

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
21 September 2006 (21.09.2006)

PCT

(10) International Publication Number
WO 2006/098494 A1

(51) International Patent Classification:
H04N 11/04 (2006.01) *H04N 1/41* (2006.01)

(21) International Application Number:
PCT/JP2006/305640

(22) International Filing Date: 15 March 2006 (15.03.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/907,080 18 March 2005 (18.03.2005) US
10/907,082 18 March 2005 (18.03.2005) US

(71) Applicant (for all designated States except US): **SHARP KABUSHIKI KAISHA** [JP/JP]; 22-22, Nagaïke-cho, Abeno-ku, Osaka-shi, Osaka, 5458522 (JP).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **SUN, Shijun. LEI, Shawmin. KATATA, Hiroyuki.**

(74) Agent: **TAKANO, Akichika**; NAGISA PATENT OFFICE, 9th Floor, Salute Bldg., 72, Yoshida-cho, Naka-ku, Yokohama-shi, Kanagawa 2310041 (JP).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

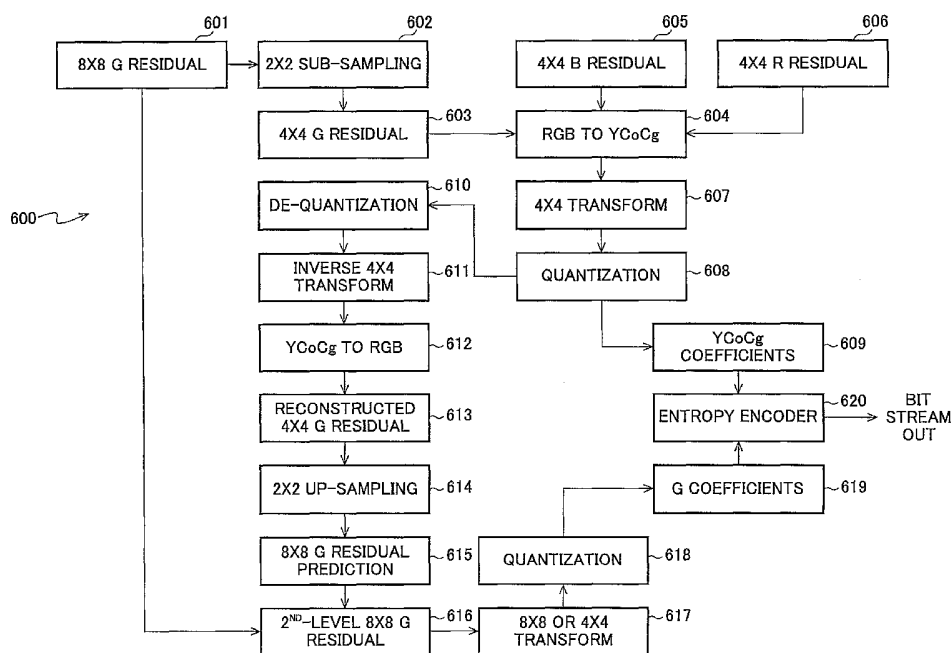
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

- with international search report
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: VIDEO COMPRESSION USING RESIDUAL COLOR TRANSFORM



(57) Abstract: A Residual Color Transform (RCT) technique directly encodes raw Red-Green-Blue (RGB) data directly without first performing a color transform. After transmission or storage, the encoded raw RGB data is directly decoded and then interpolated to generate missing RGB data.

DESCRIPTION

VIDEO COMPRESSION USING RESIDUAL COLOR TRANSFORM

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to video coding. In particular, the present invention relates to a system and a method for encoding and decoding color video data.

2. Description of the Related Art

Residual Color Transform (RCT) is a coding tool for the H.264 High 4:4:4 profile that is intended for efficient coding of video sequences in a Red-Green-Blue-format (RGB-format). [1]

[1] W.S. Kim, D. Birinov, and H. M. Kim, "Adaptive Residual Transform and Sampling," ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/SG16, Document JVT-K018, March 2004.

In this description, the term "coder" includes the concept of "encoder" and/or "decoder". Similarly, the term "coding" includes the concept of "encoding" and/or "decoding".

Figures 1 and 2 illustrate the difference between a conventional video coding system that does not use the RCT coding tool and a conventional video coding system that uses the RCT coding tool. Details regarding the encoding and decoding loops, and the prediction and compensation loops are not shown in either of Figures 1 or 2.

Figure 1, in particular, depicts a high-level functional block diagram of a conventional video coding system 100 that does not use the RCT coding tool. Conventional video coding system 100 captures Red-Green-Blue (RGB) data in a well-known manner at 101. At 102, the RGB data is converted into a YCbCr (or YCoCg) format. At 103, intra/inter prediction is performed on the YCbCr-formatted (or YCoCg-formatted) data. A spatial transform is performed at 104 and quantization is performed at 105. Entropy encoding is performed at 106. The encoded data is transmitted and/or stored, as depicted by channel/storage 107. At 108, the encoded data is entropy decoded. At 109, the entropy-decoded data is de-quantized. An inverse-spatial transform is performed at 110, and intra/inter compensation is performed at 111. At 112, the resulting YCbCr-formatted (or YCoCg-formatted) data is transformed to RGB-based data and displayed at 113. YCoCg is a format which enables reversible color transform defined in the H.264 standard.

Figure 2 depicts a high-level functional block diagram of a conventional video coding system 200 that uses the RCT coding tool for the H.264 High 4:4:4 profile. Video coding system 200 captures RGB data in a well-known manner at 201. At 202, intra/inter prediction is performed on the RGB data. At 203, the intra/inter-predicted data is converted into a YCbCr (or YCoCg) format. A spatial transform is performed at 204 and quantization is performed at 205. Entropy encoding is performed at 206. The encoded data is transmitted and/or stored, as depicted by

channel/storage 207. At 208, the encoded data is entropy decoded. At 209, the entropy-decoded data is de-quantized. An inverse-spatial transform is performed on YCbCr-formatted (or YCoCg-formatted) data at 210. At 211, the YCbCr-based data is transformed to RGB-based data. At 212, intra/inter compensation is performed, and RGB-based data is displayed at 213.

The difference between conventional video coding system 100 (Figure 1) and conventional video coding system 200 (Figure 2) is that the RCT coding tool of system 200 enables compression and decompression directly to and from the RGB space. To illustrate this, compression directly in the RGB space is depicted in Figure 2 by the sequence of functional blocks 202-204. In particular, intra/inter prediction is performed on the RGB data at 202. The intra/inter-predicted data is converted into a YCbCr (or YCoCg) format at 203. A spatial transform is then performed at 204. Decompression directly from the RGB space is depicted in Figure 2 by the sequence of functional blocks 210-212. At 210, an inverse-spatial transform is performed on YCbCr-formatted (or YCoCg-formatted) data. At 211, the YCbCr-formatted data is transformed to RGB-based data. At 212, intra/inter compensation is performed.

In contrast, the corresponding compression process in conventional video coding system 100 is depicted by functional blocks 102-104. At 102, RGB data is converted into a YCbCr (or YCoCg) format. Intra/inter prediction is performed on the YCbCr-formatted (or YCoCg-

formatted) data at 103, and a spatial transform is performed at 104. The corresponding decompression process is depicted by functional blocks 110-112 in which an inverse spatial transform is performed at 110. Intra/inter compensation is performed at 111. Lastly, the YCbCr (or YCoCg) data is transformed to RGB-based data at 112.

The color conversion in RCT at 203 in Figure 2, as an RGB-based format to a YCoCg-based format, is inside a typical coding loop, and can be considered as an extension of a conventional transform coding from a 2D spatial transform to a 3D transform (2D spatial + 1D color), but, with the same purpose of all transform coding, that is, data decorrelation and energy compaction and; consequently, easier compression. Significant improvements in rate distortion performance over a conventional coding scheme have been achieved for all three RGB color components, as demonstrated in an updated version of the RCT algorithm.

The main challenge for RCT, as RCT is applied in practice, does not related to compression, but relates to video capture and display. Moreover, the outputs of most video-capture devices currently do not support the RGB format not because extra hardware and software resources are needed internally to convert data from RGB-based data but based on the bandwidth requirements for the 4:4:4 RGB format. The term "4:4:4" here is used to express that each pixel position has three color components.

For a single-chip-color-sensor digital video camera, each pixel

actually has only one color component. Figure 3 depicts a typical Bayer mosaic filter 300, which is used for most of the popular primary-color-mosaic sensors. As depicted in Figure 3, the Green (G) sensors, or filters, cover 50 % of the pixels, while the Blue (B) and Red (R) sensors, or filters, each cover 25 % of the pixels. This format is called "raw RGB" in this specification. The numbers shown in Figure 3 will be used later to explain interleaving process of this invention shown in Figure 19.

Because each pixel position in Figure 3 has only one color component, the other two color components must be interpolated, or generated, based on existing samples. Each pixel position has three color components after the interpolation process. Then the interpolation process is a simple 1:3 data expansion. In other words, one color component is expanded to three color components at each pixel position.

Figure 4 depicts a high-level functional block diagram of a conventional video coding system 400 that provides a lossy color conversion, such as from RGB to YCbCr. At 401, RGB sensors perform RGB capture. At 402, an interpolation process of 1:3 data expansion is performed for generating missing color components. At 403, color conversion from 4:4:4 RGB to 4:4:4 YCbCr and a lossy 2:1 sub-sampling is performed for generating 4:2:0 YCbCr data. Since the sub-sampling from 4:4:4 YCbCr to 4:2:0 YCbCr makes the data rate one half, this is called "2:1 sub-sampling". Thus, the overall process up to functional block 404 results in a lossy 1:1.5 data expansion before the 4:2:0 YCbCr data is

compressed, i.e., video is encoded. The encoded data is transmitted and/or stored, as depicted by channel/storage 405. The video encoded data is then decoded at 406. Color up-sampling and color conversion occurs at 407, and the resulting data is 4:4:4 RGB displayed at 408.

What is needed is (i) a residual color transform (RCT) coding tool for a 4:2:0 RGB data (the definition of 4:2:0 RGB format will be given in the part of "detailed description of the present invention") in which compression is performed directly on 4:2:0 RGB data without data loss prior to compression and (ii) a residual color transform (RCT) coding tool for raw RGB data in which compression is performed directly on the raw RGB data without data loss prior to compression.

SUMMARY OF THE INVENTION

The present invention provides a residual color transform (RCT) coding technique for 4:2:0 RGB data or raw RGB data in which compression is performed directly on 4:2:0 RGB data or raw RGB data without first performing a color transform.

The present invention provides a Residual Color Transform (RCT) coding method for 4:2:0 Red-Green-Blue (RGB) data in which RGB data is interpolated to generate at least one missing Green color component to form 4:2:0 RGB data and then directly encoded. According to the present invention, video encoding of the 4:2:0 RGB data encodes the 4:2:0 RGB data without data loss. Additionally, interpolating RGB data

includes using a 1:1.5 expansion technique. More specifically, directly encoding the 4:2:0 RGB-based data includes sub-sampling an 8×8 Green residual to form a single 4×4 Green residual and then converting the 4×4 Green residual, a corresponding 4×4 Red residual and a corresponding 4×4 Blue residual to YCoCg-based data. The YCoCg-based data is 4×4 transformed and quantized to form YCoCg coefficients. The YCoCg coefficients are then encoded into a bitstream.

Encoding the YCoCg coefficients into the bitstream further includes de-quantizing the YCoCg coefficients and inverse 4×4 transforming the de-quantized YCoCg coefficients to reconstruct the YCoCb-based data. The YCoCg-based data is converted to RGB-based data to form 4×4 G residual data. The 4×4 G residual data is reconstructed and 2×2 up-sampled to form an 8×8 G residual prediction. A 2nd-level 8×8 Green residual is formed based on a difference between the 8×8 G residual prediction and the 8×8 Green residual. The 2nd-level 8×8 Green residuals is transformed by a 4×4 transformation or an 8×8 transformation and quantized to form Green coefficients. The Green coefficients are then encoded into the bitstream. After storage and/or transmission, the encoded 4:2:0 RGB data is directly decoded and interpolating for generating at least one of a missing Blue color component and a missing Red color component prior to display.

The present invention also provides a method of entropy decoding the bitstream to form YCoCg coefficients, de-quantizing the

YCoCg coefficients, inverse-transforming the de-quantized YCoCg coefficients to form an 8×8 Green residual prediction and 4×4 Red and Blue residuals, and forming an 8×8 Green residual from the 8×8 Green residual prediction. According to the invention, forming the 8×8 Green residual from the 8×8 Green residual prediction includes entropy decoding the bitstream to form Green coefficients, de-quantizing the Green coefficients to form an 8×8 Green residual, inverse-transforming the de-quantized Green coefficients to form a 2nd-level 8×8 Green residual, and combining the 2nd-level 8×8 Green residual with the 8×8 Green residual prediction to form the 8×8 Green residual.

The present invention also provides a video coding system that directly codes 4:2:0 RGB data using a RCT coding tool by interpolating RGB data to generate at least one missing Green color component to form 4:2:0 RGB data and then directly encoding the 4:2:0 RGB-based data. The system video encodes the 4:2:0 RGB data without data loss. The system also directly decodes the encoded 4:2:0 RGB data, and interpolates the decoded 4:2:0 RGB data for generating at least one of a missing Blue color component and a missing Red color component prior to display.

The system includes a sub-sampler sub-sampling an 8×8 Green residual to form a single 4×4 Green residual, a converter converting the 4×4 Green residual, a corresponding 4×4 Red residual and a corresponding 4×4 Blue residual to YCoCg-based data, a

transformer 4×4 transforming the YCoCg-based data, a quantizer quantizing the 4×4 transformed YCoCg-based data to form YCoCg coefficients, and an entropy encoder encoding the YCoCg coefficients into a bitstream. The system also includes a de-quantizer de-quantizing the YCoCg coefficients, an inverse transformer inverse 4×4 transforming the de-quantized YCoCg coefficients to reconstruct the YCoCb-based data, a converter converting the YCoCg-based data to RGB-based data, a reconstructor reconstructing 4×4 G residual data, an up-sampler 2×2 up-sampling the 4×4 G residual data to form an 8×8 G residual prediction, a differencer forming a 2nd-level 8×8 Green residuals based on a difference between the 8×8 G residual prediction and the 8×8 Green residuals, a second transformer transforming the 2nd-level 8×8 Green residuals by one of a 4×4 transformation and an 8×8 transformation, and a second quantizer quantizing the transformed 2nd-level 8×8 Green residual to form Green coefficients. The entropy encoder further encodes the Green coefficients into the bitstream.

The present invention also provides a decoder that includes a entropy decoder that entropy decodes the bitstream to form YCoCg coefficients, a first de-quantizer that de-quantizes the YCoCg coefficients, a first inverse-transformer that inverse-transforms the de-quantized YCoCg coefficients to form an 8×8 Green residual prediction and 4×4 Red and Blue residuals, and a residual former that forms an 8×8 Green residual from the 8×8 Green residual prediction. Additionally, the

entropy decoder entropy decodes the bitstream to form Green coefficients. The residual former includes a second de-quantizer that de-quantizes the Green coefficients to form an 8×8 Green residual, a second inverse-transformer that inverse-transforms the de-quantized Green coefficients to form a 2nd-level 8×8 Green residual, and a combiner that combines the 2nd-level 8×8 Green residual with the 8×8 Green residual prediction to form the 8×8 Green residual.

The present invention also provides a Residual Color Transform (RCT) encoding method for encoding Red-Green-Blue (RGB) data comprising video encoding raw RGB data using an RCT encoding tool. The video encoding of raw RGB data encodes the raw RGB data directly without first performing a color transform. After transmission or storage, the encoded raw RGB data is directly decoded. The decoded raw RGB data is then interpolated to generate at least one of a missing Red color component, a missing Green color component and a missing Blue color component.

One exemplary embodiment of the present invention provides a closed-loop encoding technique in which two 4×4 Green residuals are sub-sampled to form a single 4×4 Green residual. The single 4×4 Green residual, a corresponding 4×4 Red residual and a corresponding 4×4 Blue residual are converted to YCoCg-based data. The YCoCg-based data is 4×4 transformed and quantized to form YCoCg coefficient. The YCoCg coefficients are then encoded into a bitstream.

For this exemplary embodiment, encoding the YCoCg coefficients into the bitstream includes de-quantizing the YCoCg coefficients and 4×4 inverse transforming the de-quantized YCoCg coefficients to reconstruct the YCoCb-based data. The YCoCg-based data is converted to RGB-based data and 4×4 G residual data is reconstructed. The 4×4 G residual data is up-sampled to form two interleaved 4×4 G residual prediction blocks. Two 2nd-level interleaved 4×4 Green residuals are formed based on a difference between the two interleaved 4×4 G residual prediction blocks and the two interleaved 4×4 Green residual blocks. The two 2nd-level 4×4 Green residuals are 4×4 transformed. The two transformed 2nd-level 4×4 Green residuals are quantized to form Green coefficients that are encoded into the bitstream.

The present invention also provides a method for decoding the bitstream to form YCoCg coefficients in which the YCoCg coefficients are de-quantized to form de-quantized YCoCg coefficients. The de-quantized YCoCg coefficients are inverse-transformed to form YCoCg-based data. The YCoCg-based data are YCoCg-to-RGB converted to form a reconstructed 4×4 Green residual, a corresponding reconstructed 4×4 Red residual and a corresponding reconstructed 4×4 Blue residual. The reconstructed 4×4 Green residual is up-sampled to form two interleaved reconstructed 4×4 Green residual predictions. Forming two interleaved reconstructed 4×4 Green residuals includes decoding the bitstream to form 2nd-level Green coefficients, de-quantizing the 2nd-level Green

coefficients to form de-quantized 2nd-level Green coefficients, inverse-transforming the de-quantized 2nd-level Green coefficients to form two interleaved 2nd-level 4×4 Green residuals, and combining the two 2nd-level 4×4 Green residuals with the two interleaved reconstructed 4×4 Green residual predictions to form the two reconstructed 4×4 Green residuals.

Another exemplary embodiment provides an open-loop encoding technique in which two interleaved 4×4 blocks of Green prediction residuals are Haar transformed to form an averaged 4×4 G residual and a differentiated 4×4 G residual. The averaged 4×4 Green residual, a corresponding 4×4 Red residual and a corresponding 4×4 Blue residual are converted to YCoCg-based data. The YCoCg-based data is 4×4 transformed and then quantized to form YCoCg coefficients. The YCoCg coefficients are encoded into a bitstream.

For this exemplary embodiment, encoding the YCoCg coefficients into the bitstream includes transforming and then quantizing the differentiated 4×4 G residual to form Green coefficients. The Green coefficients are then entropy encoded into the bitstream.

