(12) UK Patent (19) GB (11) 2571313 (13) B

(45) Date of B Publication 21.09.2022

(54) Title of the Invention: New sample sets and new down-sampling schemes for linear component sample prediction

(51) INT CL: ***H04N 19/186*** (2014.01) ***H04N 19/50*** (2014.01)

(21) Application No: **1802972.8**

(22) Date of Filing: **23.02.2018**

(43) Date of A Publication **28.08.2019**

(56) Documents Cited:
EP 3013049 A1 **US 20170359595 A1**
J. Chen et. al., "Algorithm Description of Joint Exploration Test Model 7 (JEM 7)", published 2017, JVET. Available from https://jvet-experts.org
J. Chen et. al., "Chroma intra prediction by reconstructed luma samples", published 2010, JCTVC, available from phenix.int-evry.fr/jct/

(58) Field of Search:
As for published application 2571313 A viz:
INT CL **G06T, H04N**
Other: **EPODOC, WPI, INSPEC and Patent Fulltext** updated as appropriate

Additional Fields
Other: **None**

(72) Inventor(s):
Guillaume Laroche
Jonathan Taquet
Patrice Onno
Christophe Gisquet

(73) Proprietor(s):
Canon Kabushiki Kaisha
30-2 Shimomaruko 3-Chome, Ohta-ku, 146-8501 Tokyo, Japan

(74) Agent and/or Address for Service:
Canon Europe Limited
European Intellectual Property Group, 4 Roundwood Avenue, Stockley Park, Uxbridge, UB11 1AF, United Kingdom

GB 2571313 B

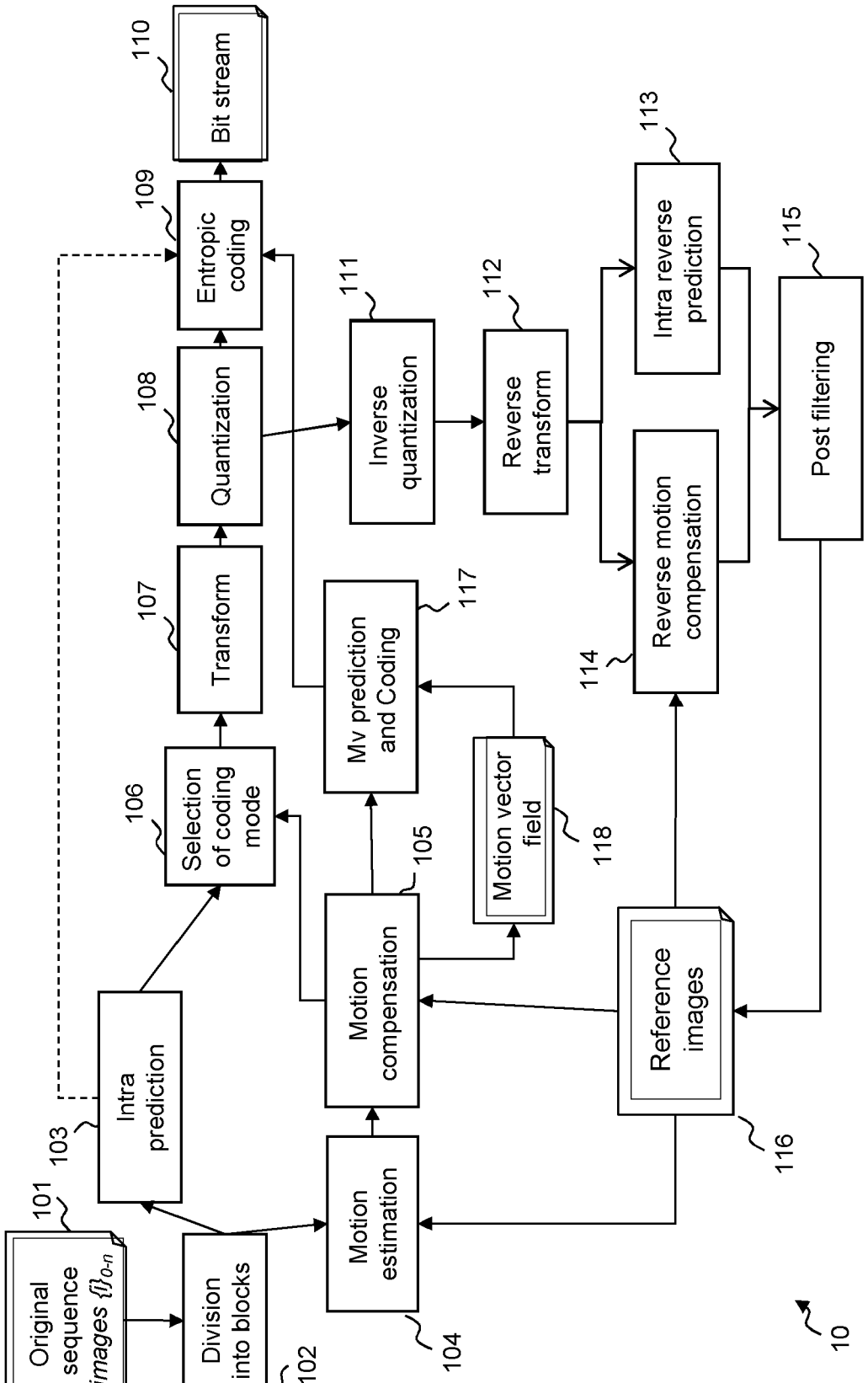

Original sequence images {I}0-n
101
Division into blocks
102
Intra prediction
103
Motion estimation
104
Motion compensation
105
Selection of coding mode
106
Transform
107
Quantization
108
Entropic coding
109
Bit stream
110
Inverse quantization
111
Reverse transform
112
Intra reverse prediction
113
Reverse motion compensation
114
Post filtering
115
Reference images
116
Mv prediction and Coding
117
Motion vector field
118
10

Fig. 1

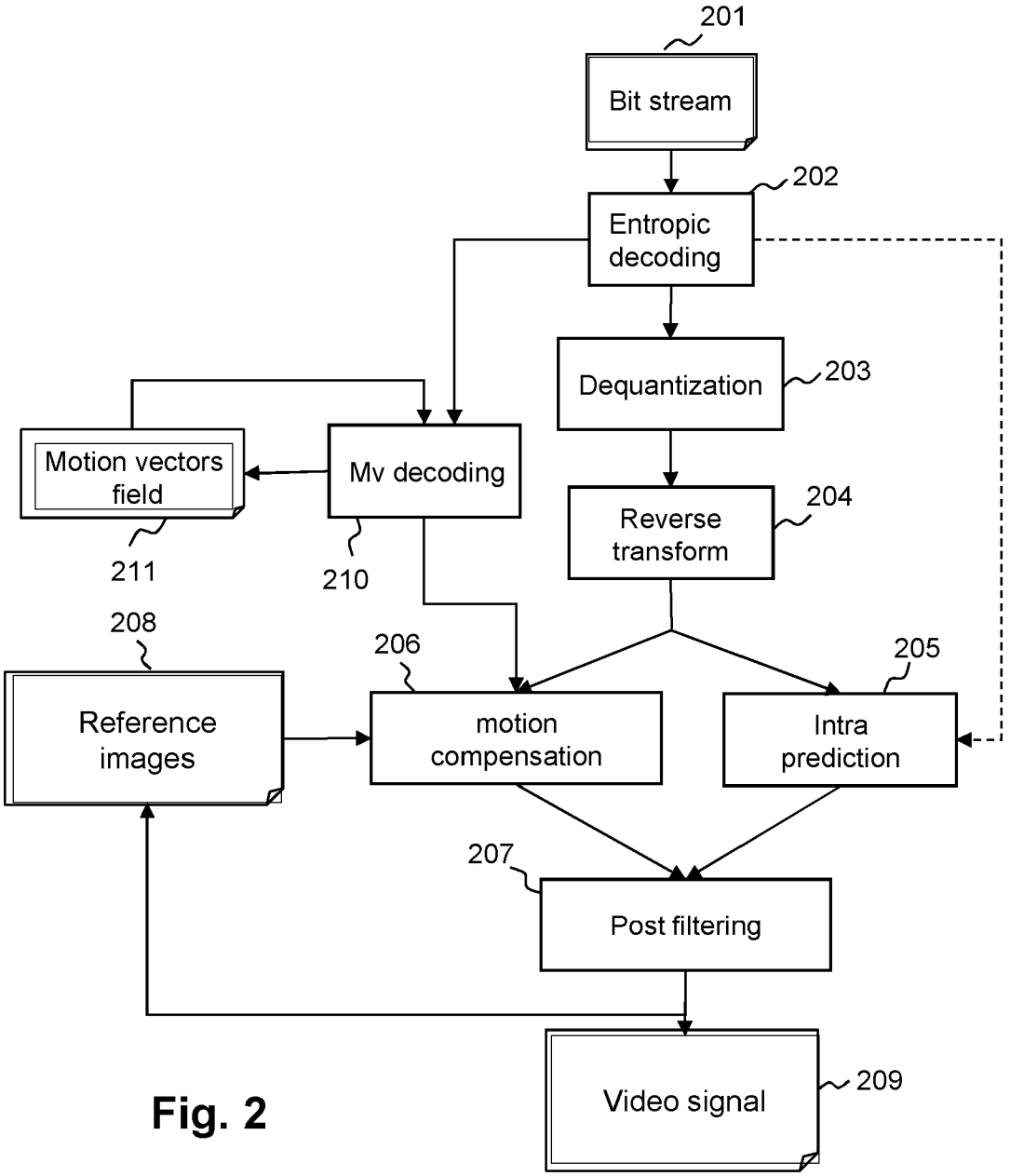
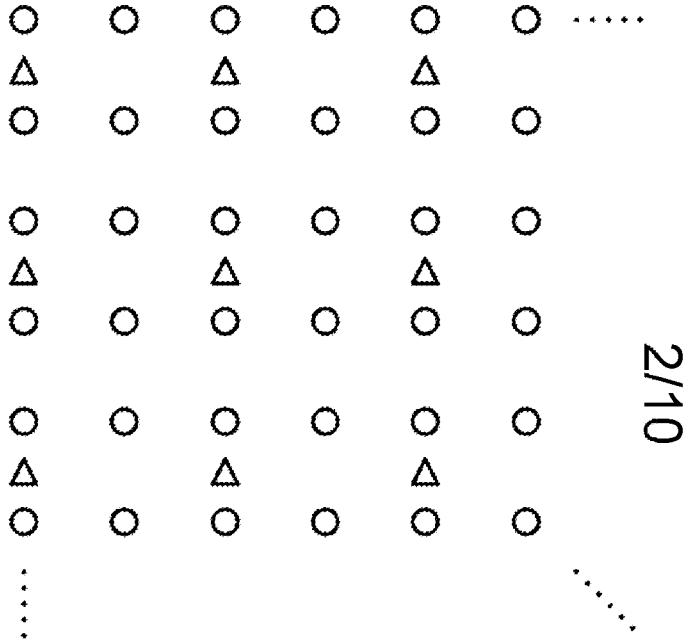


