



(51) International Patent Classification:

H04N 19/186 (2014.01) H04N 19/593 (2014.01)

H04N 19/176 (2014.01) H04N 19/11 (2014.01)

H04N 19/196 (2014.01)

(21) International Application Number:

PCT/FI2022/050735

(22) International Filing Date:

08 November 2022 (08.11.2022)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

20216170 15 November 2021 (15.11.2021) FI

(71) Applicant: NOKIA TECHNOLOGIES OY [FI/FI];
Karakaari 7, 02610 Espoo (FI).(72) Inventors: ASTOLA, Pekka; Järvensivuntie 5 A 21,
33100 Tampere (FI). LAINEMA, Jani; Kisakentäkatu 12
B 6, 33230 Tampere (FI).(74) Agent: NOKIA TECHNOLOGIES OY et al.; Mikko
Alatossava, IPR Department, Karakaari 7, 02610 Espoo
(FI).

(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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, CV, 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, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(54) Title: AN APPARATUS, A METHOD AND A COMPUTER PROGRAM FOR VIDEO CODING AND DECODING

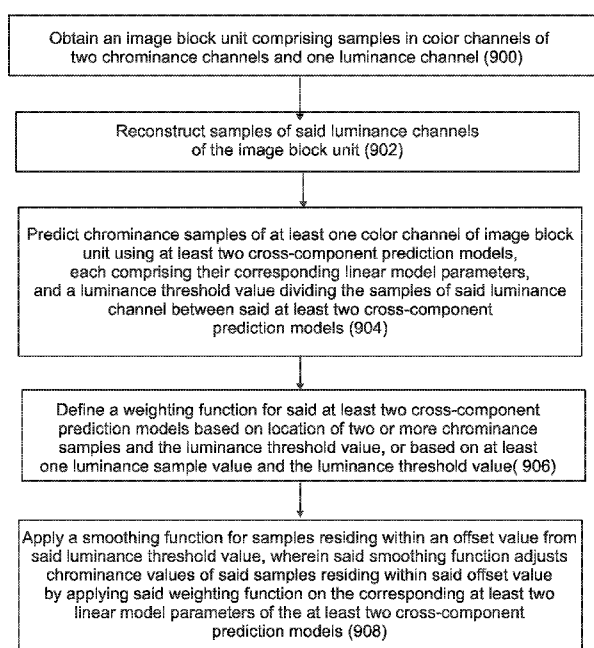


Fig. 9

(57) Abstract: A method comprising: obtaining an image block unit comprising samples in color channels of one or two chrominance channels and one luminance channel (900); reconstructing samples of said luminance channel of the image block unit (902); predicting chrominance samples of at least one color channel of image block unit using at least two cross-component prediction models, each comprising their corresponding linear model parameters, and a luminance threshold value dividing the samples of said luminance channel between said at least two cross-component prediction models (904); defining a weighting function for said at least two cross-component prediction models based on location of said two or more chrominance samples and the luminance threshold value (906); and applying a smoothing function for samples residing within an offset value from said luminance threshold value, wherein said smoothing function adjusts chrominance values of said samples residing within said offset value by applying said weighting function on the corresponding at least two linear model parameters of the at least two cross-component prediction models (908).

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))*

AN APPARATUS, A METHOD AND A COMPUTER PROGRAM FOR VIDEO CODING AND DECODING

TECHNICAL FIELD

[0001] The present invention relates to an apparatus, a method and a computer program for video coding and decoding.

BACKGROUND

[0002] In video coding, video and image samples are typically encoded using color representations such as YUV or YCbCr consisting of one luminance (luma) and two chrominance (chroma) channels. In these cases the luminance channel, representing mostly the illumination of the scene, is typically coded at certain resolution, while the chrominance channels, representing typically differences between certain color components, are often coded at a second resolution lower than that of the luminance signal. The intention of this kind of a differential representation is to decorrelate the color components and be able to compress the data more efficiently.

[0003] In Versatile Video Coding (VVC/H.266) standard, a Cross-Component Linear Model (CCLM) is used as a linear model for predicting the samples in the chroma channels (e.g. Cb and Cr). The CCLM may be extended by adding three Multi-model LM (MMLM) modes, wherein in each MMLM mode, the reconstructed neighbouring samples are classified into two classes using a threshold which is the average of the luma reconstructed neighbouring samples. The linear model of each class is derived using the Least-Mean-Square (LMS) method.

[0004] However, the two classes of luma-to-chroma models may be significantly different from each other. Hard thresholding the co-located luma samples may create colour discrepancies, artificial edges, or other artefacts within the target chroma block due to the use of two different luma-to-chroma models.

SUMMARY

[0005] Now in order to at least alleviate the above problems, an enhanced method for achieving a better correlation between the luma and chroma channels is introduced herein.

[0006] A method according to a first aspect comprises obtaining an image block unit comprising samples in color channels of one or two chrominance channels and one luminance channel; reconstructing samples of said luminance channels of the image block unit; predicting chrominance samples of at least one color channel of image block unit using at least two cross-component prediction models, each comprising their corresponding linear model parameters, and a luminance threshold value dividing the samples of said luminance channel between said at least two cross-component prediction models; defining a weighting function for said at least two cross-component prediction models based on location of two or more chrominance samples and the luminance threshold value, or based on at least one luminance sample value and the luminance threshold value; and applying a smoothing function for samples residing within an offset value from said luminance threshold value, wherein said smoothing function adjusts chrominance values of said samples residing within said offset value by applying said weighting function on the corresponding at least two linear model parameters of the at least two cross-component prediction models.

[0007] An apparatus according to a second aspect comprises means for obtaining an image block unit comprising samples in color channels of one or two chrominance channels and one luminance channel; means for reconstructing samples of said luminance channels of the image block unit; means for predicting chrominance samples of at least one color channel of image block unit using at least two cross-component prediction models, each comprising their corresponding linear model parameters, and a luminance threshold value dividing the samples of said luminance channel between said at least two cross-component prediction models; means for defining a weighting function for said at least two cross-component prediction models based on location of two or more chrominance samples and the luminance threshold value, or based on at least one luminance sample value and the luminance threshold value; and means for applying a smoothing function for samples residing within an offset value from said luminance threshold value, wherein said smoothing function adjusts chrominance values of said samples residing within said offset value by applying said weighting function on the corresponding at least two linear model parameters of the at least two cross-component prediction models.

[0008] According to an embodiment, the apparatus comprises means for adjusting the value of the weighting function to remain within a range of [0,1].

[0009] According to an embodiment, the apparatus comprises means for means for defining the weighting function in sample domain.

[0010] According to an embodiment, the smoothing function is defined in the sample domain as

$C(x,y,L_t,L,\alpha_1,\beta_1,\alpha_2,\beta_2)=w(\lfloor x,y,L \rfloor_t)(\alpha_2 L(x,y)+\beta_2)+(1-w(x,y,L_t))(\alpha_1 L(x,y)+\beta_1)$, where w is the weighting function, x and y define the position of the sample, α_1 and β_1 are the linear model parameters for the first cross-component prediction model and α_2 and β_2 are the linear model parameters for the second cross-component prediction model, and L_t is the threshold luma value.

[0011] According to an embodiment, the weighting function w is defined in the sample domain as a linear weighting function

$$w(x,y,L_t)=\gamma \cdot (L(x,y)-L_t+R),$$

where γ is a scaling term and R is the range of the reference luma samples.

[0012] According to an embodiment, the weighting function w is defined in the sample domain as a linear weighting function

$$w(x,y,\mu,\sigma)=e^{\lfloor -0.5((L(x,y)-\mu)/\sigma) \rfloor^2},$$

where $\mu=L_t+R$ and $\sigma=\gamma R$.

[0013] According to an embodiment, the smoothing function is applied in a parametric vicinity of the threshold value L_t .

[0014] According to an embodiment, the smoothing function is applied in the sample domain in unsymmetric manner only when $L(x,y) \leq L_t+T_0$ && $L(x,y) \geq L_t-T_1$ where $T_0 \neq T_1$.

[0015] According to an embodiment, the weighting function $w(\cdot)$ is defined in spatial domain as $w(\Delta x, \Delta y)$ where Δx and Δy define distance to the nearest boundary sample (x_B, y_B) .

[0016] According to an embodiment, the smoothing function C is defined in the spatial domain as

$$C(x,y,L_t,L,\alpha_1,\beta_1,\alpha_2,\beta_2)=w(\Delta x, \Delta y)(\alpha_2 L(x,y)+\beta_2)+(1-w(\Delta x, \Delta y))(\alpha_1 L(x,y)+\beta_1).$$

[0017] According to an embodiment, the smoothing function is applied only in the vicinity of the spatial boundary of two luma-to-chroma models, when $\sqrt{(\Delta_x^2+\Delta_y^2)} \leq T$ where T is a fixed, inferred or signalled parameter.

[0018] According to an embodiment, the smoothing function is applied in the spatial domain in anisotropic manner only when $\sqrt{(\omega_0 \Delta_x)^2 + (\omega_1 \Delta_y)^2} \leq T$, where ω_0, ω_1 are fixed, inferred or signalled by the encoder.

[0019] According to an embodiment, the apparatus comprises means for deriving different linear models for a beginning and an end of a luma spectrum of the multi-model cross-component prediction model.

[0020] According to an embodiment, the apparatus comprises means for means for performing a rounding and/or a clipping operation on the chroma samples values obtained as a result of the smoothing function.

[0021] The apparatuses and the computer readable storage mediums stored with code thereon, as described above, are thus arranged to carry out the above methods and one or more of the embodiments related thereto.

BRIEF DESCRIPTION OF THE DRAWINGS

[0022] For better understanding of the present invention, reference will now be made by way of example to the accompanying drawings in which:

[0023] Figure 1 shows schematically an electronic device employing embodiments of the invention;

[0024] Figure 2 shows schematically a user equipment suitable for employing embodiments of the invention;

[0025] Figure 3 further shows schematically electronic devices employing embodiments of the invention connected using wireless and wired network connections;

[0026] Figures 4a and 4b show schematically an encoder and a decoder suitable for implementing embodiments of the invention;

[0027] Figure 5 illustrates locations of the samples used for derivation of parameters for a Cross-Component Linear Model;

[0028] Figures 6a and 6b show examples of classification of luma samples into two classes in the sample domain, and in the spatial domain, respectively;

[0029] Figure 7 illustrates an example of four reference lines neighboring to a prediction block;

[0030] Figure 8 illustrates a matrix weighted intra prediction process;

[0031] Figure 9 shows a flow chart of a method according to an embodiment of the invention;

[0032] Figures 10a and 10b show examples of defining a transition area for chroma predictions in the sample domain, and in the spatial domain, respectively, according to embodiments of the invention;

[0033] Figure 11 shows an example of extending a linear model with slope α according to an embodiment of the invention; and

[0034] Figure 12 shows a schematic diagram of an example multimedia communication system within which various embodiments may be implemented.

DETAILED DESCRIPTION OF SOME EXAMPLE EMBODIMENTS

[0035] The following describes in further detail suitable apparatus and possible mechanisms for prediction of chroma samples. In this regard reference is first made to Figures 1 and 2, where Figure 1 shows a block diagram of a video coding system according to an example embodiment as a schematic block diagram of an exemplary apparatus or electronic device 50, which may incorporate a codec according to an embodiment of the invention. Figure 2 shows a layout of an apparatus according to an example embodiment. The elements of Figs. 1 and 2 will be explained next.

[0036] The electronic device 50 may for example be a mobile terminal or user equipment of a wireless communication system. However, it would be appreciated that embodiments of the invention may be implemented within any electronic device or apparatus which may require encoding and decoding or encoding or decoding video images.

[0037] The apparatus 50 may comprise a housing 30 for incorporating and protecting the device. The apparatus 50 further may comprise a display 32 in the form of a liquid crystal display. In other embodiments of the invention the display may be any suitable display technology suitable to display an image or video. The apparatus 50 may further comprise a keypad 34. In other embodiments of the invention any suitable data or user interface mechanism may be employed. For example the user interface may be implemented as a virtual keyboard or data entry system as part of a touch-sensitive display.

[0038] The apparatus may comprise a microphone 36 or any suitable audio input which may be a digital or analogue signal input. The apparatus 50 may further comprise an audio output device which in embodiments of the invention may be any one of: an earpiece 38, speaker, or an

analogue audio or digital audio output connection. The apparatus 50 may also comprise a battery (or in other embodiments of the invention the device may be powered by any suitable mobile energy device such as solar cell, fuel cell or clockwork generator). The apparatus may further comprise a camera capable of recording or capturing images and/or video. The apparatus 50 may further comprise an infrared port for short range line of sight communication to other devices. In other embodiments the apparatus 50 may further comprise any suitable short range communication solution such as for example a Bluetooth wireless connection or a USB/firewire wired connection.

[0039] The apparatus 50 may comprise a controller 56, processor or processor circuitry for controlling the apparatus 50. The controller 56 may be connected to memory 58 which in embodiments of the invention may store both data in the form of image and audio data and/or may also store instructions for implementation on the controller 56. The controller 56 may further be connected to codec circuitry 54 suitable for carrying out coding and decoding of audio and/or video data or assisting in coding and decoding carried out by the controller.

[0040] The apparatus 50 may further comprise a card reader 48 and a smart card 46, for example a UICC and UICC reader for providing user information and being suitable for providing authentication information for authentication and authorization of the user at a network.

[0041] The apparatus 50 may comprise radio interface circuitry 52 connected to the controller and suitable for generating wireless communication signals for example for communication with a cellular communications network, a wireless communications system or a wireless local area network. The apparatus 50 may further comprise an antenna 44 connected to the radio interface circuitry 52 for transmitting radio frequency signals generated at the radio interface circuitry 52 to other apparatus(es) and for receiving radio frequency signals from other apparatus(es).

[0042] The apparatus 50 may comprise a camera capable of recording or detecting individual frames which are then passed to the codec 54 or the controller for processing. The apparatus may receive the video image data for processing from another device prior to transmission and/or storage. The apparatus 50 may also receive either wirelessly or by a wired connection the image for coding/decoding. The structural elements of apparatus 50 described above represent examples of means for performing a corresponding function.

[0043] With respect to Figure 3, an example of a system within which embodiments of the present invention can be utilized is shown. The system 10 comprises multiple communication devices which can communicate through one or more networks. The system 10 may comprise any combination of wired or wireless networks including, but not limited to a wireless cellular telephone network (such as a GSM, UMTS, CDMA network etc.), a wireless local area network (WLAN) such as defined by any of the IEEE 802.x standards, a Bluetooth personal area network, an Ethernet local area network, a token ring local area network, a wide area network, and the Internet.

[0044] The system 10 may include both wired and wireless communication devices and/or apparatus 50 suitable for implementing embodiments of the invention.

[0045] For example, the system shown in Figure 3 shows a mobile telephone network 11 and a representation of the internet 28. Connectivity to the internet 28 may include, but is not limited to, long range wireless connections, short range wireless connections, and various wired connections including, but not limited to, telephone lines, cable lines, power lines, and similar communication pathways.

[0046] The example communication devices shown in the system 10 may include, but are not limited to, an electronic device or apparatus 50, a combination of a personal digital assistant (PDA) and a mobile telephone 14, a PDA 16, an integrated messaging device (IMD) 18, a desktop computer 20, a notebook computer 22. The apparatus 50 may be stationary or mobile when carried by an individual who is moving. The apparatus 50 may also be located in a mode of transport including, but not limited to, a car, a truck, a taxi, a bus, a train, a boat, an airplane, a bicycle, a motorcycle or any similar suitable mode of transport.

[0047] The embodiments may also be implemented in a set-top box; i.e. a digital TV receiver, which may/may not have a display or wireless capabilities, in tablets or (laptop) personal computers (PC), which have hardware or software or combination of the encoder/decoder implementations, in various operating systems, and in chipsets, processors, DSPs and/or embedded systems offering hardware/software based coding.

[0048] Some or further apparatus may send and receive calls and messages and communicate with service providers through a wireless connection 25 to a base station 24. The base station 24 may be connected to a network server 26 that allows communication between the mobile

telephone network 11 and the internet 28. The system may include additional communication devices and communication devices of various types.

[0049] The communication devices may communicate using various transmission technologies including, but not limited to, code division multiple access (CDMA), global systems for mobile communications (GSM), universal mobile telecommunications system (UMTS), time divisional multiple access (TDMA), frequency division multiple access (FDMA), transmission control protocol-internet protocol (TCP-IP), short messaging service (SMS), multimedia messaging service (MMS), email, instant messaging service (IMS), Bluetooth, IEEE 802.11 and any similar wireless communication technology. A communications device involved in implementing various embodiments of the present invention may communicate using various media including, but not limited to, radio, infrared, laser, cable connections, and any suitable connection.

[0050] In telecommunications and data networks, a channel may refer either to a physical channel or to a logical channel. A physical channel may refer to a physical transmission medium such as a wire, whereas a logical channel may refer to a logical connection over a multiplexed medium, capable of conveying several logical channels. A channel may be used for conveying an information signal, for example a bitstream, from one or several senders (or transmitters) to one or several receivers.

[0051] An MPEG-2 transport stream (TS), specified in ISO/IEC 13818-1 or equivalently in ITU-T Recommendation H.222.0, is a format for carrying audio, video, and other media as well as program metadata or other metadata, in a multiplexed stream. A packet identifier (PID) is used to identify an elementary stream (a.k.a. packetized elementary stream) within the TS. Hence, a logical channel within an MPEG-2 TS may be considered to correspond to a specific PID value.

[0052] Available media file format standards include ISO base media file format (ISO/IEC 14496-12, which may be abbreviated ISOBMFF) and file format for NAL unit structured video (ISO/IEC 14496-15), which derives from the ISOBMFF.

[0053] Video codec consists of an encoder that transforms the input video into a compressed representation suited for storage/transmission and a decoder that can uncompress the compressed video representation back into a viewable form. A video encoder and/or a video decoder may also be separate from each other, i.e. need not form a codec. Typically encoder discards some

information in the original video sequence in order to represent the video in a more compact form (that is, at lower bitrate).

