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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
17 March 2016 (17.03.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/040116 A1

(51) International Patent Classification:
H04N 19/139 (2014.01) *H04N 19/14* (2014.01)
H04N 19/176 (2014.01) *H04N 19/137* (2014.01)
H04N 19/124 (2014.01) *H04N 19/154* (2014.01)
H04N 19/136 (2014.01)

(21) International Application Number:
PCT/US2015/048353

(22) International Filing Date:
3 September 2015 (03.09.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/049,342 11 September 2014 (11.09.2014) US
14/532,947 4 November 2014 (04.11.2014) US
62/078,181 11 November 2014 (11.11.2014) US
62/158,523 7 May 2015 (07.05.2015) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:
US 14/532,947 (CIP)
Filed on 4 November 2014 (04.11.2014)

(71) Applicant: **EUCLID DISCOVERIES, LLC** [US/US]; 30 Monument Square, Suite 212, Concord, MA 01742 (US).

(72) Inventors: **LEE, Nigel**; 727 Hammond Street, Chestnut Hill, MA 02467 (US). **PARK, Sangseok**; 3620 Dresage Lane, Flower Mound, TX 75022 (US). **TUN, Myo**; 10812 Sedalia Drive, McKinney, Texas 75070 (US). **KOTTKE, Dane, P.**; 18 Marigold Place, Durham, North Carolina 27705 (US). **LEE, Jeyun**; 11125 Old Quarry Road, Austin, TX 78717 (US). **WEED, Christopher**; 33 Brimstone Lane, Sudbury, Massachusetts 01776 (US).

(74) Agents: **FESSENDEN, Giovanna, H.** et al.; Hamilton, Brook, Smith & Reynolds, P.C., 530 Virginia Rd, P.O. Box 9133, Concord, MA 01742-9133 (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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, 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, 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,

[Continued on next page]

(54) Title: PERCEPTUAL OPTIMIZATION FOR MODEL-BASED VIDEO ENCODING

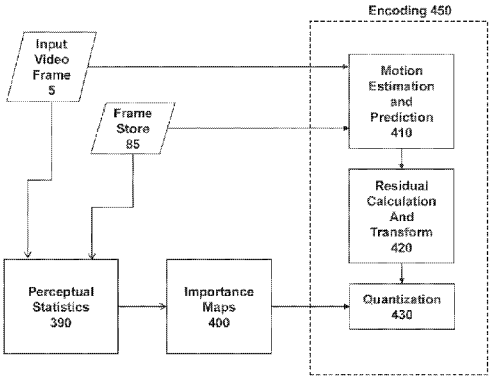

Figure 8B: Application of importance maps to modify quantization in video encoding

(57) Abstract: Perceptual statistics may be used to compute importance maps that indicate which regions of a video frame are important to the human visual system. Importance maps may be applied to the video encoding process to enhance the quality of encoded bitstreams. The temporal contrast sensitivity function (TCSF) may be computed from the encoder's motion vectors. Motion vector quality metrics may be used to construct a true motion vector map (TMVM) that can be used to refine the TCSF. Spatial complexity maps (SCMs) can be calculated from metrics such as block variance, block luminance, SSIM, and edge strength, and the SCMs can be combined with the TCSF to obtain a unified importance map. Importance maps may be used to improve encoding by modifying the criterion for selecting optimum encoding solutions or by modifying the quantization for each target block to be encoded.

WO 2016/040116 A1

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))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

PERCEPTUAL OPTIMIZATION FOR MODEL-BASED VIDEO ENCODING

RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application No.: 62/158,523 filed on May 7, 2015 and U.S. Provisional Application 62/078,181 filed on November 11, 2014. This application is also a Continuation-in-Part (CIP) of U.S. Application No.: 14/532,947, filed November 4, 2014, which claims the benefit of U.S. Provisional Application No. 61/950,784, filed March 10, 2014 and U.S. Provisional Application No. 62/049,342, filed September 11, 2014. The entire teachings of the above referenced applications are incorporated herein by reference.

BACKGROUND

[0002] Video compression can be considered the process of representing digital video data in a form that uses fewer bits when stored or transmitted. Video encoding can achieve compression by exploiting redundancies in the video data, whether spatial, temporal, or color-space. Video compression processes typically segment the video data into portions, such as groups of frames and groups of pels, to identify areas of redundancy within the video that can be represented with fewer bits than required by the original video data. When these redundancies in the data are exploited, greater compression can be achieved. An encoder can be used to transform the video data into an encoded format, while a decoder can be used to transform encoded video back into a form comparable to the original video data. The implementation of the encoder/decoder is referred to as a codec.

[0003] Standard encoders divide a given video frame into non-overlapping *coding units* or *macroblocks* (rectangular regions of contiguous pels) for encoding. The macroblocks (herein referred to more generally as "input blocks" or "data blocks") are typically processed in a traversal order of left to right and top to bottom in a video frame. Compression can be achieved when input blocks are predicted and encoded using previously-coded data. The process of encoding input blocks using spatially neighboring samples of previously-coded blocks within the same frame is referred to as *intra-prediction*. Intra-prediction attempts to exploit spatial redundancies in the data. The encoding of input blocks using similar regions from previously-coded frames, found using a motion estimation process, is referred to as *inter-prediction*. Inter-prediction attempts to exploit temporal redundancies in the data. The motion estimation process can generate a motion vector that specifies, for example, the

location of a matching region in a reference frame relative to an input block that is being encoded. Most motion estimation processes consist of two main steps: *initial motion estimation*, which provides an first, rough estimate of the motion vector (and corresponding temporal prediction) for a given input block, and *fine motion estimation*, which performs a local search in the neighborhood of the initial estimate to determine a more precise estimate of the motion vector (and corresponding prediction) for that input block.

[0004] The encoder may measure the difference between the data to be encoded and the prediction to generate a residual. The residual can provide the difference between a predicted block and the original input block. The predictions, motion vectors (for inter-prediction), residuals, and related data can be combined with other processes such as a spatial transform, a quantizer, an entropy encoder, and a loop filter to create an efficient encoding of the video data. The residual that has been quantized and transformed can be processed and added back to the prediction, assembled into a decoded frame, and stored in a framestore. Details of such encoding techniques for video will be familiar to a person skilled in the art.

[0005] MPEG-2 (H.262) and H.264 (MPEG-4 Part 10, Advanced Video Coding [AVC]), hereafter referred to as MPEG-2 and H.264, respectively, are two codec standards for video compression that achieve high quality video representation at relatively low bitrates. The basic coding units for MPEG-2 and H.264 are 16×16 macroblocks. H.264 is the most recent widely-accepted standard in video compression and is generally thought to be twice as efficient as MPEG-2 at compressing video data.

[0006] The basic MPEG standard defines three types of frames (or pictures), based on how the input blocks in the frame are encoded. An I-frame (intra-coded picture) is encoded using only data present in the frame itself and thus consists of only intra-predicted blocks. A P-frame (predicted picture) is encoded via forward prediction, using data from previously-decoded I-frames or P-frames, also known as *reference frames*. P-frames can contain either intra blocks or (forward-)predicted blocks. A B-frame (bi-predicted picture) is encoded via bi-directional prediction, using data from both previous and subsequent frames. B-frames can contain intra, (forward-)predicted, or bi-predicted blocks.

[0007] A particular set of reference frames is termed a Group of Pictures (GOP). The GOP contains only the decoded pels within each reference frame and does not include information as to how the input blocks or frames themselves were originally encoded (I-frame, B-frame, or P-frame). Older video compression standards such as MPEG-2 use one reference frame (in the past) to predict P-frames and two reference frames (one past, one

future) to predict B-frames. By contrast, more recent compression standards such as H.264 and HEVC (High Efficiency Video Coding) allow the use of multiple reference frames for P-frame and B-frame prediction. While reference frames are typically temporally adjacent to the current frame, the standards also allow reference frames that are not temporally adjacent.

[0008] Conventional inter-prediction is based on block-based motion estimation and compensation (BBMEC). The BBMEC process searches for the best match between the *target block* (the current input block being encoded) and same-sized regions within previously-decoded reference frames. When such a match is found, the encoder may transmit a motion vector, which serves as a pointer to the best match's position in the reference frame. For computational reasons, the BBMEC search process is limited, both temporally in terms of reference frames searched and spatially in terms of neighboring regions searched. This means that "best possible" matches are not always found, especially with rapidly changing data.

[0009] The simplest form of the BBMEC process initializes the motion estimation using a (0, 0) motion vector, meaning that the initial estimate of a target block is the co-located block in the reference frame. Fine motion estimation is then performed by searching in a local neighborhood for the region that best matches (i.e., has lowest error in relation to) the target block. The local search may be performed by exhaustive query of the local neighborhood (termed here *full block search*) or by any one of several "fast search" methods, such as a diamond or hexagonal search.

[0010] An improvement on the BBMEC process that has been present in standard codecs since later versions of MPEG-2 is the *enhanced predictive zonal search* (EPZS) method [Tourapis, A., 2002, "Enhanced predictive zonal search for single and multiple frame motion estimation," *Proc. SPIE 4671, Visual Communications and Image Processing*, pp. 1069-1078]. The EPZS method considers a set of motion vector candidates for the initial estimate of a target block, based on the motion vectors of neighboring blocks that have already been encoded, as well as the motion vectors of the co-located block (and neighbors) in the previous reference frame. The EPZS method hypothesizes that the video's motion vector field has some spatial and temporal redundancy, so it is logical to initialize motion estimation for a target block with motion vectors of neighboring blocks, or with motion vectors from nearby blocks in already-encoded frames. Once the set of initial estimates has been gathered, the EPZS method narrows the set via approximate rate-distortion analysis, after which fine motion estimation is performed.

[0011] For any given target block, the encoder may generate multiple inter-predictions to choose from. The predictions may result from multiple prediction processes (e.g., BBMEC, EPZS, or model-based schemes). The predictions may also differ based on the subpartitioning of the target block, where different motion vectors are associated with different subpartitions of the target block and the respective motion vectors each point to a subpartition-sized region in a reference frame. The predictions may also differ based on the reference frames to which the motion vectors point; as noted above, recent compression standards allow the use of multiple reference frames. Selection of the best prediction for a given target block is usually accomplished through *rate-distortion optimization*, where the best prediction is the one that minimizes the rate-distortion metric $D+\lambda R$, where the *distortion* D measures the error between the target block and the prediction, while the *rate* R quantifies the cost (in bits) to encode the prediction and λ is a scalar weighting factor.

[0012] Historically, *model-based* compression schemes have also been proposed to avoid the limitations of BBMEC prediction. These model-based compression schemes (the most well-known of which is perhaps the MPEG-4 Part 2 standard) rely on the detection and tracking of objects or features (defined generally as "components of interest") in the video and a method for encoding those features/objects separately from the rest of the video frame. Feature/object detection/tracking occurs independently of the spatial search in standard motion estimation processes, so feature/object tracks can give rise to a different set of predictions than achievable through standard motion estimation.

SUMMARY

[0013] Such feature/object-based model-based compression schemes, however, suffer from the challenges associated with segmenting video frames into object vs. non-object (or feature vs. non-feature) regions. First, because objects can be of arbitrary size, their *shapes* need to be encoded in addition to their texture (color content). Second, the tracking of multiple moving objects can be difficult, and inaccurate tracking causes incorrect segmentation, usually resulting in poor compression performance. A third challenge is that not all video content is composed of objects or features, so there needs to be a fallback encoding scheme when objects/features are not present.

[0014] Co-pending U.S. Patent Application No. 61/950,784, filed November 4, 2014 (herein "the '784 Application") presents a model-based compression scheme that avoids the segmentation challenge noted above. The continuous block tracker (CBT) of the '784

application does not detect objects and features, eliminating the need to segment objects and features from the non-object/non-feature background. Instead the CBT tracks all input blocks ("macroblocks") in the video frame as if they are regions of interest by combining frame-to-frame motion estimates into continuous tracks. In so doing, the CBT models motion in the video, achieving the benefits of higher-level modeling of the data to improve inter-prediction while avoiding the challenges of segmentation.

[0015] Other model-based compression approaches model the response of the human visual system (HVS) to the content in the video data as *importance maps* that indicate which parts of a video frame are most noticeable to human perception. Importance maps take on values for each input or data block in a video frame, and the importance map values for any given block may change from frame to frame throughout the video. Generally, importance maps are defined such that higher values indicate more important data blocks.

[0016] One type of importance map is the *temporal contrast sensitivity function* (TCSF) [de Lange, H., 1954, "Relationship between critical flicker frequency and a set of low frequency characteristics of the eye," *J. Opt. Soc. Am.*, 44:380-389], which measures the response of the HVS to temporally periodic stimuli and reveals that certain temporal characteristics in the data are noticeable to human observers. These temporal characteristics are related to the motion in the data, and the TCSF predicts that the most noticeable type of motion in the data is "moderate" motion that corresponds to neither very high nor very low temporal frequencies.

[0017] It is important to note that the TCSF requires accurate measurement of the velocities of moving content in the video to generate accurate temporal contrast values. These velocities can be approximated by computing optical flow, which describes the apparent motion of video content due to camera motion and/or object motion. However, most standard video encoders employ motion estimation processes that optimize compression efficiency rather than accurately computing optical flow.

[0018] Another type of importance map is based on spatial contrast sensitivity and measures the HVS response to spatial characteristics such as brightness, edges, spatial frequencies, and color. The *spatial-contrast sensitivity function* (SCSF) [see, e.g., Barten, P., 1999, *Contrast Sensitivity of the Human Eye and Its Effects on Image Quality*, SPIE Press], also known simply as the contrast sensitivity function (CSF), measures spatial contrast that is significant to the HVS and has been applied successfully in the JPEG 2000 image compression standard to reduce image compression artifacts. Objects and features are also

typically detected with the aid of spatial contrast measures (e.g., the presence of edges as indicated by spatial frequency gradients). While spatial contrast sensitivity has been studied and exploited in the context of image compression (e.g., the JPEG 2000 codec) and many video compression processes based on object and feature detection have been proposed, temporal contrast sensitivity as represented in the TCSF has not previously been applied to video compression.

[0019] Some disclosed inventive embodiments apply importance maps to video compression to enhance the quality of video encoding. In one example embodiment, within a standard video encoding processing stream, temporal frequency is computed by using structural similarity (SSIM) in the colorspace domain to approximate wavelength and the encoder's motion vectors to approximate velocity. Temporal frequency then serves as an input to the temporal contrast sensitivity function (TCSF), which can be computed for every data block to generate a temporal importance map that indicates which regions of the video frame are most noticeable to human observers.

