, a default value may be used that may be configurable based on content type, content provider policy, client device type and capabilities, etc. Responsive to the control inputs, BWA-TS functionality is operative to generate or otherwise indicate a selected set of tiles based on the bandwidth and tile weights (block **1754**). A tile combining/stitching and stream generation functionality (also referred to as TC-SG module in some embodiments) is operative to receive the selected tile set (block **1760**), which may be concatenated as set forth hereinabove. Accordingly, in one implementation, a video slice header is concatenated with select tiles and appropriately modified to include applicable entry point offsets (block **1762**). For purposes of tile stitching, certain operations may be performed at a Network Abstraction Layer (NAL) access unit level, where the coded video data is organized into multiple NAL units in a tiling hierarchy. A NAL access unit, which is effectively a packet that contains an integer number of bytes, may be treated as a logical substructure of an elementary stream formed by binary audio/video flows and compressed to facilitate bitstream manipulation access. In one implementation, it is the smallest data organization that is possible to be attributed in a system of synchronization involving layer compression, where MPEG decoding operations may be made, taking into account that consistency among the video parametric information (e.g., spatial/temporal redundancy, etc.) is maintained.

Continuing to refer to FIG. **17**B, at block **1764**, TC-SG module is provided with a segment of data for one frame/picture comprising combined tiles, which may be containerized in a suitable container format, e.g., MPEG-2 Transport Stream container format (M2TS; also referred to as MP2TS sometimes), MPEG 4 part 14 (MP4) container format, or ISO Base Media File Format (ISOBMFF) container format, and the like (block **1766**). A delivery server may be configured to deliver the muxed picture/frame to the client device over a suitable network (block **1768**). As set forth in the embodiment of FIG. **17**B, operations comprising BWA-TS, TC-SG and delivery service of process **1700**B may continue to take place until the delivery communications socket is closed or timed out (block **1770**). Thereafter, the 360° video session with the client device may be terminated (block **1772**).

In an example embodiment, the bandwidth allocation for an exemplary 360° immersive video session may be 19

Mb/s. The video may be encoded with full 360 video using a 128-tile grid, covering bitrates varying from a high of 105.6 Mb/s with a QP value of 16 to a low of 7 Mb/s with a QP value of 30. The higher quality tiles are targeted at the user's direct field of vision. The quality of tiles degrades (i.e., QP values rise) in proportion to the distance from the user's direct field of vision. The functionality of BWA-TS insures that the overall bandwidth of the 360 video session is not exceeded. The tile selection is based on the bitrate of each tile. In an example when the user is looking up at a cloudy sky in a scene, most of the tiles provided in that viewport are relatively high quality. The content of the tiles when looking up in such a scenario is relatively static (i.e., very little motion) and therefore not as many bits are dedicated to the low motion areas by the encoder. This results in the ability to show tiles from the highest quality video encoding with a QP value of 16. When the bandwidth allocation for the 360 video is reduced (for example from 19 Mb/s to 7 Mb/s), the quality of the tiles is also reduced. In the foregoing example, the highest quality tiles in the direct field of vision may have a bitrate of 22.4 Mb/s with a QP value of 22.

FIG. **18**A illustrates a tile-weighted frame **1800**A comprising 16-by-8 array of tiles, wherein each tile is assigned a weight based a gaze vector of {0.783, 0.396, −0.481} provided by a client device in an example implementation. Reference numeral **1802** refers to a viewport associated with the gaze, where the tiles are given highest values in accordance with the teachings of the present invention. One skilled in the art will recognize that as the viewport changes, the region of tiles with highest values also changes concomitantly. In a 360° immersive video display space based on equirectangular projection, the region of tiles with highest values thus also moves around, e.g., to the polar regions if the user is gazing directly up or down, or to the equator if the user is gazing directly in the middle of a picture. By way of illustration, FIG. **18**C depicts a 3D immersive display or viewing space **1800**C where the tiles of highest quality are near the North Pole region **1852** when the user is directly looking up, with the progressively lower quality tiles forming the remaining portion of the immersive space, wherein the lowest quality tiles are located near the South Pole region **1854**. Likewise, FIG. **18**D depicts a 3D immersive display or viewing space **1800**D where the tiles of higher quality are near the South Pole region **1854** when the user is directly looking down, with the progressively lower quality tiles spanning toward the North Pole **1852**.

FIG. **18**B illustrates a device frame buffer **1800**B in an example embodiment. Three consecutive frames **1822**A-**1822**C in the buffer are illustrated, each having a P-slice header but comprising different sets of tiles in a viewport **1820** based on the headset view. Whereas a current frame **1822**A has all I-tiles in its viewport **1820**, the following frames are shown with viewports **1820** having P-tiles.

As noted hereinabove, an aspect of the functionality of a BWA-TS module is to insure that the overall bandwidth of an example 360° immersive video session does not exceed a designated bandwidth allocation (e.g., based on network operator policies, content provider policies, subscriber/device policies, or any combination thereof), while still maximizing quality and viewing experience. Optimized tile selection having suitable bitrate qualities may therefore be configured responsive to a user's field of vision, bandwidth allocation/limitation, bitrates per tile as well as a transmit buffer model such that tiles in the direct line of sight have the best quality possible, with decreasing qualities moving farther away from the direct gaze.

FIG. **19** is a flowchart illustrative of various blocks, steps and/or acts of a BWA-TS process **1900** that may be (re) combined in one or more arrangements, with or without blocks, steps and/or acts of additional flowcharts of the present disclosure, according to one or more embodiments of the present invention. As set forth at block **1902**, process **1900** may commence with or responsive to receiving, retrieving or otherwise obtaining one or more stream manifest files provided by a 360° video asset packager (e.g., packager **214** in FIG. **2**) with respect to a plurality of tile-encoded streams that may be generated according to a BIE or PE scheme. In general, the manifest files may include information or data describing various characteristics of tile groupings per frame, including location URLs, bitrates, slice/block type, media type, etc., for each tile-encoded bitstream corresponding to a particular one of a plurality of bitrate representations of a media input stream. In one arrangement, manifests may be organized in a hierarchical manner, i.e., with certain manifests for describing overall coded bitstreams, while other manifests are provided for describing individual tiles in a stream. As set forth passim in the present patent application, each stream is a particular bitrate representation of the source media having a video quality, e.g., related to or controlled by a corresponding QP used for the bitrate representation, and/or targeted bitrate, or other indicia, wherein each frame of a tile-encoded bitstream comprises an array of tiles organized into at least one slice per frame, wherein a plurality of frames form a GOP structure of the tile-encoded bitstream. At block **1904**, process **1900** proceeds to receiving, retrieve or otherwise obtain gaze vector information, and responsive thereto, determines tile weights corresponding to an array of tiles forming a frame, e.g., based on the gaze vector or by default settings. At block **1906**, process **1900** proceeds to receive, retrieve or otherwise obtain variant weights corresponding to the plurality of bitrate representations or associated tile-encoded bitstreams of the media input stream. In one arrangement, the variant weights may be defined as a policy-based property of the streams where higher quality stream representations (i.e., variants) are accorded a higher priority or weight that may be used in further computations involving weight based knapsack packing selections. At block **1908**, a determination is made with respect to an adequacy metric value as a function of a variant weight and a tile weight for each tile/GOP-tuple combination over a set of frames across a GOP structure for each of the tile-encoded bitstreams. At block **1910**, process **1900** proceeds to selecting tiles having different bitrate qualities from corresponding tile-encoded bitstreams for assembling a frame, responsive at least in part to the adequacy metric values, wherein the bitrate qualities of the selected tiles are optimized to satisfy a transmit buffer model for transmitting a multiplexed video output stream. Thereafter, a list of the selected tiles may be provided to a tile stitcher for generating a frame containing the selected tiles as part of the muxed video output stream (block **1912**). Where the tile stitching is performed in a device-side embodiment, the selected tiles may be provided to the client device in an example embodiment, as noted elsewhere in the present patent application.

An example stream-level manifest for purposes of an embodiment of the present invention is illustrated below:

```
<?xml version="1.0"?>
<TiledMediaDefinition>
  <TileGroup>
```

-continued


```
    <Representation tiles=“128” columns=“16” rows=“8” height=“2160”
id=“1” mimeType=“video/H265” width=“3840” QP=“16”>
      <URL>race360-105Mbs.hevc</URL>
    </Representation>
    <Representation tiles=“128” columns=“16” rows=“8” height=“2160”
id=“2” mimeType=“video/H265” width=“3840” QP=“18”>
      <URL>race360-64_6Mbs.hevc</URL>
    </Representation>
    <Representation tiles=“128” columns=“16” rows=“8” height=“2160”
id=“3” mimeType=“video/H265” width=“3840” QP=“20”>
      <URL>race360-39Mbs.hevc</URL>
    </Representation>
    <Representation tiles=“128” columns=“16” rows=“8” height=“2160”
id=“4” mimeType=“video/H265” width=“3840” QP=“22”>
      <URL>race360-24_4Mbs.hevc</URL>
    </Representation>
    <Representation tiles=“128” columns=“16” rows=“8” height=“2160”
id=“5” mimeType=“video/H265” width=“3840” QP=“24”>
      <URL>race360-16_2Mbs.hevc</URL>
    </Representation>
    <Representation tiles=“128” columns=“16” rows=“8” height=“2160”
id=“6” mimeType=“video/H265” width=“3840” QP=“26”>
      <URL>race360-11_4Mbs.hevc</URL>
    </Representation>
    <Representation tiles=“128” columns=“16” rows=“8” height=“2160”
id=“7” mimeType=“video/H265” width=“3840” QP=“28”>
id=“7” mimeType=“video/H265” width=“3840” QP=“28”>
      <URL>race360-8_6Mbs.hevc</URL>
    </Representation>
    <Representation tiles=“128” columns=“16” rows=“8” height=“2160”
id=“8” mimeType=“video/H265” width=“3840” QP=“30”>
      <URL>race360-7Mbs.hevc</URL>
    </Representation>
  </TileGroup>
</TiledMediaDefinition>
```


An example lower-level manifest based on DASH-MPD for purposes of an embodiment of the present invention involving multiple phase encoded streams is illustrated below:


```
<?xml version=“1.0” encoding=“UTF-8” ?>
- <MPD xmlns:xsi=“http://www.w3.org/2001/XMLSchema-instance”
  xmlns=“urn:mpeg:dash:schema:mpd:2011”
  xsi:schemaLocation=“urn:mpeg:dash:schema:mpd:2011 DASH-
  MPD.xsd” type=“static” mediaPresentationDuration=“PT654S”
  minBufferTime=“PT2S” profiles=“urn:mpeg:dash:profile:isoff-on-
  demand:2011”>
  - <Period tag=“batman-plain”>
    - <AdaptationSet mimeType=“audio/mpegts”>
      - <Representation>
        <BaseURL>batman-audio.ts</BaseURL>
      </Representation>
      </AdaptationSet>
    - <AdaptationSet mimeType=“video/mp4” framePeriod=“1/24”>
      - <Representation label=“batman qp 16 @ 38.0”
        weight=“37.99” graphNote=“16”>
        <BaseURL phase=“0” label=“batman phase 0 qp
          16”>batman-phase-0-qp16.mp4</BaseURL>
        <BaseURL phase=“1” label=“batman phase 1 qp
          16”>batman-phase-1-qp16.mp4</BaseURL>
        <BaseURL phase=“2” label=“batman phase 2 qp
          16”>batman-phase-2-qp16.mp4</BaseURL>
        <BaseURL phase=“3” label=“batman phase 3 qp
          16”>batman-phase-3-qp16.mp4</BaseURL>
        <BaseURL phase=“4” label=“batman phase 4 qp
          16”>batman-phase-4-qp16.mp4</BaseURL>
        <BaseURL phase=“5” label=“batman phase 5 qp
          16”>batman-phase-5-qp16.mp4</BaseURL>
        <BaseURL phase=“6” label=“batman phase 6 qp
          16”>batman-phase-6-qp16.mp4</BaseURL>
        <BaseURL phase=“7” label=“batman phase 7 qp
          16”>batman-phase-7-qp16.mp4</BaseURL>
        <BaseURL phase=“8” label=“batman phase 8 qp
          16”>batman-phase-8-qp16.mp4</BaseURL>
        <BaseURL phase=“9” label=“batman phase 9 qp
```