[0054] Typical hybrid video encoders, for example many encoder implementations of ITU-T H.263 and H.264, encode the video information in two phases. Firstly pixel values in a certain picture area (or “block”) are predicted for example by motion compensation means (finding and indicating an area in one of the previously coded video frames that corresponds closely to the block being coded) or by spatial means (using the pixel values around the block to be coded in a specified manner). Secondly the prediction error, i.e. the difference between the predicted block of pixels and the original block of pixels, is coded. This is typically done by transforming the difference in pixel values using a specified transform (e.g. Discrete Cosine Transform (DCT) or a variant of it), quantizing the coefficients and entropy coding the quantized coefficients. By varying the fidelity of the quantization process, encoder can control the balance between the accuracy of the pixel representation (picture quality) and size of the resulting coded video representation (file size or transmission bitrate).

[0055] In temporal prediction, the sources of prediction are previously decoded pictures (a.k.a. reference pictures). In intra block copy (IBC; a.k.a. intra-block-copy prediction), prediction is applied similarly to temporal prediction but the reference picture is the current picture and only previously decoded samples can be referred in the prediction process. Inter-layer or inter-view prediction may be applied similarly to temporal prediction, but the reference picture is a decoded picture from another scalable layer or from another view, respectively. In some cases, inter prediction may refer to temporal prediction only, while in other cases inter prediction may refer collectively to temporal prediction and any of intra block copy, inter-layer prediction, and inter-view prediction provided that they are performed with the same or similar process than temporal prediction. Inter prediction or temporal prediction may sometimes be referred to as motion compensation or motion-compensated prediction.

[0056] Motion compensation can be performed either with full sample or sub-sample accuracy. In the case of full sample accurate motion compensation, motion can be represented as a motion vector with integer values for horizontal and vertical displacement and the motion compensation process effectively copies samples from the reference picture using those displacements. In the case of sub-sample accurate motion compensation, motion vectors are represented by fractional or decimal values for the horizontal and vertical components of the

motion vector. In the case a motion vector is referring to a non-integer position in the reference picture, a sub-sample interpolation process is typically invoked to calculate predicted sample values based on the reference samples and the selected sub-sample position. The sub-sample interpolation process typically consists of horizontal filtering compensating for horizontal offsets with respect to full sample positions followed by vertical filtering compensating for vertical offsets with respect to full sample positions. However, the vertical processing can be also be done before horizontal processing in some environments.

[0057] Inter prediction, which may also be referred to as temporal prediction, motion compensation, or motion-compensated prediction, reduces temporal redundancy. In inter prediction the sources of prediction are previously decoded pictures. Intra prediction utilizes the fact that adjacent pixels within the same picture are likely to be correlated. Intra prediction can be performed in spatial or transform domain, i.e., either sample values or transform coefficients can be predicted. Intra prediction is typically exploited in intra coding, where no inter prediction is applied.

[0058] One outcome of the coding procedure is a set of coding parameters, such as motion vectors and quantized transform coefficients. Many parameters can be entropy-coded more efficiently if they are predicted first from spatially or temporally neighboring parameters. For example, a motion vector may be predicted from spatially adjacent motion vectors and only the difference relative to the motion vector predictor may be coded. Prediction of coding parameters and intra prediction may be collectively referred to as in-picture prediction.

[0059] Figs. 4a and 4b show an encoder and a decoder suitable for employing embodiments of the invention. A video codec consists of an encoder that transforms an input video into a compressed representation suited for storage/transmission and a decoder that can decompress the compressed video representation back into a viewable form. Typically, the encoder discards and/or loses some information in the original video sequence in order to represent the video in a more compact form (that is, at lower bitrate). An example of an encoding process is illustrated in Figure 4a. Figure 4a illustrates an image to be encoded (I^n); a predicted representation of an image block (P^n); a prediction error signal (D^n); a reconstructed prediction error signal (D^m); a preliminary reconstructed image (I^m); a final reconstructed image (R^m); a transform (T) and inverse transform (T^{-1}); a quantization (Q) and inverse quantization (Q^{-1}); entropy encoding (E);

a reference frame memory (RFM); inter prediction (P_{inter}); intra prediction (P_{intra}); mode selection (MS) and filtering (F).

[0060] An example of a decoding process is illustrated in Figure 4b. Figure 4b illustrates a predicted representation of an image block (P^m); a reconstructed prediction error signal (D^m); a preliminary reconstructed image (I^m); a final reconstructed image (R^m); an inverse transform (T^{-1}); an inverse quantization (Q^{-1}); an entropy decoding (E^{-1}); a reference frame memory (RFM); a prediction (either inter or intra) (P); and filtering (F).

[0061] Many hybrid video encoders encode the video information in two phases. Firstly pixel values in a certain picture area (or “block”) are predicted for example by motion compensation means (finding and indicating an area in one of the previously coded video frames that corresponds closely to the block being coded) or by spatial means (using the pixel values around the block to be coded in a specified manner). Secondly the prediction error, i.e. the difference between the predicted block of pixels and the original block of pixels, is coded. This is typically done by transforming the difference in pixel values using a specified transform (e.g. Discrete Cosine Transform (DCT) or a variant of it), quantizing the coefficients and entropy coding the quantized coefficients. By varying the fidelity of the quantization process, encoder can control the balance between the accuracy of the pixel representation (picture quality) and size of the resulting coded video representation (file size or transmission bitrate). Video codecs may also provide a transform skip mode, which the encoders may choose to use. In the transform skip mode, the prediction error is coded in a sample domain, for example by deriving a sample-wise difference value relative to certain adjacent samples and coding the sample-wise difference value with an entropy coder.

[0062] Entropy coding/decoding may be performed in many ways. For example, context-based coding/decoding may be applied, where in both the encoder and the decoder modify the context state of a coding parameter based on previously coded/decoded coding parameters. Context-based coding may for example be context adaptive binary arithmetic coding (CABAC) or context-based variable length coding (CAVLC) or any similar entropy coding. Entropy coding/decoding may alternatively or additionally be performed using a variable length coding scheme, such as Huffman coding/decoding or Exp-Golomb coding/decoding. Decoding of coding parameters from an entropy-coded bitstream or codewords may be referred to as parsing.

[0063] The phrase along the bitstream (e.g. indicating along the bitstream) may be defined to refer to out-of-band transmission, signalling, or storage in a manner that the out-of-band data is associated with the bitstream. The phrase decoding along the bitstream or alike may refer to decoding the referred out-of-band data (which may be obtained from out-of-band transmission, signalling, or storage) that is associated with the bitstream. For example, an indication along the bitstream may refer to metadata in a container file that encapsulates the bitstream.

[0064] The H.264/AVC standard was developed by the Joint Video Team (JVT) of the Video Coding Experts Group (VCEG) of the Telecommunications Standardization Sector of International Telecommunication Union (ITU-T) and the Moving Picture Experts Group (MPEG) of International Organisation for Standardization (ISO) / International Electrotechnical Commission (IEC). The H.264/AVC standard is published by both parent standardization organizations, and it is referred to as ITU-T Recommendation H.264 and ISO/IEC International Standard 14496-10, also known as MPEG-4 Part 10 Advanced Video Coding (AVC). There have been multiple versions of the H.264/AVC standard, integrating new extensions or features to the specification. These extensions include Scalable Video Coding (SVC) and Multiview Video Coding (MVC).

[0065] Version 1 of the High Efficiency Video Coding (H.265/HEVC a.k.a. HEVC) standard was developed by the Joint Collaborative Team – Video Coding (JCT-VC) of VCEG and MPEG. The standard was published by both parent standardization organizations, and it is referred to as ITU-T Recommendation H.265 and ISO/IEC International Standard 23008-2, also known as MPEG-H Part 2 High Efficiency Video Coding (HEVC). Later versions of H.265/HEVC included scalable, multiview, fidelity range, three-dimensional, and screen content coding extensions which may be abbreviated SHVC, MV-HEVC, REXT, 3D-HEVC, and SCC, respectively.

[0066] Versatile Video Coding (VVC) (MPEG-I Part 3), a.k.a. ITU-T H.266, is a video compression standard developed by the Joint Video Experts Team (JVET) of the Moving Picture Experts Group (MPEG), (formally ISO/IEC JTC1 SC29 WG11) and Video Coding Experts Group (VCEG) of the International Telecommunication Union (ITU) to be the successor to HEVC/H.265.

[0067] Some key definitions, bitstream and coding structures, and concepts of H.264/AVC and HEVC are described in this section as an example of a video encoder, decoder, encoding

method, decoding method, and a bitstream structure, wherein the embodiments may be implemented. Some of the key definitions, bitstream and coding structures, and concepts of H.264/AVC are the same as in HEVC – hence, they are described below jointly. The aspects of the invention are not limited to H.264/AVC or HEVC, but rather the description is given for one possible basis on top of which the invention may be partly or fully realized.

[0068] Similarly to many earlier video coding standards, the bitstream syntax and semantics as well as the decoding process for error-free bitstreams are specified in H.264/AVC and HEVC. The encoding process is not specified, but encoders must generate conforming bitstreams. Bitstream and decoder conformance can be verified with the Hypothetical Reference Decoder (HRD). The standards contain coding tools that help in coping with transmission errors and losses, but the use of the tools in encoding is optional and no decoding process has been specified for erroneous bitstreams.

[0069] The elementary unit for the input to an H.264/AVC or HEVC encoder and the output of an H.264/AVC or HEVC decoder, respectively, is a picture. A picture given as an input to an encoder may also be referred to as a source picture, and a picture decoded by a decoder may be referred to as a decoded picture.

[0070] The source and decoded pictures are each comprised of one or more sample arrays, such as one of the following sets of sample arrays:

- Luma (Y) only (monochrome).
- Luma and two chroma (YCbCr or YCgCo).
- Green, Blue and Red (GBR, also known as RGB).
- Arrays representing other unspecified monochrome or tri-stimulus color samplings (for example, YZX, also known as XYZ).

[0071] In H.264/AVC and HEVC, a picture may either be a frame or a field. A frame comprises a matrix of luma samples and possibly the corresponding chroma samples. A field is a set of alternate sample rows of a frame and may be used as encoder input, when the source signal is interlaced. Chroma sample arrays may be absent (and hence monochrome sampling may be in use) or chroma sample arrays may be subsampled when compared to luma sample arrays.

Chroma formats may be summarized as follows:

- In monochrome sampling there is only one sample array, which may be nominally considered the luma array.

- In 4:2:0 sampling, each of the two chroma arrays has half the height and half the width of the luma array.
- In 4:2:2 sampling, each of the two chroma arrays has the same height and half the width of the luma array.
- In 4:4:4 sampling when no separate color planes are in use, each of the two chroma arrays has the same height and width as the luma array.

[0072] In H.264/AVC and HEVC, it is possible to code sample arrays as separate color planes into the bitstream and respectively decode separately coded color planes from the bitstream. When separate color planes are in use, each one of them is separately processed (by the encoder and/or the decoder) as a picture with monochrome sampling.

[0073] A partitioning may be defined as a division of a set into subsets such that each element of the set is in exactly one of the subsets.

[0074] When describing the operation of HEVC encoding and/or decoding, the following terms may be used. A coding block may be defined as an $N \times N$ block of samples for some value of N such that the division of a coding tree block into coding blocks is a partitioning. A coding tree block (CTB) may be defined as an $N \times N$ block of samples for some value of N such that the division of a component into coding tree blocks is a partitioning. A coding tree unit (CTU) may be defined as a coding tree block of luma samples, two corresponding coding tree blocks of chroma samples of a picture that has three sample arrays, or a coding tree block of samples of a monochrome picture or a picture that is coded using three separate color planes and syntax structures used to code the samples. A coding unit (CU) may be defined as a coding block of luma samples, two corresponding coding blocks of chroma samples of a picture that has three sample arrays, or a coding block of samples of a monochrome picture or a picture that is coded using three separate color planes and syntax structures used to code the samples. A CU with the maximum allowed size may be named as LCU (largest coding unit) or coding tree unit (CTU) and the video picture is divided into non-overlapping LCUs.

[0075] A CU consists of one or more prediction units (PU) defining the prediction process for the samples within the CU and one or more transform units (TU) defining the prediction error coding process for the samples in the said CU. Typically, a CU consists of a square block of samples with a size selectable from a predefined set of possible CU sizes. Each PU and TU can be further split into smaller PUs and TUs in order to increase granularity of the prediction and

prediction error coding processes, respectively. Each PU has prediction information associated with it defining what kind of a prediction is to be applied for the pixels within that PU (e.g. motion vector information for inter predicted PUs and intra prediction directionality information for intra predicted PUs).

[0076] Each TU can be associated with information describing the prediction error decoding process for the samples within the said TU (including e.g. DCT coefficient information). It is typically signalled at CU level whether prediction error coding is applied or not for each CU. In the case there is no prediction error residual associated with the CU, it can be considered there are no TUs for the said CU. The division of the image into CUs, and division of CUs into PUs and TUs is typically signalled in the bitstream allowing the decoder to reproduce the intended structure of these units.

[0077] In HEVC, a picture can be partitioned in tiles, which are rectangular and contain an integer number of LCUs. In HEVC, the partitioning to tiles forms a regular grid, where heights and widths of tiles differ from each other by one LCU at the maximum. In HEVC, a slice is defined to be an integer number of coding tree units contained in one independent slice segment and all subsequent dependent slice segments (if any) that precede the next independent slice segment (if any) within the same access unit. In HEVC, a slice segment is defined to be an integer number of coding tree units ordered consecutively in the tile scan and contained in a single NAL unit. The division of each picture into slice segments is a partitioning. In HEVC, an independent slice segment is defined to be a slice segment for which the values of the syntax elements of the slice segment header are not inferred from the values for a preceding slice segment, and a dependent slice segment is defined to be a slice segment for which the values of some syntax elements of the slice segment header are inferred from the values for the preceding independent slice segment in decoding order. In HEVC, a slice header is defined to be the slice segment header of the independent slice segment that is a current slice segment or is the independent slice segment that precedes a current dependent slice segment, and a slice segment header is defined to be a part of a coded slice segment containing the data elements pertaining to the first or all coding tree units represented in the slice segment. The CUs are scanned in the raster scan order of LCUs within tiles or within a picture, if tiles are not in use. Within an LCU, the CUs have a specific scan order.

[0078] The decoder reconstructs the output video by applying prediction means similar to the encoder to form a predicted representation of the pixel blocks (using the motion or spatial information created by the encoder and stored in the compressed representation) and prediction error decoding (inverse operation of the prediction error coding recovering the quantized prediction error signal in spatial pixel domain). After applying prediction and prediction error decoding means the decoder sums up the prediction and prediction error signals (pixel values) to form the output video frame. The decoder (and encoder) can also apply additional filtering means to improve the quality of the output video before passing it for display and/or storing it as prediction reference for the forthcoming frames in the video sequence.

[0079] The filtering may for example include one more of the following: deblocking, sample adaptive offset (SAO), and/or adaptive loop filtering (ALF). H.264/AVC includes a deblocking, whereas HEVC includes both deblocking and SAO.

[0080] In typical video codecs the motion information is indicated with motion vectors associated with each motion compensated image block, such as a prediction unit. Each of these motion vectors represents the displacement of the image block in the picture to be coded (in the encoder side) or decoded (in the decoder side) and the prediction source block in one of the previously coded or decoded pictures. In order to represent motion vectors efficiently those are typically coded differentially with respect to block specific predicted motion vectors. In typical video codecs the predicted motion vectors are created in a predefined way, for example calculating the median of the encoded or decoded motion vectors of the adjacent blocks. Another way to create motion vector predictions is to generate a list of candidate predictions from adjacent blocks and/or co-located blocks in temporal reference pictures and signalling the chosen candidate as the motion vector predictor. In addition to predicting the motion vector values, it can be predicted which reference picture(s) are used for motion-compensated prediction and this prediction information may be represented for example by a reference index of previously coded/decoded picture. The reference index is typically predicted from adjacent blocks and/or co-located blocks in temporal reference picture. Moreover, typical high efficiency video codecs employ an additional motion information coding/decoding mechanism, often called merging/merge mode, where all the motion field information, which includes motion vector and corresponding reference picture index for each available reference picture list, is predicted and used without any modification/correction. Similarly, predicting the motion field information is

carried out using the motion field information of adjacent blocks and/or co-located blocks in temporal reference pictures and the used motion field information is signalled among a list of motion field candidate list filled with motion field information of available adjacent/co-located blocks.

[0081] In typical video codecs the prediction residual after motion compensation is first transformed with a transform kernel (like DCT) and then coded. The reason for this is that often there still exists some correlation among the residual and transform can in many cases help reduce this correlation and provide more efficient coding.

[0082] Video coding standards and specifications may allow encoders to divide a coded picture to coded slices or alike. In-picture prediction is typically disabled across slice boundaries. Thus, slices can be regarded as a way to split a coded picture to independently decodable pieces. In H.264/AVC and HEVC, in-picture prediction may be disabled across slice boundaries. Thus, slices can be regarded as a way to split a coded picture into independently decodable pieces, and slices are therefore often regarded as elementary units for transmission. In many cases, encoders may indicate in the bitstream which types of in-picture prediction are turned off across slice boundaries, and the decoder operation takes this information into account for example when concluding which prediction sources are available. For example, samples from a neighboring CU may be regarded as unavailable for intra prediction, if the neighboring CU resides in a different slice.

[0083] An elementary unit for the output of an H.264/AVC or HEVC encoder and the input of an H.264/AVC or HEVC decoder, respectively, is a Network Abstraction Layer (NAL) unit. For transport over packet-oriented networks or storage into structured files, NAL units may be encapsulated into packets or similar structures. A bytestream format has been specified in H.264/AVC and HEVC for transmission or storage environments that do not provide framing structures. The bytestream format separates NAL units from each other by attaching a start code in front of each NAL unit. To avoid false detection of NAL unit boundaries, encoders run a byte-oriented start code emulation prevention algorithm, which adds an emulation prevention byte to the NAL unit payload if a start code would have occurred otherwise. In order to enable straightforward gateway operation between packet- and stream-oriented systems, start code emulation prevention may always be performed regardless of whether the bytestream format is in use or not. A NAL unit may be defined as a syntax structure containing an indication of the type

of data to follow and bytes containing that data in the form of an RBSP interspersed as necessary with emulation prevention bytes. A raw byte sequence payload (RBSP) may be defined as a syntax structure containing an integer number of bytes that is encapsulated in a NAL unit. An RBSP is either empty or has the form of a string of data bits containing syntax elements followed by an RBSP stop bit and followed by zero or more subsequent bits equal to 0.

