



(43) International Publication Date  
02 January 2020 (02.01.2020)

(51) International Patent Classification:  
H04N 19/80 (2014.01) H04N 19/86 (2014.01)

(21) International Application Number:  
PCT/EP2019/066320

(22) International Filing Date:  
20 June 2019 (20.06.2019)

(25) Filing Language:  
English

(26) Publication Language:  
English

(30) Priority Data:  
1810787.0 29 June 2018 (29.06.2018) GB

(71) Applicant: CANON KABUSHIKI KAISHA [JP/JP];  
30-2, Shimomaruko 3-chome, Ohta-ku, TOKYO, 146-8501 (JP).

(71) Applicant (for AE only): CANON EUROPE LIMITED [GB/GB]; 3 The Square, Stockley Park, UXBRIDGE MIDDLESEX UB11 1ET (GB).

(72) Inventors: LAROCHE, Guillaume; 5 La Basse Romerais, 35250 SAINT AUBIN D'AUBIGNE (FR). TAQUET, Jonathan; La Frohardière, 35160 TALENSAC (FR). GISQUET, Christophe; 43 Rue du Martin Pêcheur, 35690 ACIGNE (FR). ONNO, Patrice; 5 rue Meslé, 35700 RENNES (FR).

(74) Agent: SANTARELLI; 49 avenue des Champs-Élysées, 75008 PARIS (FR).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN,

(54) Title: METHODS AND DEVICES FOR PERFORMING SAMPLE ADAPTIVE OFFSET (SAO) FILTERING

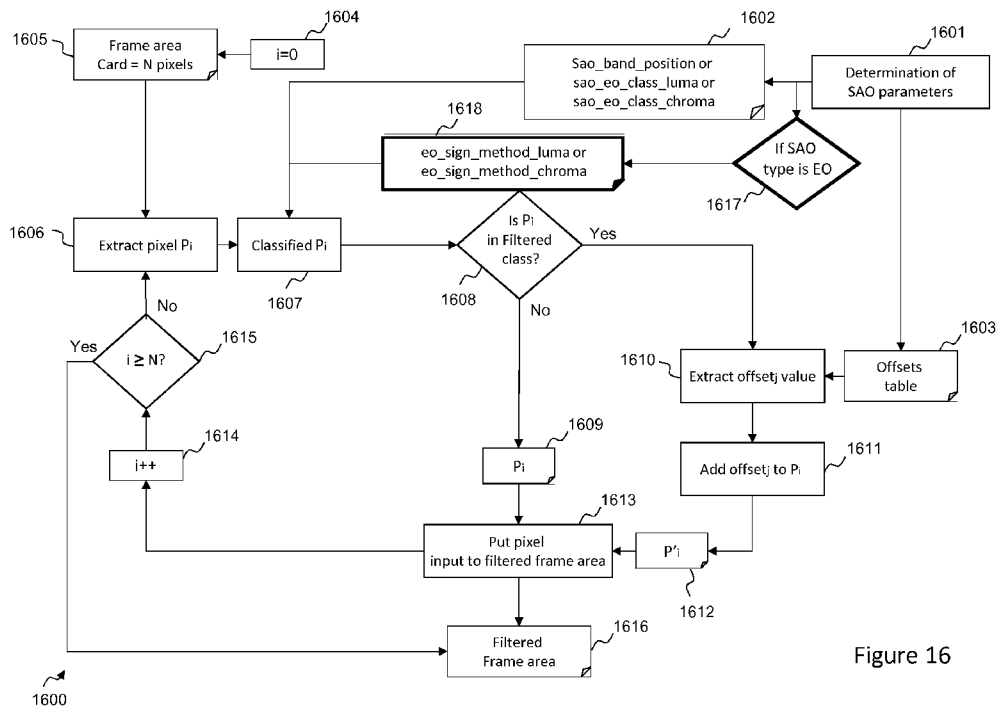


Figure 16

(57) Abstract: The present invention relates to methods for improving the coding efficiency of images subjected to the SAO filtering. According to a first aspect of the invention, there is provided a method of performing sample adaptive offset (SAO) filtering on an image comprising a plurality of image parts, the method comprising: obtaining an edge offset to perform a SAO filtering; applying an edge offset classification to at least one pixel of the image using a predetermined sign method for determining an index of the edge offset; wherein the predetermined sign method is signalled in the bitstream. Correspondingly, there is provided a method of providing an edge offset to perform a sample adaptive offset (SAO) filtering on an image, the method comprising: computing an edge offset to perform the SAO filtering; selecting a sign method among a plurality of possible sign methods for determining an index of the edge offset; and signalling the selected sign method in the bitstream thereby allowing the edge offset classification to be performed according



HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

**Published:**

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

METHODS AND DEVICES FOR PERFORMING SAMPLE ADAPTIVE OFFSET (SAO)  
FILTERING

FIELD OF THE INVENTION

5           The present invention relates to video coding and decoding.

BACKGROUND OF THE INVENTION

10           Recently, the Joint Video Experts Team (JVET), a collaborative team formed by MPEG and ITU-T Study Group 16's VCEG, commenced work on a new video coding standard referred to as Versatile Video Coding (VVC). The goal of VVC is to provide significant improvements in compression performance over the existing HEVC standard (i.e., typically twice as much as before) and to be completed in 2020. The main target applications and services include — but not limited to — 360-degree and high-dynamic-range (HDR) videos. In total, JVET evaluated responses from 32 organizations using formal subjective tests conducted by independent test labs. Some proposals demonstrated compression efficiency gains of typically 40% or more when compared to using HEVC. Particular effectiveness was shown on ultra-high definition (UHD) video test material. Thus, we may expect compression efficiency gains well-beyond the targeted 50% for the final standard.

20           The JVET exploration model (JEM) uses all the HEVC tools. One of these tools is sample adaptive offset (SAO) filtering. However, SAO is less efficient in the JEM reference software than in the HEVC reference software. This arises from fewer evaluations and from signalling inefficiencies compared to other loop filters.

25           In document JVET-J0024, a modify SAO edge offset classification is disclosed. This modification gives interesting results when low delay P configuration is tested but gives losses for low delay B configuration on top of the VTM software established for the VVC standard.

30           It is desirable to improve the coding efficiency of images subjected to the SAO filtering.

SUMMARY OF THE INVENTION

          The present invention has been devised to address one or more of the foregoing concerns.

According to a **first aspect of the invention**, there is provided a method of performing sample adaptive offset (SAO) filtering on an image comprising a plurality of image parts, the method comprising:

- obtaining an edge offset to perform a SAO filtering;
  - 5 applying an edge offset classification to at least one pixel of the image using a predetermined sign method for determining an index of the edge offset;
- wherein the predetermined sign method is signalled in the bitstream.

Correspondingly, there is provided a method of providing an edge offset to perform a sample adaptive offset (SAO) filtering on an image, the method comprising:

- 10 computing an edge offset to perform the SAO filtering;
- selecting a sign method for determining an index of the edge offset among a plurality of possible sign methods ; and
- signalling the selected sign method in the bitstream thereby allowing the edge offset classification to be performed according to the selected sign method.

- 15 Therefore, the methods according to the first aspect of the invention make it possible to improve the coding efficiency of images subjected to the SAO filtering.

Also, they provide a better flexibility compared to the prior art where the sign method is the same for all edge offsets. Consequently, these methods are better adapted to the quantization error and video content.

- 20 Optional features of the first aspect are further defined in the dependent appended claims.

According to embodiments, the predetermined sign method is signalled in the bitstream at CTU level.

- 25 An advantage of the signalization of the edge sign method at CTU level is an efficient adaptation according to the frame/slice content and quantization error which produces a coding efficiency improvement compared to the prior art.

According to embodiments, the predetermined sign method is signalled in the bitstream at frame, slice or sequence level.

- 30 An advantage of this embodiment may be that coding efficiency is improved when the frame/slice content and quantization error are relatively homogeneous inside the current slice/frame. A main advantage of the signaling at slice header is that the number of possible values for the sign method may be higher than for a CTU level signaling method.

- 35 According to embodiments, the predetermined sign method is CABAC coded.

An advantage of this implementation may be that the coding efficiency is increased while the frame/slice content and quantization error are relatively homogeneous inside the current slice/frame. As a result, one Edge offset sign method may be selected for many CTUs.

5           According to embodiments, the predetermined sign method is predicted.

An advantage of the second embodiment and all its sub-embodiments is that coding efficiency is increased due to the exploitation of the spatial correlation of the signal content and similar quantization error between CTU of the current slice/frame. This is because when the signal of the current slice is similar CTU by CTU and when the  
10           quantization error is also similar, the best selected sign methods at encoder side are often similar.

          According to embodiments, the predetermined sign method is selected at the encoder side to minimize the rate-distortion cost.

          An advantage of this embodiment may be that it should reach the maximum  
15           coding efficiency for the slice/frame level signaling.

          According to embodiments, the predetermined sign method is selected at the encoder side according to parameters of a current slice or frame.

          According to embodiments, the predetermined sign method defines a sign function taking into account a number N corresponding to a limit value.

20           According to embodiments, the predetermined sign method is selected at the encoder side based on a plurality of limit values N that are not consecutive.

          These embodiments may be very interesting when the signalling is at CTU level or inferior, as consecutive values do not correspond to the statistics of the natural content and related quantization error. One advantage is a coding efficiency  
25           improvement.

          According to embodiments, the predetermined sign method is selected at the encoder side based on a plurality of limit values N that depend on the slice type.

          According to embodiments, the predetermined sign method is selected at the encoder side based on a plurality of limit values N that depend on color components.

30           An advantage of this embodiment may be a coding efficiency improvement thanks to an adapted set of N values for each colour component type.

          According to embodiments, the predetermined sign method is selected at the encoder side based on a plurality of limit values N that depend on a Luma component only.

An advantage of this embodiment may be a coding efficiency improvement by removing the sign method selected for the chroma component.

According to embodiments, the predetermined sign method is selected at the encoder side based on a plurality of limit values N that are signalled in a slice header.

5 An advantage of this embodiment may be an increase of coding efficiency by setting the useful set at encoder side for each slice.

According to embodiments, the predetermined sign method is selected at the encoder side based on a plurality of limit values N.

10 According to embodiments, the plurality of limit values N depend on a color component.

According to a **second aspect of the invention**, there is provided a method of performing sample adaptive offset (SAO) filtering on an image comprising a plurality of image parts, the method comprising:

15 obtaining a plurality of edge offsets to perform a SAO filtering;  
wherein a sign of at least one edge offset of the plurality is signalled in the bitstream.

20 According to embodiments, this method may also comprise applying an edge offset classification to pixels of the image, wherein the edge offset classification takes into account a sign of at least one edge offset of the plurality which is signalled in the bitstream.

Correspondingly, there is provided a method of providing an edge offset to perform a sample adaptive offset (SAO) filtering on an image, the method comprising:  
25 obtaining a plurality of edge offsets to perform the SAO filtering;  
computing an index of the edge offsets by applying, for each edge offset, a sign method to a sign parameter N ; and  
signalling a sign of at least one edge offset of the plurality in the bitstream.

Therefore, the methods according to the second aspect of the invention make it possible to improve the coding efficiency of images subjected to the SAO filtering.

30 Optional features of the second aspect are further defined in the dependent appended claims.

According to embodiments, the sign of the edge offset is signalled for all the edge offsets of the plurality for which a sign parameter N to use in a sign function for determining an index of the edge offset is positive the sign of the edge offset is strictly positive.

According to embodiments, the sign of the edge offset is signalled for all the edge offsets of the plurality if all the sign parameters N to use in a sign function for determining an index of the edge offsets are higher than a predetermined threshold.

5 An advantage of this embodiment may be a coding efficiency increase compared to the implicit sign signalling of offsets for Edge offset classification when the value N used in the sign method is high.

According to embodiments, the at least one edge offset the sign of which is signalled belongs to the peak and valley configurations.

10 According to embodiments, the sign of the at least one edge offset is signalled whatever is a predetermined sign method used to determine an index of the edge offset.

According to embodiments, the sign the edge offset is signalled only for the edge offsets for which the sign parameters N to use for determining an index of these edge offsets are higher than a predetermined threshold.

15 According to embodiments, the sign the edge offset is signalled only for the edge offsets for which the sign parameters N to use for determining an index of these edge offsets are different from zero.

According to embodiments, the sign the edge offset is signalled only for the luma component.

20 An advantage of this embodiment may be a coding efficiency improvement compared to the signalling of signs for all colour components. This is because for Chroma component the implicit signalling may be more efficient than the explicit signalling of signs.

25 According to a **third aspect of the invention**, there is provided a method of performing sample adaptive offset (SAO) filtering on an image comprising a plurality of image parts, the method comprising:

obtaining a plurality of edge offsets to perform a SAO filtering;

wherein a sign of at least one edge offset of the plurality is signalled in the bitstream; and

30 wherein obtaining a plurality of edge offsets comprises predicting at least one edge offset the sign of which is signalled in the bitstream.

Correspondingly, there is provided a method of providing an edge offset to perform a sample adaptive offset (SAO) filtering on an image, the method comprising:

obtaining a plurality of edge offsets to perform the SAO filtering;

35 obtaining, for each edge offset, a sign of the edge offset;

wherein the method further comprises signalling the sign of at least one edge offset of the plurality in the bitstream; and

wherein obtaining a plurality of edge offsets comprises predicting at least one edge offset the sign of which is signalled in the bitstream.

5           Therefore, the methods according to the third aspect of the invention make it possible to improve the coding efficiency of images subjected to the SAO filtering.

Optional features of the third aspect are further defined in the dependent appended claims.

10           According to embodiments, the at least one edge offset is predicted from a default value.

According to embodiments, the default value is transmitted in the bitstream.

According to embodiments, the default value is predetermined for at least one sequence, for example all sequences.

15           According to embodiments, the method further comprises setting the sign of the default value.

According to embodiments, the default value depends on a sign parameter N to use in a sign function for determining an index of the edge offset.

20           An advantage of this embodiment may be a coding efficiency increase compared to a fix value for all edge offset sign methods. This is because the absolute offset values increase when the N value of the sign method increases.

25           According to embodiments, the method further comprises applying an edge offset classification to pixels of the image in order to classify the pixels in four categories, each category being associated with a given edge offset, and absolute predictor values of the edge offsets associated with the first and last categories corresponding to the peak and valley are higher than absolute predictor values of the edge offsets associated with the other categories.

30           This embodiment may be advantageous notably because the absolute average values of offsets O1 and O4 are often larger than the absolute average values of offsets O2 and O3. Consequently, this embodiment may provide a better coding efficiency.

35           According to embodiments, the method further comprises applying an edge offset classification to pixels of the image in order to classify the pixels in four categories, each category being associated with a given edge offset, and absolute predictor values of the edge offsets associated with the first and last categories corresponding to the peak and valley are the same.

According to embodiments, absolute predictor values of the edge offsets associated with the other categories are the same.

Advantageously, the process is simplified and is in line with the statistics selection of offsets values.

5           According to embodiments, the edge offset classification takes into account the sign of at least one edge offset of the plurality which is signalled in the bitstream.

          According to embodiments, the at least one edge offset is predicted from a sign parameter N to use in a sign function for determining an index of the edge offset.

10           According to embodiments, at least some edge offsets of the plurality the sign of which is signalled in the bitstream are not predicted.

          According to embodiments, the method further comprises applying an edge offset classification to pixels of the image in order to classify the pixels in four categories, each category being associated with a given edge offset, and only the edge offsets associated with the first and last categories corresponding to the peak and valley are  
15           predicted.

          An advantage of this embodiment may be additional coding efficiency. This is because the absolute average value of offset O1 and O4 is in average larger than offsets O2 and O3. In addition, the absolute values for O2 and O3 are often inferior to 1. Consequently, there may be no need to predict them and to signal a sign.

20           According to embodiments, the edge offsets are predicted for the luma component only.

          An advantage of this embodiment may be additional coding efficiency. This is because the absolute offset values are generally higher for luma than for chroma components. The prediction of Luma component may be efficient enough. This  
25           embodiment may thus provide a simplification as the prediction of Chroma offsets is avoided.

          According to embodiments, the method further comprises applying an edge offset classification to pixels of the image in order to classify the pixels in four categories, each category being associated with a given edge offset, and only the edge offsets  
30           associated with the first and last categories corresponding to the peak and valley, and the edge offsets for the luma component are predicted.

          According to embodiments, the at least one edge offset is predicted from a neighbouring edge offset value.

According to embodiments, the at least one edge offset is predicted from a neighbouring edge offset value only if the neighbouring edge offset is determined using the same sign method.

According to a **fourth aspect of the invention**, there is provided a method of performing sample adaptive offset (SAO) filtering on an image comprising a plurality of image parts, the method comprising:

5 obtaining a plurality of edge offsets to perform a SAO filtering;  
wherein obtaining a plurality of edge offsets comprises predicting at least one edge offset; and  
10 wherein a sign of the at least one edge offset is implicit and the minimum value of the edge offset is set to a default value different from zero.

Correspondingly, there is provided a method of providing an edge offset to perform a sample adaptive offset (SAO) filtering on an image, the method comprising:

15 computing a plurality of edge offsets to perform the SAO filtering;  
wherein computing a plurality of edge offsets comprises predicting at least one edge offset, and  
wherein a sign of the at least one edge offset is implicit and the minimum value of the edge offset is set to a default value different from zero.

Therefore, the methods according to the fourth aspect of the invention make  
20 it possible to improve the coding efficiency of images subjected to the SAO filtering.

As it does not require adding a sign of the offset as a syntax element, methods according to this aspect may be considered as a simplification compared to embodiments in which the sign is transmitted.

Optional features of the fourth aspect are further defined in the dependent  
25 appended claims.

According to embodiments, the at least one edge offset is predicted from a default value.

According to embodiments, the default value is transmitted in the bitstream.

According to embodiments, the default value is predetermined for at least  
30 one sequence, for example all sequences.

According to embodiments, the method further comprises setting the sign of the default value.

According to embodiments, the default value depends on a sign parameter N to use in a sign function for determining an index of the edge offset.

An advantage of this embodiment may be a coding efficiency increase compared to a fix value for all edge offset sign methods. This is because the absolute offset values increase when the N value of the sign method increases.

According to embodiments, the method further comprises applying an edge offset classification to pixels of the image in order to classify the pixels in four categories, each category being associated with a given edge offset, and absolute predictor values of the edge offsets associated with the first and last categories corresponding to the peak and valley are higher than absolute predictor values of the edge offsets associated with the other categories.

This embodiment may be advantageous notably because the absolute average values of offsets O1 and O4 are often larger than the absolute average values of offsets O2 and O3. Consequently, this embodiment may provide a better coding efficiency.

According to embodiments, the method further comprises applying an edge offset classification to pixels of the image in order to classify the pixels in four categories, each category being associated with a given edge offset, and absolute predictor values of the edge offsets associated with the first and last categories corresponding to the peak and valley are the same.

According to embodiments, absolute predictor values of the edge offsets associated with the other categories are the same.

Advantageously, the process is simplified and is in line with the statistics selection of offsets values.

According to embodiments, the edge offset classification takes into account the sign of at least one edge offset of the plurality which is signalled in the bitstream.

According to embodiments, the at least one edge offset is predicted from a sign parameter N to use in a sign function for determining an index of the edge offset.

According to embodiments, at least some edge offsets of the plurality are not predicted.

According to embodiments, the method further comprises applying an edge offset classification to pixels of the image in order to classify the pixels in four categories, each category being associated with a given edge offset, and only the edge offsets associated with the first and last categories corresponding to the peak and valley are predicted.