-continued


```
          16”>batman-phase-9-qp16.mp4</BaseURL>
        <BaseURL phase=“10” label=“batman phase 10 qp
          16”>batman-phase-10-qp16.mp4</BaseURL>
        <BaseURL phase=“11” label=“batman phase 11 qp
          16”>batman-phase-11-qp16.mp4</BaseURL>
        <BaseURL phase=“12” label=“batman phase 12 qp
          16”>batman-phase-12-qp16.mp4</BaseURL>
        <BaseURL phase=“13” label=“batman phase 13 qp
          16”>batman-phase-13-qp16.mp4</BaseURL>
        <BaseURL phase=“14” label=“batman phase 14 qp
          16”>batman-phase-14-qp16.mp4</BaseURL>
        </Representation>
      —
      <Representation label=“batman qp 18 @ 28.9”
        weight=“28.88” graphNote=“18”>
        <BaseURL phase=“0” label=“batman phase 0 qp
          18”>batman-phase-0-qp18.mp4</BaseURL>
        <BaseURL phase=“1” label=“batman phase 1 qp
          18”>batman-phase-1-qp18.mp4</BaseURL>
        <BaseURL phase=“2” label=“batman phase 2 qp
          18”>batman-phase-2-qp18.mp4</BaseURL>
        <BaseURL phase=“3” label=“batman phase 3 qp
          18”>batman-phase-3-qp18.mp4</BaseURL>
        <BaseURL phase=“4” label=“batman phase 4 qp
          18”>batman-phase-4-qp18.mp4</BaseURL>
        <BaseURL phase=“5” label=“batman phase 5 qp
          18”>batman-phase-5-qp18.mp4</BaseURL>
        <BaseURL phase=“6” label=“batman phase 6 qp
          18”>batman-phase-6-qp18.mp4</BaseURL>
        <BaseURL phase=“7” label=“batman phase 7 qp
          18”>batman-phase-7-qp18.mp4</BaseURL>
        <BaseURL phase=“8” label=“batman phase 8 qp
          18”>batman-phase-8-qp18.mp4</BaseURL>
        <BaseURL phase=“9” label=“batman phase 9 qp
          18”>batman-phase-9-qp18.mp4</BaseURL>
        <BaseURL phase=“10” label=“batman phase 10 qp
          18”>batman-phase-10-qp18.mp4</BaseURL>
        <BaseURL phase=“11” label=“batman phase 11 qp
          18”>batman-phase-11-qp18.mp4</BaseURL>
        <BaseURL phase=“12” label=“batman phase 12 qp
          18”>batman-phase-12-qp18.mp4</BaseURL>
        <BaseURL phase=“13” label=“batman phase 13 qp
          18”>batman-phase-13-qp18.mp4</BaseURL>
        <BaseURL phase=“14” label=“batman phase 14 qp
          18”>batman-phase-14-qp18.mp4</BaseURL>
        </Representation>
      ...
      </Representation>
      </AdaptationSet>
    </Period>
  </MPD>
```


FIG. **20** is a flowchart illustrative of additional blocks, steps and/or acts with respect to an example tile selection and bandwidth annealing process according to an embodiment of the present invention. In one arrangement, a knapsack combinatorial optimization may be used for tile selection and annealing based on the inputs comprising gaze vectors, bandwidth allocation/limitation, stream weights, etc., as pointed out previously. At block **2002**, process **2000** executing at a server or node associated with video optimization commences with or responsive to receiving a request for a 360° immersive video session. At block **2004**, process **2000** proceeds to retrieve or otherwise obtain tiled stream manifest definitions so as to be able to determine all aspects of the video characteristics based on deep-level inspection and processing in order to extract the needed tiles, which may be effectuated by way of parsing the steam manifests (block **2006**). A grid layout is determined for each stream, e.g., columns and rows per frame (block **2008**). In an example variation, process **2000** may register with a network management and orchestration node to receive notification messages relative to allocated/determined bandwidth for the requested session (block **2010**). If a bandwidth allocation is received (block **2012**), a further determination may be made whether gaze vector information is received (block **2014**). Thereafter, tile weights are determined based on the gaze vector information (block **2016**). Tile selection may be performed as a knapsack annealing process responsive to available bandwidth allocation notification (block **2018**). At block **2020**, selected tiles are provided to a tile stitching process (executing at the server or at the client device).

FIGS. **21**A and **21**B are flowcharts illustrative of additional blocks, steps and/or acts with respect to further aspects of a tile selection and bandwidth annealing process according to an example embodiment of the present invention. In particular, process **2100**A shown in FIG. **21**A exemplifies a relatively simpler knapsack annealing process, which may be computationally more expensive that can result in about approximately 1 second for tile splicing. At block **2102**, the tiles are initialized to a lowest quality. An adequacy metric may be determined as a ratio between a stream variant weight and a tile weight, which may be provided with respect to all <tile,GOP>-tuples or combinations (block **2104**). A determination is made with respect to upgrading the <tile, GOP>-tuple having the least adequacy (i.e., most inadequacy), as set forth at block **2108**. A determination is made whether a transmit buffer model is violated or satisfied (block **2110**). If the buffer model is not satisfied (i.e., violated), that tile/GOP combination may be disqualified for upgrades and the process flow returns to considering the next tile/GOP combination for upgrading, as set forth at block **2112**. If the buffer model is not violated, the tile/GOP combination is upgraded in quality (block **2114**). The foregoing process may be iteratively performed until there are no non-disqualified tile/GOP combinations at less than maximum quality (block **2116**). If none, process **2100**A is completed by sending selected tiles to a tile mux and stream generation process, as set forth at block **2118**.

Turning to FIG. **21**B, a performance-optimized tile selection and annealing process **2100**B is shown, which in some implementations may result in faster tile section, resulting in overall tile splicing times around 10 milliseconds or so. Broadly, a penalty factor may be imposed with respect to I-tile upgrades (which are costlier than P-tile upgrades as I-tiles pack more data) and a "naïve" upgrade sequence may precede initially where tile upgrades are not checked against the transmit buffer model regardless of whether the upgrades comply with an adequacy metric. Further, as tiles in the ROI/viewport get upgraded first and the rest of the tiles of a frame are upgraded/updated subsequently, an example embodiment may factor in a penalty based on where the tile position is. For example, if the tile position is close to the gaze vector, the penalty associated with that position may be lower. Further, penalty may also be related to the tile position as a balance between the quality/type of the tile to be upgraded vs. where it is in the frame. The effects of a penalty factor or combination may be incorporated in the annealing process by suitably modulating the adequacy metric used in the naïve upgrade sequence in an example embodiment.

Similar to the embodiment of FIG. **21**A, the tiles of all video encodings are initialized to a lowest quality (block **2132**). An adequacy metric may be determined as a ratio between a stream variant weight and a tile weight, multiplied by a penalty factor, which may be provided with respect to all <tile,GOP>-tuples or combinations (block **2136**). At block **2134**, a heap structure (e.g., as a large pool of memory) may be configured for containing adequacy values for all <tile,GOP>-tuples. At block **2138**, a least adequate tile is pulled from the heap and recorded on a naïve upgrade sequence or process. If the tile quality can be upgraded more (block **2140**), it is performed and an adequacy metric for the upgraded tile is determined (block **2142**). The foregoing process may be executed in an iterative loop until the heap is empty and all the tiles that can be upgraded have been upgraded (block **2144**). A binary search sequence may be effectuated to on the naïve sequence to find a last valid state that obeys a given transmit buffer model (block **2146**), which may be used as a starting tile state (block **2148**). A new upgrade heap may be configured for containing the tile/GOP states (block **2150**). A least adequate tile/GOP combination is pulled from the heap (block **2152**) and validated against the transmit buffer model (block **2154**). If the pulled tile/GOP cannot satisfy the buffer model, it is disqualified from future upgrades (block **2158**). Otherwise, a determination is made if it can be upgraded more (block **2156**). If so, an adequacy value for the upgraded tile/GOP combination that satisfies the transmit buffer model is determined (block **2160**). The foregoing operations are performed iteratively until the new upgrade heap is empty, as set forth at block **2162**. If so, process **2100**B is completed by sending selected tiles to a tile mux and stream generation process, as set forth at block **2164**.

Example annealing processes set forth herein advantageously facilitate gradual refreshing of frames when a viewport or bandwidth is changed, thereby allowing for the ability to minimize latency in increasing quality based on a user's field of vision and at the same time not overload the bandwidth. Typically, when attempting to perform quality changes on all tiles at the same time, several issues may be encountered due to the result of changing P-tiles for I-tiles at the same time, which are expensive in terms of encoded bitrate. On the other hand, performing this substitution with a minimal client buffer can cause too much delay in delivering the I-slices/frames.

In an example embodiment that employs gradual refreshing, the video streams do not have I-frames (except for the initial I-frame or any other special frames like Instant Decode Refresh or IDR frames). Instead, a video stream has I-blocks or I-tiles that may be distributed throughout a time sequence so that any particular spot on the screen gets an I-block at regular intervals, e.g., by way of phase-encoded streams as described in detail in the earlier sections of the present patent application. Thus, in such a scenario, there is no frame where all the pixels are refreshed by I-blocks. By performing gradual refresh annealing, example embodiments of the present invention can be advantageously configured to level out frame sizes (i.e., in terms of the amount of coded image data) and reduce the bandwidth consequences of injecting an I-frame to upgrade the quality of tiles that enter the FoV or viewport. Whereas a PE scheme may allow selective early refreshes of a tile in a time/frame sequence, it may impose certain bandwidth cost (e.g., due to having multiple I-tiles in a frame, which can cause an increase in the required bandwidth for that time interval corresponding to the transport of that video frame). However, an example embodiment involving PE can be configured such that the advantage of having a steadier level of bytes/frame overweighs such costs.

Over time in a frame sequence, a PE-based embodiment may allow manipulation of the phases of the various tiles around until the I-tiles are roughly evenly distributed in time again. Such a capability can be configured to be user- and/or content-dependent with respect to when this redistribution occurs as it requires the user to keep their field of view steady long enough for it to occur. In order to choose tiles to fill the bandwidth, an example embodiment may involve modeling the byte sizes of frames stretching 3 GOPs into the future (this choice is arbitrary) and performing hypothetical early refreshes (HER) based on the buffer model (e.g., with 3 GOPs in a look-ahead scenario). Based on the embodiments set forth in FIGS. **21**A and **21**B, it can be seen that such a process starts by picking the minimum-quality stream for all the tiles and then considers each GOP of tiles, both for the current frame and future frames, and evaluates whether upgrading that GOP will violate any bandwidth constraints (which are a combination of individual frame sizes and buffer considerations). If considering upgrading a current (as opposed to future) tile-GOP combination above the quality of an already-delivered I-frame, an embodiment of the present invention may temporally realign this tile to start with an I-frame (which may affect the rest of the frames in a splicing window). Once the list of possible upgrades is obtained, they may be weighted according to the quality and the tile's position in the FoV (so tiles near the center of vision will be favored for upgrades). In one implementation, the foregoing upgrade step may be repeated until buffer constraints make any more upgrades impossible.