[0084] NAL units consist of a header and payload. In H.264/AVC and HEVC, the NAL unit header indicates the type of the NAL unit

[0085] In HEVC, a two-byte NAL unit header is used for all specified NAL unit types. The NAL unit header contains one reserved bit, a six-bit NAL unit type indication, a three-bit `nuh_temporal_id_plus1` indication for temporal level (may be required to be greater than or equal to 1) and a six-bit `nuh_layer_id` syntax element. The `temporal_id_plus1` syntax element may be regarded as a temporal identifier for the NAL unit, and a zero-based `TemporalId` variable may be derived as follows: $\text{TemporalId} = \text{temporal_id_plus1} - 1$. The abbreviation TID may be used to interchangeably with the `TemporalId` variable. `TemporalId` equal to 0 corresponds to the lowest temporal level. The value of `temporal_id_plus1` is required to be non-zero in order to avoid start code emulation involving the two NAL unit header bytes. The bitstream created by excluding all VCL NAL units having a `TemporalId` greater than or equal to a selected value and including all other VCL NAL units remains conforming. Consequently, a picture having `TemporalId` equal to `tid_value` does not use any picture having a `TemporalId` greater than `tid_value` as inter prediction reference. A sub-layer or a temporal sub-layer may be defined to be a temporal scalable layer (or a temporal layer, TL) of a temporal scalable bitstream, consisting of VCL NAL units with a particular value of the `TemporalId` variable and the associated non-VCL NAL units.

`nuh_layer_id` can be understood as a scalability layer identifier.

[0086] NAL units can be categorized into Video Coding Layer (VCL) NAL units and non-VCL NAL units. VCL NAL units are typically coded slice NAL units. In HEVC, VCL NAL units contain syntax elements representing one or more CU.

[0087] A non-VCL NAL unit may be for example one of the following types: a sequence parameter set, a picture parameter set, a supplemental enhancement information (SEI) NAL unit, an access unit delimiter, an end of sequence NAL unit, an end of bitstream NAL unit, or a filler data NAL unit. Parameter sets may be needed for the reconstruction of decoded pictures,

whereas many of the other non-VCL NAL units are not necessary for the reconstruction of decoded sample values.

[0088] Parameters that remain unchanged through a coded video sequence may be included in a sequence parameter set. In addition to the parameters that may be needed by the decoding process, the sequence parameter set may optionally contain video usability information (VUI), which includes parameters that may be important for buffering, picture output timing, rendering, and resource reservation. In HEVC a sequence parameter set RBSP includes parameters that can be referred to by one or more picture parameter set RBSPs or one or more SEI NAL units containing a buffering period SEI message. A picture parameter set contains such parameters that are likely to be unchanged in several coded pictures. A picture parameter set RBSP may include parameters that can be referred to by the coded slice NAL units of one or more coded pictures.

[0089] In HEVC, a video parameter set (VPS) may be defined as a syntax structure containing syntax elements that apply to zero or more entire coded video sequences as determined by the content of a syntax element found in the SPS referred to by a syntax element found in the PPS referred to by a syntax element found in each slice segment header.

[0090] A video parameter set RBSP may include parameters that can be referred to by one or more sequence parameter set RBSPs.

[0091] The relationship and hierarchy between video parameter set (VPS), sequence parameter set (SPS), and picture parameter set (PPS) may be described as follows. VPS resides one level above SPS in the parameter set hierarchy and in the context of scalability and/or 3D video. VPS may include parameters that are common for all slices across all (scalability or view) layers in the entire coded video sequence. SPS includes the parameters that are common for all slices in a particular (scalability or view) layer in the entire coded video sequence, and may be shared by multiple (scalability or view) layers. PPS includes the parameters that are common for all slices in a particular layer representation (the representation of one scalability or view layer in one access unit) and are likely to be shared by all slices in multiple layer representations.

[0092] VPS may provide information about the dependency relationships of the layers in a bitstream, as well as many other information that are applicable to all slices across all (scalability or view) layers in the entire coded video sequence. VPS may be considered to comprise two parts, the base VPS and a VPS extension, where the VPS extension may be optionally present.

[0093] Out-of-band transmission, signaling or storage can additionally or alternatively be used for other purposes than tolerance against transmission errors, such as ease of access or session negotiation. For example, a sample entry of a track in a file conforming to the ISO Base Media File Format may comprise parameter sets, while the coded data in the bitstream is stored elsewhere in the file or in another file. The phrase along the bitstream (e.g. indicating along the bitstream) or along a coded unit of a bitstream (e.g. indicating along a coded tile) may be used in claims and described embodiments to refer to out-of-band transmission, signaling, or storage in a manner that the out-of-band data is associated with the bitstream or the coded unit, respectively. The phrase decoding along the bitstream or along a coded unit of a bitstream or alike may refer to decoding the referred out-of-band data (which may be obtained from out-of-band transmission, signaling, or storage) that is associated with the bitstream or the coded unit, respectively.

[0094] A SEI NAL unit may contain one or more SEI messages, which are not required for the decoding of output pictures but may assist in related processes, such as picture output timing, rendering, error detection, error concealment, and resource reservation.

[0095] A coded picture is a coded representation of a picture.

[0096] In HEVC, a coded picture may be defined as a coded representation of a picture containing all coding tree units of the picture. In HEVC, an access unit (AU) may be defined as a set of NAL units that are associated with each other according to a specified classification rule, are consecutive in decoding order, and contain at most one picture with any specific value of `nuh_layer_id`. In addition to containing the VCL NAL units of the coded picture, an access unit may also contain non-VCL NAL units. Said specified classification rule may for example associate pictures with the same output time or picture output count value into the same access unit.

[0097] A bitstream may be defined as a sequence of bits, in the form of a NAL unit stream or a byte stream, that forms the representation of coded pictures and associated data forming one or more coded video sequences. A first bitstream may be followed by a second bitstream in the same logical channel, such as in the same file or in the same connection of a communication protocol. An elementary stream (in the context of video coding) may be defined as a sequence of one or more bitstreams. The end of the first bitstream may be indicated by a specific NAL unit, which may be referred to as the end of bitstream (EOB) NAL unit and which is the last NAL unit

of the bitstream. In HEVC and its current draft extensions, the EOB NAL unit is required to have `nuh_layer_id` equal to 0.

[0098] In H.264/AVC, a coded video sequence is defined to be a sequence of consecutive access units in decoding order from an IDR access unit, inclusive, to the next IDR access unit, exclusive, or to the end of the bitstream, whichever appears earlier.

[0099] In HEVC, a coded video sequence (CVS) may be defined, for example, as a sequence of access units that consists, in decoding order, of an IRAP access unit with `NoRaslOutputFlag` equal to 1, followed by zero or more access units that are not IRAP access units with

`NoRaslOutputFlag` equal to 1, including all subsequent access units up to but not including any subsequent access unit that is an IRAP access unit with `NoRaslOutputFlag` equal to 1. An IRAP access unit may be defined as an access unit in which the base layer picture is an IRAP picture. The value of `NoRaslOutputFlag` is equal to 1 for each IDR picture, each BLA picture, and each IRAP picture that is the first picture in that particular layer in the bitstream in decoding order, is the first IRAP picture that follows an end of sequence NAL unit having the same value of `nuh_layer_id` in decoding order. There may be means to provide the value of

`HandleCraAsBlaFlag` to the decoder from an external entity, such as a player or a receiver, which may control the decoder. `HandleCraAsBlaFlag` may be set to 1 for example by a player that seeks to a new position in a bitstream or tunes into a broadcast and starts decoding and then starts decoding from a CRA picture. When `HandleCraAsBlaFlag` is equal to 1 for a CRA picture, the CRA picture is handled and decoded as if it were a BLA picture.

[0100] In HEVC, a coded video sequence may additionally or alternatively (to the specification above) be specified to end, when a specific NAL unit, which may be referred to as an end of sequence (EOS) NAL unit, appears in the bitstream and has `nuh_layer_id` equal to 0.

[0101] A group of pictures (GOP) and its characteristics may be defined as follows. A GOP can be decoded regardless of whether any previous pictures were decoded. An open GOP is such a group of pictures in which pictures preceding the initial intra picture in output order might not be correctly decodable when the decoding starts from the initial intra picture of the open GOP. In other words, pictures of an open GOP may refer (in inter prediction) to pictures belonging to a previous GOP. An HEVC decoder can recognize an intra picture starting an open GOP, because a specific NAL unit type, CRA NAL unit type, may be used for its coded slices. A closed GOP is such a group of pictures in which all pictures can be correctly decoded when the decoding starts

from the initial intra picture of the closed GOP. In other words, no picture in a closed GOP refers to any pictures in previous GOPs. In H.264/AVC and HEVC, a closed GOP may start from an IDR picture. In HEVC a closed GOP may also start from a BLA_W_RADL or a BLA_N_LP picture. An open GOP coding structure is potentially more efficient in the compression compared to a closed GOP coding structure, due to a larger flexibility in selection of reference pictures.

[0102] A Decoded Picture Buffer (DPB) may be used in the encoder and/or in the decoder. There are two reasons to buffer decoded pictures, for references in inter prediction and for reordering decoded pictures into output order. As H.264/AVC and HEVC provide a great deal of flexibility for both reference picture marking and output reordering, separate buffers for reference picture buffering and output picture buffering may waste memory resources. Hence, the DPB may include a unified decoded picture buffering process for reference pictures and output reordering. A decoded picture may be removed from the DPB when it is no longer used as a reference and is not needed for output.

[0103] In many coding modes of H.264/AVC and HEVC, the reference picture for inter prediction is indicated with an index to a reference picture list. The index may be coded with variable length coding, which usually causes a smaller index to have a shorter value for the corresponding syntax element. In H.264/AVC and HEVC, two reference picture lists (reference picture list 0 and reference picture list 1) are generated for each bi-predictive (B) slice, and one reference picture list (reference picture list 0) is formed for each inter-coded (P) slice.

[0104] Many coding standards, including H.264/AVC and HEVC, may have decoding process to derive a reference picture index to a reference picture list, which may be used to indicate which one of the multiple reference pictures is used for inter prediction for a particular block. A reference picture index may be coded by an encoder into the bitstream in some inter coding modes or it may be derived (by an encoder and a decoder) for example using neighboring blocks in some other inter coding modes.

[0105] Motion parameter types or motion information may include but are not limited to one or more of the following types:

- an indication of a prediction type (e.g. intra prediction, uni-prediction, bi-prediction) and/or a number of reference pictures;
- an indication of a prediction direction, such as inter (a.k.a. temporal) prediction, inter-layer prediction, inter-view prediction, view synthesis prediction (VSP), and inter-

component prediction (which may be indicated per reference picture and/or per prediction type and where in some embodiments inter-view and view-synthesis prediction may be jointly considered as one prediction direction) and/or

- an indication of a reference picture type, such as a short-term reference picture and/or a long-term reference picture and/or an inter-layer reference picture (which may be indicated e.g. per reference picture)
- a reference index to a reference picture list and/or any other identifier of a reference picture (which may be indicated e.g. per reference picture and the type of which may depend on the prediction direction and/or the reference picture type and which may be accompanied by other relevant pieces of information, such as the reference picture list or alike to which reference index applies);
- a horizontal motion vector component (which may be indicated e.g. per prediction block or per reference index or alike);
- a vertical motion vector component (which may be indicated e.g. per prediction block or per reference index or alike);
- one or more parameters, such as picture order count difference and/or a relative camera separation between the picture containing or associated with the motion parameters and its reference picture, which may be used for scaling of the horizontal motion vector component and/or the vertical motion vector component in one or more motion vector prediction processes (where said one or more parameters may be indicated e.g. per each reference picture or each reference index or alike);
- coordinates of a block to which the motion parameters and/or motion information applies, e.g. coordinates of the top-left sample of the block in luma sample units;
- extents (e.g. a width and a height) of a block to which the motion parameters and/or motion information applies.

[0106] In comparison to the previous video coding standards, Versatile Video Codec (H.266/VVC) introduces a plurality of new coding tools, such as the following:

- Intra prediction
 - 67 intra mode with wide angles mode extension
 - Block size and mode dependent 4 tap interpolation filter
 - Position dependent intra prediction combination (PDPC)

- Cross component linear model intra prediction (CCLM)
 - Multi-reference line intra prediction
 - Intra sub-partitions
 - Weighted intra prediction with matrix multiplication
- Inter-picture prediction
 - Block motion copy with spatial, temporal, history-based, and pairwise average merging candidates
 - Affine motion inter prediction
 - sub-block based temporal motion vector prediction
 - Adaptive motion vector resolution
 - 8x8 block-based motion compression for temporal motion prediction
 - High precision (1/16 pel) motion vector storage and motion compensation with 8-tap interpolation filter for luma component and 4-tap interpolation filter for chroma component
 - Triangular partitions
 - Combined intra and inter prediction
 - Merge with MVD (MMVD)
 - Symmetrical MVD coding
 - Bi-directional optical flow
 - Decoder side motion vector refinement
 - Bi-prediction with CU-level weight
- Transform, quantization and coefficients coding
 - Multiple primary transform selection with DCT2, DST7 and DCT8
 - Secondary transform for low frequency zone
 - Sub-block transform for inter predicted residual
 - Dependent quantization with max QP increased from 51 to 63
 - Transform coefficient coding with sign data hiding
 - Transform skip residual coding
- Entropy Coding
 - Arithmetic coding engine with adaptive double windows probability update
- In loop filter
 - In-loop reshaping

- Deblocking filter with strong longer filter
- Sample adaptive offset
- Adaptive Loop Filter
- Screen content coding:
 - Current picture referencing with reference region restriction
- 360-degree video coding
 - Horizontal wrap-around motion compensation
- High-level syntax and parallel processing
 - Reference picture management with direct reference picture list signalling
 - Tile groups with rectangular shape tile groups

[0107] The partitioning in VVC is carried out similarly to HEVC, i.e., each picture is divided into coding tree units (CTUs). A picture may also be divided into slices, tiles, bricks and sub-pictures. CTU may be split into smaller CUs using quaternary tree structure. Each CU may be divided using quad-tree and nested multi-type tree including ternary and binary split. However, there are specific rules to infer partitioning in picture boundaries, and the redundant split patterns are disallowed in nested multi-type partitioning.

[0108] In the new coding tools listed above, the cross-component linear model (CCLM) prediction mode is used in the VVC to reduce the cross-component redundancy. Therein, the chroma samples are predicted based on the reconstructed luma samples of the same CU by using a linear model as follows:

$$\text{pred}_C(i, j) = \alpha \cdot \text{rec}_L'(i, j) + \beta \quad (\text{Eq. 1a})$$

[0109] where $\text{pred}_C(i, j)$ represents the predicted chroma samples in a CU and $\text{rec}_L'(i, j)$ represents the downsampled reconstructed luma samples of the same CU.

[0110] Alternatively, the below equation may be used for CCLM:

$$\text{pred}_C(i, j) = \alpha \cdot \text{rec}_L'^{(i, j)} \gg k + \beta \quad (\text{Eq. 1b})$$

where $\gg$ operation denotes a bit shifting to right by value k .

[0111] The CCLM parameters (α and β) are derived with at most four neighbouring chroma samples and their corresponding down-sampled luma samples. Suppose the current chroma block dimensions are $W \times H$, then W' and H' are set as

- $W' = W, H' = H$ when LM mode is applied;

- $W' = W + H$ when LM-A mode is applied;
- $H' = H + W$ when LM-L mode is applied;

[0112] Herein, LM-A mode refers to linear model_{above}, where only the above template (i.e. sample values from neighbouring positions above the CU) is used to calculate the linear model coefficients. To get more samples, the above template is extended to $(W+H)$. LM-L mode, in turn, refers to linear model_{left}, where only left template (i.e. sample values from neighbouring positions left to the CU) is used to calculate the linear model coefficients. To get more samples, the left template is extended to $(H+W)$. For a non-square block, the above template is extended to $W+W$, the left template is extended to $H+H$.

[0113] The above neighbouring positions are denoted as $S[0, -1] \dots S[W' - 1, -1]$ and the left neighbouring positions are denoted as $S[-1, 0] \dots S[-1, H' - 1]$. Then the four samples are selected as

- $S[W' / 4, -1]$, $S[3 * W' / 4, -1]$, $S[-1, H' / 4]$, $S[-1, 3 * H' / 4]$ when LM mode is applied and both above and left neighbouring samples are available;
- $S[W' / 8, -1]$, $S[3 * W' / 8, -1]$, $S[5 * W' / 8, -1]$, $S[7 * W' / 8, -1]$ when LM-A mode is applied or only the above neighbouring samples are available;
- $S[-1, H' / 8]$, $S[-1, 3 * H' / 8]$, $S[-1, 5 * H' / 8]$, $S[-1, 7 * H' / 8]$ when LM-L mode is applied or only the left neighbouring samples are available;

[0114] The four neighbouring luma samples at the selected positions are down-sampled and compared four times to find two smaller values: $x0A$ and $x1A$, and two larger values: $x0B$ and $x1B$. Their corresponding chroma sample values are denoted as $y0A$, $y1A$, $y0B$ and $y1B$. Then x_A , x_B , y_A and y_B are derived as:

$$X_a = (x_{0A}^0 + x_{1A}^1 + 1) \ggg 1; X_b = (x_{0B}^0 + x_{1B}^1 + 1) \ggg 1; Y_a = (y_{0A}^0 + y_{1A}^1 + 1) \ggg 1; Y_b = (y_{0B}^0 + y_{1B}^1 + 1) \ggg 1 \quad (\text{Eq. 2})$$

[0115] Finally, the linear model parameters α and β are obtained according to the following equations:

$$\alpha = \frac{Y_a - Y_b}{X_a - X_b} \quad (\text{Eq. 3})$$

$$\beta = Y_b - \alpha \cdot X_b \quad (\text{Eq. 4})$$

[0116] Figure 5 shows an example of the location of the left and above samples and the sample of the current block involved in the CCLM mode.

[0117] The division operation to calculate parameter α is implemented with a look-up table. To reduce the memory required for storing the table, the diff value (difference between

maximum and minimum values) and the parameter α are expressed by an exponential notation. For example, diff is approximated with a 4-bit significant part and an exponent. Consequently, the table for $1/\text{diff}$ is reduced into 16 elements for 16 values of the significant part as follows:

$$\text{DivTable} [] = \{ 0, 7, 6, 5, 5, 4, 4, 3, 3, 2, 2, 1, 1, 1, 1, 0 \} \quad (\text{Eq. 5})$$

[0118] This provides the benefit of both reducing the complexity of the calculation as well as the memory size required for storing the needed tables.