[0020] In a further example embodiment, information about the relative quality of the motion vectors generated by the encoder can be computed at different points in the encoding process and then used to generate a *true motion vector map* that outputs, for each target block, how reliable its motion vector is. The true motion vector map, which takes on values of 0 or 1, can then be used as a mask to refine the TCSF, such that the TCSF is not used for target blocks whose motion vectors are not accurate (i.e., the true motion vector map is 0).

[0021] In a further embodiment, *spatial complexity maps* (SCMs) can be calculated from metrics such as block variance, block luminance, and edge detection to determine the spatial contrast of a given target block relative to its neighbors. In another embodiment, information from the SCMs can be combined with the TCSF to obtain a composite, unified importance map. The combination of spatial and temporal contrast information in the unified importance map effectively balances both aspects of human visual response.

[0022] In one example embodiment, the unified importance map (including information from both the TCSF and SCM) is used to weight the distortion part of the standard rate-distortion metric, $D+\lambda R$. This results in a modified rate-distortion optimization that is weighted toward solutions that fit the relative perceptual importance of each target block, either low-distortion solutions when the importance map is closer to its maximum or low-rate solutions when the importance map is closer to its minimum. In an alternative embodiment, either the TCSF or SCM may be used individually for the above purpose.

[0023] In another example embodiment, the TCSF (with true motion vector refinement) and SCM can be used to modify the block-level quantization of the encoder. In target blocks where the importance maps take on high values, the quantization parameter is reduced relative to the frame quantization parameter, resulting in higher quality for those blocks. In target blocks where the importance maps take on low values, the quantization parameter is increased relative to the frame quantization parameter, resulting in lower quality for those blocks. In an alternative embodiment, either the TCSF or SCM may be used individually for the above purpose.

[0024] While the TCSF can be computed for any encoder that incorporates inter-prediction and generates motion vectors (used by the TCSF to approximate the velocity of the content in the video), application of the TCSF to video compression is most effective within a model-based compression framework such as the continuous block tracker (CBT) of the '784 Application that provides accurate determination of which motion vectors are true motion vectors. As noted above, most standard video encoders compute motion vectors that optimize compression efficiency rather than reflecting true motion. By contrast, the CBT provides both motion vectors suitable for high compression efficiency and modeling information that maximizes the effectiveness of the TCSF.

[0025] Some example inventive embodiments are structured so that the resulting bitstream is compliant with any video compression standard – including, but not limited to, MPEG-2, H.264, and HEVC – that employs block-based motion estimation followed by transform, quantization, and entropy encoding of residual signals. The present invention can also be applied to non-standard video encoders that are not block-based, as long as the encoder incorporates inter-prediction and generates motion vectors.

[0026] Some example embodiments may include methods and systems of encoding video data, as well as any codecs (encoders/decoders) for implementing the same. A plurality of video frames having non-overlapping target blocks may be processed by an encoder. The plurality of video frames may be encoded by the encoder using importance maps, such that the importance maps modify the quantization, as well as the encoding quality of each target block to be encoded in each video frame.

[0027] The importance maps may be formed using at least one of: temporal information or spatial information. If both temporal and spatial information are used, the importance map is considered a unified importance map. The importance maps may be configured so that they indicate/identify/represent parts of a video frame in the plurality of video frames that are the

most noticeable to human perception. Specifically, in blocks where the importance maps take on high values, the block quantization parameter (QP) is reduced relative to the frame quantization parameter QP_{frame}, resulting in higher quality for those blocks; and in target blocks where the importance maps take on low values, the block quantization parameter is increased relative to the frame quantization parameter QP_{frame}, resulting in lower quality for those blocks.

[0028] The spatial information may be provided by a rule-based spatial complexity map (SCM) in which the initial step is to determine which target blocks in the frame have higher variance than the average block variance in the frame, var_{frame} . For such blocks, a QP value may be assigned that is higher than the frame quantization parameter QP_{frame} , with the block QP assignment QP_{block} scaled linearly between QP_{frame} and the maximum quantization parameter QP_{max} , based on how much higher the block variance var_{block} is than var_{frame} .

[0029] The temporal information may preferably be provided by a temporal contrast sensitivity function (TCSF) that indicates which target blocks are most temporally noticeable to a human observer and a true motion vector map (TMVM) that indicates which target blocks correspond to foreground data. It should be noted that the TCSF may only be considered valid for those target blocks identified as foreground data.

[0030] A high-variance block may have its block QP assignment QP_{block} further refined by the TCSF and TMVM, such that if the TMVM identifies a target block as foreground data and the TCSF has a log contrast sensitivity value less than 0.5 for that block, QP_{block} is raised by 2.

[0031] The SCM may include luminance masking, in which target blocks that are either very bright (luminance above 170) or very dark (luminance below 60) have their block quantization parameters QP_{block} adjusted back to QP_{max} . The SCM may include dynamic determination of QP_{max} based on the quality level of the encoded video, where quality is measured using an average structural similarity (SSIM) calculation of target blocks in Intra (I) frames, together with the average block variance var_{frame} of such frames; such that when the measured quality is low, the value of QP_{max} is lowered to something closer to QP_{frame}.

[0032] Very-low-variance blocks may be assigned fixed, low QP values QP_{block} to ensure high-quality encoding in those regions, such that the lower the block variance, the lower the value of QP_{block} (and the higher the quality). The assignment of low QP values QP_{block} for very-low-variance blocks may be fixed first for I frames and then determined for P and B frames using the ipratio and pbratio parameters. Blocks that are low-variance but do not

qualify as very-low-variance are examined to determine whether quality enhancement is needed for those blocks; in that an initial estimate of the block QP, QP_{block} , is calculated by average the QP values of neighboring, already encoded blocks to the left, top-left, right, and top-right of the current block. An estimate of the SSIM of the current block, $SSIM_{est}$, may be calculated from the SSIM values of neighboring, already-encoded blocks to the left, top-left, right, and top-right of the current block. The value of QP_{block} may be lowered by 2 if $SSIM_{est}$ is lower than 0.9.

[0033] In some embodiments, the quality enhancement is only applied to those blocks that are identified as foreground data by the TMVM and for which the TCSF has log contrast sensitivity value greater than 0.8. The TMVM may be set to 1 only for foreground data.

[0034] In some embodiments, the temporal frequency of the TCSF is computed by using SSIM in the colorspace domain between the target block and its reference block to approximate wavelength and by using motion vector magnitudes and the framerate to approximate velocity.

[0035] The TCSF may be calculated over multiple frames, such that the TCSF for the current frame is a weighted average of the TCSF maps over recent frames, with more recent frames receiving higher weighting.

[0036] The foreground data may be identified by computing the difference between the encoder motion vector for a given target block and the global motion vector for that block, such that blocks with sufficiently large differences are determined to be foreground data.

[0037] For data blocks that are identified as foreground data, the encoder motion vector may be subtracted from the global motion vector to obtain a differential motion vector, and it is the magnitude of the differential motion vector that is used in calculating the temporal frequency of the TCSF.

[0038] Computer-based methods, codecs (encoders/decoders), and other computer systems and apparatus for processing video data may embody the foregoing principles of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0039] The foregoing will be apparent from the following more particular description of example embodiments of the invention, as illustrated in the accompanying drawings in which like reference characters refer to the same parts throughout the different views. The drawings

are not necessarily to scale, with emphasis instead placed on illustrating embodiments of the present invention.

[0040] FIG. 1 is a block diagram depicting a standard encoder configuration.

[0041] FIG. 2 is a block diagram depicting the steps involved in inter-prediction for general encoders.

[0042] FIG. 3 is a block diagram depicting the steps involved in initial motion estimation via continuous block tracking.

[0043] FIG. 4 is a block diagram depicting unified motion estimation via a combination of continuous block tracking and enhanced predictive zonal search.

[0044] FIG. 5 is a plot depicting a recent measurement of the temporal contrast sensitivity function by Wooten et al [2010].

[0045] FIG. 6 is a block diagram depicting the calculation of structural similarity (SSIM) in CIE 1976 Lab colorspace, according to an embodiment of the invention.

[0046] FIG. 7 is a block diagram depicting the general application of perceptual statistics to improve the perceptual quality of video encodings, according to an embodiment of the invention.

[0047] FIG. 8A is a block diagram depicting the use of perceptual statistics to modify inter-prediction via continuous block tracking to improve the perceptual quality of video encodings, according to an embodiment of the invention.

[0048] FIG. 8B is a block diagram depicting an example process of encoding using importance maps to modify block quantization.

[0049] FIG. 9A is a schematic diagram of a computer network environment in which embodiments are deployed.

[0050] FIG. 9B is a block diagram of the computer nodes in the network of FIG. 9A.

DETAILED DESCRIPTION

[0051] The teachings of all patents, published applications and references cited herein are incorporated by reference in their entirety. A description of example embodiments of the invention follows.

[0052] The invention can be applied to various standard encodings. In the following, unless otherwise noted, the terms “conventional” and “standard” (sometimes used together with “compression,” “codecs,” “encodings,” or “encoders”) can refer to MPEG-2, MPEG-4, H.264, or HEVC. “Input blocks” are referred to without loss of generality as the basic coding

unit of the encoder and may also sometimes be referred to interchangeably as "data blocks" or "macroblocks." The current input block being encoded is referred to as a "target block."

Video Encoding and Inter-Prediction via Continuous Block Tracking

[0053] The encoding process may convert video data into a compressed, or encoded, format. Likewise, the decompression or decoding process may convert compressed video back into an uncompressed, or raw, format. The video compression and decompression processes may be implemented as an encoder/decoder pair commonly referred to as a *codec*.

[0054] FIG. 1 is a block diagram of a standard transform-based, motion-compensated encoder. The encoder in FIG. 1 may be implemented in a software or hardware environment, or combination thereof. The encoder may include any combination of components, including, but not limited to, a motion estimation module 15 that feeds into an inter-prediction module 20, an intra-prediction module 30, a transform and quantization module 60, an inverse transform and quantization module 70, an in-loop filter 80, a frame store 85, and an entropy encoding module 90. For a given input video block 10 ("input block" for short, or macroblock or "data block"), the purpose of the prediction modules (both inter-prediction and intra-prediction) is to generate the best predicted signal 40 for the input block. The predicted signal 40 is subtracted from the input block 10 to create a prediction residual 50 that undergoes transform and quantization 60. The quantized coefficients 65 of the residual then get passed to the entropy encoding module 90 for encoding into the compressed bitstream. The quantized coefficients 65 also pass through the inverse transform and quantization module 70, and the resulting signal (an approximation of the prediction residual) gets added back to the predicted signal 40 to create a reconstructed signal 75 for the input block 10. The reconstructed signal 75 may be passed through an in-loop filter 80 such as a deblocking filter, and the (possibly filtered) reconstructed signal becomes part of the frame store 85 that aids prediction of future input blocks. The function of each of the components of the encoder shown in FIG. 1 is well known to one of ordinary skill in the art.

[0055] FIG. 2 depicts the steps in standard inter-prediction (30 in FIG. 1), where the goal is to encode new data using previously-decoded data from earlier frames, taking advantage of temporal redundancy in the data. In inter-prediction, an input block 10 from the frame currently being encoded (also called the *target frame*) is "predicted" from a region of the same size within a previously-decoded *reference frame*, stored in the frame store 85 from FIG. 1. The two-component vector indicating the (x, y) displacement between the location of

the input block in the frame being encoded and the location of its matching region in the reference frame is termed a *motion vector*. The process of *motion estimation* thus involves determining the motion vector that best links an input block to be encoded with its matching region in a reference frame.

[0056] Most inter-prediction processes begin with *initial motion estimation* (110 in FIG. 2), which generates one or more rough estimates of "good" motion vectors 115 for a given input block. This is followed by an optional *motion vector candidate filtering* step 120, where multiple motion vector candidates can be reduced to a single candidate using an approximate rate-distortion metric. In rate-distortion analysis, the best motion vector candidate (prediction) is chosen as the one that minimizes the rate-distortion metric $D+\lambda R$, where the *distortion* D measures the error between the input block and its matching region, while the *rate* R quantifies the cost (in bits) to encode the prediction and λ is a scalar weighting factor. The actual rate cost contains two components: *texture bits*, the number of bits needed to encode the quantized transform coefficients of the residual signal (the input block minus the prediction), and *motion vector bits*, the number of bits needed to encode the motion vector. Note that motion vectors are usually encoded differentially, relative to already-encoded motion vectors. In the early stages of the encoder, texture bits are not available, so the rate portion of the rate-distortion metric is approximated by the motion vector bits, which in turn are approximated as a *motion vector penalty factor* dependent on the magnitude of the differential motion vector. In the motion vector candidate filtering step 120, then, the approximate rate-distortion metric is used to select either a single "best" initial motion vector or a smaller set of "best" initial motion vectors 125. The initial motion vectors 125 are then refined with *fine motion estimation* 130, which performs a local search in the neighborhood of each initial estimate to determine a more precise estimate of the motion vector (and corresponding prediction) for the input block. The local search is usually followed by *subpixel refinement*, in which integer-valued motion vectors are refined to half-pixel or quarter-pixel precision via interpolation. The fine motion estimation block 130 produces a set of refined motion vectors 135.

[0057] Next, for a given fine motion vector 135, a mode generation module 140 generates a set of candidate predictions 145 based on the possible encoding modes of the encoder. These modes vary depending on the codec. Different encoding modes may account for (but are not limited to) interlaced vs. progressive (field vs. frame) motion estimation, direction of the reference frame (forward-predicted, backward-predicted, bi-predicted), index of the

reference frame (for codecs such as H.264 and HEVC that allow multiple reference frames), inter-prediction vs. intra-prediction (certain scenarios allowing reversion to intra-prediction when no good inter-predictions exist), different quantization parameters, and various subpartitions of the input block. The full set of prediction candidates 145 undergoes "final" rate-distortion analysis 150 to determine the best single candidate. In "final" rate-distortion analysis, a precise rate-distortion metric $D+\lambda R$ is used, computing the prediction error D for the distortion portion (usually calculated as sum of squared errors [SSE]) and the actual encoding bits R (from the entropy encoding 90 in FIG. 1) for the rate portion. The final prediction 160 (or 40 in FIG. 1) is the one that has lowest rate-distortion score $D+\lambda R$ among all the candidates, and this final prediction is passed to the subsequent steps of the encoder, along with its motion vector and other encoding parameters.

[0058] FIG. 3 depicts how initial motion estimation can be performed during inter-prediction via *continuous block tracking* (CBT). CBT is useful when there is a gap of greater than one frame between the target frame and the reference frame from which temporal predictions are derived. For MPEG-2, a typical GOP structure of IBBPBBP (consisting of intra-predicted I-frames, bi-predicted B-frames, and forward-predicted P-frames) allows reference frames as many as three frames away from the current frame, as B-frames cannot act as reference frames in MPEG-2. In H.264 and HEVC, which allow multiple reference frames for each frame to be encoded, the same GOP structure allows reference frames to be located six or more frames away from the current frame. For longer GOP structures (e.g., seven B-frames in-between each P-frame), reference frames can be located even further from the target frame. When there is a greater-than-one-frame gap between the current frame and the reference frame, continuous tracking enables the encoder to capture motion in the data in a way that standard temporal prediction methods cannot, allowing CBT to produce superior temporal predictions.