An advantage of this embodiment may be additional coding efficiency. This is because the absolute average value of offset O1 and O4 is in average larger than

offsets O2 and O3. In addition, the absolute values for O2 and O3 are often inferior to 1. Consequently, there may be no need to predict them.

According to embodiments, the edge offsets are predicted for the luma component only.

5 An advantage of this embodiment may be additional coding efficiency. This is because the absolute offset values are generally higher for luma than for chroma components. The prediction of Luma component may be efficient enough. This embodiment may thus provide a simplification as the prediction of Chroma offsets is avoided.

10 According to embodiments, the method further comprises applying an edge offset classification to pixels of the image in order to classify the pixels in four categories, each category being associated with a given edge offset, and only the edge offsets associated with the first and last categories corresponding to the peak and valley, and the edge offsets for the luma component are predicted.

15 According to embodiments, the at least one edge offset is predicted from a neighbouring edge offset value.

According to embodiments, the at least one edge offset is predicted from a neighbouring edge offset value only if the neighbouring edge offset is determined using the same sign method.

20 According to a **further aspect of the present invention**, there is provided a method of encoding an image comprising performing sample adaptive offset (SAO) filtering using one of the methods aforementioned.

25 According to a **further aspect of the present invention**, there is provided a method of decoding an image comprising performing sample adaptive offset (SAO) filtering using one of the methods aforementioned.

According to a **further aspect of the present invention**, there is provided a device for performing sample adaptive offset (SAO) filtering on an image comprising a plurality of image parts, the device comprising a microprocessor configured for performing the steps of one of the methods aforementioned.

30 According to a **further aspect of the present invention**, there is provided an encoder comprising the device aforementioned.

According to a **further aspect of the present invention**, there is provided a decoder comprising the device aforementioned.

According to a further aspect of the present invention, there is provided a system comprising an encoder as aforementioned and a decoder as aforementioned configured to communicate through a communication network.

The further aspects of the present invention have optional features and advantages similar to the first, second, third and fourth aspects above-mentioned.

Since the present invention may be implemented in software, the present invention may be embodied as computer readable code for provision to a programmable apparatus on any suitable carrier medium, and in particular a suitable tangible carrier medium or suitable transient carrier medium. A tangible carrier medium may comprise a storage medium such as a floppy disk, a CD-ROM, a hard disk drive, a magnetic tape device or a solid state memory device or 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.

#### 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** is a diagram for use in explaining a coding structure used in HEVC;

**Figure 2** is a block diagram schematically illustrating a data communication system in which one or more embodiments of the invention may be implemented;

**Figure 3** is a block diagram illustrating components of a processing device in which one or more embodiments of the invention may be implemented;

**Figure 4** is a flow chart illustrating steps of an encoding method according to embodiments of the invention;

**Figure 5** is a flow chart illustrating steps of a loop filtering process of in accordance with one or more embodiments of the invention;

**Figure 6** is a flow chart illustrating steps of a decoding method according to embodiments of the invention;

**Figures 7A and 7B** are diagrams for use in explaining edge-type SAO filtering in HEVC;

**Figure 8** is a diagram for use in explaining band-type SAO filtering in HEVC;

**Figure 9** is a flow chart illustrating the steps of a process to decode SAO parameters according to the HEVC specifications;

**Figure 10** is a flow chart illustrating in more detail one of the steps of the Figure 9 process;

**Figure 11** is a flow chart illustrating how SAO filtering is performed on an image part according to the HEVC specifications;

**Figure 12** is a flow chart illustrating steps carried out an encoder to determine SAO parameters for the CTUs of a group (frame or slice);

5 **Figure 13** is a flow chart illustrating steps of an example of a statistics computed at the encoder side that can be applied for the Edge Offset type filter;

**Figure 14** shows steps of a first encoding process;

**Figure 15** shows steps of a second encoding process;

10 **Figure 16** is a flow chart illustrating how SAO filtering is performed on an image part according to a first embodiment of the first aspect of the present invention;

**Figure 17** illustrates steps of a process of parsing of SAO parameters in the bitstream at the decoder side according to the first embodiment shown in Figure 16 ;

**Figure 18** illustrates steps of a process of parsing of SAO parameters in the bitstream at the decoder side according to a second embodiment;

15 **Figure 19** illustrates steps of a process of parsing of SAO parameters in the bitstream at the decoder side according to a third embodiment;

**Figure 20** illustrates an example of selection of a `eo_sign_method_X` by an encoder;

20 **Figure 21** is a flow chart illustrating how SAO filtering is performed on an image part according to a fourth embodiment of the first aspect of the present invention;

**Figure 22** is a flow chart illustrating how SAO filtering is performed on an image part according to a fifth embodiment of the first aspect of the present invention;

**Figure 23** illustrates steps of a process of parsing of SAO parameters in the bitstream at the decoder side according to a sixth embodiment ;

25 **Figure 24** is a flow chart illustrating steps of an example of a statistics computed at the encoder side that can be applied for the Edge Offset type filter;

**Figure 25** is a flow chart illustrating steps carried out an encoder to determine SAO parameters for the CTUs of a group (frame or slice);

30 **Figure 26** is a flow chart illustrating steps of a process to decode SAO parameters according to a second embodiment of the second aspect;

**Figures 27A and 27B** are diagrams for use in explaining edge-type SAO filtering according to embodiments;

**Figure 28** is a flow chart illustrating steps of a process to decode SAO parameters according to a third embodiment of the second aspect;

**Figure 29** is a flow chart illustrating steps of a process to decode SAO parameters according to a fourth embodiment of the second aspect;

**Figure 30** is a flow chart illustrating steps of a process to decode SAO parameters according to a fifth embodiment of the second aspect;

5 **Figure 31** is a flow chart illustrating steps of a process to decode SAO parameters according to a sixth embodiment of the second aspect;

**Figure 32** is a flow chart illustrating steps of a process to decode SAO parameters according to a seventh embodiment of the second aspect;

10 **Figure 33** illustrates steps of a prediction process that may be performed during a (SAO) filtering embodiment at decoder side according to a first embodiment of the third aspect;

**Figure 34** illustrates steps of a prediction process that may be performed during a (SAO) filtering embodiment at decoder side according to a second embodiment of the third aspect;

15 **Figure 35** illustrates steps of a prediction process that may be performed during a (SAO) filtering embodiment at decoder side according to a third embodiment of the third aspect;

20 **Figure 36** illustrates steps of a prediction process that may be performed during a (SAO) filtering embodiment at decoder side according to a fourth embodiment of the third aspect;

**Figure 37** illustrates steps of a prediction process that may be performed during a (SAO) filtering embodiment at decoder side according to a fifth embodiment of the third aspect;

25 **Figure 38** illustrates steps of a prediction process that may be performed during a (SAO) filtering embodiment at decoder side according to a sixth embodiment of the third aspect;

**Figure 39** is a diagram showing a system comprising an encoder or a decoder and a communication network according to embodiments of the present invention.

30

#### DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION

**Figure 1** relates to a coding structure used in the High Efficiency Video Coding (HEVC) video standard. A video sequence 1 is made up of a succession of digital images  $i$ . Each such digital image is represented by one or more matrices. The matrix coefficients represent pixels.

35

An image 2 of the sequence may be divided into slices 3. A slice may in some instances constitute an entire image. These slices are divided into non-overlapping Coding Tree Units (CTUs). A Coding Tree Unit (CTU) is the basic processing unit of the High Efficiency Video Coding (HEVC) video standard and conceptually corresponds in structure to macroblock units that were used in several previous video standards. A CTU is also sometimes referred to as a Largest Coding Unit (LCU). A CTU has luma and chroma component parts, each of which component parts is called a Coding Tree Block (CTB). These different color components are not shown in Figure 1.

A CTU is generally of size 64 pixels x 64 pixels. Each CTU may in turn be iteratively divided into smaller variable-size Coding Units (CUs) using a quadtree decomposition.

Coding units are the elementary coding elements and are constituted by two kinds of sub-unit called a Prediction Unit (PU) and a Transform Unit (TU). The maximum size of a PU or TU is equal to the CU size. A Prediction Unit corresponds to the partition of the CU for prediction of pixels values. Various different partitions of a CU into PUs are possible as shown by 606 including a partition into 4 square PUs and two different partitions into 2 rectangular PUs. A Transform Unit is an elementary unit that is subjected to spatial transformation using DCT. A CU can be partitioned into TUs based on a quadtree representation 607.

Each slice is embedded in one Network Abstraction Layer (NAL) unit. In addition, the coding parameters of the video sequence are stored in dedicated NAL units called parameter sets. In HEVC and H.264/AVC two kinds of parameter sets NAL units are employed: first, a Sequence Parameter Set (SPS) NAL unit that gathers all parameters that are unchanged during the whole video sequence. Typically, it handles the coding profile, the size of the video frames and other parameters. Secondly, a Picture Parameter Set (PPS) NAL unit includes parameters that may change from one image (or frame) to another of a sequence. HEVC also includes a Video Parameter Set (VPS) NAL unit which contains parameters describing the overall structure of the bitstream. The VPS is a new type of parameter set defined in HEVC, and applies to all of the layers of a bitstream. A layer may contain multiple temporal sub-layers, and all version 1 bitstreams are restricted to a single layer. HEVC has certain layered extensions for scalability and multiview and these will enable multiple layers, with a backwards compatible version 1 base layer.

**Figure 2** illustrates a data communication system in which one or more embodiments of the invention may be implemented. The data communication system comprises a transmission device, in this case a server 201, which is operable to transmit data packets of a data stream to a receiving device, in this case a client terminal 202, via a data communication network 200. The data communication network 200 may be a Wide Area Network (WAN) or a Local Area Network (LAN). Such a network may be for example a wireless network (Wifi / 802.11a or b or g), an Ethernet network, an Internet network or a mixed network composed of several different networks. In a particular embodiment of the invention the data communication system may be a digital television broadcast system in which the server 201 sends the same data content to multiple clients.

The data stream 204 provided by the server 201 may be composed of multimedia data representing video and audio data. Audio and video data streams may, in some embodiments of the invention, be captured by the server 201 using a microphone and a camera respectively. In some embodiments data streams may be stored on the server 201 or received by the server 201 from another data provider, or generated at the server 201. The server 201 is provided with an encoder for encoding video and audio streams in particular to provide a compressed bitstream for transmission that is a more compact representation of the data presented as input to the encoder.

In order to obtain a better ratio of the quality of transmitted data to quantity of transmitted data, the compression of the video data may be for example in accordance with the HEVC format or H.264/AVC format.

The client 202 receives the transmitted bitstream and decodes the reconstructed bitstream to reproduce video images on a display device and the audio data by a loud speaker.

Although a streaming scenario is considered in the example of Figure 2, it will be appreciated that in some embodiments of the invention the data communication between an encoder and a decoder may be performed using for example a media storage device such as an optical disc.

In one or more embodiments of the invention a video image is transmitted with data representative of compensation offsets for application to reconstructed pixels of the image to provide filtered pixels in a final image.

**Figure 3** schematically illustrates a processing device 300 configured to implement at least one embodiment of the present invention. The processing device 300

may be a device such as a micro-computer, a workstation or a light portable device. The device 300 comprises a communication bus 313 connected to:

-a central processing unit 311, such as a microprocessor, denoted CPU;  
-a read only memory 307, denoted ROM, for storing computer programs for  
5 implementing the invention;

-a random access memory 312, 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 of encoding a sequence of digital images and/or the method of decoding a bitstream according to  
10 embodiments of the invention; and

-a communication interface 302 connected to a communication network 303 over which digital data to be processed are transmitted or received

Optionally, the apparatus 300 may also include the following components:

-a data storage means 304 such as a hard disk, for storing computer  
15 programs for implementing methods of one or more embodiments of the invention and data used or produced during the implementation of one or more embodiments of the invention;

-a disk drive 305 for a disk 306, the disk drive being adapted to read data from the disk 306 or to write data onto said disk;

20 -a screen 309 for displaying data and/or serving as a graphical interface with the user, by means of a keyboard 310 or any other pointing means.

The apparatus 300 can be connected to various peripherals, such as for example a digital camera 320 or a microphone 308, each being connected to an input/output card (not shown) so as to supply multimedia data to the apparatus 300.

25 The communication bus provides communication and interoperability between the various elements included in the apparatus 300 or connected to it. The representation of the bus is not limiting and in particular the central processing unit is operable to communicate instructions to any element of the apparatus 300 directly or by means of another element of the apparatus 300.

30 The disk 306 can be replaced by any information medium such as for example a compact disk (CD-ROM), rewritable or not, a ZIP disk or a memory card and, in general terms, by an information storage means that can be read by a microcomputer or by a microprocessor, integrated or not into the apparatus, possibly removable and adapted to store one or more programs whose execution enables the method of

encoding a sequence of digital images and/or the method of decoding a bitstream according to the invention to be implemented.

The executable code may be stored either in read only memory 306, on the hard disk 304 or on a removable digital medium such as for example a disk 306 as described previously. According to a variant, the executable code of the programs can be received by means of the communication network 303, via the interface 302, in order to be stored in one of the storage means of the apparatus 300 before being executed, such as the hard disk 304.

The central processing unit 311 is adapted to control and direct the execution of the instructions or portions of software code of the program or programs according to the invention, instructions that are stored in one of the aforementioned storage means. On powering up, the program or programs that are stored in a non-volatile memory, for example on the hard disk 304 or in the read only memory 306, are transferred into the random access memory 312, which then contains the executable code of the program or programs, as well as registers for storing the variables and parameters necessary for implementing the invention.

In this embodiment, the apparatus is a programmable apparatus which uses software to implement the invention. However, alternatively, the present invention may be implemented in hardware (for example, in the form of an Application Specific Integrated Circuit or ASIC).

**Figure 4** illustrates a block diagram of an encoder according to at least one embodiment of the invention. The encoder is represented by connected modules, each module being adapted to implement, for example in the form of programming instructions to be executed by the CPU 311 of device 300, at least one corresponding step of a method implementing at least one embodiment of encoding an image of a sequence of images according to one or more embodiments of the invention.

An original sequence of digital images  $i_0$  to  $i_n$  401 is received as an input by the encoder 400. Each digital image is represented by a set of samples, known as pixels. It is recalled that a pixel comprises three components and that a sample designates one of them.

A bitstream 410 is output by the encoder 400 after implementation of the encoding process. The bitstream 410 comprises a plurality of encoding units or slices, each slice comprising a slice header for transmitting encoding values of encoding parameters used to encode the slice and a slice body, comprising encoded video data.

The input digital images  $i_0$  to  $i_{401}$  are divided into blocks of pixels by module 402. The blocks correspond to image portions and may be of variable sizes (e.g. 4x4, 8x8, 16x16, 32x32, 64x64, 128x128 pixels and several rectangular block sizes can be also considered). A coding mode is selected for each input block. Two families of coding modes are provided: coding modes based on spatial prediction coding (Intra prediction), and coding modes based on temporal prediction (Inter coding, Merge, SKIP). The possible coding modes are tested.

Module 403 implements an Intra prediction process, in which the given block to be encoded is predicted by a predictor computed from pixels of the neighbourhood of said block to be encoded. An indication of the selected Intra predictor and the difference between the given block and its predictor is encoded to provide a residual if the Intra coding is selected.

Temporal prediction is implemented by motion estimation module 404 and motion compensation module 405. Firstly a reference image/picture from among a set of reference images/pictures 416 is selected, and a portion of the reference image/picture, also called reference area or image portion, which is the closest area to the given block to be encoded, is selected by the motion estimation module 404. Motion compensation module 405 then predicts the block to be encoded using the selected area. The difference between the selected reference area and the given block, also called a residual block, is computed by the motion compensation module 405. The selected reference area is indicated by a motion vector.

Thus in both cases (spatial and temporal prediction), a residual is computed by subtracting the prediction from the original block.

In the INTRA prediction implemented by module 403, a prediction direction is encoded. In the temporal prediction, at least one motion vector is encoded.

Information relative to the motion vector and the residual block is encoded if the Inter prediction is selected. To further reduce the bitrate, assuming that motion is homogeneous, the motion vector is encoded by difference with respect to a motion vector predictor. Motion vector predictors of a set of motion information predictors is obtained from the motion vectors field 418 by a motion vector prediction and coding module 417.

The encoder 400 further comprises a selection module 406 for selection of the coding mode by applying an encoding cost criterion, such as a rate-distortion criterion. In order to further reduce redundancies a transform (such as DCT) is applied by transform module 407 to the residual block, the transformed data obtained is then quantized by quantization module 408 and entropy encoded by entropy encoding module

409. Finally, the encoded residual block of the current block being encoded is inserted into the bitstream 410.

The encoder 400 also performs decoding of the encoded image in order to produce a reference image for the motion estimation of the subsequent images. This enables the encoder and the decoder receiving the bitstream to have the same reference frames. The dequantization module 411 performs dequantization of the quantized data, followed by an inverse transform by inverse transform module 412. The intra prediction module 413 uses the prediction information to determine which predictor to use for a given block and the motion compensation module 414 actually adds the residual obtained by module 412 to the reference area obtained from the set of reference images/pictures 416.

Post filtering is then applied by module 415 to filter the reconstructed frame of pixels. In the embodiments of the invention an SAO loop filter is used in which compensation offsets are added to the pixel values of the reconstructed pixels of the reconstructed image

**Figure 5** is a flow chart illustrating steps of loop filtering process according to at least one embodiment of the invention. In an initial step 51, the encoder generates the reconstruction of the full frame. Next, in step 52 a deblocking filter is applied on this first reconstruction in order to generate a deblocked reconstruction 53. The aim of the deblocking filter is to remove block artifacts generated by residual quantization and block motion compensation or block Intra prediction. These artifacts are visually important at low bitrates. The deblocking filter operates to smooth the block boundaries according to the characteristics of two neighboring blocks. The encoding mode of each block, the quantization parameters used for the residual coding, and the neighboring pixel differences in the boundary are taken into account. The same criterion/classification is applied for all frames and no additional data is transmitted. The deblocking filter improves the visual quality of the current frame by removing blocking artifacts and it also improves the motion estimation and motion compensation for subsequent frames. Indeed, high frequencies of the block artifact are removed, and so these high frequencies do not need to be compensated for with the texture residual of the following frames.

After the deblocking filter, the deblocked reconstruction is filtered by a sample adaptive offset (SAO) loop filter in step 54 using SAO parameters determined in accordance with embodiments of the invention. The resulting frame 55 may then be filtered with an adaptive loop filter (ALF) in step 56 to generate the reconstructed frame 57 which will be displayed and used as a reference frame for the following Inter frames.

In step 54 each pixel of the frame region is classified into a class or group. The same offset value is added to every pixel value which belongs to a certain class or group.

The derivation of the SAO parameters for the sample adaptive offset filtering in different embodiments of the present invention will be explained in more detail hereafter with reference to Figures 12 to 38.

**Figure 6** illustrates a block diagram of a decoder 60 which may be used to receive data from an encoder according an embodiment of the invention. The decoder is represented by connected modules, each module being adapted to implement, for example in the form of programming instructions to be executed by the CPU 311 of device 300, a corresponding step of a method implemented by the decoder 60.

The decoder 60 receives a bitstream 61 comprising encoding units, each one being composed of a header containing information on encoding parameters and a body containing the encoded video data. As explained with respect to Figure 4, the encoded video data is entropy encoded, and the motion vector predictors' indexes are encoded, for a given block, on a predetermined number of bits. The received encoded video data is entropy decoded by module 62. The residual data are then dequantized by module 63 and then an inverse transform is performed by module 64 to obtain pixel values.