The present invention also provides a method for decoding the bitstream to form YCoCg coefficients. The YCoCg coefficients are de-quantized to form de-quantized YCoCg coefficients. The de-quantized YCoCg coefficients are inverse-transformed to form YCoCg-based data. The YCoCg-based data are YCoCg-to-RGB converted to form a

reconstructed 4×4 Green residual, a corresponding reconstructed 4×4 Red residual and a corresponding reconstructed 4×4 Blue residual. Two reconstructed interleaved 4×4 Green residuals are formed from the reconstructed 4×4 Green residual. Forming two reconstructed interleaved 4×4 Green residuals includes decoding the bitstream to form 2nd-level Green coefficients, de-quantizing the 2nd-level Green coefficients to form de-quantized 2nd-level Green coefficients, inverse-transforming the de-quantized 2nd-level Green coefficients to form a 2nd-level 4×4 Green residual, and inverse-Haar transforming the 2nd-level 4×4 Green residual with the reconstructed 4×4 Green residual to form the two interleaved reconstructed 4×4 Green residuals.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not by limitation in the accompanying figures in which like reference numerals indicate similar elements and in which:

Figure 1 depicts a high-level functional block diagram of a conventional video coding system that does not use the RCT coding tool;

Figure 2 depicts a high-level functional block diagram of a conventional video coding system that uses the RCT coding tool for the H.264 High 4:4:4 profile;

Figure 3 depicts a typical Bayer mosaic filter, which is used for most of the popular primary-color-mosaic sensors;

Figure 4 depicts a high-level functional block diagram of a conventional video coding system that provides a lossy color conversion, such as from RGB to YCbCr;

Figure 5 depicts a high-level block diagram of a video coding system according to the present invention;

Figure 6 shows a flow diagram of a closed-loop residual encoding technique using RCT for 4:2:0 RGB data according to the present invention;

Figure 7 shows a block diagram of a closed-loop residual encoding technique using RCT for 4:2:0 RGB data according to the present invention;

Figure 8 shows a flow diagram of a closed-loop residual decoding technique using RCT for 4:2:0 RGB data according to the present invention;

Figure 9 shows a block diagram of a closed-loop residual decoding technique using RCT for 4:2:0 RGB data according to the present invention;

Figure 10 shows a block diagram of an open-loop residual encoding technique using RCT for 4:2:0 RGB data according to the present invention;

Figure 11 shows a block diagram of an open-loop residual decoding technique using RCT for 4:2:0 RGB data according to the present invention;

Figure 12 shows an example of a sub-band analysis operation;

Figure 13 shows another example of a sub-band analysis operation;

Figure 14 shows a block diagram of another open-loop residual encoding technique using RCT for 4:2:0 RGB data according to the present invention;

Figure 15 shows a block diagram of another open-loop residual decoding technique using RCT for 4:2:0 RGB data according to the present invention;

Figure 16 shows an example of a division performed using a selector;

Figure 17 depicts a high-level block diagram of a video coding system according to the present invention;

Figure 18 shows a flow diagram of a closed-loop residual encoding technique using RCT for raw RGB data according to the present invention;

Figure 19 depicts two exemplary interleaved 4×4 G residuals, an exemplary 4×4 B residual and an exemplary 4×4 R residual for the present invention;

Figure 20 shows a data expansion to the out side of the 4×4 block for filtering;

Figure 21 shows a block diagram of a closed-loop residual encoding technique using RCT for raw RGB data according to the present

invention;

Figure 22 shows a flow diagram of a closed-loop residual decoding technique using RCT for raw RGB data according to the present invention;

Figure 23 shows a block diagram of a closed-loop residual decoding technique using RCT for raw RGB data according to the present invention;

Figure 24 shows a flow diagram of an open-loop residual encoding technique using RCT for raw RGB data according to the present invention;

Figure 25 shows a block diagram of an open-loop residual encoding technique using RCT for raw RGB data according to the present invention;

Figure 26 shows a flow diagram of an open-loop residual decoding technique using RCT for raw RGB data according to the present invention;

Figure 27 shows a block diagram of an open-loop residual decoding technique using RCT for raw RGB data according to the present invention;

Figure 28 shows a block diagram of another open-loop residual encoding technique using RCT for raw RGB data according to the present invention; and

Figure 29 shows a block diagram of another open-loop

residual encoding technique using RCT for raw RGB data according to the present invention.

DETAILED DESCRIPTION OF THE PRESENT INVENTION

The present invention provides a Residual Color Transform (RCT) coding tool for 4:2:0 RGB in which compression is performed directly on 4:2:0 RGB without data loss prior to compression.

Figure 5 depicts a high-level block diagram of a video coding system according to the present invention. In particular, RGB sensors perform RGB capture in a well-known manner at 501. If the size of an encoding block is 8×8 pixels, the raw RGB data in the encoding block contains 32 Green data in a quincunx pattern, 4×4 Blue data in a grid pattern and 4×4 Red data in a grid pattern as shown in Figure 3. Interpolation and generation of missing Green data is performed at 502 using a 1:1.5 expansion. As a result, the full resolution Green data and the original Blue and Red data together form the 4:2:0 RGB data. Then, the 4:2:0 RGB data in the encoding block contains 8×8 Green data, 4×4 Blue data and 4×4 Red data all in a grid pattern. The encoding process at 503 operates directly on the 4:2:0 RGB data using the RCT coding tool. The sampling positions of the RGB data are different within each pixel and the positions can change from picture to picture. Consequently, the R/B sampling positions are signaled in the bitstream at sequence and/or each picture and are then used for motion vector interpolation and final

display rendering. For example, a zero-motion motion compensation for R/B might actually correspond to a non-zero motion in G.

The encoded 4:2:0 RGB data is then transmitted and/or stored, as depicted by channel/storage 504. The decoding process operates directly on the 4:2:0 RGB data at 505. At 506, interpolation is performed for generating missing Blue and Red color components. The resulting data is RGB displayed at 507.

Interpolation for the Blue and Red color components (functional block 506) is deferred in the present invention until the bitstreams have been decoded. Additionally, it should be noted that the Blue and Red color component interpolation (functional block 506) could be part of a post-processing for video decoding at 505 or part of a preprocessing for RGB display at 507.

Figure 6 shows a flow diagram 600 of a residual encoding technique according to the present invention using RCT for 4:2:0 RGB data. Flow diagram 600 corresponds to the second part of processes that occur in block 503 in Figure 5, which also includes as its first part of processes Intra/Inter Prediction, which is similar to block 202 in Figure 2 except that the Prediction is done in the present invention based on 4:2:0 RGB data, not on grid-pattern RGB data. Thus, the process depicted in Figure 6 corresponds only to the residual encoding, including transforms (spatial and RCT), quantization, and entropy encoding modules. Prediction and motion compensation are done in the 4:2:0 RGB domain

and are not depicted in Figure 6. Block 601 in Figure 6 represents the 8×8 block of Green (G) prediction residuals that are obtained in the first part of block 503 in Figure 5. At block 602, the 8×8 block of G prediction residuals is sub-sampled using 2×2 sub-sampling to produce a 4×4 block of G residuals at block 603. The sub-sampling at block 602 could be, for example, an averaging operation. Alternatively, any low-pass or decimation filtering technique could be used for block 602. At block 604, the 4×4 G residuals together with the 4×4 Blue (B) residuals (block 605) and the 4×4 block of Red (R) residuals (block 606) are converted from RGB-based data to YCoCg-based data. At block 607, the YCoCg data goes through a 4×4 transformation and is quantized at block 608 to produce YCoCg coefficients at block 609. The YCoCg coefficients are encoded into bitstreams by an entropy encoder at block 620.

The YCoCg coefficients generated at block 608 are de-quantized at block 610 and inverse 4×4 transformed at block 611 to reconstruct the YCoCg-based data before being converted to RGB-based data at block 612 to form a reconstructed 4×4 G residual at block 613. The 4×4 G residual is 2×2 up-sampled at block 614 to form an 8×8 G prediction residual at block 615. The up-sampling process at block 614 could be, for example, a duplicative operation. Alternatively, any interpolation filtering technique could be used for block 614. The differences between the 8×8 G residual at block 601 and the 8×8 G residual prediction at block 615 are used to form the second-level 8×8 G

residual at block 616. The second-level 8×8 G residual goes through a transformation at 617 and a quantization process at block 618 to form Green (G) coefficients at block 619. The G coefficients are encoded into bitstreams by the entropy encoder at block 620.

Figure 7 is a block diagram of a video encoder which includes a residual encoder corresponding to the residual encoding technique of Figure 6. At block 701, RGB sensors perform RGB capture to create raw RGB data. At block 702, missing Green data in the raw RGB data is interpolated and 4:2:0 RGB data is created. At block 703, Intra/Inter prediction is performed to generate residual 4:2:0 RGB data. This functional block includes inter prediction portion and intra prediction portion. The inter prediction portion contains frame memories for motion compensation prediction and motion estimation portion to generate motion vectors.

Next, each encoding block of 8×8 pixels is encoded by the technique of the present invention. At block 708, 8×8 Green residual data is sub-sampled to generate 4×4 Green residual data and 4×4 RGB residual data is color transformed to generate 4×4 YCoCg residual data at block 704. In block diagrams of this description, even residual data such as 4×4 Green residual data is shown as " 4×4 G" for simplicity.

Then 4×4 spatial transform is performed to generate 4×4 YCoCg coefficient data at block 705 and it is quantized at block 706. The 4×4 YCoCg quantized coefficient data is encoded into bitstream at block

707 and, at the same time, de-quantized to generate 4×4 YCoCg de-quantized coefficient data at block 709.

The de-quantized coefficient data is then 4×4 inverse transformed to generate 4×4 YCoCg reconstructed data at block 710 and the data is converted 4×4 RGB reconstructed residual data at block 711. The 4×4 G reconstructed residual data is up-sampled to generate 8×8 G interpolated residual data at 712. This process of reconstruction is called local decoding at an encoder.

The difference data (second-level 8×8 G residual) between the 8×8 G residual at block 703 and the 8×8 G interpolated residual data at 712 is 8×8 (or 4×4) transformed at block 713 and quantized at block 714 to generate 8×8 G quantized coefficient data. The 8×8 G quantized coefficient data is encoded into bitstream at block 707.

Figure 8 shows a flow diagram 800 of a closed-loop residual decoding technique using RCT for 4:2:0 RGB data according to the present invention. The process depicted in Figure 8 corresponds only to the residual decoding, including inverse transforms (spatial and RCT), de-quantization, and entropy decoding modules. Prediction and motion compensation are done in the raw RGB domain and are not depicted in Figure 8.

At block 801, a bitstream is entropy decoded by an entropy decoder to form G coefficients at 802 and YCoCg coefficients at 807. The G coefficients are de-quantized at 803 and an 8×8 or a 4×4 inverse

transform is performed at 804 to form a 2nd-level 8×8 G residual at 805.

The YCoCg coefficients at 807 are de-quantized at 808 and 4×4 inverse transformed at 809. At 810 the YCoCg coefficients are transformed to RGB-based data. A reconstructed 4×4 B residual is formed at 811, a reconstructed 4×4 R residual is formed at 812, and a reconstructed 4×4 G residual is formed at 813. The reconstructed 4×4 G residual is up-sampled at 814 to form an 8×8 G residual prediction.

At 806, the 2nd-level 8×8 G residual (at 805) are summed with the 8×8 G residual prediction (at 815) to form a reconstructed 8×8 G residuals at 816.

Figure 9 is a block diagram of a video decoder which includes a residual decoder corresponding to the residual decoding technique of Figure 8. The bitstream is decoded at block 901 to generate 8×8 G quantized coefficient data and 4×4 YCoCg quantized coefficient data. The 4×4 YCoCg quantized coefficient data is converted to 8×8 G interpolated residual data and 4×4 Red and Blue reconstructed residual data at blocks from 904 to 907 which are identical to the blocks from 709 to 712 in Figure 7. The 8×8 G quantized coefficient data is de-quantized at block 902, 8×8 (or 4×4) inverse transformed at block 903 and added by the data from block 907 to generate 8×8 G residual data.

At block 908, 4:2:0 RGB data is reconstructed from 8×8 G residual data and 4×4 Red and Blue residual data by performing

Intra/Inter prediction.

The technique shown in Figure 6, Figure 7, Figure 8 and Figure 9 is called closed-loop residual coding technique in which the quantized data is reconstructed by the local decoder and the second-level residual is encoded at an encoder. The next example shown in Figure 10 and Figure 11 is an open-loop residual coding technique in which the local decoder and the second-level residual do not exist.

Figure 10 is a block diagram of a video encoder for open-loop residual encoding technique. The blocks from 1001 to 1006 are identical to the blocks from 701 to 706 in Figure 7.

At block 1008 in Figure 10, 8×8 Green residual data is divided to LL, LH, HL and HH sub-band data by a sub-band analysis. Figure 12 illustrates the example of the sub-band analysis operation as follows.

- (1) The original 8×8 data of Figure 12(a) is divided to 4×8 low band (L) and high band (H) data by a horizontal sub-band analysis.
- (2) The low band data of Figure 12(b) is further divided to 4×4 low-low band (LL) and high-low band (HL) data by a vertical sub-band analysis.
- (3) The high band data of Figure 12(b) is further divided to 4×4 low-high band (LH) and high-high band (HH) data by a vertical sub-band analysis.

Then, the sub-band analysis derives four sub-bands, LL, LH, HL and HH as shown in Figure 12(c).

Although the above example of sub-band analysis is performed directly on 8×8 data, different type of sub-band analysis can be taken. Figure 13 illustrates another example of the sub-band analysis operation in which the sub-band analysis starts from a frame data with horizontal size x and vertical size y as follows.

- (1) The Green data in a frame of Figure 13(a) is first divided to LL, LH, HL and HH sub-band data of Figure 13(b). The size of each sub-band data is $(x/2) \times (y/2)$.
- (2) Each sub-band data is divided to 4×4 sub blocks.
- (3) Four 4×4 sub blocks, each from LL, LH, HL and HH sub-band data which are located at the same position in the frame are gathered to form 8×8 data shown in Figure 13(c).

At block 1004, 4×4 Red and Blue residual data and 4×4 low-low band data of Green residual are color transformed to generate 4×4 YCoCg residual data. On the other hand, other three 4×4 sub-band data are 4×4 spatial transformed at block 1009, quantized at block 1010 and encoded into bitstream at block 1007.

Figure 11 is a block diagram of a video decoder for open-loop residual coding technique. The blocks from 1104 to 1106 and 1108 are identical to the blocks from 904 to 906 and 908 in Figure 9.

At block 1101 in Figure 11, the bitstream is decoded to derive three 4×4 G quantized coefficient data and 4×4 YCoCg quantized coefficient data. The 4×4 G quantized coefficient data are de-quantized at

block 1102 and are 8×8 (or 4×4) inverse transformed to generate three 4×4 G residuals (sub-bands LH, HL and HH).

At block 1107, 8×8 Green residual data is derived from LL, LH, HL and HH sub-band data by a sub-band synthesis operation. The operation is an inverse process of the sub-band analysis process. For example, LL and HL band of Figure 12(c) is synthesized to generate low band data of Figure 12(b) by a vertical sub-band synthesis. The high band data of Figure 12(b) is also derived by a vertical sub-band synthesis. Then 8×8 Green residual data of Figure 12(a) is derived by a horizontal sub-band analysis.

In this example of open-loop technique, since 48 samples are transformed at block 1009 instead of 64 samples at 713, the number of samples to be processed in the spatial transform can be reduced. Also, the local decoder which includes the blocks from 709 to 712 is not necessary. Then the processing power and the hard ware complexity can be decreased as well.

Figure 14 and Figure 15 correspond to another example of the open-loop residual coding technique. The difference from the example shown in Figure 10 and Figure 11 is that the sub-band analysis is replaced by a simple selector (1408) and the sub-band synthesis is replaced by a simple integrator (1507).

At block 1408 in Figure 14, 8×8 Green residual data is divided to four 4×4 Green residual data blocks. One 4×4 Green residual

data is selected from these four data and transmitted to block 1404. The remaining 4×4 Green residual data are transmitted to block 1409. Figure 16 illustrates an example of the division performed at block 1408. In Figure 16(a), the pixel position x has a coordinate of $(2m, 2n)$ supposed that the position is expressed by integer from 0 to 7, where m and n are integers. Similarly, the positions y , z and w are $(2m+1, 2n)$, $(2m, 2n+1)$ and $(2m+1, 2n+1)$ respectively. Then the 8×8 data is divided to four 4×4 data blocks as shown in Figure 16(b).

In Figure 15, all blocks except block 1507 performs as same as in Figure 11. At block 1507, the reconstructed 4×4 Green residual data from block 1506 and three reconstructed 4×4 Green residual data from block 1503 are integrated by the inverse process of the selector at block 1408 in Figure 14. The four 4×4 data in Figure 16(b) is integrated to form 8×8 data in Figure 16(a).

Although this example can be implemented whichever position is selected to be sent to block 1404, the selection may be adapted so that the efficiency of the color transform at block 1404 is optimized.

As shown from Figure 3 and Figure 16, the pixel position of Blue data corresponds to $(2m+1, 2n)$ and the correlation is higher for Green data at $(2m+1, 2n)$ than for other Green data. Similarly, the correlation of Red data is higher for Green data at $(2m, 2n+1)$ than for other data. Therefore, the adaptive selection depending on the intensity of Red, Blue and Green data may optimize the transform efficiency at block

1404.

In this example, since simple selector and integrator are used instead of sub-band analysis and synthesis processes, the processing power and the hard ware complexity can be decreased. However, low pass filter may be performed before sub-sampling to reduce the aliasing effect.

Though RGB data is converted to YCoCg data in the above explanations, other color format such as YCbCr or YUV can be used as well in the present invention.

There are several considerations that should be kept in mind when designing a codec for use with the present invention. For example, because the down-sampling process at block 602 and the up-sampling process at block 614 in Figure 6 are a normative part of a codec, symbols encoded into the bitstream are required at the sequence/picture level so that the correct up-sampling and down-sampling are selected.

Another consideration would be that for coefficient coding, there are four total components that require coding: Y, Co, Cg, and the 2nd-level G. Separate Quantization Parameter (QP) values should be defined for each of the four components. In particular, the QPs for Y and for the 2nd-level G could be different. Coded block patterns (cbp) parameters should similarly be defined for each of the four components. Yet another consideration would be that for G intra prediction, 8×8 prediction modes are preferred; while for R/B, 4×4 intra modes could be used for the 4:2:0 format.

The present invention provides a Residual Color Transform (RCT) coding tool for raw RGB data in which compression is performed directly on raw RGB data without first performing a color transform first.

Figure 17 depicts a high-level block diagram of a video coding system according to the present invention. In particular, RGB sensors perform RGB capture in a well-known manner at 1701. The encoding process at 1702 operates directly on the raw RGB data using the RCT encoding tool. The sampling positions of the RGB data are different within each pixel and the positions can change from picture to picture. Consequently, in one exemplary embodiment of the present invention the RGB sampling positions are signaled in the bitstream at sequence and/or each picture and are then used for motion-vector interpolation and final display rendering (i.e., interpolation of missing RGB data). For example, a zero-motion motion compensation for R/B might actually correspond to a non-zero motion in G.

The encoded raw RGB data is then transmitted and/or stored, as depicted by channel/storage 1703. The decoding process operates directly on the RGB data at 1704. At 1705, interpolation is performed for generating missing RGB color components. The resulting data is RGB displayed at 1706.

Interpolation for the RGB color components (functional block 1705) is deferred in the present invention until the bitstreams have been decoded. Additionally, it should be noted that the RGB color component

interpolation (functional block 1705) could be part of a post-processing for video decoding at 1704 or part of a preprocessing for RGB display at 1706.

Figure 18 shows a flow diagram 1800 of a closed-loop residual encoding technique according to the present invention using RCT for raw RGB data. Flow diagram 1800 corresponds to the second part of processes that occur in block 1702 in Figure 17, which also includes as its first part of processes Intra/Inter Prediction, which is similar to block 202 in Figure 2 except that the Prediction is done in the present invention based on raw RGB data, not on grid-pattern RGB data. Thus, the process depicted in Figure 18 corresponds only to the residual encoding, including transforms (spatial and RCT), quantization, and entropy encoding modules. Prediction and motion compensation are done in the raw RGB domain and are not depicted in Figure 18. Block 1801 in Figure 18 represents two interleaved 4×4 blocks of Green (G) prediction residuals.