[0059] The first step in CBT is to perform *frame-to-frame tracking* (210 in FIG. 3). For each input block 10 in a frame, motion vectors are calculated in both the backward direction to the previous frame in the frame buffer 205 and the forward direction to the next frame in the frame buffer. In one embodiment, frame-to-frame tracking operates on frames from the original source video, not reconstructed reference frames. This is advantageous because source video frames are not corrupted by quantization and other coding artifacts, so tracking based on source video frames more accurately represents the true motion field in the video.

Frame-to-frame tracking may be carried out using either conventional block-based motion estimation (BBME) or hierarchical motion estimation (HME).

[0060] The result of frame-to-frame tracking is a set of frame-to-frame motion vectors 215 that signify, for each input block in a frame, the best matching region in the most recent frame in the frame buffer 205, and, for each block of the most recent frame in the frame buffer 205, the best matching region in the current frame. Continuous tracking 220 then aggregates available frame-to-frame tracking information to create continuous tracks across multiple reference frames for each input block. Details of how to perform continuous tracking are found in the '784 Application, which is incorporated by reference herein in its entirety. The output of continuous tracking 220 are the continuous block tracking (CBT) motion vectors 225 that track all input blocks in the current frame being encoded to their matching regions in past reference frames. The CBT motion vectors are the initial motion vectors (125 in FIG. 2) for the CBT, and they can be refined with fine motion estimation (130 in FIG. 2) as noted above.

[0061] FIG. 4 depicts how the CBT can be combined with the EPZS method to create a *unified* motion estimation process, according to an embodiment of the invention. In FIG. 4, CBT generates its motion vectors through frame-to-frame tracking 210 and continuous tracking 220 for initial motion estimation 110, followed by local search and subpixel refinement 250 for fine motion estimation 130. EPZS generates its initial motion vectors through a candidate generation module 230, followed by a candidate filtering module 240, with the filtering carried out via approximate rate-distortion analysis as detailed above. This is followed by fine motion estimation 130 via local search and subpixel refinement 260. The resulting CBT motion vector 255 and EPZS motion vector 265 are both passed forward to the remaining inter-prediction steps (mode generation 140 and final rate-distortion analysis 150 in FIG. 2) to determine the overall "best" inter-prediction.

[0062] In an alternative embodiment, the CBT and EPZS motion vector candidates 255 and 265 in FIG. 4 may be supplemented by additional candidates, including (but not limited to) random motion vectors, the (0, 0) motion vector, and the so-called "median predictor." The random motion vector may have fine motion estimation 130 applied to it to find the best candidate in its local neighborhood. The (0, 0) motion vector is one of the initial candidates in EPZS, but it is not always selected after EPZS candidate filtering (240 in FIG. 4), and even if it selected after candidate filtering, fine motion estimation 130 may result in a motion vector other than (0, 0). Explicitly including the (0, 0) motion vector (with no accompanying

fine motion estimation) as a candidate for final rate-distortion analysis ensures that at least one low-magnitude, "low-motion" candidate is considered. Similarly, the "median predictor" is also one of the initial candidates in EPZS, but it is also not always selected after EPZS candidate filtering (240 in FIG. 4). The median predictor is defined as the median of the motion vectors previously calculated in the data blocks to the left, top, and top right of the data block currently being encoded. Explicitly including the median predictor (with no accompanying fine motion estimation) as a candidate for final rate-distortion analysis can be especially beneficial for encoding spatially homogeneous ("flat") regions of the video frame. In this alternative embodiment, then, five or more motion vector candidates may be passed forward to the remaining inter-prediction steps (mode generation 140 and final rate-distortion analysis 150 in FIG. 2), including (but not limited to) a CBT-derived motion vector, an EPZS-derived motion vector, a motion vector derived from a random motion vector, the (0, 0) motion vector, and the median predictor.

Computation of Importance Maps for Video Encoding

[0063] Perceptual statistics may be used to compute importance maps that indicate which regions of a video frame are important to the human visual system (HVS).

[0064] One example of a perceptual statistic is the so-called *temporal contrast sensitivity function* (TCSF), which models the response of the human visual system (HVS) to temporally periodic stimuli. As noted in the Background section above, the concept of the TCSF has been around since the 1950s (when it was introduced as a "temporal modulation transfer function"), but it has not been applied to video compression before. FIG. 5 shows a recent measurement of the TCSF [Wooten, B. et al., 2010, "A practical method of measuring the temporal contrast sensitivity function," *Biomedical Optical Express*, 1(1):47-58], displaying the log of the temporal contrast sensitivity as a function of the log of frequency. The measured data points (the circles in FIG. 5) are fit with a 3rd-degree polynomial (the solid line in FIG. 5), which is then used for all TCSF calculations below. The TCSF predicts that the highest response of the human visual system (HVS) is for moderate frequencies, while HVS response falls off slightly for low frequencies and rapidly for high frequencies.

[0065] Application of the TCSF to video compression requires a method of calculating temporal frequency, which is the input to the TCSF (horizontal axis in FIG. 5). One way of calculating frequency, according to an embodiment of the invention, is described in the

following. Frequency f is given by $f = v/\lambda$, where v is velocity and λ is wavelength. In one embodiment, the velocity v (in units of pixels/s) associated with the content of any data block can be calculated from the magnitude of the motion vectors generated by the encoder (e.g., 135 in FIG. 2, 215 or 225 in FIG. 3, or 255 or 265 in FIG. 4) as $v = |MV| * framerate/N$, where $|MV|$ is the magnitude of the motion vector associated with the data block, *framerate* is the number of frames per second at which the video has been generated, and N is the number of frames between the reference frame pointed to by the motion vector and the current frame.

[0066] A suitable approximation for the wavelength λ can be derived from a computation of structural similarity (SSIM) [Wang, Z. et al., 2004, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. on Image Processing*, 13(4):600-612], computed in CIE 1976 Lab colorspace [www://en.wikipedia.org/wiki/Lab_color_space]. Computation of SSIM in the Lab colorspace is described in FIG. 6. SSIM is computed between a target block 300 (the current data block to be encoded) and the reference block 310 to which its motion vector points. The video data processed by encoder is usually represented in a standard space such as YUV 420, so the next step is to convert both the target block (320) and the reference block (330) into CIE 1976 Lab space, using any of the methods commonly found in the literature. Next, the error ΔE (340) between the target block and the reference block in Lab space is computed as $\Delta E = \sqrt{(L_T - L_R)^2 + (a_T - a_R)^2 + (b_T - b_R)^2}$, where the T subscript stands for "target block" and the R subscript stands for "reference block." Finally, the SSIM 360 between the error ΔE and the zero matrix of the same dimension is computed to serve as a measure of the colorspace variation of the data. SSIM as originally defined takes on values between -1 and 1, with values of 1 indicating perfect similarity (no spatial distinction). For the purpose of converting SSIM to wavelength λ, one can use spatial dissimilarity $DSSIM = (1 - SSIM)/2$, which takes on values between 0 and 1, where 0 corresponds to small wavelengths (maximum spatial similarity) and 1 corresponds to large wavelengths (minimum spatial similarity). To convert SSIM to units of pixels, one can multiply the value of SSIM to the number of pixels in the block for which it is calculated. In one embodiment, the SSIM block size is 8×8, so the DSSIM value is multiplied by 64. The final calculation for frequency is thus given by $f = |MV| * framerate/[N * 64 * (1 - SSIM)/2]$.

[0067] Once the frequency is calculated for a given target block, the TCSF value for that block can be determined from the curve fit (solid line) in FIG. 5. The TCSF takes on values between 0 and 1.08 in log10 scale or between 1 and 11.97 on an absolute scale. With different blocks in a frame taking on different TCSF values, the aggregate set of TCSF values over all the blocks in a frame forms an *importance map*, with high values indicating blocks that are perceptually important from a temporal contrast perspective and low values indicating blocks that are perceptually unimportant.

[0068] In a further embodiment, the values of the TCSF from recent frames may be averaged for each data block to prevent the TCSF-based importance map from fluctuating too much from frame to frame. For example, one such calculation of the average TCSF, $TCSF_{avg}$, might be $TCSF_{avg} = 0.7 \cdot TCSF_{cur} + 0.3 \cdot TCSF_{prev}$, where $TCSF_{cur}$ is the TCSF value from the current frame and $TCSF_{prev}$ is the TCSF value from the most recently encoded previous frame. The TCSF calculation is more robust when averaged in this way.

[0069] In a further embodiment, information about the relative quality of the motion vectors generated by the encoder can be computed at different points in the encoding process and then used to generate a *true motion vector map* (TMVM) that outputs, for each data block, how reliable its motion vector is. The true motion vector map, which takes on values of 0 or 1, can then be used as a mask to refine the TCSF, such that the TCSF is not used for data blocks whose motion vectors are not accurate (i.e., whose TMVM values are 0).

[0070] In one embodiment, motion vector accuracy can be determined by estimating a global motion model for a given video frame, applying the motion model to each of the data blocks in the frame to determine a global motion vector for each data block, and then comparing the global motion vector with the encoder's motion vector for that data block. Global motion may be estimated from the aggregate set of encoding motion vectors from the frame, fitted to either a six-parameter or eight-parameter affine motion model. If the global motion vector and encoder motion vector for a given data block are the same (or similar), the encoder motion vector is deemed accurate (and TMVM=1 for that data block). If the two vectors are not the same, one can compare their prediction errors (measured in terms of sum of square error [SSE] or sum of absolute difference [SAD]). If one of the errors is low and the other is high, the motion vector whose error is low is used for encoding and deemed accurate (TMVM=1).

[0071] In an alternative embodiment, the magnitude of the difference between the global motion vector and encoder motion vector for a given data block is used to identify that the

data block is *foreground* data, meaning that the content in the data block is moving differently than the rest of the frame (the background). In this embodiment, the TMVM is set to 1 – and the TCSF is applied – *only* for foreground data. In a further embodiment, for data blocks that are identified as foreground data, the encoder motion vector is subtracted from the global motion vector to obtain a differential motion vector, and it is the magnitude of the differential motion vector (not the encoder motion vector) that is used to calculate frequency for the TCSF (see the expression above, substituting |DMV| for |MV|, where DMV = differential motion vector).

[0072] In another embodiment, *motion vector symmetry* may be used to refine the TMVM. Motion vector symmetry [Bartels, C. and de Haan, G., 2009, "Temporal symmetry constraints in block matching," *Proc. IEEE 13th Int'l. Symposium on Consumer Electronics,* pp. 749-752] is defined as the relative similarity of pairs of counterpart motion vectors when the temporal direction of the motion estimation is switched, is a measure of the quality of calculated motion vectors (the higher the symmetry, the better the motion vector quality). The "symmetry error vector" is defined as the difference between the motion vector obtained through forward-direction motion estimation and the motion vector obtained through backward-direction motion estimation. Low motion vector symmetry (a large symmetry error vector) is often an indicator of the presence of complex phenomena such as occlusions (one object moving in front of another, thus either covering or revealing the background object), motion of objects on or off the video frame, and illumination changes, all of which make it difficult to derive accurate motion vectors.

[0073] In one embodiment, low symmetry is declared when the symmetry error vector is larger in magnitude than half the extent of the data block being encoded (e.g., larger in magnitude than an (8, 8) vector for a 16×16 macroblock). In another embodiment, low symmetry is declared when the symmetry error vector is larger in magnitude than a threshold based on motion vector statistics derived during the tracking process, such as the mean motion vector magnitude plus a multiple of the standard deviation of the motion vector magnitude in the current frame or some combination of recent frames. In one embodiment, data blocks whose motion vectors have low symmetry as defined above are automatically assigned a TMVM value of 0, while other data blocks retain their previous TMVM value from comparison of the global motion vector with the encoder motion vector.

[0074] Blocks that are flat, while having high spatial contrast sensitivity, tend to give rise to unreliable motion vectors because of the well-known aperture problem (see

http://en.wikipedia.org/wiki/Motion_perception#The_aperture_problem) in calculating motion vectors. Flat blocks may be detected, for example, using an edge detection process (where a flat block would be declared if no edges are detected in a data block) or by comparing the variance of a data block to a threshold (low variance less than the threshold would indicate a flat block). In one embodiment, block flatness may be used to modify the TMVM calculated as above. For example, a block may be reassigned a TMVM value of 0 if it is detected as a flat block.

[0075] In one embodiment, the TMVM may be used as a mask to refine the TCSF, which depends on having reliable motion vectors. Since the TMVM has values of 0 or 1, block-by-block multiplication of the TMVM value for a block with the TCSF value for that block has the effect of masking the TCSF. For blocks where the TMVM value is 0, the TCSF is "turned off," since the motion vector the TCSF relies on for its calculation is unreliable. For blocks where the TMVM value is 1, the TCSF calculation is considered reliable and used with confidence in any of the ways described above.

[0076] In another set of embodiments, spatial contrast maps can be generated instead of, or in addition to, the temporal contrast map (the TCSF as described above).

In the present invention, simple metrics are used to measure spatial contrast, the opposite of which is termed here "spatial complexity." In one embodiment, *block variance*, measured for both the luma and chroma components of the data, is used to measure the spatial complexity of a given input block. If an input block has high variance, it is thought to be spatially complex and less noticeable to the HVS, and thus it has low spatial contrast.

[0077] In another embodiment, *block luminance*, measured for the luma component of the data, is used to refine the variance measurement of spatial complexity. If an input block has low variance (low spatial complexity, high spatial contrast) but is either very bright or very dark, the block is automatically considered to have low spatial contrast, overriding its previously-measured high spatial contrast. The reason for this is that very dark and very bright regions are not noticeable to the HVS. The luma thresholds for classifying a block as very bright or very dark are application specific, but typical values for 8-bit video are "above 170" for very bright and "below 60" for very dark.

[0078] Block variance, modified by block luminance as described above, may be calculated for all the input blocks of a video frame to form a spatial contrast map (SCM) that indicates regions of high and low noticeability to the HVS in terms of spatial contrast.

[0079] In one embodiment, the SCM can be combined with the TCSF (refined by the TMVM) to form a unified importance map. The unified map may be formed, for example, by block-by-block multiplication of the SCM value for a block with the TCSF value for that block, with both the SCM and TCSF appropriately normalized. In another embodiment, the SCM may be used in place of the TCSF. In another embodiment, the SCM may be used to refine the TCSF. For example, in a block of high complexity, the SCM value may override the TCSF value for that block, whereas in a block of low complexity, the TCSF value for that block may be used directly.