The mode data indicating the coding mode are also entropy decoded and based on the mode, an INTRA type decoding or an INTER type decoding is performed on the encoded blocks of image data.

In the case of INTRA mode, an INTRA predictor is determined by intra prediction module 65 based on the intra prediction mode specified in the bitstream.

If the mode is INTER, the motion prediction information is extracted from the bitstream so as to find the reference area used by the encoder. The motion prediction information is composed of the reference frame index and the motion vector residual. The motion vector predictor is added to the motion vector residual in order to obtain the motion vector by motion vector decoding module 70.

Motion vector decoding module 70 applies motion vector decoding for each current block encoded by motion prediction. Once an index of the motion vector predictor, for the current block has been obtained the actual value of the motion vector associated with the current block can be decoded and used to perform motion compensation by module 66. The reference image portion indicated by the decoded motion vector is extracted from a reference image 68 to perform the motion

compensation 66. The motion vector field data 71 is updated with the decoded motion vector in order to be used for the prediction of subsequent decoded motion vectors.

Finally, a decoded block is obtained. Post filtering is applied by post filtering module 67 similarly to post filtering module 815 applied at the encoder as described with reference to Figure 5. A decoded video signal 69 is finally provided by the decoder 60.

The aim of SAO filtering is to improve the quality of the reconstructed frame by sending additional data in the bitstream in contrast to the deblocking filter where no information is transmitted. As mentioned above, each pixel is classified into a predetermined class or group and the same offset value is added to every pixel sample of the same class/group. One offset is encoded in the bitstream for each class. SAO loop filtering has two SAO types: an Edge Offset (EO) type and a Band Offset (BO) type. An example of Edge Offset type is schematically illustrated in Figures 7A and 7B, and an example of Band Offset type is schematically illustrated in Figure 8.

In HEVC, SAO filtering is applied CTU by CTU. In this case the parameters needed to perform the SAO filtering (set of SAO parameters) are selected for each CTU at the encoder side and the necessary parameters are decoded and/or derived for each CTU at the decoder side. This offers the possibility of easily encoding and decoding the video sequence by processing each CTU at once without introducing delays in the processing of the whole frame. Moreover, when SAO filtering is enabled, only one SAO type is used: either the Edge Offset type filter or the Band Offset type filter according to the related parameters transmitted in the bitstream for each classification. One of the SAO parameters in HEVC is an SAO type parameter `sao_type_idx` which indicates for the CTU whether EO type, BO type or no SAO filtering is selected for the CTU concerned.

The SAO parameters for a given CTU can be copied from the upper or left CTU, for example, instead of transmitting all the SAO data. One of the SAO parameters in HEVC is a `sao_merge_up` flag, which when set indicates that the SAO parameters (other than the `sao_merge_up` flag) for the subject CTU should be copied from the upper CTU. Another of the SAO parameters in HEVC is a `sao_merge_left` flag, which when set indicates that the SAO parameters for the subject CTU should be copied from the left CTU.

SAO filtering may be applied independently for different color components (e.g. YUV) of the frame. For example, one set of SAO parameters may be provided for the luma component Y and another set of SAO parameters may be provided for both chroma components U and V in common. Also, within the set of SAO parameters one or more SAO parameters may be used as common filtering parameters for two or more

color components, while other SAO parameters are dedicated (per-component) filtering parameters for the color components. For example, in HEVC, the SAO type parameter `sao_type_idx` is common to U and V, and so is a EO class parameter which indicates a class for EO filtering (see below), whereas a BO class parameter which indicates a group of classes for BO filtering has dedicated (per-component) SAO parameters for U and V.

As described in Figure 10, when the SAO type is an Edge type (603), the sign of the offset is implicit as shown in Figure 7B. So the sign is not read (610). This implicit sign determination was adopted during HEVC standardization because, the sign signaling doesn't improve the coding efficiency of SAO. In the current VTM software, if the implicit signs are replaced by explicit signs, as in the initial version of HEVC, a decrease of coding efficiency is observed.

A description of the Edge Offset type in HEVC is now provided with reference to **Figures 7A** and **7B**.

Edge Offset type involves determining an edge index for each pixel by comparing its pixel value to the values of two neighboring pixels. Moreover, these two neighboring pixels depend on a parameter which indicates the direction of these two neighboring pixels with respect to the current pixel. These directions are the 0-degree (horizontal direction), 45-degree (diagonal direction), 90-degree (vertical direction) and 135-degree (second diagonal direction). These four directions are schematically illustrated in Figure 7A.

The table of Figure 7B gives the offset value to be applied to the pixel value of a particular pixel "C" according to the value of the two neighboring pixels  $C_{n1}$  and  $C_{n2}$  at the decoder side.

When the value of C is less than the two values of neighboring pixels  $C_{n1}$  and  $C_{n2}$ , the offset to be added to the pixel value of the pixel C is "+ O1". When the pixel value of C is less than one pixel value of its neighboring pixels (either  $C_{n1}$  or  $C_{n2}$ ) and C is equal to one value of its neighbors, the offset to be added to this pixel sample value is "+ O2".

When the pixel value of c is less than one of the pixel values of its neighbors ( $C_{n1}$  or  $C_{n2}$ ) and the pixel value of C is equal to one value of its neighbors, the offset to be applied to this pixel sample is "- O3". When the value of C is greater than the two values of  $C_{n1}$  or  $C_{n2}$ , the offset to be applied to this pixel sample is "- O4".

When none of the above conditions is met on the current sample and its neighbors, no offset value is added to the current pixel C as depicted by the Edge Index value "2" of the table.

It is important to note that for the particular case of the Edge Offset type, the absolute value of each offset (O1, O2, O3, O4) is encoded in the bitstream. The sign to be applied to each offset depends on the edge index (or the Edge Index in the HEVC specifications) to which the current pixel belongs. According to the table represented in  
 5 Figure 7B, for Edge Index 0 and for Edge Index 1 (O1, O2) a positive offset is applied. For Edge Index 3 and Edge Index 4 (O3, O4), a negative offset is applied to the current pixel.

In the HEVC specifications, the direction for the Edge Offset amongst the four directions of Figure 7A is specified in the bitstream by a "sao\_eo\_class\_luma" field  
 10 for the luma component and a "sao\_eo\_class\_chroma" field for both chroma components U and V.

The SAO Edge Index corresponding to the index value is obtained by the following formula:  $\text{EdgeIndex} = \text{sign}(C - C_{n2}) - \text{sign}(C_{n1} - C) + 2$

where the definition of the function  $\text{sign}(\cdot)$  is given by the following  
 15 relationships

$$\text{sign}(x) = 1, \text{ when } x > 0$$

$$\text{sign}(x) = -1, \text{ when } x < 0$$

$$\text{sign}(x) = 0, \text{ when } x = 0.$$

In order to simplify the Edge Offset determination for each pixel, the difference between the pixel value of C and the pixel value of both its neighboring pixels C<sub>n1</sub> and C<sub>n2</sub> can be shared for current pixel C and its neighbors. Indeed, when SAO Edge Offset filtering is applied using a raster scan order of pixels of the current CTU or frame, the term  $\text{sign}(C_{n1} - C)$  has already computed for the previous pixels (to be precise  
 20 it was computed as  $C' - C_{n2}'$  at a time when the current pixel C' at that time was the present neighboring pixel C<sub>n1</sub> and the neighboring pixel C<sub>n2</sub>' was what is now the current pixel C). As a consequence this  $\text{sign}(c_{n1} - c)$  does not need to be computed again.

A description of the Band Offset type is now provided with reference to Figure  
 30 8.

Band Offset type in SAO also depends on the pixel value of the sample to be processed. A class in SAO Band offset is defined as a range of pixel values. Conventionally, for all pixels within a range, the same offset is added to the pixel value. In the HEVC specifications, the number of offsets for the Band Offset filter is four for

each reconstructed block or frame area of pixels (CTU), as schematically illustrated in Figure 8.

One implementation of SAO Band offset splits the full range of pixel values into 32 ranges of the same size. These 32 ranges are the bands (or classes) of SAO Band offset. The minimum value of the range of pixel values is systematically 0 and the maximum value depends on the bit depth of the pixel values according to the following relationship  $\text{Max} = 2^{\text{Bitdepth}} - 1$ . Classifying the pixels into 32 ranges of the full interval includes 5 bits checking needed to classify the pixels values for fast implementation i.e. only the 5 first bits (5 most significant bits) are checked to classify a pixel into one of the 32 classes/ ranges of the full range.

For example, when the bitdepth is 8 bits per pixel, the maximum value of a pixel can be 255. Hence, the range of pixel values is between 0 and 255. For this bitdepth of 8 bits, each band or class contains 8 pixel values.

According to the HEVC specifications, a group 40 of bands, represented by the grey area (40), is used, the group having four successive bands 41, 42, 43 and 44, and information is signaled in the bitstream to identify the position of the group, for example the position of the first of the 4 bands. The syntax element representative of this position is the "sao\_band\_position" field in the HEVC specifications. This corresponds to the start of band 41 in Figure 8. According to the HEVC specifications, 4 offsets corresponding respectively to the 4 bands are signaled in the bitstream.

**Figure 9** is a flow chart illustrating the steps of a process to decode SAO parameters according to the HEVC specifications. The process of Figure 9 is applied for each CTU to generate a set of SAO parameters for all components. In order to avoid encoding one set of SAO parameters per CTU (which is very costly), a predictive scheme is used for the CTU mode. This predictive mode involves checking if the CTU on the left of the current CTU uses the same SAO parameters (this is specified in the bitstream through a flag named "sao\_merge\_left\_flag"). If not, a second check is performed with the CTU above the current CTU (this is specified in the bitstream through a flag named "sao\_merge\_up\_flag"). This predictive technique enables the amount of data representing the SAO parameters for the CTU mode to be reduced. Steps of the process are set out below.

In step 503, the "sao\_merge\_left\_flag" is read from the bitstream 502 and decoded. If its value is true, then the process proceeds to step 504 where the SAO parameters of left CTU are copied for the current CTU. This enables the types for YUV of the SAO filter for the current CTU to be determined in step 508.

If the outcome is negative in step 503 then the "sao\_merge\_up\_flag" is read from the bitstream and decoded. If its value is true, then the process proceeds to step 505 where the SAO parameters of the above CTU are copied for the current CTU. This enables the types of the SAO filter for the current CTU to be determined in step 508.

5 If the outcome is negative in step 505, then the SAO parameters for the current CTU are read and decoded from the bitstream in step 507 for the Luma Y component and both U and V components (501) (551) for the type. The offsets for Chroma are independent.

The details of this step are described later with reference to Figure 10. After  
10 this step, the parameters are obtained and the type of SAO filter is determined in step 508.

In subsequent step 511 a check is performed to determine if the three colour components (Y and U & V) for the current CTU have been processed. If the outcome is positive, the determination of the SAO parameters for the three components is complete  
15 and the next CTU can be processed in step 510. Otherwise, (Only Y was processed) U and V are processed together and the process restarts from initial step 512 previously described.

**Figure 10** is a flow chart illustrating steps of a process of parsing of SAO parameters in the bitstream 601 at the decoder side. In an initial step 602, the  
20 "sao\_type\_idx\_X" syntax element is read and decoded. The code word representing this syntax element can use a fixed length code or could use any method of arithmetic coding. The syntax element sao\_type\_idx\_X enables determination of the type of SAO applied for the frame area to be processed for the colour component Y or for both Chroma components U & V. For example, for a YUV 4:2:0 sequence, two components are  
25 considered: one for Y, and one for U and V. The "sao\_type\_idx\_X" can take 3 values as follows depending on the SAO type encoded in the bitstream. '0' corresponds to no SAO, '1' corresponds to the Band Offset case illustrated in Figure 8 and '2' corresponds to the Edge Offset type filter illustrated in Figures 3A and 3B.

Incidentally, although YUV color components are used in HEVC (sometimes  
30 called Y, Cr and Cb components), it will be appreciated that in other video coding schemes other color components may be used, for example RGB color components. The techniques of the present invention are not limited to use with YUV color components and can be used with RGB color components or any other color components.

In the same step 602, a test is performed to determine if the “sao\_type\_idx\_X” is strictly positive. If “sao\_type\_idx\_X” is equal to “0” signifying that there is no SAO for this frame area (CTU) for Y if X is set equal to Y and that there is no SAO for this frame area for U and V if X is set equal to U and V. The determination of the SAO parameters is complete and the process proceeds to step 608. Otherwise if the  
5 “sao\_type\_idx” is strictly positive, this signifies that SAO parameters exist for this CTU in the bitstream.

Then the process proceeds to step 606 where a loop is performed for four iterations. The four iterations are carried in step 607 where the absolute value of offset j  
10 is read and decoded from the bitstream. These four offsets correspond either to the four absolute values of the offsets (O1, O2, O3, O4) of the four Edge indexes of SAO Edge Offset (see Figure 7B) or to the four absolute values of the offsets related to the four ranges of the SAO band Offset (see Figure 8).

Note that for the coding of an SAO offset, a first part is transmitted in the  
15 bitstream corresponding to the absolute value of the offset. This absolute value is coded with a unary code. The maximum value for an absolute value is given by the following formula:

$$\text{MAX\_abs\_SAO\_offset\_value} = (1 \ll (\text{Min}(\text{bitDepth}, 10) - 5)) - 1$$

where  $\ll$  is the left (bit) shift operator.

20 This formula means that the maximum absolute value of an offset is 7 for a pixel value bitdepth of 8 bits, and 31 for a pixel value bitdepth of 10 bits and beyond.

The current HEVC standard amendment addressing extended bitdepth video sequences provides similar formula for a pixel value having a bitdepth of 12 bits and beyond. The absolute value decoded may be a quantized value which is dequantized  
25 before it is applied to pixel values at the decoder for SAO filtering. An indication of use or not of this quantification is transmitted in the slice header.

For Edge Offset type, only the absolute value is transmitted because the sign can be inferred as explained previously.

For Band Offset type, the sign is signaled in the bitstream as a second part  
30 of the offset if the absolute value of the offset is not equal to 0. The bit of the sign is bypassed when CABAC is used.

After step 607, the process proceeds to step 603 where a test is performed to determine if the type of SAO corresponds to the Band Offset type (sao\_type\_idx\_X == 1).

If the outcome is positive, the signs of the offsets for the Band Offset mode are decoded in steps 609 and 610, except for each offset that has a zero value, before the following step 604 is performed in order to read in the bitstream and to decode the position “sao\_band\_position\_X” of the SAO band as illustrated in Figure 8.

5 If the outcome is negative in step 603 (“sao\_type\_idx\_X” is set equal to 2), this signifies that the Edge Offset type is used. Consequently, the Edge Offset class (corresponding to the direction 0, 45, 90 and 135 degrees) is extracted from the bitstream 601 in step 605. If X is equal to Y, the read syntax element is “sao\_eo\_class\_luma” and if X is set equal to U and V, the read syntax element is “sao\_eo\_class\_chroma”.

10 When the four offsets have been decoded, the reading of the SAO parameters is complete and the process proceeds to step 608.

**Figure 11** is a flow chart illustrating how SAO filtering is performed on an image part according to the HEVC specifications, for example during the step 67 in Figure 6. In HEVC, this image part is a CTU. This same process 700 is also applied in  
15 the decoding loop (step 415 in Figure 4) at the encoder in order to produce the reference frames used for the motion estimation and compensation of the following frames. This process is related to the SAO filtering for one color component (thus suffix “\_X” in the syntax elements has been omitted below).

An initial step 701 comprises determining the SAO filtering parameters  
20 according to processes depicted in Figures 9 and 10. The SAO filtering parameters are determined by the encoder and the encoded SAO parameters are included in the bitstream. Accordingly, on the decoder side in step 701 the decoder reads and decodes the parameters from the bitstream. Step 701 obtains the sao\_type\_idx and if it equals 1 also obtains the sao\_band\_position 702 and if it equals 2 also obtains the  
25 sao\_eo\_class\_luma or sao\_eo\_class\_chroma (according to the color component processed). If the element sao\_type\_idx is equal to 0 the SAO filtering is not applied. Step 701 obtains also an offsets table 703 of the 4 offsets.

A variable  $i$ , used to successively consider each pixel  $P_i$  of the current block or frame area (CTU), is set to 0 in step 704. Incidentally, “frame area” and “image area”  
30 are used interchangeably in the present specification. A frame area in this example is a CTU in the HEVC standard. In step 706, pixel  $P_i$  is extracted from the frame area 705 which contains N pixels. This pixel  $P_i$  is classified in step 707 according to the Edge offset classification described with reference to **Figures 7A & 7B** or Band offset

classification as described with reference to **Figure 8**. The decision module 708 tests if  $P_i$  is in a class that is to be filtered using the conventional SAO filtering.

If  $P_i$  is in a filtered class, the related class number  $j$  is identified and the related offset value  $Offset_j$  is extracted in step 710 from the offsets table 703. In the case of the conventional SAO filtering this  $Offset_j$  is then added to the pixel value  $P_i$  in step 711 in order to produce the filtered pixel value  $P'_i$  712. This filtered pixel  $P'_i$  is inserted in step 713 into the filtered frame area 716.

If  $P_i$  is not in a class to be SAO filtered then  $P_i$  (709) is inserted in step 713 into the filtered frame area 716 without filtering.

After step 713, the variable  $i$  is incremented in step 714 in order to filter the subsequent pixels of the current frame area 705 (if any – test 715). After all the pixels have been processed ( $i \geq N$ ) in step 715, the filtered frame area 716 is reconstructed and can be added to the SAO reconstructed frame (see frame 68 of **Figure 6** or 416 of **Figure 4**).

As noted above, the JVET exploration model (JEM) for the future VVC standard uses all the HEVC tools. One of these tools is sample adaptive offset (SAO) filtering. However, SAO is less efficient in the JEM reference software than in the HEVC reference software. This arises from fewer evaluations and from signalling inefficiencies compared to other loop filters.

Embodiments of the present invention described below are intended to improve the coding efficiency of SAO by using various techniques for deriving one or more SAO parameters of an image part in a current image from one or more SAO parameters of a collocated image part in a reference image. These techniques may be referred to as temporal derivation techniques for SAO parameters. Further embodiments described below are intended to improve the coding efficiency of SAO by using various techniques for deriving one or more SAO parameters of an image part in an image from one or more SAO parameters of another image part of the same image. These techniques may be referred to as spatial derivation techniques for SAO parameters.

**Figure 12** is a flow chart illustrating steps carried out an encoder to determine SAO parameters for the CTUs of a group (frame or slice). The process starts with a current CTU (1101). First the statistics for all possible SAO types and classes are accumulated in the variable CTUStats (1102). The process of Step 1102 is described below with reference to Figure 13. According to the value set in the variable CTUStats,

the RD cost for the SAO merge Left is evaluated if the Left CTU is in the current Slice (1103) as the RD cost of the SAO Merge UP (1104). Thanks to the statistics in CTUStats (1102), new SAO parameters are evaluated for Luma (1105) and for both Chroma components (1109). (Both Chroma components because the Chroma components share the same SAO type in the HEVC standard). For each SAO type (1106), the best RD offsets and other parameters for Band offset classification are obtained (1107). Steps 1107 and 1110 are explained below for Edge and Band classification with reference to Figure 14 and Figure 15 respectively. All RD costs are computed thanks to their respective SAO parameters (1108). In the same way for both Chroma components, the optimal RD offsets and parameters are selected (1111). All this RD costs are compared in order to select the best SAO parameters set (1115). These RD costs are also compared to disable SAO independently for the Luma and the Chroma components (1113, 1114). The use of a new SAO parameters set (1115) is compared to the SAO parameters set "Merging" or sharing (1116) from the left and up CTU.