[0119] To match the chroma sample locations for 4:2:0 video sequences, two types of downsampling filter are applied to luma samples to achieve 2 to 1 downsampling ratio in both horizontal and vertical directions. The selection of downsampling filter is specified by a SPS level flag. The two downsampling filters are as follows, which are corresponding to “type-0” and “type-2” content, respectively:

$$\text{Rec}_L'(i, j) = \left[\frac{\text{rec}_L(2i-1, 2j-1) + 2 \cdot \text{rec}_L(2i-1, 2j) + \text{rec}_L(2i+1, 2j-1) + \text{rec}_L(2i+1, 2j)}{\text{rec}_L(2i-1, 2j) + 2 \cdot \text{rec}_L(2i, 2j) + \text{rec}_L(2i+1, 2j) + 4} \right] \gg 3 \quad (\text{Eq. 6})$$

$$\text{rec}_L'(i, j) = \left[\frac{\text{rec}_L(2i, 2j-1) + \text{rec}_L(2i-1, 2j) + 4 \cdot \text{rec}_L(2i, 2j) + \text{rec}_L(2i+1, 2j) + \text{rec}_L(2i, 2j+1) + 4}{\text{rec}_L(2i+1, 2j) + \text{rec}_L(2i, 2j+1) + 4} \right] \gg 3 \quad (\text{Eq. 7})$$

[0120] It is noted that only one luma line (general line buffer in intra prediction) is used to make the down-sampled luma samples when the upper reference line is at the CTU boundary.

[0121] This parameter computation is performed as part of the decoding process and is not just as an encoder search operation. As a result, no syntax is used to convey the α and β values to the decoder.

[0122] For chroma intra mode coding, a total of 8 intra modes are allowed for chroma intra mode coding. Those modes include five traditional intra modes and three cross-component linear model modes (CCLM, LM_A, and LM_L). Chroma mode signalling and derivation process are shown in Table 1. Chroma mode coding directly depends on the intra prediction mode of the corresponding luma block. Since separate block partitioning structure for luma and chroma components is enabled in I slices, one chroma block may correspond to multiple luma blocks. Therefore, for Chroma DM mode, the intra prediction mode of the corresponding luma block covering the center position of the current chroma block is directly inherited.

Chroma prediction mode	Corresponding luma intra prediction mode				
	0	50	18	1	X (0 <= X <= 66)
0	66	0	0	0	0
1	50	66	50	50	50
2	18	18	66	18	18
3	1	1	1	66	1
4	0	50	18	1	X
5	81	81	81	81	81
6	82	82	82	82	82
7	83	83	83	83	83

Table 1.

[0123] A single binarization table is used regardless of the value of `sps_cclm_enabled_flag` as shown in Table 2.

Value of <code>intra_chroma_pred_mode</code>	Bin string
4	00
0	0100
1	0101
2	0110
3	0111
5	10
6	110
7	111

Table 2.

[0124] In Table 2, the first bin indicates whether it is regular (0) or LM modes (1). If it is LM mode, then the next bin indicates whether it is LM_CHROMA (0) or not. If it is not LM_CHROMA, next 1 bin indicates whether it is LM_L (0) or LM_A (1). For this case, when `sps_cclm_enabled_flag` is 0, the first bin of the binarization table for the corresponding `intra_chroma_pred_mode` can be discarded prior to the entropy coding. Or, in other words, the first bin is inferred to be 0 and hence not coded. This single binarization table is used for both `sps_cclm_enabled_flag` equal to 0 and 1 cases. The first two bins in Table 2 are context coded with its own context model, and the rest bins are bypass coded.

[0125] In addition, in order to reduce luma-chroma latency in dual tree, when the 64x64 luma coding tree node is partitioned with Not Split (and intra sub-partitions (ISP) is not used for the

64x64 CU) or QT, the chroma CUs in 32x32 / 32x16 chroma coding tree node are allowed to use CCLM in the following way:

- If the 32x32 chroma node is not split or partitioned QT split, all chroma CUs in the 32x32 node can use CCLM
- If the 32x32 chroma node is partitioned with Horizontal BT, and the 32x16 child node does not split or uses Vertical BT split, all chroma CUs in the 32x16 chroma node can use CCLM.

[0126] In all the other luma and chroma coding tree split conditions, CCLM is not allowed for chroma CU.

[0127] Multi-model LM (MMLM)

[0128] The CCLM included in VVC is extended by adding three Multi-model LM (MMLM) modes. In each MMLM mode, the reconstructed neighbouring samples are classified into two classes using a threshold which is the average of the luma reconstructed neighbouring samples. The linear model of each class is derived using the Least-Mean-Square (LMS) method. For the CCLM mode, the LMS method is also used to derive the linear model. Figures 6a and 6b illustrate two luma-to-chroma models obtained for luma (Y) threshold of 17 in sample domain and spatial domain, respectively. Each luma-to-chroma model has its own linear model parameters α and β . As can be seen in Figure 6b, each luma-to-chroma model corresponds to a spatial segmentation of the content (i.e., they correspond to different objects or textures in the scene).

[0129] Multiple reference line (MRL) intra prediction

[0130] Multiple reference line (MRL) intra prediction uses more reference lines for intra prediction. In Figure 7, an example of 4 reference lines is depicted, where the samples of segments A and F are not fetched from reconstructed neighbouring samples but padded with the closest samples from Segment B and E, respectively. HEVC intra-picture prediction uses the nearest reference line (i.e., reference line 0). In MRL, 2 additional lines (reference line 1 and reference line 3) are used.

[0131] The index of selected reference line (`mrl_idx`) is signalled and used to generate intra predictor. For reference line `idx`, which is greater than 0, only include additional reference line modes in MPM list and only signal mpm index without remaining mode. The reference line

index is signalled before intra prediction modes, and Planar mode is excluded from intra prediction modes in case a nonzero reference line index is signalled.

[0132] MRL is disabled for the first line of blocks inside a CTU to prevent using extended reference samples outside the current CTU line. Also, PDPC is disabled when additional line is used. For MRL mode, the derivation of DC value in DC intra prediction mode for non-zero reference line indices are aligned with that of reference line index 0. MRL requires the storage of 3 neighboring luma reference lines with a CTU to generate predictions. The CCLM tool also requires 3 neighboring luma reference lines for its down-sampling filters. The definition of MLR to use the same 3 lines is aligned as CCLM to reduce the storage requirements for decoders.

[0133] Intra sub-partitions (ISP)

[0134] The intra sub-partitions (ISP) divides luma intra-predicted blocks vertically or horizontally into 2 or 4 sub-partitions depending on the block size. For example, minimum block size for ISP is 4×8 (or 8×4). If block size is greater than 4×8 (or 8×4) then the corresponding block is divided by 4 sub-partitions. It has been noted that the $M \times 128$ (with $M \leq 64$) and $128 \times N$ (with $N \leq 64$) ISP blocks could generate a potential issue with the 64×64 VDP. For example, an $M \times 128$ CU in the single tree case has an $M \times 128$ luma TB and two corresponding $M/2 \times 64$ chroma TBs. If the CU uses ISP, then the luma TB will be divided into four $M \times 32$ TBs (only the horizontal split is possible), each of them smaller than a 64×64 block. However, in the current design of ISP chroma blocks are not divided. Therefore, both chroma components will have a size greater than a 32×32 block. Analogously, a similar situation could be created with a $128 \times N$ CU using ISP. Hence, these two cases are an issue for the 64×64 decoder pipeline. For this reason, the CU sizes that can use ISP is restricted to a maximum of 64×64 . All sub-partitions fulfil the condition of having at least 16 samples.

[0135] Matrix weighted Intra Prediction (MIP)

[0136] Matrix weighted intra prediction (MIP) method is a newly added intra prediction technique into VVC. For predicting the samples of a rectangular block of width W and height H , matrix weighted intra prediction (MIP) takes one line of H reconstructed neighbouring boundary samples left of the block and one line of W reconstructed neighbouring boundary samples above the block as input. If the reconstructed samples are unavailable, they are generated as it is done in the conventional intra prediction. The generation of the prediction signal is based on the

following three steps, which are averaging, matrix vector multiplication and linear interpolation as shown in Figure 8.

[0137] Inter prediction in VVC

[0138] A merge list may include the following candidates:

- a) Spatial MVP from spatial neighbour CUs
- b) Temporal MVP from collocated CUs
- c) History-based MVP from a FIFO table
- d) Pairwise average MVP (using the candidates already in the list)
- e) Zero MVs.

[0139] Merged mode with motion vector difference (MMVD) is to signal MVDs and a resolution index after signaling merge candidate. In Symmetric MVD, motion information of list-1 are derived from motion information of list-0 in bi-prediction case.

[0140] In Affine prediction, several motion vectors are indicated/signaled for different corners of a block, which are used to derive the motion vectors of sub-block. In affine merge, affine motion information of a block is generated based on the normal or affine motion information of the neighboring blocks.

[0141] In Sub-block-based temporal motion vector prediction, motion vectors of sub-blocks of the current block are predicted from a proper subblocks in the reference frame which are indicated by the motion vector of a spatial neighboring block (if available).

[0142] In Adaptive motion vector resolution (AMVR), precision of MVD is signaled for each CU.

[0143] In Bi-prediction with CU-level weight, an index indicated the weight values for weighted average of two prediction block. Bi-directional optical flow (BDOF) refines the motion vectors in bi-prediction case. BDOF generates two prediction blocks using the signaled motion vectors. Then a motion refinement is calculated two minimize the error between two prediction blocks using their gradient values. The final prediction blocks are refined using the motion refinement and gradient values.

[0144] The CCLM is extended by the addition of the three Multi-model LM (MMLM) modes, as described above, may be referred to as a part of the Enhanced Compression Model (ECM). In each MMLM mode, the reconstructed neighbouring samples are classified into two classes using a threshold which is the average of the luma reconstructed neighbouring samples. The linear

model of each class is derived using the Least-Mean-Square (LMS) method. For the CCLM mode, the LMS method is also used to derive the linear model.

[0145] However, the two classes of luma-to-chroma models may be significantly different from each other. Hard thresholding the co-located luma samples may create colour discrepancies, artificial edges, or other artefacts within the target chroma block due to the use of two different luma-to-chroma models.

[0146] Now an improved method for achieving a better correlation between the luma and chroma channels is introduced.

[0147] A method according to an aspect is shown in Figure 9, where the method comprises obtaining (900) an image block unit comprising samples in color channels of one or two chrominance channels and one luminance channel; reconstructing (902) samples of said luminance channels of the image block unit; predicting (904) chrominance samples of at least one color channel of image block unit using at least two cross-component prediction models, each comprising their corresponding linear model parameters, and a luminance threshold value dividing the samples of said luminance channel between said at least two cross-component prediction models; defining (906) a weighting function for said at least two cross-component prediction models based on location of two or more chrominance samples and the luminance threshold value, or based on at least one luminance sample value and the luminance threshold value; and applying (908) a smoothing function for samples residing within an offset value from said luminance threshold value, wherein said smoothing function adjusts chrominance values of said samples residing within said offset value by applying said weighting function on the corresponding at least two linear model parameters of the at least two cross-component prediction models.

[0148] Thus, the method starts by carrying out the multi-model cross-component linear model algorithm (MMLM) for the image block unit, thereby obtaining parameters for the at least two luma-to-chroma models (i.e., classes) and the threshold value dividing the luma samples into their corresponding luma-to-chroma models of said at least two luma-to-chroma models. Now a transition area (a.k.a. overlapping area or boundary area) is defined for the samples residing in the vicinity of the threshold value, wherein the transition area may comprise samples residing within a positive and/or negative offset value from the threshold value. A weighting function is defined for said at least two luma-to-chroma models to be applied for predicting chroma samples

within said transition area, wherein the weighting function controls the strength of each luma-to-chroma model. Then a smoothing function, which is a linear interpolator including the weighting function, is applied to control the final blend of the separate luma-to-chroma models in predicting the chroma samples within said transition area.

[0149] Accordingly, blending of two or more chroma samples (predicted by their corresponding luma-to-chroma models) is performed by applying a parametric smoothing function in the transition area of the given at least two luma-to-chroma models. The use of the weighting and the smoothing functions ensures a smooth transition between two (or more) luma-to-chroma models and mitigates the unwanted effects of an abrupt switch of luma-to-chroma models during the prediction of chroma samples.

[0150] As described above, the linear luma-to-chroma model parameters α and β are obtained for the current CCLM. Similarly, in case of at least two luma-to-chroma models (corresponding to an instance of MMLM), corresponding linear model parameters α_1 and β_1 are obtained for the first luma-to-chroma model and α_2 and β_2 are obtained for the second luma-to-chroma model. Moreover, a threshold luma value L_t is defined, for example as the average of the luma reconstructed neighbouring samples, for classifying the reconstructed neighbouring samples into the separate luma-to-chroma models corresponding to separate classes of samples.

[0151] According to an embodiment, the value of the weighting function $w(\cdot)$ is adjusted to remain within a range of $[0,1]$. This enables to use the value of the weighting function and its complementary value within the range of $[0,1]$ as the weights for controlling the effects of the separate luma-to-chroma models on the final blend of the prediction of chroma samples within the transition area.

[0152] According to an embodiment, the weighting function $w(\cdot)$ is defined in the sample domain. Thus, the operations are performed on the luma and chroma intensity values of the samples. This is illustrated in Figure 10a, where the threshold value L_t is utilised in defining a cross-over point between the two luma-to-chroma models, and the transition area (area of overlapped models) is depicted as an ellipse.

[0153] According to an embodiment, the smoothing function $C(\cdot)$ is defined in the sample domain as

$$C(x, y, L_t, L, \alpha_1, \beta_1, \alpha_2, \beta_2) = w(x, y, L_t)(\alpha_2 L(x, y) + \beta_2) + (1 - w(x, y, L_t))(\alpha_1 L(x, y) + \beta_1).$$

[0154] According to an embodiment, the weighting function $w(x, y, L_t)$ is defined in the sample domain as a linear weighting function

$$w(x, y, L_t) = \gamma \cdot (L(x, y) - L_t + R),$$

where γ is a scaling term and R is the range of the reference luma samples. The proper choice of R and γ , wherein γ may be defined, for example, as $R/2$, ensure that the output of the function is in the necessary $[0,1]$ range as needed by the smoothing function. The linear weighting function $w(x, y, L_t)$ uses a simple linear blending in the sample domain and considers only the distance of the current co-located luma sample to the threshold value L_t .

[0155] According to an embodiment, the weighting function $w(x, y, \mu, \sigma)$ is defined in the sample domain as a non-linear weighting function

$$w(x, y, \mu, \sigma) = e^{-0.5 \left(\frac{L(x, y) - \mu}{\sigma} \right)^2},$$

where $\mu = L_t + R$ and $\sigma = \gamma R$. Herein, for example a value $\gamma = \frac{2}{3} R$ may be used. The use of an exponential function allows the weight of the less probable model to decay faster than in the linear case.

[0156] According to an embodiment, the weighting function $w(\cdot)$ is defined in spatial domain as $w(\Delta x, \Delta y)$ where Δx and Δy define distance to the nearest boundary sample (x_B, y_B) . Thus, in the spatial domain the operations are performed on the horizontal and vertical coordinate values of the samples. This is illustrated in Figure 10b, where the threshold value L_t are the borders between the lighter and the darker areas, and the transition areas TA are depicted as the shadowed areas extending beyond both borders.

[0157] According to an embodiment, the smoothing function C is defined in the spatial domain as

$$C(x, y, L_t, L, \alpha_1, \beta_1, \alpha_2, \beta_2) = w(\Delta x, \Delta y)(\alpha_2 L(x, y) + \beta_2) + (1 - w(\Delta x, \Delta y))(\alpha_1 L(x, y) + \beta_1).$$

[0158] According to an embodiment, the weighting function is defined in the spatial domain as

$$w(\Delta x, \Delta y) = \varepsilon - \gamma \sqrt{(\Delta x)^2 + (\Delta y)^2}$$

where ε is an offset term and γ is a scaling term. With a proper choice of γ it can be ensured that $w(\Delta x, \Delta y) \in [0,1]$. The offset term ε is defined so that the models get non-zero but equal weights at the boundary samples where $\Delta x = 0, \Delta y = 0$, and can be for example $\varepsilon=0.5$.

[0159] According to an embodiment, different linear models may be derived for the beginning and the end of the luma spectrum of the luma-to-chroma model. Two cut-off luma sample values y_0 and y_1 may be determined based the reference luma sample values used in derivation of the luma-to-chroma model, as depicted in Figure 11. For luma values below the lower cut-off value y_0 “bottom range” linear model parameters α_b, β_b can be derived from the first linear model applicable to samples below the L_t threshold and having parameters α_1, β_1 . Similarly, for a “top range” linear model applicable above the higher cut-off value y_1 the linear model parameters α_t, β_t can be derived from the parameters α_2, β_2 of the second linear model in effect for luma sample values above the L_t threshold. In the case only one linear model is determined from the reference samples, the top range and bottom range linear model can be derived from the parameters α, β of that single model. Advantageously the slope parameters for the bottom and top range models α_b and α_t are derived from the slope parameters of first and second linear models by scaling the corresponding slope parameters α_1 and α_2 by a factor s , where s is selected to be smaller than one:

$$\begin{aligned}\alpha_b &= s\alpha_1 \\ \alpha_t &= s\alpha_2\end{aligned}$$

[0160] Or in the case of a single reference sample originating model:

$$\begin{aligned}\alpha_b &= s\alpha \\ \alpha_t &= s\alpha\end{aligned}$$

[0161] Scaling factor s can be selected to be, for example, 0.5 corresponding to applying half the slope in the end areas of the luma spectrum. Using a value smaller than one for s can stabilize the mapping function and provide more reliable prediction results in the range where there are no reference samples guiding the mapping. Scaling parameter s can be either pre-selected, or it can be signalled or derived from a video bitstream. Alternatively, the slope parameter can be adjusted in different ways, for example using look-up tables, piecewise linear or non-linear functions.

