(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 January 2018 (25.01.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/017599 A1

(51) International Patent Classification:
H04N 19/597 (2014.01) *H04N 19/154* (2014.01)

(21) International Application Number:
PCT/US2017/042646

(22) International Filing Date:
18 July 2017 (18.07.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

62/364,197	19 July 2016 (19.07.2016)	US
62/367,404	27 July 2016 (27.07.2016)	US
62/454,547	03 February 2017 (03.02.2017)	US
62/466,712	03 March 2017 (03.03.2017)	US

(71) Applicant: **VID SCALE, INC.** [US/US]; 200 Bellevue Parkway, Suite 300, Wilmington, Delaware 19809 (US).

(72) Inventors: **VISHWANATH, Bharath**; 775 Camino Del Sur H3, Goleta, California 93117 (US). **HE, Yuwen**; 13542 Silver Vine Path, San Diego, California 92130 (US). **YE, Yan**; 5001 Pearlman Way, San Diego, California 92130 (US). **XIU, Xiaoyu**; 9115 Judicial Drive, Suite #4109, San Diego, California 92122 (US).

(74) Agent: **IRVINE III, Robert J.**; Invention Mine LLC, 216 South Jefferson Street, Suite 102, Chicago, Illinois 60661 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,

(54) Title: QUALITY EVALUATION SYSTEM AND METHOD FOR 360-DEGREE VIDEO

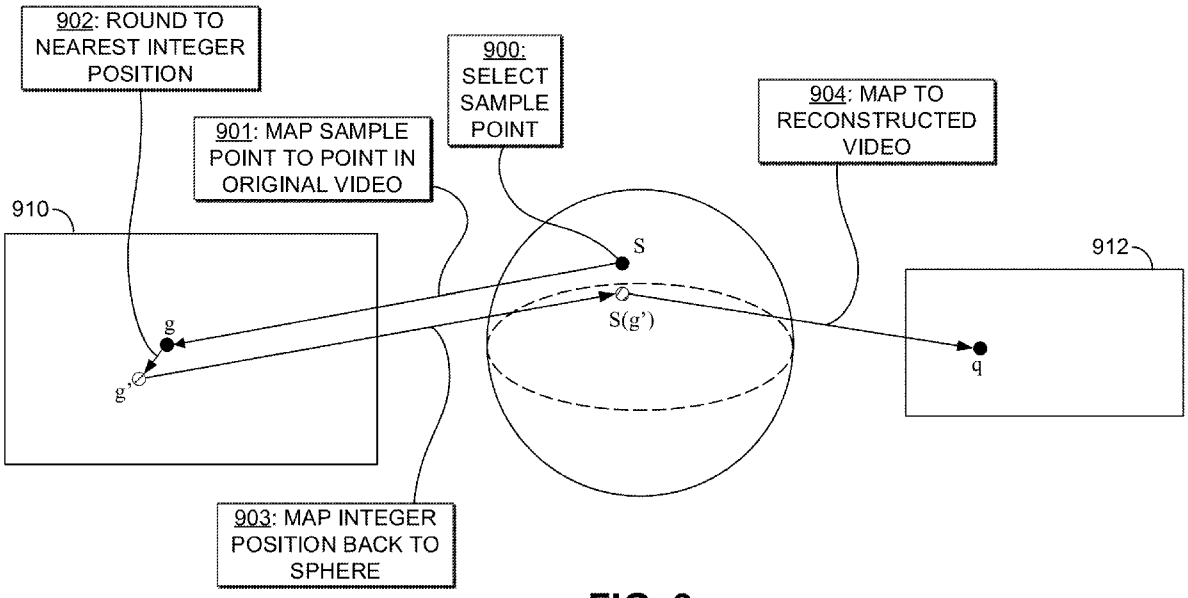


FIG. 9

(57) Abstract: Systems and methods are described herein for determining a distortion metric for encoding of spherical video. In spherical video, there is a mapping between a given geometry of samples and respective points on a unit sphere. In some embodiments, distortion is measured at each sample of interest, and the distortion of each sample is weighted by the area on the unit sphere associated with the sample. In some embodiments, a plurality of points on the unit sphere are selected, and the points are mapped to a nearest sample on the given geometry. Distortion is calculated at the nearest sample points and is weighted by a latitude-dependent weighting based on the latitude of the respective nearest sample point. The latitude-dependent weighting may be based on a viewing probability for that latitude.

WO 2018/017599 A1

[Continued on next page]

SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

QUALITY EVALUATION SYSTEM AND METHOD FOR 360-DEGREE VIDEO

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The present application is a non-provisional filing of, and claims benefit under 35 U.S.C. §119(e) from, the following four U.S. Provisional Patent Applications, all of which are entitled "Quality Evaluation System and Method for 360-Degree Video": Serial No. 62/364,197, filed July 19, 2016; Serial No. 62/367,404, filed July 27, 2016; Serial No. 62/454,547, filed February 3, 2017; and Serial No. 62/466,712, filed March 3, 2017. These four U.S. Provisional Patent Applications are incorporated herein by reference in their entirety.

BACKGROUND

[0002] Virtual reality (VR) is starting to come out of research labs and into our daily lives. VR has many application areas: healthcare, education, social networking, industry design/training, game, movies, shopping, entertainment, etc. It is capable of bringing immersive experience for the user and is thus known as immersive multimedia. It creates an artificial environment, and the user feels that he is present in the environment. A user's experience is further improved by sensory and other interactions like posture, gesture, eye gaze, voice, etc. To allow the user to interact with objects in the VR world in a natural way, a VR system also may provide haptic feedback to the user. To create the virtual environment, the user in some systems is presented with a 360-degree video, with a 360-degree viewing in horizontal direction and 180-degree viewing in vertical direction. At the same time, VR and 360-degree video are being considered to be the future direction for media consumption beyond Ultra High Definition (UHD) service. In order to improve the quality of 360-degree video in VR and to standardize the processing chain for client interoperability, an ad hoc group, belonging to MPEG-A (multimedia application format) Part-19, was set up in ISO/IEC/MPEG to work on the requirements and potential technologies for omnidirectional media application format in early 2016. Another ad hoc group, free view TV (FTV), recently issued exploration experiments for 360-degree 3D video applications. One major goal for FTV is to test the performance of two solutions: (1) 360-degree video (omnidirectional video) based systems, and (2) multi-view based systems. The joint video exploration team (JVET) from MPEG and ITU-T, which is exploring the new technologies for next generation video coding standard, issued a call for test sequences including VR.

[0003] The industry is working on improving the quality and user experience of various aspects in the VR processing chain, including capturing, processing, display and applications. On the capturing side, a VR system uses multiple cameras system to capture the scene from different

divergent views (e.g. 6~12 views). Those views are stitched together to form 360-degree video in high resolution (e.g. 4K or 8K). On the client or user side, the current virtual reality system usually consists of a computation platform, head-mounted display (HMD), and head tracking sensors. The computation platform is in charge of receiving and decoding 360-degree video, and generating the viewport for display. Two pictures, one for each eye, are rendered for the viewport. The two pictures are displayed in the HMD for stereo viewing. A lens is used to magnify the image displayed in the HMD for better viewing. The head tracking sensor constantly keeps track of the viewer's head orientation and feeds the orientation information to the system to display the viewport picture for that orientation. Some VR systems may provide a specialized touch device for viewer to interact with objects in the virtual world.

[0004] There are various existing VR systems available in the market. One system is the Rift provided by Oculus, and Gear VR, which is a product from Samsung and Oculus. The Rift is driven by a powerful workstation with good GPU support. Gear VR is a light VR system, which uses smartphone as computation platform, HMD display and head tracking sensor. A second system is the HTC Vive system. Rift and Vive have similar performance. The spatial HMD resolution is 2160x1200, refresh rate is 90Hz, and the field of view (FOV) is 110 degrees. The sampling rate for head tracking sensor is 1000Hz, which can capture very fast movement. Google also has a simple VR system called Cardboard. It consists of lenses and a cardboard frame. Like the Gear VR, it is driven by smartphone. In terms of 360-degree video streaming service, YouTube and Facebook are among the early providers.

[0005] The quality of experience such as interactivity and haptics feedback are still in need of further improvements in these current VR systems. For example, today's HMD is still too big and not convenient to wear. The current resolution of 2160x1200 for stereoscopic views provided by the HMDs is not sufficient, and could cause dizziness and discomfort for some users. Hence, further increase in resolution is necessary. Further, combining the feeling from vision in VR environment with force feedback in the real world is one direction to enhance VR experience. A VR roller coaster is an example application.

[0006] Many companies are working on 360-degree video compression and delivery system, and they have their own solutions. For example, Google YouTube provided a channel for 360-degree video streaming based on dynamic adaptive streaming for HTTP (DASH). Facebook also has solutions for 360-degree video delivery. One way to provide 360-degree video delivery is to represent the 360-degree information using a sphere geometry structure. For example, the synchronized multiple views captured by the multiple cameras are stitched on the sphere as one

integral structure. Then the sphere information is projected to 2D planar surface with a given geometry conversion process.

SUMMARY

[0007] Systems and methods are described herein for determining a distortion metric for encoding of spherical video. In spherical video, there is a mapping between a grid of samples and respective points on a unit sphere. In some embodiments, distortion is measured at each sample of interest, and the distortion of each sample is weighted by the area on the unit sphere associated with the sample. In some embodiments, a plurality of points on the unit sphere are selected, and the points are mapped to a nearest sample on the sample grid. Distortion is calculated at the nearest sample points and/or is weighted by a latitude-dependent weighting based on the latitude of the respective nearest sample point. The latitude-dependent weighting may be based on a viewing probability for that latitude.

[0008] In an exemplary embodiment, a method is provided for generating an area-weighted spherical peak signal-to-noise ratio (AW-SPSNR) for at least a selected portion of a coded spherical video, where the spherical video is associated with a mapping between regions on a unit sphere and samples in a grid. In the method, an area-weighted distortion is determined for each of a plurality of samples in the grid, wherein the area-weighted distortion is the unweighted distortion at the sample multiplied by the area of the region of the unit sphere associated with the respective sample. A sum of weighted distortions (SWD) is calculated by summing the determined area-weighted distortions. A sum of weights (SW) is calculated by summing the areas of the region of the unit sphere associated with the respective sample. A peak value P is determined from among the plurality of samples. The AW-SPSNR may be calculated as:

$$\text{AW-SPSNR}(c) = 10 \cdot \log(P^2 / (\text{SWD}/\text{SW})).$$

[0009] In some exemplary embodiments, a substantially spherical video is coded, with at least one coding-related decision being made using a rate-distortion metric determined based at least in part on the AW-SPSNR.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] A more detailed understanding may be had from the following description, presented by way of example in conjunction with the accompanying drawings.

[0011] FIGs. 1A-1B provide a schematic illustration of sphere geometry projection to a 2D plane using an equirectangular projection. FIG. 1A illustrates sphere sampling in longitude and latitude.

FIG. 1B illustrates a 2D plane with equirectangular projection. The point P on the sphere is projected to point q in the 2D plane.

[0012] FIG. 2 illustrates uneven vertical sampling in 3D space with equal latitude intervals.

[0013] FIG. 3 illustrates a sphere geometry representation with cubemap projection, PX (0), NX (1), PY (2), NY (3), PZ (4), NZ (5).

[0014] FIG. 4 is a schematic illustration of comparison of a ground truth signal with coded panorama videos as described in M. Yu, H. Lakshman, B. Girod, "A Framework to Evaluate Omnidirectional Video Coding Schemes", *IEEE International Symposium on Mixed and Augmented Reality,* 2015.

[0015] FIG. 5 is a schematic illustration of a sampling grid for 4:2:0 chroma format, where chroma is located at "D_x" positions and luma is located at "U_x" positions.

[0016] FIG. 6 is a flow chart illustrating intermediate and end-to-end quality evaluation in SPSNR.

[0017] FIG. 7 is a schematic illustration of a method for calculating SPSNR, particularly for use when the original video and the reconstructed video have the same projection format and resolution.

[0018] FIG. 8 is a schematic illustration of an alternative method for calculating SPSNR, particularly for use when the original video and the reconstructed video have different projection formats and/or resolutions.

[0019] FIG. 9 is a schematic illustration of an alternative method for calculating SPSNR, particularly when the original video and the reconstructed video have different projection formats and/or resolutions. Interpolation may be used in obtaining a sample value for the reconstructed video.

[0020] FIG. 10 is a functional block diagram depicting an exemplary video encoder.

[0021] FIG. 11 illustrates an exemplary wireless transmit/receive unit (WTRU) that may be employed as a video encoder and/or an apparatus for video quality evaluation in some embodiments.

[0022] FIG. 12 illustrates an exemplary network entity that may be employed as a video encoder and/or an apparatus for video quality evaluation in some embodiments.

DETAILED DESCRIPTION

[0023] A detailed description of illustrative embodiments will now be provided with reference to the various Figures. Although this description provides detailed examples of possible

implementations, it should be noted that the provided details are intended to be by way of example and in no way limit the scope of the application.

Equirectangular Projection (ERP).

[0024] FIG. 1A illustrates sphere sampling using longitudes (ϕ) and latitudes (θ). FIG. 1B illustrates the sphere being projected to a 2D plane using equirectangular projection. The longitude ϕ in the range $[-\pi, \pi]$ is known as yaw, and latitude θ in the range $[-\pi/2, \pi/2]$ is known as pitch in aviation, where π is the ratio of a circle's circumference to its diameter. For ease of explanation, the coordinates (x, y, z) are used to represent a point's location in 3D space, and the coordinates (ue, ve) are used to represent a point's location in a 2D plane. The equirectangular projection can be represented mathematically in Equations (1) and (2):

$$ue = (\phi/(2\times\pi)+0.5)\times W \quad (1)$$

$$ve = (0.5 - \theta/\pi)\times H \quad (2)$$

In Equations (1) and (2), W and H are the width and height of the 2D planar picture. As shown in FIGs. 1A-1B, the point P, the cross point between longitude L4 and latitude A1 on the sphere, is mapped to a unique point q in the 2D plane using the Equation (1) and (2). The point q in the 2D plane can be projected back to the point P on the sphere via inverse projection. The field of view (FOV) in FIG. 1B shows an example in which the FOV in the sphere is mapped to the 2D plane with the view angle along the X axis being about 110 degrees.