It should be appreciated that an example upgrade process may move around in time and in space depending on the look-ahead GOP modeling. In one arrangement, each tile may have a 3-4 GOP horizon, which can each be upgraded as the process is iterated, where future GOP upgrades are for potential future enhancements for early refreshes covering 3-4 GOPs in to the future.

In considering a HER-based implementation, a few potential metrics may be identified and/or employed to obtain a suitable trade-off: (i) dead air, (ii) maximum buffer level, and (iii) end buffer level, among others. In one example implementation, the maximum buffer level may be weighted as a leading criterion for HER upgrades where adequate bandwidth may be freed up to allow tile-GOP quality upgrades.

As set forth in the embodiment of FIG. **21**B, once the end is reached in the upgrading iterations, a slice/frame can be muxed using a set of tiles, whereby the byte size of the muxed slice/slice may be calculated and its effect on the transmit buffer may be recorded so that the next slice/frame is accurately constrained in accordance with the given transmit buffer model. When the next time a frame is spliced (e.g., the user gaze has changed, thereby causing adjustments to be made), the knapsack annealing process may be repeated wherein one extra frame is modeled relative to the previous operation, which can validate and/or fine-tune the knapsack/annealing process.

Skilled artisans will recognize that a heap memory structure employed in the embodiment of FIG. **21**B is particularly advantageous for keeping track of upgradable tiles because recalculating the score of the tile-GOP upgrades on every iteration may be avoided. As noted previously, an adequacy metric is defined for scoring tiles, which is used in choosing which tile to upgrade, wherein parameters such as variant_weight, tile_weight and penalty are provided in a suitable mathematical relationship to capture a desirable upgrade scenario. As such, the variant_weight parameter may be defined as a property of an encoded stream and higher quality stream variants (having lower QPs) have a higher variant_weight. Some example variant weights are {1/QP}, {100-QP}, or a value defined in the manifest examples set forth above, or it could be the bitrate of the entire stream. The tile_weight may also be provided as a function of the tile's position relative to the gaze as set forth above. In general, tiles in the user's direct FoV or ROI/viewport may be accorded higher tile_weights. The example adequacy metric formulation set forth in the embodiments of FIGS. **21**A/B is configured such that as the stream quality increases, the adequacy value also increases, and the tiles closer to the gaze vector have lower adequacy than tiles of the same quality farther from the gaze vector (which configures the annealing process to upgrade tiles closer to the gaze vector before upgrading tiles away from the gaze vector).

Further, example embodiments also include a penalty factor in scoring the tiles for an upgrade process as noted above. In one arrangement, a penalty may be imposed when an early refresh with an I-tile is required wherein a tile in the current GOP is to be upgraded beyond the quality it had in the previous slice/frame. Such a penalty has the effect of increasing that tile's adequacy which delays the upgrade relative to other tiles in the heap. This allows tile upgrades when the gaze has changed enough but defers early refreshes in marginal cases.

It will be apparent to one skilled in the art that additional/alternative formulations may also be used for scoring tile upgrades in some variations within the scope of the present invention.

FIG. **22** is illustrative of a transmit buffer model process for use in a tile selection and bandwidth annealing arrangement according to an example embodiment of the present invention. In general, a transmit buffer model may be configured to be consistent with a frame rate depending on the implementation (e.g., 30 fps, 60 fps, etc.), wherein a temporal variation of how data is added into and transmitted out of a buffer may be parameterized in order to determine whether and when there might be an overflow (i.e., a violation). In the example transmit buffer model **2200**, b_0 is starting buffer level, b_i is the size of buffer before adding an access unit or NAL unit, n_i is the size of the access/NAL unit, and a_i is the size of buffer after adding an access/NAL unit, where $a_i=b_i+n_i$, for $i\geq 1$. Assuming a transmit rate of r, and $\Delta t=1/\text{frame rate}$, the following relationship obtains:

$$b_{i+1}=\mathrm{Max}\{0,a_i-r(t_{i+1}-t_1)\}$$

A buffer_size parameter may be defined as follows:

$$\text{buffer\_size}=r(\text{latency\_frames})\Delta t$$

According to the foregoing model, if $\mathrm{Max}(a_i)>$ buffer_size, it may be indicated as a buffer overflow condition. Thus, as different n_i are being added pursuant to a tile upgrade process, the buffer end point level can be checked against a calculated buffer_size in order to insure that no buffer violations are engendered in the upgrade process.

Turning to FIG. **23**, depicted therein is an arrangement **2300** where a client UE device may be configured to perform certain aspects of 360° immersive video optimization for purposes of an embodiment of the present patent disclosure. User **2310** having a suitable 360° display device is operative with a connected UE device **2302** that includes a video optimization client module **2306** and a connected player **2308** disposed to generate suitable playback signals to the display device. In one embodiment, player **2308** may comprise an HEVC or AV1 player configured with appropriate video decoder **2314**, display renderer **2316**, audio decoder **2318**, and sound renderer **2320**. Similar to an example embodiment set forth hereinabove, a gaze tracking module **2312** may be provided with the connected UE device **2302**, which may be configured to consume 360° immersive video content delivered over the Internet **2304** in an ABR streaming environment.

Client optimization module **2306** preferably includes a 360° immersive video interface module **2321** comprising a manifest parser **2328**, a video tile and audio stream downloader **2330**, a bandwidth estimation module **2326** and a tile selection module **2324**, which may be configured to operate in a manner similar to the embodiments set forth hereinabove with suitable device-centric modifications, mutatis mutandis. An HEVC tile/audio request **2344** may be generated to a network location, e.g., a content provider network or a cloud-based storage, via the Internet **2304**, based on a manifest **2340** with respect to a particular content. Requested video tiles and audio data may be received via path **2342**. Gaze vector information provided to the immersive video interface module **2321** from the gaze tracking module **2312** (e.g., via path **2322**) may be utilized along with bandwidth estimation in selecting tiles per frame, which may be provided via a video signal path **2331** to a dynamically allocated video buffer **2332**. Likewise, corresponding audio segments may be provided to an audio buffer **2336** via an audio signal path **2338**. Tiles of different qualities may be provided to a tile combiner **2334**, which generates a muxed encoded video stream **2346** to the player's video decoder **2314**. Encoded audio stream **2348** may be generated from the audio buffer **2336** to the audio decoder **2318**. Decoded audio and video data provided to the respective renderers **2320**, **2316** of the player **2308** are rendered appropriately for display/presentation in an immersive environment effectuated by the user's display device, essentially similar to the example embodiments set forth previously.

FIG. **24** depicts a block diagram of a computer-implemented apparatus that may be (re)configured and/or (re) arranged as a platform, node or element to effectuate one or more aspects of 360° immersive video processing, preparation and tile selection optimization according to an embodiment of the present invention. Depending on implementation and/or network architecture, apparatus **2400** may be configured or otherwise integrated in different arrangements suitable for operation at one or more hierarchical levels of an example environment (e.g., as shown in FIGS. **1** and **2**). One or more processors **2402** may be provided as part of a suitable computer architecture for providing overcall control of the apparatus **2400**, wherein processor(s) **2402** may be configured to execute various program instructions stored in appropriate memory modules or blocks, e.g., persistent memory **2408**, including additional modules or blocks specific to media preparation, preprocessing, BIE/PE-based tile encoding including adaptive bitrate encoding/transcoding, optimized tile selection and bandwidth annealing, tiled media packaging, tile stitching, etc. as described in detail hereinabove. For example, such modules may include tile-based PE/BIE encoder **2404**, ABR encoder/transcoder **2406**, GV processing and tile weight processing module **2413**, tile selection and annealing module **2416**, packager and manifest generator **2410**, projection mapper **2418**, and the like. Also, a packaged media database **2419** may be provided in an example embodiment depending on the implementation of apparatus **2400**. Accordingly, various network interfaces, e.g., I/F **2414-1** to **2414**-L, operative for effectuating communications with network infrastructure elements including video back office elements, DRM entities, origin servers, client controller nodes, source media nodes, management nodes, and cache databases as well as interfaces **2412-1** to **2412**-K for effectuating communications sessions with one or more downstream nodes, e.g., including delivery servers, DSLAM/CMTS elements, RAN infrastructure elements, premises gateway nodes, etc., may be provided as part of the apparatus **2400** depending on the network hierarchical level and/or integration.

FIG. **25** depicts a block diagram of an example client UE device or subscriber station **2500** configured for performing various client-side processes according to one or more embodiments of the present patent disclosure. Client device **2500** is generally representative of various viewing devices illustrated in one or more Figures described above, and may include appropriate hardware/software components and subsystems configured for performing any of the device-side processes (either individually or in any combination thereof) with respect to media request generation, gaze vector generation, tile selection and bandwidth estimation, among others, depending on implementation. One or more microcontrollers/processors **2502** are provided for the overall control of the client device **2500** and for the execution of various stored program instructions embodied in one or more persistent memory modules that may be part of a memory subsystem **2511** of the device **2500**. For example, 360° immersive video client applications **2513**A including VR applications may be operative with a bandwidth estimator **2513**B and associated tile selector **2513**C, that may be provided as part of the memory subsystem **2511**. A manifest parser **2517** may be provided to facilitate the generation of media requests to appropriate locations. Controller/processor complex referred to by reference numeral **2502** may also be representative of other specialty processing modules such as graphic processors, video processors, digital signal processors (DSPs), and the like, operating in association with suitable video and audio interfaces (not specifically shown). Appropriate network interfaces such as network I/F modules **2504** and **2506** involving or operating with tuners, demodulators, descramblers, MPEG/H.264/H.265/AV1 decoders/demuxes may be included for processing and interfacing with IPTV and other content signals received via a DSL/CMTS network **2598** or a satellite network **2596**. Where an STB is configured as an example client device or application, suitable demodulators may also be included. One or more media players **2514** may be provided for operating in conjunction with the other subsystems of the client device **2500**, e.g., user interface **2520**, which may be further configured with additional subsystems for facilitating user control over media playback, including channel change requests and any trick mode operations. For example, client/user control functions may include pausing, resuming, fast forwarding, rewinding, seeking, bookmarking, etc. with respect to a particular 360-degree immersive video asset that is being played. Example media players may be configured to operate with one or more A/V coder/decoder (codec) functionalities based on known or hereto unknown standards or specifications.

Other I/O or interfaces such as an immersive display interface **2515**, touch screen or keypad interface **2520**, USB/HDMI ports **2518**, Ethernet I/F **2508**, and short-range and wide area wireless connectivity interfaces **2512** may also be provided depending on device configuration. Various motion detection and gaze tracking sensors **2516** may also be included, some of which may comprise gyroscopes, accelerometers, position sensors, etc. A hard disk drive (HDD) or local DVR system **2510** may be included in an example implementation for local storage of various program assets. A suitable power supply block **2522** may include AC/DC power conversion to provide power for the device **2500**. It should be appreciated that the actual power architecture for the device **2500** may vary by the hardware platform used, e.g., depending upon the core SoC (System-on-Chip), memory, analog front-end, analog signal chain components and interfaces used in the specific platform, and the like.

In further aspects of the present invention, embodiments are set forth below relating to supporting multisession 360° immersive video services in an example network environment where a shared bandwidth pipe may be dedicated for carrying multiple 360° immersive video sessions concurrently. By way of illustration, a sliced network architecture involving 5G communications is discussed immediately below to provide a context for such additional embodiments. However, it will be understood by skilled artisans that other network architectures and technologies, with or without network slicing, may also be configured for practicing an embodiment involving multisession 360° immersive video services according to the teachings herein. In general, a network architecture that supports a dedicated bandwidth for a group of multiple clients in a bandwidth-constrained session management framework (e.g., using a virtual bandwidth pipe configured to serve a plurality of subscribers operating tethered or untethered client devices) may be arranged in combination with the tile selection and bandwidth management scheme set forth hereinabove for an individual 360° immersive video session such that appropriate amounts of bandwidth may be assigned to each of the sessions in a multisession environment as will be described in detail further below.