Application of Importance Maps for Video Encoding

[0080] Importance maps as described above may be applied to the video encoding process to enhance the quality of encoded bitstreams, either for general encoders (FIG. 2) or for the CBT encoder (FIG. 3).

[0081] FIG. 7 depicts the general application of importance maps to video encoding. The input video frame 5 and frame store 85 are used to generate perceptual statistics 390 that are then applied to form importance maps 400 as described above, the TCSF (refined by the TMVM) and/or the SCM. The perceptual statistics 390 may include (but are not limited to) motion vector magnitudes, block variance, block luminance, edge detection, and global motion model parameters. The input video frame 5 and frame store 85 are also inputted as usual to the encoding of the video frame in 450, which includes the usual encoding steps (in FIG. 2, motion estimation 15, inter-prediction 20, intra-prediction 30, transform and quantization 60, and entropy encoding 90). In FIG. 7, however, the encoding 450 is enhanced by the importance maps 400, as described below.

[0082] FIG. 8A depicts the specific application of importance maps to enhance video encoding using the CBT. FIG. 8A shows initial motion estimation (110 in FIG. 2) via the frame-to-frame tracking 210 and continuous tracking 220 steps from CBT. Fine motion estimation 130 is then applied to the global CBT motion vectors 225, with the same fine motion estimation steps of local search and subpixel refinement (250 in FIG. 4). This is again followed by a mode generation module 140 that generates a set of candidate predictions 145 based on the possible encoding modes of the encoder. As in FIG. 4, EPZS and other non-model-based candidates such as the (0, 0) motion vector and the median predictor may also be generated in parallel as part of a unified motion estimation framework (these other candidates are not shown in FIG. 8A to simplify the diagram). Returning to FIG. 8A, the full

set of prediction candidates 145, including all encoding modes for CBT candidates and possibly all encoding modes for other, non-model-based candidates, again undergoes "final" rate-distortion analysis 155 to determine the best single candidate. In "final" rate-distortion analysis, a precise rate-distortion metric $D+\lambda R$ is used, computing the prediction error D for the distortion portion and the actual encoding bits R (from the entropy encoding 90 in FIG. 1) for the rate portion. The final prediction 160 (or 40 in FIG. 1) is passed to the subsequent steps of the encoder, along with its motion vector and other encoding parameters.

[0083] In FIG. 8A, perceptual statistics 390 can be calculated from the motion vectors derived from frame-to-frame motion tracking 210 and then applied to form importance maps 400 as described above, which are then inputted into the final rate-distortion analysis 155. Again, the perceptual statistics 390 may include (but are not limited to) motion vector magnitudes, block variance, block luminance, edge detection, and global motion model parameters.

[0084] In one embodiment, importance maps are used to modify the rate-distortion optimization criterion accordingly. In a standard encoder (see FIG. 2), the full set of prediction candidates 145 for a given input block 10 undergoes "final" rate-distortion analysis 150 to determine the best single candidate. In "final" rate-distortion analysis, a precise rate-distortion metric $D+\lambda R$ is used, computing the prediction error D for the distortion portion and the actual encoding bits R (from the entropy encoding 90 in FIG. 1) for the rate portion. The candidate with the lowest score for the rate-distortion metric $D+\lambda R$ becomes the final prediction 160 for the given input block 10. In one embodiment of the invention, for the perceptually-optimized encoders of FIGS. 7 or 8, the importance map IM is calculated in 400 and the final rate-distortion analysis 155 uses a modified rate-distortion metric $D \cdot IM+\lambda R$. In the modified rate-distortion metric, the IM value for a given input block multiplies the distortion term, assigning more importance to low-distortion solutions the higher the IM value is, since a high IM value indicates that the corresponding input block is perceptually important. The importance map may include the TCSF (possibly refined by the TMVM), the SCM, or a composite of both.

[0085] In a further embodiment to the above, the distortion D in the rate distortion metric may be computed as a weighted sum of SSE (sum of squared errors, the "standard" method calculating distortion) and SSIM, calculated in YUV space. The weighting γ can be computed adaptively so that the average SSIM value over the first few (or most recent few)

frames of the video, $SSIM_{avg}$, equals the average SSE value over the first few (or most recent few) frames of the video, SSE_{avg}: $\gamma \cdot SSIM_{avg} = SSE_{avg}$. For each input block, the modified rate-distortion metric would then be $(SSE + \gamma \cdot SSIM) \cdot IM + 2\lambda R$, where the multiple of 2 in front of the λR term accounts for the fact that there are two distortion terms. The inclusion of SSIM in the distortion measurements provides further accounting for HVS perception in the rate-distortion optimization, as SSIM accounts for structural information in the data.

[0086] In another set of embodiments, importance maps (e.g., the TCSF with TMVM refinement and the SCM) may be used to modify the block quantization of the encoder in addition to (or instead of) modifying the rate-distortion optimization. Quantization controls the relative quality at which a given data block is encoded; highly-quantized data results in poorer quality encoded output, while less-quantized data results in higher quality encoded output. The amount of quantization is controlled by a quantization parameter, QP. Standard encoders assign different QP values QP_{frame} to different frame types, with I-frames being encoded with the smallest QP (highest quality), B-frames being encoded with the highest QP (lowest quality), and P-frames being encoded with an intermediate QP (intermediate quality).

[0087] The above technique then represents a method of encoding a plurality of video frames having nonoverlapping target blocks, by using importance maps to modify the quantization (and thus affecting the encoding quality) of each target block in each video frame. The importance maps may be configured using temporal information (the TCSF with TMVM refinement), spatial information, or a combination of the two (i.e., a unified importance map). Because the importance maps indicate which parts of each video frame are most noticeable to human perception, the importance map values should modify the QP for each target block as follows: (i) for blocks where the importance maps take on high values, the block QP is reduced relative to QP_{frame}, resulting in higher quality for those blocks; (ii) for blocks where the importance maps take on low values, the block QP is increased relative to the frame quantization parameter QP_{frame}, resulting in lower quality for those blocks.

[0088] FIG. 8B shows an example process for using importance maps 400 to modify quantization during encoding. At 400, importance maps may be configured/created using temporal information and/or spatial information derived from perceptual statistics 390. Temporal information, for instance, may be provided by a temporal contrast sensitivity function (TCSF) that indicates which target blocks are most temporally noticeable to a human

observer and a true motion vector map (TMVM) that indicates which target blocks correspond to foreground data, with the TCSF only considered valid for those target blocks identified as foreground data. Spatial information, for instance, may be provided by a rule-based spatial complexity map (SCM).

[0089] The importance maps 400 are then used to modify the quantization step 430 within the encoding 450, as described above. In blocks where the importance maps take on high values, the block quantization parameter (QP) is reduced relative to the frame quantization parameter QP_{frame}, resulting in higher encoding quality for those blocks. In blocks where the importance maps take on low values, the block quantization parameter is increased relative to the frame quantization parameter QP_{frame}, resulting in lower encoding quality for those blocks. By using the information from the importance maps, quantization may be modified in a way that improves the encoding quality of each target block to be encoded in each of the video frames.

[0090] In one embodiment, the TCSF map for a given frame can be used to adjust the frame QP on a block-by-block basis. One method of calculating the block QP, QP_{block}, is to relate the adjustment to the full TCSF map in the frame, following the method of [Li, Z. et al, 2011, "Visual attention guided bit allocation in video compression, J. of Image and Vision Computing, 29(1):1-14]. The resulting equation is given by $QP_{block} = [TCSF_{frame}/(TCSF_{block} \times M)] \cdot QP_{frame}$, where $TCSF_{frame}$ is the sum of TCSF values for all blocks in the frame, $TCSF_{block}$ is the TCSF value for the given block, QP_{frame} is the frame QP, and M is the number of blocks in the frame. In a further embodiment, the multiplication factor $[TCSF_{frame}/(TCSF_{block} \times M)]$ may be scaled to prevent the final values of QP_{block} from becoming too high or too low relative to QP_{frame}.

[0091] In an alternative embodiment, the block-by-block adjustment of the QP via the TCSF map can be accomplished without reference to the full TCSF map for the frame. In this embodiment, the calculation of QP_{block} is simpler: $QP_{block} = QP_{frame}/TCSF_{block}$. In one embodiment, the resulting value of QP_{block} is clipped so that it does not exceed a predetermined maximum or minimum QP value for the frame: $QP_{min} \leq QP_{block} \leq QP_{max}$.

[0092] In another embodiment, the outputs of the SCM may be used to modify the quantization parameter on a block-by-block basis using a rule-based approach. This embodiment begins by assigning blocks with high variance a high QP value (low quality),

because highly-complex regions are less noticeable to the HVS. Blocks with low variance are assigned a low QP value (high quality), because less-complex regions are more noticeable to the HVS. In one embodiment, the QP assignment for a given block is bounded by the frame's maximum and minimum QP values, QP_{max} and QP_{min}, and is scaled linearly based on the block variance relative to the variance of other blocks in the frame. In an alternative embodiment, only those blocks having variance *higher* than the average variance of the entire frame are assigned QP values between the frame QP, QP_{frame}, and QP_{max}, with the assignment scaled linearly such that $QP_{block} = [(var_{block} - var_{frame})/var_{block}]*(QP_{max} - QP_{frame}) + QP_{frame}$. In this alternative embodiment, the QP assignment for high-variance blocks may be further refined by the TCSF. For example, if the block is considered a foreground data in the TMVM and the TCSF has a log contrast sensitivity value (vertical axis in FIG. 5) less than 0.5, meaning that the block is temporally unimportant, QP_{block} is raised by 2. In an alternative embodiment, an edge detection process can be applied and blocks containing edges can have their QPs adjusted to QP_{min}, overwriting the previously-assigned QPs from spatial complexity, because edges are particularly noticeable to the HVS. In a further embodiment, blocks that are either very bright or very dark can have their QPs adjusted to QP_{max}, again by overwriting the previously-assigned QPs from variance and (if applicable) from edge detection, because very dark or very bright regions are not noticeable to the HVS. This process is known as *luminance masking*.

[0093] In a further embodiment to the above, the value of QP_{max} for high-variance blocks may be determined dynamically based on the quality level of the encoded video. The idea is that low-quality encodings cannot afford any quality drop in high-variance blocks, so QP_{max} should be closer to QP_{frame}, whereas high-quality encodings can afford an increased QP_{max} for high-variance blocks, to save bits. The quality of the encoding may be updated at each I (Intra) frame by calculating the average SSIM of blocks having variance within 5% of the average frame variance, with higher SSIM values corresponding to greater values of QP_{max}. In an alternative embodiment, the average SSIM is adjusted by the average variance of the frame, so that the quality indicator is calculated as the product of the average SSIM and the average frame variance.

[0094] In a further embodiment to the above, very-low-variance blocks (corresponding to flat regions, which are especially visible to the HVS), may be assigned fixed, low QP values to ensure high-quality encoding in those regions. For example, for I (Intra) frames, blocks with variance between 0 and 10 may be assigned QP=28, blocks with variance between 10

and 30 may be assigned QP=30, and blocks with variance between 30 and 60 may be assigned QP=32. QP assignments for blocks in P and B frames may then be derived from the above QPs using the ipratio and pbratio parameters.

[0095] In a further embodiment to the above, low variance blocks (for example, those having variance between 60 and the average frame variance) are assigned the frame QP, QP_{frame} and then examined to determine whether further quality enhancement is needed. In one embodiment, one can detect *blockiness* artifacts by comparing the spatial complexity and luminance of both the *reconstructed* pixels and the *original* pixels from the current (target) block being encoded with the spatial complexity and luminance of previously-encoded surrounding blocks (e.g., blocks to the left, top-left, top, and top-right when available). If there is a large difference between the spatial complexity and luminance measures of the *reconstructed* pixels of the target block and the corresponding measures of neighboring blocks, but there is no such difference in spatial complexity and luminance between the *original* pixels of the target block and that of the neighboring blocks, then the target block is considered "blocky." In this case, the block's QP value is decreased (e.g., decreased by 2) to improve the encoding quality of the block. In another embodiment, the estimated quality of the target block is calculated by averaging the SSIM and QP values of previously-encoded surrounding blocks (e.g., blocks to the left, top-left, right, and top-right when available). The average QP value, QP_{avg}, is the estimated QP, QP_{block}, for the target block. If the average SSIM value, $SSIM_{est}$, is lower than 0.9, $QP_{block} = QP_{avg}$ is lowered by 2, increasing its quality. In a further embodiment, if the target block is identified as foreground data by the TMVM, then QP_{block} is lowered by 2 *only* if the TCSF has a log contrast sensitivity value (vertical axis in FIG. 5) greater than 0.8, meaning that the block is temporally important.

[0096] The methods outlined above may use temporal importance maps (the TCSF, with or without TMVM refinement), spatial importance maps (the SCM), or both. If both temporal and spatial importance maps are used, the result is termed *a unified importance map*.

[0097] Importance maps, generated from perceptual statistics as described above, can be applied to any video compression framework that uses motion compensation to produce motion vectors, such that both rate-distortion analysis and quantization are enhanced to produce visually superior encodings for the same encoding sizes. The use of importance maps for video compression does not require specific application to the continuous block tracker (CBT) as detailed above. However, the CBT provides the additional capability of

accurately determining which motion vectors are true motion vectors, so importance maps are more *effective* in a CBT-based encoding framework. The particular reason for this is that the CBT's frame-to-frame motion vectors (from frame-to-frame tracking 210 in FIG. 8A) are generated from the *original* frames of the video and not the reconstructed frames. The frame store 85 in FIG. 2 and FIG. 7 for general encoders contains reconstructed frames generated from the encoding process, but the frame store 205 in FIG. 3, FIG. 4, and FIG. 8A contains the original video frames. Because of this, the CBT's frame-to-frame tracking (210 in FIGS. 3, 4, and 8) is better able to track the true motion of the video, and its frame-to-frame motion vectors generate more accurate true motion vector maps. By contrast, a general encoder's motion vectors are selected to optimize rate-distortion (compression) performance and may not reflect the true motion of the video.

[0098] It should also be noted that importance maps, once generated, may be applied to intra-predicted frames as well, either by modifying the rate-distortion optimization among intra-prediction modes or by modifying the block-level quantization, following the techniques described above. For all-intra encoders, however, computation of the TCSF requires a separate encoding module (such as frame-to-frame tracking 210 in FIG. 8A) to generate motion vectors for each data block in the video frame.

Digital Processing Environment

[0099] Example implementations of the present invention may be implemented in a software, firmware, or hardware environment. FIG. 9A illustrates one such environment. Client computer(s)/devices 950 (e.g., mobile phones or computing devices) and a cloud 960 (or server computer or cluster thereof) provide processing, storage, encoding, decoding, and input/output devices executing application programs and the like.