Equal-Area Projection.

[0025] Since the sampling density increases in ERP in areas close to the pole, equal-area projection (EAP) compensates for this increasing sampling density by decreasing the vertical sampling density in areas near the pole. In particular, the vertical sampling density is set to $\cos(\theta)$. In this way, the combined horizontal and vertical sampling density is a constant.

Cubemap.

[0026] In a cubemap projection, a sphere of unit diameter is circumscribed in a cube of unit side lengths. A rectilinear projection of spherical points onto the faces of cube with a field of view of 90° forms a cube map projection. FIG. 3 shows an example of a projection for a cubemap.

Limitations of Some Signal-to-Noise Metrics.

[0027] Spherical video may be presented as a panoramic video in different projection formats. The process of mapping sphere to the panoramic video leads to different sampling densities on the sphere. For example, in ERP, the sampling density approaches infinity toward the poles, and for

cubemap, the sampling density is greater at the corner of the faces than at the centers of the cube faces. This scenario is different from a normal 2D video. Thus, for spherical videos, there is no straightforward technique for evaluating the videos, since they are in different projection formats and they might be encoded at different resolutions. Also, different points on the sphere have different viewing probabilities. For example, the points at the equator are more likely to be viewed as compared to the points at the poles. It would be beneficial to provide a meaningful metric which can measure the quality of video without being biased by the sampling densities and other factors. In this direction, M. Yu et al. have proposed different metrics which include Spherical Peak Signal-to-Noise Ratio (SPSNR), Weighted Spherical PSNR (W-SPSNR) and Latitude Spherical PSNR (L-SPSNR) in M. Yu, H. Lakshman, B. Girod, "A Framework to Evaluate Omnidirectional Video Coding Schemes," *IEEE International Symposium on Mixed and Augmented Reality,* 2015, which is incorporated herein by reference in its entirety. These may be described as follows.

[0028] *Spherical PSNR (SPSNR).* In SPSNR, uniformly sampled set of points on a unit sphere are mapped onto two videos to be compared, respectively. Error at these mapped coordinates on the panoramic videos is computed. One benefit of this metric is that panoramic videos in any projection formats can be compared in a fair manner. The metric overcomes the problem of different sampling densities in different projection formats. FIG. 4 depicts this concept. Here, a point **s** from a uniformly sampled set of points on unit sphere is mapped on to panorama videos, which are shown as q and r in the decoded videos Pano1 and Pano2 in FIG. 4, respectively. Pano1 is one projection format such as equal-area, and Pano2 is another projection format such as cubemap. These signals are compared against the ground truth sample g. To measure the distortion of Pano1, the difference between g and q is calculated. And to measure the distortion of Pano2, the difference between g and r is calculated. For example, to measure the loss in geometry conversion, going from ERP to cubemap and back to ERP, the original ERP will be the ground truth and the reconstructed ERP will be equivalent to the decoded video in the FIG. 4.

[0029] *Weighted Spherical PSNR (W-SPSNR).* To calculate W-SPSNR, start with the same set of points as S-PSNR and map it onto the panoramic video. However in calculating the total error, weigh the error component of a sample according to the probability of a user viewing that sample.

[0030] *Latitude Spherical PSNR (L-SPSNR).* In L-SPSNR, the 3D map of weights used in W-SPSNR is marginalized along the longitudes so as to get the weights only along the latitude. These latitude weights are used to weigh the errors based on their latitude positions.

[0031] PSNR can be used as one objective quality metric for 2D planar picture quality evaluation. However, many projections of 360-degree video do not have the property of even sampling because of non-planar geometrical structure, such as sphere, cubemap, cylinder and pyramid.

Directly applying PSNR for 360-degree video quality evaluation is biased and does not provide a good representation of subjective viewing quality. SPSNR and L-SPSNR were proposed to solve the uneven sampling problem for 360-degree video. However, when SPSNR or L-SPSNR are used for quality evaluation, the following four problems can arise.

[0032] First, the sampling density in ERP is uneven. In particular, the sampling density approaches infinity in regions approaching the pole. The top portion of a picture corresponding to the North Pole and the bottom portion corresponding to the South Pole are "stretched," which indicates that the equirectangular sampling in the 2D spatial domain is uneven. In SPSNR, by using uniformly sampled points on a unit sphere for comparing 360 videos in ERP, many samples are neglected in the comparison. In particular, a lesser number of samples are considered for comparison towards the pole. It would thus be beneficial to come up with a new metric that takes all the samples into account and gives a meaningful PSNR.

[0033] In ERP, different samples cover differently-sized solid angles on a sphere. The solid angle is proportional to the area on the sphere associated with that sample. In exemplary embodiments, all the available samples can be taken, and each sample error can be weighed based on the solid angle spanned by the sample. This metric may be referred to as "Area Weighted Spherical PSNR". Area Weighted Spherical PSNR, as a metric, thus considers all the samples available at hand.

[0034] Second, in mapping a point in the uniformly sampled set of points on unit sphere to a projected 360 video, it is possible to land in a point which might not be present on the sampling grid. While it is possible to use interpolation filters to estimate the value of corresponding mapped point, the error measured by using interpolation methods apart from nearest neighbor will thus introduce an additional component of interpolation error. This becomes severe if this metric is used to compare the quality of 360-degree videos in various projections. In presenting a metric, it is beneficial for the metric to reflect the difference between the actual samples rather than the interpolated samples.

[0035] Third, In L-SPSNR, when nearest neighbor is used for getting the sample on the sampling grid, the error is weighted by the latitude weight of the new interpolated sample point in the sampling grid. However, the latitude weight may not correctly match the position of the interpolated nearest neighbor sample. A point in the uniformly sampled set of points on unit sphere is mapped to a projected 360 video. The sample value is derived with nearest neighbor method for interpolation. And the weight is derived with the latitude weight of the position of the mapped point, instead of the position of the point on the sampling grid that is used to derive the sample value directly.

[0036] Fourth, prior error metrics do not satisfactorily account for calculation of the error of the chroma components. For the case of chroma 4:4:4, a similar approach as luma component can be used for chroma components. However, for the case of subsampled chroma components, the sampling grid of chroma and luma might be different. For example, for the 4:2:0 chroma format, a typical sampling grid relationship between luma and chroma is shown in FIG. 5, where chroma is located in the grid marked as "D_x", and luma is located in the grid marked as "U_x". As shown, the chroma sampling grid is aligned with the luma sampling grid in the horizontal direction, but has a 0.5-sample offset in the vertical direction. Proper care has to be taken in deriving the latitude weights for the chroma components by accounting for these offsets between luma and chroma components.

Overview of Exemplary Embodiments.

[0037] In this disclosure, exemplary quality evaluation methods for 360-degree video are disclosed. Disclosed herein are a new error metric calculation methods. The present disclosure further provides a view-port based weighted average PSNR. In some embodiments, overlapped view-ports are predefined to cover the sphere. For each view-port at certain viewing angle, the view-port is a 2D planar picture projected from sphere. A PSNR may then be calculated based on the view-port and its reference generated with the reference sphere. Subsequently, all PSNR values of view-ports are averaged with the weight determined based on viewing probability, for example, the weight is larger for those view-ports close to the equator. In some embodiments, a weighting as described in M. Yu et al. is used.

[0038] Disclosed herein are new metrics for quality measurement. Techniques are further disclosed for addressing the problems in quality evaluation metrics mentioned in the previous section. Conventional metrics such as PSNR do not consider the uneven sampling issue in projection space such as ERP. Embodiments using the metrics disclosed herein take into consideration the issues of uneven sampling.

[0039] In some embodiments, metrics disclosed herein are used in rate-distortion optimization when an ERP picture is coded. In the existing rate-distortion optimization of encoders using H.264 and/or HEVC, the distortion is evaluated with the sum of square error (SSE), which directly related to PSNR. In cases where the picture is coded in ERP format, the distortion calculation may be replaced by the weighted distortion calculation proposed in exemplary embodiments, which address the sampling issue in ERP. Specifically, given a coding unit with size NxM, the distortion of the block can be calculated as:

$$\text{Dist(CU)} = k \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (w_{n,m} \cdot D_{n,m}) \; ; \; n, m \in CU$$

$$k = \frac{1}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} w_{n,m}} ; \; n, m \in CU$$

where $w_{n,m}$ is the weight for the sample, $D_{n,m}$ is the distortion of the sample, and k is the normalization factor. The normalized factor k may also be evaluated globally to reduce the computation complexity.

$$k = \frac{1}{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} w_{i,j}} ; \; i, j \in picture \; (W \cdot H)$$

In exemplary embodiments, the modified rate-distortion optimization may be applied in the determination of various coding parameters (e.g. motion vector, mode, quantization parameter, tree structure) for use in processes such as motion estimation, inter or intra mode decision, quantization, and tree structure (quadtree, binary-tree) splitting decision.

Area Weighted Spherical PSNR (AW-SPSNR).

[0040] Exemplary embodiments disclosed herein make use of a metric referred to herein as Area Weighted Spherical PSNR (AW-SPSNR) to compare panoramic videos for 360-degree video quality evaluation. As mentioned above, Spherical PSNR compares two videos based on only on a subset of the available set of samples. However, in the case of ERP, embodiments disclosed herein operate to consider the entire set of samples available in the picture and weigh the errors according to the solid angle covered by the samples. Let (ue, ve) be the coordinate location of the pixel on ERP. Its corresponding latitude and longitude position can be obtained using the following formulas.

$$\phi = (\text{ue}/W - 0.5) \cdot 2\pi \tag{3}$$

$$\theta = (0.5 - \text{ve}/H) \cdot \pi \tag{4}$$

W and H are the width and height of ERP picture, respectively. For this sample at (ϕ, θ), the area covered by the sample on the unit sphere given delta ϕ and delta θ is calculated as follows.

$$solid_angle(\phi, \theta) = cos(\theta) \cdot |d\theta| \cdot |d\phi| \tag{5}$$

The area is dependent only on the latitude because (ϕ, θ) is evenly sampled. Thus for all the samples at a given latitude θ, the error may be weighted by cos(θ).

[0041] For samples of infinitesimal size, the sum of all these weights would equal to the total surface area of the unit sphere, which is 4π, and weights could be normalized using division by 4π. The sum of weights for samples of finite size, however, is not exactly 4π. Thus, to normalize the weights in some embodiments, the weights of each sample are summed, and the weights are normalized with the resulting actual sum.

[0042] In some embodiments disclosed herein, the distortion of samples on the sphere is measured by considering even sampling. Suppose (uei, vei) is the i-th sample position on a ERP picture; W, H are the width and height of the ERP picture, respectively; Ref(c, x, y) and I(c, x, y) are the reference picture and the picture to be evaluated, respectively, of component c, where c may be luma, Cb or Cr; SWD is the sum of weighted distortion; and SW is the sum of weight.

[0043] In an exemplary method, the following steps are performed. The distortion (squared error) Di is calculated for the i-th sample point:

$$Di = [Ref(c, uei, vei) - I(c, uei, vei)]^2$$

The latitude of the position of the i-th sample point is calculated:

$$\theta i = (0.5 - vei/H) \cdot \pi$$

The distortion is weighted with the obtained latitude weight. The sum of weighted distortion (SWD) and the sum of weights (SW) are incremented by the weighted distortion (wDi) and by the current weight $\cos(\theta i)$, respectively.

$$wDi = \cos(\theta i) \cdot Di$$

$$SWD = SWD + wDi$$

$$SW = SW + \cos(\theta i)$$

After all points are evaluated, the AW-SPSNR of component c may be calculated as follows, where P is the peak value of the sample value.

$$AW\text{-}SPSNR(c) = 10 \cdot \log(P^2 / (SWD/SW))$$

[0044] It may be noted that the value of AW-SPSNR is represented in decibels in the equation above. It should be understood that in other embodiments, AW-SPSNR is represented using a measure other than decibels.

[0045] AW-SPNSR may be calculated over any desired subset of samples in the projection picture. For example, it may be calculated over the full set of samples in an encoded/decoded ERP representation (to evaluate overall distortion of the encoded representation), or it may be calculated using only the set of samples in a particular coding unit (CU) or prediction unit (PU), or some other subset of samples relevant to a particular coding decision (e.g., to support the making of rate-distortion optimized decisions within a video encoder).

[0046] For other projections, the area calculation in Equation (5) is changed according to projection format, so the weight may be different from that in ERP.

[0047] Set forth below is one embodiment of a general procedure to calculate the weights for any general geometry, and exemplary cases of evaluating the weights for EAP and cubemap.

AW-SPSNR Weights for a General Projection Format.