[0162] The lower cut-off value may be generated as a function of the lowest reference luma sample value and the higher cut-off value may be generated as a function of the highest reference

luma sample value found in a set of reference values. Alternatively, the cut-off values may be determined as a function of multiple reference luma values. For example, the average of two or more reference luma sample values may be used. The function to determine the lower cut-off value may include deducting an offset from a luma value and the function to determine the higher cut-off value may include adding an offset to a luma value.

[0163] The same weighting function $w(\cdot)$ that is used for blending between linear models derived from reference samples can be used also to smooth the transition from those models to the models generated for the bottom and top ranges of the luma spectrum. Alternatively, a different weighting function or no weighting can be selected to be applied for transitions to bottom and top ranges.

[0164] In the following, embodiments relating to various implementation options are disclosed.

[0165] In an embodiment, any linear or non-linear function can be used as the weighting function $w(\cdot)$.

[0166] In an embodiment, a bilateral weighting function $w(\cdot)$ can be used to combine the properties of both sample and spatial domain weighting functions within a single non-linear function.

[0167] In an embodiment, the parameterization of the weighting function $w(\cdot)$ and/or smoothing function $C(\cdot)$ can include local or global information such as block dimensions, block location, QP, prediction mode or any other information concerning the prediction or decoding of the current chroma block, co-located luma block, frame or sequence.

[0168] In an embodiment, the smoothing function can be applied to all chroma blocks, its use is inferred by the decoder or a flag is signalled by the encoder to determine the chroma blocks where the smoothing function is applied.

[0169] In an embodiment, the smoothing function may be applied in a parametric vicinity (i.e., the overlapping area) of the threshold value L_t . For example only samples corresponding to $L(x, y) \leq L_t + T$ && $L(x, y) \geq L_t - T$ are considered, where T is a fixed, inferred or signalled by the encoder and && corresponds to logical AND.

[0170] In an embodiment, the overlapping area for either sample or spatial domain (or both) can be defined by an ellipse, polygon, or any other geometric construct.

[0171] In an embodiment, the overlapping area can be explicitly signalled by the encoder.

[0172] In an embodiment, the smoothing function may be applied in the sample domain in unsymmetric manner so that smoothing function $C(\cdot)$ is only applied when $L(x, y) \leq L_t + T_0$ & $L(x, y) \geq L_t - T_1$ where $T_0 \neq T_1$.

[0173] In an embodiment, the smoothing function $C(\cdot)$ may be applied only in the vicinity of the spatial boundary (i.e., the overlapping area) of two luma-to-chroma models, for example when $\sqrt{\Delta_x^2 + \Delta_y^2} \leq T$ where T is a fixed, inferred or signalled parameter.

[0174] In an embodiment, the smoothing function $C(\cdot)$ may be applied in the spatial domain in anisotropic manner so that the smoothing function is only applied when

$$\sqrt{(\omega_0 \Delta_x)^2 + (\omega_1 \Delta_y)^2} \leq T, \text{ where } \omega_0, \omega_1 \text{ are fixed, inferred or signalled by the encoder.}$$

[0175] In an embodiment, both the sample domain and spatial domain implementations can be used simultaneously, or they can be used in a selective manner where the usage of either (or both) method(s) is inferred by the decoder or signalled by the encoder.

[0176] In an embodiment, N luma-to-chroma models may be considered in cascaded form so that the predicted chroma sample becomes $C(x, y, \dots) = (C_1(x, y, \dots) + C_2(x, y, \dots) + \dots C_N(x, y, \dots))/N$ where $C_i(x, y, \dots) = w_i(x, y, L_{t_i})(\alpha_{i+1}L(x, y) + \beta_{i+1}) + (1 - w_i(x, y, L_{t_i})(\alpha_iL(x, y) + \beta_i))$, for $i = 1, \dots, N$.

[0177] In an embodiment of the cascaded form, the sum $C(x, y, \dots) = \frac{1}{N} \sum C_i(\cdot)$ can be divided into parallel operations with respect to the computation of each individual $C_i(\cdot)$.

[0178] In an embodiment, the weighting function $w(\cdot)$ can be selected from a group of pre-defined weighting functions. As an example, a group of two weighting functions are the linear and the exponential weighting functions. The selected weighting function is either inferred by the decoder or signalled by the encoder.

[0179] In an embodiment, the weighting function(s) $w(\cdot)$ can be signalled by the encoder.

[0180] In an embodiment, the weighting function(s) $w(\cdot)$ can be applied in sequential or parallel order with respect to chroma samples of a block.

[0181] In an embodiment, the smoothing function(s) $C(\cdot)$ can be applied in sequential or parallel order with respect to chroma samples of a block.

[0182] In an embodiment, the Cb and Cr components may use different weighting functions $w(\cdot)$. For example, Cb might use a linear weighting function while Cr uses a non-linear

weighting function or vice-versa. The component-wise selection of weighting functions is either inferred at the decoder or signalled by the encoder.

[0183] In an embodiment, the smoothing and weighting functions in sample or spatial domain or both may also be applied for two channel data, RGB or hyperspectral data that has more than three components.

[0184] In an embodiment, a linear model applied below a first cut-off luma value is configured to use a slope parameter smaller than a linear model applicable above the first cut-off luma value; and a linear model applied above a second cut-off luma value is configured to use a slope parameter smaller than a linear model applicable below the second cut-off luma value.

[0185] In an embodiment, a linear model applied below a first cut-off luma value is configured to use half the slope parameter a linear model applicable above the first cut-off luma value uses; and a linear model applied above a second cut-off luma value is configured to use half the slope parameter a linear model applicable below the second cut-off luma value uses.

[0186] In an embodiment, the first and the second cut-off luma values are determined based on the lowest and highest reference luma values used in deriving at least one linear luma-to-chroma mapping function.

[0187] According to an embodiment, a rounding and/or a clipping operation is performed on the chroma samples values obtained as a result of the smoothing function. Thus, the chroma samples values obtained as a result of the smoothing function may not be usable as such for further processing, but a minor adjustment may be required.

[0188] It is noted the methods and the embodiments related thereto can be applied to any image or video prediction tool that involves applying linear or non-linear model or models that are derived on the reference samples and are to be used in prediction of a sample or a group of samples such as a block in an image or video frame.

[0189] The methods and the related embodiments can be implemented in different ways. For example, the order of operations described above can be changed or the operations can be interleaved in different ways. Also, different additional operations can be applied in different stages of the processing. For example, there may be additional filtering, scaling, mapping or other processing applied to the final or intermediate result of described operations. The final or intermediate result of the operations described above may also be further combined with results of other operations.

[0190] According to an embodiment, the chroma blocks may also correspond to any of the red, green, or blue color components of the RGB color space.

[0191] According to an embodiment, the luma block may also correspond to any of the red, green, or blue color components of the RGB color space.

[0192] According to an embodiment the luma channel may instead correspond to a chroma channel or any other channel such as any of the RGB channels or any channel of any other colorspace or any hyperspectral channel.

[0193] According to an embodiment, the chroma channels may instead correspond to any other channels such as any of the RGB channels or any channels of any other color space or any hyperspectral channels.

[0194] An apparatus according to an aspect comprises means for obtaining an image block unit comprising samples in color channels of one or two chrominance channels and one luminance channel; means for reconstructing samples of said luminance channels of the image block unit; means for predicting chrominance samples of at least one color channel of image block unit using at least two cross-component prediction models, each comprising their corresponding linear model parameters, and a luminance threshold value dividing the samples of said luminance channel between said at least two cross-component prediction models; means for defining a weighting function for said at least two cross-component prediction models based on location of two or more chrominance samples and the luminance threshold value, or based on at least one luminance sample value and the luminance threshold value; and means for applying a smoothing function for samples residing within an offset value from said luminance threshold value, wherein said smoothing function adjusts chrominance values of said samples residing within said offset value by applying said weighting function on the corresponding at least two linear model parameters of the at least two cross-component prediction models.

[0195] As a further aspect, there is provided an apparatus comprising: at least one processor and at least one memory, said at least one memory stored with code thereon, which when executed by said at least one processor, causes the apparatus to perform at least: obtain an image block unit comprising samples in color channels of one or two chrominance channels and one luminance channel; reconstruct samples of said luminance channels of the image block unit; predict chrominance samples of at least one color channel of image block unit using at least two cross-component prediction models, each comprising their corresponding linear model

parameters, and a luminance threshold value dividing the samples of said luminance channel between said at least two cross-component prediction models; define a weighting function for said at least two cross-component prediction models based on location of two or more chrominance samples and the luminance threshold value, or based on at least one luminance sample value and the luminance threshold value; and apply a smoothing function for samples residing within an offset value from said luminance threshold value, wherein said smoothing function adjusts chrominance values of said samples residing within said offset value by applying said weighting function on the corresponding at least two linear model parameters of the at least two cross-component prediction models.

[0196] Such apparatuses may comprise e.g. the functional units disclosed in any of the Figures 1, 2, 4a, and 4b for implementing the embodiments.

[0197] Such an apparatus further comprises code, stored in said at least one memory, which when executed by said at least one processor, causes the apparatus to perform one or more of the embodiments disclosed herein.

[0198] Figure 12 is a graphical representation of an example multimedia communication system within which various embodiments may be implemented. A data source 1510 provides a source signal in an analog, uncompressed digital, or compressed digital format, or any combination of these formats. An encoder 1520 may include or be connected with a pre-processing, such as data format conversion and/or filtering of the source signal. The encoder 1520 encodes the source signal into a coded media bitstream. It should be noted that a bitstream to be decoded may be received directly or indirectly from a remote device located within virtually any type of network. Additionally, the bitstream may be received from local hardware or software. The encoder 1520 may be capable of encoding more than one media type, such as audio and video, or more than one encoder 1520 may be required to code different media types of the source signal. The encoder 1520 may also get synthetically produced input, such as graphics and text, or it may be capable of producing coded bitstreams of synthetic media. In the following, only processing of one coded media bitstream of one media type is considered to simplify the description. It should be noted, however, that typically real-time broadcast services comprise several streams (typically at least one audio, video and text sub-titling stream). It should also be noted that the system may include many encoders, but in the figure only one encoder 1520 is represented to simplify the description without a lack of generality. It should be

further understood that, although text and examples contained herein may specifically describe an encoding process, one skilled in the art would understand that the same concepts and principles also apply to the corresponding decoding process and vice versa.

[0199] The coded media bitstream may be transferred to a storage 1530. The storage 1530 may comprise any type of mass memory to store the coded media bitstream. The format of the coded media bitstream in the storage 1530 may be an elementary self-contained bitstream format, or one or more coded media bitstreams may be encapsulated into a container file, or the coded media bitstream may be encapsulated into a Segment format suitable for DASH (or a similar streaming system) and stored as a sequence of Segments. If one or more media bitstreams are encapsulated in a container file, a file generator (not shown in the figure) may be used to store the one more media bitstreams in the file and create file format metadata, which may also be stored in the file. The encoder 1520 or the storage 1530 may comprise the file generator, or the file generator is operationally attached to either the encoder 1520 or the storage 1530. Some systems operate “live”, i.e. omit storage and transfer coded media bitstream from the encoder 1520 directly to the sender 1540. The coded media bitstream may then be transferred to the sender 1540, also referred to as the server, on a need basis. The format used in the transmission may be an elementary self-contained bitstream format, a packet stream format, a Segment format suitable for DASH (or a similar streaming system), or one or more coded media bitstreams may be encapsulated into a container file. The encoder 1520, the storage 1530, and the server 1540 may reside in the same physical device or they may be included in separate devices. The encoder 1520 and server 1540 may operate with live real-time content, in which case the coded media bitstream is typically not stored permanently, but rather buffered for small periods of time in the content encoder 1520 and/or in the server 1540 to smooth out variations in processing delay, transfer delay, and coded media bitrate.

[0200] The server 1540 sends the coded media bitstream using a communication protocol stack. The stack may include but is not limited to one or more of Real-Time Transport Protocol (RTP), User Datagram Protocol (UDP), Hypertext Transfer Protocol (HTTP), Transmission Control Protocol (TCP), and Internet Protocol (IP). When the communication protocol stack is packet-oriented, the server 1540 encapsulates the coded media bitstream into packets. For example, when RTP is used, the server 1540 encapsulates the coded media bitstream into RTP packets according to an RTP payload format. Typically, each media type has a dedicated RTP

payload format. It should be again noted that a system may contain more than one server 1540, but for the sake of simplicity, the following description only considers one server 1540.

[0201] If the media content is encapsulated in a container file for the storage 1530 or for inputting the data to the sender 1540, the sender 1540 may comprise or be operationally attached to a "sending file parser" (not shown in the figure). In particular, if the container file is not transmitted as such but at least one of the contained coded media bitstream is encapsulated for transport over a communication protocol, a sending file parser locates appropriate parts of the coded media bitstream to be conveyed over the communication protocol. The sending file parser may also help in creating the correct format for the communication protocol, such as packet headers and payloads. The multimedia container file may contain encapsulation instructions, such as hint tracks in the ISOBMFF, for encapsulation of the at least one of the contained media bitstream on the communication protocol.

[0202] The server 1540 may or may not be connected to a gateway 1550 through a communication network, which may e.g. be a combination of a CDN, the Internet and/or one or more access networks. The gateway may also or alternatively be referred to as a middle-box. For DASH, the gateway may be an edge server (of a CDN) or a web proxy. It is noted that the system may generally comprise any number gateways or alike, but for the sake of simplicity, the following description only considers one gateway 1550. The gateway 1550 may perform different types of functions, such as translation of a packet stream according to one communication protocol stack to another communication protocol stack, merging and forking of data streams, and manipulation of data stream according to the downlink and/or receiver capabilities, such as controlling the bit rate of the forwarded stream according to prevailing downlink network conditions. The gateway 1550 may be a server entity in various embodiments.

[0203] The system includes one or more receivers 1560, typically capable of receiving, demodulating, and de-capsulating the transmitted signal into a coded media bitstream. The coded media bitstream may be transferred to a recording storage 1570. The recording storage 1570 may comprise any type of mass memory to store the coded media bitstream. The recording storage 1570 may alternatively or additively comprise computation memory, such as random access memory. The format of the coded media bitstream in the recording storage 1570 may be an elementary self-contained bitstream format, or one or more coded media bitstreams may be encapsulated into a container file. If there are multiple coded media bitstreams, such as an audio

stream and a video stream, associated with each other, a container file is typically used and the receiver 1560 comprises or is attached to a container file generator producing a container file from input streams. Some systems operate “live,” i.e. omit the recording storage 1570 and transfer coded media bitstream from the receiver 1560 directly to the decoder 1580. In some systems, only the most recent part of the recorded stream, e.g., the most recent 10-minute excerpt of the recorded stream, is maintained in the recording storage 1570, while any earlier recorded data is discarded from the recording storage 1570.

[0204] The coded media bitstream may be transferred from the recording storage 1570 to the decoder 1580. If there are many coded media bitstreams, such as an audio stream and a video stream, associated with each other and encapsulated into a container file or a single media bitstream is encapsulated in a container file e.g. for easier access, a file parser (not shown in the figure) is used to decapsulate each coded media bitstream from the container file. The recording storage 1570 or a decoder 1580 may comprise the file parser, or the file parser is attached to either recording storage 1570 or the decoder 1580. It should also be noted that the system may include many decoders, but here only one decoder 1570 is discussed to simplify the description without a lack of generality

[0205] The coded media bitstream may be processed further by a decoder 1570, whose output is one or more uncompressed media streams. Finally, a renderer 1590 may reproduce the uncompressed media streams with a loudspeaker or a display, for example. The receiver 1560, recording storage 1570, decoder 1570, and renderer 1590 may reside in the same physical device or they may be included in separate devices.

[0206] A sender 1540 and/or a gateway 1550 may be configured to perform switching between different representations e.g. for switching between different viewports of 360-degree video content, view switching, bitrate adaptation and/or fast start-up, and/or a sender 1540 and/or a gateway 1550 may be configured to select the transmitted representation(s). Switching between different representations may take place for multiple reasons, such as to respond to requests of the receiver 1560 or prevailing conditions, such as throughput, of the network over which the bitstream is conveyed. In other words, the receiver 1560 may initiate switching between representations. A request from the receiver can be, e.g., a request for a Segment or a Subsegment from a different representation than earlier, a request for a change of transmitted scalability layers and/or sub-layers, or a change of a rendering device having different

capabilities compared to the previous one. A request for a Segment may be an HTTP GET request. A request for a Subsegment may be an HTTP GET request with a byte range. Additionally or alternatively, bitrate adjustment or bitrate adaptation may be used for example for providing so-called fast start-up in streaming services, where the bitrate of the transmitted stream is lower than the channel bitrate after starting or random-accessing the streaming in order to start playback immediately and to achieve a buffer occupancy level that tolerates occasional packet delays and/or retransmissions. Bitrate adaptation may include multiple representation or layer up-switching and representation or layer down-switching operations taking place in various orders.

[0207] A decoder 1580 may be configured to perform switching between different representations e.g. for switching between different viewports of 360-degree video content, view switching, bitrate adaptation and/or fast start-up, and/or a decoder 1580 may be configured to select the transmitted representation(s). Switching between different representations may take place for multiple reasons, such as to achieve faster decoding operation or to adapt the transmitted bitstream, e.g. in terms of bitrate, to prevailing conditions, such as throughput, of the network over which the bitstream is conveyed. Faster decoding operation might be needed for example if the device including the decoder 1580 is multi-tasking and uses computing resources for other purposes than decoding the video bitstream. In another example, faster decoding operation might be needed when content is played back at a faster pace than the normal playback speed, e.g. twice or three times faster than conventional real-time playback rate.

[0208] In the above, some embodiments have been described with reference to and/or using terminology of HEVC and/or VVC. It needs to be understood that embodiments may be similarly realized with any video encoder and/or video decoder.

[0209] In the above, where the example embodiments have been described with reference to an encoder, it needs to be understood that the resulting bitstream and the decoder may have corresponding elements in them. Likewise, where the example embodiments have been described with reference to a decoder, it needs to be understood that the encoder may have structure and/or computer program for generating the bitstream to be decoded by the decoder. For example, some embodiments have been described related to generating a prediction block as part of encoding. Embodiments can be similarly realized by generating a prediction block as part

of decoding, with a difference that coding parameters, such as the horizontal offset and the vertical offset, are decoded from the bitstream than determined by the encoder.