[00100] Client computer(s)/devices 950 can also be linked through communications network 970 to other computing devices, including other client devices/processes 950 and server computer(s) 960. Communications network 970 can be part of a remote access network, a global network (e.g., the Internet), a worldwide collection of computers, Local area or Wide area networks, and gateways that currently use respective protocols (TCP/IP, Bluetooth, etc.) to communicate with one another. Other electronic devices/computer network architectures are suitable.

[00101] Embodiments of the invention may include means for encoding, tracking, modeling, filtering, tuning, decoding, or displaying video or data signal information. FIG. 9B

is a diagram of the internal structure of a computer/computing node (e.g., client processor/device/mobile phone device/tablet 950 or server computers 960) in the processing environment of FIG. 9A, which may be used to facilitate encoding such videos or data signal information. Each computer 950, 960 contains a system bus 979, where a bus is a set of actual or virtual hardware lines used for data transfer among the components of a computer or processing system. Bus 979 is essentially a shared conduit that connects different elements of a computer system (e.g., processor, encoder chip, decoder chip, disk storage, memory, input/output ports, etc.) that enables the transfer of data between the elements. Attached to the system bus 979 is an I/O device interface 982 for connecting various input and output devices (e.g., keyboard, mouse, displays, printers, speakers, etc.) to the computer 950, 960. Network interface 986 allows the computer to connect to various other devices attached to a network (for example, the network illustrated at 970 of FIG. 9A). Memory 990 provides volatile storage for computer software instructions 992 and data 994 used to implement a software implementation of the present invention (e.g., codec: encoder/decoder).

[00102] Disk storage 995 provides non-volatile storage for computer software instructions 998 (equivalently "OS program") and data 994 used to implement an embodiment of the present invention: it can also be used to store the video in compressed format for long-term storage. Central processor unit 984 is also attached to system bus 979 and provides for the execution of computer instructions. Note that throughout the present text, “computer software instructions” and "OS program” are equivalent.

[00103] In one example, an encoder may be configured with computer readable instructions 992 to encode video data using importance maps formed from temporal information or spatial information. The importance maps may be configured to provide a feedback loop to an encoder (or elements thereof) to optimize the encoding/decoding of video data.

[00104] In one embodiment, the processor routines 992 and data 994 are a computer program product, with an encoder (generally referenced 992), including a computer readable medium capable of being stored on a storage device 994 which provides at least a portion of the software instructions for the encoder.

[00105] The computer program product 992 can be installed by any suitable software installation procedure, as is well known in the art. In another embodiment, at least a portion of the encoder software instructions may also be downloaded over a cable, communication, and/or wireless connection. In other embodiments, the encoder system software is a

computer program propagated signal product 907 (in Fig. 9A) embodied on a nontransitory computer readable medium, which when executed can be implemented as a propagated signal on a propagation medium (e.g., a radio wave, an infrared wave, a laser wave, a sound wave, or an electrical wave propagated over a global network such as the Internet, or other network(s)). Such carrier media or signals provide at least a portion of the software instructions for the present invention routines/program 992.

[00106] In alternate embodiments, the propagated signal is an analog carrier wave or digital signal carried on the propagated medium. For example, the propagated signal may be a digitized signal propagated over a global network (e.g., the Internet), a telecommunications network, or other network. In one embodiment, the propagated signal is transmitted over the propagation medium over a period of time, such as the instructions for a software application sent in packets over a network over a period of milliseconds, seconds, minutes, or longer. In another embodiment, the computer readable medium of computer program product 992 is a propagation medium that the computer system 950 may receive and read, such as by receiving the propagation medium and identifying a propagated signal embodied in the propagation medium, as described above for the computer program propagated signal product.

[00107] While this invention has been particularly shown and described with references to example embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the scope of the invention encompassed by the appended claims.

CLAIMS

What is claimed is:

1. A method of encoding a plurality of video frames having non-overlapping target blocks, the method comprising:

encoding the plurality of video frames using importance maps, such that the importance maps modify quantization affecting the encoding quality of each target block to be encoded in each video frame, the importance maps being formed by:

configuring the importance maps using temporal information and spatial information; and

computationally causing the importance maps indicate which parts of a video frame in the plurality of video frames which are most noticeable to human perception, such that: (i) in blocks where the importance maps take on high values, the block quantization parameter (QP) is reduced relative to the frame quantization parameter QP_{frame}, resulting in higher quality for those blocks, and (ii) in target blocks where the importance maps take on low values, the block quantization parameter is increased relative to the frame quantization parameter QP_{frame}, resulting in lower quality for those blocks.

2. The method as in Claim 1, wherein the spatial information is provided by a rule-based spatial complexity map (SCM) in which the initial step is to determine which target blocks in the frame have higher variance than the average block variance in the frame, var_{frame}; and

for such blocks, to assign a QP value higher than the frame quantization parameter QP_{frame}, with the block QP assignment QP_{block} scaled linearly between QP_{frame} and the maximum quantization parameter QP_{max}, based on how much higher the block variance var_{block} is than var_{frame}.

3. The method as in Claim 1, wherein the temporal information is provided by

a temporal contrast sensitivity function (TCSF) that indicates which target blocks are most temporally noticeable to a human observer and

a true motion vector map (TMVM) that indicates which target blocks correspond to foreground data,

with the TCSF only considered valid for those target blocks identified as foreground data.

4. The method as in Claim 2, wherein a high-variance block has its block QP assignment QP_{block} further refined by the TCSF and TMVM, such that if the TMVM identifies a target block as foreground data and the TCSF has a log contrast sensitivity value less than 0.5 for that block, QP_{block} is raised by 2.

5. The method as in Claim 2, wherein the SCM further includes luminance masking, in which target blocks that are either very bright (luminance above 170) or very dark (luminance below 60) have their block quantization parameters QP_{block} adjusted back to QP_{max} .

6. The method as in Claim 2, wherein the SCM further includes dynamic determination of QP_{max} based on the quality level of the encoded video,

where quality is measured using an average structural similarity (SSIM) calculation of target blocks in Intra (I) frames, together with the average block variance var_{frame} of such frames;

such that when the measured quality is low, the value of QP_{max} is lowered to something closer to QP_{frame}.

7. The method as in Claim 2, wherein very-low-variance blocks are assigned fixed, low QP values QP_{block} to ensure high-quality encoding in those regions, such that the lower the block variance, the lower value of QP_{block} (and the higher the quality).

8. The method as in Claim 7, wherein the assignment of low QP values QP_{block} for very-low-variance blocks is fixed first for I frames and then determined for P and B frames using the ipratio and pbratio parameters.

9. The method as in Claim 7, wherein blocks that are low-variance but do not qualify as very-low-variance are examined to determine whether quality enhancement is needed for those blocks;

such that an initial estimate of the block QP, QP_{block} , is calculated by average the QP values of neighboring, already encoded blocks to the left, top-left, right, and top-right of the current block;

an estimate of the SSIM of the current block, $SSIM_{est}$, is calculated from the SSIM values of neighboring, already-encoded blocks to the left, top-left, right, and top-right of the current block;

and the value of QP_{block} is lowered by 2 if $SSIM_{est}$ is lower than 0.9.

10. The method as in Claim 9, wherein the quality enhancement is only applied to those blocks that are identified as foreground data by the TMVM and for which the TCSF has log contrast sensitivity value greater than 0.8.

11. The method as in Claim 3, wherein the temporal frequency of the TCSF is computed by using SSIM in the colorspace domain between the target block and its reference block to approximate wavelength and by using motion vector magnitudes and the framerate to approximate velocity.

12. The method as in Claim 3, wherein the TCSF is calculated over multiple frames, such that the TCSF for the current frame is a weighted average of the TCSF maps over recent frames, with more recent frames receiving higher weighting.

13. The method as in Claim 3, wherein the TMVM is set to 1 only for foreground data.

14. The method as in Claim 13, wherein foreground data is identified by computing the difference between the encoder motion vector for a given target block and the global motion vector for that block, such that blocks with sufficiently large differences are determined to be foreground data.

15. The method as in Claim 14, wherein for data blocks that are identified as foreground data, the encoder motion vector is subtracted from the global motion vector to obtain a differential motion vector, and it is the magnitude of the differential motion vector that is used in calculating the temporal frequency of the TCSF.

16. The method as in Claim 3, wherein the TCSF is computed from motion vectors from an encoder.

17. The method as in Claim 1, wherein if the importance map is configured with the temporal information and spatial information, the importance map is a unified importance map.

18. A system of encoding video data, the system comprising:

a codec using importance maps to encode a plurality of video frames having non-overlapping target blocks; and

the importance maps configured to modify quantization affecting the encoding quality of each target block to be encoded in each video frame, the importance maps being formed by:

configuring the importance maps using temporal information and spatial information, where an importance map that is configured with the temporal information and spatial information is a unified importance map; and

computationally causing the importance maps to indicate parts of a video frame in the plurality of video frames which are most noticeable to human perception, such that: (i) in blocks where the importance maps take on high values, the block quantization parameter (QP) is reduced relative to the frame quantization parameter QP_{frame}, resulting in higher quality for those blocks, and (ii) in target blocks where the importance maps take on low values, the block quantization parameter is increased relative to the frame quantization parameter QP_{frame}, resulting in lower quality for those blocks.

19. The encoder as in Claim 18, wherein the spatial information is provided by a rule-based spatial complexity map (SCM) in which the initial step is to determine which target blocks in the frame have higher variance than the average block variance in the frame, var_{frame}; and

for such blocks, to assign a QP value higher than the frame quantization parameter QP_{frame}, with the block QP assignment QP_{block} scaled linearly between QP_{frame} and the maximum quantization parameter QP_{max}, based on how much higher the block variance var_{block} is than var_{frame}.

20. The encoder as in Claim 18, wherein the temporal information is provided by

a temporal contrast sensitivity function (TCSF) that indicates which target blocks are most temporally noticeable to a human observer and

a true motion vector map (TMVM) that indicates which target blocks correspond to foreground data,

with the TCSF only considered valid for those target blocks identified as foreground data.

21. The encoder as in Claim 19, wherein a high-variance block has its block QP assignment QP_{block} further refined by the TCSF and TMVM, such that if the TMVM identifies a target block as foreground data and the TCSF has a log contrast sensitivity value less than 0.5 for that block, QP_{block} is raised by 2.

22. The encoder as in Claim 19, wherein the SCM further includes luminance masking, in which target blocks that are either very bright (luminance above 170) or very dark (luminance below 60) have their block quantization parameters QP_{block} adjusted back to QP_{max} .

23. The encoder as in Claim 19, wherein the SCM further includes dynamic determination of QP_{max} based on the quality level of the encoded video,

where quality is measured using an average structural similarity (SSIM) calculation of target blocks in Intra (I) frames, together with the average block variance var_{frame} of such frames;

such that when the measured quality is low, the value of QP_{max} is lowered to something closer to QP_{frame}.

24. The encoder as in Claim 19, wherein very-low-variance blocks are assigned fixed, low QP values QP_{block} to ensure high-quality encoding in those regions, such that the lower the block variance, the lower value of QP_{block} (and the higher the quality).

25. The encoder as in Claim 24, wherein the assignment of low QP values QP_{block} for very-low-variance blocks is fixed first for I frames and then determined for P and B frames using the ipratio and pbratio parameters.

26. The system as in Claim 19, wherein blocks that are low-variance but do not qualify as very-low-variance are examined to determine whether quality enhancement is needed for those blocks;

such that an initial estimate of the block QP, QP_{block} , is calculated by average the QP values of neighboring, already encoded blocks to the left, top-left, right, and top-right of the current block;

an estimate of the SSIM of the current block, $SSIM_{est}$, is calculated from the SSIM values of neighboring, already-encoded blocks to the left, top-left, right, and top-right of the current block;

and the value of QP_{block} is lowered by 2 if $SSIM_{est}$ is lower than 0.9.

27. The system as in Claim 26, wherein the quality enhancement is only applied to those blocks that are identified as foreground data by the TMVM and for which the TCSF has log contrast sensitivity value greater than 0.8.

28. The system as in Claim 20, wherein the temporal frequency of the TCSF is computed by using SSIM in the colorspace domain between the target block and its reference block to approximate wavelength and by using motion vector magnitudes and the framerate to approximate velocity.

29. The system as in Claim 20, wherein the TCSF is calculated over multiple frames, such that the TCSF for the current frame is a weighted average of the TCSF maps over recent frames, with more recent frames receiving higher weighting.

30. The system as in Claim 20, wherein the TMVM is set to 1 only for foreground data.

31. The system as in Claim 30, wherein foreground data is identified by computing the difference between the encoder motion vector for a given target block and the global motion vector for that block, such that blocks with sufficiently large differences are determined to be foreground data.

32. The system as in Claim 20, wherein for data blocks that are identified as foreground data, the encoder motion vector is subtracted from the global motion vector to obtain a differential motion vector, and it is the magnitude of the differential motion vector that is used in calculating the temporal frequency of the TCSF.

33. The system as in Claim 20, wherein the TCSF is computed from motion vectors from the encoder.

34. The system as in Claim 18, wherein if the importance map is configured with the temporal information and spatial information, the importance map is a unified importance map.