Figure 19 depicts two exemplary interleaved 4×4 G residuals 1901 and 1902, an exemplary 4×4 B residual 1903 and an exemplary 4×4 R residual 1904.

At block 1802, the two interleaved 4×4 blocks of G prediction residuals is sub-sampled to produce a 4×4 block of G residual at block 1803. The sub-sampling at block 1802 could be, for example, an averaging operation. Alternatively, any low-pass or decimation filtering technique could be used for block 1802. In case of the averaging operation, the sub-sampled 4×4 G residual may be calculated by :

$$G_s(i, j) = (G_o(i, j) + G_e(i, j)) / 2.$$

In the above equations, (i, j) represents horizontal coordinate and vertical coordinate of data position in each 4×4 block. G_e and G_o are 4×4 blocks respectively shown by 1901 and 1902 in Figure 19. G_s is a sub-sampled 4×4 G residual. In case of the low-pass or decimation filtering technique, the sub-sampled 4×4 G residual may be calculated by :

$$G_s(i, j) = (a * G_o(i, j) + b * G_e(i, j) + b * G_e(i-1, j) + b * G_e(i, j-1) + b * G_e(i-1, j-1)) / (a + 4 * b),$$

where "a" and "b" are filtering coefficients and "(a+4*b)" is a normalization factor. The sample of 1902 in Figure 19 may be expanded as shown in Figure 20 to get the data out side of the 4×4 block. The samples with broken line are the expanded data.

At block 1804, the 4×4 G residual together with the 4×4 Blue (B) residual (block 1805) and the 4×4 block of Red (R) residual (block 1806) are converted from RGB-based data to YCoCg-based data (in three 4×4 blocks). At block 1807, the YCoCg data goes through a 4×4 transformation and is quantized at block 1808 to produce YCoCg coefficients at block 1809. The YCoCg coefficients are encoded into bitstreams by an entropy encoder at block 1820.

The YCoCg coefficients generated at block 1808 are de-quantized at block 1810 and inverse 4×4 transformed at block 1811 to reconstruct the YCoCg-based data before being converted to RGB-based data at block 1812 to form a reconstructed 4×4 G residual at block 1813.

The 4×4 G residual is up-sampled at block 1814 to form two interleaved 4×4 G residual predictions at block 1815. The up-sampling process at block 1814 could be, for example, a duplicative operation. Alternatively, any interpolation filtering technique could be used for block 1814. The differences between the two interleaved 4×4 G residuals at block 1801 and the two interleaved 4×4 G residual predictions at block 1815 are used to form the two 2nd-level 4×4 G residuals at block 1816. The two 2nd-level 4×4 G residuals go through two 4×4 transformations at 1817 and a quantization process at block 1818 to form Green (G) coefficients at block 1819. The G coefficients are encoded into bitstreams by the entropy encoder at block 1820.

Figure 21 is a block diagram of a video encoder which includes a residual encoder corresponding to the residual encoding technique of Figure 18.

At block 2101, RGB sensors perform RGB capture to create raw RGB data. At block 2102, Intra/Inter prediction is performed to generate residual raw RGB data. This functional block includes inter prediction portion and intra prediction portion. The inter prediction portion contains frame memories for motion compensation prediction and motion estimation portion to generate motion vectors.

The 4×4 Green residual data, which is generated by sub-sampling from two 4×4 Green residual data at block 2107, and the 4×4 Red and Blue residual data which is from block 2102 are input to block

2103.

Blocks from 2103 to 2105 are identical to blocks from 704 to 706 in Figure 7. The 4×4 YCoCg quantized coefficient data derived in this part is encoded into bitstream at block 2106.

Blocks from 2108 to 2110 are identical to blocks from 709 to 711 in Figure 7. The 4×4 G reconstructed residual data derived in this part is up-sampled to generate two 4×4 G interpolated residual data at 2111. This process of reconstruction is called local decoding at an encoder.

The difference data (second-level $2 \times 4 \times 4$ G residual) between two 4×4 G residual at block 2102 and two 4×4 G interpolated residual data at 2111 is 4×4 transformed at block 2112 and quantized at block 2113 to generate 4×4 G quantized coefficient data. The 4×4 G quantized coefficient data is encoded into bitstream at block 2106.

Figure 22 shows a flow diagram 2200 of a closed-loop residual decoding technique using RCT for raw RGB data according to the present invention, which is a decoder corresponding to the residual encoding technique of Figure 18. The process depicted in Figure 22 corresponds only to the residual decoding, including inverse transforms (spatial and RCT), de-quantization, and entropy decoding modules. Prediction and motion compensation are done in the raw RGB domain and are not depicted in Figure 22.

At block 2201, a bitstream is entropy decoded by an entropy decoder to form 2nd-level G coefficients at 2202 and YCoCg coefficients at

2207. The 2nd-level G coefficients are de-quantized at 2203 and two 4×4 inverse transforms are performed at 2204 to form two 2nd-level 4×4 G residuals at 2205.

The YCoCg coefficients at 2207 are de-quantized at 2208 and 4×4 inverse transformed at 2209 to form YCoCg-based data. At 2210 the YCoCg-based data are converted to RGB-based data including a reconstructed 4×4 B residual at 2211, a reconstructed 4×4 R residual at 2212, and a reconstructed 4×4 G residual at 2213. The reconstructed 4×4 G residual is up-sampled at 2214 to form two interleaved 4×4 G residual predictions at 2215.

At 2206, the two 2nd-level 4×4 G residuals (at 2205) are summed with the two interleaved 4×4 G residual predictions (at 2215) to form two reconstructed 4×4 G residuals at 2216.

Figure 23 is a block diagram of a video decoder which includes a residual decoder corresponding to the residual decoding technique of Figure 22.

The bitstream is decoded at block 2301 to generate two 4×4 G quantized coefficient data and 4×4 YCoCg quantized coefficient data. The 4×4 YCoCg quantized coefficient data is converted to two 4×4 G interpolated residual data and 4×4 Red and Blue reconstructed residual data at blocks from 2304 to 2307 which are identical to blocks from 2108 to 2111 in Figure 21. The two 4×4 G quantized coefficient data is de-

quantized at block 2302, 4×4 inverse transformed at block 2303 and added by the data from block 2307 to generate two 4×4 G residual data.

At block 2308, raw RGB data is reconstructed from two 4×4 Green residual data and 4×4 Red and Blue residual data by performing Intra/Inter prediction.

The encoding of G samples described in connection with Figure 18 and Figure 21 is a closed-loop technique. Figure 24 shows a flow diagram 2400 of an open-loop residual encoding technique using RCT for raw RGB data according to the present invention. Block 2401 in Figure 24 represents two interleaved 4×4 blocks of Green (G) prediction residuals, similar to block 1801 in Figure 18. At 2402, the two interleaved 4×4 blocks of G residuals are Haar transformed to form an averaged 4×4 G residual at 2403 and a differentiated 4×4 G residual at 2410 as follows :

$$Ga(i, j) = (Ge(i, j) + Go(i, j)) / 2 ,$$

$$Gd(i, j) = (Ge(i, j) - Go(i, j)) / 2 .$$

In the above equations, (i, j) represents horizontal coordinate and vertical coordinate of data position in each 4×4 block. Ge and Go are 4×4 blocks respectively shown by 1901 and 1902 in Figure 19. Ga is an averaged 4×4 G residual and Gd is a differentiated 4×4 G residual.

The averaged 4×4 G residual at 2403 is a simple average of the two closest G pixels in the two interleaved 4×4 G residuals. At block 2404, the averaged 4×4 G residual together with the 4×4 Blue (B)

residual (block 2405) and the 4×4 block of Red (R) residual (block 2406) are converted from RGB-based data to YCoCg-based data. At block 2407, the YCoCg data goes through a 4×4 transformation and is quantized at block 2408 to produce YCoCg coefficients at block 2409. The YCoCg coefficients are encoded into a bitstream by an entropy encoder at block 2414.

Returning to block 2402, the difference of the two 4×4 interleaved G residuals is used for form a differentiated 4×4 G residual block at 2410, which goes through the second-level of residual encoding, that is, a 4×4 transform at 2411 and quantization at 2412 that are similar to steps 1817 and 1818 in Figure 18, except that only one 4×4 data block need to be transformed and quantized here. The G coefficients at 2413 are encoded into bitstreams by the entropy encoder at block 2414. Blocks 1810-1816 of Figure 18 are not needed in the open-loop approach of Figure 24 because the reconstructed pixels are not needed.

Figure 25 is a block diagram of a video encoder which includes a residual encoder corresponding to the open-loop residual encoding technique of Figure 24.

Blocks from 2501 to 2505 are identical to blocks from 2101 to 2105 and the resulting 4×4 YCoCg quantized coefficient data derived in this part is encoded into bitstream at block 2506.

At block 2507, two interleaved 4×4 G residual data is transformed by Haar transform to generate an averaged 4×4 G residual

and a differentiated 4×4 G residual. The averaged 4×4 G residual is inputted to block 2503 to be encoded by RCT encoding technique together with 4×4 R and B residuals.

On the other hand, the differentiated 4×4 G residual is 4×4 transformed at block 2508 and quantized at block 2509 to generate 4×4 G quantized coefficient data. The 4×4 G quantized coefficient data is encoded into bitstream at block 2506.

Figure 26 shows a flow diagram 2600 of an open-loop residual decoding technique using RCT for raw RGB data according to the present invention, which is a decoder corresponding to the residual encoding technique of Figure 24 and Figure 25. The process depicted in Figure 26 corresponds only to the residual decoding, including inverse transforms (spatial and RCT), de-quantization, and entropy decoding modules. Prediction and motion compensation are done in the raw RGB domain and are not depicted in Figure 26.

At block 2601, a bitstream is entropy decoded by an entropy decoder to form 2nd-level G coefficients at 2602 and YCoCg coefficients at 2607. The 2nd level G coefficients are de-quantized at 2603 and 4×4 inverse transformed at 2604 to form a 2nd-level 4×4 G residual at 2605.

The YCoCg coefficients at 2607 are de-quantized at 2608 and 4×4 inverse transformed at 2609. At 2610 the YCoCg-based data are converted to RGB-based data including a reconstructed 4×4 B residual at 2611, a reconstructed 4×4 R residual at 2612, and a reconstructed 4×4 G

residual at 2613.

At 2606, the 2nd-level 4×4 G residual (at 2605) are inverse Haar transformed with the reconstructed 4×4 G residual (at 2613) to form two interleaved reconstructed 4×4 G residuals at 2614 as follows:

$$Ge(i, j) = Ga(i, j) + Gd(i, j) ,$$

$$Go(i, j) = Ga(i, j) - Gd(i, j) ,$$

where (i, j) represents horizontal coordinate and vertical coordinate of data position in each 4×4 block. Ge and Go are interleaved reconstructed 4×4 blocks. Ga is an averaged 4×4 G residual at 2606 and Gd is a differentiated 4×4 G residual at 2613.

Figure 27 is a block diagram of a video decoder which includes a residual decoder corresponding to the open-loop residual decoding technique of Figure 24 and Figure 25.

The bitstream is decoded at block 2701 to generate 4×4 G quantized coefficient data and 4×4 YCoCg quantized coefficient data. The 4×4 YCoCg quantized coefficient data is de-quantized at block 2704 and inverse transformed at block 2705. Then at block 2706, the data is converted to averaged 4×4 G residual and 4×4 Red and Blue residual. On the other hand, the 4×4 G quantized coefficient data is de-quantized at block 2702 and inverse transformed at block 2703 to generate differentiated 4×4 G residual data.

At block 2707, the averaged 4×4 G residual and the differentiated 4×4 G residual are inverse Haar transformed to generate

two interleaved 4×4 G residual. At block 2708, raw RGB data is reconstructed from two interleaved 4×4 G residual and 4×4 Red and Blue residual data by performing Intra/Inter prediction.

Figure 28 and Figure 29 correspond to another example of the open-loop residual coding technique using RCT for raw RGB data. The difference from the example shown in Figure 25 and Figure 27 is that the Haar transform is replaced by a simple selector (2807) and the inverse Haar transform is replaced by a simple integrator (2907).

At block 2807 in Figure 28, one 4×4 data is selected from two 4×4 interleaved residual G shown in Figure 19 and transmitted to block 2803. The other 4×4 data is transmitted to block 2808. The remaining blocks perform the same way as in Figure 25.

Although this example can be implemented whichever block is selected to be sent to block 2803, the selection may be adapted so that the efficiency of the color transform at block 2803 is optimized by the same reason explained for block 1408 of Figure 14.

In Figure 29, all blocks except block 2907 perform as same as in Figure 27. At block 2907, the reconstructed 4×4 Green residual data from block 2906 and the reconstructed 4×4 Green residual data from block 2903 are integrated by the inverse process of the selector at block 2807 in Figure 28.

In this example, since simple selector and integrator are used instead of Haar transform/inverse transform processes, the processing

power and the hardware complexity can be decreased.

There are several considerations that should be kept in mind when a codec is designed for use with the present invention. For example, because the sub-sampling process at block 1802 and the up-sampling process at block 1814 in Figure 18 are a normative part of a codec, symbols encoded into the bitstream are required at the sequence/picture level so that the correct up-sampling and sub-sampling are selected.

Another consideration would be that for coefficient coding, there are four total components that require coding: Y, Co, Cg, and the 2nd-level G. Separate Quantization Parameter (QP) values should be defined for each of the four components. In particular, the QPs for Y and for the 2nd-level G could be different. Coded block patterns (cbp) parameters should similarly be defined for each of the four components. Yet another consideration would be that for R/B intra prediction, 4×4 intra prediction modes are preferred; while for G, two 4×4 intra modes could be used. Alternatively, a set of intra prediction modes could be developed.

The G pixels are sampled in a quincunx pattern. Consequently, sub-pixel interpolation for motion prediction for G residuals is different from sub-pixel interpolation for motion prediction for the R or B pixels, which are sampled in a usual grid pattern. Accordingly, there are many possible interpolation methods designed for a quincunx pattern that could be used.

Though RGB data is converted to YCoCg data in the above

explanations, other color format such as YCbCr or YUV can be used as well in the present invention. Also, the other color components than RGB may be applied for the current invention, such as four components of RGB with white, four components of Y(yellow), M(magenta), C(cyan) with black, six components of RGB and YMC, and so on.

The current invention generally be applied for video data with at least three color components such as RGB where the sampling rate of at least one component is greater than other components. For example, 4:2:0 RGB format contains 8×8 Green data, 4×4 Blue data and 4×4 Red data in a block of 8×8 pixels. Then in case of 4:2:0 RGB format, the sampling rate of G is four times higher than that of B and R. In case of raw RGB format, the sampling rate of G is two times higher than that of B and R.

The current invention can be applied for still image coding. In this case, the Intra/Inter Prediction block in Figures 7, 9, 10, 11, 14, 15, 21, 23, 25, 27, 28 and 29 may be replaced by Intra Prediction.

Generally, the Intra/Inter Prediction block in the encoder may be replaced by a converter which converts a first RGB data to a second RB data and a second G data. When the converter performs intra/inter prediction such as 703, the second RB data and the second G data are the residual RB data and the residual G data. When the converter performs low pass pre-filtering, the second RB data and the second G data are the filtered RB data and the filtered G data.

Similarly, the Intra/Inter Prediction block in the decoder such as block 908 may be replaced by a converter which converts a second RB data and a second G data to a first RGB data. When the converter of an encoder is an intra/inter predictor, the converter of a decoder also is an intra/inter predictor. When the converter of an encoder is a pre-filter, the converter of a decoder may be a post filter.

Generally, blocks 708, 1008, 1408, 2107, 2507 and 2807 can be replaced by a converter which converts the second G data to a third G data. When the converter is block 708 or 2107, the third G data is sub-sampled G residual data. When the convert is block 1008, the third G data includes sub-bands (LL, LH, HL and HH) of G data. And when the converter is block 1008, 1408, 2507 or 2807, the third G data includes two separated G data. These converters generate G data with smaller sampling rate than the input. The converted sampling rate is same as the sampling rate of other color components(R and B) which enables color transforming at block 704, block 1004 and so on.

Similarly, blocks 907, 1107, 1507, 2307, 2707 and 2907 can be replaced by a converter which converts the third G data to the second G data. These converters generate G data with same sampling rate to the G data in the first RGB data.

Generally, a set of blocks from 704 to 707 in Figure 7 comprises an encoder for RGB data which includes color transforming from RGB to YCoCg. Other examples of the encoder are a set of blocks

from 1004 to 1007 in Figure 10, a set of blocks from 1404 to 1407 in Figure 14, a set of blocks from 2103 to 2106 in Figure 21, a set of blocks from 2503 to 2506 in Figure 25 and a set of blocks from 2803 to 2806 in Figure 28.

Similarly, a general decoder for RGB data which includes color transforming from YCoCg to RGB can be considered. The examples of the general decoder are as follows.

- blocks 901, 904 to 906 in Figure 9
- blocks 1101, 1104 to 1106 in Figure 11
- blocks 1501, 1504 to 1506 in Figure 15
- blocks 2301, 2304 to 2306 in Figure 23
- blocks 2701, 2704 to 2706 in Figure 27
- blocks 2901, 2904 to 2906 in Figure 29

Also, general encoder and decoder for G data can be considered and the examples are as follows.

[Examples of encoder]

- blocks 709 to 713 and subtracter in Figure 7
- blocks 1009 and 1010 in Figure 10
- blocks 1409 and 1410 in Figure 14
- blocks 2108 to 2113 and subtracter in Figure 21
- blocks 2508 and 2509 in Figure 25
- blocks 2808 and 2809 in Figure 28

[Examples of decoder]

- blocks 902 and 903 and adder in Figure 9

- blocks 1102 and 1103 in Figure 11
- blocks 1502 and 1503 in Figure 15
- blocks 2302 and 2303 and adder in Figure 23
- blocks 2702 and 2703 in Figure 27
- blocks 2902 and 2903 in Figure 29

Although the foregoing invention has been described in some detail for purposes of clarity of understanding, it will be apparent that certain changes and modifications may be practiced that are within the scope of the appended claims. Accordingly, the present embodiments are to be considered as illustrative and not restrictive, and the invention is not to be limited to the details given herein, but may be modified within the scope and equivalents of the appended claims.

CLAIMS

1. A video encoder for encoding a first color data with at least three color components of a first color system, comprising:

a first converter converting the first color data to a second color data with at least two color components and a third color data with at least one color component;

a second converter converting the third color data to a fourth color data;

a first encoder encoding the second color data and the fourth color data; and

a second encoder encoding the third color data,

wherein the first encoder includes a transformer transforming the second color data and the fourth color data to a color data of a second color system.

2. The video encoder according to claim 1, wherein:

the second encoder includes a local decoder for the first encoder; and

the second encoder includes a subtractor calculating the difference between the color data converted from the decoded fourth color

data and the third color data.

3. A video encoder for encoding a first color data with at least three color components of a first color system, comprising:

a first converter converting the first color data to a second color data with at least two color components and a third color data with at least one color component;

a second converter converting the third color data to a fourth color data and a fifth color data;

a first encoder encoding the second color data and the fourth color data; and

a second encoder encoding the fifth color data,

wherein the first encoder includes a transformer transforming the second color data and the fourth color data to a color data of a second color system.