[0210] The embodiments of the invention described above describe the codec in terms of separate encoder and decoder apparatus in order to assist the understanding of the processes involved. However, it would be appreciated that the apparatus, structures and operations may be implemented as a single encoder-decoder apparatus/structure/operation. Furthermore, it is possible that the coder and decoder may share some or all common elements.

[0211] Although the above examples describe embodiments of the invention operating within a codec within an electronic device, it would be appreciated that the invention as defined in the claims may be implemented as part of any video codec. Thus, for example, embodiments of the invention may be implemented in a video codec which may implement video coding over fixed or wired communication paths.

[0212] Thus, user equipment may comprise a video codec such as those described in embodiments of the invention above. It shall be appreciated that the term user equipment is intended to cover any suitable type of wireless user equipment, such as mobile telephones, portable data processing devices or portable web browsers.

[0213] Furthermore elements of a public land mobile network (PLMN) may also comprise video codecs as described above.

[0214] In general, the various embodiments of the invention may be implemented in hardware or special purpose circuits, software, logic or any combination thereof. For example, some aspects may be implemented in hardware, while other aspects may be implemented in firmware or software which may be executed by a controller, microprocessor or other computing device, although the invention is not limited thereto. While various aspects of the invention may be illustrated and described as block diagrams, flow charts, or using some other pictorial representation, it is well understood that these blocks, apparatus, systems, techniques or methods described herein may be implemented in, as non-limiting examples, hardware, software, firmware, special purpose circuits or logic, general purpose hardware or controller or other computing devices, or some combination thereof.

[0215] The embodiments of this invention may be implemented by computer software executable by a data processor of the mobile device, such as in the processor entity, or by hardware, or by a combination of software and hardware. Further in this regard it should be

noted that any blocks of the logic flow as in the Figures may represent program steps, or interconnected logic circuits, blocks and functions, or a combination of program steps and logic circuits, blocks and functions. The software may be stored on such physical media as memory chips, or memory blocks implemented within the processor, magnetic media such as hard disk or floppy disks, and optical media such as for example DVD and the data variants thereof, CD.

[0216] The memory may be of any type suitable to the local technical environment and may be implemented using any suitable data storage technology, such as semiconductor-based memory devices, magnetic memory devices and systems, optical memory devices and systems, fixed memory and removable memory. The data processors may be of any type suitable to the local technical environment, and may include one or more of general purpose computers, special purpose computers, microprocessors, digital signal processors (DSPs) and processors based on multi-core processor architecture, as non-limiting examples.

[0217] Embodiments of the inventions may be practiced in various components such as integrated circuit modules. The design of integrated circuits is by and large a highly automated process. Complex and powerful software tools are available for converting a logic level design into a semiconductor circuit design ready to be etched and formed on a semiconductor substrate.

[0218] Programs, such as those provided by Synopsys, Inc. of Mountain View, California and Cadence Design, of San Jose, California automatically route conductors and locate components on a semiconductor chip using well established rules of design as well as libraries of pre-stored design modules. Once the design for a semiconductor circuit has been completed, the resultant design, in a standardized electronic format (e.g., Opus, GDSII, or the like) may be transmitted to a semiconductor fabrication facility or "fab" for fabrication.

[0219] The foregoing description has provided by way of exemplary and non-limiting examples a full and informative description of the exemplary embodiment of this invention. However, various modifications and adaptations may become apparent to those skilled in the relevant arts in view of the foregoing description, when read in conjunction with the accompanying drawings and the appended claims. However, all such and similar modifications of the teachings of this invention will still fall within the scope of this invention.

CLAIMS:

1. An apparatus comprising

means for obtaining an image block unit comprising samples in color channels of one or two chrominance channels and one luminance channel;

means for reconstructing samples of said luminance channels of the image block unit;

means for predicting chrominance samples of at least one color channel of image block unit using at least two cross-component prediction models, each comprising their corresponding linear model parameters, and a luminance threshold value dividing the samples of said luminance channel between said at least two cross-component prediction models;

means for defining a weighting function for said at least two cross-component prediction models based on location of two or more chrominance samples and the luminance threshold value, or based on at least one luminance sample value and the luminance threshold value; and

means for applying a smoothing function for samples residing within an offset value from said luminance threshold value, wherein said smoothing function adjusts chrominance values of said samples residing within said offset value by applying said weighting function on the corresponding at least two linear model parameters of the at least two cross-component prediction models.

2. The apparatus according to claim 1, comprising

means for adjusting the value of the weighting function to remain within a range of [0,1].

3. The apparatus according to claim 1 or 2, comprising

means for defining the weighting function in sample domain.

4. The apparatus according to claim 3, wherein the smoothing function is defined in the sample domain as

$C(x, y, L_t, L, \alpha_1, \beta_1, \alpha_2, \beta_2) = w(x, y, L_t)(\alpha_2 L(x, y) + \beta_2) + (1 - w(x, y, L_t))(\alpha_1 L(x, y) + \beta_1)$, where w is the weighting function, x and y define the position of the sample, α_1 and β_1 are the linear model parameters for the first cross-component prediction model and α_2 and β_2 are the linear model parameters for the second cross-component prediction model, and L_t is the threshold luma value.

5. The apparatus according to claim 4, wherein the weighting function w is defined in the sample domain as a linear weighting function

$$w(x, y, L_t) = \gamma \cdot (L(x, y) - L_t + R),$$

where γ is a scaling term and R is the range of the reference luma samples.

6. The apparatus according to claim 4, wherein the weighting function w is defined in the sample domain as a linear weighting function

$$w(x, y, \mu, \sigma) = e^{-0.5 \left(\frac{L(x, y) - \mu}{\sigma} \right)^2},$$

where $\mu = L_t + R$ and $\sigma = \gamma R$.

7. The apparatus according to any preceding claim, wherein the smoothing function is applied in a parametric vicinity of the threshold value L_t .

8. The apparatus according to any preceding claim, wherein the smoothing function is applied in the sample domain in unsymmetric manner only when $L(x, y) \leq L_t + T_0$ & $L(x, y) \geq L_t - T_1$ where $T_0 \neq T_1$.

9. The apparatus according to claim 1 or 2, wherein the weighting function $w(\cdot)$ is defined in spatial domain as $w(\Delta x, \Delta y)$ where Δx and Δy define distance to the nearest boundary sample (x_B, y_B) .

10. The apparatus according to claim 9, wherein the smoothing function C is defined in the spatial domain as

$$C(x, y, L_t, L, \alpha_1, \beta_1, \alpha_2, \beta_2) = w(\Delta x, \Delta y)(\alpha_2 L(x, y) + \beta_2) + (1 - w(\Delta x, \Delta y))(\alpha_1 L(x, y) + \beta_1).$$

11. The apparatus according to claim 9 or 10, wherein the smoothing function is applied only in the vicinity of the spatial boundary of two luma-to-chroma models, when $\sqrt{\Delta_x^2 + \Delta_y^2} \leq T$ where T is a fixed, inferred or signalled parameter.

12. The apparatus according to claim 9 or 10, wherein the smoothing function is applied in the spatial domain in anisotropic manner only when $\sqrt{(\omega_0 \Delta_x)^2 + (\omega_1 \Delta_y)^2} \leq T$, where ω_0, ω_1 are fixed, inferred or signalled by the encoder.

13. The apparatus according to any preceding claim, further comprising
means for deriving different linear models for a beginning and an end of a luma spectrum of the multi-model cross-component prediction model.

14. The apparatus according to any preceding claim, further comprising
means for performing a rounding and/or a clipping operation on the chroma samples values obtained as a result of the smoothing function

15. A method comprising
obtaining an image block unit comprising samples in color channels of one or two chrominance channels and one luminance channel;
reconstructing samples of said luminance channels of the image block unit;
predicting chrominance samples of at least one color channel of image block unit using at least two cross-component prediction models, each comprising their corresponding linear model parameters, and a luminance threshold value dividing the samples of said luminance channel between said at least two cross-component prediction models;
defining a weighting function for said at least two cross-component prediction models based on location of said two or more chrominance samples and the luminance threshold value; and
applying a smoothing function for samples residing within an offset value from said luminance threshold value, wherein said smoothing function adjusts chrominance values of said samples residing within said offset value by applying said weighting function on the

corresponding at least two linear model parameters of the at least two cross-component prediction models.