Fig. 2



Luma samples

Chroma samples

Fig. 3

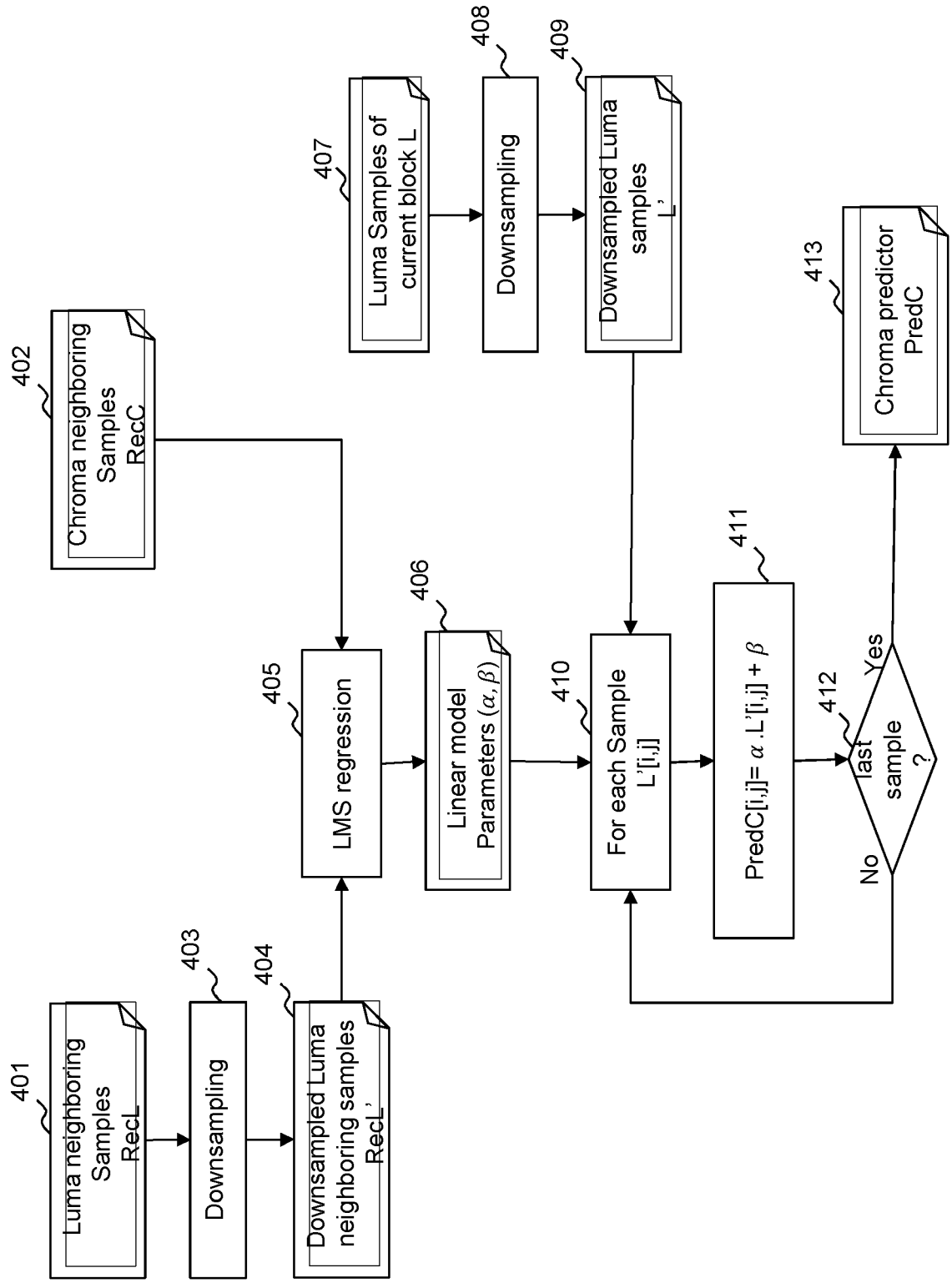

401
Luma neighboring Samples RecL
403
Downsampling
404
Downsampled Luma neighboring samples RecL'
402
Chroma neighboring Samples RecC
405
LMS regression
406
Linear model Parameters (α, β)
407
Luma Samples of current block L
408
Downsampling
409
Downsampled Luma samples L'
410
For each Sample L'[i,j]
411
PredC[i,j]= α .L'[i,j] + β
412
last sample ?
Yes
No
413
Chroma predictor PredC

Fig. 4

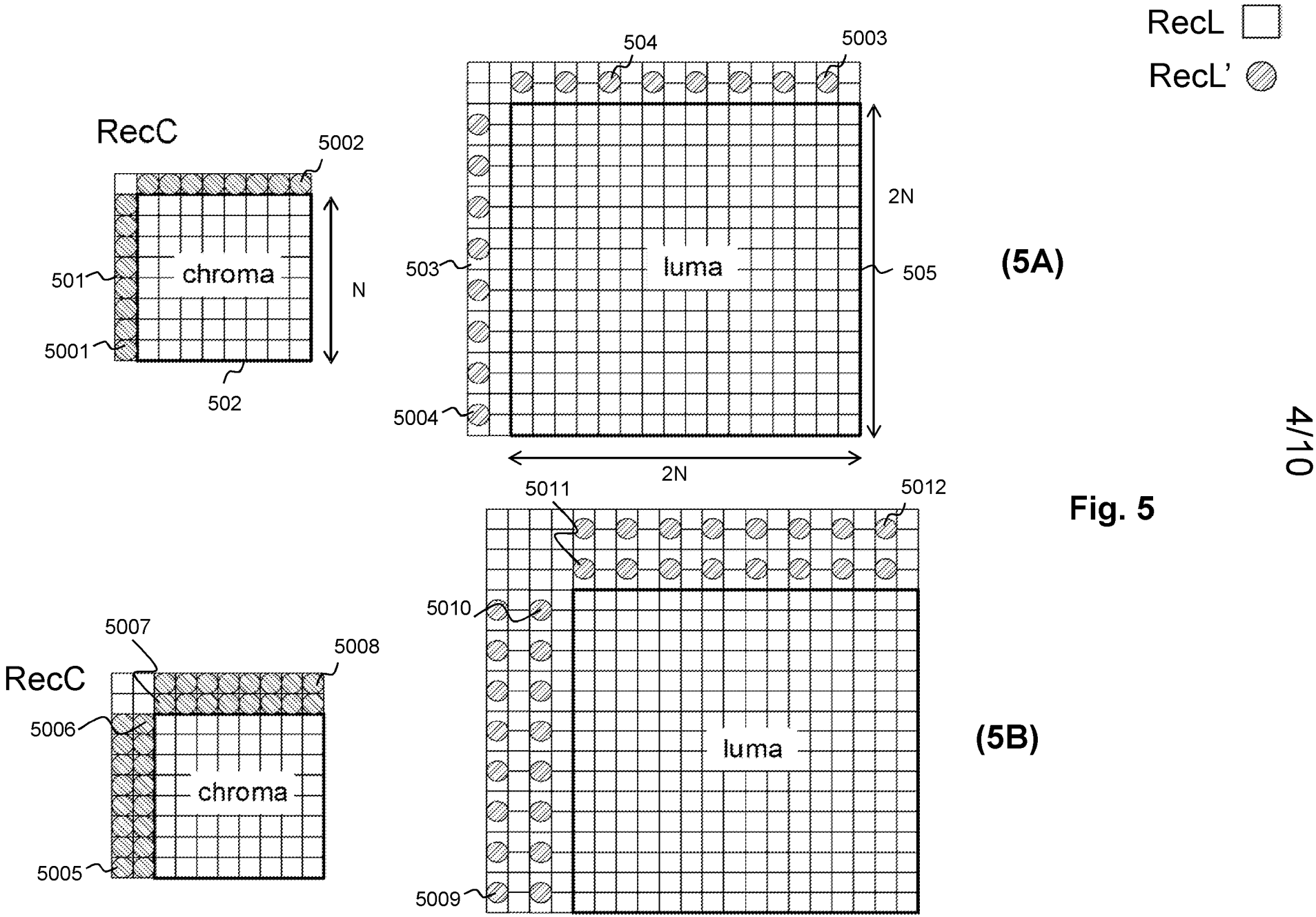

RecL
RecL'
RecC
5002
chroma
501
N
5001
502
504
5003
2N
luma
503
505
5004
2N
(5A)
5011
5012
5010
luma
5009
5007
5008
RecC
5006
chroma
5005
(5B)

Fig. 5

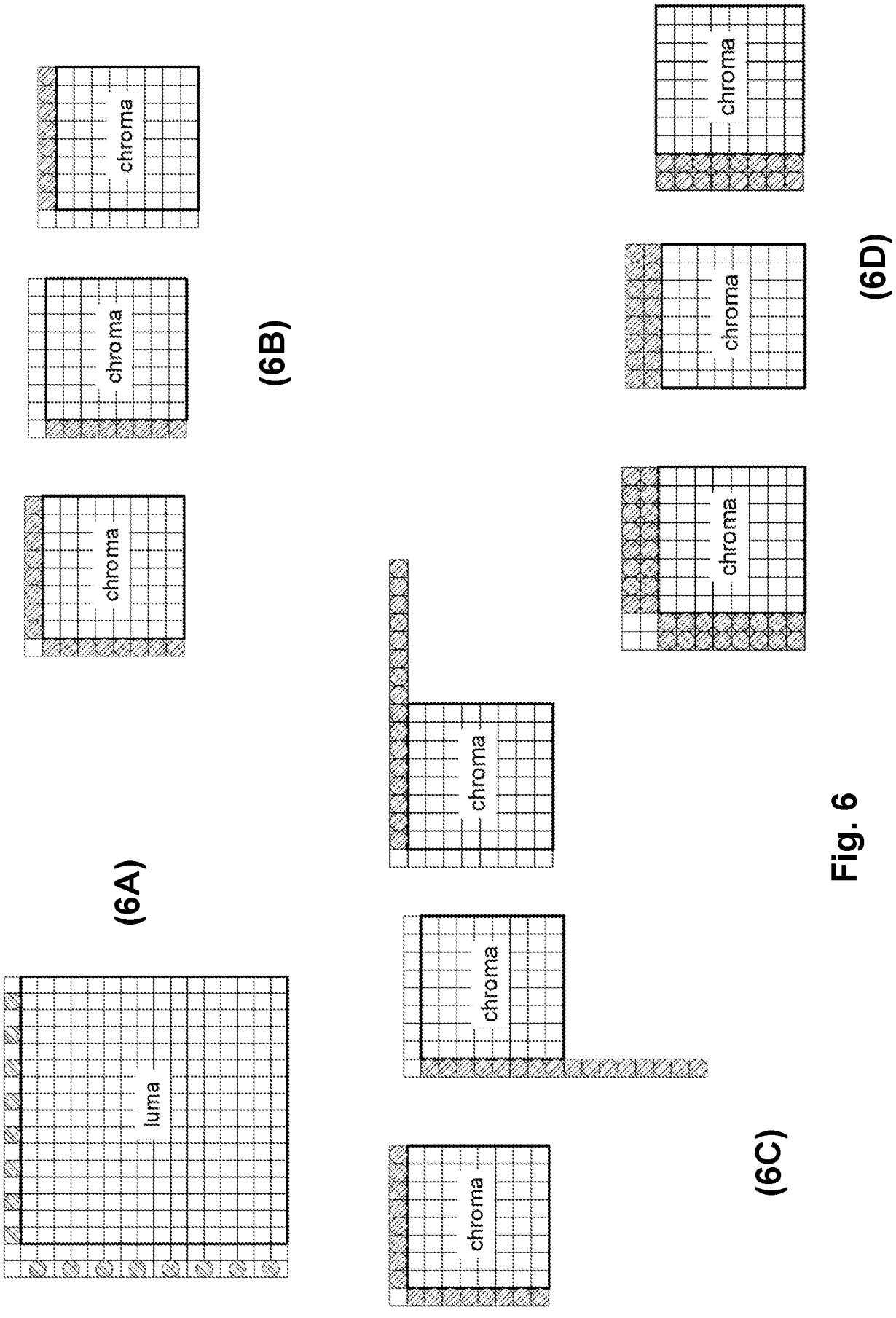
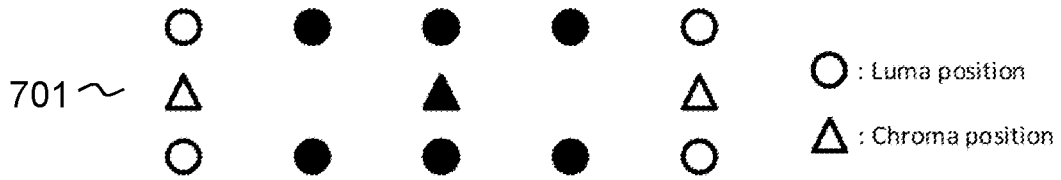

luma
chroma
chroma
chroma
chroma
chroma
chroma
chroma
chroma
(6A)
(6B)
(6C)
(6D)