**Figure 13** is a flow chart illustrating steps of an example of a statistics computed at the encoder side that can be applied for the Edge Offset type filter, in the case of the conventional SAO filtering. The similar approach may also be used for the Band Offset type filter.

**Figure 13** illustrates the setting of the variable CTUStats containing all information needed to derive each best rate distortion offsets for each class. Moreover, it illustrates the selection of the best SAO parameters set for the current CTU. For each colour component Y, U, V (or RGB) (811) each SAO type is evaluated. For each SAO type (812) the variables  $Sum_j$  and  $SumNbPix_j$  are set to zero in an initial step 801. The current frame area 803 contains N pixels.

$j$  is the current range number to determine the four offsets (related to the four edge indexes shown in **Figure 7B** for Edge Offset type or to the 32 ranges of pixel values shown in **Figure 8** for Band Offset type).  $Sum_j$  is the sum of the differences between the pixels in the range  $j$  and their original pixels.  $SumNbPix_j$  is the number of pixels in the frame area, the pixel value of which belongs to the range  $j$ .

In step 802, a variable  $i$ , used to successively consider each pixel  $P_i$  of the current frame area, is set to zero. Then, the first pixel  $P_i$  of the frame area 803 is extracted in step 804. In step 805, the class of the current pixel is determined by checking the conditions defined in **Figure 7B**. Then a test is performed in step 805. During step

805, a check is performed to determine if the class of the pixel value  $P_i$  corresponds to the value “none of the above” of **Figure 7B**.

If the outcome is positive, then the value “i” is incremented in step 808 in order to consider the next pixels of the frame area 803.

5                   Otherwise, if the outcome is negative in step 806, the next step is 807 where the related  $SumNbPix_j$  (i.e. the sum of the number of pixels for the class determined in step 805) is incremented and the difference between  $P_i$  and its original value  $P_i^{org}$  is added to  $Sum_j$ . In the next step 808, the variable  $i$  is incremented in order to consider the next pixels of the frame area 803.

10                   Then a test is performed to determine if all pixels have been considered and classified. If the outcome is negative, the process loops back to step 804 described above. Otherwise, if the outcome is positive, the process proceeds to step 810 where the variable CTUStats for the current colour component X and the SAO type SAO\_type and the current class  $j$  are set equal to  $Sum_j$  for the first value and  $SumNbPix_j$  for the  
 15 second value. These variables can be used to compute for example the optimal offset parameter  $Offset_j$  of each class  $j$ . This offset  $Offset_j$  may be the average of the differences between the pixels of class  $j$  and their original values. Thus,  $Offset_j$  is given by the following formula:

$$Offset_j = \frac{Sum_j}{SumNbPix_j}$$

20                   Note that the offset  $Offset_j$  is an integer value. As a consequence, the ratio defined in this formula may be rounded, either to the closest value or using the ceiling or floor function.

Each offset  $Offset_j$  is an optimal offset  $Oopt_j$  in terms of distortion.

To evaluate an RD cost for a merge of SAO parameters, the encoder uses  
 25 the statistics set in table CTUStats. According to the following examples for the SAO Merge Left and by considering the type for Luma Left\_Type\_Y and the four related offsets  $O\_Left\_0$ ,  $O\_Left\_1$ ,  $O\_Left\_2$ ,  $O\_Left\_3$ , the distortion can be obtained by the following formula:

Distortion\_Left\_Y =  
 30                   (CTUStats[Y][ Left\_Type\_Y][0][1] x  $O\_Left\_0$  x  $O\_Left\_0$  - CTUStats[Y][ Left\_Type\_Y][0][0] x  $O\_Left\_0$  x 2)>> Shift

```

+ (CTUStats[Y][ Left_Type_Y][1][1] x O_Left_1 x O_Left_1 - CTUStats[Y][
Left_Type_Y][1][0] x O_Left_1 x 2)>> Shift
+ (CTUStats[Y][ Left_Type_Y][2][1] x O_Left_2 x O_Left_2 - CTUStats[Y][
Left_Type_Y][2][0] x O_Left_2 x 2)>> Shift
5 + (CTUStats[Y][ Left_Type_Y][3][1] x O_Left_3 x O_Left_3 - CTUStats[Y][
Left_Type_Y][3][0] x O_Left_3 x 2)>> Shift

```

The variable Shift is designed for a distortion adjustment. The distortion should be negative as SAO is a post filtering.

The same computing is applied for Chroma components. The Lambda of the Rate distortion cost is fixed for the three components. For an SAO parameters merged with the left CTU, the rate is only 1 flag which is CABAC coded.

The encoding process illustrated in **Figure 14** is applied in order to find the best offset in terms of rate distortion criterion, offset referred to as ORD<sub>j</sub>. This process is applied in steps 1109 to 1112.

In an initial step 901 of the encoding process of Figure 14, the rate distortion value J<sub>j</sub> is initialized to the maximum possible value. Then a loop on O<sub>j</sub> from O<sub>optj</sub> to 0 is applied in step 902. Note that O<sub>j</sub> is modified by 1 at each new iteration of the loop. If O<sub>optj</sub> is negative, the value O<sub>j</sub> is incremented and if O<sub>optj</sub> is positive, the value O<sub>j</sub> is decremented. The rate distortion cost related to O<sub>j</sub> is computed in step 903 according to the following formula:

$$J(O_j) = \text{SumNbPix}_j \times O_j \times O_j - \text{Sum}_j \times O_j \times 2 + \lambda R(O_j)$$

where λ is the Lagrange parameter and R(O<sub>j</sub>) is a function which provides the number of bits needed for the code word associated with O<sub>j</sub>.

Formula ‘SumNbPix<sub>j</sub> x O<sub>j</sub> x O<sub>j</sub> - Sum<sub>j</sub> x O<sub>j</sub> x 2’ gives the improvement in terms of the distortion provided by the use of the offset O<sub>j</sub>. If J(O<sub>j</sub>) is inferior to J<sub>j</sub> then J<sub>j</sub> = J(O<sub>j</sub>) and ORD<sub>j</sub> is equal to O<sub>j</sub> in step 904. If O<sub>j</sub> is equal to 0 in step 905, the loop ends and the best ORD<sub>j</sub> for the class j is selected.

This algorithm of Figures 18 and 19 provides a best ORD<sub>j</sub> for each class j. This algorithm is repeated for each of the four directions of Figure 7A. Then the direction that provides the best rate distortion cost (sum of J<sub>j</sub> for each direction) is selected as the direction to be used for the current CTU.

This algorithm (Figures 18 and 19) for selecting the offset values at the encoder side for the Edge offset tool can be easily applied to the Band Offset filter to select the best position (SAO\_band\_position) where j is in the interval [0,32] instead of the interval [1,4] in Figure 13. It involves changing the value 4 to 32 in modules 801, 810,

811. More specifically, for the 32 classes of Figure 8, the parameter Sumj (j=[0,32]) is computed. This corresponds to computing for each range j, the difference between the current pixel value (Pi) and its original value (Porgi), each pixel of the image belonging to a single range j. Then the best offset in terms of rate distortion ORDj is computed for the 32 classes, with the same process as described in Figure 14.

The next step involves finding the best position of the SAO band position of Figure 8. This is determined with the encoding process set out in **Figure 15**. The RD cost Jj for each range has been computed with the encoding process of Figure 14 with the optimal offset ORDj in terms of rate distortion. In Figure 15, in an initial step 1001 the rate distortion value J is initialized to the maximum possible value. Then a loop on the 28 positions j of 4 consecutive classes is run in step 1002. Next, the variable Jj corresponding to the RD cost of the band (of 4 consecutive classes) is initialized to 0 in step 1003. Then the loop on the four consecutive offset j is run in step 1004. Ji is incremented by the RD costs of the four classes Jj in step 1005 (j=i to i+4).

If this cost Ji is inferior to the Best RD cost J, J is set to Ji, and sao\_band\_position = i in step 1007, and the next step is step 1008.

Otherwise, the next step is step 1008.

In the document JVET-J0024, it is proposed to modify the SAO edge classification. The following table illustrates this new proposed SAO edge classification.

EdgeIndex	Condition
0	$c < a - 3 \ \&\& \ c < b - 3$
1	$( c < a - 3 \ \&\& \  c-b  \leq 3 ) \    \ (  c - a  \leq 3 \ \&\& \ c < b - 3 )$
3	$( c > a + 3 \ \&\& \  c-b  \leq 3 ) \    \ (  c - a  \leq 3 \ \&\& \ c > b + 3 )$
4	$c > a + 3 \ \&\& \ c > b + 3$
2	None of the above

In this table, c is the value of the sample c in Figure 7A and Figure 7B, a is the value of the “left” sample corresponding to Cn1 in Figure 7A and Figure 7B and b is the value of the “right” sample corresponding to Cn1 in Figure 7A and Figure 7B. The symbol && is the operator AND. The symbol || is the operator OR.

.If this proposed modification is compared with the SAO edge offset classification of HEVC provided in the following table, the modification is only a modification of the sign function computation.

EdgeIndex	Condition
0	$c < a \ \&\& \ c < b$
1	$( c < a \ \&\& \ c=b ) \    \ ( c = a \ \&\& \ c < b )$



According to this first aspect, the encoder selects a sign method among a plurality of possible sign methods. The best one is signalled in the bitstream so that the decoder applies the related Edge offset classification with this selected sign method.

There are several possibilities to define a sign method. One of the possibilities may be to define a sign function “sign” which takes into account a number N corresponding to the limit value. This sign function may be given for example by the following formula:

$$\text{SignN}(\text{val}, N) = (\text{val} < -N ? -1 : (\text{val} > N ? 1 : 0))$$

where the definition of the function signN(..) may be given by the following relationships :

$$\begin{aligned} \text{signN}(\text{val}, N) &= -1 \text{ if "val" } < -N \\ \text{signN}(\text{val}, N) &= 0 \text{ if } -N \leq \text{"val"} \leq N \\ \text{signN}(\text{val}, N) &= +1 \text{ if "val" } > N \end{aligned}$$

In this case, the EdgeIndex of the SAO edge offset classification may be obtained according to the following formula:

$$\text{EdgeIndex} = \text{signN}(C - Cn2, N) - \text{signN}(C - Cn2, N) + 2$$

And the SAO edge classification may thus become:

EdgeIndex	Condition
0	$c < a - N \ \&\& \ c < b - N$
1	$(c < a - N \ \&\& \  c-b  \leq N) \    \ ( c - a  \leq N \ \&\& \ c < b - N)$
3	$(c > a + N \ \&\& \  c-b  \leq N) \    \ ( c - a  \leq N \ \&\& \ c > b + N)$
4	$c > a + N \ \&\& \ c > b + N$
2	None of the above

An advantage of this embodiment may be that coding efficiency is increased compared to the traditional HEVC Edge offset classification. Consequently, the first embodiment is more flexible than the prior art. The coding efficiency increase provided by this first embodiment comes from an additional flexibility which is adapted to the quantization error and video content.

The function signN is only an implementation example, and another possibility may be to keep the same sign function as defined for HEVC and to divide the input value "val" by the value N. In this case, the EdgeIndex of the SAO edge offset classification may be obtained according to the following formula:

$$\text{EdgeIndex} = \text{sign}((C - Cn2)/N) - \text{sign}((C - Cn2)/N) + 2$$

According to alternative embodiments, the skilled person can change this division by a multiplication and a shift, for instance as follows:

$$\text{EdgeIndex} = \text{sign}(((C - C_{n2}) * R) \gg L) - \text{sign}(((C - C_{n2}) * R) \gg L) + 2$$

where R and L are derived from the N value.

5 **Figure 16** is a flow chart illustrating how SAO filtering is performed on an image part according to a first embodiment of the first aspect of the present invention.

For example, the process 1600 may be performed during step 67 in Figure 6. It is also performed in the decoding loop (step 415 in Figure 4) at the encoder in order to produce the reference frames used for the motion estimation and compensation of the following frames. This process is related to the SAO filtering for one color component (thus suffix “\_X” in the syntax elements may have been omitted below).

In this first embodiment, the predetermined sign method is signalled in the bitstream at CTU level. Hence, the signaling is performed at CTU level, i.e. the image part to which the process 1600 is applied is a CTU.

15 Compared to the process shown in Figure 11, the present process comprises two additional modules 1617 and 1618 which are described below.

An initial step 1601 comprises determining the SAO filtering parameters according to processes depicted in Figures 9 and 10. The SAO filtering parameters are determined by the encoder and the encoded SAO parameters are included in the bitstream. Accordingly, on the decoder side in step 1601, the decoder reads and decodes the parameters from the bitstream. Step 1601 obtains the `sao_type_idx` and if it equals 1 also obtains the `sao_band_position` 1602 and if it equals 2 also obtains the `sao_eo_class_luma` or `sao_eo_class_chroma` (according to the color component processed). If the element `sao_type_idx` is equal to 0 the SAO filtering is not applied.

25 At step 1601, the decoder also checks whether the SAO type is Edge Offset (EO). According to the first embodiment, a flag is inserted inside the bitstream for each SAO edge offset at CTU level. For this purpose, it is defined an index `eo_sign_method` representing the value N which will be used as the second parameter of the `signN` function for the Edge classification of current CTB samples. Next, the `eo_sign_method` is identified between `eo_sign_method_Luma` and `eo_sign_method_Chroma` 1618. As for other Edge offset parameters, the `eo_sign_method_Luma` may be independent from the `eo_sign_method_Chroma`. Step 1601 obtains also an offsets table 1603 of the 4 offsets.

35 A variable `i`, used to successively consider each pixel  $P_i$  of the current block or frame area (CTU), is set to 0 in step 1604. Incidentally, “frame area” and “image area”

are used interchangeably in the present specification. A frame area in this example is a CTU. In step 1606, pixel  $P_i$  is extracted from the frame area 1605 which contains  $N$  pixels. This pixel  $P_i$  is classified in step 1607 according to the Edge offset classification described with reference to Figure 27A or 27B or Band offset classification as described with reference to Figure 8. The decision module 1608 tests if  $P_i$  is in a class that is to be filtered using the conventional SAO filtering.

If  $P_i$  is in a filtered class, the related class number  $j$  is identified and the related offset value  $Offset_j$  is extracted in step 1610 from the offsets table 1603. In the case of the conventional SAO filtering this  $Offset_j$  is then added to the pixel value  $P_i$  in step 1611 in order to produce the filtered pixel value  $P_i'$  1612. This filtered pixel  $P_i'$  is inserted in step 1613 into the filtered frame area 1616.

If  $P_i$  is not in a class to be SAO filtered then  $P_i$  (1609) is inserted in step 1613 into the filtered frame area 1616 without filtering.

After step 1613, the variable  $i$  is incremented in step 1614 in order to filter the subsequent pixels of the current frame area 1605 (if any – test 1615). After all the pixels have been processed ( $i \geq N$ ) in step 1615, the filtered frame area 1616 is reconstructed and can be added to the SAO reconstructed frame (see frame 68 of Figure 6 or 416 of Figure 4).

**Figure 17** illustrates steps of a process of parsing of SAO parameters in the bitstream 1701 at the decoder side according to the first embodiment shown in Figure 16.

Compared to the process shown in Figure 10, the present process comprises a new module 1711 which is described below.

In an initial step 1702, the "sao\_type\_idx\_X" syntax element is read and decoded. The code word representing this syntax element can use a fixed length code or could use any method of arithmetic coding. The syntax element saotype\_idx\_X enables determination of the type of SAO applied for the frame area to be processed for the colour component Y or for both Chroma components U & V. The techniques of the present invention are not limited to use with YUV color components and can be used with RGB color components or any other color components.

In the same step 1702, a test is performed to determine if the "sao\_type\_idx\_X" is strictly positive. If "sao\_type\_idx\_X" is equal to "0" signifying that there is no SAO for this frame area (CTU) for Y if X is set equal to Y and that there is no SAO for this frame area for U and V if X is set equal to U and V. The determination of the SAO parameters is complete and the process proceeds to step 1708. Otherwise if

the "sao\_type\_idx" is strictly positive, this signifies that SAO parameters exist for this CTU in the bitstream.

Then the process proceeds to step 1706 where a loop is performed for four iterations. The four iterations are carried in step 1707 where the absolute value of offset  
5 j is read and decoded from the bitstream. These four offsets correspond either to the four absolute values of the offsets (O1, O2, O3, O4) of the four Edge indexes of SAO Edge Offset (see Figure 27A or 27B) or to the four absolute values of the offsets related to the four ranges of the SAO band Offset (see Figure 8).

After step 1707, the process proceeds to step 1703 where a test is performed  
10 to determine if the type of SAO corresponds to the Band Offset type (sao\_type\_idx\_X == 1).

If the outcome is positive, the signs of the offsets for the Band Offset mode are decoded in steps 1709 and 1710, except for each offset that has a zero value, before the following step 1704 is performed in order to read in the bitstream and to decode the  
15 position "sao\_band\_position\_X" of the SAO band as illustrated in Figure 8.

If the outcome is negative in step 1703 ("sao\_type\_idx\_X" is set equal to 2), this signifies that the Edge Offset type is used. According to the present embodiment, the decoder in this case reads the eo\_sign\_method\_X from the bitstream 1701 in step 1711. If X is equal to Y, the read syntax element is "eo\_sign\_method\_luma" and if X is  
20 set equal to U and V, the read syntax element is "eo\_sign\_method\_chroma".

It may be noted that, in this example, the eo\_sign\_method\_X is for both Chroma components (U and V) as some other parameters shared between both Chroma components. But an individual eo\_sign\_method syntax element may also be considered especially if the input video signal type is RGB.

25 The eo\_sign\_method syntax element may be encoded with a fixed length code representing the number of possible values for the parameter N. It has been determined that 4 possible values for N may provide an interesting compromise between encoding complexity and coding efficiency. Hence, this syntax element may be coded with 2 bits.

30 Next, the Edge Offset class (corresponding to the direction 0, 45, 90 and 135 degrees) is extracted from the bitstream 1701 in step 1705. If X is equal to Y, the read syntax element is "sao\_eo\_class\_luma" and if X is set equal to U and V, the read syntax element is "sao\_eo\_class\_chroma".

When the four offsets have been decoded, the reading of the SAO  
35 parameters is complete and the process proceeds to step 1708.

An advantage of the signalization of the edge sign method at CTU level is an efficient adaptation according to the frame/slice content and quantization error which produces a coding efficiency improvement compared to the prior art.

According to a possible implementation, the predetermined sign method is  
5 CABAC coded, i.e the flag `eo_sign_method_X` may be arithmetically coded with the traditional CABAC. An advantage of this implementation may be that the coding efficiency is increased while the frame/slice content and quantization error are relatively homogeneous inside the current slice/frame. As a result, one Edge offset sign method may be selected for many CTUs.

10 **Figure 18** illustrates steps of a process of parsing of SAO parameters in the bitstream 1801 at the decoder side according to a second embodiment.

According to the second embodiment, the predetermined sign method is predicted. More specifically, the Edge offset sign method syntax element `eo_sign_method_X` may be predicted from another sign method syntax element.

15 Compared to the process shown in Figure 17, the present process comprises two additional modules 1812 and 1813 which are described below.

In an initial step 1802, the "`sao_type_idx_X`" syntax element is read and decoded. The code word representing this syntax element can use a fixed length code or could use any method of arithmetic coding. The syntax element `sao_type_idx_X`  
20 enables determination of the type of SAO applied for the frame area to be processed for the colour component Y or for both Chroma components U & V. The techniques of the present invention are not limited to use with YUV color components and can be used with RGB color components or any other color components.