From a functional and structural point of view, a network slice is an independent end-to-end logical network that runs on a shared physical infrastructure, configured to provide a negotiated service quality with respect to the services hosted thereon (e.g., multisession 360° immersive video services). A network slice arrangement could span across multiple parts and/or hierarchical levels of the network, e.g., subscriber end station, access network, core network and transport network, and could also be deployed across multiple network operators, with suitable slice mapping functions configured to connect slices at level to slices of adjacent levels. Further, a network slice at a particular level, e.g., access network, core network, etc., may be built based on a plurality of network functions (NFs), which may be virtualized (VNFs) and/or cloud-centric in certain arrangements.

In general, a network slice may comprise dedicated and/or shared physical/virtualized resources such as, e.g., processing, storage, and bandwidth, and may be provided with structural/logical isolation from other network slices of a network environment. A network slice management framework comprising a service instance layer, a network slice instance layer and a resource layer may be provided for facilitating various slice-centric requirements in an example arrangement, e.g., creating and managing network slice instances; creating and managing services-to-slices mapping/pairing functions; creating and managing inter-level slice mapping/pairing functions (for instance, one access slice to one core network slice, or one access slice to many core network slices, etc.); resource management and allocation; deployment of suitable fault, configuration, accounting, performance and security (FCAPS) functions, and the like.

It will therefore be realized that network slicing allows core/access networks to be logically separated, with each slice providing customized connectivity, and potentially where all slices may be running on the same, shared infrastructure or on separate infrastructures as the operator(s) may require. Whereas such logical slicing of the resources may be deployed in various communications infrastructures, networks such as 5G networks that support a flexible decoupling between control plane and user plane (leading to a full separation in certain implementations) are more adaptable for leveraging end-to-end slicing, particularly when coupled with software-defined networking (SDN) and cloud-centric implementations. For applications such as 360-degree video services, implementation in a dedicated 5G network slice portion having a shared/dedicated bandwidth resource for video delivery can be particularly advantageous, especially in view of the extremely low latency of 5G networks along with higher bandwidth 5G offers, resulting in the ability to deliver a better QoE for a plurality of video sessions sharing the slice bandwidth capacity.

Although 5G networks offer a higher level of bandwidth for bandwidth intensive applications such as 360-degree video services, several constraints remain especially with respect to multisession environments involving several media streams that may even be generated based on highly efficient compression technologies. For example, to fully optimize the quality in a VR headset, which is currently a 2K-resolution device, requires an 8K stream in today's implementations. These streams require roughly 35-40 Mbs encoded bitrate with HEVC to maximize the QoE for the user. A 5G implementation may be configured to support a fairly large number of sessions given these existing bitrates, e.g., a 2 Gbs network slice could be set aside for 360° video sessions that would allow approximately ~50 concurrent video sessions to be run at the highest quality within the 2 Gbs slice. As network slices having a dedicated bandwidth capacity can be configured on a tower-by-tower basis, the 50 or so sessions may also be established on a tower-by-tower basis in one implementation.

Due to the ongoing improvements in the man-machine interface (MMI) technologies, high performance headsets having 4K and even 8K resolutions are expected to be deployed in the near future, which may require a full 360 video resolutions of 16K and 32K, respectively, to achieve the best possible quality. Even using the latest compression technologies, video streams for supporting such qualities may require bitrates well over 800 to 1000 Mbs each. With a given slice bandwidth capacity of 2 Gbs, the number of multiple sessions having the highest quality that can be supported per tower dramatically drops to just around 13 sessions. Although client-driven bitrate selection may be provided in an ABR-based implementation at an individual session level, several lacunae remain with respect to managing bandwidth allocation at the slice level in a multisession 360-degree video delivery scenario.

Broadly, embodiments herein are directed to a bandwidth optimization scheme in a network environment configured to support multiple 360-degree sessions in a dedicated bandwidth pipe, e.g., supported in a slice architecture configured to serve a plurality of subscribers (potentially in different premises, domains, geographical areas, etc., and potentially using myriad access technologies, devices, and the like), where bitrate qualities of tiles for individual video streams are optimized based on the respective users' gaze vector information and individual bitrate caps. In addition, the overall bitrate allocations for the video sessions corresponding to the individual video streams may also be optimized based on appropriate bandwidth management techniques, e.g., weighted allocation, etc.

Turning to FIG. **26**, depicted therein is an example network environment **2600** configured for providing multiple 360° immersive video sessions according to an embodiment of the present invention. Skilled artisans will recognize that various portions of the network environment **2600** involving media preparation, tile-based encoding/transcoding, packaging, media asset storage, etc., are similar to the corresponding portions of the network environment **200** described above in detail in reference to a single 360° immersive video session, except that multiple media source streams **2602-1** to **2602**-N are provided to one or more CDN origin servers **2604** coupled to a CDN **2606** that is configured to distribute the media streams to one or more edge servers **2608** for subsequent processing and packaging. Accordingly, at least portions of the description of FIG. **2** are equally applicable here, mutatis mutandis, wherein suitable tile encoder(s) **2610** are operative to generate groups of pluralities of tiled adaptive bitrate streams **2612**(N) corresponding to respective media source streams **2602-1** to **2602**-N. Likewise, one or more video packagers **2614**(N) may be configured to package the respective pluralities of tile-encoded bitstreams (e.g., in phase-encoded form or in block-intra encoded form) and generate suitable stream manifests **2641**(N) for storage at one or more storage nodes **2640**(N).

In accordance with the teachings of the present invention, a network node **2616** may be configured to interface with a delivery network management portion **2632** operative to support a dedicated bandwidth pipe optimized for carrying multiple 360° immersive video sessions with a plurality of subscribers/client devices. In one arrangement, the dedicated bandwidth pipe may exist within a larger network pipe operative to support various other services, e.g., telephony, data services, non-360° video services, and the like. In another arrangement, the dedicated bandwidth pipe for multisession 360° immersive video may be configured as a standalone pipe. Furthermore, regardless of whether configured as part of a larger network pipe or as a standalone pipe, the dedicated bandwidth pipe may be implemented in a non-sliced network architecture or a network slice architecture, as noted previously. Similar to the arrangement set forth in FIG. **2**, a video optimization system **2615** comprising a plurality of modules or subsystems is operative in association with a video back office system **2638** for effectuating multisession 360° immersive video services with respect to the multiple subscribers/devices, here exemplified by client device **2636**, via suitable access network infrastructure(s), including a dedicated network/slice portion **2630**. In one embodiment, a bandwidth manager **2699** is operative with a plurality of tile selection (TS) modules **2618-1** to **2618**-N for facilitating optimization of tile selections for individual video streams corresponding to the multiple 360° immersive video sessions as well as optimizing an overall bandwidth allocated to service the multiple 360° immersive video sessions. On individual session/stream level, TS modules **2618-1** to **2618**-N are operative similar to the TS and bandwidth annealing processes set forth hereinabove with respect to selecting appropriate tiles. Accordingly, for respective streams/sessions, tiles **2697-1** to **2697**-N are selected that may be provided to respective tile mux/stitcher modules **2695-1** to **2695**-N for generating corresponding muxed streams **2693-1** to **2693**-N. One or more delivery servers **2691-1** to **2691**-N are operative to interface with delivery network management portion **2632** coupled to the dedicated network/slice portion **2630** for effectuating media delivery with respect to the plurality of 360° immersive video sessions.

In one arrangement, the dedicated network/slice portion **2630** of the network environment **2600** may be implemented as a logical network slice comprising at least a portion of a 5G communications network architecture, a fixed wireless network architecture, a Next Generation Network (NGN) architecture, a Digital Subscriber Line (DSL) network architecture, a Data Over Cable Service Interface Specification (DOCSIS)-compliant Cable Modem Termination System (CMTS) architecture, a switched digital video (SDV) network architecture, Fiber Optic Services (FiOS) network architecture, and a Hybrid Fiber-Coaxial (HFC) network architecture.

In an embodiment where a sliced network architecture is implemented, network node **2616** and/or associated delivery network management portion **2632** may include a management and control loop interface to leverage the advantages of network slicing including, e.g., slice policy control as well as analytics-driven "to-device" bandwidth reporting. An example implementation may therefore involve an interface operative with a management and control loop framework which may be referred to as a Control, Orchestration, Management, Policy, and Analytics (COMPA) entity operative in association with a sliced network. Depending on the COMPA interfacing, an embodiment of the present invention may be configured to generate and provide suitable controls that may be reported to the video optimization system/node **2616**/**2615** in order to dynamically (re)define/(re)size the pipes to the network slice and adjust/readjust the individual bandwidth allocations of the multiple 360° immersive video sessions accordingly. Alternatively or additionally, the video optimization system/node **2616**/**2615** may be configured to report the size of the bandwidth pipe to a COMPA system associated with the network slice, whereby the COMPA system sizes the dedicated slice's bandwidth accordingly. For flow control, an example COMPA system may be configured with suitable network interfaces allowing the video optimization system/node **2616**/**2615** to set the amount of bandwidth on a per flow basis within the slice or operate on a basis where the actual slice is not managed per flow and the delivery system generates the streams based on the internal policy management within the larger slice. Accordingly, in one arrangement, whether the video optimization system is controlling the data flows or not, the system may be configured to manage the bandwidth allocated on a per flow basis, e.g., as part of the functionality of bandwidth manager **2699** coupled to the tile selection modules **2618-1** to **2618**-N described hereinabove.

In an embodiment involving the 5G mobile network architecture, a "to-device" bandwidth reporting may be provided to facilitate advanced analytics and device metrics on per-device basis. In the case where an embodiment is configured to allocate a set amount of bandwidth, it will be appreciated that there may be situations where a particular client device is experiencing lower bandwidth than the system's allocation due to poor transmission conditions, e.g., low signal-to-noise ratio (SNR). In such a case, an embodiment of the present invention may be configured to generate a control signal to the network to report the actual bandwidth to the client device on a dynamic basis, which may be facilitated through the COMPA interface associated therewith.

As noted above, COMPA is derived from performing the key tasks of Control, Orchestration, Management, Policy, and Analytics with respect to a network slice or slices instantiated on a shared/private network infrastructure, which may be configured to provide a modular framework as well as an architecture for a federation of control units exhibiting a common underlying control signaling loop pattern. In one arrangement, COMPA may be configured to operate at different levels in a carrier network. At any given layer, it orchestrates and manages the layers resources to deliver services. Those resources may in turn contribute to the services exposed by the receiving layer. Accordingly, a recursive pattern with multiple, nested levels of control may be implemented in a sliced network architecture configured to support multisession 360° video services. In one embodiment, the following underlying building blocks of COMPA may be provided: Control/Orchestration/Management (COM) executing control decisions; Policy (P) responsible for intelligence and dynamic governance to achieve control decisions; and Analytics (A) providing real time insights to influence control decisions. The COMPA architecture may be configured to define a number of sample federated domain layers (responsibility domains) with specific functionalities such as, e.g.: "Customer/Tenant" (service and cloud user), "Network Functions" (RAN, fixed access, etc.), "Cloud Infrastructure" (Data Centers), and "Transport Networks" (Access, Metro, Edge, Core). Such operational layers may be supported by one or more cross-cutting "Multi-Layer Federation" and "Business" layers for continuous operations across the federated domain layers.

Additional details regarding COMPA architecture may be found in the publication entitled "Using the COMPA Autonomous Architecture for Mobile Network Security", by Fallon et al., published in May 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), and at: https://www.ericsson.com/res/thecompany/docs/publications/ericsson_review/2014/er-evolved-network-architecture.pdf, incorporated by reference herein.