Fig. 6


701
: Luma position
: Chroma position

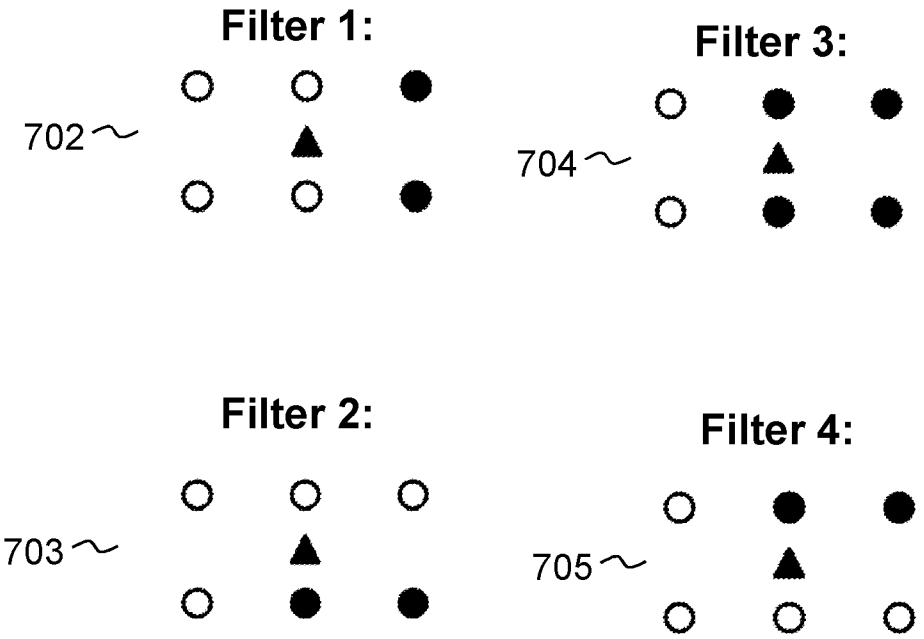

Filter 1:
702
Filter 3:
704
Filter 2:
703
Filter 4:
705

Fig. 7

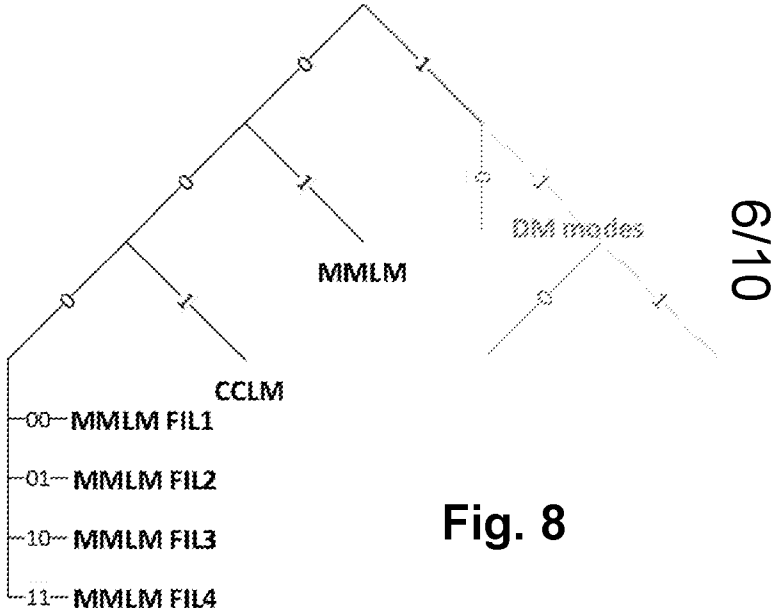

0
1
0
1
0
1
0
1
DM modes
MMLM
CCLM
00 MMLM FIL1
01 MMLM FIL2
10 MMLM FIL3
11 MMLM FIL4

Fig. 8

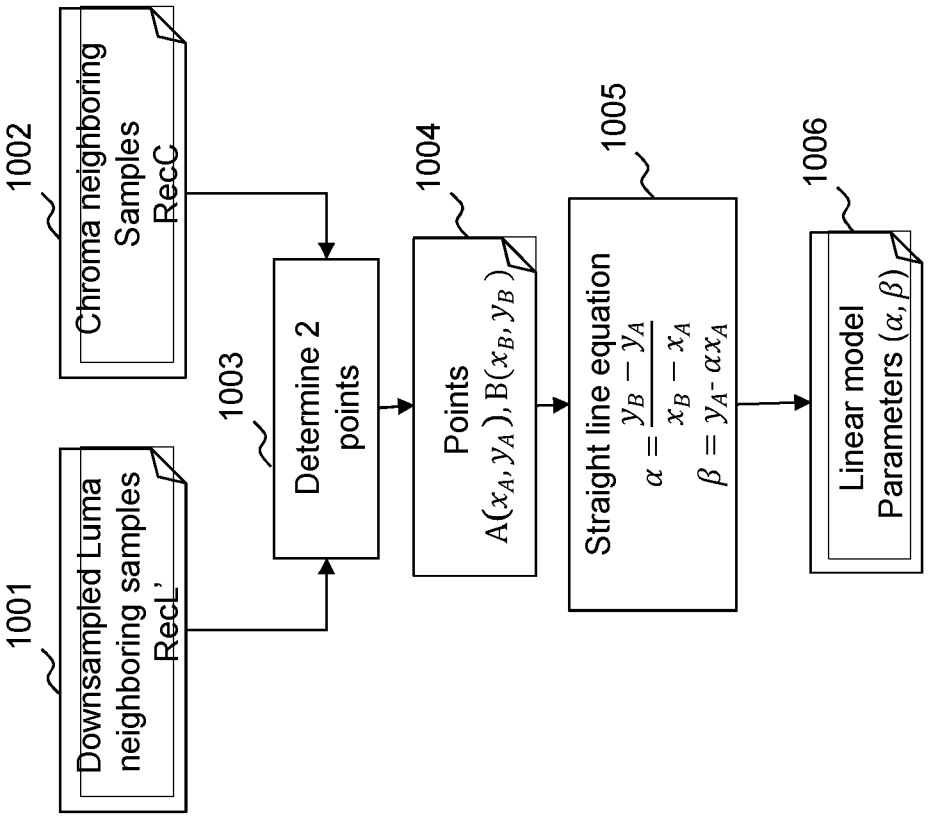

1001
Downsampled Luma neighboring samples RecL'
1002
Chroma neighboring Samples RecC
1003
Determine 2 points
1004
Points
$A(x_A, y_A), B(x_B, y_B)$
1005
Straight line equation
$\alpha = \frac{y_B - y_A}{x_B - x_A}$
$\beta = y_A - \alpha x_A$
1006
Linear model Parameters (α, β)

Fig. 10

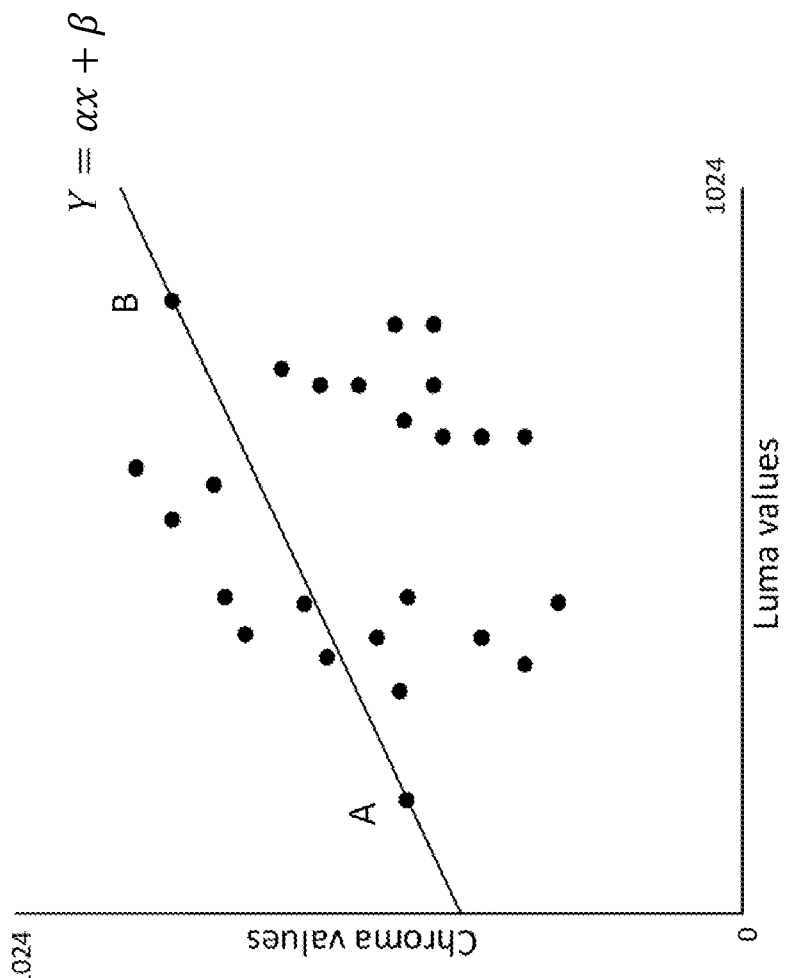

$Y = \alpha x + \beta$
B
A
1024
Chroma values
0
1024
Luma values

Fig. 9

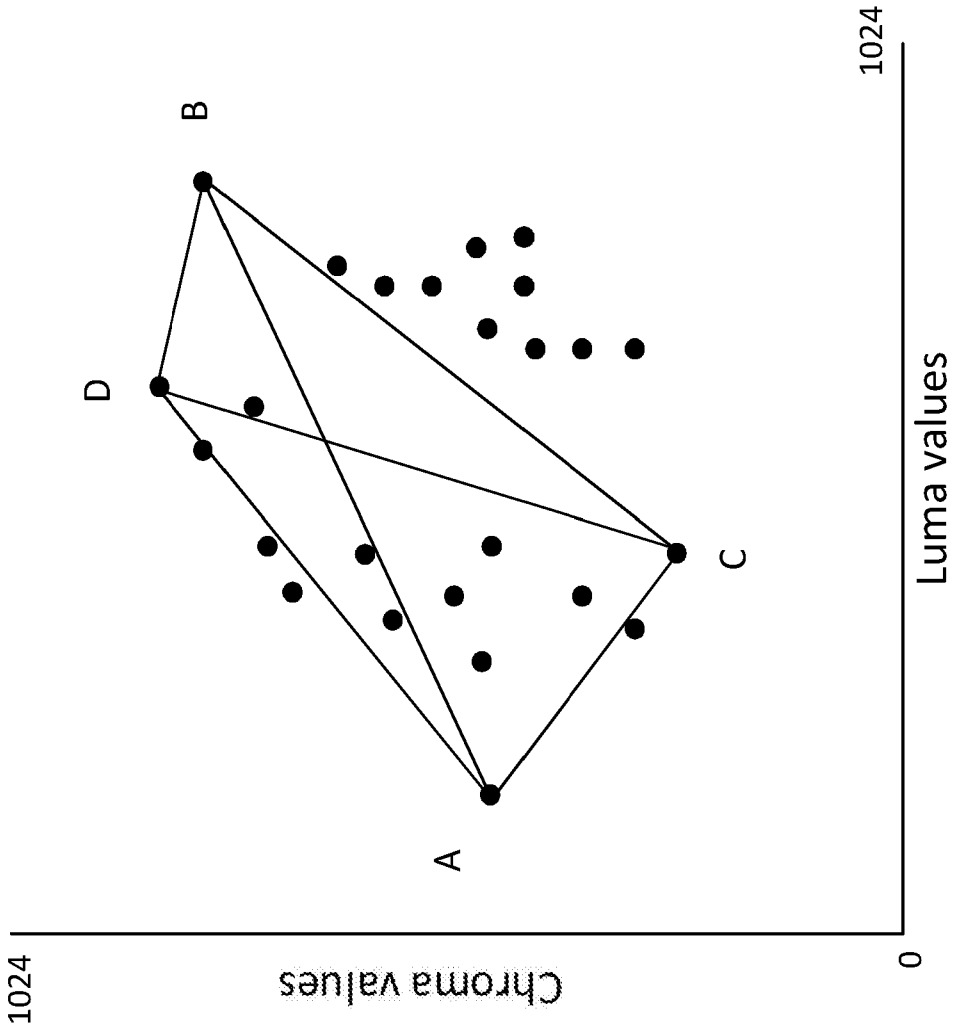


Fig. 11

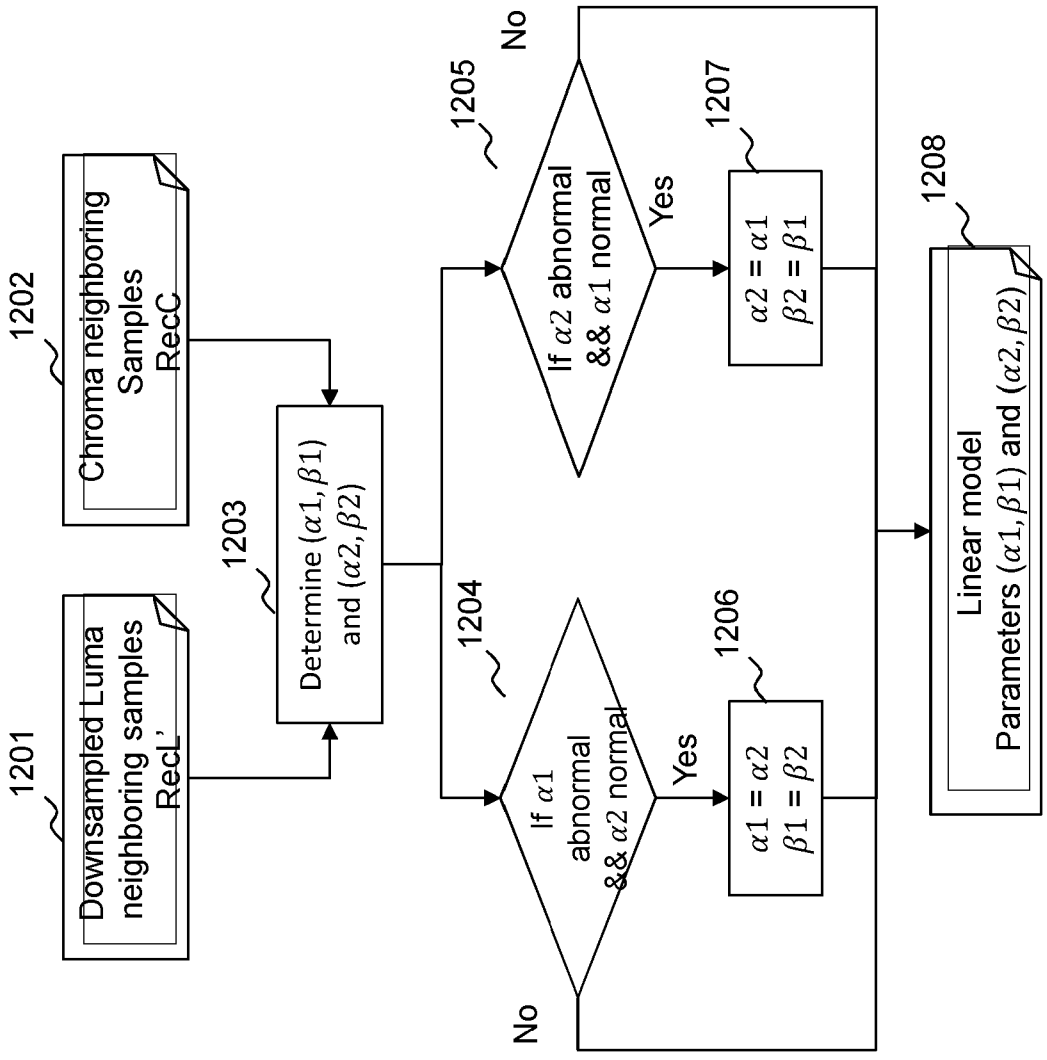

1201
Downsampled Luma neighboring samples RecL'
1202
Chroma neighboring Samples RecC
1203
Determine $(\alpha 1, \beta 1)$ and $(\alpha 2, \beta 2)$
1204
If $\alpha 1$ abnormal && $\alpha 2$ normal
Yes
No
1206
$\alpha 1 = \alpha 2$
$\beta 1 = \beta 2$
1205
If $\alpha 2$ abnormal && $\alpha 1$ normal
Yes
No
1207
$\alpha 2 = \alpha 1$
$\beta 2 = \beta 1$
1208
Linear model Parameters $(\alpha 1, \beta 1)$ and $(\alpha 2, \beta 2)$