4. The video encoder according to claim 1 or claim 3, wherein:

the first color data includes red, green and blue color components;

the second color data includes red and blue color components;

the fourth color data includes green color component;

the number of samples of the green color component in the

first color data is larger than that of the red color component and the blue color component in the first color data respectively; and

the number of samples of the green color component in the fourth color data is as same as that of the red color component and the blue color component in the second color data respectively.

5. The video encoder according to claim 3, wherein:

the first converter performs sub-band analysis;

the fourth color data contains a low-low band generated by the sub-band analysis; and

the fifth color data contains a low-high band, a high-low band and a high-high band generated by the sub-band analysis.

6. The video encoder according to claim 3, wherein:

the first converter divides the third color data to the fourth color data and the fifth color data.

7. The video encoder according to claim 3, wherein:

the first converter performs orthogonal transform;

the fourth color data contains a DC data generated by the orthogonal transform; and

the fifth color data contains an AC data generated by the orthogonal transform.

8. A video decoder for decoding an encoded color data,
comprising:

a first decoder decoding a second color data with at least two color components of a first color system and a fourth color data with at least one color component of the first color system;

a second decoder decoding a third color data with at least one color component of the first color system; and

a converter converting the second color data and the third color data to a first color data with at least three color components of a first color system,

wherein the first decoder includes a transformer transforming a color data of a second color system to the second color data and the fourth color data.

9. The video decoder according to claim 8, wherein:

the second decoder includes an adder adding the color data converted from the fourth color data and a difference data decoded in the second decoder.

10. A video decoder for decoding an encoded color data,
comprising:

a first decoder decoding a second color data with at least two color components of a first color system and a fourth color data with at

least one color component of the first color system;

a second decoder decoding a fifth color data with at least one color component of the first color system;

a first converter converting the fourth color data and the fifth color data to a third color data; and

a second converter converting the second color data and the third color data to a first color data with at least three color components of a first color system,

wherein the first decoder includes a transformer transforming a color data of a second color system to the second color data and the fourth color data.

11. The video decoder according to claim 8 or claim 10, wherein:

the first color data includes red, green and blue color components;

the second color data includes red and blue color components;

the fourth color data includes green color component;

the number of samples of the green color component in the first color data is larger than that of the red color component and the blue color component in the first color data respectively; and

the number of samples of the green color component in the

fourth color data is as same as that of the red color component and the blue color component in the second color data respectively.

12. The video decoder according to claim 10, wherein:

the converter performs sub-band synthesis;

the fourth color data contains a low-low band for the sub-band synthesis; and

the fifth color data contains a low-high band, a high-low band and a high-high band for the sub-band synthesis.

13. The video decoder according to claim 10, wherein:

the converter integrates the fourth color data and fifth color data to generate the third color data.

14. The video decoder according to claim 10, wherein:

the converter performs inverse orthogonal transform;

the fourth color data contains a DC data for the inverse orthogonal transform; and

the fifth color data contains an AC data for the inverse orthogonal transform.

15. A Residual Color Transform (RCT) encoding method for 4:2:0 Red-Green-Blue (RGB) data, comprising:

interpolating RGB data to generate at least one missing

Green color component from 4:2:0 RGB data; and

directly encoding the 4:2:0 RGB-based data.

16. A video encoding system directly encoding 4:2:0 RGB data using an RCT encoding tool by interpolating RGB data to generate at least one missing Green color component from 4:2:0 RGB data and directly encoding the 4:2:0 RGB-based data.

17. The system according to claim 16, wherein the system directly encodes the 4:2:0 RGB data without data loss.

18. The video encoding system according to claim 16, further comprising:

a sub-sampler sub-sampling an 8×8 Green residual to form a single 4×4 Green residual;

a converter converting the 4×4 Green residual, a corresponding 4×4 Red residual and a corresponding 4×4 Blue residual to YCoCg-based data;

a transformer 4×4 transforming the YCoCg-based data;

a quantizer quantizing the 4×4 transformed YCoCg-based data to form YCoCg coefficients; and

an entropy encoder encoding the YCoCg coefficients into a bitstream.

19. The system according to claim 18, further comprising:

a de-quantizer de-quantizing the YCoCg coefficients;

an inverse transformer inverse 4×4 transforming the de-quantized YCoCg coefficients to reconstruct the YCoCb-based data;

a converter converting the YCoCg-based data to RGB-based data;

a reconstructor reconstructing 4×4 G residual data;

an up-sampler 2×2 up-sampling the 4×4 G residual data to form an 8×8 G residual prediction;

a differencer forming a 2nd-level 8×8 Green residuals based on a difference between the 8×8 G residual prediction and the 8×8 Green residuals;

a second transformer transforming the 2nd-level 8×8 Green residuals by one of a 4×4 transformation and an 8×8 transformation;

and

a second quantizer quantizing the transformed 2nd-level 8×8 Green residual to form Green coefficients,

wherein the entropy encoder further encodes the Green coefficients into the bitstream.

20. A system decoding directly encoded 4:2:0 RGB data and interpolating the decoded 4:2:0 RGB data for generating at least one of a missing Blue color component and a missing Red color component.

21. The system according to claim 20, wherein the system further displays the directly decoded and interpolated 4:2:0 RGB data.

22. The system according to claim 20, further comprising a decoder that includes:

an entropy decoder entropy decoding a bitstream of directly encoded 4:2:0 RGB data to form YCoCg coefficients;

a first de-quantizer de-quantizing the YCoCg coefficients;

an inverse-transformer inverse-transforming the de-quantized YCoCg coefficients to form an 8×8 Green residual prediction and 4×4 Red and Blue residuals; and

a residual former forming an 8×8 Green residual from the 8×8 Green residual prediction.

23. The system according to claim 22, wherein the entropy decoder further entropy decodes the bitstream to form Green coefficients, and

wherein the residual former includes:

a second de-quantizer de-quantizing the Green coefficients to form an 8×8 Green residual;

a second inverse-transformer inverse-transforming the de-quantized Green coefficients to form a 2nd-level 8×8 Green residual; and

a combiner combining the 2nd-level 8×8 Green residual with

the 8×8 Green residual prediction to form the 8×8 Green residual.

24. A Residual Color Transform (RCT) encoding method for encoding Red-Green-Blue (RGB) data, comprising:

encoding raw RGB data using an RCT encoding tool.

25. A video encoding system directly encoding raw RGB data using a RCT encoding tool.

26. The system according to claim 25, wherein the system encodes the raw RGB data directly without first performing a color transform.

27. The system according to claim 25, wherein the video encoding system performs a closed-loop encoding technique.

28. The system according to claim 27, further comprising:

a sub-sampler sub-sampling two 4×4 Green residuals to form a single 4×4 Green residual;

a converter converting the single 4×4 Green residual, a corresponding 4×4 Red residual and a corresponding 4×4 Blue residual to YCoCg-based data;

a 4×4 transformer 4×4 transforming the YCoCg-based data;

a quantizer quantizing the 4×4 transformed YCoCg-based data to form quantized YCoCg coefficients; and

an entropy encoder encoding the quantized YCoCg coefficients into a bitstream.

29. The system according to claim 28, further comprising:

a de-quantizer de-quantizing the quantized YCoCg coefficients;

an inverse 4×4 transformer inverse 4×4 transforming the de-quantized YCoCg coefficients to reconstruct the YCoCb-based data;

a YCoCg-to-RGB converter converting the YCoCg-based data to RGB-based data including a reconstructed 4×4 G residual;

an up-sampler up-sampling the reconstructed 4×4 G residual to form two 4×4 G residual predictions;

a differencer forming two 2nd-level 4×4 Green residuals based on a difference between the two 4×4 G residual predictions and the two 4×4 Green residuals;

a second 4×4 transformer 4×4 transforming the two 2nd-level 4×4 Green residuals; and

a second quantizer quantizing the two transformed 2nd-level 4×4 Green residuals to form quantized Green coefficients, and

wherein the entropy encoder further encodes the Green coefficients into the bitstream.

30. The system according to claim 25, wherein the video encoding system performs an open-loop encoding technique.

31. The system according to claim 30, further comprising:

a Haar transformer Haar transforming two interleaved 4×4 blocks of Green prediction residuals to form an averaged 4×4 G residual and a differentiated 4×4 G residual;

a converter converting the averaged 4×4 Green residual, a corresponding 4×4 Red residual and a corresponding 4×4 Blue residual to YCoCg-based data;

a 4×4 transformer 4×4 transforming the YCoCg-based data;

a quantizer quantizing the 4×4 transformed YCoCg-based data to form quantized YCoCg coefficients; and

an entropy encoder encoding the quantized YCoCg coefficients into a bitstream.

32. The system according to claim 31, further comprising:

a second 4×4 transformer 4×4 transforming a differentiated 4×4 G residual; and

a second quantizer quantizing the transformed differentiated 4×4 G residual to form quantized Green coefficients,

wherein the entropy encoder further encodes the quantized Green coefficients into the bitstream.

33. A decoder, comprising:

an entropy decoder entropy decoding a bitstream to form YCoCg coefficients, the bitstream having raw RGB data video encoded using an RCT encoding tool;

a first de-quantizer de-quantizing the YCoCg coefficients to form de-quantized YCoCg coefficients;

an inverse-transformer inverse-transforming the de-quantized YCoCg coefficients to form YCoCg-based data;

a YCoCg-to-RGB converter converting the YCoCg-based data to form a reconstructed 4×4 Green residual, a corresponding reconstructed 4×4 Red residual and a corresponding reconstructed 4×4 Blue residual; and

an up-sampler up-sampling the reconstructed 4×4 Green residual to form two reconstructed 4×4 Green residual predictions.

34. The decoder according to claim 33, wherein the entropy decoder entropy decodes the bitstream to form 2nd-level Green coefficients, and

wherein the decoder further includes:

a second de-quantizer de-quantizing the 2nd-level Green coefficients to form de-quantized 2nd-level Green coefficients;

a second inverse transformer inverse-transforming the de-quantized 2nd-level Green coefficients to form two 2nd-level 4×4 Green residuals, and

a residual former combining the two 2nd-level 4×4 Green residuals with the two reconstructed 4×4 Green residual predictions to form the two reconstructed 4×4 Green residuals.

35. A decoder, comprising:

an entropy decoder entropy decoding a bitstream to form YCoCg coefficients, the bitstream having raw RGB data video encoded using an RCT encoding tool;

a first de-quantizer de-quantizing the YCoCg coefficients to form de-quantized YCoCg coefficients;

a first inverse transformer inverse-transforming the de-quantized YCoCg coefficients to form YCoCg-based data; and

a YCoCg-to-RGB converter converting the YCoCg-based data to form a reconstructed 4×4 Green residual, a corresponding reconstructed 4×4 Red residual and a corresponding reconstructed 4×4 Blue residual.

36. The decoder according to claim 35, wherein the entropy decoder entropy decodes the bitstream to form 2nd-level Green coefficients, and

wherein the decoder further comprises:

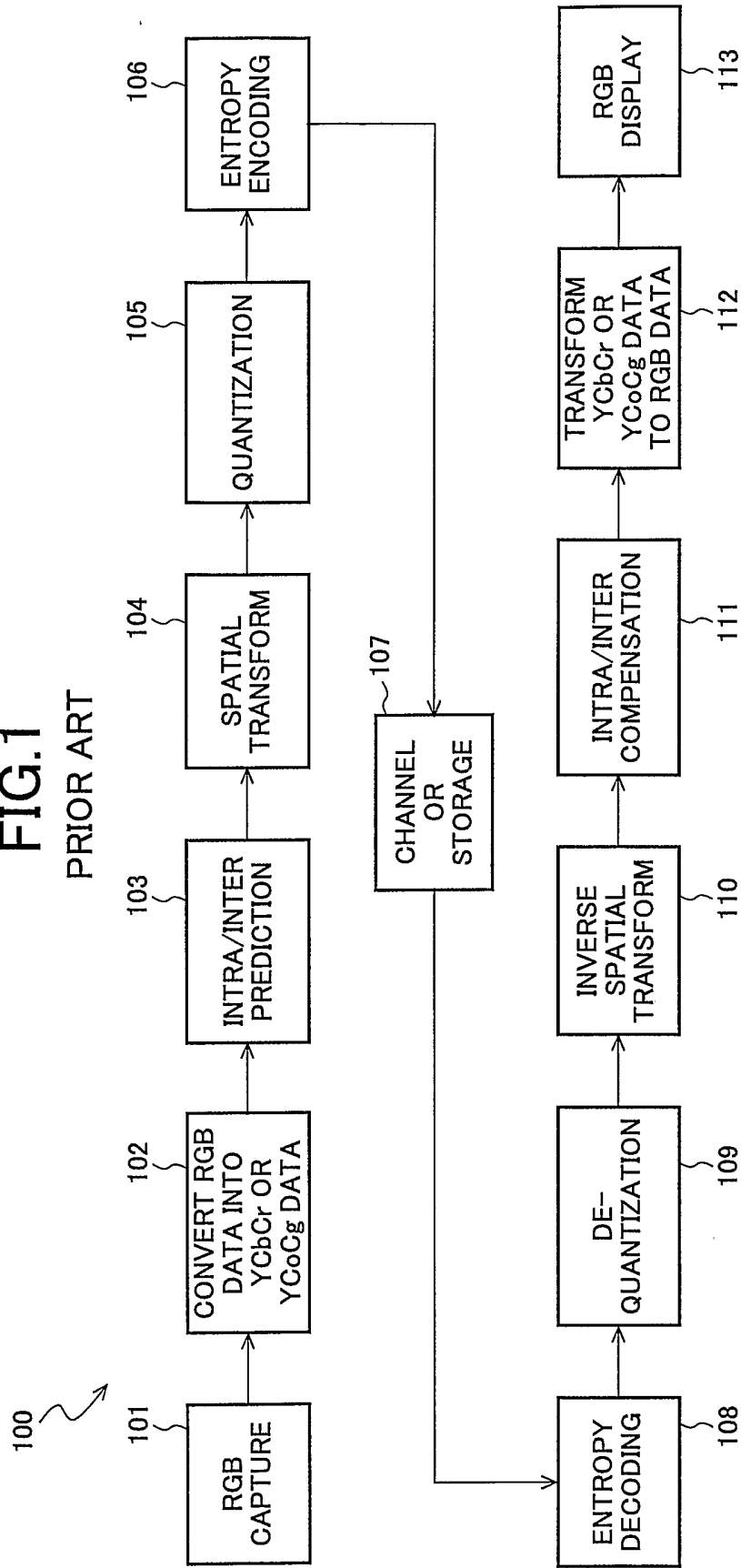
a second de-quantizer de-quantizing the 2nd-level Green coefficients to form de-quantized 2nd-level Green coefficients; and

a second inverse-transformer inverse-transforming the de-quantized 2nd-level Green coefficients to form a 2nd-level 4×4 Green residual, and

wherein an inverse Haar transformer inverse Haar transforms the 2nd-level 4×4 Green residual and the reconstructed 4×4 Green residual to form the two reconstructed 4×4 Green residuals.

FIG.1

PRIOR ART



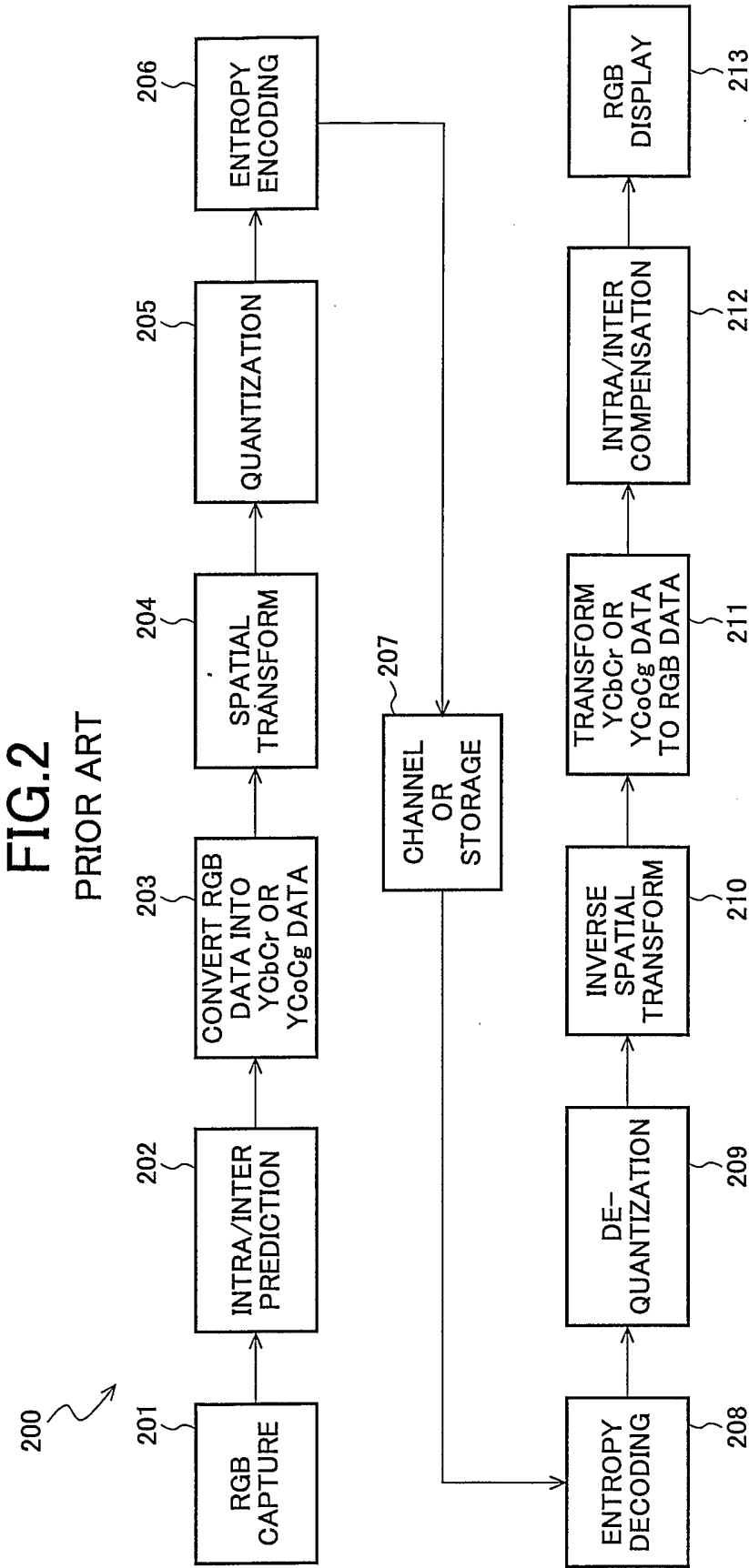
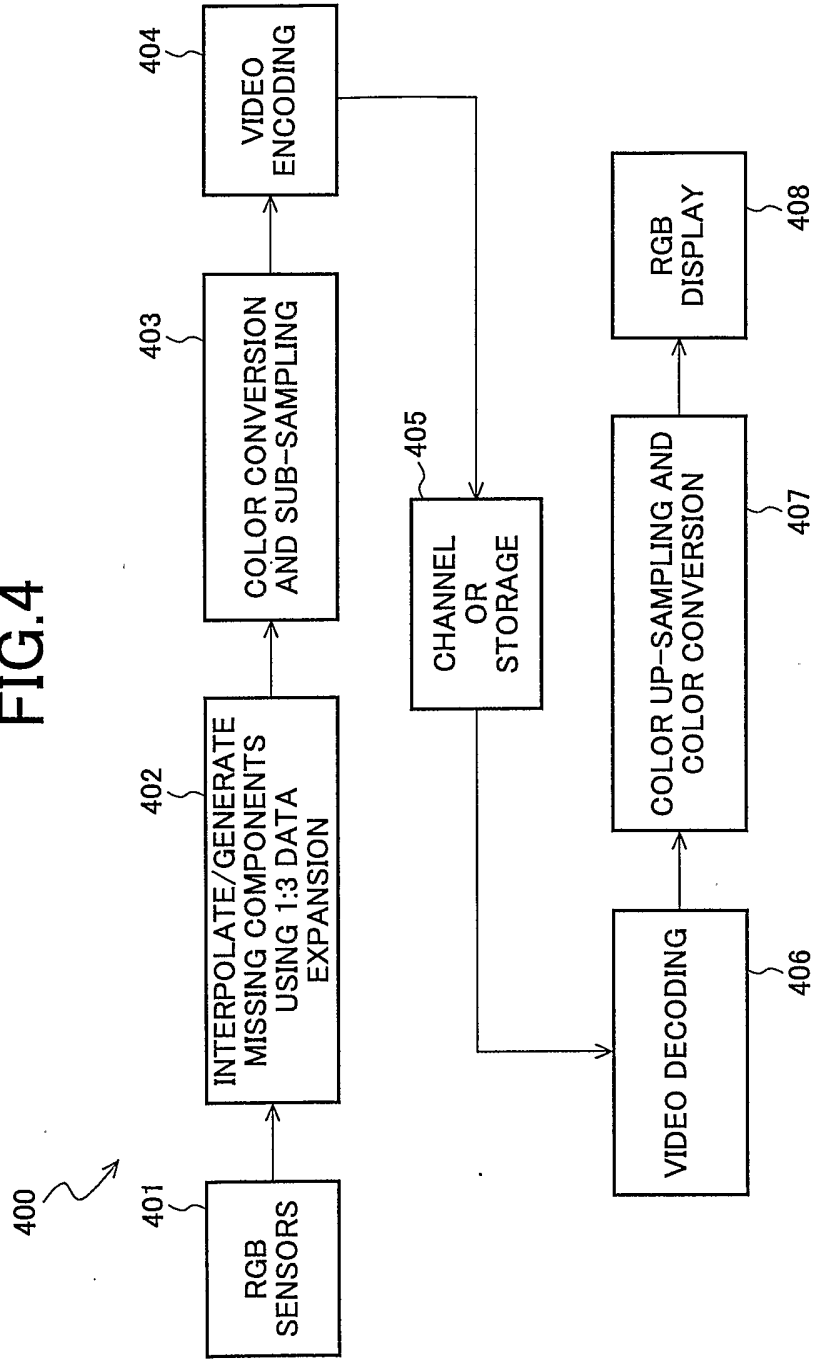