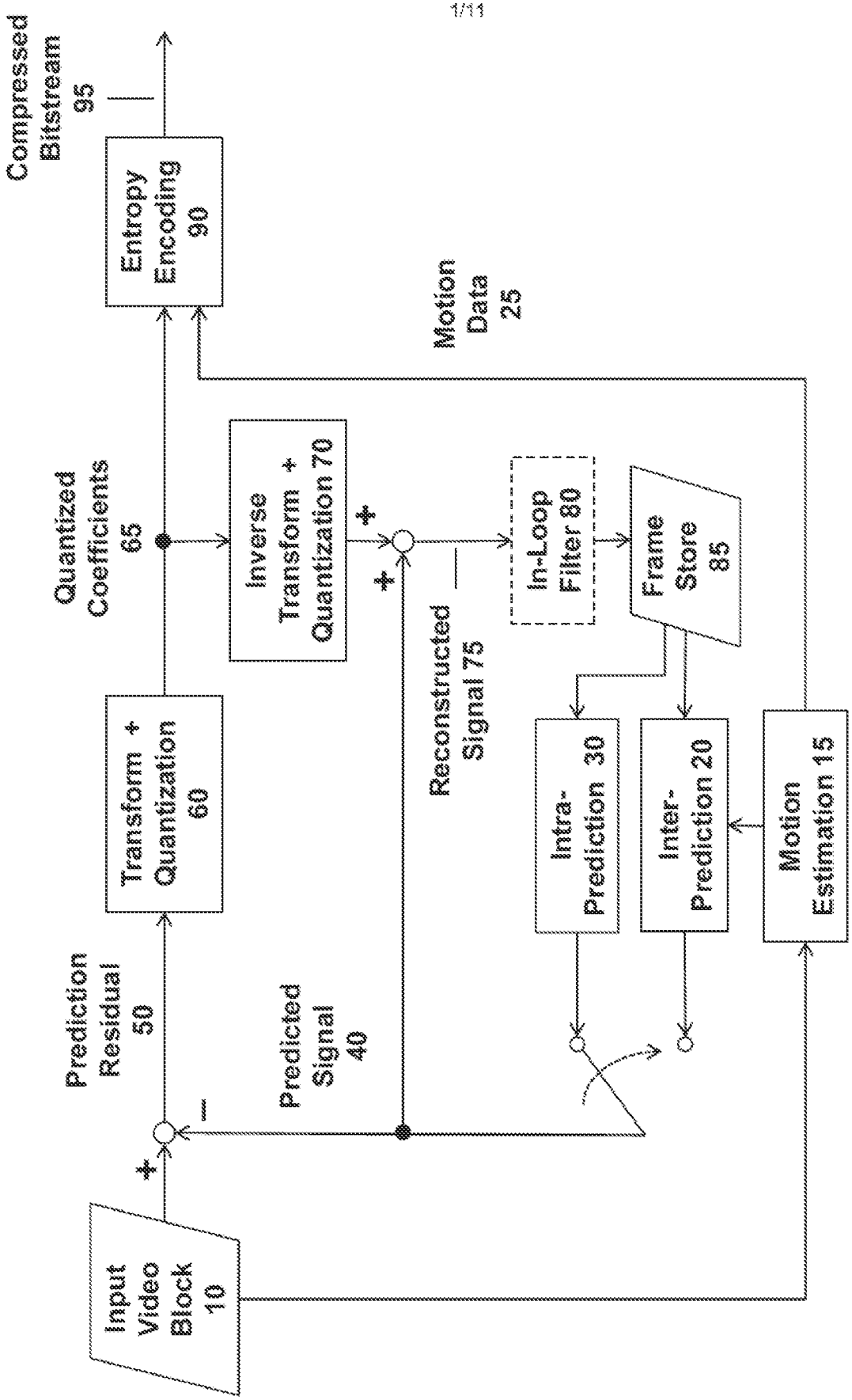

Figure 1: Basic encoder configuration

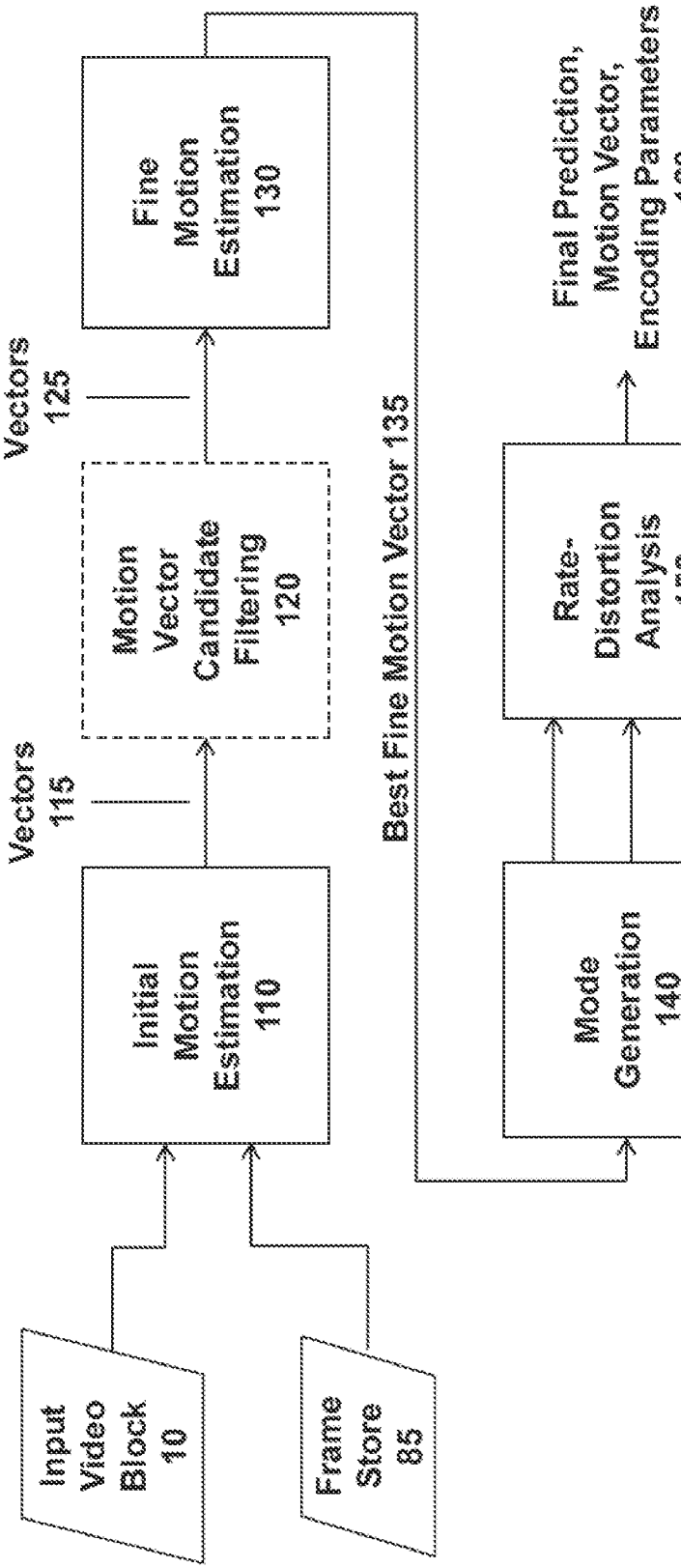

Figure 2: Inter-prediction steps

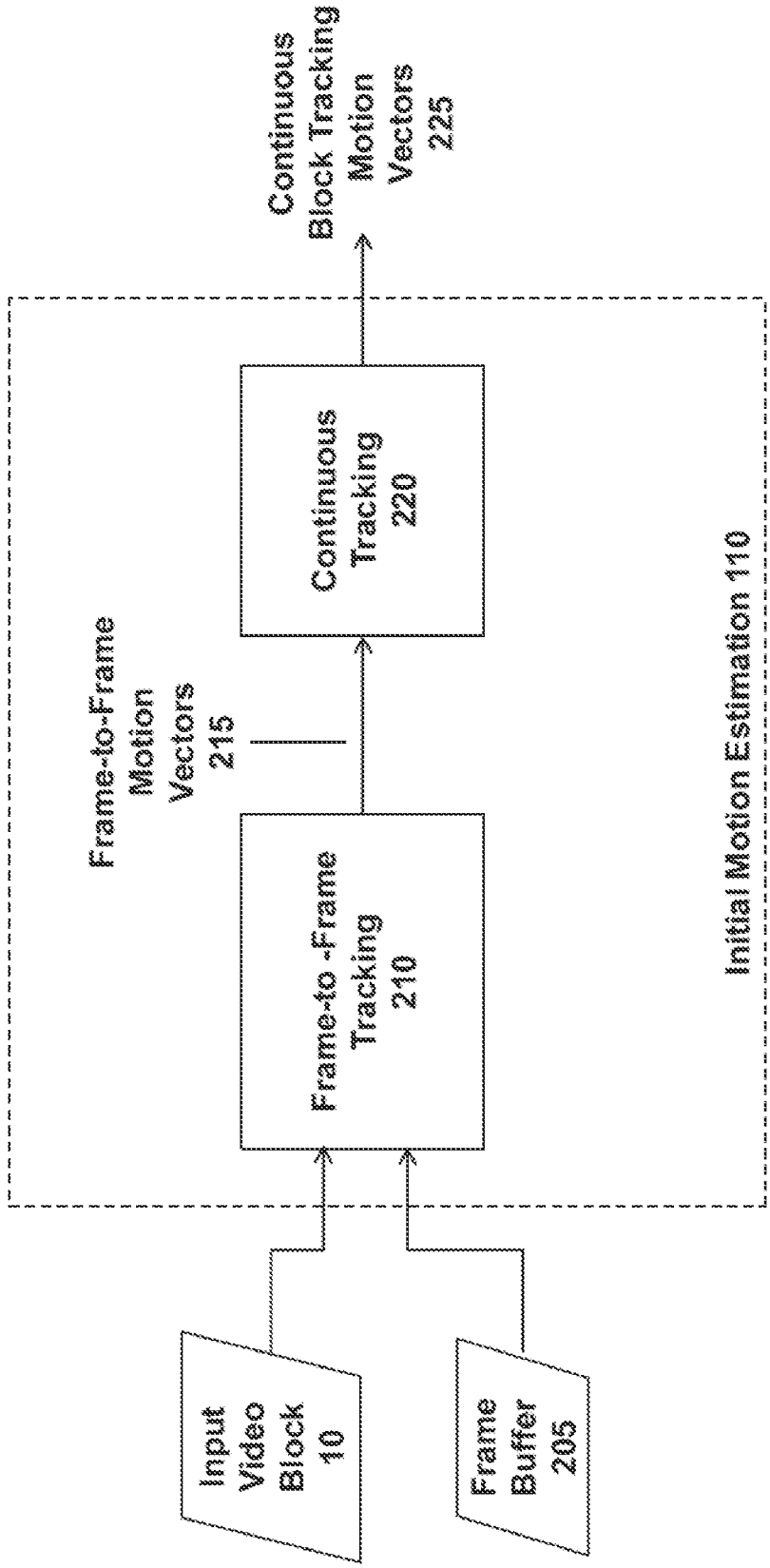

Figure 3: Initial motion estimation via continuous block tracking

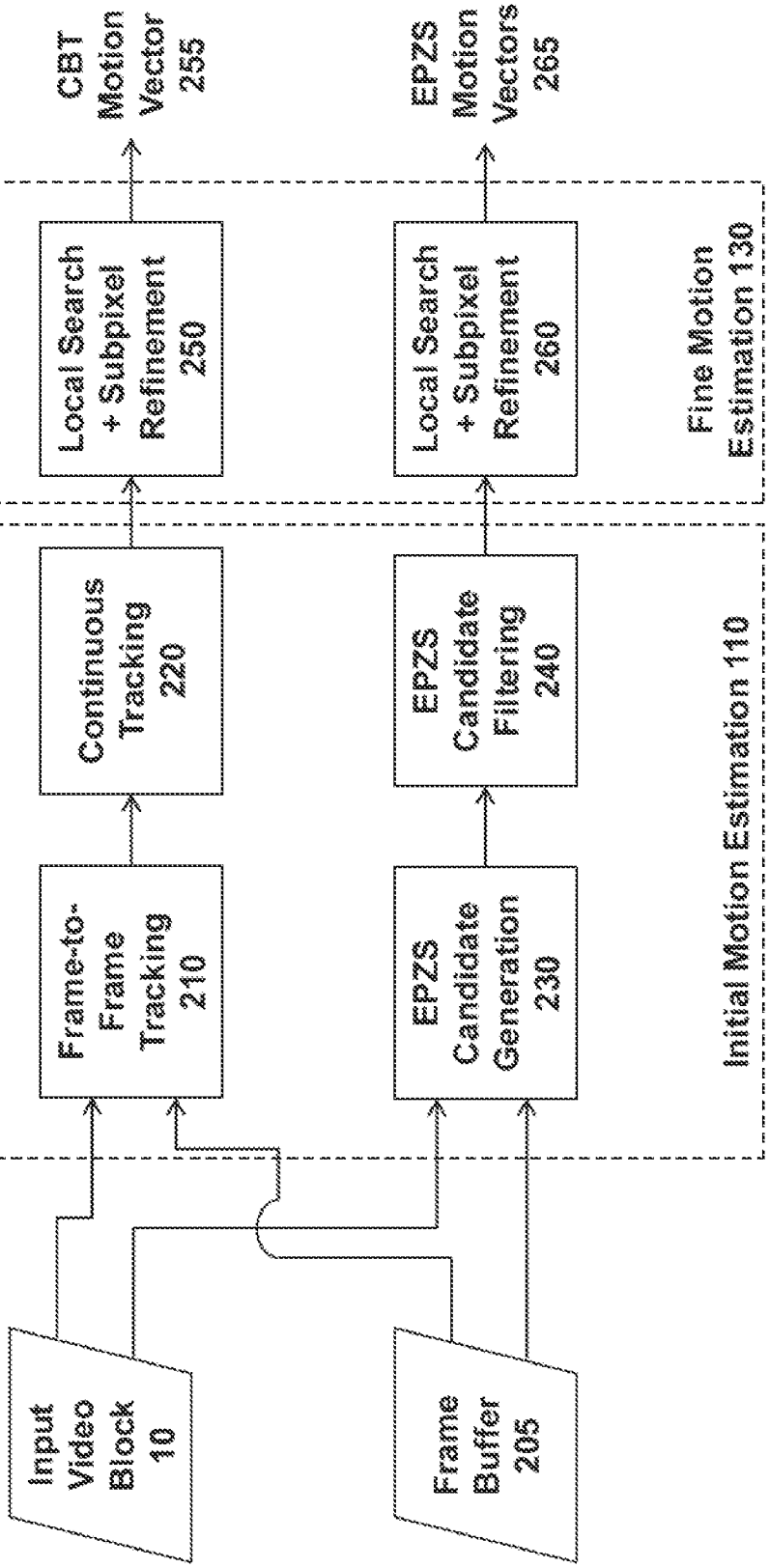

Figure 4: Unified motion estimation via combination of continuous block tracking (CBT) and enhanced predictive zonal search (EPZS)