Fig. 12

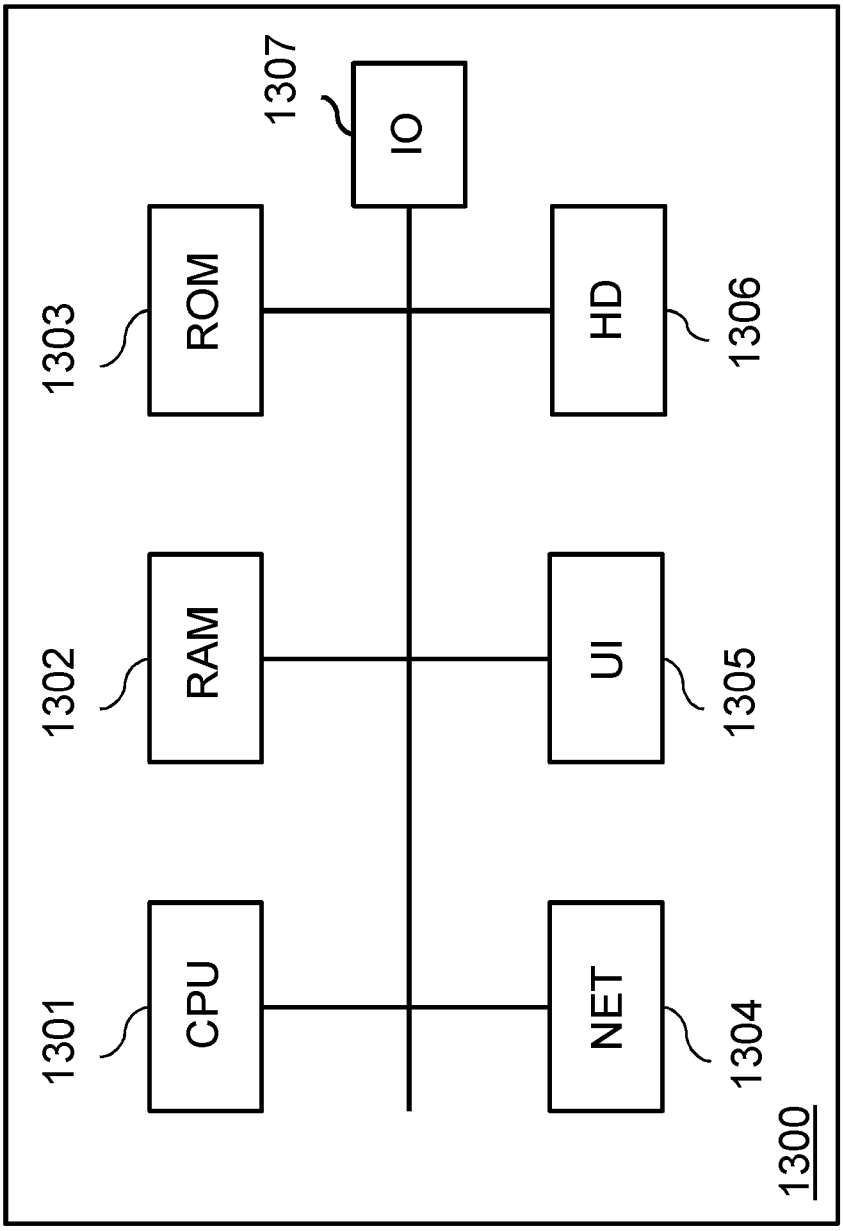

1307
IO
1303
ROM
HD
1306
1302
RAM
UI
1305
1301
CPU
NET
1304
1300

Fig. 13

NEW SAMPLE SETS AND NEW DOWN-SAMPLING SCHEMES FOR LINEAR COMPONENT SAMPLE PREDICTION

DOMAIN OF THE INVENTION

The present invention regards the encoding or decoding of blocks of a given video component, in particular the intra prediction of such component blocks or obtaining the samples of such blocks. The invention finds applications in obtaining blocks of a component, typically blocks of a chroma component, of video data from samples of another component, typically luma samples.

BACKGROUND OF THE INVENTION

Predictive encoding of video data is based on the division of frames into blocks of pixels. For each block of pixels, a predictor block is searched for in available data. The predictor block may be a block in a reference frame different from the current one in INTER coding modes, or generated from neighbouring pixels in the current frame in INTRA coding modes. Different encoding modes are defined according to different ways of determining the predictor block. The result of the encoding is a signalling of the predictor block and a residual block consisting in the difference between the block to be encoded and the predictor block.

Regarding INTRA coding modes, various modes are usually proposed, such as a Direct Current (DC) mode, a planar mode and angular modes. Each of them seeks to predict samples of a block using previously decoded boundary samples from spatially neighbouring blocks.

The encoding may be performed for each component forming the pixels of the video data. Although RGB (for Red-Green-Blue) representation is well-known, the YUV representation is preferably used for the encoding to reduce the inter-channel redundancy. According to these encoding modes, a block of pixels may be considered as composed of several, typically three, component blocks. An RGB pixel block is composed of an R component block containing the values

of the R component of the pixels of the block, a G component block containing the values of the G component of these pixels, a B component block containing the values of the B component of these pixels. Similarly, a YUV pixel block is composed of a Y component block (luma), a U component block (chroma) and a V component block (also chroma). Another example is YCbCr, where Cb and Cr are also known as chroma components. However, inter-component (also known as cross-component) correlation is still observed locally.

To improve compression efficiency, the usage of Cross-Component Prediction (CCP) has been studied in the state of this art. The main application of CCP concerns luma-to-chroma prediction. It means that the luma samples have already been encoded and reconstructed from encoded data (as the decoder does) and that chroma is predicted from luma. However, variants use CCP for chroma-to-chroma prediction or more generally for first-component to second-component prediction (including RGB).

The Cross-Component Prediction may apply directly to a block of chroma pixels or may apply to a residual chroma block (meaning the difference between a chroma block and a chroma block predictor).

The Linear Model (LM) mode uses a linear model to predict chroma from luma as a chroma intra prediction mode, relying on one or two parameters, slope (α) and offset (β), to be determined. The chroma intra predictor is thus derived from reconstructed luma samples of a current luma block using the linear model with the parameters.

The linearity, i.e. parameters α and β, is derived from the reconstructed causal samples, in particular from a neighbouring chroma sample set comprising reconstructed chroma samples neighbouring the current chroma block to predict and from a neighbouring luma sample set comprising luma samples neighbouring the current luma block.

Specifically, for an NxN chroma block, the N neighbours of the above row and the N neighbours of the left column are used to form the neighbouring chroma sample set for derivation.

The neighbouring luma sample set is also made of N neighbouring samples just above the corresponding luma block and N neighbouring samples on the left side of the luma block.

It is known to reduce the size of the video data to encode without significant degradation of visual rendering, by sub-sampling the chroma components. Known subsampling modes are labelled 4:1:1, 4:2:2, 4:2:0.

In the situation where the video chroma data are subsampled, the luma block corresponding to the NxN chroma block is bigger than NxN. In that case, the neighbouring luma sample set is down-sampled to match the chroma resolution. The chroma intra predictor to predict the chroma samples in the current NxN chroma block has to be generated using the linear model with the one or more parameters α and β derived and the reconstructed luma samples of the current luma block that are previously down-sampled to match chroma resolution. The down-sampling of the reconstructed luma samples to chroma resolution makes it possible to retrieve the same number of samples as the chroma samples to form both the luma sample set and the chroma intra predictor.

The chroma intra predictor is thus subtracted from the current chroma block to obtain a residual chroma block that is encoded at the encoder. Conversely, at the decoder, the chroma intra predictor is added to the received residual chroma block in order to retrieve the chroma block, also known as reconstruction of the decoded block. This may also involve clipping for results of the addition going out of the sample range.

Sometimes, the residual chroma block is negligible and thus not considered during encoding. In that case, the above-mentioned chroma intra predictor is used as the chroma block itself. As a consequence, the above LM mode makes it possible to obtain a sample for a current block of a given component from an associated (i.e. collocated or corresponding) reconstructed sample of a block of another component in the same frame using a linear model with one or more parameters. The sample is obtained using the linear model with the one or more parameters derived and the associated reconstructed samples in the block of the other component. If needed, the block of the other component

is made of samples down-sampled to match the block resolution of the current component. While the block of the current component is typically a chroma block and the block of the other component a luma block, this may not be the case. For the sake of clarity and simplicity, the examples given here focus on the prediction of a chroma block from a luma block, it should be clear that the described mechanism may apply to any component prediction from another component.

The Joint Exploration Model (JEM) of the Joint Video Exploration Team (JVET) adds six Cross-Component (luma-to-chroma) linear model modes to the conventional intra prediction modes already known. All these modes compete against each other to predict or generate the chroma blocks, the selection being usually made based on a rate-distortion criterion at the encoder end.

The six Cross-Component (luma-to-chroma) linear model modes differ from each other by different down-sampling schemes used to down-sample the reconstructed luma samples and/or by different sample sets of samples from which the parameters α and β are derived.

For instance, the sample sets may be made of the two lines (i.e. rows and columns) of samples neighbouring the current luma or chroma block, these lines being parallel and immediately adjacent to each one of the top and/or left boundaries of the current luma or chroma block at chroma resolution. Such exemplary sample set is described in publication US 9,736,487.

Other exemplary sample sets are also disclosed in publications US 9,288,500 and US 9,462,273.

The down-sampling schemes used in JEM include a 6-tap filter determining a down-sampled reconstructed luma sample from six reconstructed luma samples but also three 2-tap filters that select either the top right and bottom right samples from among the six reconstructed luma samples, or the bottom and bottom right samples, or the top and top right samples, and a 4-tap filter that selects the top, top right, bottom and bottom right samples of the six reconstructed luma samples.

SUMMARY OF THE INVENTION

The JEM is complex in terms of processing. For instance, it requires a complex derivation of the linear model parameters for the computation of the chroma predictor block samples.

The present invention has been devised to address one or more of the foregoing concerns. It concerns an improved method for obtaining a chroma sample for a current chroma block, possibly through chroma intra prediction.

According to a first aspect of the present invention, there is provided a method of deriving a linear model as set out in any of accompanying claims 1 to 10. According to a second aspect, there is provided a method of obtaining a chromas sample as set out in any of accompanying claims 11 to 13. According to a third aspect there is provided a method of encoding one or more images into a bitstream as set out in accompanying claim 14 and 15. According to a fourth aspect, there is provided a method of decoding one or more images from a bitstream as set out in claims 16 and 17.

According to a fifth aspect of the present invention, there is provided a device for deriving a linear model as set out in any of accompanying claims 18 to 27.

According to a sixth aspect of the present invention, there is provided a device for obtaining a chroma sample for a chroma block as set out in any of accompanying claims 28 to 30.

According to seventh aspect of the present invention, there is provided a device for encoding images as set out in accompanying claim 31.

According to an eight aspect of the present invention, there is provided a device for decoding images as set out in accompanying claim 32.

According to a ninth aspect of the present invention, there is provided a computer program product as set out in accompanying claim 33. According to a tenth aspect of the present invention, there is provided a computer-readable medium as set out in accompanying claim 34. According to an eleventh aspect there is provided a computer program as set out in accompanying claim 35.

Other features of embodiments of the invention are defined in the description which follows. Some of these features are explained here below with reference to a method, while they can be transposed into system features dedicated to the device.

At least parts of the methods according to the invention may be computer implemented. Accordingly, the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a “processor and a memory”, "circuit", "module" or "system". Furthermore, the present invention may take the form of a computer program product embodied in any tangible medium of expression having computer usable program code embodied in the medium.

Since the present invention can be implemented in software, the present invention can be embodied as computer readable code for provision to a programmable apparatus on any suitable carrier medium. A tangible carrier medium may comprise a storage medium such as a hard disk drive, a magnetic tape device or a solid state memory device and the like. A transient carrier medium may include a signal such as an electrical signal, an electronic signal, an optical signal, an acoustic signal, a magnetic signal or an electromagnetic signal, e.g. a microwave or RF signal.