[0048] Embodiments disclosed herein accommodate different weight calculations for different projection formats. Exemplary embodiments proceed to determine weights as follows. Consider a mapping in which (xg, yg) are the coordinates of a point in a given geometry space and (θ, ϕ) are the corresponding latitude and longitude position of this sample on the unit sphere. The geometry mapping may be expressed as functions *f* and *g*, where *f* and *g* are different for different types of projection. The functions *f* and *g* satisfy the following relationship.

$$\theta = f(xg, yg) \tag{6}$$

$$\phi = g(xg, yg) \tag{7}$$

[0049] In order to compute the area on the unit sphere, dθ and dϕ are computed. Since θ and ϕ are functions of both xg and yg, the partial derivatives are first computed and then the total derivatives are computed. Let ∂θ/∂xg and ∂θ/∂yg be the partial derivatives of θ with regard to xg and yg, respectively. Similarly, let ∂ϕ/∂xg and ∂ϕ/∂yg be the partial derivatives of ϕ with regard to xg and yg, respectively. The computation of dθ and dϕ may then be as follows:

$$d\theta = \left(\frac{\partial\theta}{\partial xg}\right)dxg + \left(\frac{\partial\theta}{\partial yg}\right)dyg \tag{8}$$

$$d\phi = \left(\frac{\partial\phi}{\partial xg}\right)dxg + \left(\frac{\partial\phi}{\partial yg}\right)dyg \tag{9}$$

[0050] To determine the area spanned by the sample, the values of |dϕ| and |dθ| may be determined as follows:

$$|d\theta| = \left(\left(\left(\frac{\partial\theta}{\partial xg}\right)^2 + \left(\frac{\partial\theta}{\partial yg}\right)^2\right)^{\frac{1}{2}}\right)dx \tag{10}$$

$$|d\phi| = \left(\left(\left(\frac{\partial\phi}{\partial xg}\right)^2 + \left(\frac{\partial\phi}{\partial yg}\right)^2\right)^{\frac{1}{2}}\right)dx \tag{11}$$

[0051] The area may then be computed using equation (5).

AW-SPSNR Weights for EAP.

[0052] In some embodiments, AW-SPSNR is determined for systems using equal-area projection. Let (ue, ve) be the pixel coordinates on the EAP frame. The geometry mapping between the pixel coordinates and the position on the unit sphere is as follows:

$$\phi = (\frac{ue}{W} - 0.5) \times 2\pi \quad (12)$$

$$\theta = \sin^{-1}(2 \times (0.5 - \frac{ve}{H})) \quad (13)$$

[0053] In order to compute the solid angle, the derivatives of ϕ and θ are used specifically:

$$d\phi = (\frac{1}{W}) \times 2\pi \times due \quad (14)$$

$$d\theta = (\frac{2}{H \times cos\theta}) \times dve \quad (15)$$

[0054] Correspondingly the solid angle or equivalently the area spanned by the sample at (ue, ve) is:

$$Area = (due \times dve \times \pi)/(W \times H) \quad (16)$$

[0055] As seen, ue and ve are equally sampled. Thus, the area is the same for all samples.

AW-SPSNR Weights for Cubemap.

[0056] Cubemap has six symmetric faces. The weights derived for one face can be used for all the six faces. The calculation is done for the face ABCD in FIG. 3. Let x and y be the normalized coordinates of the point Ps on the face ABCD, where x and y range from -1 to +1. The relation between x,y and ϕ, θ is as follows:

$$\phi = \tan^{-1}(x) \quad (17)$$

$$\theta = \tan^{-1}(y \times cos(\tan^{-1}(x))) \quad (18)$$

[0057] Since ϕ is a function of x alone, its derivative can be computed as:

$$d\phi = \cos^2(\phi) dx \quad (19)$$

[0058] θ is a function of both x and y. Next the partial derivatives ∂θ/∂x and ∂θ/∂y are computed and then the total derivative. The partial derivatives are:

$$\frac{\partial\theta}{\partial x} = -sin(\theta)cos(\theta)sin(\phi)cos(\phi) \quad (20)$$

$$\frac{\partial\theta}{\partial y} = \cos^2(\theta)\, cos(\phi) \quad (21)$$

[0059] The total derivative can then be computed as:

$$d\theta = \left(\frac{\partial\theta}{\partial x}\right)dx + \left(\frac{\partial\theta}{\partial y}\right)dy \quad (22)$$

[0060] However, still to be calculated is the length of dθ. The Euclidian norm may be calculated assuming dx and dy to be equal as:

$$|d\theta| = \left(\left(\left(\frac{\partial\theta}{\partial x}\right)^2 + \left(\frac{\partial\theta}{\partial y}\right)^2\right)^{\frac{1}{2}}\right)dx \quad (23)$$

[0061] The area of the sample may be determined as follows in Eq. (24), and this area may be used as the weight of the sample in embodiments disclosed herein:

$$Area = cos(\theta) \times cos^2(\phi) \times \left(\left(\left(sin(\theta)cos(\theta)sin(\phi)cos(\phi)\right)^2 + \left(cos^2(\theta)\, cos(\phi)\right)^2\right)^{\frac{1}{2}}\right)dxdy \quad (24)$$

Interpolation Method for Quality Comparison.

[0062] In the implementation of M. Yu et al., *supra*, samples from the set of uniformly sampled set of points on the unit sphere are mapped onto the projection geometry. If the mapped point is not on the integer sampling grid, various interpolation filters, including bilinear and bicubic filters, are used to obtain the pixel value at the mapped positions. The ultimate goal in quality comparison should be the comparison of the samples on the sampling grid. Thus, it is desirable to use nearest neighbor interpolation in quality comparison.

Alternative Implementations of SPSNR.

[0063] SPSNR may be used for 360 video quality evaluation in different ways. In JVET 360 common test conditions and evaluation procedures, SPSNR is applied in two ways: intermediate and end to end quality evaluation, as shown in FIG. 6. For end to end quality evaluation, the reference for SPSNR calculation is the original video, and the test is reconstructed video after decoding and projection format conversion. Thus, the two inputs to SPSNR calculation, that is, the reference signal and the test signal, have the same projection format and resolution. For intermediate quality evaluation, the reference is still the original high resolution video, but the test is the reconstructed video right after decoding but before projection format conversion. Since the inverse projection format conversion has not yet been applied, the test signal may have a different projection format and/or resolution compared to that of reference video. Thus, the intermediate quality evaluation is also referred to as “cross format” quality evaluation.

[0064] For SPSNR calculation as shown in FIG. 4, the point S from the set of points uniformly sampled on the sphere is mapped to the point g in the original video and the point q in the reconstructed video Pano1. The error between the sample value at point g and the sample value at point q is calculated for the quality evaluation between original video and reconstructed video. The sample value at point g or q may be derived with the sample at its nearest integer sampling point if the point g or q is not at an integer sampling position to avoid introducing additional interpolation error in quality evaluation. Suppose g' and q' are the nearest integer sampling point of g and q, respectively. The sample error calculation process is summarized as following steps, illustrated in FIG. 7.

[0065] In step 700, a sample point is S is selected from a set of points that is substantially uniformly sampled on a sphere.

[0066] In step 701, map a point S (S is from the set of points uniformly sampled on a sphere) to point g in the original video 710 in the projection plane.

[0067] In step 702, round point g to nearest neighbor point g' at an integer sampling position.

[0068] In step 703, map point S from the set of points uniformly sampled on the sphere to point q in the reconstructed video 712 in the projection plane.

[0069] In step 704, round point q to nearest neighbor point q' at an integer sampling position; and subsequently calculate the error between the sample value at point g' and the sample value at point q'.

[0070] In FIG. 7, as shown by the dashed lines, coordinate mapping from the sphere to the projection plane can be performed in the reverse direction, that is, the points g' and q' on the projection plane can be mapped back to S(g') and S(q') on the sphere. If the original video and the reconstructed video have the same projection format and the same resolution, S(g') and S(q') will be mapped back at the same position on sphere (as indicated by S' in FIG. 7). However, if the original video and the reconstructed video have either different projection formats or different resolutions or both different resolution and different projection formats, after the inverse coordinate mapping, S(g') and S(q') may correspond to different positions on the sphere (not shown in FIG. 7). This makes the quality evaluation inaccurate, because the sample errors will effectively be calculated between the positions g' and q', which do not correspond to the same positions on the sphere. In order to reduce this error due to non-aligned spherical coordinates, exemplary sample error calculation methods for SPSNR with nearest neighbor are proposed as follows. Such embodiments may be used when, for example, the original video and the reconstructed video are not in the same projection format or do not have the same resolution.

[0071] An exemplary error calculation method is illustrated in FIG. 8. A method as illustrated in FIG. 8 may operate to minimize the distance of the two points between which the sample error is calculated, in order to reduce the inaccuracy caused by the non-aligned spherical coordinates due to nearest neighbor rounding in the projection plane. As illustrated in FIG. 8, an exemplary SPSNR method may be performed as follows.

[0072] In step 800, a sample point S is selected from a set of points that is substantially uniformly sampled on a sphere.

[0073] In step 801, map point S to point q in the reconstructed video 812.

[0074] In step 802, round point q to nearest neighbor point q' at an integer sampling position in the reconstructed video domain.

[0075] In step 803, perform inverse coordinate mapping of the point q' back onto the sphere at S(q').

[0076] In step 804, perform coordinate mapping from the spherical coordinate S(q') to the original video projection domain 810 at the position g.

[0077] In step 805, round point g to nearest neighbor point g' at an integer sampling position, and subsequently calculate the error between the sample value at point g' in the original video and the sample value at point q' in the reconstructed video.

[0078] As in other embodiments described herein, such a method may be performed for each of a plurality of points S selected on the sphere, with squared errors for each point being summed to generate a distortion metric. In some embodiments, the calculated errors are weighted in the summation process, using techniques disclosed herein or other weighting schemes.

[0079] In some embodiments, a method of error calculation may be performed as follows, in which the roles of the original video and reconstructed video are reversed as compared to the above steps. A sample point S is selected from a set of points that is substantially uniformly sampled on a sphere (analogous to step 800). Point S is mapped to point q in the original video (analogous to step 801, except that video 812 now represents the original video). Point q is rounded to nearest neighbor point q' at an integer sampling position in the original video domain (analogous to step 802, except in the original video). Inverse coordinate mapping of the point q' back onto the sphere at S(q') is performed (analogous to step 803). Coordinate mapping is performed from the spherical coordinate S(q') to the reconstructed video projection domain at the position g (analogous to step 804, except that video 810 now represents the reconstructed video). Point g is rounded to the nearest neighbor point g' at an integer sampling position (analogous to step 805), and the error is

calculated between the sample value at point g' in the reconstructed video and the sample value at point q' in the original video.

[0080] Using a method as shown in FIG. 8, the sample error is calculated between the points q' and g'. The distance between spherical point S(q') and spherical point S(g') on sphere is measured with the distance between g (which maps to the coordinate q' in the reconstructed video projection domain) and g' in the original/reconstructed video projection domain. Examples of coordinate points such as may be used in the method of FIG. 7 are also shown in FIG. 8 as g_0 (directly mapped from the spherical coordinate S) and g_0' (rounded coordinate based on g_0). The method of FIG. 7 effectively measures the sample error between the points q' and g_0'. The distance between spherical point S(q') and spherical point S(g_0') on sphere is measured with the distance between g (which maps to the coordinate q' in the reconstructed video projection domain) and g_0' in the original/reconstructed video projection domain. In contrast, because by definition g' is the nearest neighbor of point g, a method according to FIG. 8 can operate to minimize the distance between the two points between which the sample error is calculated.

[0081] In the method described with respect to FIG. 8, SPSNR may be calculated without necessarily considering the resolutions of the original and reconstructed video in their projection domain(s). In some embodiments, however, effects due to the original video and the reconstructed video having different resolutions are taken into consideration. In some such embodiments, the original and the reconstructed video may have the same projection format. In an exemplary method for use when the videos have the same projection format, the video having the lower resolution is selected (from between the original video and the reconstructed video). A point S on the sphere is mapped to point q in the selected lower-resolution video, as depicted in step 801 of FIG. 8. The rounding of coordinate g to g' in step 805 is thus performed in the higher-resolution video. Performing the final rounding of the coordinate g in the higher resolution video incurs a smaller rounding error, since the sampling grid in the higher resolution video is denser and more accurate.

[0082] In some embodiments, a method as described with respect to FIG. 8 may be applied in a case where the original and the reconstructed video also have different projection formats. In one such embodiment, step 801 is performed on the video (original or reconstructed) with the lower resolution, and the rounding of step 805 is performed in the video with higher resolution. When projection formats are different, higher resolution may not always translate to denser sampling in all areas on the sphere. However, from an overall perspective, higher resolution generally represents a denser and more accurate sampling grid, and may be better suited for final rounding in step 805.

[0083] Compared to other techniques, the methods described with respect to FIG. 8 perform additional steps of 2D-to-3D and 3D-to-2D coordinate mapping. This may lead to increased computational complexity. However, given that the projection formats and resolutions of the original video and reconstructed video are fixed, and the set of points on the sphere (S) is also fixed for SPSNR calculation, the coordinate mapping can be pre-calculated and stored in a lookup table, and re-used on a frame-by-frame basis to reduce computation complexity.

[0084] In some embodiments, methods as described with respect to FIG. 8 may be employed in situations where the original video and reconstructed video have either different projection format or different resolution. Such methods may further be implemented in cases where the luma and chroma components have different resolutions, such as 4:2:0 chroma format. In such instances, a method as described with respect to FIG. 8 may be applied for each component's quality evaluation separately. For example, a method may be applied with luma component resolution and sampling grid to perform luma quality evaluation, and the method may be separately applied with chroma component resolution and sampling grid to perform chroma quality evaluation.

[0085] In exemplary methods described above, the sphere point S(g') corresponding to g' in the original video and the sphere point S(q') corresponding to q' in the reconstructed video may not be located at the same position on the sphere. This is the result of the rounding operation in step 805. In particular, the point S(q') on the sphere is likely to be different from S(g') when the projection format of reconstructed video and/or the resolution of the reconstruction video is different from that of the original video. This results in the SPSNR not being measured using the same set of sphere points for those reconstructed videos having different projection formats and/or different resolutions. In embodiments described below, the position misalignment on the sphere between reference sample and reconstructed sample is addressed using interpolation in the reconstructed video. One such embodiment is illustrated in FIG. 9 and includes the following steps.

[0086] In step 900, a sample point S is selected from a set of points that is substantially uniformly sampled on a sphere.

[0087] In step 901, map point S (where S is from the set of points uniformly sampled on sphere) to point g in the original video 910.

[0088] In step 902, round point g to nearest neighbor point g' at an integer sampling position in the original video domain.