1/10

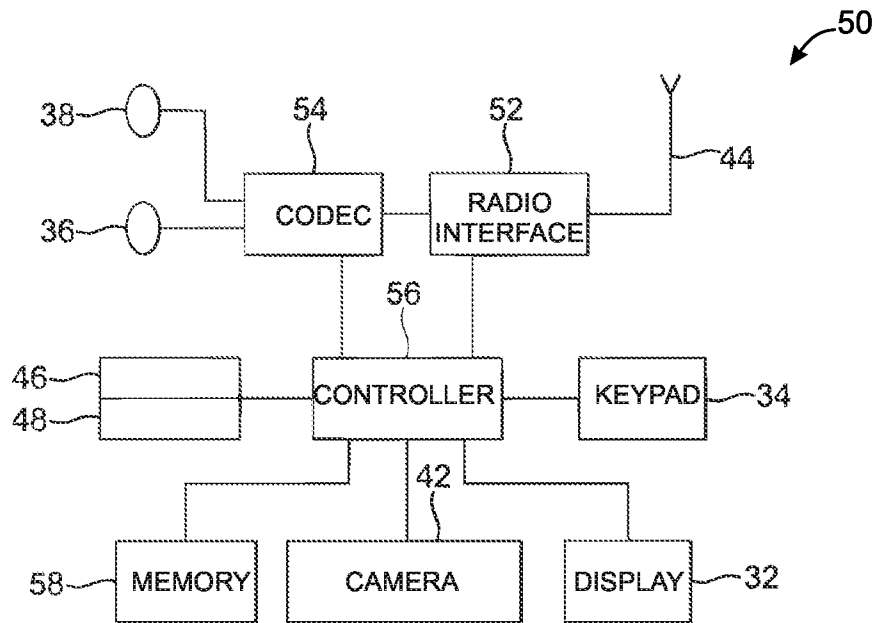


Fig. 1

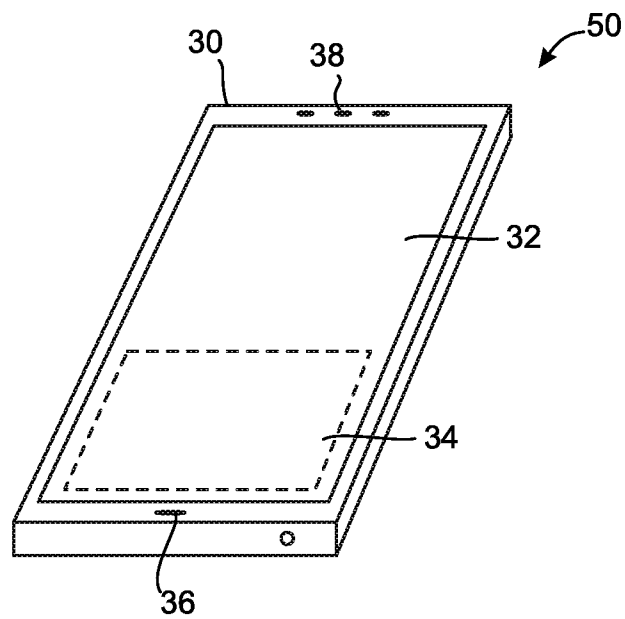


Fig. 2

2/10

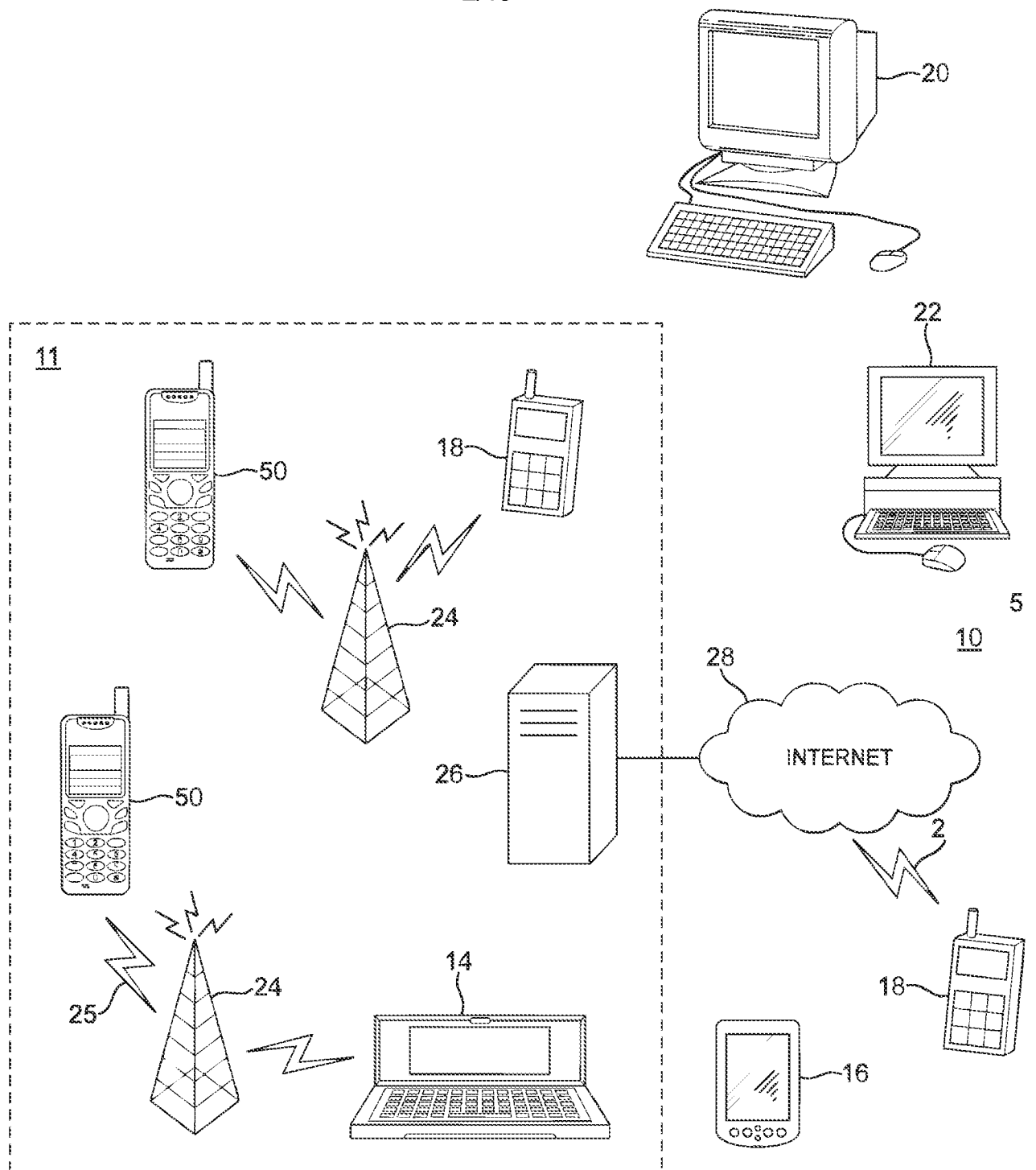


Fig. 3

3/10

/

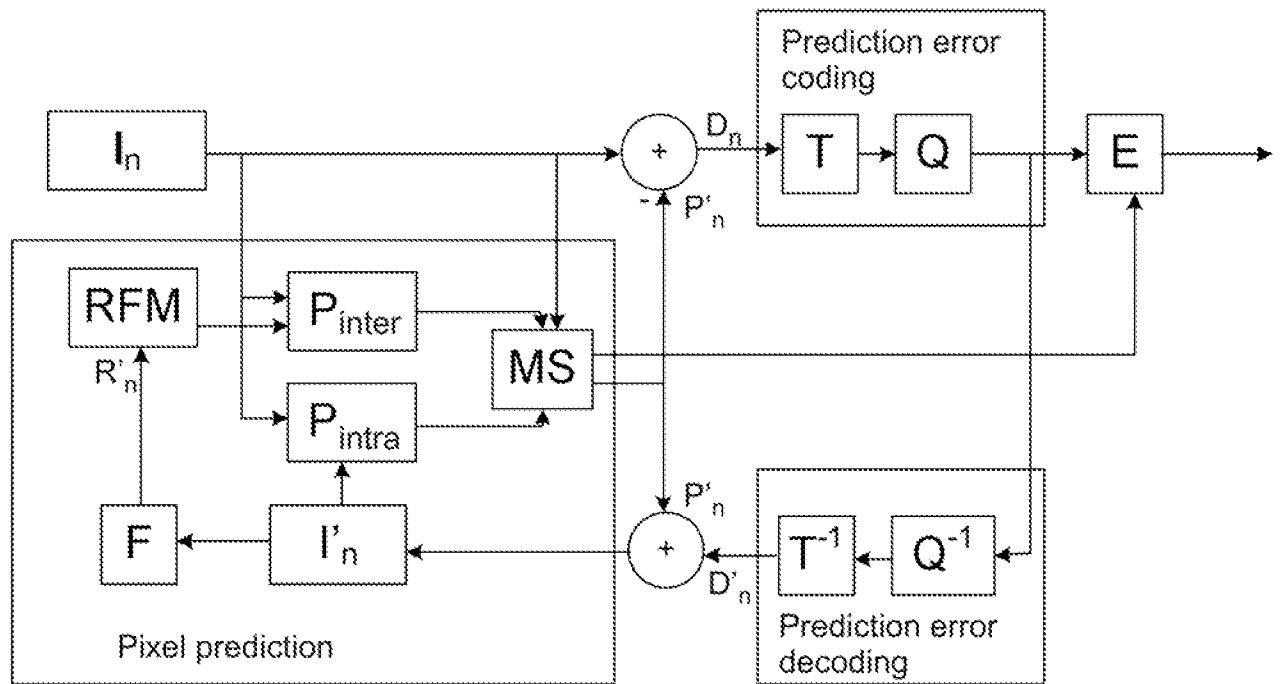


Fig. 4a

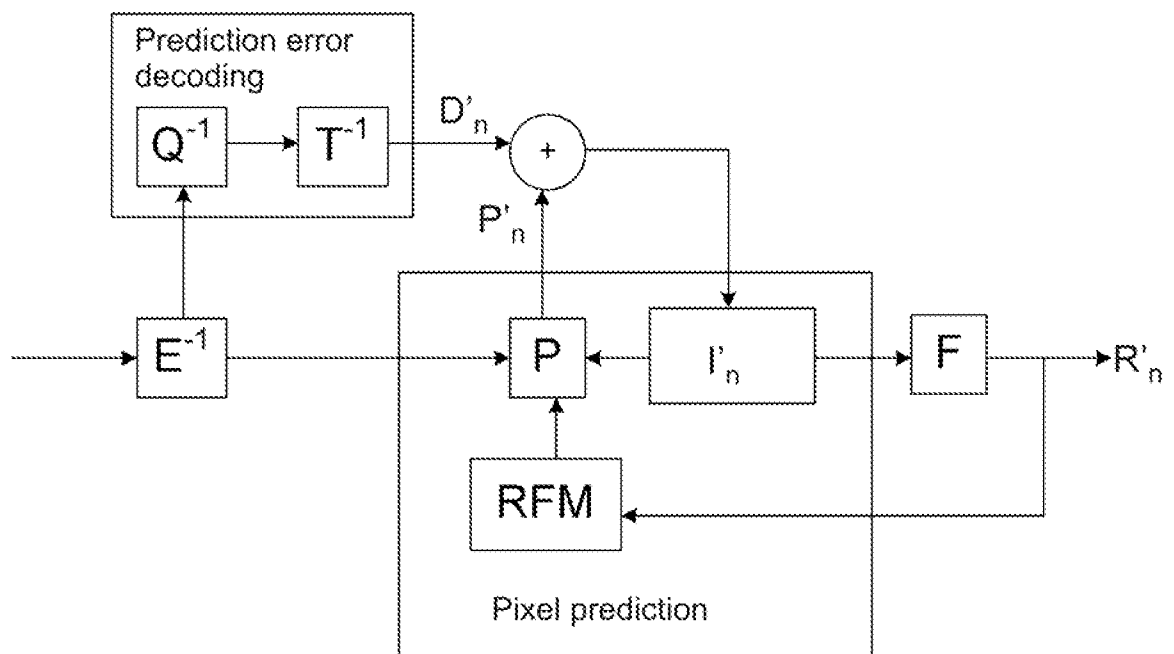


Fig. 4b

4/10

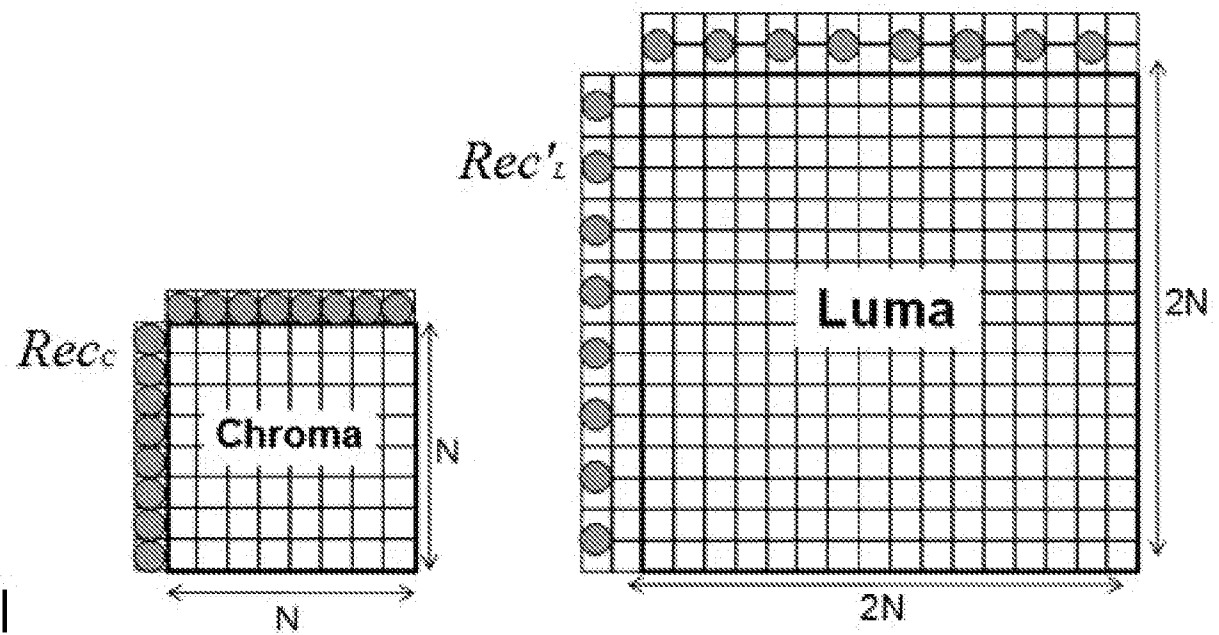


Fig. 5

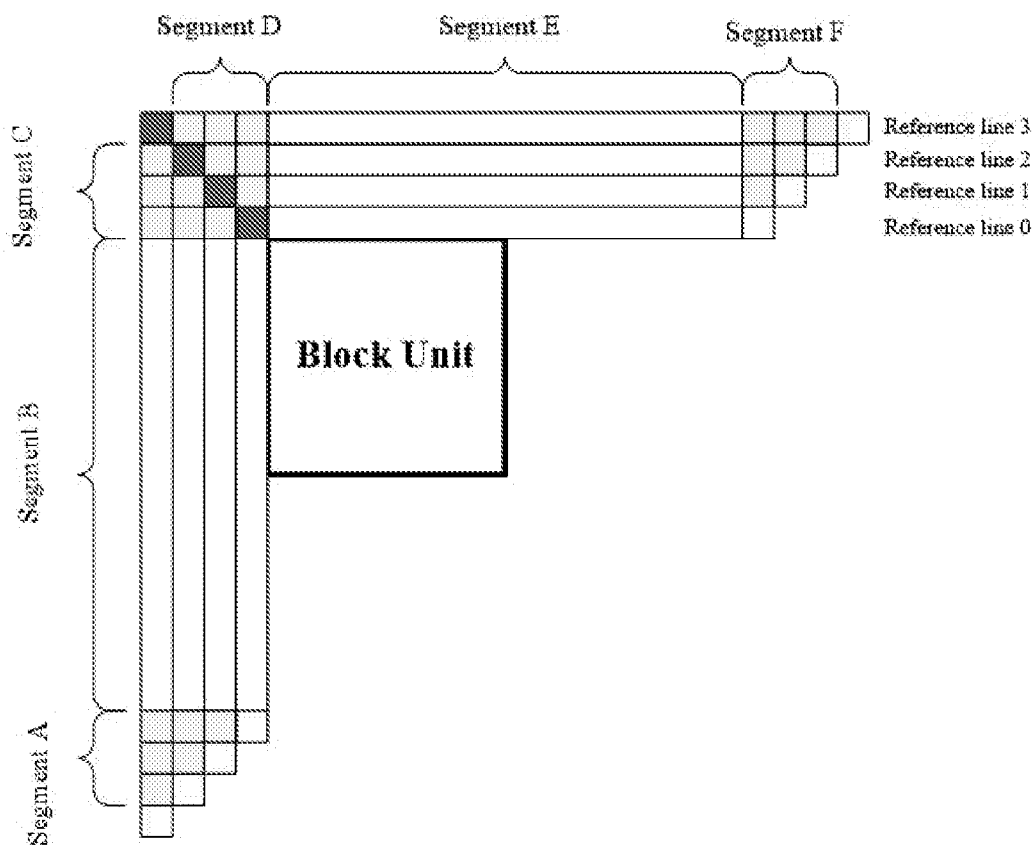
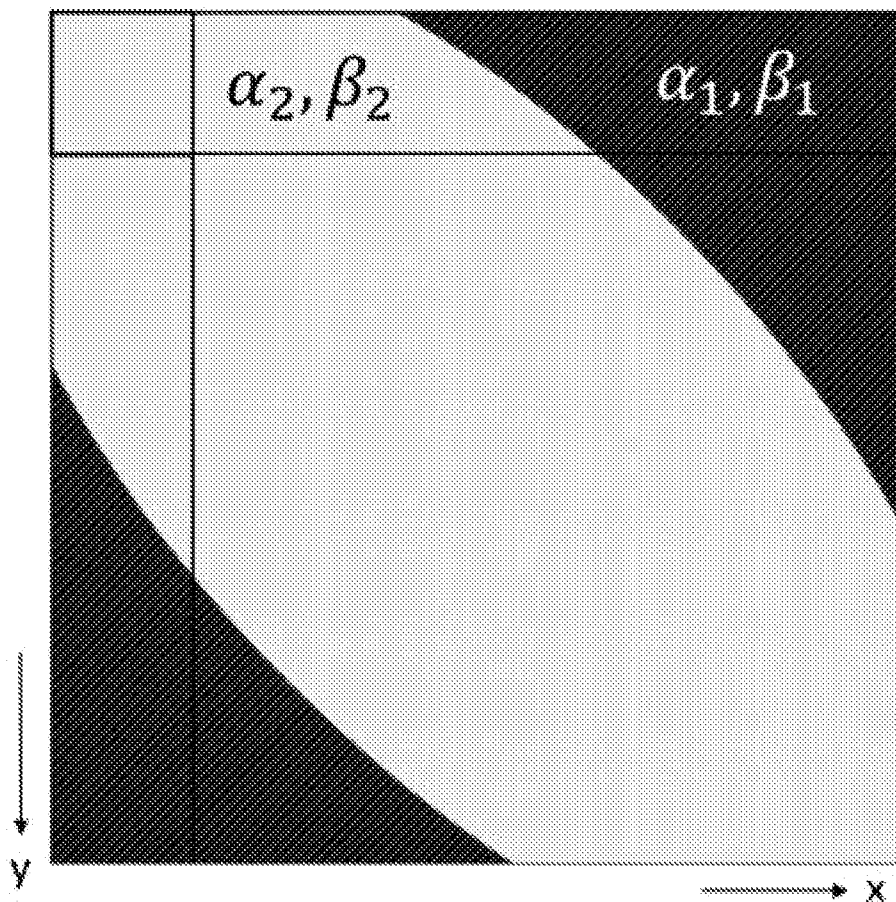
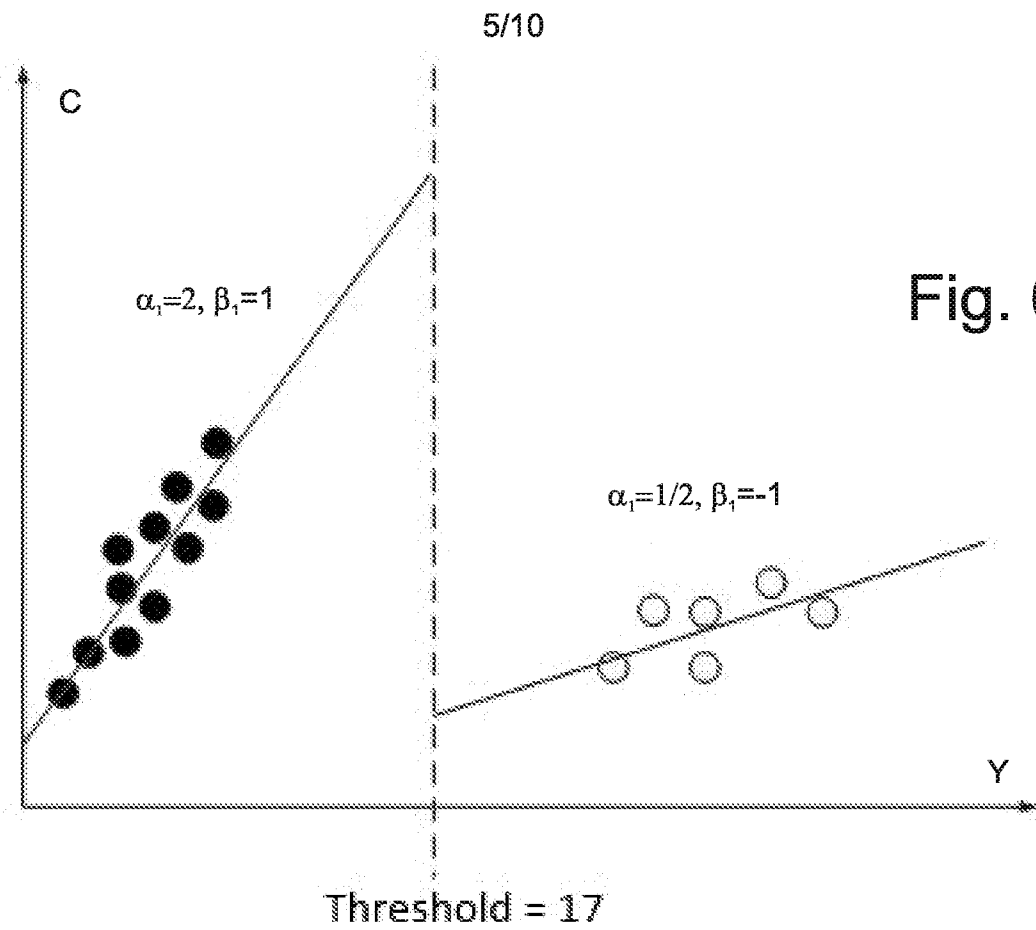