➔ Continues at page 13

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention will now be described, by way of example only, and with reference to the following drawings in which:

Figure 1 illustrates a video encoder logical architecture;

Figure 2 illustrates a video decoder logical architecture corresponding to the video encoder logical architecture illustrated in **Figure 1**;

Figure 3 schematically illustrates examples of a YUV sampling scheme for 4:2:0 sampling;

Figure 4 illustrates, using a flowchart, general steps for generating a block predictor using the LM mode, performed either by an encoder or a decoder;

Figure 5 schematically illustrates a chroma block and an associated or collocated luma block, with down-sampling of the luma samples, and neighbouring chroma and luma samples, as known in prior art;

Figure 6 schematically illustrates exemplary sample sets for LM parameter derivation as known in prior art;

Figure 7 illustrates some down-sampling filters known in prior art;

Figure 8 illustrates exemplary coding of signalling flags to signal LM modes;

Figure 9 illustrates points of luma and chroma neighboring samples and a straight line representing the linear model parameters obtained in one embodiment of the invention;

Figure 10 illustrates the main steps of a process of the simplified LM derivation in one embodiment of the invention;

Figure 11 illustrates several points of luma and chroma neighboring samples and segments used to determine best two points in some embodiments of the invention;

Figure 12 illustrates the main steps of a process of a MMLM derivation in one embodiment of the invention; and

Figure 13 is a schematic block diagram of a computing device for implementation of one or more embodiments of the invention.

DETAILLED DESCRIPTION OF EMBODIMENTS

Figure 1 illustrates a video encoder architecture. In the video encoder, an original sequence **101** is divided into blocks of pixels **102** called coding blocks or coding units for HEVC. A coding mode is then affected to each block. There are two families of coding modes typically used video coding: the coding modes based on spatial prediction or "INTRA modes" **103** and the coding modes based on temporal prediction or "INTER modes" based on motion estimation **104** and motion compensation **105**.

An INTRA coding block is generally predicted from the encoded pixels at its causal boundary by a process called INTRA prediction. The predictor for each pixel of the INTRA coding block thus forms a predictor block. Depending on which pixels are used to predict the INTRA coding block, various INTRA modes are proposed: for example, DC mode, a planar mode and angular modes.

While **Figure 1** is directed to a general description of a video encoder architecture, it is to be noted that a pixel corresponds here to an element of an image, that typically consists of several components, for example a red component, a green component, and a blue component. An image sample is an element of an image, that comprises only one component.

Temporal prediction first consists in finding in a previous or future frame, called the reference frame **116**, a reference area which is the closest to the coding block in a motion estimation step **104**. This reference area constitutes the

predictor block. Next this coding block is predicted using the predictor block to compute the residue or residual block in a motion compensation step **105**.

In both cases, spatial and temporal prediction, a residue or residual block is computed by subtracting the obtained predictor block from the coding block.

In the INTRA prediction, a prediction mode is encoded.

In the temporal prediction, an index indicating the reference frame used and a motion vector indicating the reference area in the reference frame are encoded. However, in order to further reduce the bitrate cost related to motion vector encoding, a motion vector is not directly encoded. Indeed, assuming that motion is homogeneous, it is particularly advantageous to encode a motion vector as a difference between this motion vector, and a motion vector (or motion vector predictor) in its surroundings. In the H.264/AVC coding standard for instance, motion vectors are encoded with respect to a median vector computed from the motion vectors associated with three blocks located above and on the left of the current block. Only a difference, also called residual motion vector, computed between the median vector and the current block motion vector is encoded in the bitstream. This is processed in module "Mv prediction and coding" **117**. The value of each encoded vector is stored in the motion vector field **118**. The neighbouring motion vectors, used for the prediction, are extracted from the motion vector field **118**.

The HEVC standard uses three different INTER modes: the Inter mode, the Merge mode and the Merge Skip mode, which mainly differ from each other by the signalling of the motion information (i.e. the motion vector and the associated reference frame through its so-called reference frame index) in the bit-stream **110**. For the sake of simplicity, motion vector and motion information are conflated below. Regarding motion vector prediction, HEVC provides several candidates of motion vector predictor that are evaluated during a rate-distortion competition in order to find the best motion vector predictor or the best motion information for respectively the Inter or the Merge mode. An index corresponding to the best predictors or the best candidate of the motion information is inserted in the bitstream **110**. Thanks to this signalling, the decoder can derive the same

set of predictors or candidates and uses the best one according to the decoded index.

The design of the derivation of motion vector predictors and candidates contributes to achieving the best coding efficiency without large impact on complexity. Two motion vector derivations are proposed in HEVC: one for Inter mode (known as Advanced Motion Vector Prediction (AMVP)) and one for the Merge modes (known as Merge derivation process).

Next, the coding mode optimizing a rate-distortion criterion for the coding block currently considered is selected in module **106**. In order to further reduce the redundancies within the obtained residue data, a transform, typically a DCT, is applied to the residual block in module **107**, and a quantization is applied to the obtained coefficients in module **108**. The quantized block of coefficients is then entropy coded in module **109** and the result is inserted into the bit-stream **110**.

The encoder then performs decoding of each of the encoded blocks of the frame for the future motion estimation in modules **111** to **116**. These steps allow the encoder and the decoder to have the same reference frames **116**. To reconstruct the coded frame, each of the quantized and transformed residual blocks is inverse quantized in module **111** and inverse transformed in module **112** in order to provide the corresponding "reconstructed" residual block in the pixel domain. Due to the loss of the quantization, this "reconstructed" residual block differs from original residual block obtained at step **106**.

Next, according to the coding mode selected at **106** (INTER or INTRA), this "reconstructed" residual block is added to the INTER predictor block **114** or to the INTRA predictor block **113**, to obtain a "pre-reconstructed" block (coding block).

Next, the "pre-reconstructed" blocks are filtered in module **115** by one or several kinds of post filtering to obtain "reconstructed" blocks (coding blocks). The same post filters are integrated at the encoder (in the decoding loop) and at the decoder to be used in the same way in order to obtain exactly the same reference frames at encoder and decoder ends. The aim of this post filtering is to remove compression artefacts.

Figure 2 illustrates a video decoder architecture corresponding to the video encoder architecture illustrated in **Figure 1**.

The video stream **201** is first entropy decoded in a module **202**. Each obtained residual block (coding block) is then inverse quantized in a module **203** and inverse transformed in a module **204** to obtain a "reconstructed" residual block. This is similar to the beginning of the decoding loop at the encoder end.

Next, according to the decoding mode indicated in the bitstream **201** (either INTRA type decoding or an INTER type decoding), a predictor block is built.

In case of INTRA mode, an INTRA predictor block is determined **205** based on the INTRA prediction mode specified in the bit-stream **201**.

In case of INTER mode, the motion information is extracted from the bitstream during the entropy decoding **202**. The motion information is composed, for example in HEVC and JVET, of a reference frame index and a motion vector residual.

A motion vector predictor is obtained in the same way as done by the encoder (from neighbouring blocks) using already computed motion vectors stored in motion vector field data **211**. It is thus added **210** to the extracted motion vector residual block to obtain the motion vector. This motion vector is added to the motion vector field data **211** in order to be used for the prediction of the next decoded motion vectors.

The motion vector is also used to locate the reference area in the reference frame **206** which is the INTER predictor block.

Next, the "reconstructed" residual block obtained at **204** is added to the INTER predictor block **206** or to the INTRA predictor block **205**, to obtain a "pre-reconstructed" block (coding block) in the same way as the decoding loop of the encoder.

Next, this "pre-reconstructed" block is post filtered in module **207** as done at the encoder end (signalling of the post filtering to use may be retrieved from bitstream **201**).

A "reconstructed" block (coding block) is thus obtained which forms the de-compressed video **209** as the output of the decoder.

The above-described encoding/decoding process may be applied to monochrome frames. However, most common frames are colour frames generally made of three arrays of colour samples, each array corresponding to a "colour component", for instance R (red), G (green) and B (blue). A pixel of the image comprises three collocated/corresponding samples, one for each component.

R, G, B components have usually high correlation between them. It is thus very common in image and video compression to decorrelate the colour components prior to processing the frames, by converting them in another colour space. The most common format is the YUV (YCbCr) where Y is the luma (or luminance) component, and U (Cb) and V (Cr) are chroma (or chrominance) components.

To reduce the amount of data to process, some colour components of the colour frames may be subsampled, resulting in having different sampling ratios for the three colour components. A subsampling scheme is commonly expressed as a three part ratio J:a:b that describes the number of luma and chroma samples in a conceptual 2-pixel-high region. 'J' defines the horizontal sampling reference of the conceptual region (i.e. a width in pixels), usually 4. 'a' defines the number of chroma samples (Cr, Cb) in the first row of J pixels, while 'b' defines the number of (additional) chroma samples (Cr, Cb) in the second row of J pixels.

With the subsampling schemes, the number of chroma samples is reduced compared to the number of luma samples.

The 4:4:4 YUV or RGB format does not provide subsampling and corresponds to a non-subsampled frame where the luma and chroma frames have the same size W x H.

The 4:0:0 YUV or RGB format has only one colour component and thus corresponds to a monochrome frame.

Exemplary sampling formats are the following.

The 4:2:0 YUV format has half as many chroma samples as luma samples in the first row, and no chroma samples in the second row. The two chroma frames are thus W/2-pixel wide and H/2-pixel height, where the luma frame is W x H.

The 4:2:2 YUV format has half as many chroma samples in the first row and half as many chroma samples in the second raw, as luma samples. The two chroma frames are thus W/2-pixel wide and H-pixel height, where the luma frame is W x H.

The 4:1:1 YUV format has 75% fewer chroma samples in the first row and 75% fewer chroma samples in the second row, than the luma samples. The two chroma frames are thus W/4-pixel wide and H-pixel height, where the luma frame is W x H.

When subsampled, the positions of the chroma samples in the frames are shifted compared to the luma sample positions.

Figure 3 illustrates an exemplary positioning of chroma samples (triangles) with respect to luma samples (circles) for a 4:2:0 YUV frame.

The encoding process of **Figure 1** may be applied to each colour-component frame of an input frame.

Due to correlations between the colour components (between RGB or remaining correlations between YUV despite the RGB-to-YUV conversion), Cross-Component Prediction (CCP) methods have been developed to exploit these (remaining) correlations in order to improve coding efficiency.

CCP methods can be applied at different stages of the encoding or the decoding process, in particular either at a first prediction stage (to predict a current colour component) or at a second prediction stage (to predict a current residual block of a component).

One known CCP method is the LM mode, also referred as to CCLM (Cross-Component Linear Model prediction). It is used to predict both chroma components Cb and Cr (or U and V) from the luma Y, more specifically from the reconstructed luma (at the encoder end or at the decoder end). One predictor is

generated for each component. The method operates at a (chroma and luma) block level, for instance at CTU (coding tree unit), CU (coding unit) level, PU (prediction unit) level, sub-PU or TU (transform unit) level.

Figure 4 illustrates as an example, using a flowchart, general steps for generating a block predictor using the LM mode, performed either by the encoder (used as reference below) or the decoder.

In the description below, an exemplary first component is chroma while an exemplary second component is luma.

Considering a current chroma block **502** (**Figure 5A**) to encode or decode and its associated or corresponding (i.e. "collocated") luma block **505** (i.e. of the same CU for instance) in the same frame, the encoder (or the decoder) receives, in step **401**, a neighbouring luma sample set RecL comprising luma samples **503** neighbouring the current luma block, and receives a neighbouring chroma sample set RecC comprising chroma samples **501** neighbouring the current chroma block, denoted **402**. It is to be noted that for some chroma sampling formats and chroma phase, the luma samples **504** and **503** are not directly adjacent to luma block **505** as depicted in **Figure 5A**. For example in **Figure 5A**, to obtain the left row RecL' (**503**), only the second left row is needed and not the direct left row. In the same way, for the up line **504** the second up line is also considered for the down-sampling of luma sample as depicted in **Figure 5A**.

When a chroma sampling format is used (e.g. 4:2:0, 4:2:2, etc.), the neighbouring luma sample set is down-sampled at step **403** into RecL' **404** to match chroma resolution (i.e. the sample resolution of the corresponding chroma frame/block). RecL' thus comprises reconstructed luma samples **504** neighbouring the current luma block that are down-sampled. Thanks to the down-sampling, RecL' and RecC comprise the same number 2N of samples (chroma block 502 being N x N). Yet, particular down-samplings of the luma border exist in the prior art where less samples are needed to obtain RecL'. In addition, even if RecL and RecC have the same resolution, RecL' can be seen as the denoised version of RecL, through the use of a low-pass convolution filter.

In the example of **Figure 5A**, the neighbouring luma and chroma sample sets are made of the down-sampled top and left neighbouring luma samples and of the top and left neighbouring chroma samples, respectively. More precisely each of the two sample sets is made of the first line immediately adjacent to the left boundary and the first line immediately adjacent to the top boundary of their respective luma or chroma block. Due to down-sampling (4:2:0 in **Figure 5A**), the single line of neighbouring luma samples RecL' is obtained from two lines of non down-sampled reconstructed luma samples RecL (left or up).

US 9,565,428 suggests using sub-sampling which selects a single sample, only for the up line (i.e. adjacent to the top boundary of the luma block) and not for the luma block itself (as described below with reference to step **408**). The proposed sub-sampling is illustrated in **Figure 6A**. The motivation for this approach is to reduce the line buffer of the up line.