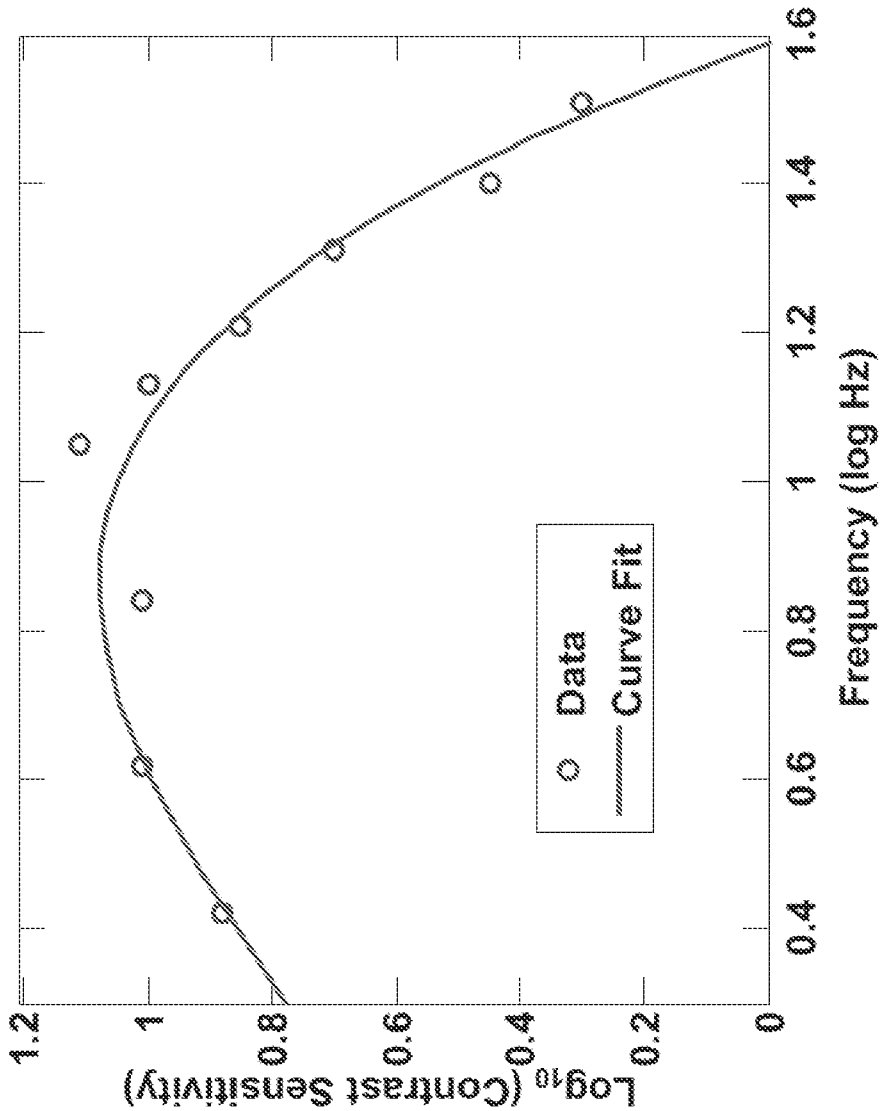

Figure 5: Temporal contrast sensitivity function, parafoveal target [Wooten, 2010]

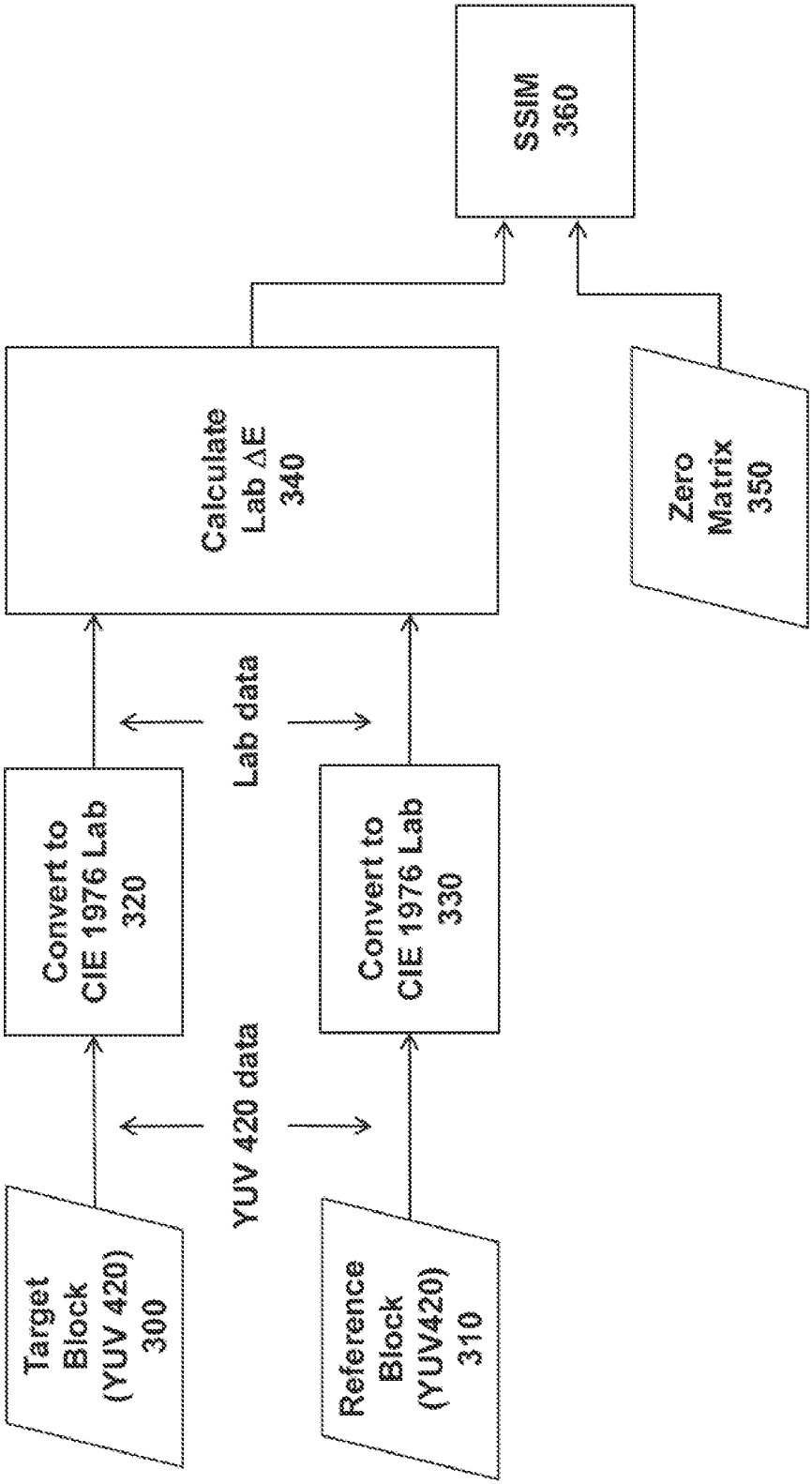

Figure 6: Calculation of SSIM in CIE 1976 Lab colorspace

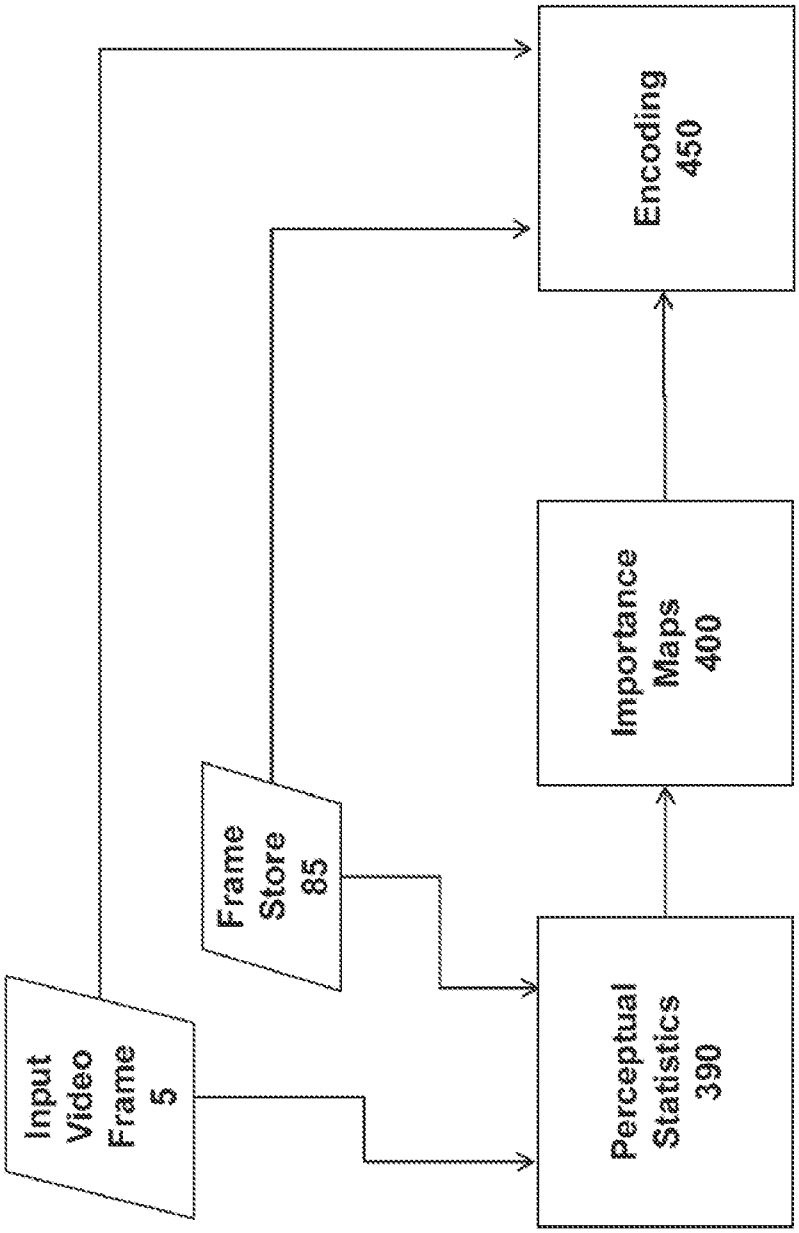

Figure 7: General application of importance maps to video encoding

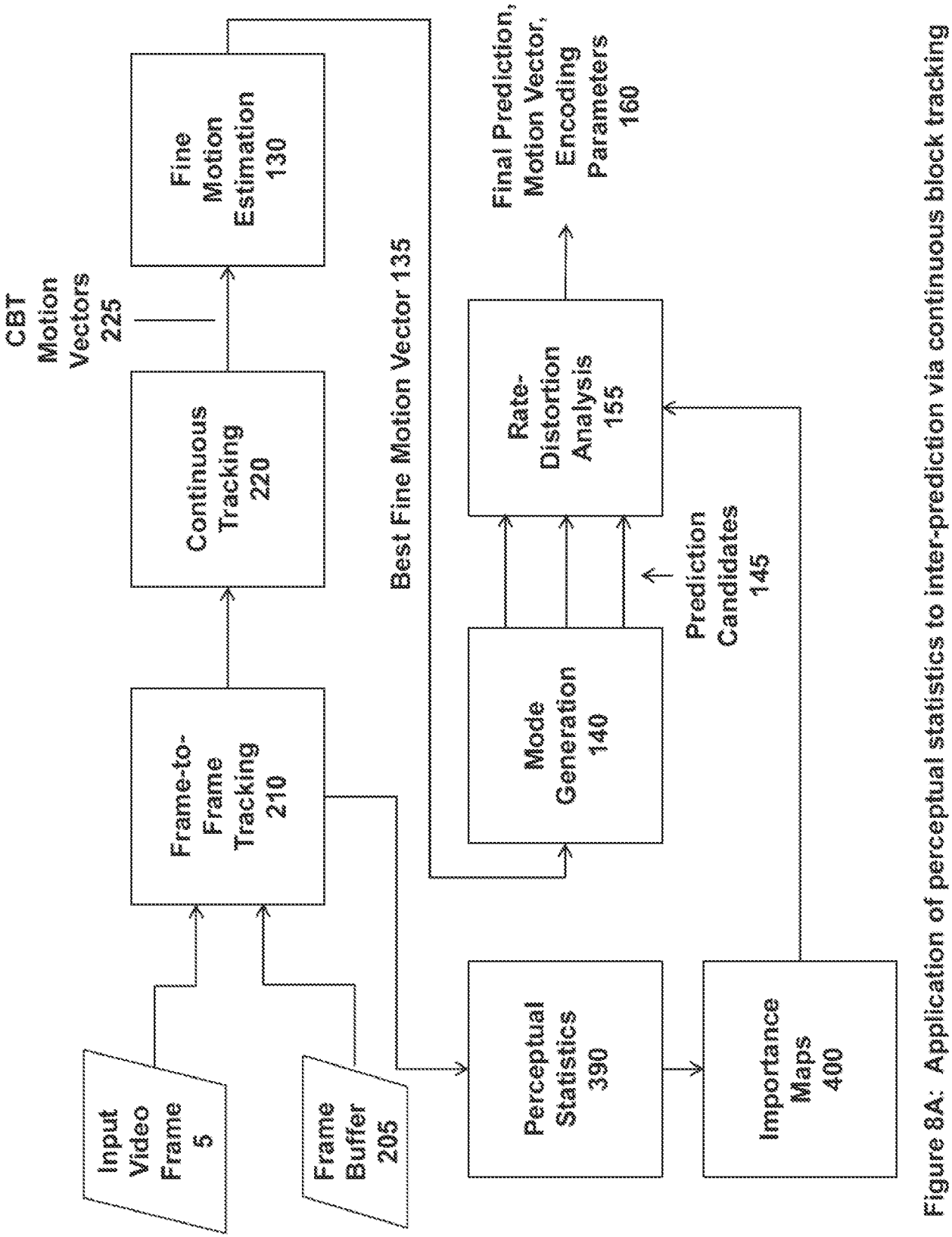

Figure 8A: Application of perceptual statistics to inter-prediction via continuous block tracking