[0089] In step 903, perform inverse coordinate mapping of the point g' back onto the sphere at S(g').

[0090] In step 904, perform coordinate mapping from the spherical coordinate S(g') to the reconstructed video projection domain 912 at the position q. Apply interpolation to derive the sample value at point q using its neighboring sample values at integer sampling positions if q is not located at an integer sampling position. No interpolation is applied if q is at an integer sampling position. The error between the sample value at point g' in the original video and the interpolated sample value at point q in the reconstructed video is then calculated.

[0091] In a method such as the one described above, the use of interpolation instead of rounding avoids potential issues with having sphere point S(g') and sphere point S(q') being different. In this way, distortion is measured between the sphere points S(g') and S(q), which are aligned on the sphere. As a result, SPSNR calculations performed using a method such as that of FIG. 9 refer to the same set of points on the sphere regardless of the projection formats and resolution of the reconstructed video. When the reconstructed video has the same projection format and resolution as original video, a method as in FIG. 9 will achieve the same results as the methods of FIGs. 7 and 8.

[0092] An SPSNR calculation using a technique described with respect to FIG. 4 applies the interpolation filtering to both the original signal and reconstructed video. The interpolation filter may change the characteristics of original signal and thus will affect the PSNR calculation. The method described with respect to FIG. 7 does not apply interpolation filtering, and it can be used when the original video and reconstructed have the same projection format and resolution. The sample position misalignment on the sphere between the reference sample and the reconstructed sample degrades the SPSNR calculation when the original video and reconstructed video do not have the same projection format and resolution. The misalignment also makes SPSNR comparison unreliable when comparing different projection formats or the same projection format with different resolution. The method described with respect to FIG. 8 relieves the problem of sample position misalignment; however, it does not completely resolve the problem because of rounding operation. The method described with respect to FIG. 9 addresses the problem of sample position misalignment by applying the interpolation filtering to the reconstructed video as appropriate. The method of FIG. 9 does not apply the interpolation filtering to the original signal, so it will not change the characteristics of the original signal for PSNR calculation. As a result, a method such as that illustrated with respect to FIG. 9 may be used in calculating an SPSNR that is appropriate for cross-format quality comparison.

Latitude Weight for Nearest Neighbor Interpolation.

[0093] In the implementation in M. Yu et al., *supra*, for the calculation of L-SPSNR, a sample from the uniformly sampled set of points on sphere is mapped onto the projection geometry, and

various interpolation filters are used to obtain the sample at this mapped location. When nearest neighbor interpolation is used, the interpolated sample is at a different location than the position of the original sample point mapped onto the projection geometry. In deriving the latitude weight, the latitude of the original sample point on the sphere is used. As discussed above, this results in a mismatch between the distortion, which is measured as squared error between the reference sample and the sample to be evaluated, and the associated weight.

[0094] In embodiments disclosed herein, the following method may be used in weighted distortion computation for L-SPSNR calculation for luma component or chroma components in 4:4:4 chroma format.

[0095] A set of substantially uniformly distributed points is selected on the unit sphere. The number of points selected may vary in different embodiments. Any one of a variety of techniques may be used to select substantially uniform points on the unit sphere, including random selection or selection using algorithmic techniques such as spiral point selection or charged particle simulation. Suppose (ui, vi) is i-th sample point position of the uniformly sampled set of points on the unit sphere; (uei, vei) is a mapped position on an ERP picture; W, H are the width and height of the ERP picture, respectively; Ref(c, x, y) and I(c, x, y) are the reference picture and the picture to be evaluated, respectively, of component c, where c may be, e.g., luma, Cb or Cr; $W(\theta)$ is the weight function to define the weight at the latitude θ. SWD is the sum of weighted distortion, and SW is the sum of weights.

[0096] For the i-th sample point from the uniformly sampled set of points on the unit sphere, map the sample point onto an ERP picture as follows:

$$uei = W \cdot (ui/2\pi + 0.5)$$

$$vei = H \cdot (0.5 - vi/\pi)$$

The nearest neighbor (uni, vni) of the mapped point is found on the sampling grid:

$$uni = NN(uei)$$

$$vni = NN(vei)$$

The distortion (squared error) Di is calculated at this nearest neighbor sample point:

$$Di = [Ref(c, uni, vni) - I(c, uni, vni)]^2$$

The latitude of the position of the nearest neighbor sample point is found:

$$\theta i = (0.5 - vni/H) \cdot \pi$$

The latitude weight of the new latitude is obtained. In embodiments in which $W(\theta)$ is defined by a look-up table, interpolation may be applied to derive the weight at the input latitude θ.

$$wi = W(\theta i)$$

The distortion is weighted with the obtained latitude weight, and the weighted distortion and the weight are accumulated.

$$wDi = wi * Di$$

$$SWD = SWD + wDi$$

$$SW = SW + wi$$

After all points from the uniformly sampled set of points on the unit sphere are evaluated, the L-SPSNR of component c is calculated as follows. P is the peak value of the sample value.

$$L\text{-}SPSNR(c) = 10 \cdot \log(P^2 / (SWD/SW))$$

In an embodiment using this method, the distortion and the weight are aligned at the same position on the sphere. As a result, the calculated value of L-SPNSR may more accurately reflect video quality.

[0097] It may be noted that the value of L-SPSNR is represented in decibels in the equation above. It should be understood that in other embodiments, L-SPSNR is represented using a measure other than decibels.

PSNR for Chroma Components.

[0098] In the case of videos with sub-sampled chroma components, additional care should be taken in deriving latitude weight in calculating L-SPSNR. Exemplary embodiments are disclosed herein with reference to 4:2:0 subsampling since it is widely used, although it is contemplated that the techniques described herein may also be applied to alternative chroma subsampling schemes. Initially, an overview of the 4:2:0 chroma sub-sampling grids is provided below, followed by a workflow for calculating L-SPSNR for chroma components. For 4:2:0 chroma format, there are four kinds of chroma sampling grid placement relative to luma grid for progressive video format. These are defined in HEVC and H.264/AVC as chroma sample location types. The chroma sample location type is defined in the following table.

Table 1. Chroma sample location type definition

Chroma sample location type	Vertical phase shift relative to luma sample location	Horizontal phase shift relative to luma sample location
0	0.5	0
1	0.5	0.5
2	0	0
3	0	0.5

[0099] Type 0 is most widely used for 4:2:0 chroma format. It has a misalignment of 0.5 samples in the vertical direction.

[0100] Exemplary steps that may be used in calculating L-SPSNR for chroma components include the following. A point i is selected from the uniformly sampled set of points on the unit sphere and is mapped onto the ERP picture as follows.

$$uei = (W/2)\cdot(ui/2\pi+0.5)$$

$$vei = (H/2)\cdot(0.5-vi/\pi)$$

The nearest neighbor position of the mapped point is found on the chroma sampling grid.

$$uni = NN(uei)$$

$$vni = NN(vei)$$

The distortion (squared error) is calculated at this nearest neighbor sample point for component c (either Cb or Cr);

$$Di = [Ref(c, uni, vni) - I(c, uni, vni)]^2$$

The position of the nearest neighbor is found on the luma grid.

$$uni_l = (NN(uni\cdot 2)>>1) <<1$$

$$vni_l = (NN(vni\cdot 2)>>1) <<1$$

The horizontal and vertical offsets listed in Table 1 are added based on the sampling pattern; offset_x and offset_y are the horizontal and vertical offset, respectively.

$$uni_c = uni_l + offset_x$$

$$vni_c = vni_l + offset_y$$

After adding offsets, the latitude of the new location is found.

$$\theta i = (0.5- vni_c/H)\cdot\pi$$

The latitude weight of the new latitude is found.

$$wi = W(\theta i)$$

The distortion is weighted with the obtained latitude weight, and the weighted distortion and the weight are accumulated.

$$wDi = wi\cdot Di$$

$$SWD = SWD + wDi$$

$$SW = SW + wi$$

After all points in the uniformly sampled set of points are evaluated, the L-SPSNR of component c is calculated as follows. P is the peak value of the sample value.

$$L\text{-}SPSNR(c) = 10\cdot\log(P^2 / (SWD/SW))$$

[0101] It may be noted that the value of L-SPSNR is represented in decibels in the equation above. It should be understood that in other embodiments, L-SPSNR is represented using a measure other than decibels.

[0102] While the foregoing exemplary embodiments illustrate in detail methods for calculating L-SPSNR, it should be understood that analogous methods may be employed for calculating other quality metrics and are contemplated within the scope of this disclosure. For example, the weight values may be generated in a variety of ways, including custom weights, for different weighted SPNSR. For example, in an embodiment, such weights are derived across a wide variety of sequences, and these weights are fixed and applied to new sequences. In such embodiments, among other embodiments, pre-trained weights may be used for rate-distortion optimization decisions in a newly encoded sequence, even if that sequence does not contribute to the training of the weights. As for AW-SPSNR, the weighted SPSNR calculation method described above can also be applied in the determination of coding parameters that are based on rate-distortion (R-D) optimization (e.g. motion vector, mode, quantization parameter, tree structure) determination. Such decisions may be used in, for example, motion estimation, inter or intra mode decisions, quantization and tree (quadtree, binary-tree) structure splitting decisions. In some embodiments, S-PSNR and/or L-PSNR measures may be used in a R-D-based decision process. In some such embodiments, a block being coded is mapped to a corresponding region on the unit sphere. Instead of using sample points from across the entire unit sphere, only points within that corresponding region on the unit sphere are used for S-PSNR and/or L-PSNR calculation. The resulting S-PSNR and/or L-PSNR measure may be used for R-D optimization.

Exemplary Video Encoder.

[0103] Embodiments disclosed herein, like the HEVC and JEM software, are built upon a block-based hybrid video coding framework.

[0104] FIG. 10 is a functional block diagram of a block-based hybrid video encoding system. The input video signal 1002 is processed block by block. In HEVC, extended block sizes (called a "coding unit" or CU) are used to efficiently compress high resolution (1080p and beyond) video signals. In HEVC, a CU can be up to 64x64 pixels, and bigger block size up to 256x256 is allowed in JEM. A CU can be further partitioned into prediction units (PU), for which separate prediction methods are applied. For each input video block (MB or CU), spatial prediction (1060) and/or temporal prediction (1062) may be performed. Spatial prediction (or "intra prediction") uses pixels from the already coded neighboring blocks in the same video picture/slice to predict the current video block. Spatial prediction reduces spatial redundancy inherent in the video signal. Temporal prediction (also referred to as "inter prediction" or "motion compensated prediction") uses

reconstructed pixels from the already coded video pictures to predict the current video block. Temporal prediction reduces temporal redundancy inherent in the video signal. Temporal prediction signal for a given video block is usually signaled by one or more motion vectors which indicate the amount and the direction of motion between the current block and its reference block. Also, if multiple reference pictures are supported (as is the case for the recent video coding standards such as H.264/AVC or HEVC), then for each video block, its reference picture index is sent additionally; and the reference index is used to identify from which reference picture in the reference picture store (1064) the temporal prediction signal comes.

[0105] After spatial and/or temporal prediction, the mode decision block (1080) in the encoder chooses the best prediction mode. In some embodiments, the best prediction mode is selected using a rate-distortion optimization method in which distortion is measured using one or more of the techniques described herein, such as Area Weighted PSNR. The prediction block is then subtracted from the current video block (1016); and the prediction residual is de-correlated using transform (1004) and quantized (1006) to achieve the target bit-rate. The quantized residual coefficients are inverse quantized (1010) and inverse transformed (1012) to form the reconstructed residual, which is then added back to the prediction block (1026) to form the reconstructed video block. Further in-loop filtering such as de-blocking filter and Adaptive Loop Filters may be applied (1066) on the reconstructed video block before it is put in the reference picture store (1064) and used to code future video blocks. To form the output video bit-stream 1020, coding mode (inter or intra), prediction mode information, motion information, and quantized residual coefficients are all sent to the entropy coding unit (1008) to be further compressed and packed to form the bit-stream.

Exemplary Hardware.

[0106] Note that various hardware elements of one or more of the described embodiments are referred to as "modules" that carry out (i.e., perform, execute, and the like) various functions that are described herein in connection with the respective modules. As used herein, a module includes hardware (e.g., one or more processors, one or more microprocessors, one or more microcontrollers, one or more microchips, one or more application-specific integrated circuits (ASICs), one or more field programmable gate arrays (FPGAs), one or more memory devices) deemed suitable by those of skill in the relevant art for a given implementation. Each described module may also include instructions executable for carrying out the one or more functions described as being carried out by the respective module, and it is noted that those instructions could take the form of or include hardware (i.e., hardwired) instructions, firmware instructions, software instructions, and/or the like, and may be stored in any suitable non-transitory computer-readable medium or media, such as commonly referred to as RAM, ROM, etc.

[0107] Exemplary embodiments disclosed herein are implemented using one or more wired and/or wireless network nodes, such as a wireless transmit/receive unit (WTRU) or other network entity.

[0108] FIG. 11 is a system diagram of an exemplary WTRU 1102, which may be employed as a video encoder and/or an apparatus for video quality evaluation in embodiments described herein. As shown in FIG. 11, the WTRU 1102 may include a processor 1118, a communication interface 1119 including a transceiver 1120, a transmit/receive element 1122, a speaker/microphone 1124, a keypad 1126, a display/touchpad 1128, a non-removable memory 1130, a removable memory 1132, a power source 1134, a global positioning system (GPS) chipset 1136, and sensors 1138. It will be appreciated that the WTRU 1102 may include any sub-combination of the foregoing elements while remaining consistent with an embodiment.