The linear model which is defined by one or two parameters (a slope α and an offset β) is derived from RecL' (if any, otherwise RecL) and RecC. This is step **405** to obtain the parameters **406**.

The LM parameters α and β are obtained using a least mean square-based method using the following equations:

$$\alpha = \frac{M.\sum_{i=1}^{M} RecC_i.RecL'_i - \sum_{i=1}^{M} RecC_i.\sum_{i=1}^{M} RecL'_i}{M.\sum_{i=1}^{M} RecL'^2_i - (\sum_{i=1}^{M} RecL'_i)^2} = \frac{A_1}{A_2}$$

$$\beta = \frac{\sum_{i=1}^{M} RecC_i - \alpha.\sum_{i=1}^{M} RecL'_i}{M}$$

where M is a value which depends on the size of the block considered. In general cases of square blocks as shown in the **Figures 5A** and **5B**, M=2N. However, the LM-based CCP may apply to any block shape where M is for instance the sum of the block height H plus the block width W (for a rectangular block shape).

It is to be noted that the value of M used as a weight in this equation may be adjusted to avoid computational overflows at the encoder and decoder. To be precise, when using arithmetic with 32-bit or 64-bit signed architectures, some of

the computations may sometimes overflow and thus cause unspecified behaviour (which is strictly prohibited in any cross platform standard). To face this situation, the maximum magnitude possible given inputs RecL' and RecC values may be evaluated, and M (and in turn the sums above) may be scaled accordingly to ensure that no overflow occurs.

The derivation of the parameters is usually made from the sample sets RecL' and RecC shown in **Figure 5A**.

Variations of the sample sets have been proposed.

For instance, US 9,288,500 proposes three competing sample sets, including a first sample set made of the outer line adjacent to the top boundary and the outer line adjacent to the left boundary, a second sample set made of only the outer line adjacent to the top boundary and a third sample set made of only the outer line adjacent to the left boundary. These three sample sets are shown in **Figure 6B** for the chroma block only (and thus can be transposed to the luma block).

US 9,462,273 extends the second and third sample sets to additional samples extending the outer lines (usually doubling their length). The extended sample sets are shown in **Figure 6C** for the chroma block only. This document also provides a reduction in the number of LM modes available in order to decrease the signalling costs for signalling the LM mode used in the bitstream. The reduction may be contextual, for instance based on the Intra mode selected for the associated luma block.

US 9,736,487 proposes three competing sample sets similar to those of US 9,288,500 but made, each time, of the two lines of outer neighbouring samples parallel and immediately adjacent to the boundaries considered. These sample sets are shown in **Figure 6D** for the chroma block only.

Also US 9,153,040 and the documents of the same patent family propose additional sample sets made of a single line per boundary, with less samples per line than the previous sets.

Back to the process of **Figure 4**, using the linear model with the one or more derived parameters **406**, a chroma intra predictor **413** for chroma block **502** may thus be obtained from the reconstructed luma samples **407** of the current luma block represented in **505**. Again if a chroma sampling format is used (e.g. 4:2:0, 4:2:2, etc.), the reconstructed luma samples are down-sampled at step **408** into L' **409** to match chroma resolution (i.e. the sample resolution of the corresponding chroma frame/block).

The same down-sampling as for step **403** may be used, or another one for a line buffer reason. For instance, a 6-tap filter may be used to provide the down-sampled value as a weighted sum of the top left, top, top right, bottom left, bottom and bottom right samples surrounding the down-sampling position. When some surrounding samples are missing, a mere 2-tap filter is used instead of the 6-tap filter.

Applied to reconstructed luma samples L, output L' of an exemplary 6-tap filter is obtained as follows:

$$L'[i,j] = (2 \times L[2i,2j] + 2 \times L[2i,2j+1] + L[2i-1,2j] + L[2i+1,2j] + L[2i-1,2j+1] + L[2i+1,2j+1] + 4) >> 3$$

with (i,j) being the coordinates of the sample within the down-sampled block and >> being the bit-right-shifting operation.

Also an adaptive luma down-sampling may be used as described in US 2017/0244975. Only the content of the luma block is used to determine which down-sampling filter is used for each reconstructed luma sample of the luma block. A 1-tap filter is available. The motivation of this approach is to avoid propagation of an edge in the down-sampled luma block.

Thanks to down-sampling step **408**, L' and C blocks (the set of chroma samples in chroma block 502) comprise the same number N^2 of samples (chroma block **502** being N x N).

Next, each sample of the chroma intra predictor PredC **413** is calculated using the loop **410-411-412** following the formula

$$PredC[i,j] = \alpha . L'[i,j] + \beta$$

with (i,j) being the coordinates of all samples within the chroma and luma blocks.

To avoid divisions and multiplications, the computations may be implemented using less complex methods based on look-up tables and shift operations. For instance, the actual chroma intra predictor derivation **411** may be done as follows:

$$PredC[i,j] = (A.L'[i,j]) \gg S + \beta$$

where S is an integer and A is derived from A1 and A2 (introduced above when computing α and β) using the look-up table mentioned previously. It actually corresponds to a rescaled value of α. The operation (x >> S) corresponds to the bit-right-shifting operation, equivalent to an integer division of x (with truncation) by 2^S.

When all samples of the down-sampled luma block have been parsed (**412**), the chroma intra predictor **413** is available for subtraction from chroma block **502** (to obtain a chroma residual block) at the encoder end or for addition to a chroma residual block (to obtain a reconstructed chroma block) at the decoder end.

Note that the chroma residual block may be insignificant and thus discarded, in which case the obtained chroma intra predictor **413** directly corresponds to predicted chroma samples (forming chroma block **502**).

Both standardization groups ITU-T VCEG (Q6/16) and ISO/IEC MPEG (JTC 1/SC 29/WG 11) which have defined the HEVC standard are studying future video coding technologies for the successor of HEVC in a joint collaboration effort known as the Joint Video Exploration Team (JVET). The Joint Exploration Model (JEM) contains HEVC tools and new added tools selected by this JVET group. In particular, this reference software contains some CCP tools, as described in document JVET-G1001.

In the JEM, a total of 11 intra modes are allowed for chroma coding. Those modes include five traditional intra modes and six cross-component LM modes to predict Cb from Y (signalled in bitstream **110**, **201**) and one cross-component LM mode to predict Cr from Cb.

One of the six Y-to-Cb CC LM modes is the CCLM described above, in which the neighbouring luma and chroma sample sets RecL' and RecC are each made of the first line immediately adjacent to the left boundary and the first line immediately adjacent to the top boundary of their respective luma or chroma block as shown in **Figure 5A**.

The five other Y-to-Cb CC LM modes are based on a particular derivation known as the Multiple Model (MM). These modes are labelled MMLM.

Compared to CCLM, the MMLM modes use two linear models. The neighbouring reconstructed luma samples from the RecL' set and the neighbouring chroma samples from the RecC set are classified into two groups, each group being used to derive the parameters α and β of one linear model, thus resulting in two sets of linear model parameters (α_1,β_1) and (α_2,β_2).

For instance, a threshold may be calculated as the average value of the neighbouring reconstructed luma samples forming RecL'. Next, a neighbouring luma sample with RecL'[i,j] ≤ threshold is classified into group 1; while a neighbouring luma sample with RecL'[i,j] > threshold is classified into group 2.

Next, the chroma intra predictor (or the predicted chroma samples for current chroma block **602**) is obtained according to the following formulas:

$$PredC[i,j] = \alpha_1 . L'[i,j] + \beta_1, \qquad if\ L'[i,j] \leq threshold$$

$$PredC[i,j] = \alpha_2 . L'[i,j] + \beta_2, \qquad if\ L'[i,j] > threshold$$

In addition, compared to CCLM, the MMLM modes use neighbouring luma and chroma sample sets RecL' and RecC, each made of the two lines of outer neighbouring samples parallel and immediately adjacent to the left and top boundaries of the block considered. An example is shown in **Figure 5B** illustrating a 4:2:0 sampling format for which the two lines of neighbouring luma samples are obtained (using down-sampling) from four lines of non down-sampled reconstructed luma samples.

The five MMLM modes differ from each other by five different down-sampling filters to down-sample the reconstructed luma samples to match chroma resolution (to obtain RecL' and/or L').

A first MMLM mode relies on the same 6-tap filter as used in CCLM (see the 6 black points in reference **701** of **Figure 7**). The second to fourth MMLM modes rely on 2-tap filters which respectively provide the down-sampled value as a weighted sum of:

- the top right and bottom right samples of the six samples (used by the 6-tap filter) surrounding the down-sampling position (see filter 1, **702** of **Figure 7**):

  $L'[i,j] = (L[2i+1,2j] + L[2i+1,2j+1] + 1) \gg 1$ (the same applies with RecL),

- the bottom and bottom right samples of the six samples (used by the 6-tap filter) surrounding the down-sampling position (see filter 2, **703** of **Figure 7**):

  $L'[i,j] = (L[2i,2j+1] + L[2i+1,2j+1] + 1) \gg 1$, and

- the top and top right samples of the six samples (used by the 6-tap filter) surrounding the down-sampling position (see filter 4, **705** of **Figure 7**):

  $L'[i,j] = (L[2i,2j] + L[2i+1,2j] + 1) \gg 1.$

The fifth MMLM mode relies on 4-tap filter which provides the down-sampled value as the weighted sum of the top, top right, bottom and bottom right samples of the six samples (used by the 6-tap filter) surrounding the down-sampling position (see filter 3, **704** of **Figure 7**):

$$L'[i,j] = (L[2i,2j] + L[2i,2j+1] + L[2i+1,2j] + L[2i+1,2j+1] + 2) \gg 2.$$

As indicated above, the CCLM or MMLM mode has to be signalled in the bitstream **110** or **201**. **Figure 8** illustrates an exemplary LM mode signalling of JEM. A first binary flag indicates whether the current block is predicted using an LM mode or other intra modes, including so-called DM modes. In case of LM mode, the six possible LM modes need to be signalled. The first MMLM mode (using the 6-tap filter) is signalled with one second binary flag set to 1. This second binary flag is set to 0 for the remaining modes, in which case a third binary flag is set to 1 to signal the CCLM mode and is set to 0 for the remaining MMLM

modes. Two additional binary flags are then used to signal one of the four remaining MMLM modes.

One mode is signalled for each chroma component.

The Cb-to-Cr CCLM mode introduced above is used in DM modes, and applies at residual level. Indeed, a DM mode uses for chroma the intra mode which was used by luma in a predetermined location. Traditionally, a coding mode like HEVC uses one single DM mode, co-located with the top-left corner of the CU. Without going in too many details, and for the sake of clarity, JVET provides several such locations. This mode is then used to determine the prediction method, therefore creating a usual intra prediction for a chroma component which, when subtracted from the reference/original data, yield aforementioned residual data. The prediction for the Cr residual is obtained from the Cb residual (ResidualCb below) by the following formula:

$$PredCr[i,j] = \alpha . ResidualCb[i,j]$$

where α is derived in a similar way as in the CCLM luma-to-chroma prediction. The only difference is the addition of a regression cost relative to a default α value in the error function so that the derived scaling factor is biased towards a default value of -0.5 as follows:

$$\alpha = \frac{M . \sum_{i=1}^{M} RecCb_i . RecCr_i - \sum_{i=1}^{M} RecCb_i . \sum_{i=1}^{M} RecL'_i + \lambda\,(-0.5)}{M . \sum_{i=1}^{M} RecCb_i^2 - (\sum_{i=1}^{M} RecCb_i)^2 + \lambda}$$

where $RecCb_i$ represents the values of neighbouring reconstructed Cb samples, $RecCr_i$ represents the neighbouring reconstructed Cr samples, and

$$\lambda = \sum_{i=1}^{M} RecCb_i^2 \gg 9.$$

The known LM modes exhibit a large computation complexity, in particular when deriving the linear model parameter using least square based methods.

The present invention seeks to improve the situation in term of coding efficiency and/or computational complexity.

The invention is based on the replacement of the derivation of a linear model used to compute chroma predictor block samples from luma block samples by determination of the parameters of the linear model based on the equation of

a straight line. The straight line is defined by two sample pairs defined based on reconstructed sample pairs in the neighbourhood of the block. First the two sample pairs to be used are determined. Then the parameters of the linear model are determined from these two sample pairs. By limiting to two the number of sample pairs used in the determination of the linear model, the use of a least mean square method can be avoided. The proposed method is therefore less computation intensive than the known method using a least mean square method.

Figure 9 illustrates the principle of this method by considering here the minimum and the maximum of luma sample values in the set of sample pairs in the neighborhood of the current block. All the sample pairs are drawn on the figure according to their chroma value and their luma value. Two different points, namely point A and point B are identified on the figure, each point corresponding to a sample pair. Point A corresponds to the sample pair with the lowest luma value x_A from RecL' and y_A its collocated chroma value from RecC. Point B corresponds to the sample pair with the highest luma value x_B and y_B its collocated chroma value.

Figure 10 gives the flow chart of a proposed method to derive the linear model parameters. This flow chart is a simplified version of **Figure 4**. The method is based on the neighboring luma samples RecL' obtained in step **1001** and chroma samples RecC obtained in step **1002**.