In one embodiment, various messages, requests and responses relative to a particular 360° media session in the multisession network environment **2600** may be set forth as follows. Broadly, device-generated messages may comprise a request **2665** for a 360° immersive video session as well as gaze vector(s) **2667** (e.g., at various subsequent times), which are received at the dedicated delivery network/slice management interface **2632**. Responsively, the management interface **2632** is operative to generate a request **2687** for the requested video asset to the video back office system **2638**, including other relevant parameters, e.g., display resolution and other device characteristics. The received request is processed by the video back office system **2638**, which engages in a messaging session **2652** with the video optimization system **2615** with respect to the requested 360° media asset, wherein a message may be generated for providing a session ID and associated location/manifest information for the requested 360° media as well as a weight/priority configured for the requested session. The session ID as well as session URL information may be provided to the management interface **2632** via a response message **2689**. The management interface **2632** is further operative to provide the gaze vector information and associated session ID for the user **2636** via a message **2685**. Additionally, the management interface **2632** is operative to provide an indication of the dedicated network/slice's bandwidth pipe size as well as a dynamic bandwidth level for the requested video session having the session ID to the video optimization system/node **2616**/**2615**, as indicated by messages **2681** and **2683**, respectively.

As will be set forth further below, the bandwidth manager **2699** of video optimization system/node **2616**/**2615** may be configured to execute a bandwidth optimization/annealing process for determining a bandwidth amount that can be allocated to the requested video session without violating the applicable bandwidth constraints (either at the individual session bitrate caps or suggested/recommended limits, overall dedicated bandwidth size, or both). The tile selection subsystem of video optimization system/node **2616**/**2615** may be configured to operate in response to control messages relative to the assigned bandwidth allocation, user gaze vector information, or both, for selecting tiles having different video qualities, which may be combined or stitched into frames in order to generate a muxed tile-encoded video output stream for the requested video session in a manner similar to the processes set forth previously, e.g., processes **1700**B and **2000** in FIGS. **17**B and **20**, respectively, inter alia.

FIG. **27** depicts an example network slice architecture **2700** that may be configured to support multisession 360° immersive video services according to an embodiment of the present invention. A 360° immersive video content source network **2702** is operatively coupled to one or more external networks **2704** that are coupled to a virtualized core network **2706** that may be configured to support plurality of core network slices **2710-1** to **2710**-N. Each of the core network slices **2710-1** to **2710**-N may be coupled to a corresponding radio access network slice, each RAN slice having specific QoS, bandwidth, resource allocation, etc. By way of example, RAN slices **2716-1** to **2716**-N may be configured to support a virtualized RAN infrastructure **2712** that supports a variety of services and associated end users, e.g., multisession 360° immersive video devices **2710**, Voice-over-LTE (VoLTE) devices **2722**, M2M/IoT devices **2724** that may include autonomous vehicle communications, sensor communications, etc., as well as connected medical instrumentation devices **2726**, among others. As illustrated in this FIGURE, an end-to-end slice **2728** comprising a RAN slice and associated core network slice that are suitably coupled in an operator's slice mapping functionality may be configured to support a virtual pipe of dedicated bandwidth for managing delivery of multisession 360° immersive video services.

FIG. **28** depicts an example network slice arrangement **2800** operative with a multisession network slice video optimization system in accordance with an embodiment. A 360° immersive video client device **2850** is exemplary of a plurality of client devices operatively associated with a network slice **2802** optimized for 360° immersive video as set forth in the present patent application. By way of illustration, the network slice **2802** may be configured to support a COMPA system **2814** that may include a resource monitor subsystem **2816**, an analytics subsystem **2818**, a slice size policy subsystem **2820** (e.g., with respect to a bandwidth size dedicated to the slice) as well as a per-stream flow control policy subsystem **2822**. A multisession network slice (MNS) video optimization system **2824** may be configured as an embodiment operative with or part of the video optimization system **2615** described above in reference to FIG. **26**. In one arrangement, the network slice **2802** may be configured to support an edge subnet with tile cache **2828** as well as a tile encoder/packager subnet **2826**, which interfaces to a CDN edge 360-video stream subnet **2830** that provides source media streams **2832**. Accordingly, one skilled in the art will recognize upon reference hereto that the example network slice arrangement **2800** is a slice-based implementation of at least a portion of the network environment **2600** shown in FIG. **26**, wherein encoded tiles **2836** are provided to the MNS video optimization system **2824** based on one or more tiled video asset packages **2834** corresponding to the multiple video streams/sessions supported by the example network slice arrangement **2800**. MNS video optimization system **2824** is further operative to receive slice size indication **2840** and session bitrate indication **2838** from COMPA subsystem **2814**, which may be configured to receive a dynamic bandwidth reporting **2842** from example video client device **2850**. MNS video optimization system **2824** is further operative to receive user gaze vector information **2844** from example video client device **2850**. Responsive at least in part to the various control inputs, an optimized 360° immersive video bitstream **2846** may be provided to the client device **2850** for decoding and playback.

In general operation, the overall functionality of COMPA subsystem **2814** of the example network slice arrangement **2800** may be configured to provide a guaranteed bitrate, adjust to a limited device bitrate and manage the multiple sessions flowing within a 360° managed delivery network slice. Depending on implementation, COMPA subsystem **2814** and MNS video optimization system **2824** may interact in a number of ways to facilitate the transfer of appropriate control signaling with respect to slice bandwidth policy control. In one arrangement, COMPA subsystem **2814** may be configured to provide policy control indications in a push model that includes sending a dedicated network slice size and dynamic bandwidth level for a 360° managed video session with session ID or flow ID to MNS video optimization system **2824**. In another arrangement, MNS video optimization system **2824** may send a request to COMPA subsystem **2814** for a slice policy control, which is then received in response (e.g., request/response model or pull model). Skilled artisans will clearly recognize that various other ways of transferring slice policy control information between COMPA subsystem **2813** and MNS video optimization system **2824** may also be implemented in an example embodiment of the present invention.

Because an embodiment of the 360° immersive video optimization system may require extremely low latency for both network transmission and decoding, it is desirable that the system is optimally located at the extreme edge of a network infrastructure. In a 5G implementation, it would preferably be located at a base station. Turning to FIG. **29**, depicted therein is an example 360° immersive video network layout **2900** with different latencies according to an embodiment of the present invention, wherein an access portion **2904**C, an aggregation portion **2904**B and a centralized/core portion **2904**A are exemplified for potential placement/deployment of a 360° immersive video optimization system with respect to a video content source network **2902**. Centralized portion **2904**A may include a video ingest functionality **2908** with respect to a source stream **2906**, which may be provided to a video distribution edge module **2910** coupled to a converged packet core's user plane of the aggregation portion **2904**B. Typically, the aggregation portion **2904**B may include source video caching as well as tile encoding and packaging functionalities **2914**. In an example implementation where a 360° immersive video optimization system is deployed at this level, moderate latencies of around 30-300 milliseconds may be seen. For lower latencies (e.g., around 3 milliseconds), a VNF **2916** configured for video optimization may be provided at the access portion **2904**C, where the following functionalities may be executed: viewport optimized tile selection, assembly and delivery; multisession control and policy management; and provisioning a dedicated network slice, as illustrated at block **2926**. Converged packet core's user plane portion **2918** of the access portion **2904**C, which may be operably coupled to 360-degree video VNF **2916**, may include VRAN connectivity **2922** and WiFi connectivity **2924** that may serve one or more 360-degree video client devices **2999** via suitable infrastructure elements **2997**A/B.

FIGS. **30**A-**30**C depict flowcharts illustrative of various blocks, steps and/or acts relative to bandwidth optimization in a network slice or a dedicated network environment configured to support multiple 360° immersive video sessions according to one or more embodiments of the present invention. In particular, process **3000**A set forth in FIG. **30**A depicts an embodiment of a bandwidth optimization method that may be executed at a video optimization system wherein a number of control inputs are utilized in configuring the amount of bandwidths allocated to respective 360° video sessions in a shared/dedicated bandwidth pipe. At block **3002**, a bandwidth capacity or size indication may be received with respect to a bandwidth capacity dedicated to a network portion configured for supporting a plurality of 360° video sessions in a shared bandwidth pipe. As set forth previously, each video session is operative to serve a corresponding client device for playing an immersive video asset, wherein each video frame comprises an array of tiles projected on a 3D display environment viewed by a user operating the client device. At block **3004**, a plurality of weights corresponding to the plurality of 360° video sessions may be received, e.g., based on operator, subscriber and/or content provider policy configuration control, wherein each weight is operative to indicate a priority value associated with a respective video session. At block **3006**, a plurality of bitrate caps or recommended bitrate settings may be received corresponding to the plurality of 360° video sessions, e.g., from a network slice manager, wherein each bitrate cap is indicative of a bandwidth limit associated with a respective video session. At block **3008**, responsive to the bandwidth capacity indication, total number of the plurality of 360° video sessions and corresponding pluralities of weights and bitrate caps, a determination may be made regarding respective bandwidth allocations that can be assigned to each of the multiple 360° video sessions without violating the bitrate cap of the 360° video session. At block **3010**, the respective bandwidth allocation may be provided to a tile selection process operative to select tiles for a corresponding video session on a frame-by-frame basis from a plurality of tile-encoded bitrate representations of a media input stream associated with the video session, wherein each tile-encoded bitrate representation is provided with a separate video quality. As set forth at block **3012**, the bandwidth allocation/optimization process may continue until a triggering event to affect bandwidth allocation is detected, reported or otherwise obtained.

In one embodiment, process **3000**A may also involve a step or block where bitrates of at least a portion of the selected tiles for a particular video session are optimized based on gaze vector information received from the client device associated with the video session. Process **3000**B shown in FIG. **30**B sets forth additional steps that may be augmented in an embodiment of process **3000**A. At block **3022**, a determination may be made as to whether a triggering event has occurred with respect to the shared bandwidth pipe of the network portion. If so, responsive to the determining that a triggering event has occurred, the bandwidth allocations of respective 360° video sessions may be (re)adjusted to optimize usage of the bandwidth capacity dedicated to the network portion, as set forth at block **3024**. Similar to process **3000**A, an example embodiment of process **3000**B may continue until a further triggering event is reported, e.g., in an iterative loop manner (block **3026**). In one implementation, a triggering event with respect to any combination of processes **3000**A/**3000**B and/or other bandwidth optimization processes set forth herein may comprise at least one of the following: a (new) client device commencing a new 360° video session for receiving content via the shared bandwidth pipe, an existing client device terminating/pausing an ongoing 360° video session, receiving an indication that a particular video session is experiencing a poor quality resulting in reduced bitrate level, and receiving a policy configuration change (e.g., pipe resizing) with respect to the bandwidth capacity dedicated to the bandwidth pipe, and/or any combination thereof.

Process **3000**C shown in FIG. **30**C is illustrative of still further steps, blocks, acts and/or functions that may be combined with one or more other processes according to additional/alternative variations within the scope of the present patent application. At block **3042**, a determination may be made to determine a surplus in individual bandwidth allocations assigned to respective 360° video sessions based on the corresponding bitrate caps. At block **3044**, an overall excess amount of bandwidth available in the shared bandwidth pipe based on the surplus may be obtained. At block **3046**, the overall excess amount of bandwidth may be (re)distributed to the plurality of 360° video sessions according to the respective weights associated with the plurality of 360° video sessions. At block **3048**, an embodiment of process **3000**C may continue until a further triggering event is reported, e.g., in an iterative loop manner, similar to the processes **3000**A and **3000**B set forth above.