In the same step 1802, a test is performed to determine if the  
25 "`sao_type_idx_X`" is strictly positive. If "`sao_type_idx_X`" is equal to "0" signifying that there is no SAO for this frame area (CTU) for Y if X is set equal to Y and that there is no SAO for this frame area for U and V if X is set equal to U and V. The determination of the SAO parameters is complete and the process proceeds to step 1808. Otherwise if the "`sao_type_idx`" is strictly positive, this signifies that SAO parameters exist for this  
30 CTU in the bitstream.

Then the process proceeds to step 1806 where a loop is performed for four iterations. The four iterations are carried in step 1807 where the absolute value of offset `j` is read and decoded from the bitstream. These four offsets correspond either to the four absolute values of the offsets (`O1`, `O2`, `O3`, `O4`) of the four Edge indexes of SAO Edge

Offset (see Figure 27A or 27B) or to the four absolute values of the offsets related to the four ranges of the SAO band Offset (see Figure 8).

After step 1807, the process proceeds to step 1803 where a test is performed to determine if the type of SAO corresponds to the Band Offset type (`sao_type_idx_X == 1`).

If the outcome is positive, the signs of the offsets for the Band Offset mode are decoded in steps 1809 and 1810, except for each offset that has a zero value, before the following step 1804 is performed in order to read in the bitstream and to decode the position "`sao_band_position_X`" of the SAO band as illustrated in Figure 8.

If the outcome is negative in step 1803 ("`sao_type_idx_X`" is set equal to 2), this signifies that the Edge Offset type is used. The decoder in this case reads the `eo_sign_method_X` from the bitstream 1801 in step 1811. If X is equal to Y, the read syntax element is "`eo_sign_method_luma`" and if X is set equal to U and V, the read syntax element is "`eo_sign_method_chroma`".

Then, according to this second embodiment, a predictor sign method for the color components X 1812 is added to the decoded syntax element `eo_sign_method_X` at step 1813 in order to obtain the edge offset sign method for the current CTB.

Correspondingly, at encoder side, the `Predictor_sign_method_X` has been subtracted to the `eo_sign_method_X`.

In this second embodiment, the syntax element `eo_sign_method_X` is coded with a variable length code to obtain a benefice in term of coding efficiency. This variable length code could be a unary code or Golomb code and or another method. This `Predictor_sign_method_X` may be used as a context value for a CABAC coding.

According to an alternative embodiment, a flag may indicate if the current `eo_sign_method_X` is strictly equal to the `Predictor_sign_method_X` or not. If it is not the case, the `eo_sign_method_X` may be decoded. Otherwise, the `eo_sign_method_X` is equal to the `Predictor_sign_method_X`. The `eo_sign_method_X` may thus be a fix length code.

In one sub-embodiment, the `eo_sign_method_X` may be predicted by the `eo_sign_method_X` of the left CTU if the left CTU exists (for exemple if the left CTU as an SAO parameters and if the SAO type is an Edge offset type).

In one sub-embodiment, the `eo_sign_method_X` may be predicted by the `eo_sign_method_X` of the Up CTU if the Up CTU exists.

These two sub-embodiments have the advantage to be aligned with the SAO merges derivation which are currently the SAO parameters of the left and the up CTU.

In one sub-embodiment, the `eo_sign_method_X` may be predicted by the last `eo_sign_method_X` previously decoded if it exists.

In one sub-embodiment, the `eo_sign_method_X` may be predicted by the average of several `eo_sign_method_X` previously decoded if it exists.

5           After step 1813, the Edge Offset class (corresponding to the direction 0, 45, 90 and 135 degrees) is extracted from the bitstream 1801 in step 1805. If X is equal to Y, the read syntax element is “`sao_eo_class_luma`” and if X is set equal to U and V, the read syntax element is “`sao_eo_class_chroma`”.

10           When the four offsets have been decoded, the reading of the SAO parameters is complete and the process proceeds to step 1808.

An advantage of the second embodiment and all its sub-embodiments is that coding efficiency is increased due to the exploitation of the spatial correlation of the signal content and similar quantization error between CTU of the current slice/frame. This is because when the signal of the current slice is similar CTU by CTU and when the quantization error is also similar, the best selected `eo_sign_method_X` at encoder side are often similar.

**Figure 19** illustrates steps of a process of parsing of SAO parameters in the bitstream 1901 at the decoder side according to a third embodiment.

20           According to this third embodiment, the edge offset sign method is signalled at a higher level than at the CTU level. In other words, the predetermined sign method is signalled in the bitstream at frame, slice or sequence level.

More specifically, it is signalled for the slice/frame and exploits the potential correlation between edge offset sign methods used inside a slice/frame.

25           At step 1902, the `slice_sao_luma_flag` is extracted from the bitstream 1901. If the value of the flag is set to true (=1), the `eo_sign_method_luma` is extracted at step 1903 from the bitstream 1901 and will be used for the Edge offset classification sign method for all the luma CTB of the current slice/frame. The process then goes to the next slice header syntax element (step 1906).

30           If the value of the flag `slice_sao_luma_flag` is set to false, the `slice_sao_chroma_flag` is extracted at step 1904 from the bitstream 1901. If the value of the flag is set to true (=1), the `eo_sign_method_chroma` is extracted at step 1905 from the bitstream 1901 and will be used for the Edge offset classification sign method for all the chroma CTB of the current slice/frame. Then, the process goes to the next slice header syntax element (step 1906). If the value of the flag is set to false (=0), the process goes directly to the next slice header syntax element (step 1906).

35

An advantage of this embodiment may be that coding efficiency is improved when the frame/slice content and quantization error are relatively homogeneous inside the current slice/frame. A main advantage of the signaling at slice header is that the number of possible values for the `eo_sign_method_X` may be higher than for a CTU level signaling method described with reference to Figures 17 and 18. This is because the rate at slice header may have a lower impact than a syntax element at CTU level. Consequently, the `eo_sign_method_X` may be coded with 3 bits (8 possible values) or 4 bits (16 possible values).

According to an alternative embodiment, the `eo_sign_method_X` may be signaled at sequence level. In this case, the quantization error and signal content should be similar frame by frame and CTU by CTU.

According to an embodiment, the predetermined sign method is selected at the encoder side to minimize the rate-distortion cost. Therefore, the different possible values for each `eo_sign_method_X` are evaluated with the traditional RD criterion and the value which minimized the RD (rate-distortion) cost is selected and inserted in the bitstream.

An advantage of this embodiment may be that it should reach the maximum coding efficiency for the slice/frame level signaling.

**Figure 20** illustrates an example of selection of a `eo_sign_method_X` by an encoder. It represents a frame hierarchy based on the depth (encoder parameters only) and the quantization parameters at slice level. As depicted in this figure, the `eo_sign_method_X` value increases when the quantization parameter increases.

According to this embodiment, the predetermined sign method is selected at the encoder side according to parameters of a current slice or frame. In other words, the encoder may select the `eo_sign_method_X` according to one or some parameters of the current slice/frame. These parameters may be the depth in the frame hierarchy or the quantization parameters as shown in the figure.

According to an alternative embodiment, any `eo_sign_method_X` is transmitted, for instance once at the beginning of the sequence. In this case the `eo_sign_method_X` is selected at sequence level, or pre selected to define the codec in order to be optimal for each component or for luma component and both chroma components for all sequences.

**Figure 21** is a flow chart illustrating how SAO filtering is performed on an image part according to a fourth embodiment of the first aspect of the present invention.

For instance, this process 2100 is performed during step 67 in Figure 6. It is also performed in the decoding loop (step 415 in Figure 4) at the encoder in order to produce the reference frames used for the motion estimation and compensation of the following frames. This process is related to the SAO filtering for one color component (thus suffix “\_X” in the syntax elements has been omitted below).

In this fourth embodiment, the `eo_sign_method_X` obtained does not correspond directly to the value N used as parameters in the `signN` function. The table set is used to convert the indice value `eo_sign_method_X` to this value N. For example, the predetermined sign method is selected at the encoder side based on a plurality of limit values N that are not consecutive and/or may not include 0 as a possible value.

Compared to the process shown in Figure 16, the present process comprises two additional modules 2119 and 2120, which are described below.

An initial step 2101 comprises determining the SAO filtering parameters according to processes depicted in Figures 9 and 10. The SAO filtering parameters are determined by the encoder and the encoded SAO parameters are included in the bitstream. Accordingly, on the decoder side in step 2101, the decoder reads and decodes the parameters from the bitstream. Step 2101 obtains the `sao_type_idx` and if it equals 1 also obtains the `sao_band_position` 2102 and if it equals 2 also obtains the `sao_eo_class_luma` or `sao_eo_class_chroma` (according to the colour component processed). If the element `sao_type_idx` is equal to 0 the SAO filtering is not applied.

At step 2101, the decoder also checks whether the SAO type is Edge Offset (EO). A flag is inserted inside the bitstream for each SAO edge offset at CTU level. For this purpose, it is defined an index `eo_sign_method` representing the value N which will be used as the second parameter of the `signN` function for the Edge classification of current CTB samples. Next, the `eo_sign_method` is identified between `eo_sign_method_Luma` and `eo_sign_method_Chroma` 2118. As for other Edge offset parameters, the `eo_sign_method_Luma` may be independent from the `eo_sign_method_Chroma`. Step 2101 obtains also an offsets table 2103 of the 4 offsets.

Next, according to the present embodiment, a new syntax element M is defined at step 2119 based on the syntax element `eo_sign_method_luma` or `o_edge_sign_method_chroma` and a conversion table “Set”. The table set is used to convert the indice value `eo_sign_method_X` to the value M. Examples of such a conversion table are provided below. This value M will be used in the function “`signN(x, M)`” 2120 for the Edge offset classification of samples of the current CTB.

A variable  $i$ , used to successively consider each pixel  $P_i$  of the current block or frame area (CTU), is set to 0 in step 2104. Incidentally, “frame area” and “image area” are used interchangeably in the present specification. A frame area in this example is a CTU. In step 2106, pixel  $P_i$  is extracted from the frame area 2105, which contains  $N$  pixels. The pixel  $P_i$  is classified in step 2107 according to the Edge offset classification described with reference to Figures 27A and 27B or Band offset classification as described with reference to Figure 8. The decision module 2108 tests if  $P_i$  is in a class that is to be filtered using the conventional SAO filtering.

If  $P_i$  is in a filtered class, the related class number  $j$  is identified and the related offset value  $Offset_j$  is extracted in step 2110 from the offsets table 2103. In the case of the conventional SAO filtering this  $Offset_j$  is then added to the pixel value  $P_i$  in step 2111 in order to produce the filtered pixel value  $P_i'$  2112. This filtered pixel  $P_i'$  is inserted in step 2113 into the filtered frame area 2116.

If  $P_i$  is not in a class to be SAO filtered then  $P_i$  (2109) is inserted in step 2113 into the filtered frame area 2116 without filtering.

After step 2113, the variable  $i$  is incremented in step 2114 in order to filter the subsequent pixels of the current frame area 2105 (if any – test 2115). After all the pixels have been processed ( $i \geq N$ ) in step 2115, the filtered frame area 2116 is reconstructed and can be added to the SAO reconstructed frame (see frame 68 of Figure 6 or 416 of Figure 4).

For the sake of easiness, the following examples comprises 4 values in the set of possible  $N$ . The set of possible  $N$  are the values of the table set. However, the present invention is not limited thereto.

For example, the value  $N=0$  may be excluded, and the set of  $N$  values used may be  $\{1, 2, 3, 4\}$ . This may provide a better RD compromise.

Alternatively, the values of the set of possible  $N$  may be aligned with a shift value of the value  $Val$ . For instance, this set may be  $\{0, 2, 4, 8\}$ . It corresponds to the use of the classical sign method and right shift  $M$  of the input value  $val$  (with  $M = \{0, 1, 2, 3\}$ ).

For example, the Edge index may be obtained by the following formula:

$$EdgeIndex = \text{sign}((C - C_{n2}) \gg M) - \text{sign}((C - C_{n2}) \gg M) + 2$$

It should be noted that this formula does not always provide exactly the same result as when a  $N$  among the set  $\{0, 2, 4, 8\}$  is used with the following  $\text{sign}_N$  function:

$$EdgeIndex = \text{sign}_N(C - C_{n2}, N) - \text{sign}(C - C_{n2}, N) + 2$$

This is due to the right shift operator on negative value. But the coding efficiency of both methods is similar.

In one embodiment, the set of possible N values may be {1, 2, 4, 8}. In this case, the Edge Index can be obtained with the following formula:

$$\text{EdgeIndex} = \text{sign}(C - C_{n-2}, 1 < n < (\text{eo\_sign\_method\_X})) - \text{sign}(C - C_{n-2}, 1 < n < (\text{eo\_sign\_method\_X})) + 2$$

5           Where eo\_sign\_method\_X can take 4 values 0, 1, 2, 3. This set makes it easier to derive the value N from the eo\_sign\_method\_X.

In other embodiments, the set of possible N is adapted to match the statistics of the signal content as {0, 2, 4, 6} or another example as {0, 1, 3, 6}.

10           In these examples, the set of possible values N is built so that the maximum value is not very large and the difference between the first value of the set compared to the value between the last values of the set is small.

15           All these embodiments may be very interesting when the signalling is at CTU level or inferior, as consecutive values do not correspond to the statistics of the natural content and related quantization error. One advantage is a coding efficiency improvement.

20           According to another embodiment, the predetermined sign method is selected at the encoder side based on a plurality of limit values N that depend on the slice type, i.e. the set of N values may depend on the slice type. This may be interesting since the quantization errors content depends on the slice type even if an objective measure gives the same quality measure. In this embodiment, the values contained in the set for Inter P slices are higher than the set of values used for Inter B slices.

In an alternative embodiment, at least one value in the set of possible N values for Inter P slices is higher than its corresponding values inside the set of possible N values for Inter B slices. The N values can be related to Luma only.

25           This alternative may be implemented by modifying the process shown in Figure 21, for example by using this formula in the module 2119:

$$M = \text{Set}[\text{Slice\_type}][\text{eo\_sign\_method\_X}]$$

where the Slice\_type is Intra, Inter P or Inter B.

For example, the table Set can be the following:

- 30                           Set[Intra][ ] = {0,1,2,3}  
                               Set[InterP][ ] = {1,2,4,8}  
                               Set[InterB][ ] = {0,2,4,6}

35           According to another embodiment, the predetermined sign method is selected at the encoder side based on a plurality of limit values N that depend on colour components, i.e. the set of N values depends on the colour components. In this

embodiment, the set of possible values N for luma component are different from the set of possible values N for both Chroma components. An advantage of this embodiment may be a coding efficiency improvement thanks to an adapted set of N values for each colour component type.

5                    In one additional embodiment, the set of N values for luma component contains higher values than the set of N values for Chroma components. An advantage of this embodiment may be a coding efficiency improvement thanks to an adapted set of N values for each colour component type. This is because for the chroma component the difference between neighbouring samples is lower than for luma component so it is  
10 preferable to reduce the values N for chroma rather than for Luma.

In one additional embodiment or alternative embodiment, the number of possible N values in the set for Chroma is inferior to the number of the possible N values in the set for Luma. An advantage of this embodiment may be that the coding efficiency increases because the rate related to the signalling of the edge offset sign method is  
15 reduced to a more optimal number of methods.

In one embodiment, two sets of N values are used when the input signal is YUV and one set of N values when the input signal is RGB.

These embodiments can be implemented by modifying the process shown in Figure 21, for example by using this formula in the module 2119:

20                    
$$M = \text{Set}[X][\text{eo\_sign\_method\_X}]$$

where X is equal to luma or chroma and the Slice\_type is Intra, Inter P or Inter B.

For example, the table Set may be the following:

25                    
$$\text{Set}[\text{chroma}] = \{0,1\}$$
  

$$\text{Set}[\text{Luma}] = \{0,2,4,8\}$$

**Figure 22** is a flow chart illustrating how SAO filtering is performed on an image part according to a fifth embodiment of the first aspect of the present invention.

30                    According to this fifth embodiment, the predetermined sign method is selected at the encoder side based on a plurality of limit values N that depend on a Luma component only, i.e. the N values concern the Luma component only. In practice, the set of possible N values for chroma component is thus set to 0. An advantage of this embodiment may be a coding efficiency improvement by removing the eo\_sign\_method\_chroma. This embodiment may be implemented by removing the eo\_sign\_method\_chroma from the figures previously described.

Compared to the process shown in Figure 17, the present process comprises an additional module 2212, which is described below.

In an initial step 2202, the "sao\_type\_idx\_X" syntax element is read and decoded. The code word representing this syntax element can use a fixed length code or could use any method of arithmetic coding. The syntax element sa

5 o\_type\_idx\_X enables determination of the type of SAO applied for the frame area to be processed for the colour component Y or for both Chroma components U & V. The techniques of the present invention are not limited to use with YUV colour components and can be used with RGB colour components or any other colour components.

10 In the same step 2202, a test is performed to determine if the "sao\_type\_idx\_X" is strictly positive. If "sao\_type\_idx\_X" is equal to "0" signifying that there is no SAO for this frame area (CTU) for Y if X is set equal to Y and that there is no SAO for this frame area for U and V if X is set equal to U and V. The determination of the SAO parameters is complete and the process proceeds to step 2208. Otherwise, if

15 the "sao\_type\_idx" is strictly positive, this signifies that SAO parameters exist for this CTU in the bitstream.

Then the process proceeds to step 2206 where a loop is performed for four iterations. The four iterations are carried in step 2207 where the absolute value of offset j is read and decoded from the bitstream. These four offsets correspond either to the four

20 absolute values of the offsets (O1, O2, O3, O4) of the four Edge indexes of SAO Edge Offset (see Figure 27A or 27B) or to the four absolute values of the offsets related to the four ranges of the SAO band Offset (see Figure 8).

After step 2207, the process proceeds to step 2203 where a test is performed to determine if the type of SAO corresponds to the Band Offset type (sao\_type\_idx\_X ==

25 1).

If the outcome is positive, the signs of the offsets for the Band Offset mode are decoded in steps 2209 and 2210, except for each offset that has a zero value, before the following step 2204 is performed in order to read in the bitstream and to decode the position "sao\_band\_position\_X" of the SAO band as illustrated in Figure 8.

30 If the outcome is negative in step 2203 ("sao\_type\_idx\_X" is set equal to 2), this signifies that the Edge Offset type is used. According to the present embodiment, the decoder checks whether the considered component is the Luma component at step 2212 and if so reads the eo\_sign\_method\_Luma from the bitstream 2201 in step 2211. Otherwise, the process goes directly to step 2205.

Next, the Edge Offset class (corresponding to the direction 0, 45, 90 and 135 degrees) is extracted from the bitstream 2201 in step 2205. If X is equal to Y, the read syntax element is "sao\_eo\_class\_luma" and if X is set equal to U and V, the read syntax element is "sao\_eo\_class\_chroma".

5                   When the four offsets have been decoded, the reading of the SAO parameters is complete and the process proceeds to step 2208.

**Figure 23** illustrates steps of a process of parsing of SAO parameters in the bitstream 2301 at the decoder side according to a sixth embodiment.

10                   According to this embodiment, the predetermined sign method is selected at the encoder side based on a plurality of limit values N that are signalled in a slice header, i.e. only one set of N values is transmitted in the slice header for the luma component and only one set is transmitted for the chroma components. An advantage of this embodiment may be an increase of coding efficiency by setting the useful set at encoder side for each slice.