[0109] The processor 1118 may be 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 (FPGAs) circuits, any other type of integrated circuit (IC), a state machine, and the like. The processor 1118 may perform signal coding, data processing, power control, input/output processing, and/or any other functionality that enables the WTRU 1102 to operate in a wireless environment. The processor 1118 may be coupled to the transceiver 1120, which may be coupled to the transmit/receive element 1122. While FIG. 11 depicts the processor 1118 and the transceiver 1120 as separate components, it will be appreciated that the processor 1118 and the transceiver 1120 may be integrated together in an electronic package or chip.

[0110] The transmit/receive element 1122 may be configured to transmit signals to, or receive signals from, a base station over the air interface 1116. For example, in one embodiment, the transmit/receive element 1122 may be an antenna configured to transmit and/or receive RF signals. In another embodiment, the transmit/receive element 1122 may be an emitter/detector configured to transmit and/or receive IR, UV, or visible light signals, as examples. In yet another embodiment, the transmit/receive element 1122 may be configured to transmit and receive both RF and light signals. It will be appreciated that the transmit/receive element 1122 may be configured to transmit and/or receive any combination of wireless signals.

[0111] In addition, although the transmit/receive element 1122 is depicted in FIG. 11 as a single element, the WTRU 1102 may include any number of transmit/receive elements 1122. More specifically, the WTRU 1102 may employ MIMO technology. Thus, in one embodiment, the WTRU 1102 may include two or more transmit/receive elements 1122 (e.g., multiple antennas) for transmitting and receiving wireless signals over the air interface 1116.

[0112] The transceiver 1120 may be configured to modulate the signals that are to be transmitted by the transmit/receive element 1122 and to demodulate the signals that are received by the transmit/receive element 1122. As noted above, the WTRU 1102 may have multi-mode capabilities. Thus, the transceiver 1120 may include multiple transceivers for enabling the WTRU 1102 to communicate via multiple RATs, such as UTRA and IEEE 802.11, as examples.

[0113] The processor 1118 of the WTRU 1102 may be coupled to, and may receive user input data from, the speaker/microphone 1124, the keypad 1126, and/or the display/touchpad 1128 (e.g., a liquid crystal display (LCD) display unit or organic light-emitting diode (OLED) display unit). The processor 1118 may also output user data to the speaker/microphone 1124, the keypad 1126, and/or the display/touchpad 1128. In addition, the processor 1118 may access information from, and store data in, any type of suitable memory, such as the non-removable memory 1130 and/or the removable memory 1132. The non-removable memory 1130 may include random-access memory (RAM), read-only memory (ROM), a hard disk, or any other type of memory storage device. The removable memory 1132 may include a subscriber identity module (SIM) card, a memory stick, a secure digital (SD) memory card, and the like. In other embodiments, the processor 1118 may access information from, and store data in, memory that is not physically located on the WTRU 1102, such as on a server or a home computer (not shown).

[0114] The processor 1118 may receive power from the power source 1134, and may be configured to distribute and/or control the power to the other components in the WTRU 1102. The power source 1134 may be any suitable device for powering the WTRU 1102. As examples, the power source 1134 may include one or more dry cell batteries (e.g., nickel-cadmium (NiCd), nickel-zinc (NiZn), nickel metal hydride (NiMH), lithium-ion (Li-ion), and the like), solar cells, fuel cells, and the like.

[0115] The processor 1118 may also be coupled to the GPS chipset 1136, which may be configured to provide location information (e.g., longitude and latitude) regarding the current location of the WTRU 1102. In addition to, or in lieu of, the information from the GPS chipset 1136, the WTRU 1102 may receive location information over the air interface 1116 from a base station and/or determine its location based on the timing of the signals being received from two or more nearby base stations. It will be appreciated that the WTRU 1102 may acquire location information by way of any suitable location-determination method while remaining consistent with an embodiment.

[0116] The processor 1118 may further be coupled to other peripherals 1138, which may include one or more software and/or hardware modules that provide additional features, functionality and/or wired or wireless connectivity. For example, the peripherals 1138 may include sensors such

as an accelerometer, an e-compass, a satellite transceiver, a digital camera (for photographs or video), a universal serial bus (USB) port, a vibration device, a television transceiver, a hands free headset, a Bluetooth® module, a frequency modulated (FM) radio unit, a digital music player, a media player, a video game player module, an Internet browser, and the like.

[0117] FIG. 12 depicts an exemplary network entity 1290 that may be used in embodiments of the present disclosure, for example as a video encoder and/or an apparatus for video quality evaluation. As depicted in FIG. 12, network entity 1290 includes a communication interface 1292, a processor 1294, and non-transitory data storage 1296, all of which are communicatively linked by a bus, network, or other communication path 1298.

[0118] Communication interface 1292 may include one or more wired communication interfaces and/or one or more wireless-communication interfaces. With respect to wired communication, communication interface 1292 may include one or more interfaces such as Ethernet interfaces, as an example. With respect to wireless communication, communication interface 1292 may include components such as one or more antennae, one or more transceivers/chipsets designed and configured for one or more types of wireless (e.g., LTE) communication, and/or any other components deemed suitable by those of skill in the relevant art. And further with respect to wireless communication, communication interface 1292 may be equipped at a scale and with a configuration appropriate for acting on the network side—as opposed to the client side—of wireless communications (e.g., LTE communications, Wi-Fi communications, and the like). Thus, communication interface 1292 may include the appropriate equipment and circuitry (perhaps including multiple transceivers) for serving multiple mobile stations, UEs, or other access terminals in a coverage area.

[0119] Processor 1294 may include one or more processors of any type deemed suitable by those of skill in the relevant art, some examples including a general-purpose microprocessor and a dedicated DSP.

[0120] Data storage 1296 may take the form of any non-transitory computer-readable medium or combination of such media, some examples including flash memory, read-only memory (ROM), and random-access memory (RAM) to name but a few, as any one or more types of non-transitory data storage deemed suitable by those of skill in the relevant art could be used. As depicted in FIG. 12, data storage 1296 contains program instructions 1297 executable by processor 1294 for carrying out various combinations of the various network-entity functions described herein.

[0121] Although features and elements are described above in particular combinations, one of ordinary skill in the art will appreciate that each feature or element can be used alone or in any

combination with the other features and elements. In addition, the methods described herein may be implemented in a computer program, software, or firmware incorporated in a computer-readable medium for execution by a computer or processor. Examples of computer-readable storage media include, but are not limited to, a read only memory (ROM), a random access memory (RAM), a register, cache memory, semiconductor memory devices, magnetic media such as internal hard disks and removable disks, magneto-optical media, and optical media such as CD-ROM disks, and digital versatile disks (DVDs). A processor in association with software may be used to implement a radio frequency transceiver for use in a WTRU, UE, terminal, base station, RNC, or any host computer.

CLAIMS

1. A method of generating a distortion metric for at least a selected portion of a coded spherical video, where the spherical video is associated with a mapping between regions on a unit sphere and samples in a grid, the method comprising:

determining an area-weighted distortion for each of a plurality of samples in the grid, wherein the area-weighted distortion is an unweighted distortion at the sample multiplied by an area of the region of the unit sphere associated with the respective sample; and

calculating a sum of weighted distortions (SWD) by summing the determined area-weighted distortions.

2. The method of claim 1 implemented in a video encoder, wherein the unweighted distortion for a sample is determined with reference to a corresponding sample value in an input picture, the method further comprising coding the input picture, wherein at least one coding decision is made using a rate-distortion metric based at least in part on the SWD.

3. The method of claim 1, wherein the distortion metric is an area-weighted spherical peak signal-to-noise ratio (AW-SPSNR), the method further comprising:

calculating a sum of weights (SW) by summing the areas of the region of the unit sphere associated with the respective samples;

determining a peak value P from among the plurality of samples; and

calculating the AW-SPSNR as:

$$\text{AW-SPSNR}(c) = 10 \cdot \log(P^2 / (\text{SWD}/\text{SW})).$$

4. The method of claim 3, implemented in a video encoder, wherein the unweighted distortion for a sample is determined with reference to a corresponding sample value in an input picture, the method further comprising coding the input picture, wherein at least one coding decision is made using a rate-distortion metric, and wherein the rate-distortion metric is based at least in part on the area-weighted spherical peak signal-to-noise ratio (AW-SPSNR).

5. The method of claim 2 or 4, wherein the coding decision includes selection of a motion vector.

6. The method of claim 2 or 4, wherein the coding decision includes selection of a quantization parameter.

7. The method of claim 2 or 4, wherein the coding decision includes a tree splitting decision.

8. The method of claim 2 or 4, wherein the coding decision includes selection between intra mode and inter mode.

9. The method of claim 2 or 4, wherein the unweighted distortion for a sample is the square of a difference between the sample value and the corresponding sample value in the input picture.

10. The method of claim 1 or 3, wherein the unweighted distortion for a sample is the square of a difference between the sample value and a corresponding sample value in a reference picture.

11. The method of any of claims 1-4, wherein the mapping is an equirectangular projection (ERP), and wherein the area associated with a sample is a function of a latitude position of the respective sample.

12. The method of any of claims 1-4, wherein the mapping is an equirectangular projection (ERP), and wherein the area associated with a sample is proportional to cos(θ), where θ is a latitude position of the respective sample.

13. The method of any of claims 1-4, wherein the mapping is a cubemap, and wherein the area associated with a sample is proportional to

$$cos(\theta) \times cos^2(\phi) \times \left(\left(\left(sin(\theta) cos(\theta) sin(\phi) cos(\phi) \right)^2 + \left(cos^2(\theta)\, cos(\phi) \right)^2 \right)^{\frac{1}{2}} \right)$$

where θ is a latitude position and φ is a longitude position of the respective sample.

14. The method of any of claims 1-4, wherein different samples within the plurality of samples are associated with different area weights.

15. The method of any of claims 1-4, wherein a sum of weighted distortions (SWD) is calculated separately for luma and chroma components of the spherical video.

16. A system for generating a distortion metric for at least a selected portion of a coded spherical video, where the spherical video is associated with a mapping between regions on a unit sphere and samples in a grid, the system comprising a processor and a non-transitory computer-readable

storage medium storing instructions operative, when executed on the processor, to perform functions comprising:

determining an area-weighted distortion for each of a plurality of samples in the grid, wherein the area-weighted distortion is an unweighted distortion at the sample multiplied by an area of the region of the unit sphere associated with the respective sample; and

calculating a sum of weighted distortions (SWD) by summing the determined area-weighted distortions.

17. The system of claim 16 implemented in a video encoder, wherein the unweighted distortion for a sample is determined with reference to a corresponding sample value in an input picture, and wherein the instructions are further operative to perform functions comprising:

coding the input picture, wherein at least one coding decision is made using a rate-distortion metric based at least in part on the SWD.

18. A method of generating a distortion metric for at least a portion of a coded spherical video, where the spherical video is associated with a mapping between points on a unit sphere and points in a sample grid, the method comprising:

selecting a plurality of points on the unit sphere;

mapping the selected points on the unit sphere to respective points in the sample grid;

for each of the respective points in the sample grid, selecting a nearest sample;

for each of the selected nearest samples, determining a latitude-weighted distortion, wherein the latitude-weighted distortion is the unweighted distortion at the sample multiplied by a latitude-dependent weight based on a latitude of the selected nearest sample; and

calculating a sum of weighted distortions SWD by summing the determined latitude-weighted distortions.