Fig. 7



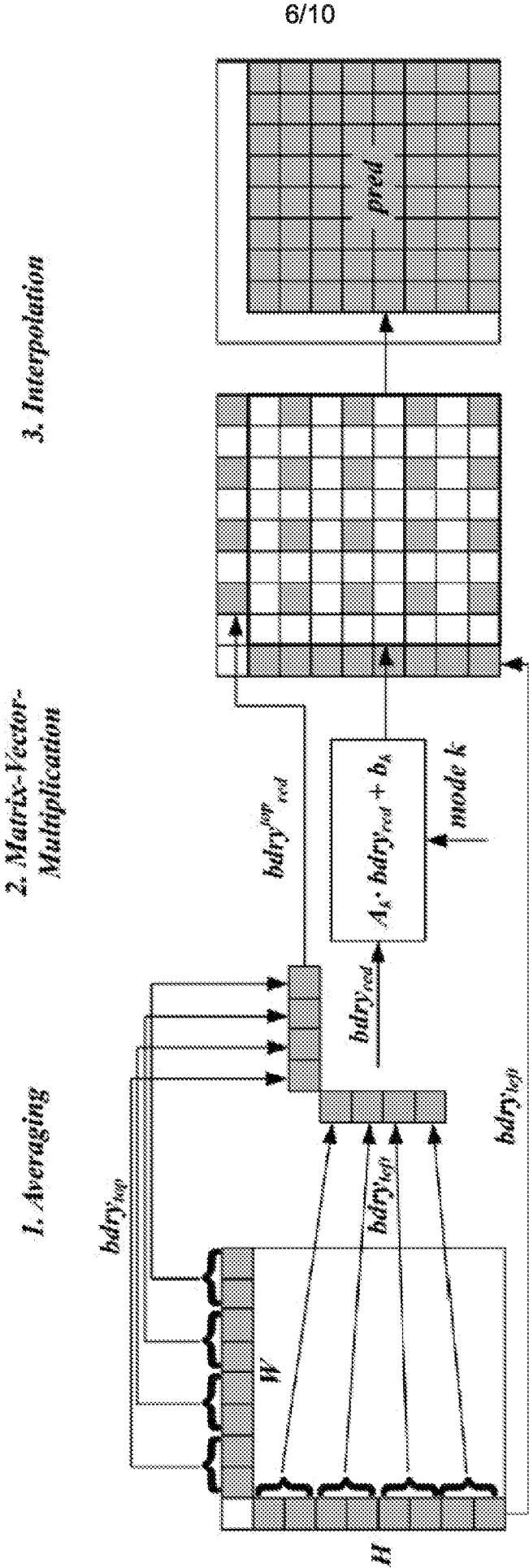


Fig. 8

7/10

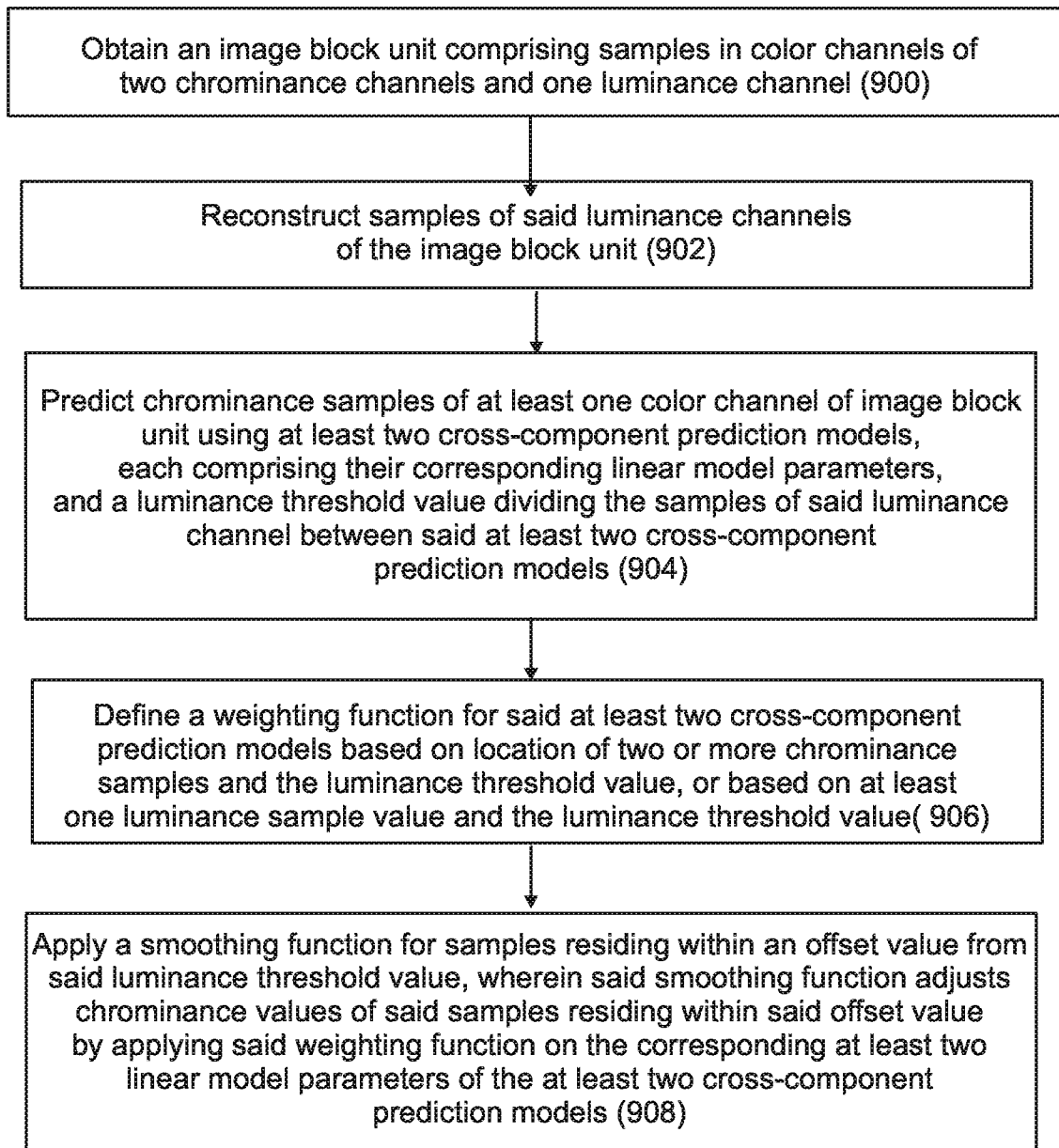


Fig. 9

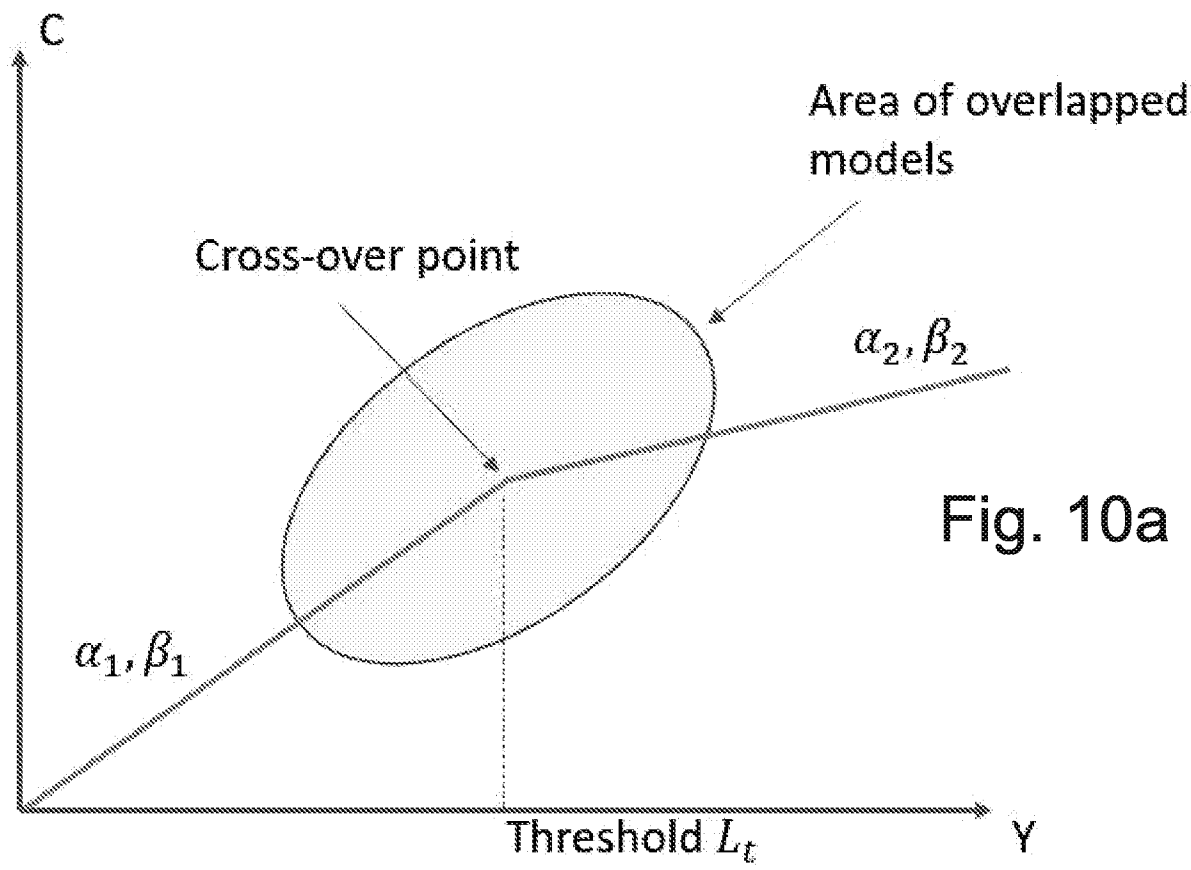


Fig. 10a

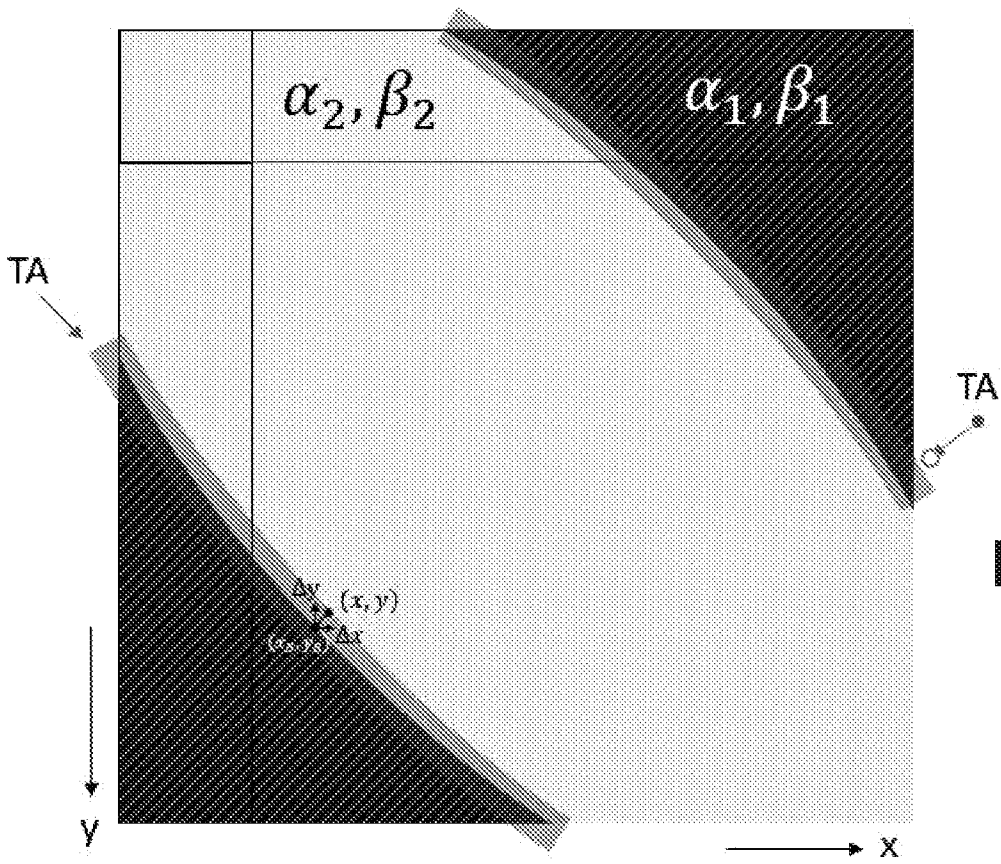


Fig. 10b

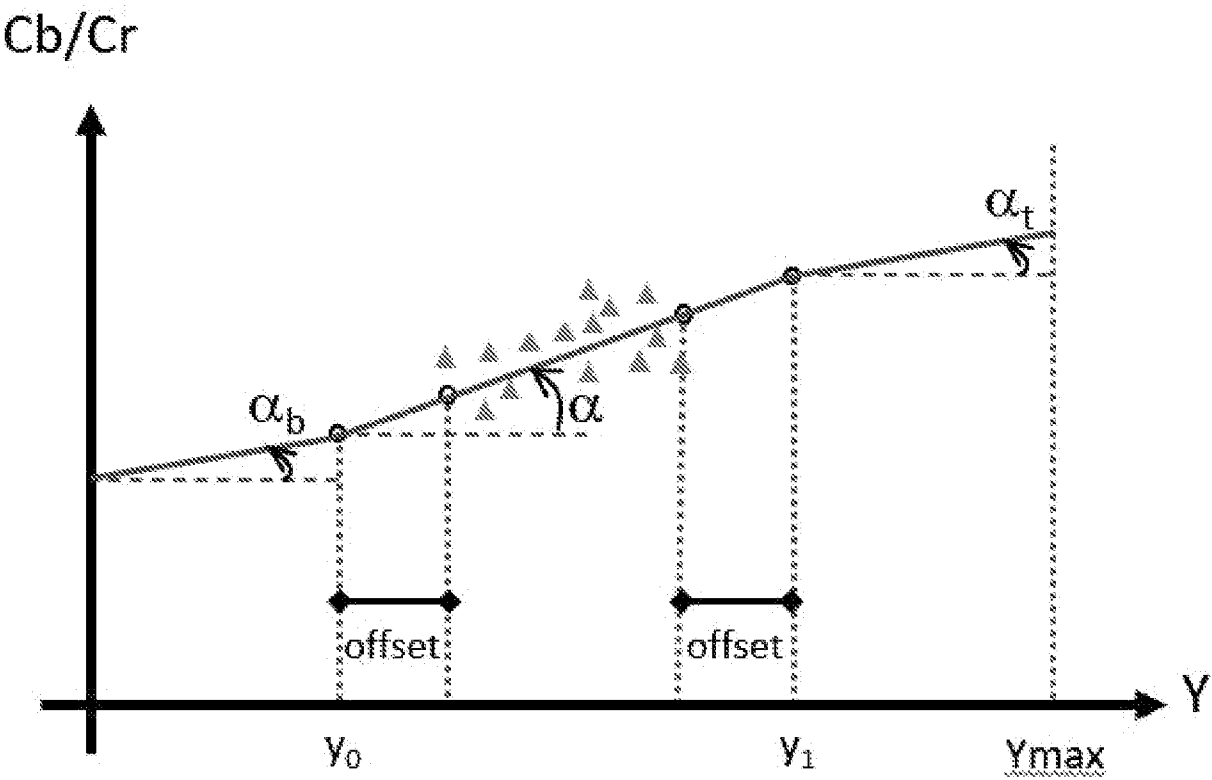


Fig. 11

10/10

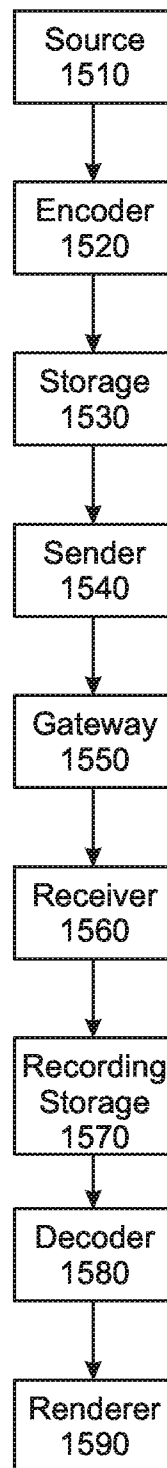


Fig. 12

INTERNATIONAL SEARCH REPORT

International application No.

PCT/FI2022/050735

A. CLASSIFICATION OF SUBJECT MATTER

See extra sheet

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)

IPC: H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
FI, SE, NO, DK

Electronic database consulted during the international search (name of database and, where practicable, search terms used)

EPODOC, EPO-Internal full-text databases, Full-text translation databases from Asian languages, WPIAP, IPRally, XP3GPP, XPAIP, XPESP, XPETSI, XPI3E, XPIEE, XPIETF, XPIOP, XPIPCOM, XPJPEG, XPMISC, XPOAC, XPRD, XPTK, COMPD, INSPEC, TDB, Internet, MPEG document management system

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2018077426 A1 (ZHANG KAI [US] et al.) 15 March 2018 (15.03.2018) paragraphs [0044], [0092], [0133]-[0135], [0149], [0151]-[0155]; Figs. 7A-7E	1-15
X	WO 2019131349 A1 (SHARP KK [JP]) 04 July 2019 (04.07.2019) Fig. 9 & machine translation into English by EPO [online] EPOQUENET TXPMTEA paragraphs [0113]-[0115], [0131]	1-15
A	US 2020366933 A1 (ZHANG KAI [US] et al.) 19 November 2020 (19.11.2020) paragraphs [0081]-[0083], [1210]-[1219]; Fig. 2	1-15

☐ Further documents are listed in the continuation of Box C.
 ☒ See patent family annex.

* Special categories of cited documents:	"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
"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"D" document cited by the applicant in the international application	"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
"E" earlier application or patent but published on or after the international filing date	"&" document member of the same patent family
"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	

Date of the actual completion of the international search 29 March 2023 (29.03.2023)	Date of mailing of the international search report 29 March 2023 (29.03.2023)
Name and mailing address of the ISA/ FI Finnish Patent and Registration Office FI-00091 PRH, FINLAND Facsimile No. +358 29 509 5328	Authorized officer Kai Willner Telephone No. +358 29 509 5000

INTERNATIONAL SEARCH REPORT
Information on Patent Family Members

International application No.
PCT/FI2022/050735

US 2018077426 A1	15/03/2018	JP 7044765 B2	30/03/2022
		BR 112019004544 A2	28/05/2019
		TW I776818 B	11/09/2022
		SG 11201900967X A	29/04/2019
		WO 2018053293 A1	22/03/2018
		EP 3513559 A1	24/07/2019
		EP 3513559 B1	04/08/2021
		US 10652575 B2	12/05/2020
		KR 20190046852 A	07/05/2019
		CN 109716771 A	03/05/2019
		JP 2019530330 A	17/10/2019
		CN 109716771 B	28/06/2022
		TW 201817236 A	01/05/2018
		ES 2884375 T3	10/12/2021

WO 2019131349 A1	04/07/2019	None	
------------------	------------	------	--

US 2020366933 A1	19/11/2020	ZA 202105403 B	26/01/2022
		CN 112997484 A	18/06/2021
		SG 11202108209Y A	30/08/2021
		SG 11202108289P A	30/08/2021
		AU 2022268297 A1	15/12/2022
		JP 2022517136 A	04/03/2022
		US 2020382769 A1	03/12/2020
		US 2021211681 A1	08/07/2021
		US 2021152837 A1	20/05/2021
		CN 115720268 A	28/02/2023
		US 10999581 B2	04/05/2021
		KR 20210087928 A	13/07/2021
		CA 3121671 A1	11/06/2020
		WO 2020094058 A1	14/05/2020
		US 10939128 B2	02/03/2021
		CA 3128769 A1	27/08/2020
		KR 20210091182 A	21/07/2021
		US 2020366910 A1	19/11/2020
		KR 20210089131 A	15/07/2021
		WO 2020169101 A1	27/08/2020
		EP 3861736 A1	11/08/2021
		JP 2022521698 A	12/04/2022
		US 11438598 B2	06/09/2022
		WO 2020192642 A1	01/10/2020
		US 2020382800 A1	03/12/2020
		WO 2020192717 A1	01/10/2020

INTERNATIONAL SEARCH REPORT
Information on Patent Family Members

International application No.
PCT/FI2022/050735

CN 112997491 A	18/06/2021
JP 2022506283 A	17/01/2022
US 2021400260 A1	23/12/2021
CA 3128429 A1	27/08/2020
CN 113632474 B	09/12/2022
WO 2020094067 A1	14/05/2020
US 2021344902 A1	04/11/2021
SG 11202105759Q A	29/06/2021
AU 2020226565 A1	19/08/2021
EP 3861738 A1	11/08/2021
EP 3903493 A1	03/11/2021
JP 7212157 B2	24/01/2023
KR 20210126009 A	19/10/2021
EP 3861739 A1	11/08/2021
WO 2020108591 A1	04/06/2020
US 11057642 B2	06/07/2021
US 2021258571 A1	19/08/2021
KR 102472756 B1	01/12/2022
JP 2022508294 A	19/01/2022
US 11025915 B2	01/06/2021
US 11019344 B2	25/05/2021
US 11595687 B2	28/02/2023
WO 2020094061 A1	14/05/2020
BR 112021015017 A2	26/10/2021
US 11438581 B2	06/09/2022
JP 2022506277 A	17/01/2022
CA 3128769 C	24/01/2023
WO 2020094059 A1	14/05/2020
ZA 202103763 B	26/10/2022
CN 113439437 A	24/09/2021
CN 113491121 A	08/10/2021
BR 112021010428 A2	24/08/2021
WO 2020094066 A1	14/05/2020
JP 2022506285 A	17/01/2022
EP 3903482 A1	03/11/2021
CN 113767631 A	07/12/2021
US 2020359051 A1	12/11/2020
CN 112956199 A	11/06/2021
CN 113170122 A	23/07/2021
CN 112997492 A	18/06/2021
CN 113039791 A	25/06/2021
CN 113170168 A	23/07/2021
US 2021297694 A1	23/09/2021
BR 112021015017 B1	26/07/2022
JP 2022506286 A	17/01/2022
US 11115655 B2	07/09/2021
WO 2020114445 A1	11/06/2020

INTERNATIONAL SEARCH REPORT
Information on Patent Family Members

International application No.
PCT/FI2022/050735

US 2021258572 A1	19/08/2021
KR 20210089133 A	15/07/2021
WO 2020094057 A1	14/05/2020
CN 113632474 A	09/11/2021
JP 7123268 B2	22/08/2022
US 2020366896 A1	19/11/2020
EP 3871415 A1	01/09/2021
KR 20210127157 A	21/10/2021
US 10979717 B2	13/04/2021
AU 2019391197 A1	17/06/2021
WO 2020169102 A1	27/08/2020
EP 3861728 A1	11/08/2021
ZA 202105356 B	25/01/2023
CN 113491121 B	06/12/2022
JP 2023033408 A	10/03/2023
US 2020252619 A1	06/08/2020
AU 2020226566 A1	19/08/2021
KR 20210089132 A	15/07/2021
CN 112997488 A	18/06/2021

CLASSIFICATION OF SUBJECT MATTER

IPC

H04N 19/186 (2014.01)**H04N 19/176** (2014.01)**H04N 19/196** (2014.01)**H04N 19/593** (2014.01)**H04N 19/11** (2014.01)