15                   At step 2302, the slice\_sao\_luma\_flag is extracted from the bitstream 2301. If the value of the flag is set to true (=1), the Max\_eo\_sign\_method\_luma is extracted at step 2303 from the bitstream 2301. This value represents the number of elements in the set. Then each element value (2305) of the set is extracted (2307) from the bitstream (2301) and will be used for the Edge offset classification sign method for all the luma  
20                   CTB of the current slice/frame. The process then goes to the next slice header syntax element (step 2306).

                  If the value of the flag slice\_sao\_luma\_flag is set to false, the slice\_sao\_chroma\_flag is extracted at step 2304 from the bitstream 2301. If the value of the flag is set to true (=1), the Max\_eo\_sign\_method\_chroma is extracted at step 2308  
25                   from the bitstream 2301. This value represents the number of elements in the set. Then each element value (2309) of the set is extracted (2310) from the bitstream (2301) and will be used for the Edge offset classification sign method for all the luma CTB of the current slice/frame. The process then goes to the next slice header syntax element (step 2306). If the value of the flag is set to false (=0), the process goes directly to the next  
30                   slice header syntax element (step 2306).

**Figure 24** is a flow chart illustrating steps 240 of an example of a statistics computed at the encoder side that can be applied for the Edge Offset type filter, in the case of the SAO filtering according to embodiments of the present invention.

In this embodiment, there is a competition between different N values at the encoder. In these embodiments, the RD cost of each possible value for eo\_sign\_method is computed in order to select the best eo\_sign\_method at encoder side.

This figure illustrates the setting of the variable CTUStats containing all information needed to derive each best rate distortion offsets for each class. Moreover, it illustrates the selection of the best SAO parameters set for the current CTU.

For each colour component Y, U, V (or RGB) (2411) each SAO type is evaluated. For each SAO type (2412) and for each eo\_sign\_method\_X (2416), the variables  $Sum_j$  and  $SumNbPix_j$  are set to zero in an initial step 2401. The current frame area 2403 contains N pixels.

$j$  is the current range number to determine the four offsets (related to the four edge indexes shown in Figure 7B for Edge Offset type or to the 32 ranges of pixel values shown in Figure 8 for Band Offset type).  $Sum_j$  is the sum of the differences between the pixels in the range  $j$  and their original pixels.  $SumNbPix_j$  is the number of pixels in the frame area, the pixel value of which belongs to the range  $j$ .

In step 2402, a variable  $i$ , used to successively consider each pixel  $P_i$  of the current frame area, is set to zero. Then, the first pixel  $P_i$  of the frame area 2403 is extracted in step 2404. In step 2405, the class of the current pixel is determined by checking the conditions defined in Figure 27A and 27B. Then a test is performed in step 2405. During step 2405, a check is performed to determine if the class of the pixel value  $P_i$  corresponds to the value “none of the above” of Figure 27A and 27B.

If the outcome is positive, then the value “i” is incremented in step 2408 in order to consider the next pixels of the frame area 2403.

Otherwise, if the outcome is negative in step 2406, the next step is 2407 where the related  $SumNbPix_j$  (i.e. the sum of the number of pixels for the class determined in step 2405) is incremented and the difference between  $P_i$  and its original value  $P_i^{org}$  is added to  $Sum_j$ . In the next step 2408, the variable  $i$  is incremented in order to consider the next pixels of the frame area 2403.

Then a test 2409 is performed to determine if all pixels have been considered and classified. If the outcome is negative, the process loops back to step 2404 described

above. Otherwise, if the outcome is positive, the process proceeds to step 2410 where the variable CTUStats for the current colour component X, the SAO type SAO\_type, the eo\_sign\_method\_X and the current class<sup>j</sup> are set equal to  $Sum_j$  for the first value and  $SumNbPix_j$  for the second value.

5 The process then loops for each eo\_sign\_method\_X (2417) until the last one (2418) and for each SAO type (2413) until the last one (2414). When the last one is reached, the process handles the next component (2415).

As described with reference to Figure 13, these variables can be used to compute for example the optimal offset parameter  $Offset_j$  of each class  $j$ . This offset  $Offset_j$  may be the average of the differences between the pixels of class  $j$  and their original values. Thus,  $Offset_j$  is given by the following formula:

$$Offset_j = \frac{Sum_j}{SumNbPix_j}$$

Note that the offset  $Offset_j$  is an integer value. Consequently, the ratio defined in this formula may be rounded, either to the closest value or using the ceiling or floor function.

Each offset  $Offset_j$  is an optimal offset  $O_{optj}$  in terms of distortion.

Thus, in Figure 24, as a new loop on the edge offset sign methods is added (2416, 2417, 2418) to obtain the statistics CTUStats (2410) for each edge offset sign method, the table CTUStats has an additional dimension to set the statistics for edge offset sign method compared to Figure 13.

To evaluate an RD cost for a merge of SAO parameters, the encoder uses the statistics set in table CTUStats. According to the following examples for the SAO Merge Left and by considering the type for Luma Left\_Type\_Y and the four related offsets O\_Left\_0, O\_Left\_1, O\_Left\_2, O\_Left\_3, the distortion can be obtained by the following formula:

Distortion\_Left\_Y =  
 (CTUStats[Y][ Left\_Type\_Y][0][1] x O\_Left\_0 x O\_Left\_0 - CTUStats[Y][ Left\_Type\_Y][0][0] x O\_Left\_0 x 2)>> Shift  
 + (CTUStats[Y][ Left\_Type\_Y][1][1] x O\_Left\_1 x O\_Left\_1 - CTUStats[Y][ Left\_Type\_Y][1][0] x O\_Left\_1 x 2)>> Shift

+ (CTUStats[Y][ Left\_Type\_Y][2][1] x O\_Left\_2 x O\_Left\_2 - CTUStats[Y][ Left\_Type\_Y][2][0] x O\_Left\_2 x 2)>> Shift  
 + (CTUStats[Y][ Left\_Type\_Y][3][1] x O\_Left\_3 x O\_Left\_3 - CTUStats[Y][ Left\_Type\_Y][3][0] x O\_Left\_3 x 2)>> Shift

5           The variable Shift is designed for a distortion adjustment. The distortion should be negative as SAO is a post filtering.

The same computing is applied for Chroma components. The Lambda of the Rate distortion cost is fixed for the three components. For an SAO parameters merged with the left CTU, the rate is only 1 flag, which is CABAC coded.

10           **Figure 25** is a flow chart illustrating steps carried out an encoder to determine SAO parameters for the CTUs of a group (frame or slice).

In this embodiment, the predetermined sign method is selected at the encoder side based on a plurality of limit values N that depend on a colour component. Therefore, the statistics obtained by the process of Figure 24 may be used to determine  
 15 the best edge class and best eo\_sign\_method for luma and respectively for both chroma components.

The process starts with a current CTU (2501). First, the statistics for all possible SAO types and classes are accumulated in the variable CTUStats (2502). The process of Step 2502 is described above with reference to Figure 24.

20           According to the value set in the variable CTUStats, the RD cost for the SAO merge Left is evaluated if the Left CTU is in the current Slice (2503) as the RD cost of the SAO Merge UP (2504). Thanks to the statistics in CTUStats (2502), new SAO parameters are evaluated for Luma (2505), for both Chroma components (2509) and for each eo\_sign\_method\_chroma (2518).

25           For each SAO type (2506) and for each eo\_sign\_method\_luma (2517), the best RD offsets and other parameters for Band offset classification are obtained (2507). All RD costs are computed thanks to their respective SAO parameters and luma sign methods (2508). In the same way for both Chroma components, the optimal RD offsets and parameters are selected (2511). All RD costs are computed thanks to their  
 30 respective SAO parameters and chroma sign methods (2512). All this RD costs are then compared in order to select the best SAO parameters set (2515). These RD costs are also compared to disable SAO independently for the Luma and the Chroma components (2513, 2514). The use of a new SAO parameters set (2515) is compared to the SAO parameters set "Merging" or sharing (2516) from the left and up CTU.

35

### Second group of embodiments (second aspect)

According to a second aspect of the invention, there is provided a method for obtaining an edge offset for performing a Sample Adaptive Offset (SAO) filtering, wherein when the “sign” parameter N is strictly positive, the sign of at least part of the offsets is explicitly signalled.

In other words, the second aspect provides a method of performing sample adaptive offset (SAO) filtering on an image comprising a plurality of image parts, the method comprising:

obtaining a plurality of edge offsets to perform a SAO filtering;

wherein the method further comprises signalling a sign of at least one edge offset of the plurality in the bitstream.

According to a first embodiment of the second aspect, the sign of the edge offset is signalled for all the edge offsets of the plurality for which a sign parameter N to use in a sign function for determining an index of the edge offset is positive.

According to other embodiments, the sign of the edge offsets is signalled when at least one of the following conditions is fulfilled:

- the classification uses a new edge sign method for this edge offset;
- the input value of the sign function is modified for the edge offset classification;
- the parameter N to use in the sign function for determining the index of the edge offset is strictly positive.

An advantage of these embodiment is a coding efficiency increase compared to the implicit sign signalling of offsets for Edge offset when the value N of the sign method is different from zero. This is an unexpected result, as it is known that the explicit sign signal of the offsets of the edge offset classification induces a decrease of coding efficiency with the traditional sign method.

**Figure 26** is a flow chart illustrating steps of a process to decode SAO parameters according to a second embodiment of the second aspect.

According to this second embodiment, the sign of the edge offset is signalled for all the edge offsets of the plurality if all the sign parameters N to use in a sign function for determining an index of the edge offsets are higher than a predetermined threshold.

Hence, the sign is signalled only when the sign parameter N is high for all offsets, i.e. when all the sign parameters N to use in the sign function for determining the indexes of these edge offsets are higher than a predetermined threshold.

Compared to the process shown in Figure 10, the present process performs an additional loop, which comprises additional modules 2611, 2612, 2613 and 2614 as described below.

5 In an initial step 2602, the "sao\_type\_idx\_X" syntax element is read and decoded. The code word representing this syntax element can use a fixed length code or could use any method of arithmetic coding. The syntax element sao\_type\_idx\_X enables determination of the type of SAO applied for the frame area to be processed for the colour component Y or for both Chroma components U & V.

10 Incidentally, although YUV colour components are used in HEVC (sometimes called Y, Cr and Cb components), it will be appreciated that in other video coding schemes other colour components may be used, for example RGB colour components. The techniques of the present invention are not limited to use with YUV colour components, and can be used with RGB colour components or any other colour components.

15 In the same step 2602, a test is performed to determine if the "sao\_type\_idx\_X" is strictly positive. If "sao\_type\_idx\_X" is equal to "0" signifying that there is no SAO for this frame area (CTU) for Y if X is set equal to Y and that there is no SAO for this frame area for U and V if X is set equal to U and V. The determination of the SAO parameters is complete and the process proceeds to step 2608. Otherwise, if  
20 the "sao\_type\_idx" is strictly positive, this signifies that SAO parameters exist for this CTU in the bitstream.

Then the process proceeds to step 2606 where a loop is performed for four iterations. The four iterations are carried in step 2607 where the absolute value of offset j is read and decoded from the bitstream. These four offsets correspond either to the four  
25 absolute values of the offsets (O1, O2, O3, O4) of the four Edge indexes of SAO Edge Offset (see Figure 27A or 27B) or to the four absolute values of the offsets related to the four ranges of the SAO band Offset (see Figure 8).

After step 2607, the process proceeds to step 2603 where a test is performed to determine if the type of SAO corresponds to the Band Offset type (sao\_type\_idx\_X ==  
30 1).

If the outcome is positive, the signs of the offsets for the Band Offset mode are decoded in steps 2609 and 2610, except for each offset that has a zero value, before the following step 2604 is performed in order to read in the bitstream and to decode the position "sao\_band\_position\_X" of the SAO band as illustrated in Figure 8.

If the outcome is negative in step 2603 (“sao\_type\_idx\_X” is set equal to 2), this signifies that the Edge Offset type is used. If X is equal to Y, the read syntax element is “sao\_eo\_class\_luma” and if X is set equal to U and V, the read syntax element is “sao\_eo\_class\_chroma”. Consequently, the Edge Offset class (corresponding to the  
5 direction 0, 45, 90 and 135 degrees) is extracted from the bitstream in step 2605.

When the four offsets have been extracted, the process proceeds to step 2611 where it is tested whether the edge offset sign method eo\_sign\_method\_X (2614), i.e. the index corresponding to the N value in the set table for the current colour component is superior to a threshold Th (2611). If so, the four signs (2612) are extracted  
10 (2613) from the bitstream. Then, the reading of the SAO parameters is complete and the process proceeds to step 2608.

An advantage of this embodiment may be a coding efficiency increase compared to the implicit sign signalling of offsets for Edge offset classification when the value N used in the sign method is high.

15 It may be noted that in this embodiment, the edge offset sign method may be the same for all CTUs of the sequence or selected slice by slice. When the edge offset sign method has always the same value, the module 2611 may not be used.

A description of the Edge Offset used in these embodiments of the second aspect is now provided with reference to **Figures 27A** and **27B**.

20 The Edge Offset type involves determining an edge index for each pixel by comparing its pixel value to the values of two neighbouring pixels. Moreover, these two neighbouring pixels depend on a parameter, which indicates the direction of these two neighbouring pixels with respect to the current pixel. These directions are the 0-degree (horizontal direction), 45-degree (diagonal direction), 90-degree (vertical direction) and  
25 135-degree (second diagonal direction), as shown in Figure 7A.

**Figure 27A** shows a table that gives the offset value to be applied to the pixel value of a particular pixel at the decoder side when the value N is superior to a threshold Th.

30 When the value of C is less than the two values of neighbouring pixels Cn1 and Cn2, the offset to be added to the pixel value of the pixel C is “O1” (O1 can be positive or negative or equal to 0). When the pixel value of C is less than one pixel value of its neighbouring pixels (either Cn1 or Cn2) and C is equal to one value of its neighbours, the offset to be added to this pixel sample value is “O2” (O2 can be positive or negative or equal to 0).

When the pixel value of  $c$  is less than one of the pixel values of its neighbours ( $C_{n1}$  or  $C_{n2}$ ) and the pixel value of  $C$  is equal to one value of its neighbours, the offset to be applied to this pixel sample is "O3" (O3 can be positive or negative or equal to 0). When the value of  $C$  is greater than the two values of  $C_{n1}$  or  $C_{n2}$ , the offset to be applied  
5 to this pixel sample is "O4" (O4 can be positive or negative or equal to 0).

When none of the above conditions is met on the current sample and its neighbours, no offset value is added to the current pixel  $C$  as depicted by the Edge Index value "2" of the table.

It is important to note that for the particular case of the Edge Offset type, the  
10 absolute value of each offset ( $O1$ ,  $O2$ ,  $O3$ ,  $O4$ ) is encoded in the bitstream. The sign is also encoded.

In this **Figure 27A**, the samples  $C$ ,  $C_{n1}$  and  $C_{n2}$  correspond to those depicted in **Figure 7A**.

**Figure 27B** shows a table that gives the offset value to be applied to the pixel  
15 value of a particular pixel at the decoder side when the value  $N$  is inferior or equal to a threshold  $Th$ .

When the value of  $C$  is less than the two values of neighbouring pixels  $C_{n1}$  and  $C_{n2}$ , the offset to be added to the pixel value of the pixel  $C$  is "O1" (O1 can be positive or negative or equal to 0). When the pixel value of  $C$  is less than one pixel value  
20 of its neighbouring pixels (either  $C_{n1}$  or  $C_{n2}$ ) and  $C$  is equal to one value of its neighbours, the offset to be added to this pixel sample value is "+ O2" (this offset is positive or equal to 0).

When the pixel value of  $c$  is less than one of the pixel values of its neighbours ( $C_{n1}$  or  $C_{n2}$ ) and the pixel value of  $C$  is equal to one value of its neighbours, the offset  
25 to be applied to this pixel sample is "- O3" (this offset is negative or equal to 0). When the value of  $C$  is greater than the two values of  $C_{n1}$  or  $C_{n2}$ , the offset to be applied to this pixel sample is "O4" (O4 can be positive or negative or equal to 0).

When none of the above conditions is met on the current sample and its neighbours, no offset value is added to the current pixel  $C$  as depicted by the Edge Index  
30 value "2" of the table.

It is important to note that for the particular case of the Edge Offset type, the absolute value of each offset ( $O1$ ,  $O2$ ,  $O3$ ,  $O4$ ) is encoded in the bitstream. The sign to be applied to each offset depends on the edge index (or the Edge Index in the HEVC specifications) to which the current pixel belongs. According to the table represented in  
35 **Figure 27B**, for Edge Index 1 ( $O2$ ) a positive offset is applied. For Edge Index 3 ( $O3$ ), a

negative offset is applied to the current pixel. For the Edge index 1 and for Edge index 4 the sign is encoded/decoded.

In this **Figure 27B**, the samples C, Cn1 and Cn2 correspond to those depicted in **Figure 7A**.

5 In one additional embodiment, the sign is CABAC coded. In addition, no context is needed. An advantage of this additional embodiment is a coding efficiency increase because the sign is often equal to the implicit sign.

**Figure 28** is a flow chart illustrating steps of a process to decode SAO parameters according to a third embodiment of the second aspect.

10 According to this third embodiment, the at least one edge offset the sign of which is signalled belongs to the peak and valley configurations. More specifically, the sign of the at least one edge offset is signalled whatever is a predetermined sign method used to determine an index of the edge offset.

In the present example, the sign of the edge offset is explicit for the peak and Valley classes and implicit for the two other classes, whatever the sign method used to know the sign of the edge offset. It means that for the offsets O1 and O4 shown in Figure 27B, a sign is transmitted in the bitstream.

15 In an initial step 2802, the "sao\_type\_idx\_X" syntax element is read and decoded. The code word representing this syntax element can use a fixed length code or could use any method of arithmetic coding. The syntax element sao\_type\_idx\_X enables determination of the type of SAO applied for the frame area to be processed for the colour component Y or for both Chroma components U & V.

25 Incidentally, although YUV colour components are used in HEVC (sometimes called Y, Cr and Cb components), it will be appreciated that in other video coding schemes other colour components may be used, for example RGB colour components. The techniques of the present invention are not limited to use with YUV colour components, and can be used with RGB colour components or any other colour components.

30 In the same step 2802, a test is performed to determine if the "sao\_type\_idx\_X" is strictly positive. If "sao\_type\_idx\_X" is equal to "0" signifying that there is no SAO for this frame area (CTU) for Y if X is set equal to Y and that there is no SAO for this frame area for U and V if X is set equal to U and V. The determination of the SAO parameters is complete and the process proceeds to step 2808. Otherwise, if the "sao\_type\_idx" is strictly positive, this signifies that SAO parameters exist for this  
35 CTU in the bitstream.

Then the process proceeds to step 2806 where a loop is performed for four iterations. The four iterations are carried in step 2807 where the absolute value of offset  $j$  is read and decoded from the bitstream. These four offsets correspond either to the four absolute values of the offsets (O1, O2, O3, O4) of the four Edge indexes of SAO Edge Offset (see Figure 27A or 27B) or to the four absolute values of the offsets related to the four ranges of the SAO band Offset (see Figure 8).

After step 2807, the process proceeds to step 2803 where a test is performed to determine if the type of SAO corresponds to the Band Offset type ( $\text{sao\_type\_idx\_X} == 1$ ).

10 If the outcome is positive, the signs of the offsets for the Band Offset mode are decoded in steps 2809 and 2810, except for each offset that has a zero value, before the following step 2804 is performed in order to read in the bitstream and to decode the position "sao\_band\_position\_X" of the SAO band as illustrated in Figure 8.