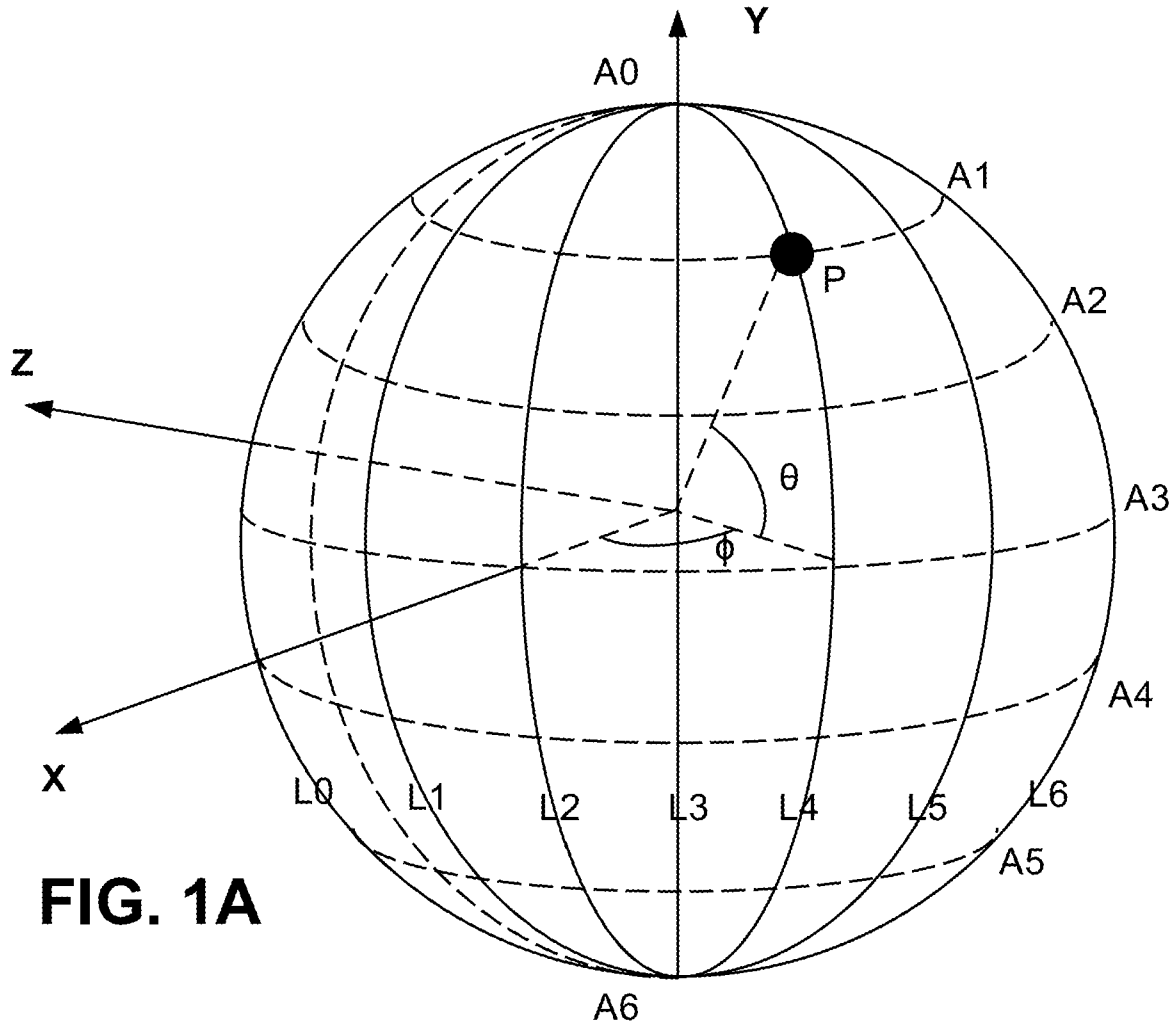

Y
A0
A1
P
A2
Z
θ
A3
ϕ
A4
X
L0
L1
L2
L3
L4
L5
L6
A5
A6

FIG. 1A

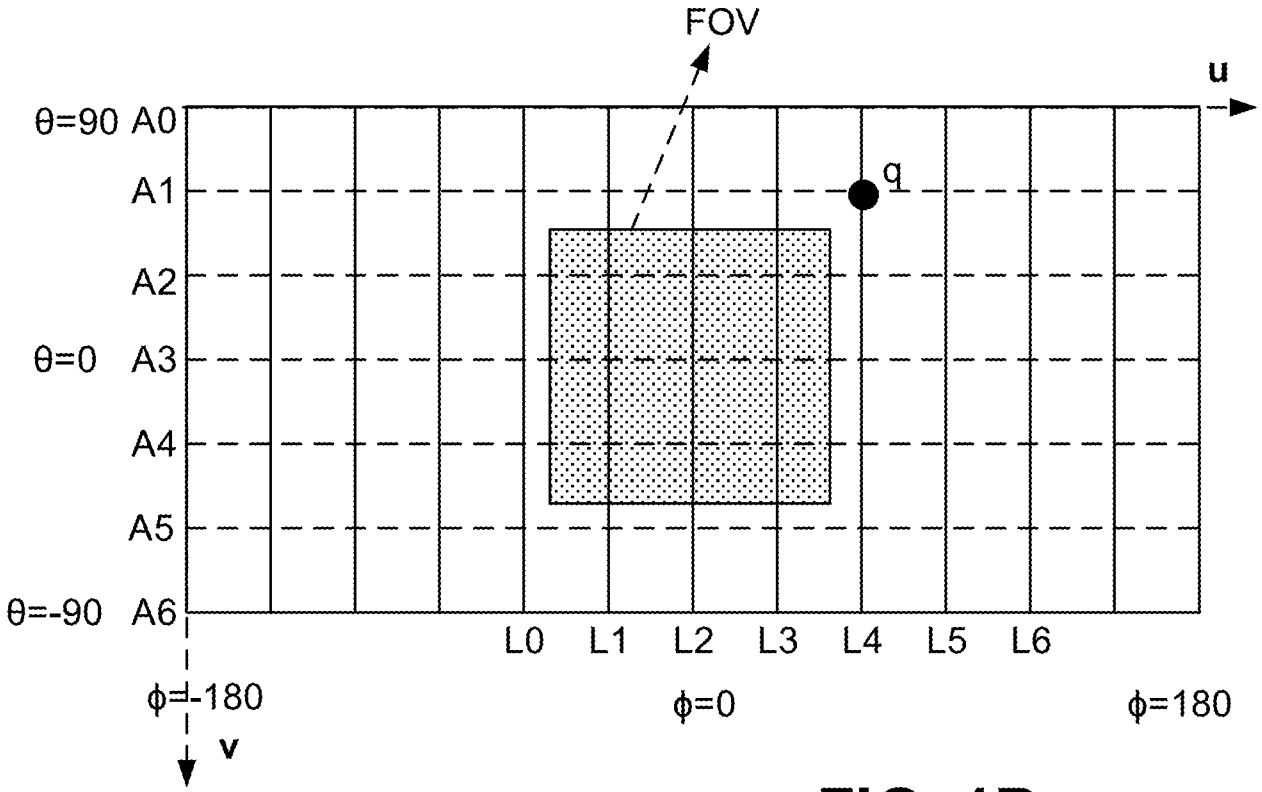
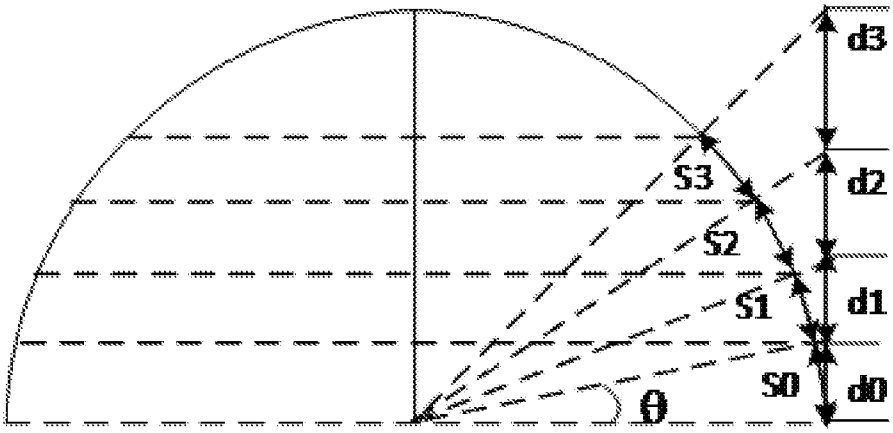

FOV
u
θ=90 A0
A1
q
A2
θ=0 A3
A4
A5
θ=-90 A6
L0 L1 L2 L3 L4 L5 L6
ϕ=-180
ϕ=0
ϕ=180
v