FIGS. **31**A-**31**D depict flowcharts illustrative of various blocks, steps and/or acts relative to additional details with respect to at least some portions of the foregoing processes according to one or more embodiments of the present invention. In particular, process **3100**A shown in FIG. **31**A sets forth a bitrate allocation process in a virtual bandwidth pipe or slice based on weights. At block **3102**, various inputs such as total slice/pipe bandwidth (B) **3104**A, number (N) of plurality of sessions **3104**B, and an array of weights (W) **3104**C corresponding thereto are retrieved, received or otherwise obtained. At block **3106**, a sum (S) of weights is determined. At block **3108**, a bandwidth array (BW) of size N is initialized for commencing an iterative loop process by initializing an increment, i (block **3110**). At block **3112**, bandwidth for i^{th} session is determined as a relative proportion or ratio of the total bandwidth based on the ratio of the weight of the i^{th} session to the sum of the weights. Bandwidths for the rest of the sessions are similarly determined in an iterative manner as set forth in block **3114** and **3116**. Upon completion of the iterative process, bandwidths are allocated to respective sessions according to the values of the BW array (block **3118**).

Skilled artisans will recognize that the allocation process **3100**A above may be dynamically affected by various triggering events as noted previously. Accordingly, for example, the number (N) of sessions and/or total slice bandwidth (B) may change from time to time, which can trigger redistribution/readjustment of the BW(i) array as needed.

In yet another variation, an example bandwidth (re) allocation process may involve applying a per-session bitrate limit for given a plurality of sessions, array of bandwidths per session and corresponding session weights, which may be formalized as an optimization process subject to certain constraints as set forth below where a goal function is to be maximized:

Goal Function:

$$\max \sum_{i=0}^{N-1} BW[i]$$

Input (Constants):
SLICE=Total slice bandwidth
Weights

$\forall_{0 \le i \le N-1} W[i]$

Bandwidth limits

$\forall_{0 \le i \le N-1} \text{limit}[i]$

Variables:
Allocated bandwidths

$\forall_{0 \le i \le N-1} \text{BW}[i]$

Constraints:

$$\sum_{i=0}^{N-1} BW[i] \le \text{SLICE}$$

//total allocated bandwidth≤total bandwidth of slice

$\forall_{0 \le i \le N-1} \text{BW}[i] \le \text{limit}[i]$

//no bandwidth overflows session limit

$$\forall_{0 \le i,j \le N-1} \text{ if } (BW[i] < \text{limit}[i] \text{ AND } BW[j] < \text{limit}[j]) \Rightarrow \frac{BW[i]}{BW[j]} = \frac{W[i]}{W[j]}$$

$$\forall_{0 \le i,j \le N-1} \text{ if } (BW[i] = \text{limit}[i] \text{ AND } BW[j] < \text{limit}[j]) \Rightarrow BW[i] \le \frac{W[i]}{W[j]} BW[j]$$

Output:
(Re)allocated bandwidths

$\forall_{0 \le i \le N-1} \text{BW}[i]$

In accordance with the above formulation, an embodiment may involve performing a pair-wise comparison between sessions (e.g., between [i] and [j] sessions) so as to determine if session [j] has a current bandwidth allocation that is less than its limit, i.e., an excess, which may be redistributed to another session, i.e., session [i]. Process **3100**B depicted in FIG. **31**B sets forth a flowchart of a method for clipping bandwidths according to per-session limits and determining an overall excess (i.e., based on a per-session surplus determination). At block **3122**, various inputs are obtained such as the number (N) of sessions (block **3124**A), an array of bandwidths (BW[i]) for the sessions (block **3124**B), and an array of bandwidth limits (L[i]) for the sessions (block **3124**C). At block **3126**, an initialization may be made with respect to an iterative loop process where individual bandwidth surpluses for each session may be computed and summed as set forth at blocks **3128**, **3130**, **3132** and **3134**. Further, where a bandwidth allocation of a session (e.g., as determined by process **3100**A) is greater than the session's limit, its bandwidth may be set or clipped at the limit (block **3130**). At block **3136**, appropriate output of excess bandwidth and initial allocation of bandwidth are provided.

As noted previously, several triggering events may be monitored and reported which may cause (re)execution of at least some of the processes set forth herein, including process **3100**B for determining dynamically varying excess bandwidth. For example, in addition to clients joining or leaving 360-degree video sessions, a bitrate notification may be reported when a client changes its bandwidth, whereby an updated limit for that particular session may be provided to update the L[i] array. Accordingly, in one embodiment, after an initial allocation of the bandwidth for a current list of sessions has been set (e.g., by example process **3100**A), an updating process may be executed/triggered either periodically or on demand to determine excess bandwidth that may be (re)distributed based on pair-wise comparisons, weight proportions, and the like. FIG. **31**C depicts an embodiment of an excess bandwidth (re)distribution process **3100**C, wherein the excess bandwidth of a dedicated pipe may be partitioned among one or more sessions that have not yet reached their respective bitrate caps (i.e., bandwidth limits). At block **3150**, various inputs are obtained, retrieved or received, which may include the number (N) of sessions (block **3152**A), array of bandwidths (BW[i]) for the sessions (block **3152**B), array of bandwidth limits (L[i]) for the sessions (block **3152**C) as well as the excess bandwidth (block **3152**D). As before, an initialization may be made (block **3154**) with respect to an iterative loop process where the excess bandwidth is apportioned for each session that has a bandwidth allocation less than its cap, as set forth at blocks **3156**, **3158**, **3160** and **3162**. Thereafter, a redistributed BW[i] array may be output as set forth at block **3164**.

In a still further embodiment, an example process **3100**D for selecting the allocated bitrates per session that takes into account both session weights and limits is set forth in FIG. **31**D, which may in one arrangement be implemented as a combination of the foregoing processes, e.g., processes **3100**A-**3100**C. As before, at block **3170**, various inputs are obtained, retrieved or received, which may include total slice bandwidth (W) (block **3172**A). the number (N) of sessions (block **3172**B), array of session weights (W[i]) (block **3172**C) as well as array of bandwidth limits (L[i]) for the sessions (block **3172**D). At block **3174**, a sum (S) of weights and a sum (Sum(L)) of limits are determined. At block **3176**, a determination is made as to whether Sum(L) is less than the total slice bandwidth (B). If so, an intermediary variable B' is set as equal to Sum(L) (block **3178**). Otherwise, the intermediary variable B' is set as equal to the total slice bandwidth (block **3180**). Using B', an embodiment of process **3100**A is executed at block **3182** for determining a bandwidth array (BW[i]). At block **3184**, an embodiment of process **3100**B may be executed to determine bandwidth excess. If an excess bandwidth is determined (block **3186**), it may be distributed across the sessions (e.g., sessions having a current bandwidth below their bitrate caps) by executing an embodiment of process **3100**C, as set forth at block **3188**, until there is no excess bandwidth left. Thereafter, the (re)adjusted bandwidth array for the sessions is provided as output (block **3190**).

Skilled artisans will recognize that the foregoing process **3100**D may be executed by a video optimization system's bandwidth manager at a quadratic computational time complexity order, i.e., ($O\{n^2\}$). In a still further embodiment of the present invention, a methodology combining weighted allocation techniques with bitrate caps in a single pass may be provided in order to improve computational complexity, wherein a "pipe sorting" process may be implemented with respect to the plurality of 360-degree sessions (each being analogized to a "pipe" within the dedicated network/slice size) using a parameter referred to herein as a "safety quotient". For purposes of this embodiment, the safety quotient may be defined as a ratio of cap/weight for each session. In an implementation of this embodiment, the lower this ratio is, the sooner the corresponding session/pipe will be capped as the available bandwidth is increased. When allocating the bandwidth based on safety quotients, an embodiment according to the teachings herein may be configured to sort the session pipes in order of increasing safety so that the least safe pipes will be evaluated first. As bandwidth is allocated, certain parameters defined as a "remaining bandwidth" as well as a "residual weight sum" (which is the sum of the weights of the pipes that have not yet been allocated) are monitored and tracked. An example embodiment implemented in Python script is set forth below by way of illustration.

```
""""
Reconciling caps in a weighted bandwidth allocation process
""""
class Pipe:
  def_init_(self, weight, cap):
  self.weight=weight
  self.cap=cap
  def overbookSafety(self):
  """"the lower the number returned by this function, the
    more likely this pipe will be affected by a data cap""""
  return self.cap/self.weight
def sim(pipes, bandwidth):
  p2=sorted(pipes, key=lambda x:x.overbookSafety( ))
  # the least-safe-most-doomed pipes will come first in the
sorted list.
  denom=sum([pipe.weight for pipe in pipes])
  remainingBandwidth=bandwidth
  print("bandwidth %.2f"% bandwidth)
  for pipe in p2:
  alloc=remainingBandwidth*pipe.weight/denom
  if alloc>pipe.cap:
    alloc=pipe.cap
  print("p(%.1f,%.1f)\tgets %.2f"%(pipe.weight, pipe.cap,
    alloc))
  remainingBandwidth-=alloc
  denom-=pipe.weight
  if (remainingBandwidth>1.0e-10):
  print("unallocated % r\n"% remainingBandwidth)
  else:
  print( )
#
#
#
  pipes=[
  Pipe(3, 4),
  Pipe(3, 3),
  Pipe(1.5, 1),
  Pipe(1.5, 2),
]
  sim(pipes, 6)
  sim(pipes, 8)
  sim(pipes, 9)
  sim(pipes, 12)
```

Example bandwidth allocation output as well as any unallocated bandwidth may comprise as follows:

bandwidth 6.00	
p(1.5,1.0)	gets 1.00
p(3.0,3.0)	gets 2.00
p(3.0,4.0)	gets 2.00
p(1.5,2.0)	gets 1.00
bandwidth 8.00	
p(1.5,1.0)	gets 1.00
p(3.0,3.0)	gets 2.80
p(3.0,4.0)	gets 2.80
p(1.5,2.0)	gets 1.40
bandwidth 9.00	
p(1.5,1.0)	gets 1.00
p(3.0,3.0)	gets 3.00

-continued

p(3.0,4.0)	gets 3.33
p(1.5,2.0)	gets 1.67
bandwidth 12.00	
p(1.5,1.0)	gets 1.00
p(3.0,3.0)	gets 3.00
p(3.0,4.0)	gets 4.00
p(1.5,2.0)	gets 2.00
unallocated 2	

The foregoing methodology only requires a single pass because the pipes are sorted in such a way that once a pipe that is not capped is reached, all subsequent pipes are guaranteed to also be uncapped due to the mathematical constraints. Because this technique depends upon a sorting phase, the overall time complexity is typically in the range of O{(n log(n)}.

In an embodiment of the above process, the sorting stage can be done only once and the pipes can be stored in a binary search tree (BST) where their key equals to their respective safety quotient (defined as cap/weight). Then, to maintain the sorted order, upon detecting or receiving a triggering change (e.g., client joins, client leaves, cap of weight of a client changes, etc.), the methodology only needs to update the pipe of this specific client in the BST, which requires O{log(n)} time. Accordingly, after initial sorting, each update requires only O{log(n)} time to maintain the sorted order and then the bandwidth allocation based on weights and limits only takes O{n} time.

FIG. **32** depicts a flowchart illustrative of various blocks, steps and/or acts relative to bandwidth optimization in a network slice configured to support multiple 360° immersive video sessions according to an embodiment of the foregoing methodology based on weighted allocation with caps and weights. Similar to processes **3100**A-D, process **3200** commences by obtaining/retrieving a number of inputs at block **3202**, which may include total slice bandwidth (W) (block **3204**A), number (N) of sessions (block **3204**B), array of session weights (W[i]) (block **3204**C) as well as array of bandwidth limits (L[i]) for the sessions (block **3204**D). At block **3206**, client sessions are sorted in ascending order based on a safety quotient. At blocks **3208** and **3210**, an initialization process is performed and a sum (S) of weights is obtained. For each client[i], a bandwidth is proportionally allocated in relation to its weight to the sum of weights (block **3212**). At block **3214**, a determination is made as to whether the client's bandwidth is greater than its cap limit. If so, its bandwidth is adjusted and the remaining slice bandwidth is reduced based on the limit accordingly (block **3216**). Otherwise, the remaining slice bandwidth is adjusted based on the allocated session bandwidth and a residual sum of weights is determined by reducing the weight of the client/session being considered (block **3218**). The foregoing updating process is performed iteratively until all sessions/clients have been assigned (block **3220**). At block **3222**, the assigned bandwidths are stored in the bandwidth array (BW[i]), where i^{th} session is allocated bandwidth amount of BW[i]). As noted previously, the allocated bandwidth amounts may be provided to respective tile selection and annealing modules for selecting tiles with appropriate bitrate qualities for each output encoded bitstream.