If the outcome is negative in step 2803 ("sao\_type\_idx\_X" is set equal to 2), 15 this signifies that the Edge Offset type is used. If X is equal to Y, the read syntax element is "sao\_eo\_class\_luma" and if X is set equal to U and V, the read syntax element is "sao\_eo\_class\_chroma". Consequently, the Edge Offset class (corresponding to the direction 0, 45, 90 and 135 degrees) is extracted from the bitstream in step 2805.

For each offset (2812), it is determined whether the offset is the peak ( $j==1$ ) 20 or the valley ( $j==4$ ) offset ie. O1 or O4 (2815). If so, the sign of the offset is extracted (2813) from the bitstream. Otherwise, the sign is implicit and set as depicted in Figure 27B. Then, the reading of the SAO parameters is complete and the process proceeds to step 2808.

25 An advantage of this embodiment may be a coding efficiency increase compared to the implicit sign signalling of offsets for Edge offset classification and a coding efficiency increase compared to the explicit sign signalling of offsets for Edge offset classification.

In another embodiment, the sign is CABAC coded. An advantage of this additional embodiment is a coding efficiency increase because the sign is often equal to 30 the implicit sign.

**Figure 29** is a flow chart illustrating steps of a process to decode SAO parameters according to a fourth embodiment of the second aspect.

According to this fourth embodiment, the sign the edge offset is signalled only for the edge offsets for which the sign parameters N to use for determining an index 35 of these edge offsets are higher than a predetermined threshold.

In other words, the sign of the edge offset is explicit (i.e. it is signalled in the bitstream) for the peak and valley classes and implicit for the two other classes. Thus, a sign is transmitted for the offsets O1 and O4 as shown in Figure 27B when N is superior or equal to a threshold.

5 Compared to the process shown in Figure 28, the present process comprises additional modules 2914 and 2911 as described below.

In an initial step 2902, the "sao\_type\_idx\_X" syntax element is read and decoded. The code word representing this syntax element can use a fixed length code or could use any method of arithmetic coding. The syntax element saotype\_idx\_X  
10 enables determination of the type of SAO applied for the frame area to be processed for the colour component Y or for both Chroma components U & V.

Incidentally, although YUV colour components are used in HEVC (sometimes called Y, Cr and Cb components), it will be appreciated that in other video coding schemes other colour components may be used, for example RGB colour  
15 components. The techniques of the present invention are not limited to use with YUV colour components, and can be used with RGB colour components or any other colour components.

In the same step 2902, a test is performed to determine if the "sao\_type\_idx\_X" is strictly positive. If "sao\_type\_idx\_X" is equal to "0" signifying that  
20 there is no SAO for this frame area (CTU) for Y if X is set equal to Y and that there is no SAO for this frame area for U and V if X is set equal to U and V. The determination of the SAO parameters is complete and the process proceeds to step 2908. Otherwise, if the "sao\_type\_idx" is strictly positive, this signifies that SAO parameters exist for this CTU in the bitstream.

25 Then the process proceeds to step 2906 where a loop is performed for four iterations. The four iterations are carried in step 2907 where the absolute value of offset j is read and decoded from the bitstream. These four offsets correspond either to the four absolute values of the offsets (O1, O2, O3, O4) of the four Edge indexes of SAO Edge Offset (see Figure 27A or 27B) or to the four absolute values of the offsets related to the  
30 four ranges of the SAO band Offset (see Figure 8).

After step 2907, the process proceeds to step 2903 where a test is performed to determine if the type of SAO corresponds to the Band Offset type (sao\_type\_idx\_X == 1).

If the outcome is positive, the signs of the offsets for the Band Offset mode  
35 are decoded in steps 2909 and 2910, except for each offset that has a zero value, before

the following step 2904 is performed in order to read in the bitstream and to decode the position "sao\_band\_position\_X" of the SAO band as illustrated in Figure 8.

If the outcome is negative in step 2903 ("sao\_type\_idx\_X" is set equal to 2), this signifies that the Edge Offset type is used. If X is equal to Y, the read syntax element  
5 is "sao\_eo\_class\_luma" and if X is set equal to U and V, the read syntax element is "sao\_eo\_class\_chroma". Consequently, the Edge Offset class (corresponding to the direction 0, 45, 90 and 135 degrees) is extracted from the bitstream in step 2905.

When the four offsets have been extracted, the process proceeds to step 2911 where it is tested whether the edge offset sign method eo\_sign\_method\_X (2914),  
10 i.e. the index corresponding to the N value (in the conversion table "set") for the current colour component is superior to a threshold Th (2911).

If so, for each of the four offsets (2912), it is determined whether the offset is a peak (j=1) or the valley (j=4) offset ie.O1 or O4 (2815). If so, the sign of the offset is extracted (2913) from the bitstream. Otherwise, the sign is implicit and set as depicted  
15 in Figure 27B. Then, the reading of the SAO parameters is complete and the process proceeds to step 2908.

An advantage of this embodiment may be a coding efficiency increase compared to the implicit sign signalling of offsets for edge offset classification and a coding efficiency increase compared to the explicit sign signalling of offsets for edge  
20 offset classification. The coding efficiency is also increased compared to the third embodiment.

According to embodiments, the sign the edge offset is signalled only for the edge offsets for which the sign parameters N to use for determining an index of these edge offsets are different from zero.

25 According to these embodiment, the threshold value is equal to 0 in order that the sign value is implicit for all offsets of the edge offset classification when the sign method is a traditional sign method (N=0) and the signs of offsets are explicit (i.e. sent in the bitstream) for the peak and valley classes when the sign method has a N>0. This may provide a very interesting coding efficiency improvement.

30 **Figure 30** is a flow chart illustrating steps of a process to decode SAO parameters according to a fifth embodiment of the second aspect.

According to this fifth embodiment, the sign the edge offset is signalled only for the luma component, i.e. the signs of offsets for the Edge offset classification are explicit for the Luma component as in Figure 27A and are implicit for Chroma  
35 components as in Figure 7B.

This may provide a coding efficiency improvement compared to the explicit signalling for all colour components.

In an initial step 3002, the "sao\_type\_idx\_X" syntax element is read and decoded. The code word representing this syntax element can use a fixed length code or could use any method of arithmetic coding. The syntax element sa

5  
o\_type\_idx\_X enables determination of the type of SAO applied for the frame area to be processed for the colour component Y or for both Chroma components U & V.

Incidentally, although YUV colour components are used in HEVC (sometimes called Y, Cr and Cb components), it will be appreciated that in other video coding schemes other colour components may be used, for example RGB colour components. The techniques of the present invention are not limited to use with YUV colour components, and can be used with RGB colour components or any other colour components.

10

In the same step 3002, a test is performed to determine if the "sao\_type\_idx\_X" is strictly positive. If "sao\_type\_idx\_X" is equal to "0" signifying that there is no SAO for this frame area (CTU) for Y if X is set equal to Y and that there is no SAO for this frame area for U and V if X is set equal to U and V. The determination of the SAO parameters is complete and the process proceeds to step 3008. Otherwise, if the "sao\_type\_idx" is strictly positive, this signifies that SAO parameters exist for this

15  
20 CTU in the bitstream.

Then the process proceeds to step 3006 where a loop is performed for four iterations. The four iterations are carried in step 3007 where the absolute value of offset j is read and decoded from the bitstream. These four offsets correspond either to the four absolute values of the offsets (O1, O2, O3, O4) of the four Edge indexes of SAO Edge Offset (see Figure 27A and 27B) or to the four absolute values of the offsets related to the four ranges of the SAO band Offset (see Figure 8).

25

After step 3007, the process proceeds to step 3003 where a test is performed to determine if the type of SAO corresponds to the Band Offset type (sao\_type\_idx\_X == 1).

If the outcome is positive, the signs of the offsets for the Band Offset mode are decoded in steps 3009 and 3010, except for each offset that has a zero value, before the following step 3004 is performed in order to read in the bitstream and to decode the position "sao\_band\_position\_X" of the SAO band as illustrated in Figure 8.

30

If the outcome is negative in step 3003 ("sao\_type\_idx\_X" is set equal to 2), this signifies that the Edge Offset type is used. If X is equal to Y, the read syntax element

35

is "sao\_eo\_class\_luma" and if X is set equal to U and V, the read syntax element is "sao\_eo\_class\_chroma". Consequently, the Edge Offset class (corresponding to the direction 0, 45, 90 and 135 degrees) is extracted from the bitstream in step 3005.

When the four offsets have been extracted, the process proceeds to step  
5 3016 where it is tested whether the current colour component X corresponds to the luma component. If so, the four signs (3012) are extracted (3013) from the bitstream. Then, the reading of the SAO parameters is complete and the process proceeds to step 3008.

If the current colour component is chroma component, the process proceeds directly to step 3008.

10 An advantage of this embodiment may be a coding efficiency improvement compared to the signalling of signs for all colour components. This is because for Chroma component the implicit signalling may be more efficient than the explicit signalling of signs.

In other embodiments, the signs may be signalled for all colour components,  
15 which simplifies the hardware implementation.

**Figure 31** is a flow chart illustrating steps of a process to decode SAO parameters according to a sixth embodiment of the second aspect.

According to this sixth embodiment, the signs are explicitly signalled for Luma when the SAO sign method, i.e. the index corresponding to the N value in the "set"  
20 table, is higher than a threshold Th.

Compared to the process shown in Figure 30, the present process comprises additional modules 3111 and 3114 as described below.

In an initial step 3102, the "sao\_type\_idx\_X" syntax element is read and decoded. The code word representing this syntax element can use a fixed length code  
25 or could use any method of arithmetic coding. The syntax element sao\_type\_idx\_X enables determination of the type of SAO applied for the frame area to be processed for the colour component Y or for both Chroma components U & V.

Incidentally, although YUV colour components are used in HEVC (sometimes called Y, Cr and Cb components), it will be appreciated that in other video  
30 coding schemes other colour components may be used, for example RGB colour components. The techniques of the present invention are not limited to use with YUV colour components, and can be used with RGB colour components or any other colour components.

In the same step 3102, a test is performed to determine if the  
35 "sao\_type\_idx\_X" is strictly positive. If "sao\_type\_idx\_X" is equal to "0" signifying that

there is no SAO for this frame area (CTU) for Y if X is set equal to Y and that there is no SAO for this frame area for U and V if X is set equal to U and V. The determination of the SAO parameters is complete and the process proceeds to step 3108. Otherwise, if the "sao\_type\_idx" is strictly positive, this signifies that SAO parameters exist for this  
5 CTU in the bitstream.

Then the process proceeds to step 3106 where a loop is performed for four iterations. The four iterations are carried in step 3107 where the absolute value of offset j is read and decoded from the bitstream. These four offsets correspond either to the four absolute values of the offsets (O1, O2, O3, O4) of the four Edge indexes of SAO Edge  
10 Offset (see Figure 27A or 27B) or to the four absolute values of the offsets related to the four ranges of the SAO band Offset (see Figure 8).

After step 3107, the process proceeds to step 3103 where a test is performed to determine if the type of SAO corresponds to the Band Offset type (sao\_type\_idx\_X == 1).  
15

If the outcome is positive, the signs of the offsets for the Band Offset mode are decoded in steps 3109 and 3110, except for each offset that has a zero value, before the following step 3104 is performed in order to read in the bitstream and to decode the position "sao\_band\_position\_X" of the SAO band as illustrated in Figure 8.

If the outcome is negative in step 3103 ("sao\_type\_idx\_X" is set equal to 2),  
20 this signifies that the Edge Offset type is used. If X is equal to Y, the read syntax element is "sao\_eo\_class\_luma" and if X is set equal to U and V, the read syntax element is "sao\_eo\_class\_chroma". Consequently, the Edge Offset class (corresponding to the direction 0, 45, 90 and 135 degrees) is extracted from the bitstream in step 3105.

When the four offsets have been extracted, the process proceeds to step  
25 3116 where it is tested whether the current colour component X corresponds to the luma component. If so, it is tested whether the edge offset sign method eo\_sign\_method\_X (3114) i.e. the index corresponding to the N value (in the conversion table "set") for the current colour component (luma) is superior to a threshold Th (3111). The threshold can be 0 or higher. If so, the four signs (3112) are extracted (3113) from the bitstream. Then,  
30 the reading of the SAO parameters is complete and the process proceeds to step 3108.

Otherwise (edge offset sign method i.e. the index corresponding to an index of the N value inferior to the threshold), the sign is implicit and set as depicted in Figure 27B. Then, the reading of the SAO parameters is complete and the process proceeds to step 3108.

If the current colour component is chroma component, the process proceeds directly to step 3108.

This embodiment provides a coding efficiency increase compared to the previous embodiments. This is because for Chroma component the implicit signalling is often more efficient than the explicit signalling of signs.

**Figure 32** is a flow chart illustrating steps of a process to decode SAO parameters according to a seventh embodiment of the second aspect.

According to this seventh embodiment, the sign of the Edge offset is explicit (signalled) for the peak and Valley classes and implicit for the two other classes when the colour component is Luma. In this case, a sign is transmitted for the offsets O1 and O4 as shown in Figure 27B.

In an initial step 3202, the "sao\_type\_idx\_X" syntax element is read and decoded. The code word representing this syntax element can use a fixed length code or could use any method of arithmetic coding. The syntax element sao\_type\_idx\_X enables determination of the type of SAO applied for the frame area to be processed for the colour component Y or for both Chroma components U & V.

Incidentally, although YUV colour components are used in HEVC (sometimes called Y, Cr and Cb components), it will be appreciated that in other video coding schemes other colour components may be used, for example RGB colour components. The techniques of the present invention are not limited to use with YUV colour components, and can be used with RGB colour components or any other colour components.

In the same step 3202, a test is performed to determine if the "sao\_type\_idx\_X" is strictly positive. If "sao\_type\_idx\_X" is equal to "0" signifying that there is no SAO for this frame area (CTU) for Y if X is set equal to Y and that there is no SAO for this frame area for U and V if X is set equal to U and V. The determination of the SAO parameters is complete and the process proceeds to step 3208. Otherwise, if the "sao\_type\_idx" is strictly positive, this signifies that SAO parameters exist for this CTU in the bitstream.

Then the process proceeds to step 3206 where a loop is performed for four iterations. The four iterations are carried in step 3207 where the absolute value of offset j is read and decoded from the bitstream. These four offsets correspond either to the four absolute values of the offsets (O1, O2, O3, O4) of the four Edge indexes of SAO Edge Offset (see Figure 27A and 27B) or to the four absolute values of the offsets related to the four ranges of the SAO band Offset (see Figure 8).

After step 3207, the process proceeds to step 3203 where a test is performed to determine if the type of SAO corresponds to the Band Offset type (`sao_type_idx_X == 1`).

If the outcome is positive, the signs of the offsets for the Band Offset mode are decoded in steps 3209 and 3210, except for each offset that has a zero value, before the following step 3204 is performed in order to read in the bitstream and to decode the position “`sao_band_position_X`” of the SAO band as illustrated in Figure 8.

If the outcome is negative in step 3203 (“`sao_type_idx_X`” is set equal to 2), this signifies that the Edge Offset type is used. If X is equal to Y, the read syntax element is “`sao_eo_class_luma`” and if X is set equal to U and V, the read syntax element is “`sao_eo_class_chroma`”. Consequently, the Edge Offset class (corresponding to the direction 0, 45, 90 and 135 degrees) is extracted from the bitstream in step 3205.

When the four offsets have been extracted, the process proceeds to step 3216 where it is tested whether the current colour component X corresponds to the luma component. If so, for each of the four offsets (3212), it is determined whether the offset is a peak ( $j=1$ ) or the valley ( $j=4$ ) offset i.e. O1 or O4 (3215). If so, the sign of the offset is extracted (3213) from the bitstream. Otherwise, the sign is implicit and set as depicted in Figure 27B. Then, the reading of the SAO parameters is complete and the process proceeds to step 3208. If the current colour component is chroma component, the process proceeds directly to step 3208.

An advantage of this embodiment may be that it may be better in term of coding efficiency because the implicit signalling for Chroma component is often more efficient than the explicit signalling of signs.

In a variant, an additional test may be performed to test whether the edge sign method i.e. the index corresponding to the N value (in the conversion table “set”) is superior to threshold th.

In another embodiment, the signalling for all colour components may allow a simplification of the hardware implementation.

### 30 **Third group of embodiments (third aspect)**

According to a third aspect of the invention, there is provided a method for obtaining an edge offset for performing a Sample Adaptive Offset (SAO) filtering, wherein when the sign of an SAO offset is explicit signalled in the bitstream, its value is predicted.

In other words, the third aspect provides a method of performing sample adaptive offset (SAO) filtering on an image comprising a plurality of image parts, the method comprising:

obtaining a plurality of edge offsets to perform a SAO filtering;

5 wherein a sign of at least one edge offset of the plurality is signalled in the bitstream; and

wherein obtaining a plurality of edge offsets comprises predicting at least one edge offset the sign of which is signalled in the bitstream.

In practice, according to embodiments of this third aspect, the offset value or  
10 more precisely the absolute offset value may be coded with a variable length code or with a fixed code arithmetically coded.

**Figure 33** illustrates steps of a prediction process that may be performed during a (SAO) filtering embodiment at decoder side according to a first embodiment of the third aspect. These steps may be performed during the process shown in Figure 11.  
15 For example, this prediction can be applied in step 701 of Figure 11. These steps may be performed during the process shown in Figure 29 to Figure 32 just before respectively step 2908 to 3208.

According to this first embodiment, the at least one edge offset is predicted from a default value. In this example, the edge offsets are predicted based on a  
20 predictors table that contains default values. An advantage of this embodiment may be a coding efficiency increase.

At the beginning of the process, a decoded offsets table (3301) is obtained. This table comprises offsets values with their respective sign. The value of the four offsets (3302) is then predicted (3303). This is done by adding the variable  $Offset\_X\_j$  of  
25 the component X (YUV) to the predictor  $Predictor\_X\_j$  using a predictor table (3304). The predictor table 3304 may comprise different predictors adapted to each type of offsets (peak, valley etc.). Correspondingly, at encoder side, the Predictor  $Predictor\_X\_j$  is subtracted to the  $Offset\_X\_j$  value. The process then proceeds to step 3305.

According to embodiments, the default value is predetermined for at least  
30 one sequence, for example all sequences, i.e. the values of the predictor table may be the same for the whole slice/frame or for all frames of the sequence or for all sequences.

According to embodiments, the table of default value is transmitted in the bitstream. For instance, these default predictor values are transmitted in the slice header. In a variant, these default predictor values are transmitted in the sequence parameters.

These embodiments offer a better coding efficiency thanks to a better flexibility.

According to an embodiment, these default predictor values are pre-determined for all sequences.

5           According to an embodiment, the sign of the default value used to predict the edge offset is set, i.e. it is fixed. For instance, the predictor values for offsets O1 and O2 are positive and the predictor values for offsets O1 and O3 are negative.

          According to an embodiment, the absolute predictor values of the edge offsets associated with the first and last categories O1 and O4 are higher than absolute predictor  
10 values of the edge offsets associated with the other categories O2 and O3.

This embodiment may be advantageous notably because the absolute average values of offsets O1 and O4 are often larger than the absolute average values of offsets O2 and O3. Consequently, this embodiment may provide a better coding efficiency.

15           According to an embodiment, the absolute predictor values of the edge offsets associated with the first and last categories (O1 and O4) are the same and absolute predictor values of the edge offsets associated with the other categories (O2 and O3) are the same.

          Advantageously, the process is simplified and is in line with the statistics  
20 selection of offsets values.

**Figure 34** illustrates steps of a prediction process that may be performed during a (SAO) filtering embodiment at decoder side according to a second embodiment of the third aspect. As previously, these steps may be performed during the process shown in Figure 10 or 11.