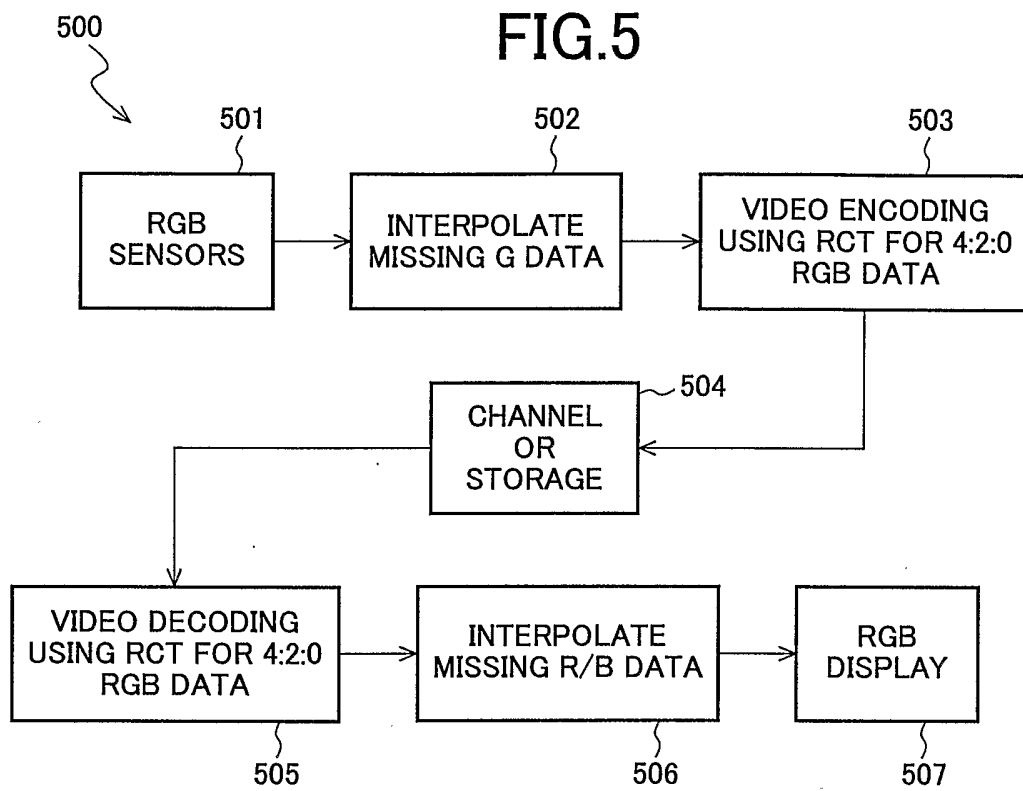
FIG. 3
PRIOR ART

300
↘

G1	B1	G2	B2	G3	B3	G4	B4
R1	G17	R2	G18	R3	G19	R4	G20
G5	B5	G6	B6	G7	B7	G8	B8
R5	G21	R6	G22	R7	G23	R8	G24
G9	B9	G10	B10	G11	B11	G12	B12
R9	G25	R10	G26	R11	G27	R12	G28
G13	B13	G14	B14	G15	B15	G16	B16
R13	G29	R14	G30	R15	G31	R16	G32