FIG. 1B



S0=S1=S2=S3, d3>d2>d1>d0

FIG. 2

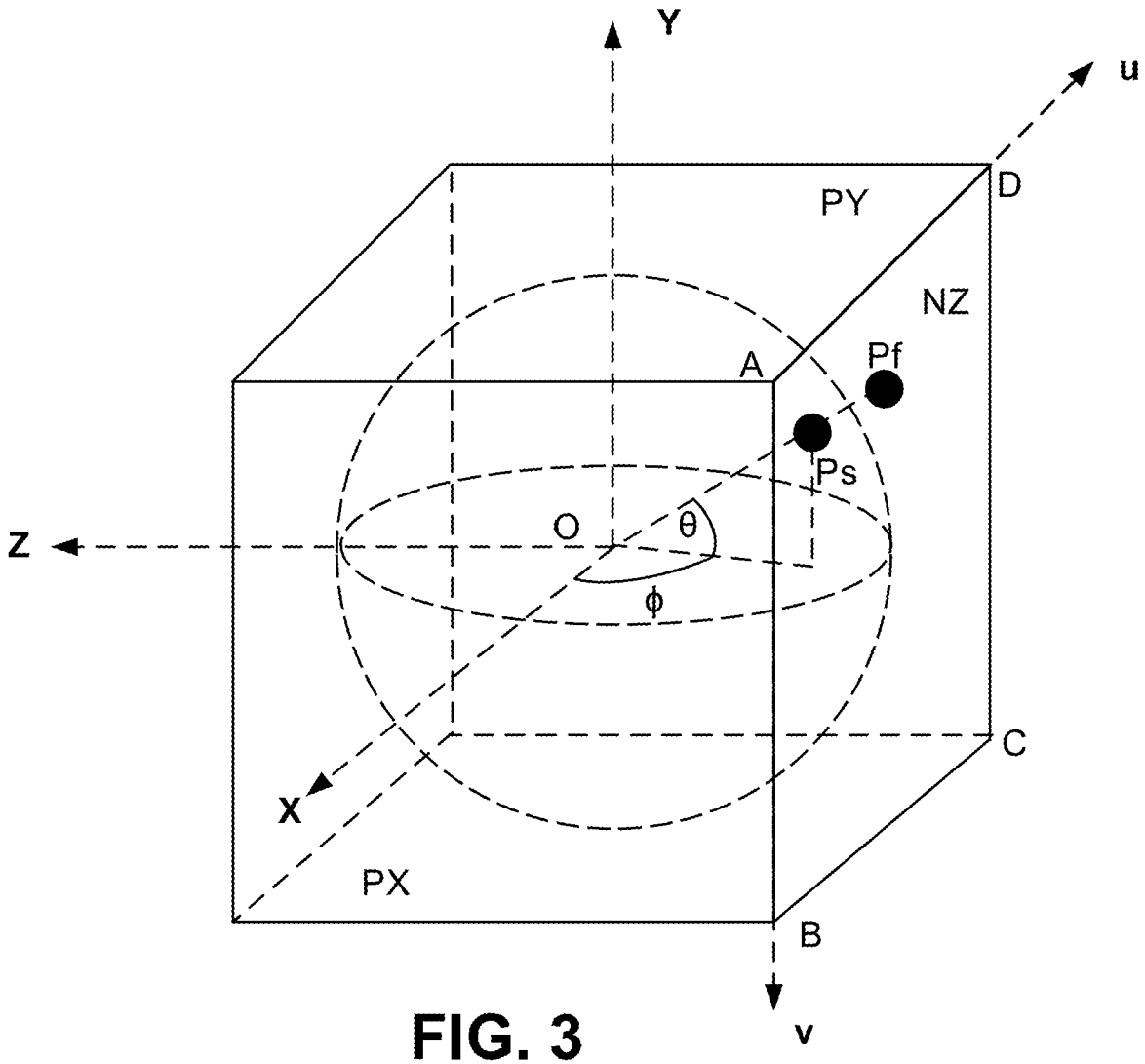


FIG. 3

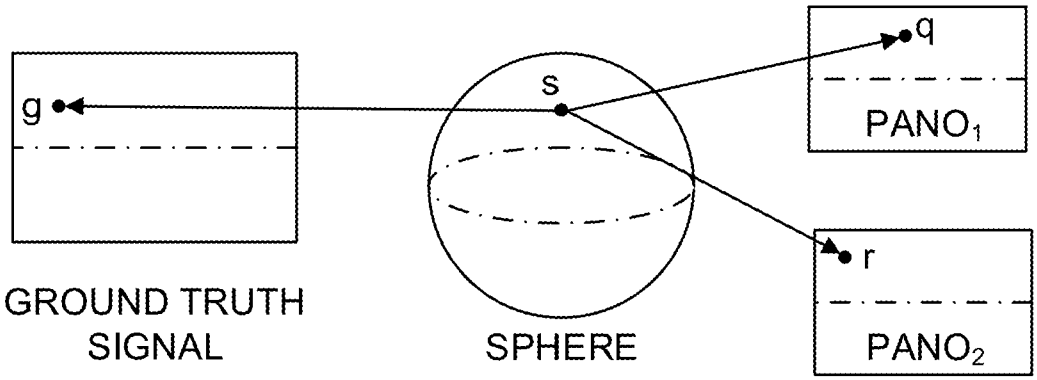

g
s
q
PANO1
r
PANO2
GROUND TRUTH SIGNAL
SPHERE

FIG. 4

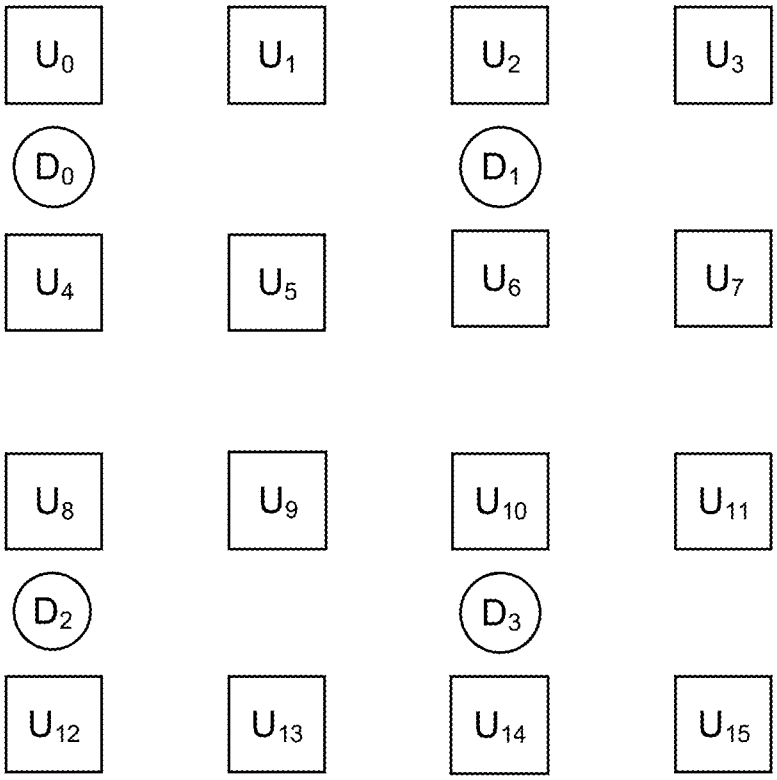

U0
U1
U2
U3
D0
D1
U4
U5
U6
U7
U8
U9
U10
U11
D2
D3
U12
U13
U14
U15

FIG. 5

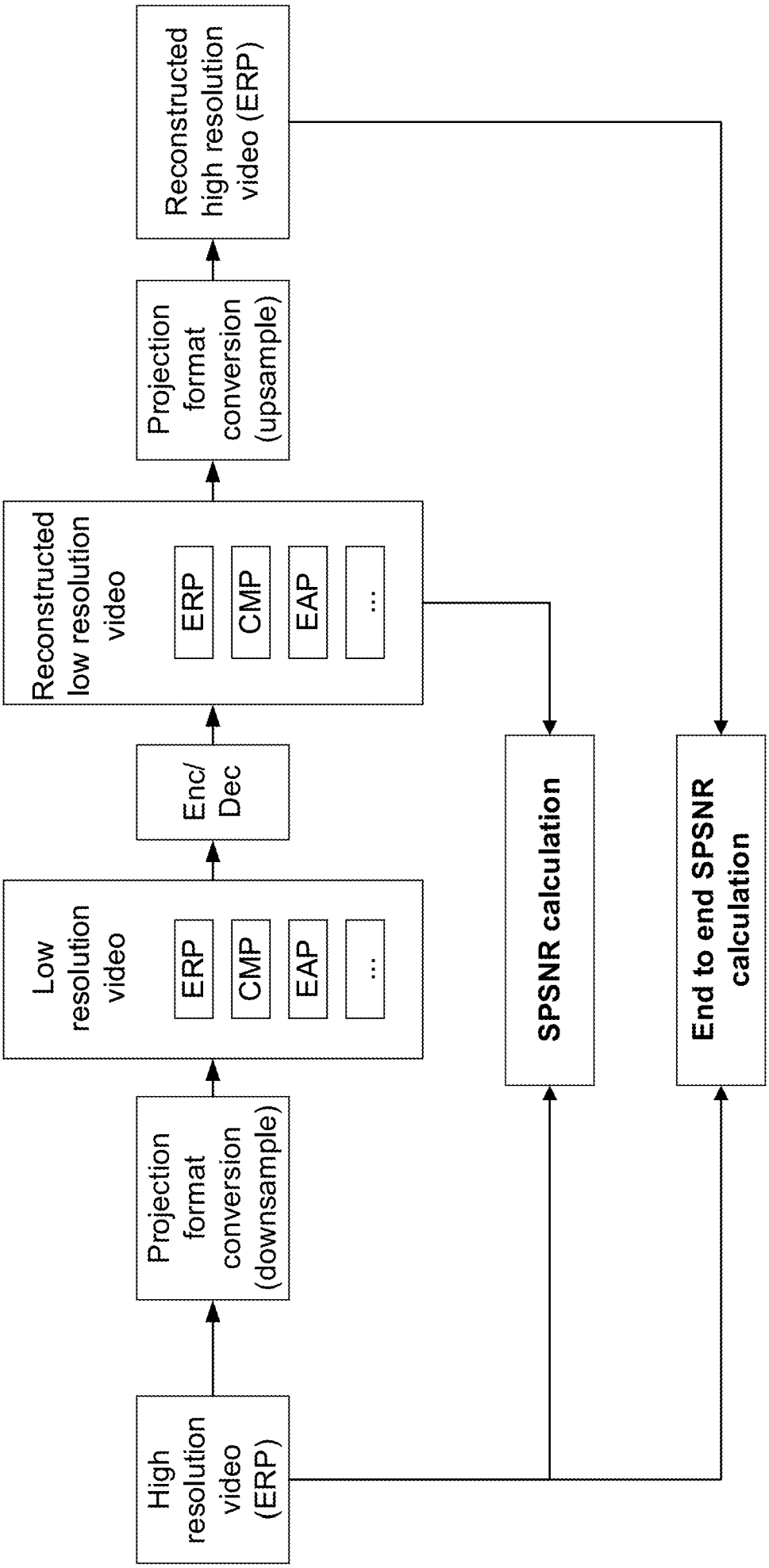

High resolution video (ERP)
Projection format conversion (downsample)
Low resolution video
ERP
CMP
EAP
...
Enc/ Dec
Reconstructed low resolution video
ERP
CMP
EAP
...
Projection format conversion (upsample)
Reconstructed high resolution video (ERP)
SPSNR calculation
End to end SPSNR calculation

FIG. 6

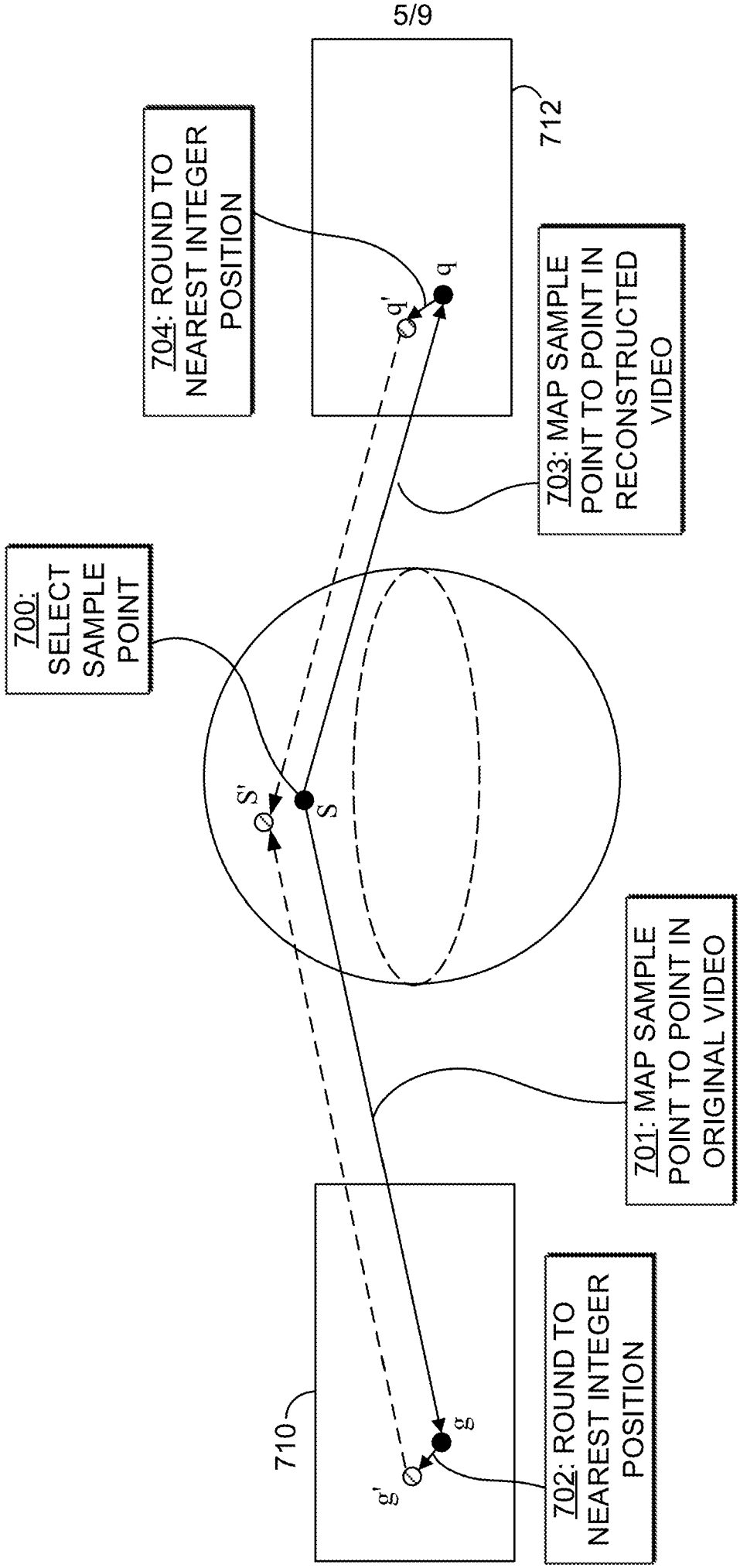

712
710
q
q'
g
g'
S
S'
704: ROUND TO NEAREST INTEGER POSITION
703: MAP SAMPLE POINT TO POINT IN RECONSTRUCTED VIDEO
700: SELECT SAMPLE POINT
701: MAP SAMPLE POINT TO POINT IN ORIGINAL VIDEO
702: ROUND TO NEAREST INTEGER POSITION

FIG. 7

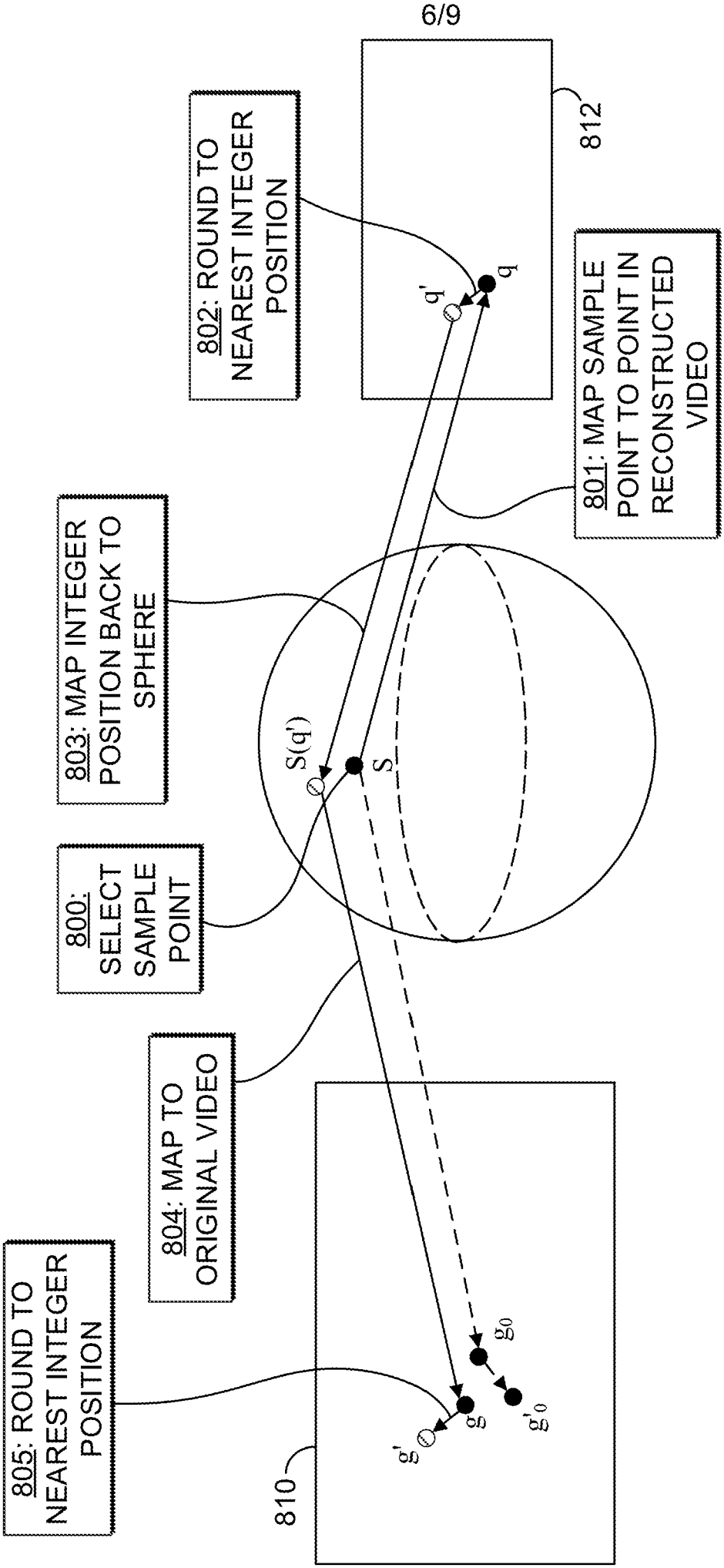

802: ROUND TO NEAREST INTEGER POSITION
812
801: MAP SAMPLE POINT TO POINT IN RECONSTRUCTED VIDEO
803: MAP INTEGER POSITION BACK TO SPHERE
800: SELECT SAMPLE POINT
804: MAP TO ORIGINAL VIDEO
805: ROUND TO NEAREST INTEGER POSITION
810
q'
q
S(q')
S
g'
g
g0
g'0