In a step **1003**, the two points A and B (**1004**) corresponding to two sample pairs are determined. In a first embodiment, these two points A and B correspond to the sample pairs with respectively the lowest and highest luma sample values x_A and x_B with their corresponding chroma sample values y_A and y_B.

Then the straight line equation which crosses the points A and B is computed in a step **1005** according to the following equation:

$$\alpha = \frac{y_B - y_A}{x_B - x_A}$$

$$\beta = y_A - \alpha x_A$$

The obtained α, β are the linear model parameters **1006** used to generate the chroma predictor.

The linear model derivation based on the LMS algorithm used in the prior art has a certain complexity. In this known method, the computation of the α parameter of the model is obtained by the following equation:

$$\alpha = \frac{M\sum_{i=1}^{M} \mathrm{Re}cC_i \,\mathrm{Re}cL'_i - \sum_{i=1}^{M} \mathrm{Re}cC_i \sum_{i=1}^{M} \mathrm{Re}cL'_i}{M\sum_{i=1}^{M} \mathrm{Re}cL'^{2}_i - \left(\sum_{i=1}^{M} \mathrm{Re}cL'_i\right)^2} = \frac{B_1 - B_2}{B_3 - B_4} = \frac{A_1}{A_2}$$

The analysis of this equation regarding the computation complexity gives the following results. The computation of B_1 requires M+1 multiplications and M sums, M being the number of sample pairs. The computation of B_2 requires 1 multiplication and 2M sums. The computation of B_3 requires M+1 multiplication and M sums and the computation of B_4 requires one multiplication and 2M sums. The computation of α corresponding to $\frac{B_1 - B_2}{B_3 - B_4}$ requires two additional sums and one division.

To compute β , one multiplication and 2M+1 sums and one division. As described previously M is the number of pairs of samples $RecC_i$ and $RecL'_i$.

The complexity of the LMS derivation of α and β is therefore $(2M + 2 + 2)$ multiplications, $(7M + 3)$ additions and two divisions.

In comparison, the analysis of the proposed method based on the computation of the equation of a straight line using only two points gives the following results. As reported, the derivation step **1005** requires only one multiplication, three sums and one division. This large complexity reduction in generating the linear model parameters is a major advantage of the proposed invention.

It should be noted that the search for the minimum and maximum values has a complexity of its own, typically related to sorting algorithm. The operation

is not completely serial: N points can be compared to N other points, generating N minimum/maximum. Then N/2 minimum and N/2 maximum points can be compared to the N/2 others, then again N/4 and so on until only the desired numbers of minimum and maximum points remain. Typically, the search for the minimum and maximum thus results in approximatively 2*N-2 comparisons (N-1 for each).

As already described, the chroma predictor can be calculated with an integer multiplication and a shift instead of a floating-point multiplication, and a division when computing the slope. This simplification consists in replacing:

$$pred_C(i,\ j) = \alpha \cdot rec_L'^{(i,\ j)} + \beta$$

By:

$$pred_C(i,\ j) = \left(L \cdot rec_L'^{(i,\ j)}\right) \gg S + \beta$$

To use only integer multiplication and shift, in one embodiment, the straight line equation is obtained as follows:

$$S = 10$$

$$L = \frac{(y_B - y_A) \ll S}{x_B - x_A}$$

$$\beta = y_A - L(x_A \gg S)$$

Please note that β refers to this equation in the following if α is replaced by L and S otherwise it refers to the traditional equation $\beta = y_A - \alpha x_A$.

Another advantage of this derivation is that the shift value S always has the same value. This is interesting especially for hardware implementation that can be made simpler in taking advantage of this property.

In yet another embodiment, the value of S is forced to be low, as L could be large, and requires larger multiplier operations. Indeed, a multiply of 8bits values by a 8-bits value is much easier to implement than e.g. a 8*16 multiplier. Typical practical values for L are often equivalent to a multiplier less than 8 bits.

However, the preferred embodiment is an implementation known as fixed point: for every value of D=(x_B - x_A), possibly quantized (e.g. the results for 2D+0 and 2D+1 are stored as a single one), the value of (1 << S)/D is stored in a table.

Preferably, these are only for the positive values, as the sign can be easily retrieved. Using an array TAB, the computation of L thus becomes:

$$L = \begin{cases} (y_B - y_A) * TAB[abs(x_B - x_A)/Q] \; if \; x_B - x_A \geq 0 \\ -1 * (y_B - y_A) * TAB[abs(x_B - x_A)/Q] \; otherwise \end{cases}$$

Q controls the quantization and thus the number of elements in the table. Using Q=1 thus means no quantization. Also note that the looked-up index can be instead $(abs(x_B - x_A) + R)/Q$, typically with R=Q/2, or a variation thereof of the division rounding. Consequently, Q is ideally a power of 2 so that the division by $Q=2^P$ is equivalent to a right-shift by P.

Finally, some of the values in that table may not be equal to 0: low values of $abs(x_B - x_A)$ or $abs(y_B - y_A)$ often result in very bad estimations of L. Pre-determined, or explicit (such as in the slice header or a parameter set such as PPS or SPS) values can be used then. For instance, for all values of D below 4, the array TAB may contains a default value, e.g. -(1<<S)/8.

For 10 bits content and Q=1, up to 2048 entries in the array are needed. By exploiting the symmetry with sign as shown above, this can be reduced to 1024. Increasing further Q would similarly reduce the size of TAB.

If some of the samples (either RecL or RecC, or both) are residual samples (i.e. resulting themselves from the difference between two blocks, possibly quantized), as it is the case in JVET with Cb to Cr prediction, then the table size (and content) can be accordingly adapted.

Figure 11 illustrates different ways of selecting two points in embodiments of the invention.

The proposed simplification of the derivation has an impact on coding efficiency. To reduce this coding efficiency loss the careful selection of the two points is a very important step.

In the first embodiment as described previously, the minimum and the maximum of the neighboring luma sample values are selected corresponding to points A and B of **Figure 11**.

In an alternative embodiment, the two selected points are points C and D of **Figure 11**, which correspond to the pair of luma and chroma samples corresponding to the minimum and the maximum of the neighboring chroma sample values. This alternative embodiment is sometimes interesting in term of coding efficiency.

In an alternative embodiment, the longest segment between segments [AB] and [CD] is determined and if the [AB] segment is longer than the [CD] segment, points A and B are selected, otherwise points C and D are selected. The length of each segment can be computed with a Euclidian distance. Yet, another distance measure can be used. This embodiment improves the coding efficiency compared to the two first ones. Indeed, when the two selected points are far, generally the generated linear model is relevant. Consequently, the generated chroma block predictor is relevant to predict the current block.

In an alternative embodiment, the longest segment among all possible segments that can be generated between A, B, C, D gives the two selected points. This corresponds to segments [AB], [CD], [AC], [AD], [CB] and [DB] as depicted in **Figure 11**. This embodiment improves the coding efficiency compared to the previous ones at the price of a higher complexity.

In a preferred embodiment, the points representing the minimum and the maximum of the RecL' luma sample values are set so as to create the A and B points and if one component of point A is equal to its corresponding component from B ($x_B = x_A$ or $y_B = y_A$), the points representing the minimum and the maximum of the chroma sample values C and D are selected. This embodiment obtains the best coding efficiency because if $x_B = x_A$ or $y_B = y_A$ then α (or L) is respectively infinite or equal to 0 and consequently the chroma predictor block is respectively unusable or equivalent to the DC prediction. This is the case as soon as either the numerator or denominator of the fraction representative of $\alpha\ (\text{or}\ L)$ is too low (for example, it may be verified the following condition: $|\alpha| < 0.1$): any error on it (such as due to quantization) even by a small amount lead to very different values of $\alpha\ (\text{or}\ L)$. In the remainder of the document, such cases, which

are basically almost horizontal or vertical slopes, lead to what is referred to as an abnormal slope, whether it is α or L.

In an additional alternative embodiment, several pairs of points, as all depicted in **Figure 11**, are tested until α is not "abnormal". This embodiment improves the coding efficiency of the previous one but it increases the computational complexity.

In one alternative embodiment, the differences between the maximum and the minimum of the two components (chroma and luma) is computed. In addition, the component with the maximum difference is selected to determine the two points defining the line for the computation of the model parameters. This embodiment is efficient when the two components are the two chroma components or two RGB components.

The selection of the two points A and B may be done on values of samples of the current block. In one embodiment, the two points for the simplified linear model derivation are set based on the sample values of the current downsampled luma block (**505** in **Figure 5**). The luma samples values of the sample pairs in the neighborhood of the block are compared to the luma sample values of the luma block. The value with the maximum occurrence is selected to create x_A and the second value with the maximum occurrence is selected to create x_B. The corresponding chroma values y_A and y_B are the average values of the collocated chroma samples in the sample pairs in the neighborhood of the block. When α (or L) is "abnormal" (equal to 0 or close to 0 ($|\alpha| < 0.1$)), x_B is one of the luma values with the less selection instead of the second most selected value. In the same way, y_B are the average values of the collocated chroma samples. This embodiment increases the coding efficiency compared to previous embodiment at the price of a highest complexity.

The selection of the two points A and B may be done on spatial positions of sample pairs.

In previous embodiments it is needed to determine the minimum and the maximum values for luma (A, B) or/and for chroma (C, D) among M pairs of luma and chroma neighboring samples. This can be considered as an additional

complexity. Therefore, for some implementations, it is preferable to obtain these two points with a minimum complexity.

In one embodiment, one linear model is generated with the chroma samples RecC (**501** on **Figure 5**) and with the downsampled luma samples of the border RecL' (**503**). The first point selected is the bottom sample of the left row, referenced **5004,** of luma and the collocated chroma sample **5001**. The second point selected is the top right luma sample **5003** and the co-located chroma sample **5002**. This selection of the two points is very simple but it is also less efficient than previous embodiments based on values.

Additionally, if one of the top or left edge does not exist, for example for block on the border of an image or a slice or is unavailable, for example for complexity or error resilience reasons, then two samples (e.g. those whose luma is **504** or **5003** on the available edge) are selected instead of the missing one. It can thus be seen that several conditions exist to select the samples.

A further embodiment is described in which, if not enough points can be selected to compute the slope, or they result in an "abnormal" α (or L), then a default point can be selected instead. This embodiment can also be applied for MMLM mode with adaptation. To create the linear model parameters for the first group, the first point is the bottom sample of the second left row (**5009**) of luma and the collocated chroma sample (**5005**). Moreover, the second point is the up luma sample of the first left row (**5010**) and the collocated chroma sample (**5006**).

To create the linear model parameters for the second group, the first point is the left sample of the first up line (**5011**) of luma and the collocated chroma sample (**5007**). And the second point is the right luma sample of the second up line (**5012**) and the collocated chroma sample (**5008**).

This embodiment simplifies the selection of the four points for the first and the second group.

In yet another embodiment, the threshold of the MMLM mode is the luma value of points **5010** or **5011**, namely the top right point of the left neighborhood and the bottom left point of the top neighborhood in the example, or an average

between these points. This additional embodiment simplifies the computation of the threshold.

In yet a further embodiment to all these embodiments related to the selection of two points, the downsampling process of luma is disabled and it is replaced by a decimation, i.e. one in two luma samples is used for the RecL' samples. In this case, steps **1001** in **Figure 10** and **1201** in **Figure 12**, which is described in detail below, would be skipped. This embodiment reduces the complexity with a minor impact on coding efficiency.

Points A, B, C and D are determined based on decoded versions of samples, and thus may not match the original samples values. This can cause abnormally short segments, or just noisy estimation, as already described when defining what is an "abnormal" slope. A and C being the two lowest points, and B and D being the two highest, instead of using any two of them, the point E defined as the average between A and C, and point F defined as the average between B and D can be used, at the cost of a few simple supplementary operations:

$$x_E = (x_A + x_C + 1) >> 1 \text{ and } y_E = (y_A + y_C + 1) >> 1$$

$$x_F = (x_B + x_D + 1) >> 1 \text{ and } y_F = (y_B + y_D + 1) >> 1$$

$$A = \frac{(y_E - y_F) \ll S}{x_E - x_F}$$

$$\beta = y_E\text{-}A.(x_E \gg S)$$

Obviously, if y_E-y_F or x_E-x_F equals 0 or is too low (i.e. the derived slope is "abnormal"), then points A, B, C and D are considered per the usual to obtain better parameters.

It should be understood from this that the two points used in the computation of the slope in the model may not be two actual points made from samples values of RecL' or RecC. This explains the use of the "determine" wording in step **1003** instead of "select".

In yet a further embodiment, for MMLM mode, if the α (or L) parameter defining the slope of one group is "abnormal", the corresponding LM parameters are set equal to the LM parameters of the other group or another group if more than two groups of LM parameters. **Figure 12** illustrates this embodiment. After

the determination of $(\alpha 1, \beta 1)$ and $(\alpha 2, \beta 2)$ defining the two models for the two groups in step **1203**, $\alpha 1$ and $\alpha 2$ are tested to check if they are equal to 0 in steps **1204** and **1205**. If it is the case, the "abnormal" slope parameter α (or L) is set equal to the other slope parameter α, similarly, the corresponding β parameter value of the other group is also used in steps **1206** and **1207**. So in that case, only one set of parameters is used whatever the value of the downsampled luma sample values of the current block, there is no comparison to the threshold, the same complexity as the CCLM mode is obtained. The advantage of this embodiment is a coding efficiency improvement with a small complexity because no additional linear model parameters need to be derived.