FIG.4





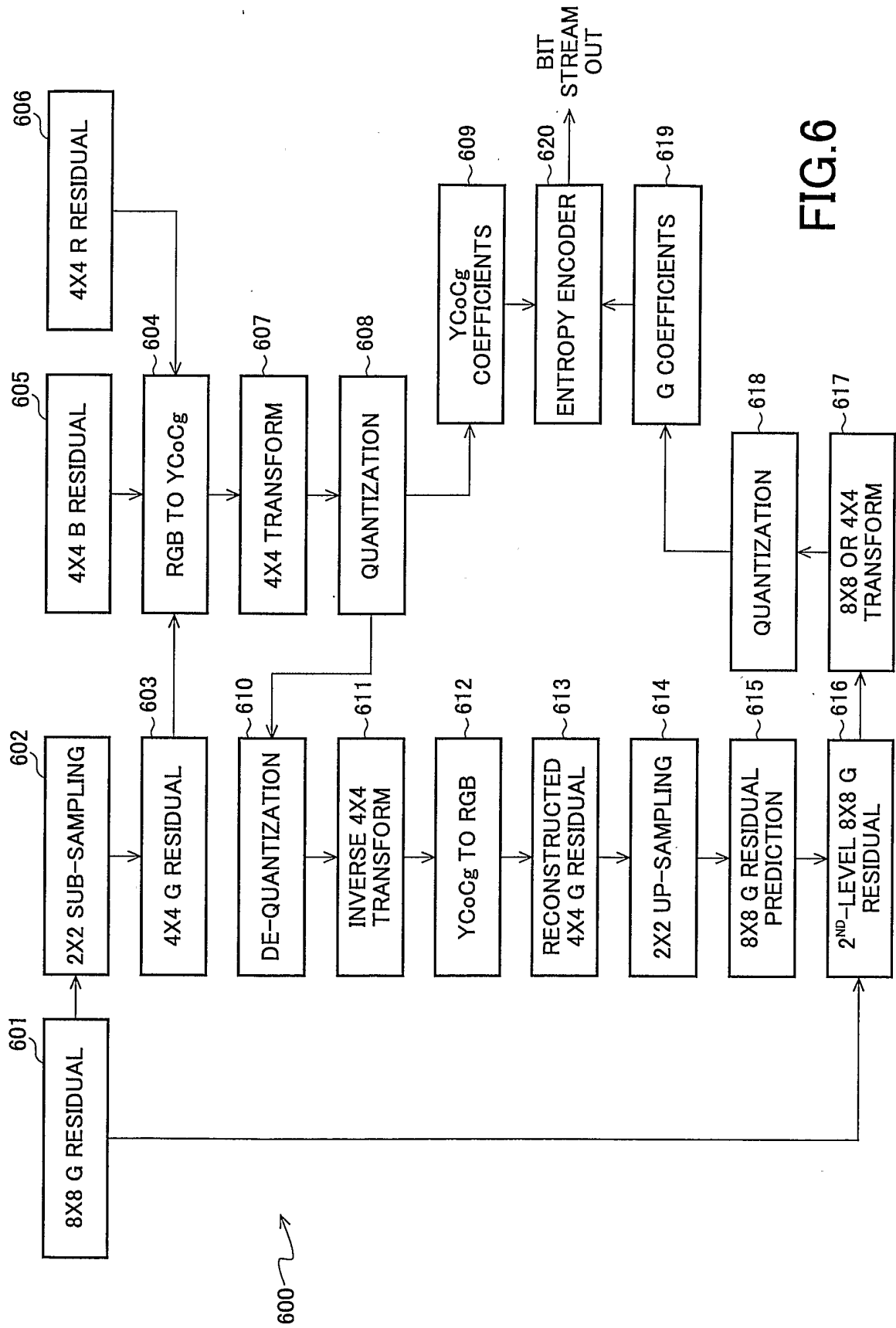


FIG.6

FIG. 7

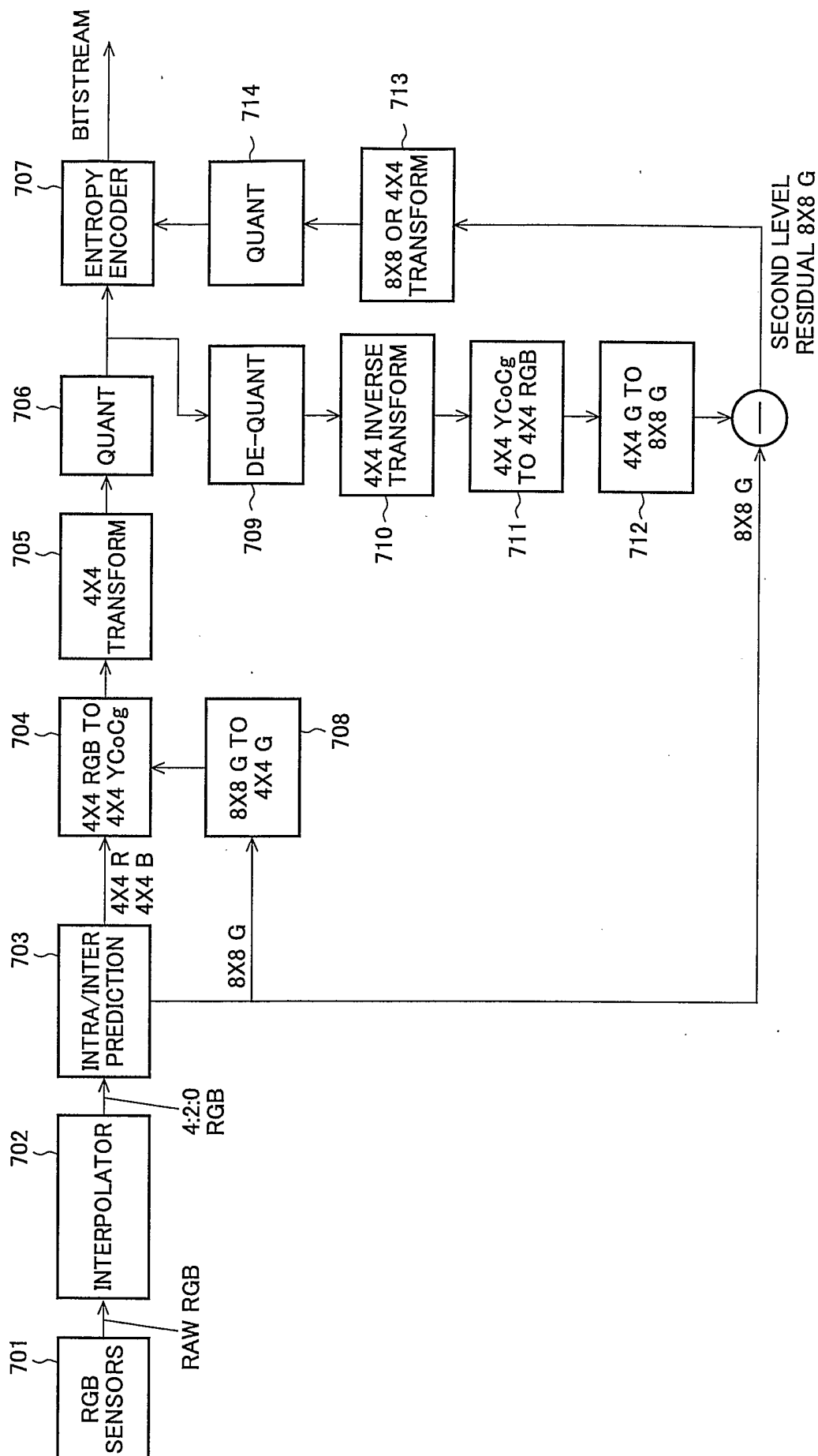


FIG. 8

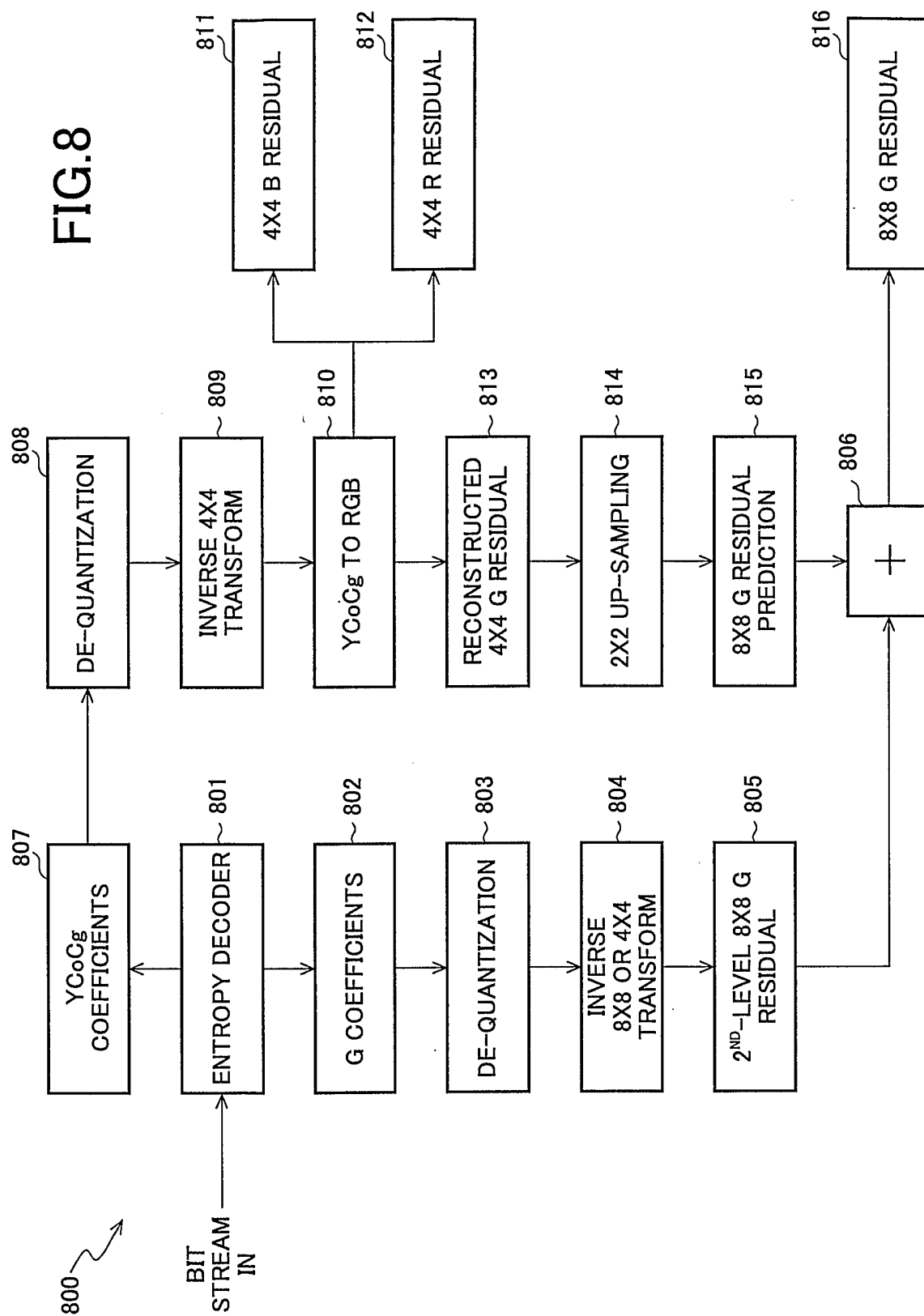


FIG.9

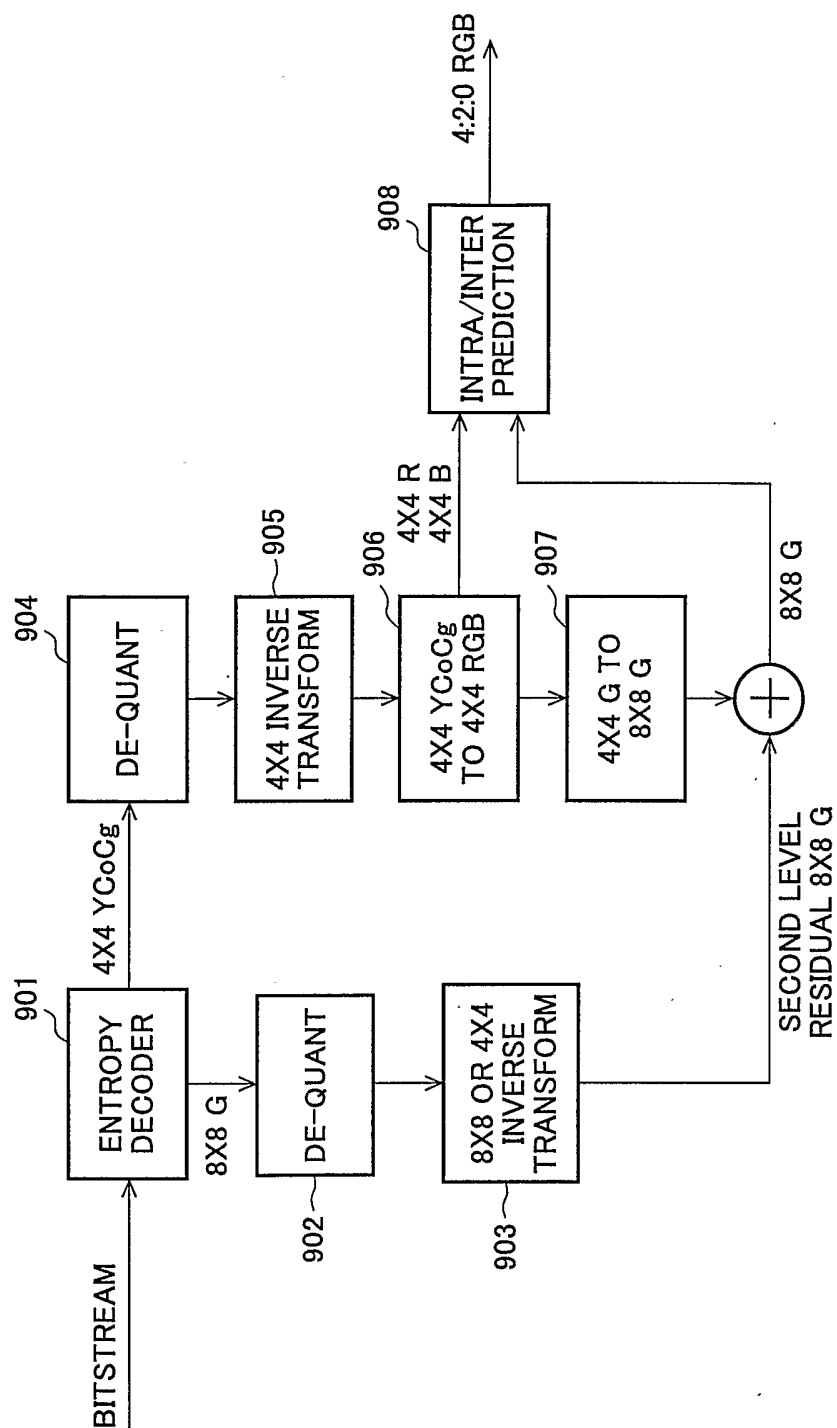


FIG.10

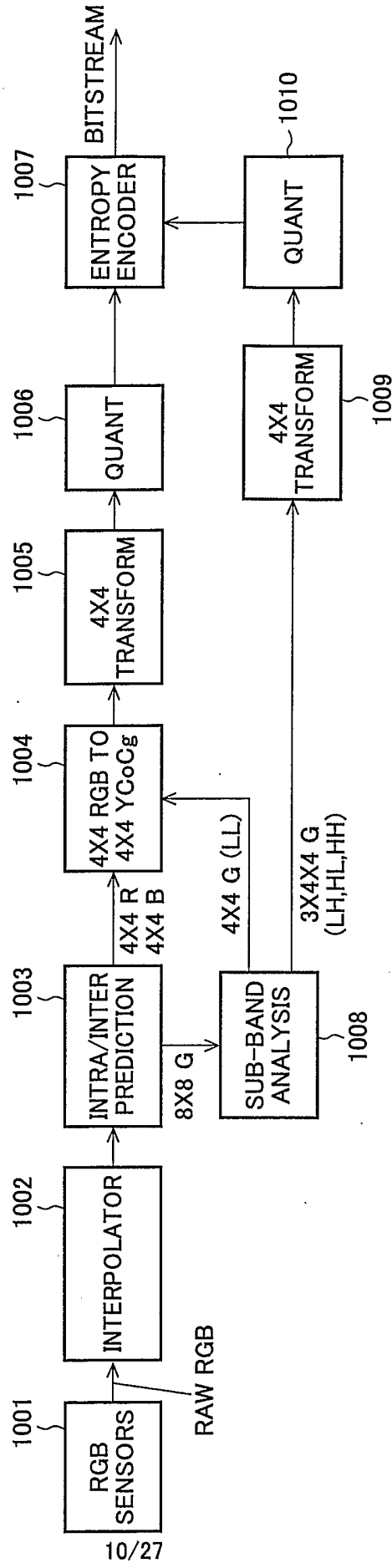


FIG.11

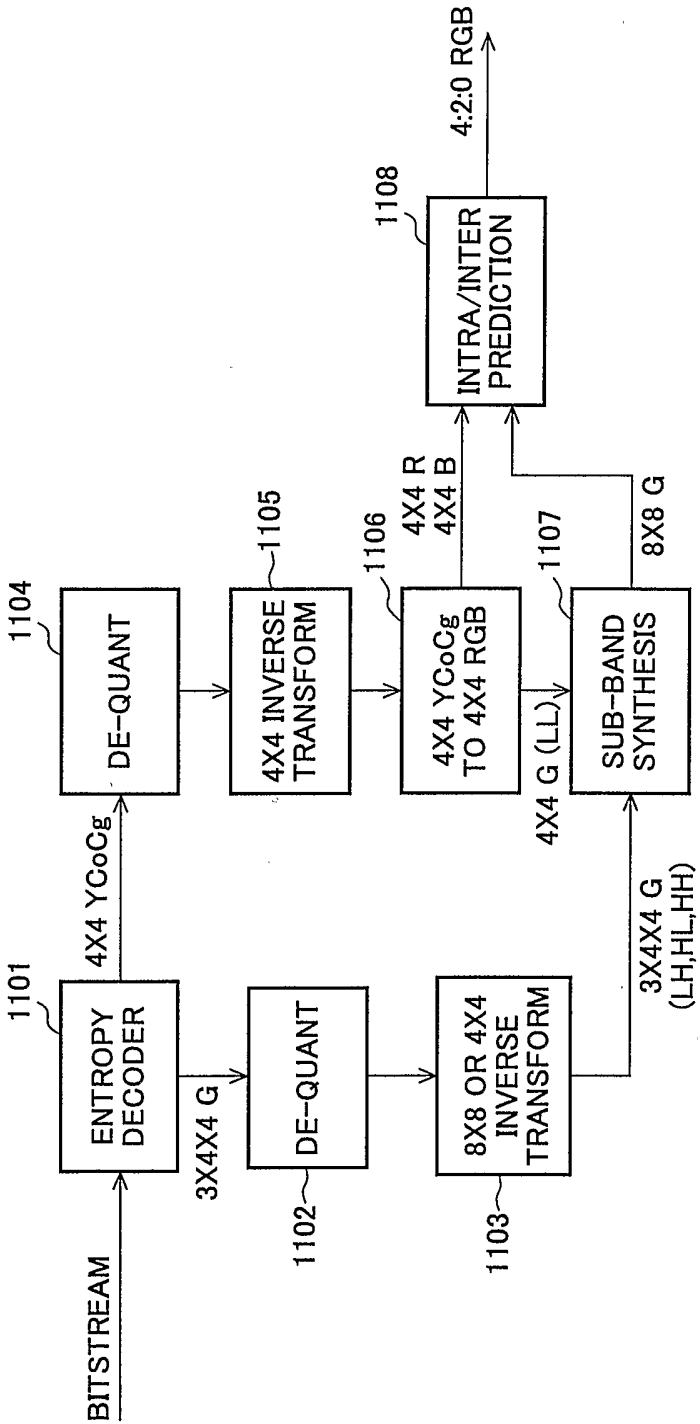


FIG.12

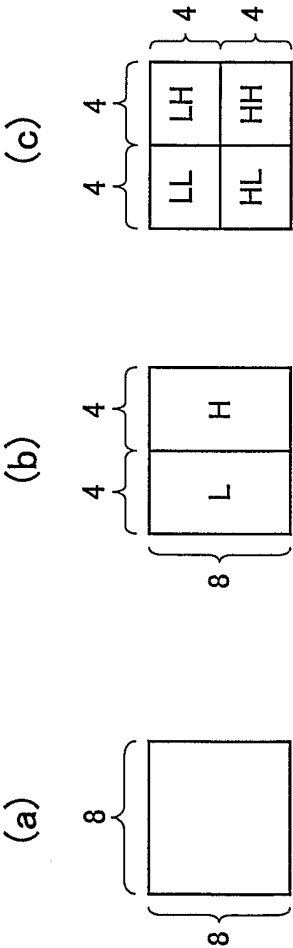


FIG.13

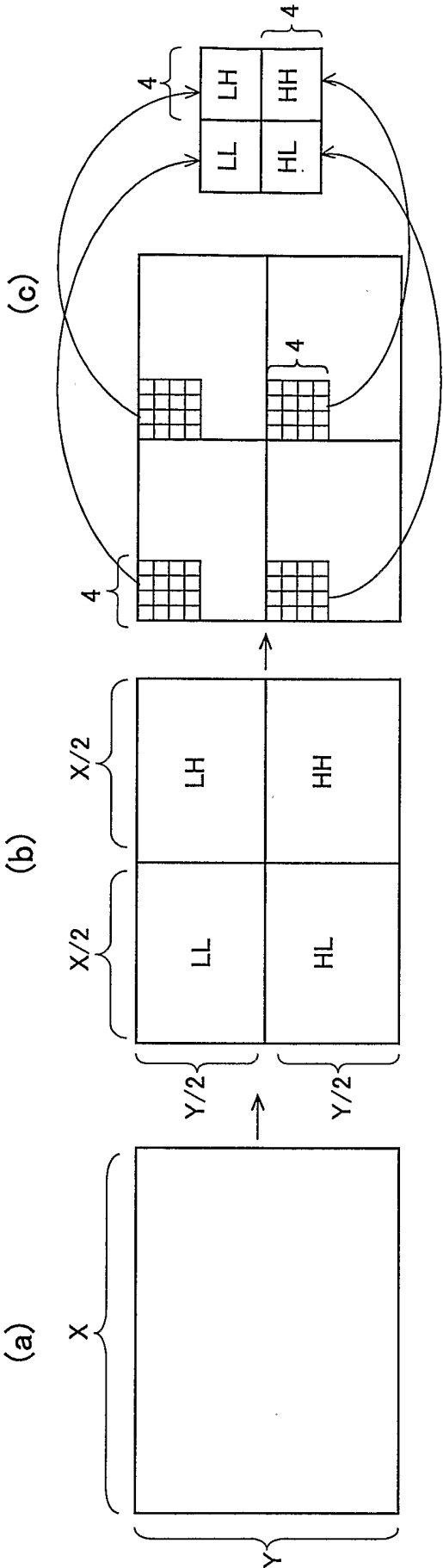


FIG.14

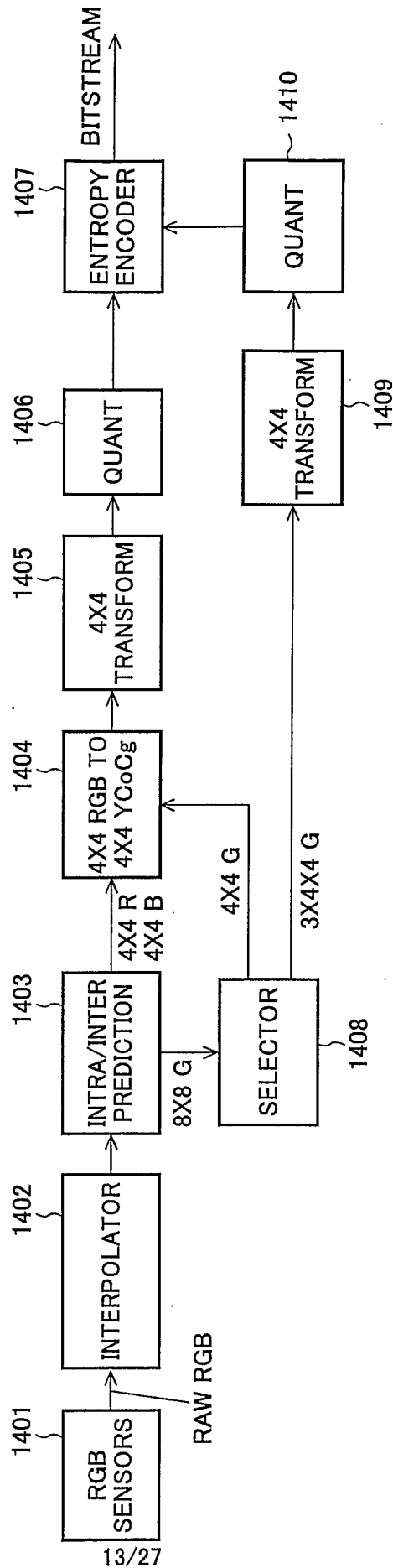


FIG.15

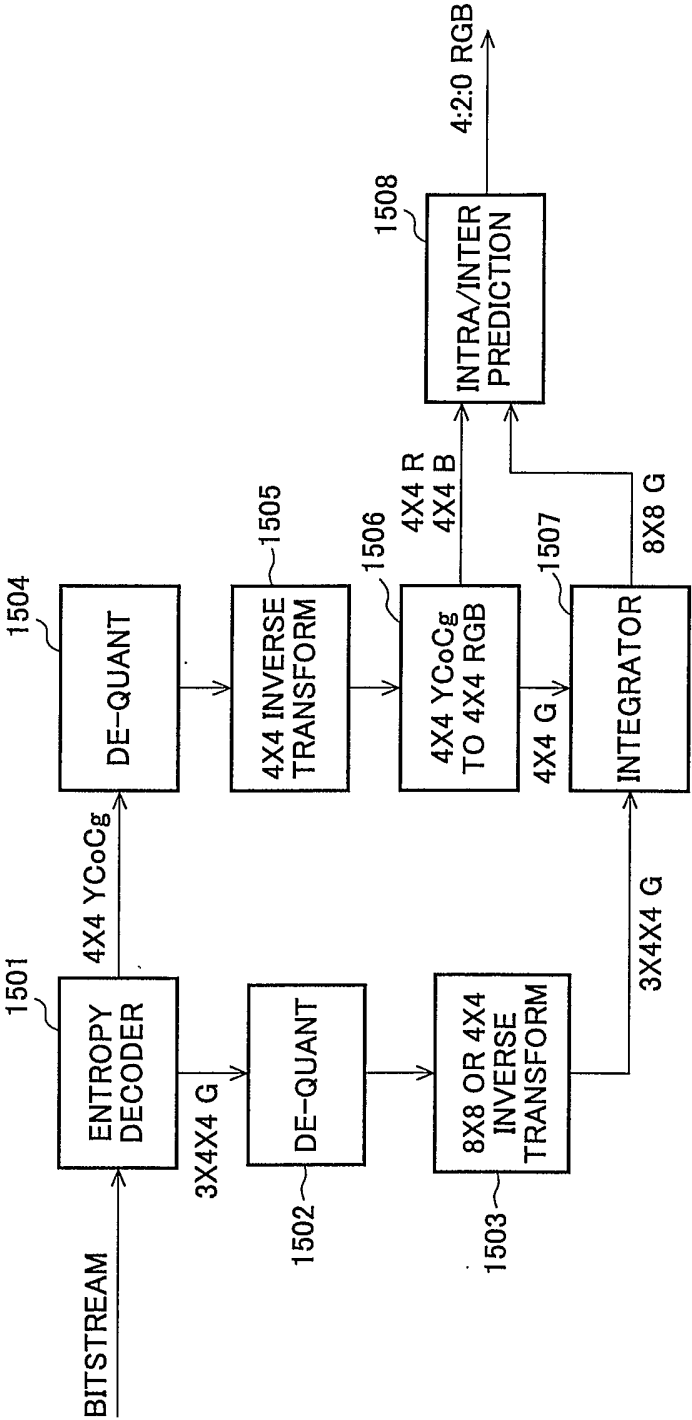
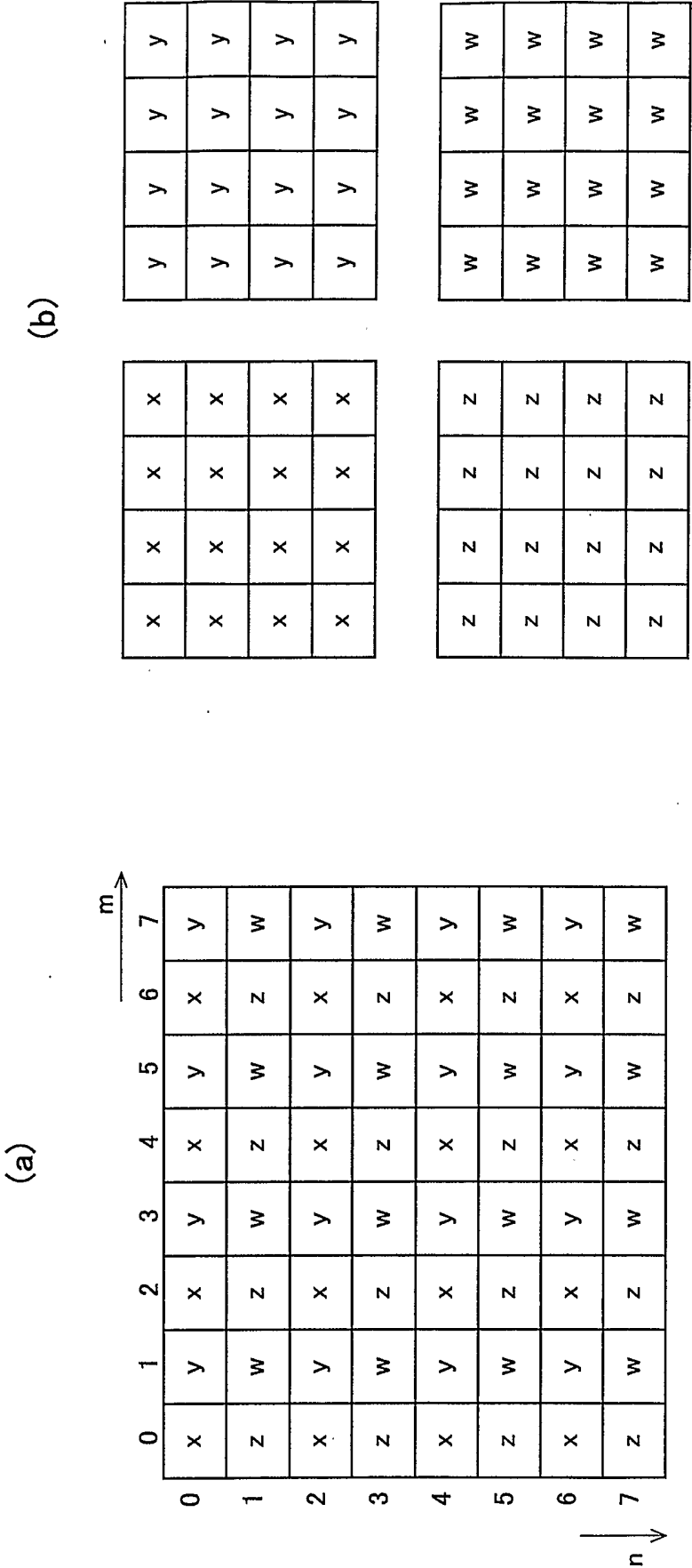
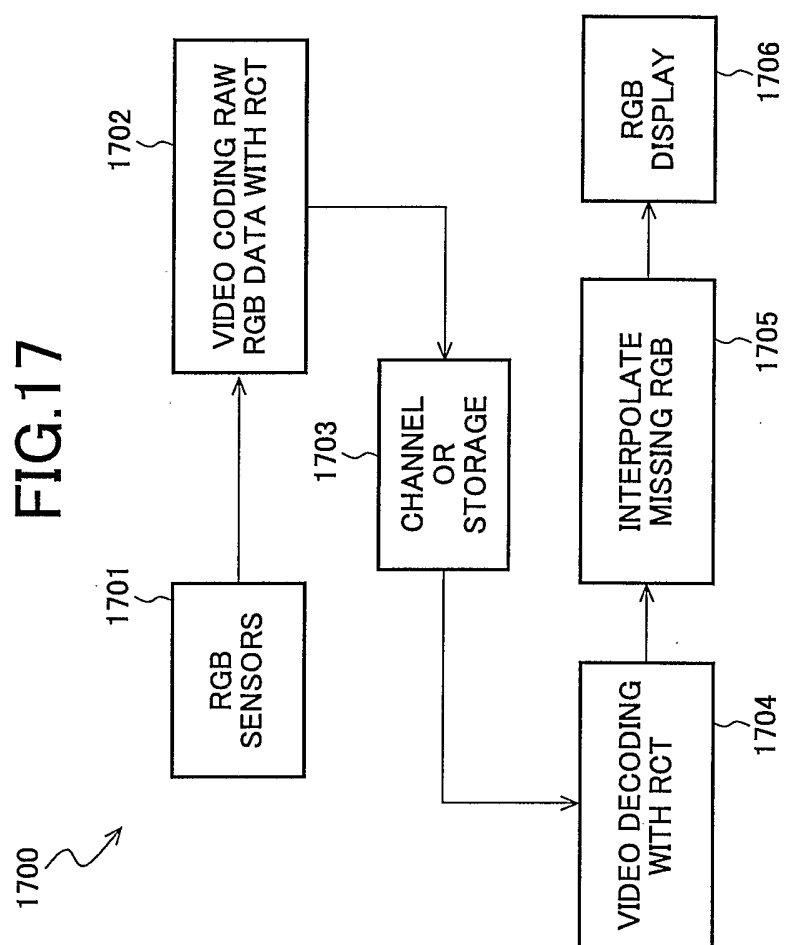