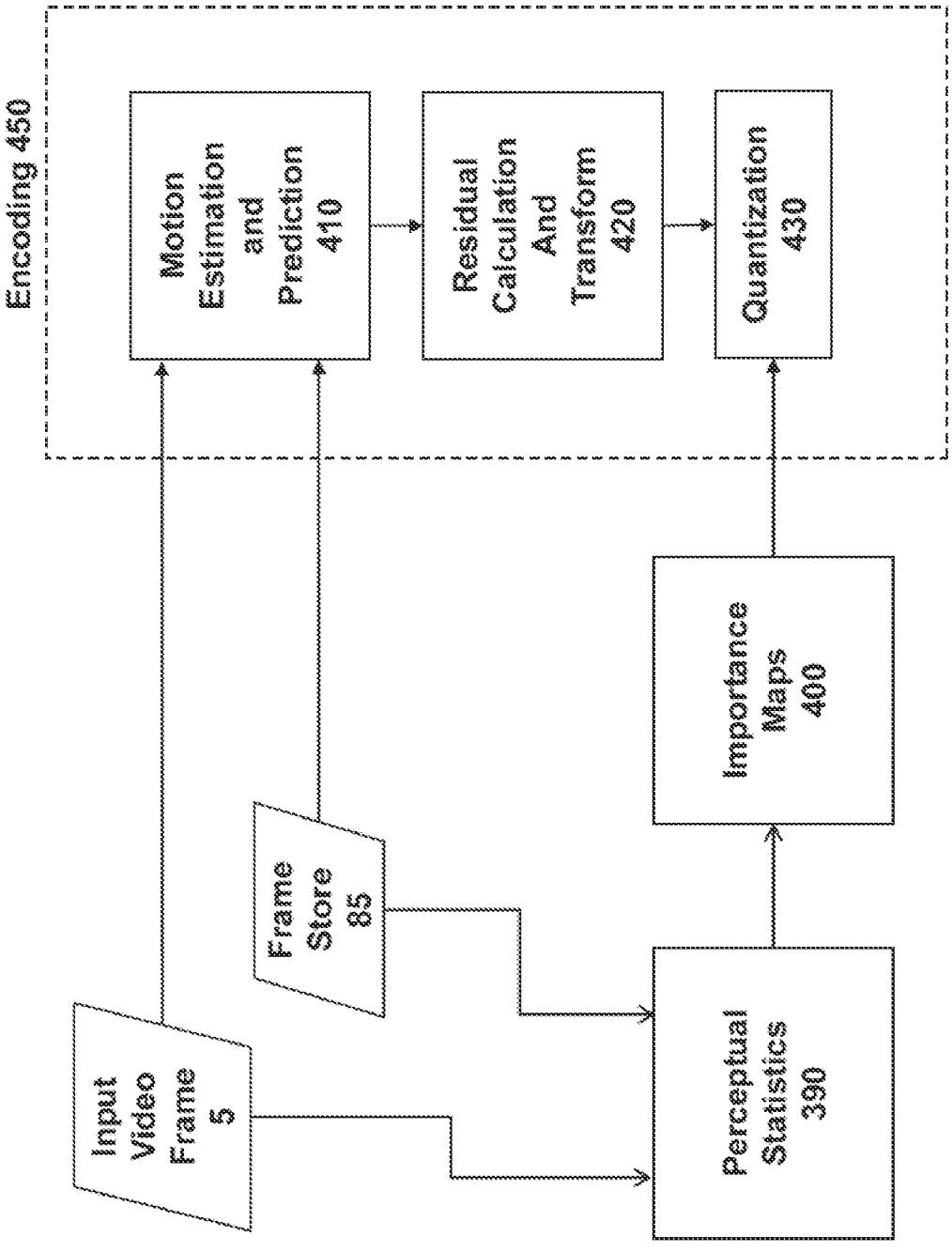

Figure 8B: Application of importance maps to modify quantization in video encoding

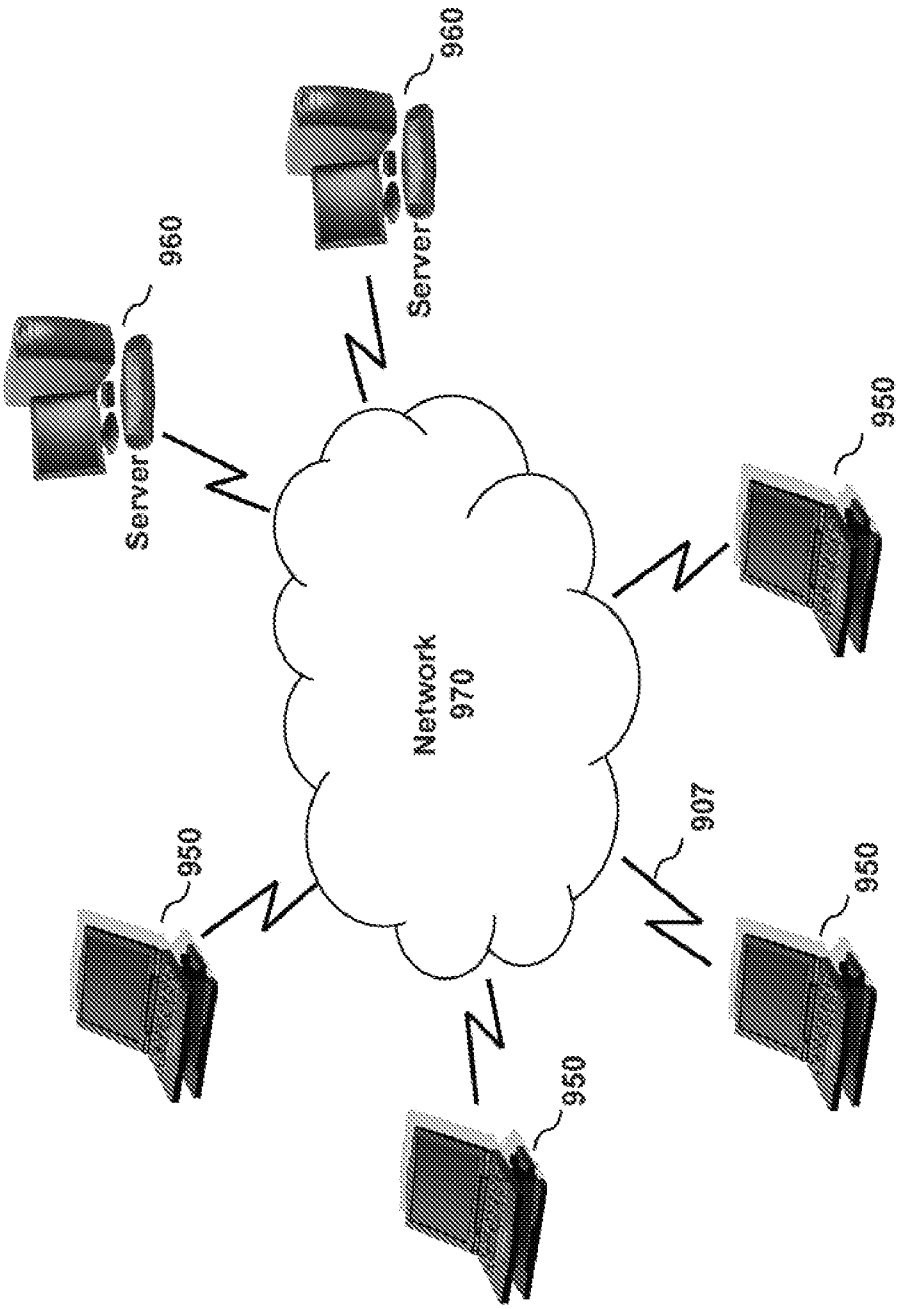

Figure 9A

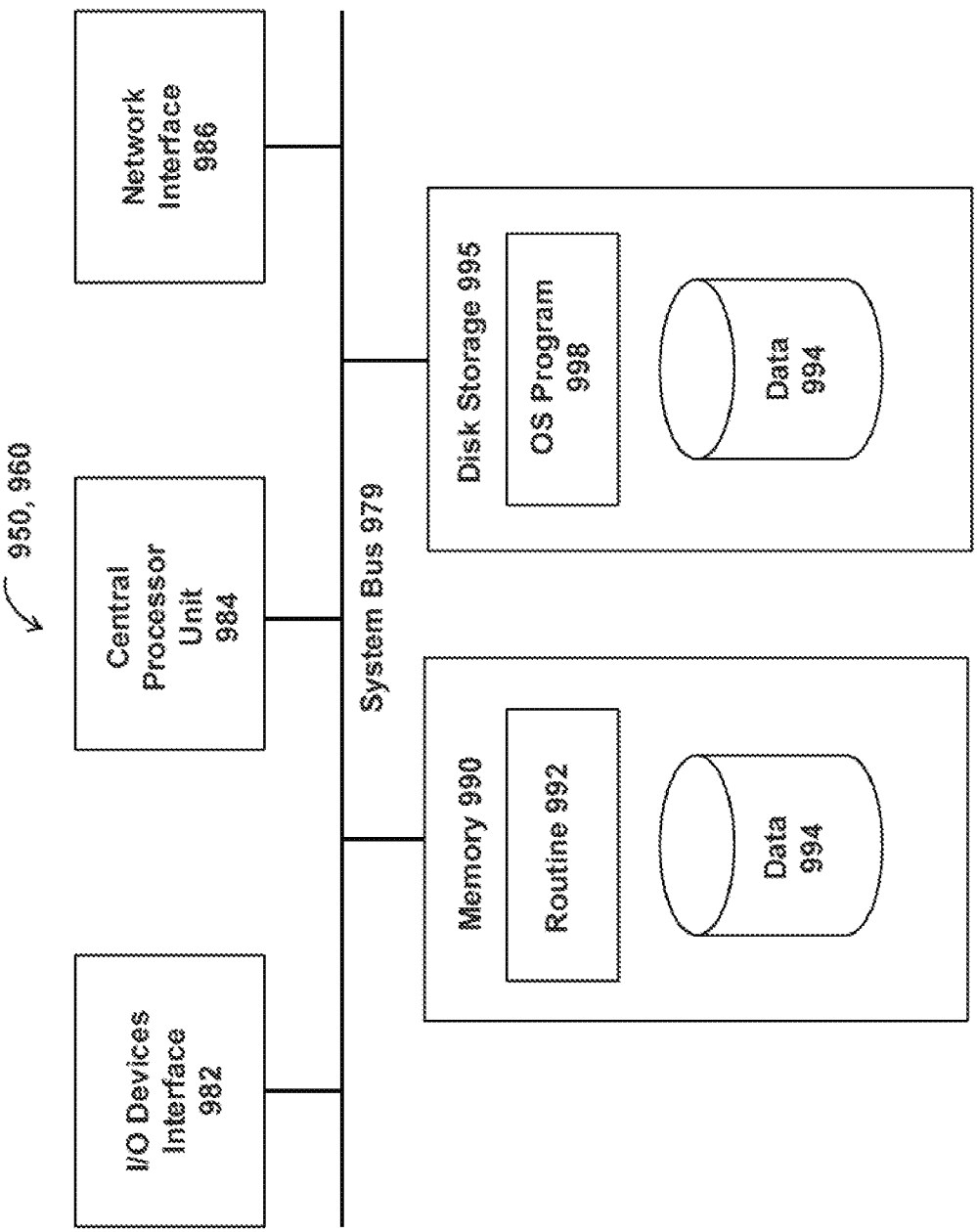
950, 960
Network Interface 986
Central Processor Unit 984
I/O Devices Interface 982
System Bus 979
Disk Storage 995
OS Program 998
Data 994
Memory 990
Routine 992
Data 994

Figure 9B

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2015/048353

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N19/139 H04N19/176 H04N19/124 H04N19/136 H04N19/14
H04N19/137 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, WPI Data, COMPENDEX, INSPEC, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ZHICHENG LI ET AL: "Visual attention guided bit allocation in video compression", IMAGE AND VISION COMPUTING, vol. 29, no. 1, 1 January 2011 (2011-01-01), pages 1-14, XP055126506, ISSN: 0262-8856, DOI: 10.1016/j.imavis.2010.07.001 cited in the application	1,17,18, 34
A	* sections 1, 2 and 5 *	2,4-10, 19,21-27

-/--

[X] Further documents are listed in the continuation of Box C.

[X] 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
1 December 2015	17/02/2016

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	Ferré, Pierre

1

Form PCT/ISA/210 (second sheet) (April 2005)

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	ANONYMOUS: "The H.264 Advanced Video Compression Standard, 2nd Edition, chapter 7, H.264 transform and coding, Iain E. Richardson", NOT KNOWN,, 20 April 2010 (2010-04-20), XP030001638, * equation 7.17 and table 7.3 *	1,18
X	US 8 135 062 B1 (COTE GUY [US]) 13 March 2012 (2012-03-13)	1,17,18, 34
A	figure 4 column 4, line 4 - line 37 column 5, line 15 - column 10, line 27	2,4-10, 19,21-27
X	US 8 737 464 B1 (ZHANG HUIPIN [US] ET AL) 27 May 2014 (2014-05-27)	1,2,4, 6-10, 17-19, 21, 23-27,34
Y	column 1, line 19 - line 30 column 3, line 44 - column 8, line 13 figures 3a,3b, 4	5,22
X	EP 1 250 012 A2 (SHARP KK [JP]) 16 October 2002 (2002-10-16) paragraph [0025] paragraph [0031] paragraph [0035] - paragraph [0051] figure 1	1,17,18, 34
Y	"Lumi masking", Wikipedia, 8 November 2006 (2006-11-08), XP055232466, Retrieved from the Internet: URL:https://web.archive.org/web/20061124153834/http://en.wikipedia.org/wiki/Lumi_masking [retrieved on 2015-12-01]	5,22
A	the whole document	1,2,4, 6-10, 17-19, 21, 23-27,34
X	CHRISTOPHER BULLA ET AL: "High Quality Video Conferencing: Region of Interest Encoding and Joint Video/Audio Analysis", INTERNATIONAL JOURNAL ON ADVANCES IN TELECOMMUNICATIONS, vol. 6, no. 3-4, 1 December 2013 (2013-12-01), pages 153-163, XP055232071, ISSN: 1942-2601 * sections II.A and II.B *	1,17,18, 34

-/--

1

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2015/048353

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CHIH-WEI TANG: "Spatiotemporal Visual Considerations for Video Coding", IEEE TRANSACTIONS ON MULTIMEDIA, IEEE SERVICE CENTER, PISCATAWAY, NJ, US, vol. 9, no. 2, 1 February 2007 (2007-02-01), pages 231-238, XP011346384, ISSN: 1520-9210, DOI: 10.1109/TMM.2006.886328 * sections III and IV *	1,17,18, 34
X	CHEN ZHENZHONG ET AL: "Perception-oriented video coding based on foveated JND model Â", PICTURE CODING SYMPOSIUM 2009; 6-5-2009 - 8-5-2009; CHICAGO,, 6 May 2009 (2009-05-06), XP030081866, * section 2 *	1,17,18, 34
A	NACCARI M ET AL: "Improving HEVC compression efficiency by intensity dependant spatial quantisation", 101. MPEG MEETING; 16-7-2012 - 20-7-2012; STOCKHOLM; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11),, no. m25398, 11 July 2012 (2012-07-11), XP030053732, the whole document	1,2, 4-10, 17-19, 21-27,34
A	US 2010/290524 A1 (LU XIAOAN [US] ET AL) 18 November 2010 (2010-11-18) paragraph [0042] - paragraph [0000]	1,2, 4-10, 17-19, 21-27,34

1

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/048353

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

1, 2, 4-10, 17-19, 21-27, 34

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1, 2, 4-10, 17-19, 21-27, 34

   How to derive spatial information and how to modify the QP using this spatial information

   ---

2. claims: 3, 11-16, 20, 28-33

   How to derive temporal information

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2015/048353

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 8135062 B1	13-03-2012	US 8135062 B1	13-03-2012
		US 2013121403 A1	16-05-2013
US 8737464 B1	27-05-2014	NONE	
EP 1250012 A2	16-10-2002	CN 1378384 A	06-11-2002
		EP 1250012 A2	16-10-2002
		JP 2002335527 A	22-11-2002
		US 2002181583 A1	05-12-2002
US 2010290524 A1	18-11-2010	NONE	