In an alternative embodiment, when one slope parameter α (or L) is "abnormal", one set of linear parameters are re-derived by considering all initial samples of the MMLM (it corresponds to the CCLM derivation with two up lines and two neighboring rows, instead of one up line and one neighboring row). This embodiment gives better coding efficiency than the previous one, but it is more complex because it is needed to re-derive a set of linear model parameters.

The simplified LM derivation with two points as described in this document is generally less efficient than the classical LMS derivation except if it does not replace all LMS derivation when several LM modes are competing.

In one embodiment, the LM derivation with two points is used only for the CCLM mode to derive a chroma block predictor. This embodiment gives coding efficiency improvements.

In one embodiment, the derivation with two points is used only for the MMLM mode, as it is the most complex prediction method.

In one embodiment, the LM derivation with two points is used for the CCLM mode and the MMLM mode to derive a chroma block predictor. This embodiment has similar coding efficiency as the JEM but it reduces the worst-case complexity by using this simplified LM derivation for the generation of the chroma block predictors. Indeed the chroma prediction based on luma is the mode presenting the worst-case complexity among the prediction linear model modes. It is more complex than the residual chroma prediction.

In one embodiment, the LM derivation with two points replaces all LMS derivations (chroma block predictor generation and residual prediction). This embodiment reduces the coding efficiency compared to JEM but it significantly decreases the complexity. Please note that these two embodiments give a coding efficiency improvement whatever the derivation method used in step **1203** for parameters.

In yet another embodiment, if one or both of the slope parameters α (or L) are "abnormal", then a default value (such as $-(1 << S)/8$) is used instead in steps **1206** and/or **1207**, and the corresponding value β is computed.

In yet a further embodiment, several LM modes are competing at encoder side, and syntax elements may signal the selected LM mode in the bitstream at decoder side. This signaling may be at the slice-level (or PPS, or SPS) to indicate which sets should be used, or at least provide the candidates for a block-level selection. At least one of the differences between these competing LM modes is the set of two points used to derive the LM parameters. The set of two points and the method to generate these two points define different LM modes in competition. For example, for one LM mode the two points are determined based on the minimum and the maximum luma values and for one another LM mode the two points are selected based on the maximum and minimum chroma values.

Another embodiment consists in defining a number of sets from possible locations as illustrated on **Figure 5**. While four such different points can lead to up to twelve different pairs, the ones resulting in the largest values for the numerator and denominator in the equation for the calculation of the slope parameter α (or L) can be preferred. The encoder builds the list of pairs, and sort them according to some criterion (such as distance in the luma component, or Cartesian distance using both luma and chroma components), possibly removing some of them (i.e. if their slope is too close to another one), and thus building the list of parameters that can be selected and signaled.

The advantage of these embodiments is a coding efficiency improvement.

The descriptions of these embodiments mention the luma and a chroma component but can easily be adapted to other components such as both chroma

components, or RGB components. According to an embodiment, the present invention is used when predicting a first chroma component sample value from a second chroma component. In another embodiment, the present invention is used when predicting a sample value of one component from more than one sample values of more than one component. It is understood that in such a case, the linear model is derived based on two points/sets, each point/set comprising a sample value of the one component, and the more than one sample values of the more than one component. For example, if two components' sample values are used to predict the one component's sample value, each point/set can be represented as a position in a 3-dimensional space, and the linear model is based on a straight line passing through the two positions in the 3-dimensional space that correspond to the two points/sets of the reconstructed sample values.

Figure 13 is a schematic block diagram of a computing device **1300** for implementation of one or more embodiments of the invention. The computing device **1300** may be a device such as a micro-computer, a workstation or a light portable device. The computing device **1300** comprises a communication bus connected to:

- a central processing unit **1301**, such as a microprocessor, denoted CPU;

- a random access memory **1302**, denoted RAM, for storing the executable code of the method of embodiments of the invention as well as the registers adapted to record variables and parameters necessary for implementing the method for encoding or decoding at least part of an image according to embodiments of the invention, the memory capacity thereof can be expanded by an optional RAM connected to an expansion port for example;

- a read only memory **1303**, denoted ROM, for storing computer programs for implementing embodiments of the invention;

- a network interface **1304** is typically connected to a communication network over which digital data to be processed are transmitted or received. The network interface **1304** can be a single network interface, or composed of a set of different network interfaces (for instance wired and wireless interfaces, or different kinds of wired or wireless interfaces). Data packets are written to the

network interface for transmission or are read from the network interface for reception under the control of the software application running in the CPU **1301**;

- a user interface **1305** may be used for receiving inputs from a user or to display information to a user;

- a hard disk **1306** denoted HD may be provided as a mass storage device;

- an I/O module **1307** may be used for receiving/sending data from/to external devices such as a video source or display.

The executable code may be stored either in read only memory **1303**, on the hard disk **1306** or on a removable digital medium such as for example a disk. According to a variant, the executable code of the programs can be received by means of a communication network, via the network interface **1304**, in order to be stored in one of the storage means of the communication device **1300**, such as the hard disk **1306**, before being executed.

The central processing unit **1301** is adapted to control and direct the execution of the instructions or portions of software code of the program or programs according to embodiments of the invention, which instructions are stored in one of the aforementioned storage means. After powering on, the CPU **1301** is capable of executing instructions from main RAM memory **1302** relating to a software application after those instructions have been loaded from the program ROM **1303** or the hard-disc (HD) **1306** for example. Such a software application, when executed by the CPU **1301**, causes the steps of the method according to the invention to be performed.

Any step of the methods according to the invention may be implemented in software by execution of a set of instructions or program by a programmable computing machine, such as a PC ("Personal Computer"), a DSP ("Digital Signal Processor") or a microcontroller; or else implemented in hardware by a machine or a dedicated component, such as an FPGA ("Field-Programmable Gate Array") especially for the Minima and Maxima selection, or an ASIC ("Application-Specific Integrated Circuit").

It is also to be noted that while some examples are based on HEVC for the sake of illustration, the invention is not limited to HEVC. For example, the present invention can also be used in any other prediction/estimation process where a relationship between two or more components' sample values can be estimated/predicted with a model, wherein the model is an approximate model determined based on at least two sets of related/associated component sample values selected from all available sets of the related/associated component sample values.

It is understood that each point corresponding to a sample pair (i.e. a set of associated sample values for different components) may be stored and/or processed in terms of an array. For example, each component's sample values may be stored in an array so that each sample value of that component is referable/accessible/obtainable by referencing an element of that array, using an index for that sample value for example. Alternatively, an array may be used to store and process each sample pairs that each sample value of the sample pairs accessible/obtainable as an element of the array.

It is also understood that any result of comparison, determination, assessment, selection, or consideration described above, for example a selection made during an encoding process, may be indicated in or determinable from data in a bitstream, for example a flag or data indicative of the result, so that the indicated or determined result can be used in the processing instead of actually performing the comparison, determination, assessment, selection, or consideration, for example during a decoding process.

Although the present invention has been described hereinabove with reference to specific embodiments, the present invention is not limited to the specific embodiments, and modifications will be apparent to a skilled person in the art, which lie within the scope of the present invention.

Many further modifications and variations will suggest themselves to those versed in the art upon making reference to the foregoing illustrative embodiments, which are given by way of example only and which are not intended to limit the scope of the invention, that being determined solely by the appended claims. In

particular the different features from different embodiments may be interchanged, where appropriate.

Each of the embodiments of the invention described above can be implemented solely or as a combination of a plurality of the embodiments. Also, features from different embodiments can be combined where necessary or where the combination of elements or features from individual embodiments in a single embodiment is beneficial.

In the claims, the word “comprising” does not exclude other elements or steps, and the indefinite article “a” or “an” does not exclude a plurality. The mere fact that different features are recited in mutually different dependent claims does not indicate that a combination of these features cannot be advantageously used.

CLAIMS

1. A method of deriving a linear model for obtaining a chroma sample of a chroma block from an associated luma sample of a luma block in the same frame, the method comprising:

determining two pairs of values for determining the linear model, each pair being defined by two variables, a first variable of said two variables corresponding to a luma sample value, a second variable of said two variables corresponding to a chroma sample value,

wherein the two pairs for determining the linear model are determined based on sample values in a neighbourhood of the chroma block or the luma block,

wherein each value in said two pairs of values for determining the linear model is determined by averaging two values based on the sample values in the neighbourhood; and

determining parameters of the linear model by calculating a difference between the luma sample values of the two pairs of values for determining the linear model and a difference between the chroma sample values of the two pairs of values for determining the linear model.

2. The method of claim 1, wherein:

- the samples of the luma block are organized into at least two groups; and
- two pairs of values for determining the linear model are determined for the definition of a linear model for each group of samples of the luma block.

3. The method of claim 2, wherein if the two pairs of values for determining the linear model determined for a group correspond to a slope

parameter lower than a predetermined threshold, then they are replaced by two pairs determined for another group.

4. The method of claim 2, wherein if the two pairs of values for determining the linear model determined for a group correspond to a slope parameter lower than a predetermined threshold, then two new pairs are determined based on the samples of all the groups considered as a single group.

5. A method of obtaining a chroma sample for a chroma block from an associated luma sample of a luma block in the same frame, the method comprising:

- defining a plurality of linear model derivation modes comprising Cross Component Linear Model modes using a single linear model and Multi Model Linear Model modes using several linear models; and
- selecting one of the linear model derivation modes for obtaining the chroma samples for a chroma block,

wherein:

- at least one of the linear model derivation modes uses a method of derivation according to any one claim from 1 to 4.

6. The method of claim 5, wherein only the Cross Component Linear Model modes use a method of derivation according to any one claim from 1 to 4.

7. The method of claim 5, wherein only the Multi Model Linear Model modes use a method of derivation according to any one claim from 1 to 4.

8. A method of encoding one or more images into a bitstream, wherein the method comprises deriving a linear model according to any one of claims 1 to 4.

9. A method of encoding one or more images into a bitstream, wherein the method comprises obtaining a chroma sample for a chroma block of the one or more images from an associated reconstructed luma sample block according to any one of claims 5 to 7.

10. A method of decoding one or more images from a bitstream, wherein the method comprises deriving a linear model according to any one of claims 1 to 4.

11. A method of decoding one or more images from a bitstream, wherein the method comprises obtaining a chroma sample for a chroma block of the one or more images from an associated reconstructed luma sample block according to any one of claims 5 to 7.

12. A device for deriving a linear model for obtaining a chroma sample of a chroma block from an associated luma sample of a luma block in the same frame, the device comprising a means for:

determining two pairs of values for determining the linear model, each pair being defined by two variables, a first variable of said two variables corresponding to a luma sample value, a second variable of said two variables corresponding to a chroma sample value,

wherein the two pairs of values for determining the linear model are determined based on a plurality of chroma sample values in a neighbourhood of the chroma block or the luma block and a plurality of corresponding down-sampled luma sample values,

wherein each value in said two pairs of values for determining the linear model is determined by averaging two values based on the sample values in the neighbourhood; and

determining parameters of the linear model by calculating a difference between the luma sample values of the two pairs of values for determining the linear model and a difference between the chroma sample values of the two pairs of values for determining the linear model.

13. The device of claim 12, wherein:

- the samples of the luma block are organized into at least two groups; and
- two pairs of values are determined for the definition of a linear model for each group of samples of the luma block.

14. The device of claim 13, wherein if the two pairs of values for determining the linear model determined for a group correspond to a slope parameter lower than a predetermined threshold, then they are replaced by two pairs determined for another group.

15. The device of claim 13, wherein if the two pairs of values for determining the linear model determined for a group correspond to a slope parameter lower than a predetermined threshold, then two new pairs are determined based on the samples of all the groups considered as a single group.

16. A device for obtaining a chroma sample for a chroma block from an associated luma sample of a luma block in the same frame, the device comprising a means for:

- defining a plurality of linear model derivation modes comprising Cross Component Linear Model modes using a single linear model and Multi Model Linear Model modes using several linear models;
- selecting one of the linear model derivation modes for obtaining the chroma samples for a chroma block;

wherein:

- at least some of the linear model derivation modes using a method of derivation according to any one claim from 1 to 4.

17. The device of claim 16, wherein only the Cross Component Linear Model modes use a method of derivation according to any one claim from 1 to 4.

18. The device of claim 16, wherein only the Multi Model Linear Model modes use a method of derivation according to any one claim from 1 to 4.

19. A device for encoding images, wherein the device comprises a means for deriving a linear model according to any one claim 1 to 4.

20. A device for decoding images, wherein the device comprises a means for deriving a linear model according to any one claim 1 to 4.

21. A computer program product for a programmable apparatus, the computer program product comprising a sequence of instructions for implementing a method according to any one of claims 1 to 11, when loaded into and executed by the programmable apparatus.

22. A computer-readable medium storing a program which, when executed by a microprocessor or computer system in a device, causes the device to perform a method according to any one of claims 1 to 11.

23. A computer program which upon execution causes the method of any one of claims 1 to 11 to be performed.