In at least some of the flowcharts relating to the several processes described above, various Python-based mathematical operators (e.g., ADD AND (+=); SUBTRACT AND (−=); etc.) are shown in the Drawing Figures of the present patent application. One skilled in the art will recognize that such operators and underlying processes may be implemented using other types of operations as long as they are logically/functionally equivalent to the illustrated processes, steps and/or blocks of an exemplary embodiment.

FIGS. **33**A and **33**B depict example virtual pipe and weighted bandwidth allocation scenarios according to an embodiment of the present invention. Example scenario **3300**A in FIG. **33**A is illustrative of a network slice or dedicated virtual pipe **3302** having a 2.0 Gbs capacity that is allocated among 59 video sessions/streams/flows, which are weighted such that a first plurality of sessions are assigned a weight of 3.0 each, a second plurality of sessions are assigned a weight of 1.5 each, and a third plurality of sessions are assigned a weight of 0.75 each. By way of illustration, three HMD client devices **3308-1** for **3308-3** are engaged in sessions having a weight of 3.0 each; two HMD client devices **3308-4** and **3308-5** are engaged in sessions having a weight of 1.5 each; and a tablet client device **3308-6** is engaged in a session with a weight of 0.75, via a dedicated network/slice infrastructure **3305**. An example embodiment of bandwidth allocation according to the present invention allocates the bandwidth to the illustrated client devices as follows: (i) a bitrate of 49.382716 Mbs each for the corresponding sessions/pipes **3304**A-**1** to **3304**A-**3**; (ii) a bitrate of 24.691358 Mbs each for corresponding sessions/pipes **3304**A-**31** to **3304**A-**32**; and (iii) a bitrate of 12.345679 Mbs for the corresponding session/pipe **3304**A-**59**.

Example scenario **3300**B in FIG. **33**B illustrates bandwidth reallocation within the network slice or dedicated virtual pipe **3302** when a session, e.g., session associated with HMD device **3308-3**, is constrained in a multisession scenario. By way of illustration, due to the network bandwidth constraints (e.g., poor SNR), the session is now limited to only 20.0 Mbs, as exemplified by the readjusted session pipe **3304**B-**3**. Excess bandwidth is (re)allocated to the remaining devices, resulting in (re)adjusted pipes for the corresponding sessions as follows: pipes **3304**B-**1** and **3304**B-**2** each having 50.126582 Mbs; pipes **3304**B-**31** and **3303**B-**32** each having 25.063291 Mbs; and pipe **3304**B-**59** having 12.531645 Mbs. One skilled in the art will recognize that a similar readjustment of the session pipes may be obtained when the sizing of network slice or dedicated virtual pipe **3302** is reduced (e.g., to 1.8 Gbs).

Based on the foregoing, it will be appreciated that embodiments herein advantageously provide a bandwidth optimization scheme for supporting multisession 360° immersive video services without sacrificing QoE in a network portion having a dedicated bandwidth capacity. As gaze vector information corresponding to each session is provided to a tile selection mechanism operating within a constrained session pipe, viewport-optimized selection of tiles with highest bitrate quality provides for a superior viewing experience even in more realistic viewing scenarios where the users' gaze may be constantly changing. Embodiments are also particularly advantageous in a sliced network architecture since an example implementation allows for the ability to add policy management on 360° immersive video flows based on content, device, subscriber policies, etc., in addition to controlling the delivery within dedicated network slices.

One skilled in the art will further recognize that various apparatuses and systems with respect to the foregoing embodiments, as well as the underlying network infrastructures set forth above may be architected in a virtualized environment according to a network function virtualization (NFV) architecture in additional or alternative embodiments of the present patent disclosure. For instance, various physical resources, databases, services, applications and functions executing within an example streaming network of the present application, including source media processing infrastructure, media containerization, PE/BIE tile encoding and packaging, etc., set forth hereinabove may be provided as virtual appliances, machines or functions, wherein the resources and applications are virtualized into suitable virtual network functions (VNFs) or virtual network elements (VNEs) via a suitable virtualization layer. Resources comprising compute resources, memory resources, and network infrastructure resources are virtualized into corresponding virtual resources wherein virtual compute resources, virtual memory resources and virtual network resources are collectively operative to support a VNF layer, whose overall management and orchestration functionality may be supported by a virtualized infrastructure manager (VIM) in conjunction with a VNF manager and an NFV orchestrator. An Operation Support System (OSS) and/or Business Support System (BSS) component may typically be provided for handling network-level functionalities such as network management, fault management, configuration management, service management, and subscriber management, etc., which may interface with VNF layer and NFV orchestration components via suitable interfaces.

Furthermore, at least a portion of an example network architecture disclosed herein may be virtualized as set forth above and architected in a cloud-computing environment comprising a shared pool of configurable virtual resources. Various pieces of hardware/software associated with PE/BIE tile encoding and packaging, bandwidth annealing and tile selection, tile muxing and containerization, multisession bandwidth management, and the like may be implemented in a service-oriented architecture, e.g., Software as a Service (SaaS), Platform as a Service (PaaS), infrastructure as a Service (IaaS) etc., with multiple entities providing different features of an example embodiment of the present invention, wherein one or more layers of virtualized environments may be instantiated on commercial off the shelf (COTS) hardware. Skilled artisans will also appreciate that such a cloud-computing environment may comprise one or more of private clouds, public clouds, hybrid clouds, community clouds, distributed clouds, multiclouds and interclouds (e.g., "cloud of clouds"), and the like.

In the above-description of various embodiments of the present disclosure, it is to be understood that the terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of the invention. Unless otherwise defined, all terms (including technical and scientific terms) used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this invention belongs. It will be further understood that terms, such as those defined in commonly used dictionaries, should be interpreted as having a meaning that is consistent with their meaning in the context of this specification and the relevant art and may not be interpreted in an idealized or overly formal sense expressly so defined herein.

At least some example embodiments are described herein with reference to block diagrams and/or flowchart illustrations of computer-implemented methods, apparatus (systems and/or devices) and/or computer program products. It is understood that a block of the block diagrams and/or flowchart illustrations, and combinations of blocks in the block diagrams and/or flowchart illustrations, can be implemented by computer program instructions that are performed by one or more computer circuits. Such computer program instructions may be provided to a processor circuit of a general purpose computer circuit, special purpose computer circuit, and/or other programmable data processing circuit to produce a machine, so that the instructions, which execute via the processor of the computer and/or other programmable data processing apparatus, transform and control transistors, values stored in memory locations, and other hardware components within such circuitry to implement the functions/acts specified in the block diagrams and/or flowchart block or blocks, and thereby create means (functionality) and/or structure for implementing the functions/acts specified in the block diagrams and/or flowchart block(s). Additionally, the computer program instructions may also be stored in a tangible computer-readable medium that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable medium produce an article of manufacture including instructions which implement the functions/acts specified in the block diagrams and/or flowchart block or blocks.

As pointed out previously, tangible, non-transitory computer-readable medium may include an electronic, magnetic, optical, electromagnetic, or semiconductor data storage system, apparatus, or device. More specific examples of the computer-readable medium would include the following: a portable computer diskette, a random access memory (RAM) circuit, a read-only memory (ROM) circuit, an erasable programmable read-only memory (EPROM or Flash memory) circuit, a portable compact disc read-only memory (CD-ROM), and a portable digital video disc read-only memory (DVD/Blu-ray). The computer program instructions may also be loaded onto or otherwise downloaded to a computer and/or other programmable data processing apparatus to cause a series of operational steps to be performed on the computer and/or other programmable apparatus to produce a computer-implemented process. Accordingly, embodiments of the present invention may be embodied in hardware and/or in software (including firmware, resident software, micro-code, etc.) that runs on a processor or controller, which may collectively be referred to as "circuitry," "a module" or variants thereof. Further, an example processing unit may include, by way of illustration, a general purpose processor, a special purpose processor, a conventional processor, a digital signal processor (DSP), a plurality of microprocessors, one or more microprocessors in association with a DSP core, a controller, a microcontroller, Application Specific Integrated Circuits (ASICs), Field Programmable Gate Array (FPGA) circuits, any other type of integrated circuit (IC), and/or a state machine. As can be appreciated, an example processor unit may employ distributed processing in certain embodiments.

Further, in at least some additional or alternative implementations, the functions/acts described in the blocks may occur out of the order shown in the flowcharts. For example, two blocks shown in succession may in fact be executed substantially concurrently or the blocks may sometimes be executed in the reverse order, depending upon the functionality/acts involved. Moreover, the functionality of a given block of the flowcharts and/or block diagrams may be separated into multiple blocks and/or the functionality of two or more blocks of the flowcharts and/or block diagrams may be at least partially integrated. Furthermore, although some of the diagrams include arrows on communication paths to show a primary direction of communication, it is to be understood that communication may occur in the opposite direction relative to the depicted arrows. Finally, other blocks may be added/inserted between the blocks that are illustrated.

It should therefore be clearly understood that the order or sequence of the acts, steps, functions, components or blocks illustrated in any of the flowcharts depicted in the drawing Figures of the present disclosure may be modified, altered, replaced, customized or otherwise rearranged within a particular flowchart, including deletion or omission of a particular act, step, function, component or block. Moreover, the acts, steps, functions, components or blocks illustrated in a particular flowchart may be inter-mixed or otherwise inter-arranged or rearranged with the acts, steps, functions, components or blocks illustrated in another flowchart in order to effectuate additional variations, modifications and configurations with respect to one or more processes for purposes of practicing the teachings of the present patent disclosure.

Although various embodiments have been shown and described in detail, the claims are not limited to any particular embodiment or example. None of the above Detailed Description should be read as implying that any particular component, element, step, act, or function is essential such that it must be included in the scope of the claims. Reference to an element in the singular is not intended to mean "one and only one" unless explicitly so stated, but rather "one or more." All structural and functional equivalents to the elements of the above-described embodiments that are known to those of ordinary skill in the art are expressly incorporated herein by reference and are intended to be encompassed by the present claims. Accordingly, those skilled in the art will recognize that the exemplary embodiments described herein can be practiced with various modifications and alterations within the spirit and scope of the claims appended below.

The invention claimed is:

1. A bandwidth optimization method in a network environment configured to support multiple 360° video sessions, the method comprising:

receiving a bandwidth capacity indication with respect to a bandwidth capacity dedicated to a network portion supporting a plurality of 360° video sessions in a shared bandwidth pipe, each session serving a corresponding client device for playing an immersive video asset, wherein each video frame comprises an array of tiles projected on a 3-dimensional (3D) display environment viewed by a user operating the client device;

receiving a plurality of weights corresponding to the plurality of 360° video sessions, each weight indicating a priority value associated with a respective video session;

receiving a plurality of bitrate caps corresponding to the plurality of 360° video sessions, each bitrate cap indicating a bandwidth limit associated with a respective video session;

responsive to the bandwidth capacity indication, total number of the plurality of 360° video sessions and corresponding pluralities of weights and bitrate caps, determining a respective bandwidth allocation that can be assigned to each 360° video session without violating the bitrate cap of the 360° video session; and

for each 360° video session, responsive to the respective bandwidth allocation assigned thereto, performing:

selecting a plurality of tiles on a frame-by-frame basis from a plurality of encoded bitstreams corresponding to a media input stream associated with the 360° video session, wherein each encoded bitstream comprises a tile-encoded bitrate representation of the media input stream, and wherein each encoded bitstream is provided with a separate video quality; and

combining the selected plurality of tiles for each frame and multiplexing the frames to form a single output stream having a mixed video quality for transmission to a corresponding client device, wherein the selected plurality of tiles of a frame comprise tiles have different video qualities.