FIG. 8

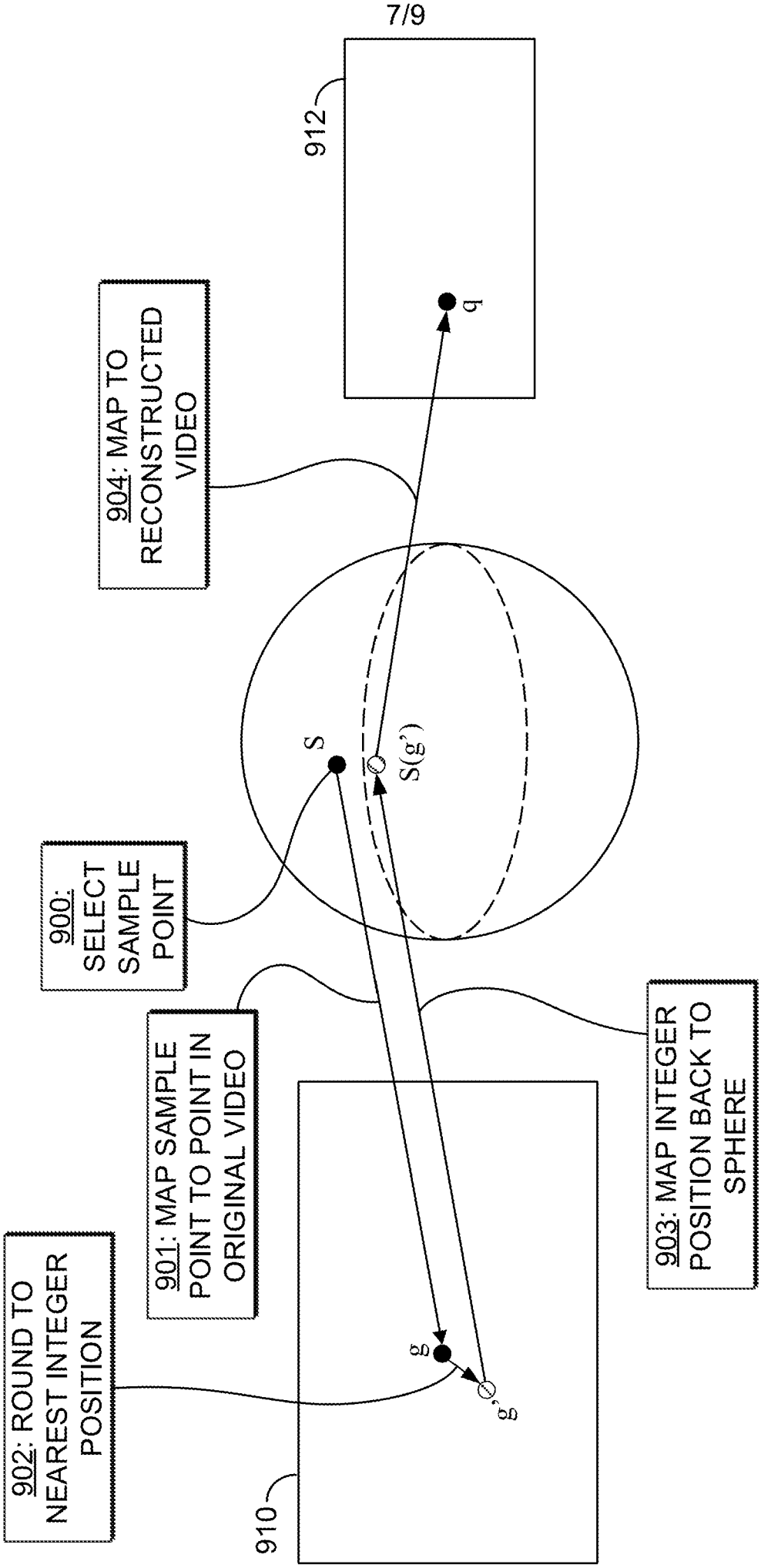

912
q
904: MAP TO RECONSTRUCTED VIDEO
S
S(g')
900: SELECT SAMPLE POINT
901: MAP SAMPLE POINT TO POINT IN ORIGINAL VIDEO
903: MAP INTEGER POSITION BACK TO SPHERE
902: ROUND TO NEAREST INTEGER POSITION
g
g'
910

FIG. 9

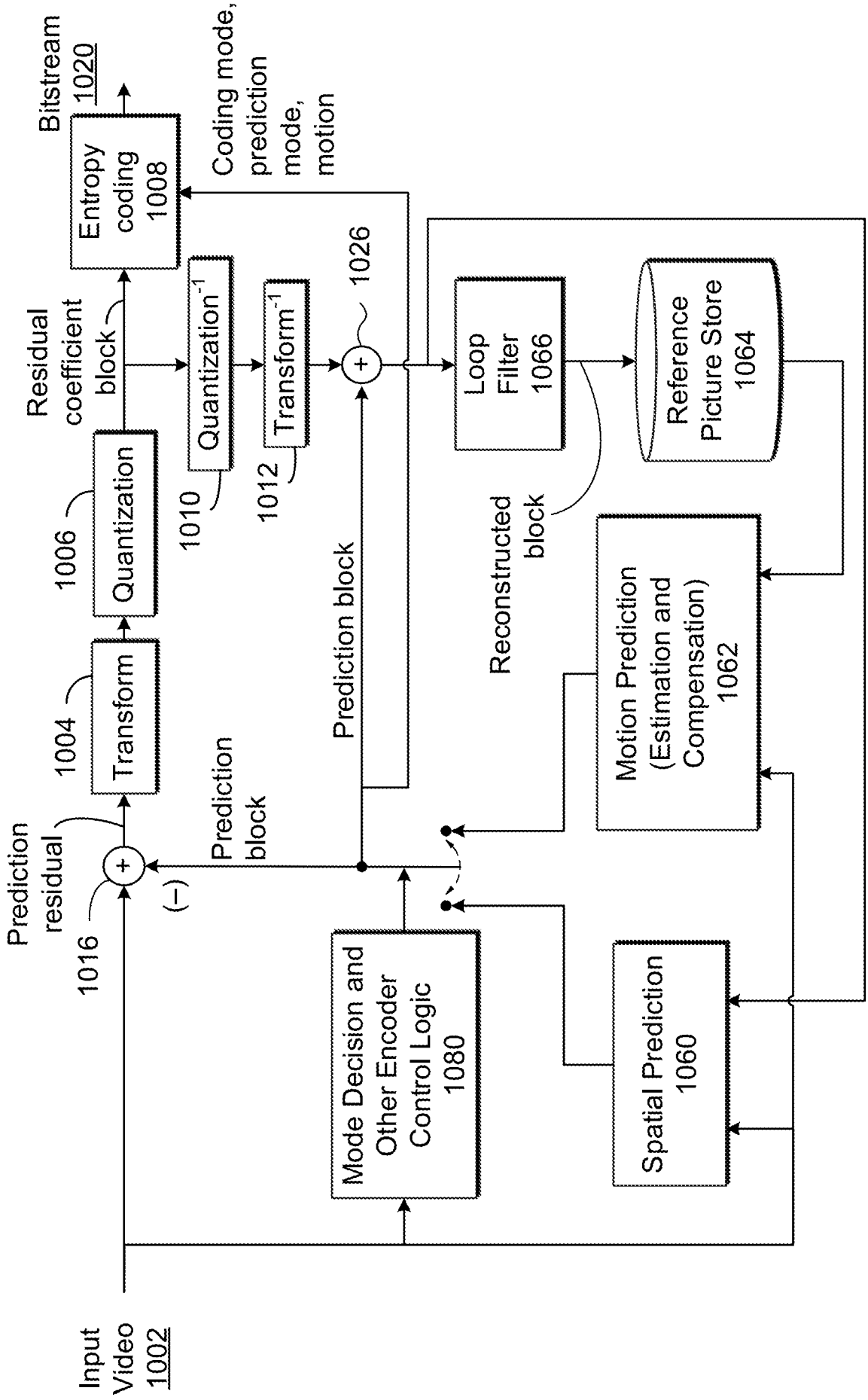

Input Video 1002
Prediction residual
1016
(–)
Prediction block
1004
Transform
1006
Quantization
Residual coefficient block
1010
Quantization^{-1}
1012
Transform^{-1}
1026
Prediction block
Entropy coding 1008
Bitstream 1020
Coding mode, prediction mode, motion
Loop Filter 1066
Reconstructed block
Reference Picture Store 1064
Motion Prediction (Estimation and Compensation) 1062
Spatial Prediction 1060
Mode Decision and Other Encoder Control Logic 1080

FIG. 10

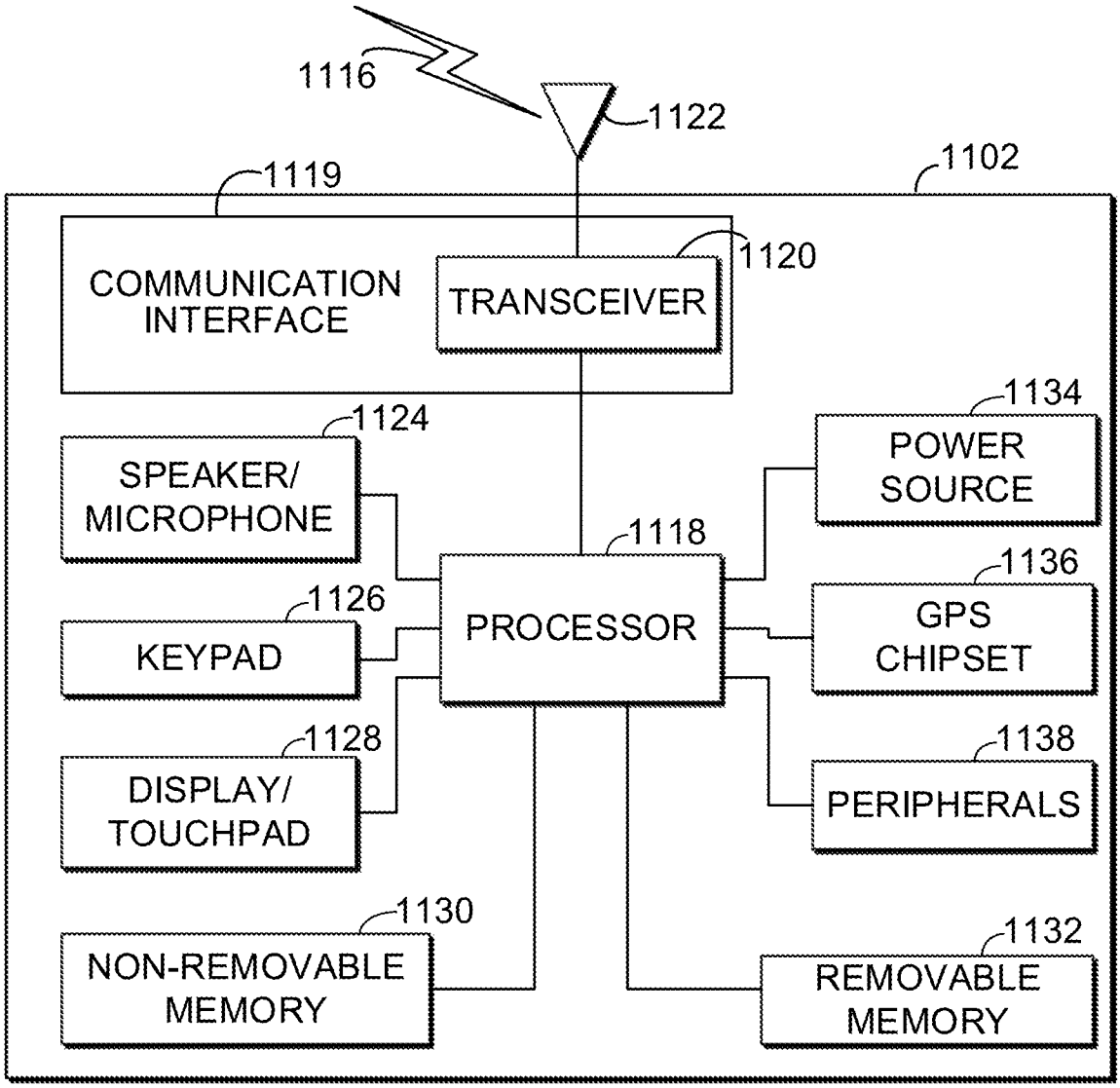

1116
1122
1102
1119
COMMUNICATION INTERFACE
TRANSCEIVER
1120
1134
POWER SOURCE
1124
SPEAKER/ MICROPHONE
1118
1136
GPS CHIPSET
1126
KEYPAD
PROCESSOR
1128
DISPLAY/ TOUCHPAD
1138
PERIPHERALS
1130
NON-REMOVABLE MEMORY
1132
REMOVABLE MEMORY

FIG. 11

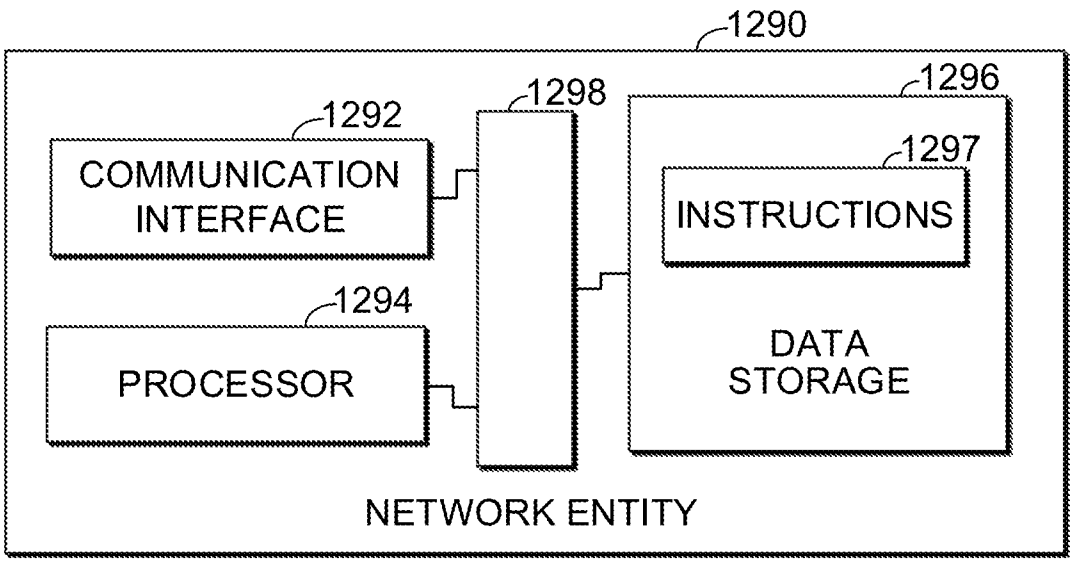

1290
1298
1296
1292
COMMUNICATION INTERFACE
1297
INSTRUCTIONS
1294
PROCESSOR
DATA STORAGE
NETWORK ENTITY

FIG. 12

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2017/042646

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N19/597 H04N19/154
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	YULE SUN ET AL: "[FTV-AHG] WS-PSNR for 360 video quality evaluation", 115. MPEG MEETING; 30-5-2016 - 3-6-2016; GENEVA; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11),, no. m38551, 27 May 2016 (2016-05-27), XP030066907, the whole document ----	1-18
X,P	VISHWANATH B ET AL: "AHG8: Area Weighted Spherical PSNR for 360 video quality evaluation", 4. JVET MEETING; 15-10-2016 - 21-10-2016; CHENGDU; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://PHENIX.INT-EVRY.FR/JVET/,, no. JVET-D0072, 6 October 2016 (2016-10-06), XP030150305, the whole document ----	1-18

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
2 October 2017	11/10/2017

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Oelbaum, Tobias

1