FIG.16





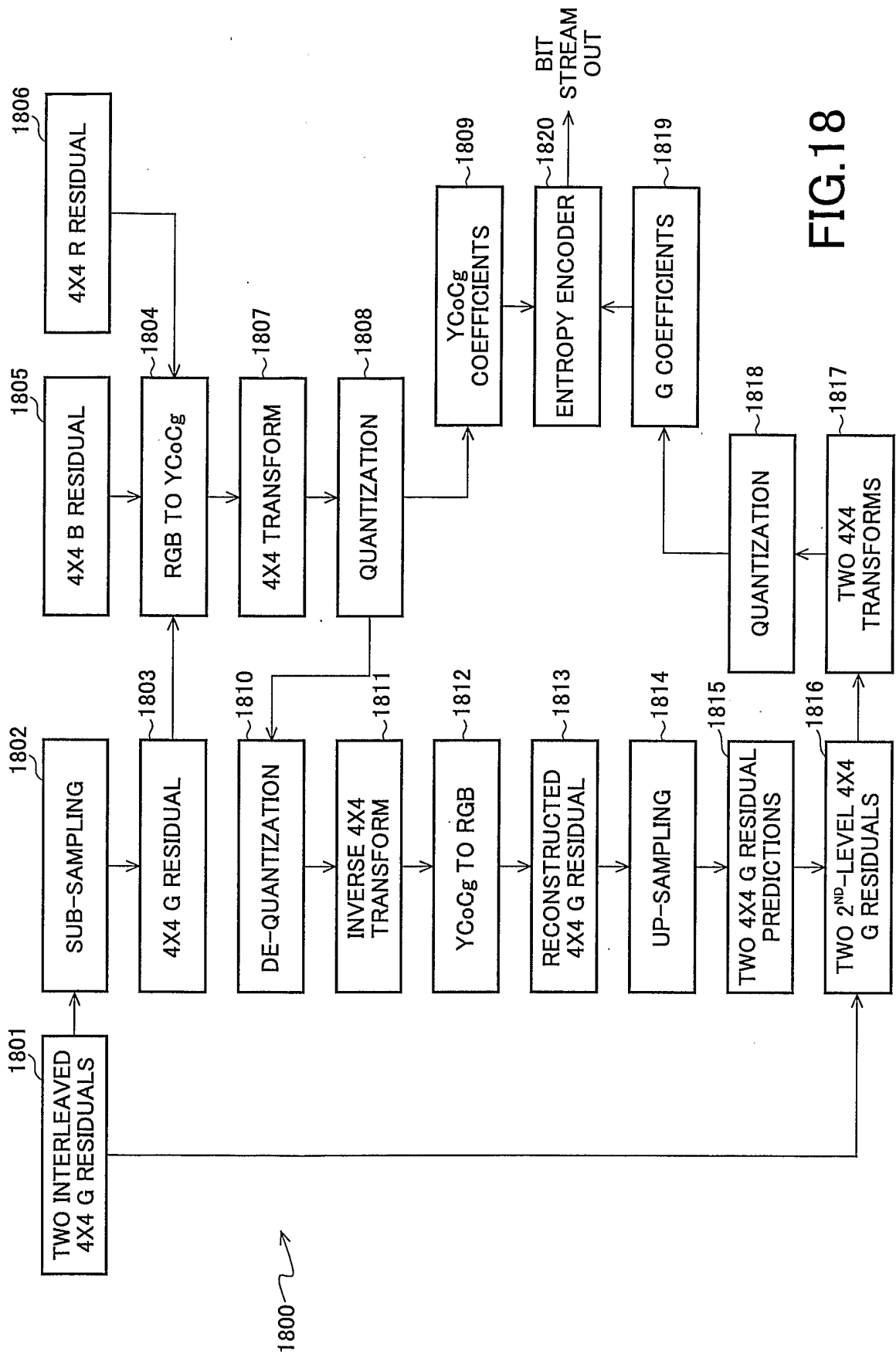


FIG. 19

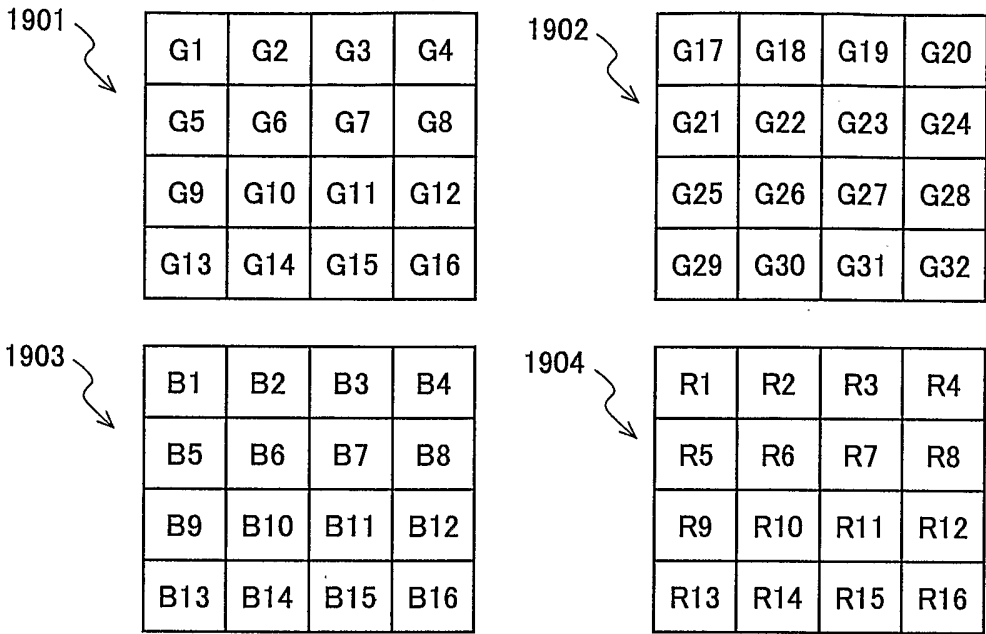


FIG. 20

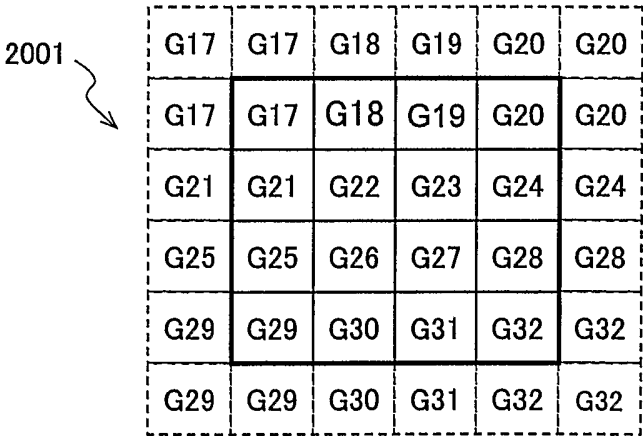
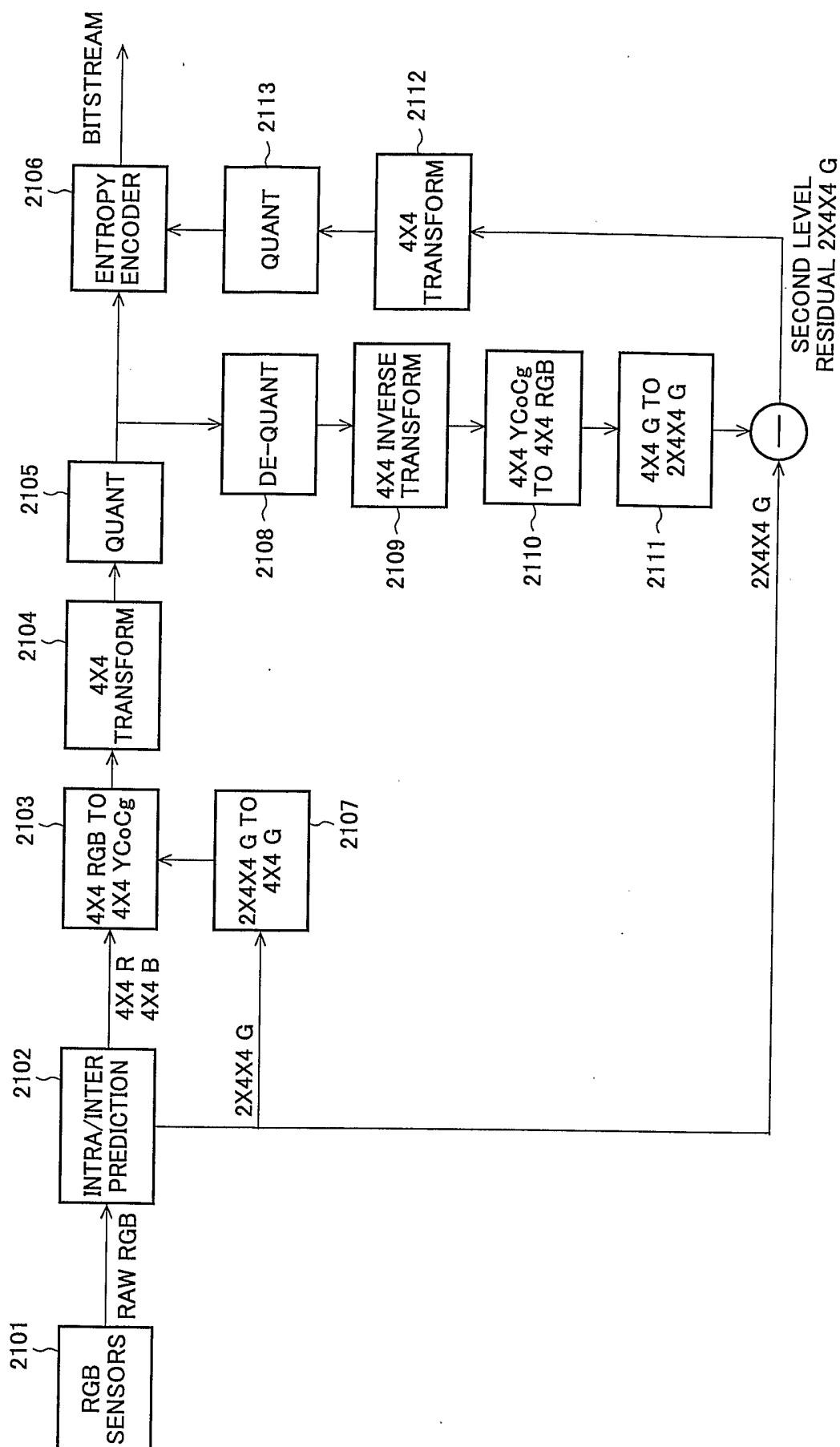