2. The method as recited in claim **1**, wherein bitrates of at least a portion of the selected plurality of tiles for a 360° video session are optimized based on gaze vector information received from the client device associated with the video session.

3. The method as recited in claim **1**, wherein the network portion supporting the plurality of 360° video sessions is configured as a logical network slice comprising at least a portion of a 5G communications network architecture, a fixed wireless network architecture, a Next Generation Network (NGN) architecture, a Digital Subscriber Line (DSL) network architecture, a Data Over Cable Service Interface Specification (DOCSIS)-compliant Cable Modem Termination System (CMTS) architecture, Fiber Optic Services (FiOS) network architecture, a switched digital video (SDV) network architecture, and a Hybrid Fiber-Coaxial (HFC) network architecture.

4. The method as recited in claim **1**, further comprising:

determining if a triggering event has occurred with respect to the shared bandwidth pipe of the network portion; and

responsive to the determining that a triggering event has occurred, adjusting the bandwidth allocations of respective 360° video sessions to optimize usage of the bandwidth capacity dedicated to the network portion.

5. The method as recited in claim **4**, wherein the triggering event comprises at least one of a new client device commencing a new 360° video session for receiving content via the shared bandwidth pipe, an existing client device terminating an ongoing 360° video session, and receiving a policy configuration change with respect to the bandwidth capacity dedicated to the bandwidth pipe.

6. The method as recited in claim **1**, wherein the plurality of encoded bitstreams comprising respective tile-encoded bitrate representations are generated as a plurality of phase-encoded bitstreams based on at least one of High Efficiency Video Coding (HEVC) H.265 compression, Alliance for Open Media (AOMedia) Video 1 (AV1) compression and H.266/Versatile Video Coding (VVC) compression.

7. The method as recited in claim **1**, wherein a portion of the encoded bitstreams comprising respective tile-encoded bitrate representations are generated as block-intra-encoded bitstreams based on at least one of High Efficiency Video Coding (HEVC) H.265 compression, Alliance for Open Media (AOMedia) Video 1 (AV1) compression and H.266/Versatile Video Coding (VVC) compression.

8. The method as recited in claim **1**, further comprising:

determining a surplus in individual bandwidth allocations assigned to respective 360° video sessions based on the corresponding bitrate caps;

obtaining an overall excess amount of bandwidth available in the shared bandwidth pipe based on the surplus; and

distributing the overall excess amount of bandwidth to the plurality of 360° video sessions according to the respective weights associated with the plurality of 360° video sessions.

9. A video server system operative in association with a 360-degree immersive video environment, the system comprising:

one or more processors; and

one or more persistent memory modules having program instructions stored thereon which, when executed by the one or more processors, perform following acts in association with one or more modules:

receive a bandwidth capacity indication with respect to a bandwidth capacity dedicated to a network portion supporting a plurality of 360° video sessions in a shared bandwidth pipe, each session serving a corresponding client device for playing an immersive video asset, wherein each video frame comprises an array of tiles projected on a 3-dimensional (3D) display environment viewed by a user operating the client device;

receive a plurality of weights corresponding to the plurality of 360° video sessions, each weight indicating a priority value associated with a respective video session;

receive a plurality of bitrate caps corresponding to the plurality of 360° video sessions, each bitrate cap indicating a bandwidth limit associated with a respective video session;

determine a respective bandwidth allocation that can be assigned to each 360° video session without violating the bitrate cap of the 360° video session responsive to the bandwidth capacity indication, total number of the plurality of 360° video sessions and corresponding pluralities of weights and bitrate caps; and

for each 360° video session, responsive to the respective bandwidth allocation assigned thereto, perform:

selecting a plurality of tiles on a frame-by-frame basis from a plurality of encoded bitstreams corresponding to a media input stream associated with the 360° video session, wherein each encoded bitstream comprises a tile-encoded bitrate representation of the media input stream, and wherein each encoded bitstream is provided with a separate video quality; and,

combining the selected plurality of tiles for each frame and multiplexing the frames to form a single output stream having a mixed video quality for transmission to a corresponding client device, wherein the selected plurality of tiles of a frame comprise tiles have different video qualities.

10. The system as recited in claim 9, wherein the program instructions further comprise instructions for optimizing bitrates of at least a portion of the selected plurality of tiles for a 360° video session based on gaze vector information received from the client device associated with the video session.

11. The system as recited in claim 9, wherein the network portion supporting the plurality of 360° video sessions is configured as a logical network slice comprising at least a portion of a 5G communications network architecture, a fixed wireless network architecture, a Next Generation Network (NGN) architecture, a Digital Subscriber Line (DSL) network architecture, a Data Over Cable Service Interface Specification (DOCSIS)-compliant Cable Modem Termination System (CMTS) architecture, a Fiber Optic Services (FiOS) network architecture, a switched digital video (SDV) network architecture, and a Hybrid Fiber-Coaxial (HFC) network architecture.

12. The system as recited in claim 9, wherein the program instructions further comprise instructions configured to:

determine if a triggering event has occurred with respect to the shared bandwidth pipe of the network portion; and

responsive to the determining that a triggering event has occurred, adjust the bandwidth allocations of respective 360° video sessions to optimize usage of the bandwidth capacity dedicated to the network portion.

13. The system as recited in claim 12, wherein the triggering event comprises at least one of a new client device commencing a new 360° video session for receiving content via the shared bandwidth pipe, an existing client device terminating an ongoing 360° video session, and receiving a policy configuration change with respect to the bandwidth capacity dedicated to the bandwidth pipe.

14. The system as recited in claim 9, wherein the plurality of encoded bitstreams comprising respective tile-encoded bitrate representations are generated as a plurality of phase-encoded bitstreams based on at least one of High Efficiency Video Coding (HEVC) H.265 compression, Alliance for Open Media (AOMedia) Video 1 (AV1) compression and H.266/Versatile Video Coding (WC) compression.

15. The system as recited in claim 9, wherein a portion of the encoded bitstreams comprising respective tile-encoded bitrate representations are generated as block-intra-encoded bitstreams based on at least one of High Efficiency Video Coding (HEVC) H.265 compression, Alliance for Open Media (AOMedia) Video 1 (AV1) compression and H.266/Versatile Video Coding (VVC) compression.

16. The system as recited in claim 9, wherein the program instructions further comprise instructions configured to:

determine a surplus in individual bandwidth allocations assigned to respective 360 video sessions based on the corresponding bitrate caps;

obtain an overall excess amount of bandwidth available in the shared bandwidth pipe based on the surplus; and

distribute the overall excess amount of bandwidth to the plurality of 360° video sessions according to the respective weights associated with the plurality of 360° video sessions.

17. A non-transitory tangible computer readable medium having program instructions stored thereon which effectuate, when executed by one or more processors of a network element, a process for bandwidth optimization method in a network environment configured to support multiple 360° video sessions, the non-transitory tangible computer readable medium comprising:

a code portion for receiving a bandwidth capacity indication with respect to a bandwidth capacity dedicated to a network portion configured for supporting a plurality of 360° video sessions in a shared bandwidth pipe, each session serving a corresponding client device for playing an immersive video asset, wherein each video frame comprises an array of tiles projected on a 3-dimensional (3D) display environment viewed by a user operating the client device;

a code portion for receiving a plurality of weights corresponding to the plurality of 360° video sessions, each weight indicating a priority value associated with a respective video session;

a code portion for receiving a plurality of bitrate caps corresponding to the plurality of 360° video sessions, each bitrate cap indicating a bandwidth limit associated with a respective video session;

a code portion, operating responsive to the bandwidth capacity indication, total number of the plurality of 360° video sessions and corresponding pluralities of weights and bitrate caps, for determining a respective bandwidth allocation that can be assigned to each 360° video session without violating the bitrate cap of the 360° video session; and

a code portion operative, for each 360° video session, responsive to for providing the respective bandwidth allocation assigned thereto, for performing:

selecting a plurality of tiles on a frame-by-frame basis from a plurality of encoded bitstreams corresponding to a media input stream associated with the 360° video session, wherein each encoded bitstream comprises a tile-encoded bitrate representation of the media input stream, and wherein each encoded bitstream is provided with a separate video quality; and

combining the selected plurality of tiles for each frame and multiplexing the frames to form a single output stream having a mixed video quality for transmission to a corresponding client device, wherein the selected plurality of tiles of a frame comprise tiles have different video qualities.

18. The non-transitory tangible computer readable medium as recited in claim **17**, further comprising a code portion for optimizing bitrates of at least a portion of the selected plurality of tiles for a 360° video session based on gaze vector information received from the client device associated with the video session.

19. The non-transitory tangible computer readable medium as recited in claim **17**, wherein the network portion supporting the plurality of 360° video sessions is configured as a logical network slice comprising at least a portion of a 5G communications network architecture, a fixed wireless network architecture, a Next Generation Network (NGN) architecture, a Digital Subscriber Line (DSL) network architecture, a Data Over Cable Service Interface Specification (DOCSIS)-compliant Cable Modem Termination System (CMTS) architecture, a Fiber Optic Services (FiOS) network architecture, a switched digital video (SDV) network architecture, and a Hybrid Fiber-Coaxial (HFC) network architecture.

20. The non-transitory tangible computer readable medium as recited in claim **17**, further comprising:

a code portion for determining if a triggering event has occurred with respect to the shared bandwidth pipe of the network portion; and

a code portion, operating responsive to the determining that a triggering event has occurred, for adjusting the bandwidth allocations of respective 360° video sessions to optimize usage of the bandwidth capacity dedicated to the network portion.

21. The non-transitory tangible computer readable medium as recited in claim **20**, wherein the triggering event comprises at least one of a new client device commencing a new 360° video session for receiving content via the shared bandwidth pipe, an existing client device terminating or pausing an ongoing 360° video session, and a policy configuration change with respect to the bandwidth capacity dedicated to the bandwidth pipe.

22. The non-transitory tangible computer readable medium as recited in claim **17**, wherein the plurality of encoded bitstreams comprising respective tile-encoded bitrate representations are generated as a plurality of phase-encoded bitstreams based on at least one of High Efficiency Video Coding (HEVC) H.265 compression, Alliance for Open Media (AOMedia) Video 1 (AV1) compression and H.266/Versatile Video Coding (VVC) compression.

23. The non-transitory tangible computer readable medium as recited in claim **17**, wherein a portion of the encoded bitstreams comprising respective tile-encoded bitrate representations are generated as block-intra-encoded bitstreams based on at least one of High Efficiency Video Coding (HEVC) H.265 compression, Alliance for Open Media (AOMedia) Video 1 (AV1) compression and H.266/Versatile Video Coding (VVC) compression.

24. The non-transitory tangible computer readable medium as recited in claim **17**, further comprising:

a code portion for determining a surplus in individual bandwidth allocations assigned to respective 360° video sessions based on the corresponding bitrate caps;

a code portion for obtaining an overall excess amount of bandwidth available in the shared bandwidth pipe based on the surplus; and

a code portion for distributing the overall excess amount of bandwidth to the plurality of 360° video sessions according to the respective weights associated with the plurality of 360° video sessions.

* * * * *