FIG.21



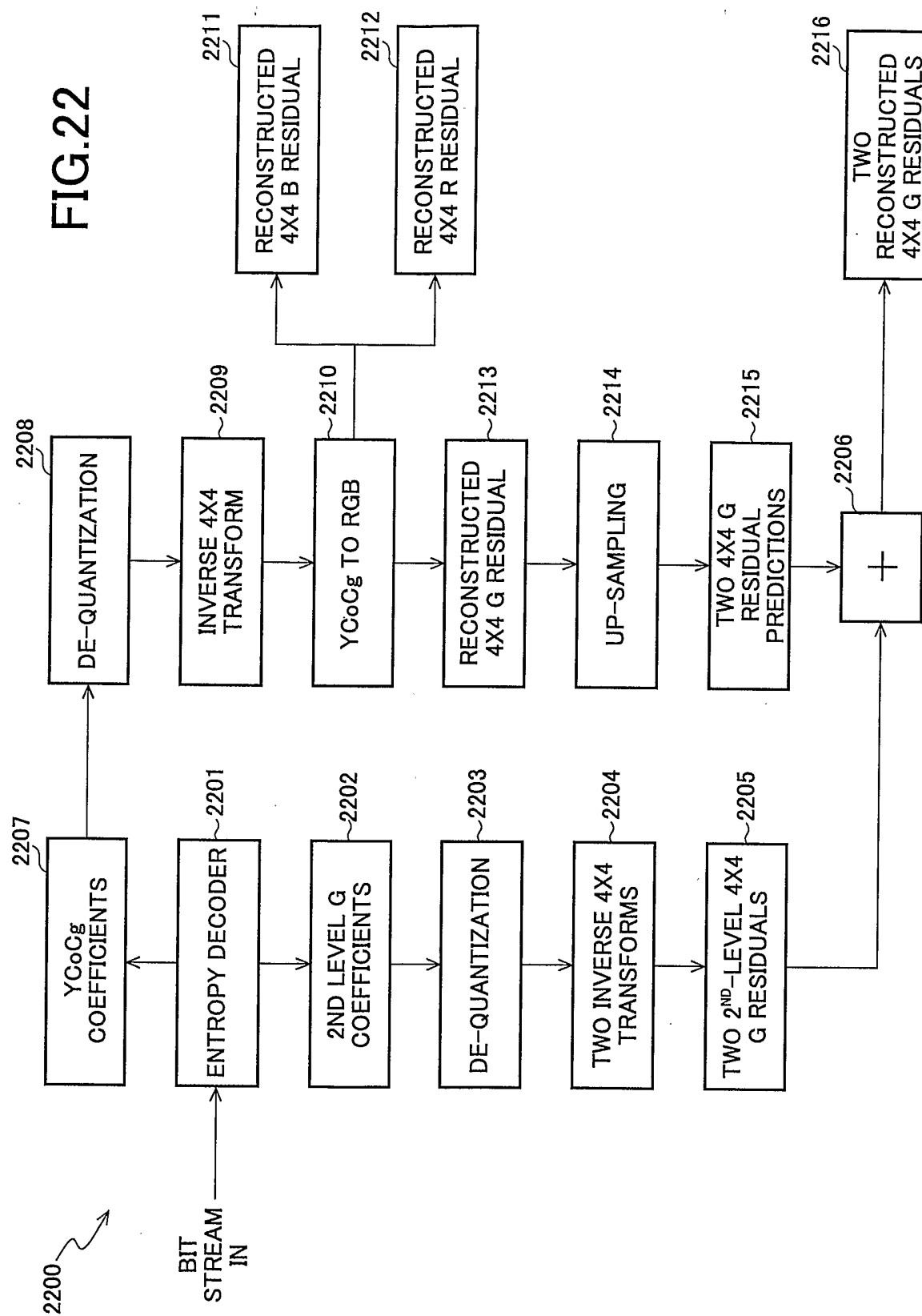


FIG.23

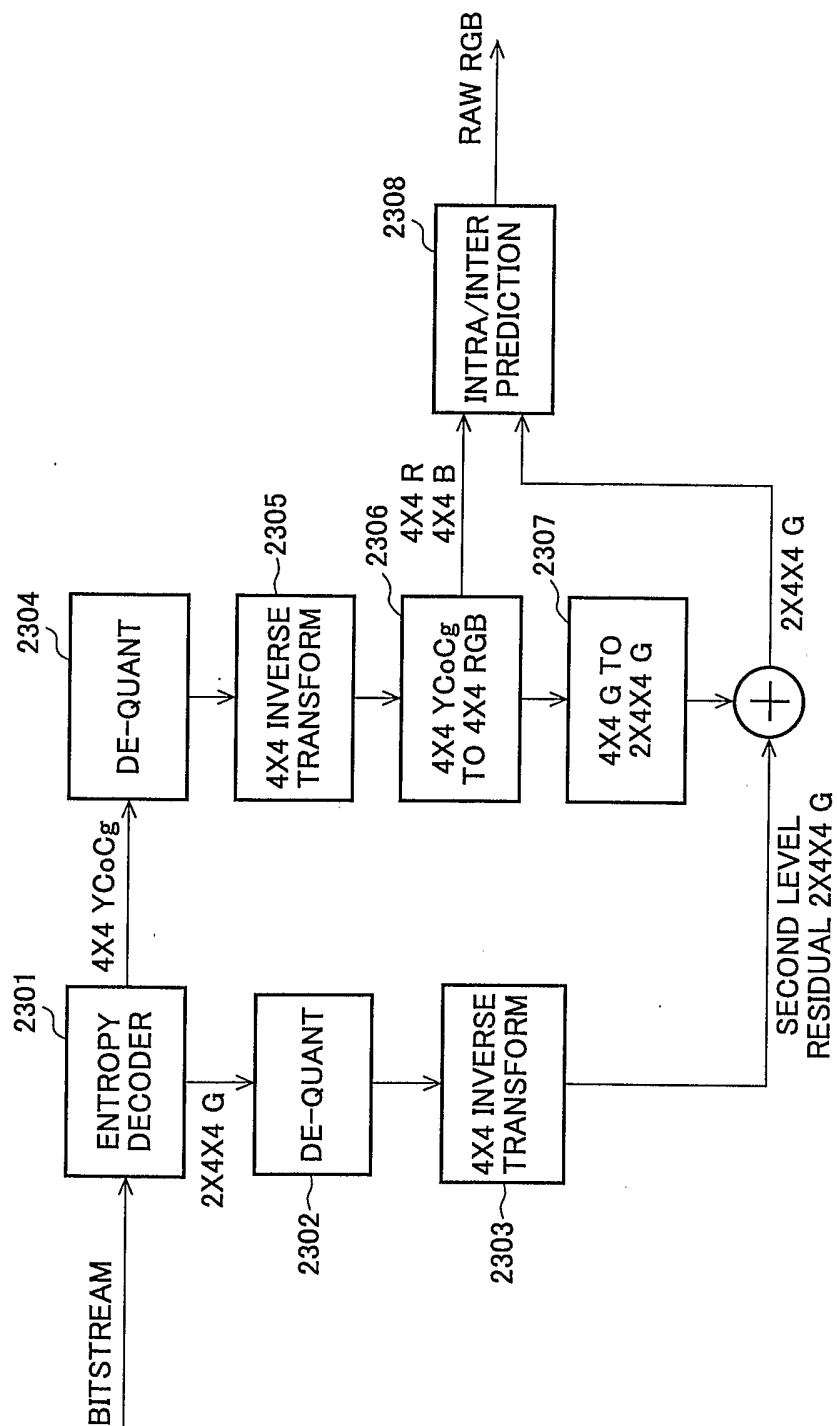


FIG.24

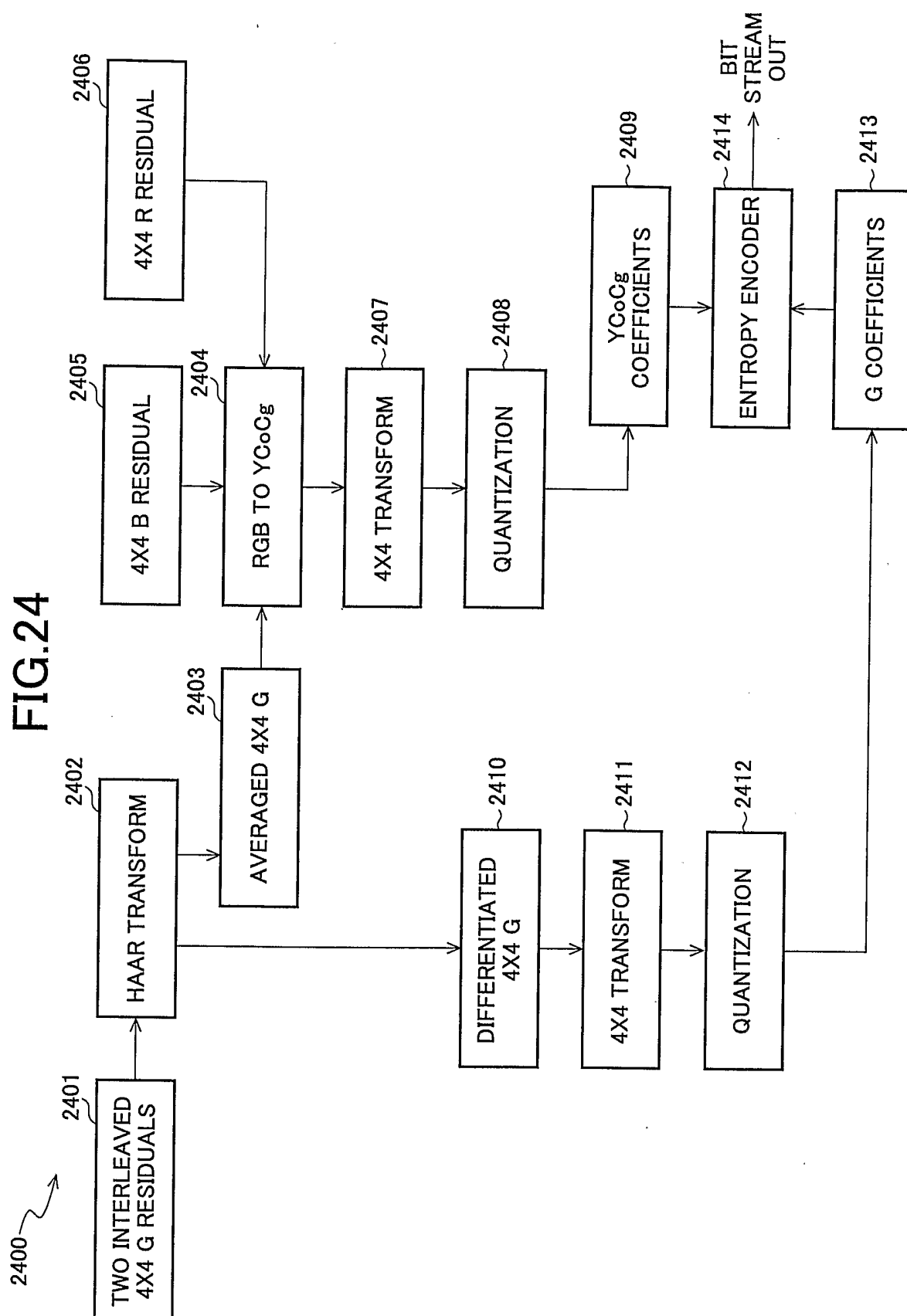


FIG.25

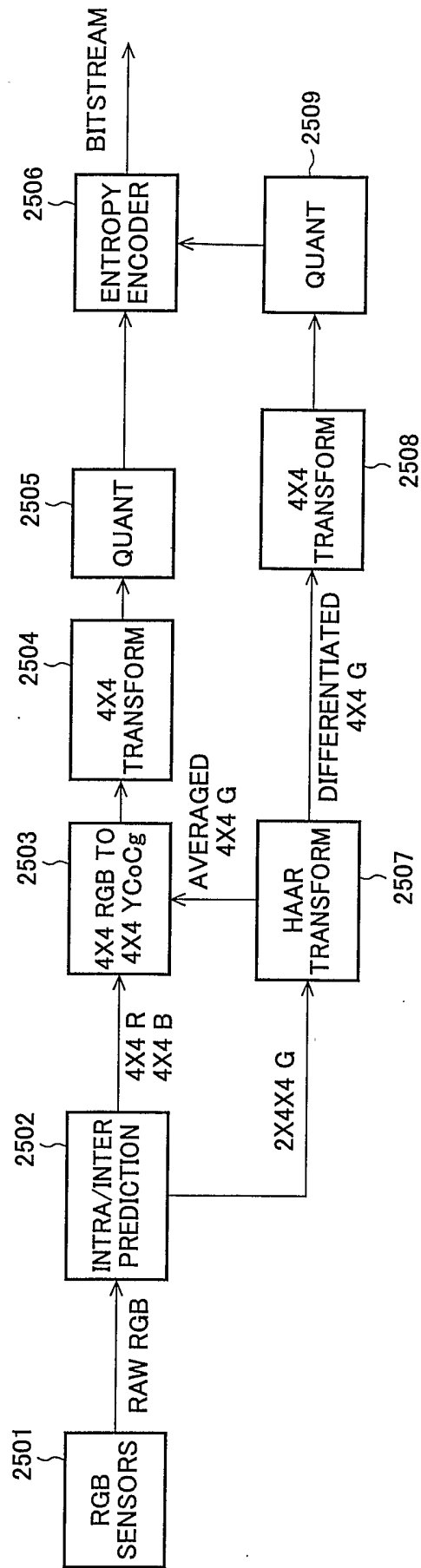


FIG.26

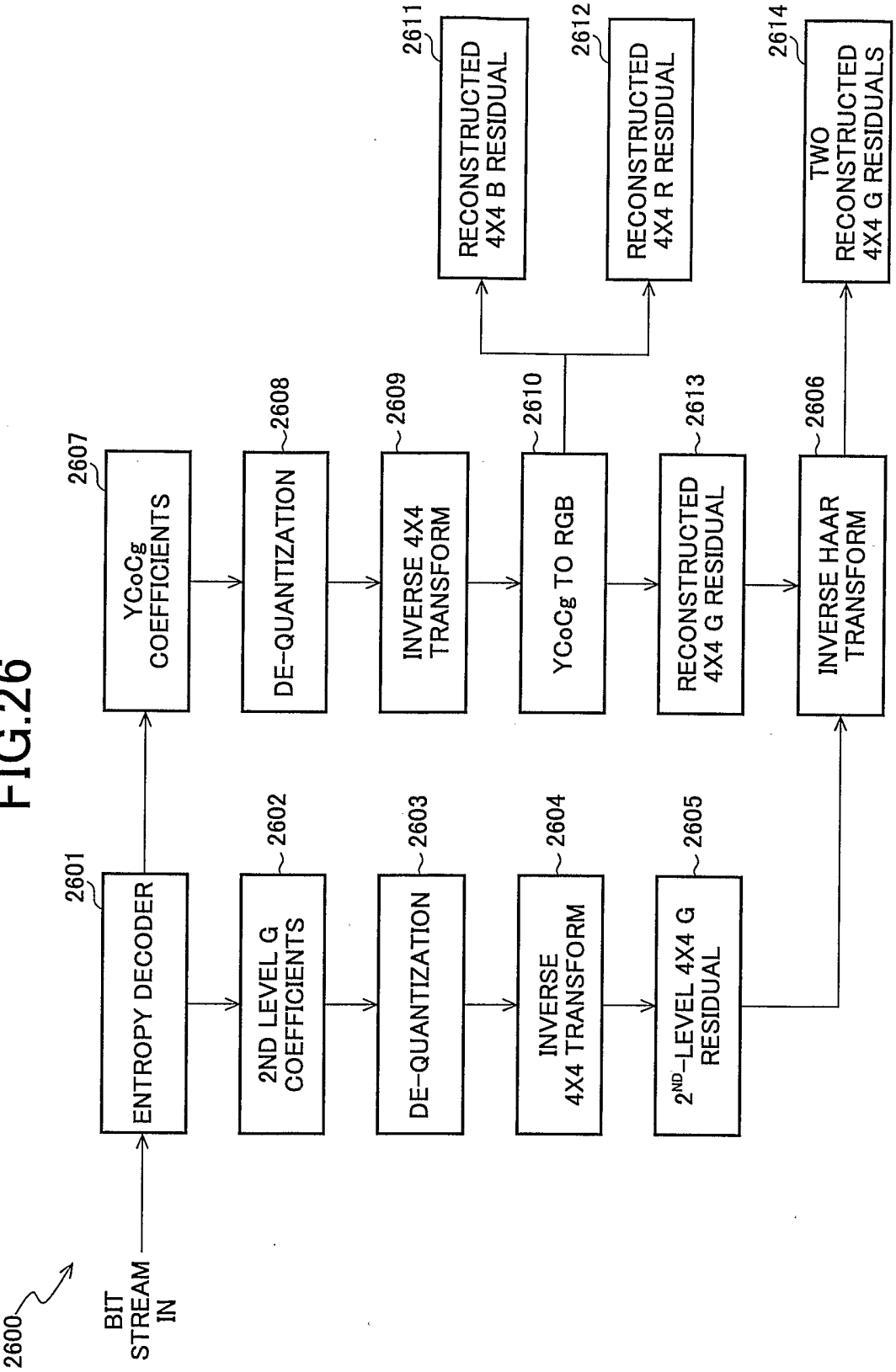


FIG.27

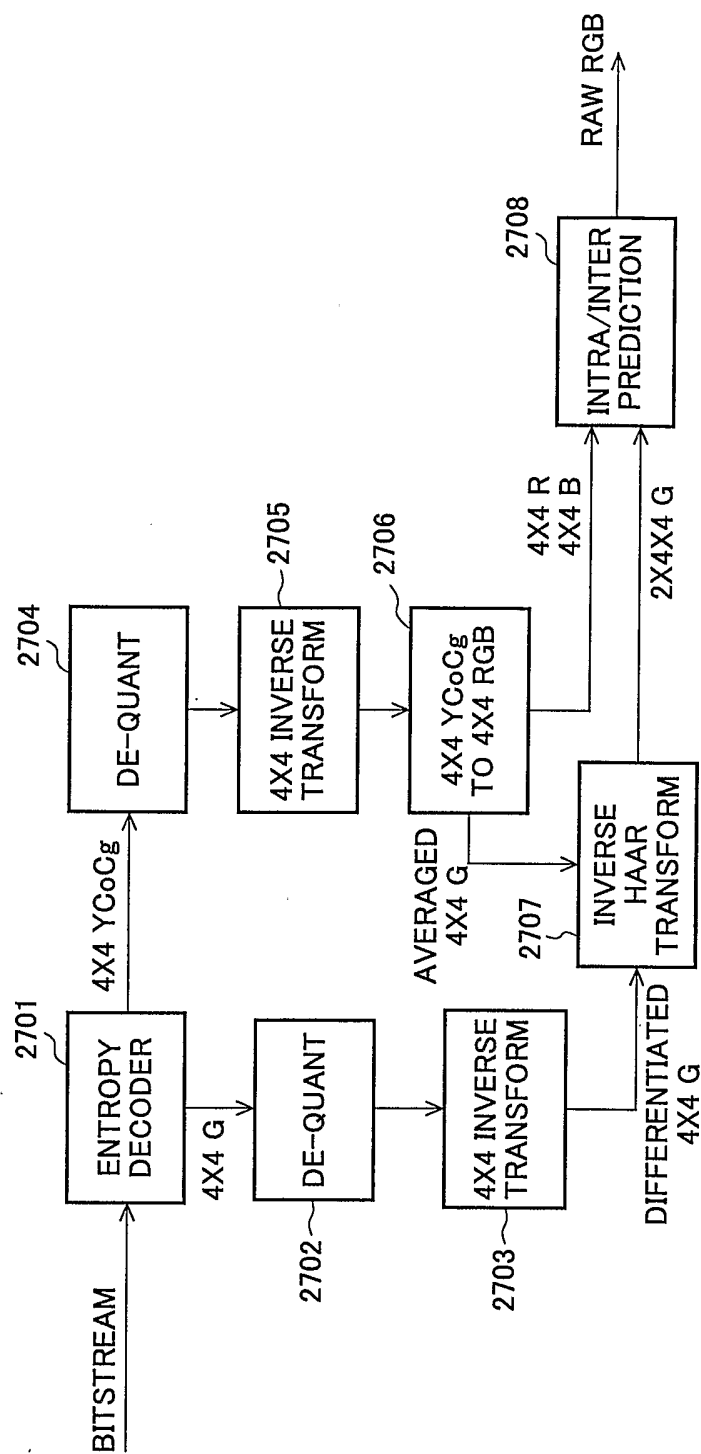


FIG.28

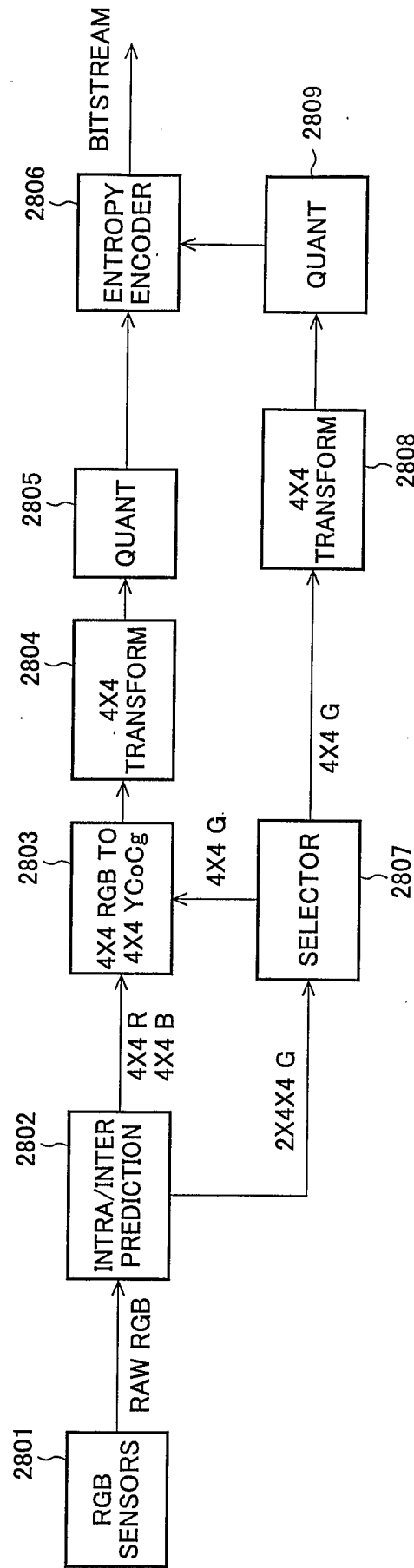
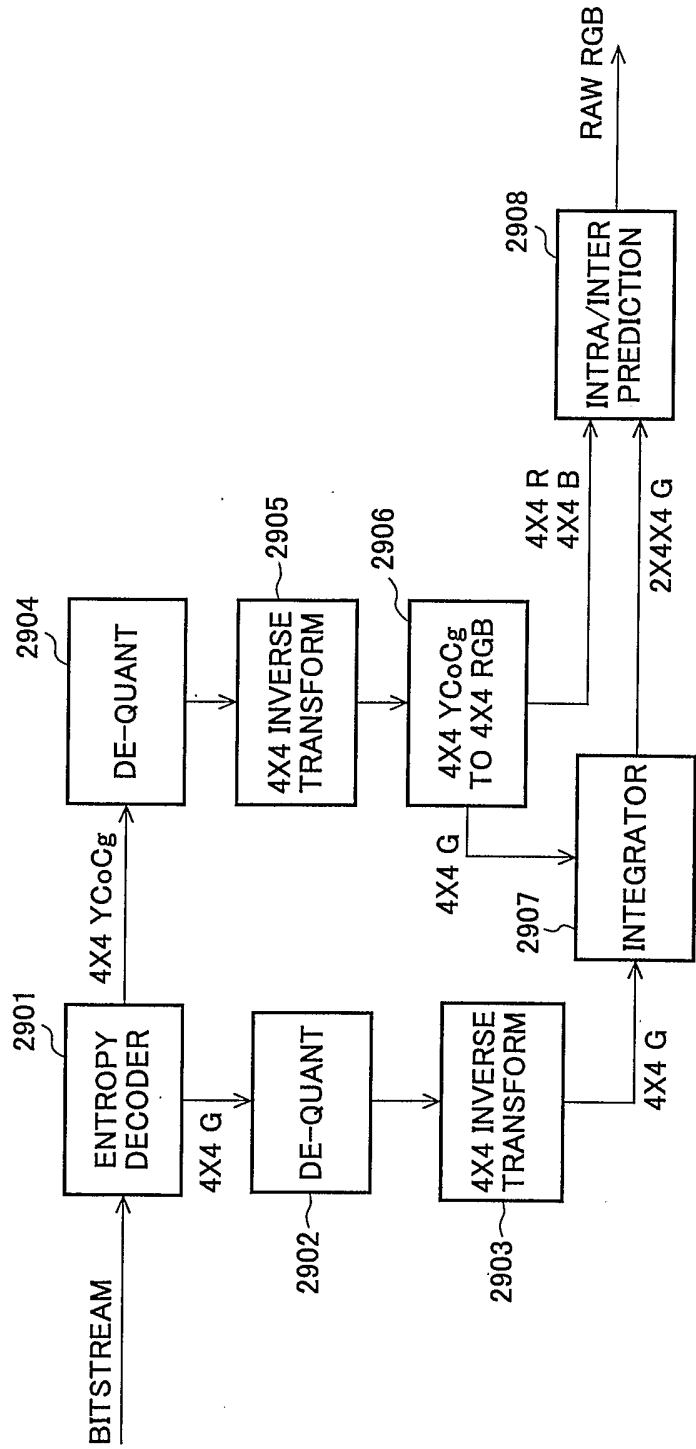


FIG.29



INTERNATIONALSEARCHREPORT

International application No.

PCT/JP2006/305640

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. H04N11/04 (2006.01) i, H04N1/41 (2006.01) i

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)

Int.Cl. H04N11/04, H04N1/41

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

Published examined utility model applications of Japan 1922-1996
 Published unexamined utility model applications of Japan 1971-2006
 Registered utility model specifications of Japan 1996-2006
 Published registered utility model applications of Japan 1994-2006

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
E, X	JP 2006-121669 A (microsoft CORP.) 2006.05.11, the	1-14
E, Y	whole of document, none	24-36
E, A		15-23
P, Y	JP 2005-160108 A (samsung electronics CORP.) 2005.06.16, the whole of document, none	24-36
A	JP 2003-324757 A (microsoft CORP.) 2003.11.14, the whole of document & US 2003/185439 A1 & EP 1349394 A2	1-36

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

* Special categories of cited documents:

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

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

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

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

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

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

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

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

“&” document member of the same patent family

Date of the actual completion of the international search

21.07.2006

Date of mailing of the international search report

01.08.2006

Name and mailing address of the ISA/JP

Japan Patent Office

3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan

Authorized officer

Naoki Sato

Telephone No. +81-3-3581-1101 Ext. 3581

5P

9562