25           According to this second embodiment, the default value depends on a sign parameter N to use in a sign function for determining an index of the edge offset.

          At the beginning of the process, a decoded offsets table (3401) is obtained. This table comprises offsets values with their respective sign. The value of the four offsets (3402) is then predicted (3403). This is done based on the predictor Predictor\_X\_j  
30 using a predictor table (3404) and the formula set[eo\_sign\_method\_X], which is based on the eo\_sign\_method\_X set table (3406). The predictor table 3404 may comprise different predictors adapted to each type of offsets (peak, valley etc..) while the table set (3406) is used to obtain the correct predictor for the current Offset\_X\_j. In this example, the predictor table has the value N for its first dimension. But if the table set is the same  
35 whatever the sequences, the conversion of the eo\_sign\_method\_X to the value N

according to the table set is not needed. In addition, if the `eo_sign_method_X` is not directly the N value, the table "set" may be used to convert the `eo_sign_method` to the correct N value.

Correspondingly, at encoder side, the correct Predictor `Predictor_X_j` is subtracted to the `Offset_X_j` value. The process then proceeds to step 3405.

An advantage of this embodiment may be a coding efficiency increase compared to a fix value for all edge offset sign methods. This is because the absolute offset values increase when the N value of the sign method increases.

According to embodiments, at least some edge offsets of the plurality the sign of which is signalled in the bitstream are not predicted, i.e. only some offsets may be predicted and the non-predicted offsets may have an explicit signalling of their signs. Figures 35, 36, 37 and 38 show four examples of such embodiments. In these examples, the steps are similar to the process shown in Figure 33 and/or 34 but they are performed only for some of the offsets.

**Figure 35** illustrates steps of a prediction process that may be performed during a (SAO) filtering embodiment at decoder side according to a third embodiment of the third aspect. As previously, these steps may be performed during the process shown in Figure 10 or 11.

According to this third embodiment, at least one edge offset is predicted from a sign parameter N to use in a sign function for determining an index of the edge offset. In addition, only some of the offsets are predicted.

For example, the offsets O1 and O4 are predicted by a predictor which has its absolute value equal to the value N of the sign method.

At the beginning of the process, a decoded offsets table (3501) is obtained. This table comprises offsets values with their respective sign. For each offset (3502), it is determined whether the offset is the peak or the valley offsets i.e. O1 or O4 (3507). If so, the offset is predicted by the value N (3503) obtained by the formula  $\text{set}[\text{eo\_sign\_method\_X}]$  based on the `eo_sign_method_X` set table (3506). According to the table shown in Figure 7B, this value N is only the absolute value of the predictor and the sign is positive if it is the offset O1 and negative if it is the offset O4. The process then proceeds to step 3505.

An advantage of this embodiment may be an increased coding efficiency.

According to other embodiments, other possible values for the absolute value of the predictor can be considered as N-1 or N-2 or N+1 or N+2, especially if the edge

offset sign method is the same for the whole frame or sequence (if there is no competition between edge offset sign methods).

**Figure 36** illustrates steps of a prediction process that may be performed during a (SAO) filtering embodiment at decoder side according to a fourth embodiment of the third aspect.

In this fourth embodiment, only the peak and valley offsets (O1 and O4) may be predicted. Thus, this process is similar to the process shown in Figure 33 but it is performed only for some of the offsets.

At the beginning of the process, a decoded offsets table (3601) is obtained. This table comprises offsets values with their respective sign. For each offset (3602), it is determined whether the offset is the peak or the valley offsets i.e. O1 or O4 (3607). If so, a variable  $\text{Offset\_X\_j}$  of the component X (YUV) is added to the predictor  $\text{Predictor\_X\_j}$  (3603) using the predictor table (3604). The process then proceeds to step 3605.

An advantage of this embodiment may be additional coding efficiency. This is because the absolute average value of offset O1 and O4 is in average larger than offsets O2 and O3. In addition, the absolute values for O2 and O3 are often inferior to 1. Consequently, there may be no need to predict them and to signal a sign.

**Figure 37** illustrates steps of a prediction process that may be performed during a (SAO) filtering embodiment at decoder side according to a fifth embodiment of the third aspect.

In this fifth embodiment, the offsets for the luma component only are predicted.

At the beginning of the process, a decoded offsets table (3701) is obtained. This table comprises offsets values with their respective sign. For each offset (3702), it is determined whether the current colour component is luma (3708). If so, the offset is predicted by adding a variable  $\text{Offset\_X\_j}$  of the component X (YUV) to the predictor  $\text{Predictor\_X\_j}$  (3703) using the predictor table (3704). The process then proceeds to step 3705.

An advantage of this embodiment may be additional coding efficiency. This is because the absolute offset values are generally higher for luma than for chroma components. The prediction of Luma component may be efficient enough. This embodiment may thus provide a simplification as the prediction of Chroma offsets is avoided.

**Figure 38** illustrates steps of a prediction process that may be performed during a (SAO) filtering embodiment at decoder side according to a sixth embodiment of the third aspect.

In this sixth embodiment, only the offsets for Luma component and offsets  
5 O1 and O4 are predicted.

At the beginning of the process, a decoded offsets table (3801) is obtained. This table comprises offsets values with their respective sign. For each offset (3802), it is determined whether the current colour component is luma (3808). If so, it is determined whether the offset is the peak or the valley offsets i.e. O1 or O4 (3807). If so, the offset  
10 is predicted by adding a variable  $Offset\_X\_j$  of the component X (YUV) to the predictor  $Predictor\_X\_j$  (3803) using the predictor table (3804). The process then proceeds to step 3805.

According to other embodiments, the at least one edge offset is predicted from a neighbouring edge offset value, only if the sign of the neighbouring edge offset is  
15 determined using the same sign method. In other words, at least some of the current edge offsets values may be predicted based on edge offset values of one of previously encoded CTU parameters, if the sign of the previous offset is determined using the same edge offset sign method.

#### 20 **Fourth group of embodiments (fourth aspect)**

According to a fourth aspect of the invention, there is provided a method for obtaining an edge offset for performing a Sample Adaptive Offset (SAO) filtering, wherein the sign of the at least one edge offset is implicit and the minimum value of the edge  
25 offset is set to a default value different from zero.

In other words, the fourth aspect provides a method of performing sample adaptive offset (SAO) filtering on an image comprising a plurality of image parts, the method comprising:

obtaining a plurality of edge offsets to perform a SAO filtering;  
30 wherein obtaining a plurality of edge offsets comprises predicting at least one edge offset; and

wherein a sign of the at least one edge offset is implicit and the minimum value of the edge offset is set to a default value different from zero.

According to one embodiment of this fourth aspect, the offsets for Edge offset  
35 classification have a minimum limit absolute value. In this embodiment, for the offsets

which have a minimum limit absolute value superior to 0, the sign of the offsets are not signalled in the bitstream.

One may note that in the current implementation of SAO, the Edge offsets are bounded as the following:

5  $O1 \geq 0$   
 $O2 \geq 0$   
 $O3 \leq 0$   
 $O4 \leq 0$

10 According to the proposed embodiment, the offsets O1, O2, O3, O4 have the minimum limit absolute values MAO1, MAO2, MAO3, MAO4 respectively. For example, the offset values O1, O2, O3, O4 are bounded as follows:

$O1 \geq MAO1$   
 $O2 \geq MAO2$   
 $O3 \leq -MAO3$   
15  $O4 \leq -MAO4$

This bound aims at limiting the rate related to offset. The absolute offset value Offset\_X\_j (607) shown in Figure 10 may be always coded from 0 to a maximum value as in the current SAO implementation. Consequently, at encoder side the absolute value Offset\_X\_j may be predicted by respectively MAO1\_X, MAO2\_X, -MAO3\_X,  
20 - MAO4\_X.

At decoder side, the same process as shown in Figure 33 is applied with the predictors Predictor\_X\_j which may be set as the following:

$Predictor\_X\_1 = MAO1$   
 $Predictor\_X\_2 = MAO2$   
25  $Predictor\_X\_3 = -MAO3$   
 $Predictor\_X\_4 = -MAO4$

An advantage of this embodiment may be a coding efficiency increase compared to the prior art. As it does not require adding a sign of the offset as a syntax element, this embodiment may be considered as a simplification compared to  
30 embodiments in which the sign is transmitted.

**Figure 39** shows a system 191 195 comprising at least one of an encoder 150 or a decoder 100 and a communication network 199 according to embodiments of the present invention. According to an embodiment, the system 195 is for processing and providing a content (for example, a video and audio content for displaying/outputting or  
35 streaming video/audio content) to a user, who has access to the decoder 100, for

example through a user interface of a user terminal comprising the decoder 100 or a user terminal that is communicable with the decoder 100. Such a user terminal may be a computer, a mobile phone, a tablet or any other type of a device capable of providing/displaying the (provided/streamed) content to the user. The system 195  
5 obtains/receives a bitstream 101 (in the form of a continuous stream or a signal – e.g. while earlier video/audio are being displayed/output) via the communication network 199. According to an embodiment, the system 191 is for processing a content and storing the processed content, for example a video and audio content processed for displaying/outputting/streaming later. The system 191 obtains/receives a content  
10 comprising an original sequence of images 151, which is received and processed (including filtering with a deblocking filter according to the present invention) by the encoder 150, and the encoder 150 generates a bitstream 101 that is to be communicated to the decoder 100 via a communication network 191. The bitstream 101 is then communicated to the decoder 100 in a number of ways, for example it may be generated  
15 in advance by the encoder 150 and stored as data in a storage apparatus in the communication network 199 (e.g. on a server or a cloud storage) until a user requests the content (i.e. the bitstream data) from the storage apparatus, at which point the data is communicated/streamed to the decoder 100 from the storage apparatus. The system 191 may also comprise a content providing apparatus for providing/streaming, to the  
20 user (e.g. by communicating data for a user interface to be displayed on a user terminal), content information for the content stored in the storage apparatus (e.g. the title of the content and other meta/storage location data for identifying, selecting and requesting the content), and for receiving and processing a user request for a content so that the requested content can be delivered/streamed from the storage apparatus to the user  
25 terminal. Alternatively, the encoder 150 generates the bitstream 101 and communicates/streams it directly to the decoder 100 as and when the user requests the content. The decoder 100 then receives the bitstream 101 (or a signal) and performs filtering with a deblocking filter according to the invention to obtain/generate a video signal 109 and/or audio signal, which is then used by a user terminal to provide the  
30 requested content to the user.

In the preceding embodiments, the functions described may be implemented in hardware, software, firmware, or any combination thereof. If implemented in software, the functions may be stored on or transmitted over, as one or more instructions or code, a computer-readable medium and executed by a hardware-based processing unit.

Computer-readable media may include computer-readable storage media, which corresponds to a tangible medium such as data storage media, or communication media including any medium that facilitates transfer of a computer program from one place to another, e.g., according to a communication protocol. In this manner, computer-readable  
5 media generally may correspond to (1) tangible computer-readable storage media, which is non-transitory, or (2) a communication medium such as a signal or carrier wave. Data storage media may be any available media that can be accessed by one or more computers or one or more processors to retrieve instructions, code and/or data structures for implementation of the techniques described in this disclosure. A computer program  
10 product may include a computer-readable medium.

By way of example, and not limitation, such computer-readable storage media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage, or other magnetic storage devices, flash memory, or any other medium that can be used to store desired program code in the form of instructions or data structures  
15 and that can be accessed by a computer. In addition, any connection is properly termed a computer-readable medium. For example, if instructions are transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio, and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless  
20 technologies such as infrared, radio, and microwave are included in the definition of medium. It should be understood, however, that computer-readable storage media and data storage media do not include connections, carrier waves, signals, or other transient media, but are instead directed to non-transient, tangible storage media. Disk and disc, as used herein, includes compact disc (CD), laser disc, optical disc, digital versatile disc  
25 (DVD), floppy disk and Blu-ray disc, where disks usually reproduce data magnetically, while discs reproduce data optically with lasers. Combinations of the above should also be included within the scope of computer-readable media.

Instructions may be executed by one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, application specific  
30 integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry. Accordingly, the term "processor," as used herein may refer to any of the foregoing structure or any other structure suitable for implementation of the techniques described herein. In addition, in some aspects, the functionality described herein may be provided within dedicated hardware and/or  
35 software modules configured for encoding and decoding, or incorporated in a combined

codec. In addition, the techniques could be fully implemented in one or more circuits or logic elements.

The techniques of this disclosure may be implemented in a wide variety of devices or apparatuses, including a wireless handset, an integrated circuit (IC) or a set of ICs (e.g., a chip set). Various components, modules, or units are described in this disclosure to emphasize functional aspects of devices configured to perform the disclosed techniques, but do not necessarily require realization by different hardware units. Rather, as described above, various units may be combined in a codec hardware unit or provided by a collection of interoperative hardware units, including one or more processors as described above, in conjunction with suitable software and/or firmware.

While the invention has been illustrated and described in detail in the drawings and foregoing description, such illustration and description are to be considered illustrative or exemplary and not restrictive, the invention being not restricted to the disclosed embodiment. Other variations to the disclosed embodiments can be understood and effected by those skilled in the art in putting into practice (i.e. performing) the claimed invention, from a study of the drawings, the disclosure and the appended claims.

Some embodiments of the first, second, third and fourth aspects may be combined.

For instance, the sign of the offsets O1 and O4 (peak and valley) may be transmitted in the bitstream for Luma and Chroma components. For instance, the sign of the offsets O1 and O4 (peak and valley) may be transmitted in the bitstream for Luma and Chroma components. In addition, the absolute predictor value may be set equal to the value N of the sign method for the offset O1 and O4 (peak and valley).

In another embodiment, the sign method may be also transmitted with a table set corresponding to the following set of N values: {0, 2, 4, 8}.

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. A single processor or other unit may fulfil the functions of several items recited in the claims. 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. Any reference signs in the claims should not be construed as limiting the scope of the invention.

CLAIMS

1. A method of performing sample adaptive offset (SAO) filtering on an image  
5 comprising a plurality of image parts, the method comprising:  
    obtaining an edge offset to perform a SAO filtering;  
    applying an edge offset classification to at least one pixel of the image using a  
    predetermined sign method for determining an index of the edge offset;  
    wherein the predetermined sign method is signalled in the bitstream.  
10
2. A method according to claim 1, wherein the predetermined sign method is signalled  
in the bitstream at CTU, frame, slice or sequence level.  
15
3. A method according to any one of the preceding claims, wherein the predetermined  
sign method is predicted.  
20
4. A method according to any one of the preceding claims, wherein the predetermined  
sign method defines a sign function taking into account a number N corresponding  
to a limit value.  
25
5. A method according to claim 4, wherein the predetermined sign method is selected  
at the encoder side based on a plurality of limit values N that are not consecutive.  
30
6. A method according to claim 4, wherein the predetermined sign method is selected  
at the encoder side based on a plurality of limit values N that depend on the slice  
type.  
35
7. A method according to claim 4, wherein the predetermined sign method is selected  
at the encoder side based on a plurality of limit values N that depend on color  
components.

8. A method according to claim 4, wherein the predetermined sign method is selected at the encoder side based on a plurality of limit values N that depend on a Luma component only.
- 5 9. A method according to claim 4, wherein the predetermined sign method is selected at the encoder side based on a plurality of limit values N that are signalled in a slice header.
10. A method according to claim 4, wherein the predetermined sign method is selected  
10 at the encoder side based on a plurality of limit values N.
11. A method of providing an edge offset to perform a sample adaptive offset (SAO) filtering on an image, the method comprising:  
computing an edge offset to perform the SAO filtering;  
15 selecting a sign method for determining an index of the edge offset among a plurality of possible sign methods; and  
signalling the selected sign method in the bitstream thereby allowing the edge offset classification to be performed according to the selected sign method.  
20
12. A method according to claim 11, wherein the sign method is signalled in the bitstream at CTU, frame, slice or sequence level.
- 25 13. A method according to any one of claims 11 to 12, wherein the sign method is predicted.
14. A method according to any one of claims 11 to 13, wherein the sign method is selected according to parameters of a current slice or frame.  
30
15. A method according to any one of claims 11 to 14, wherein the sign method defines a sign function taking into account a number N corresponding to a limit value.
- 35 16. A method of performing sample adaptive offset (SAO) filtering on an image comprising a plurality of image parts, the method comprising:

obtaining a plurality of edge offsets to perform a SAO filtering;  
wherein a sign of at least one edge offset of the plurality is signalled in the  
bitstream.

- 5 17. A method according to claim 16, wherein the sign of the edge offset is signalled for  
all the edge offsets of the plurality for which a sign parameter N to use in a sign  
function for determining an index of the edge offset is positive.
- 10 18. A method according to claim 16, wherein the sign of the edge offset is signalled for  
all the edge offsets of the plurality if all the sign parameters N to use in a sign function  
for determining an index of the edge offsets are higher than a predetermined  
threshold.
- 15 19. A method according to claim 16, wherein the at least one edge offset the sign of  
which is signalled belongs to the peak and valley configurations.
- 20 20. A method according to claim 16, wherein the sign of the at least one edge offset is  
signalled whatever is a predetermined sign method used to determine an index of  
the edge offset.
21. A method according to claim 16, wherein the sign of the edge offset is signalled only  
for the edge offsets for which the sign parameters N to use for determining an index  
of these edge offsets are higher than a predetermined threshold.
- 25 22. A method according to claim 16, wherein the sign of the edge offset is signalled only  
for the edge offsets for which the sign parameters N to use for determining an index  
of these edge offsets are different from zero.
- 30 23. A method according to any one of claims 16 to 22, wherein the sign of the edge offset  
is signalled only for the luma component.
24. A method of providing an edge offset to perform a sample adaptive offset (SAO)  
filtering on an image, the method comprising:  
obtaining a plurality of edge offsets to perform the SAO filtering;  
35 computing an index of the edge offsets by applying, for each edge offset, a sign  
method to a sign parameter N; and

signalling a sign of at least one edge offset of the plurality in the bitstream.

25. A method of performing sample adaptive offset (SAO) filtering on an image comprising a plurality of image parts, the method comprising:
- 5       obtaining a plurality of edge offsets to perform a SAO filtering;  
      wherein a sign of at least one edge offset of the plurality is signalled in the bitstream; and  
      wherein obtaining a plurality of edge offsets comprises predicting at least one edge offset the sign of which is signalled in the bitstream.
- 10
26. A method according to claim 25, wherein the at least one edge offset is predicted from a default value.
27. A method according to claim 26, wherein the default value is transmitted in the
- 15       bitstream.
28. A method according to claim 26, wherein the default value is predetermined for at least one sequence.
- 20
29. A method according to claim 26, further comprising setting the sign of the default value.
30. A method according to claim 26, wherein the default value depends on a sign parameter N to use in a sign function for determining an index of the edge offset.
- 25
31. A method according to claim 25, further comprising:
- applying an edge offset classification to pixels of the image in order to classify the pixels in four categories, each category being associated with a given edge offset, and  
      wherein absolute predictor values of the edge offsets associated with the first and
- 30       last categories corresponding to the peak and valley are higher than absolute predictor values of the edge offsets associated with the other categories.
32. A method according to claim 25, further comprising:
- applying an edge offset classification to pixels of the image in order to classify the
- 35       pixels in four categories, each category being associated with a given edge offset, and

wherein absolute predictor values of the edge offsets associated with the first and last categories corresponding to the peak and valley are the same.

33. A method according to claim 32, wherein absolute predictor values of the edge  
5 offsets associated with the other categories are the same.

34. A method according to claim 31 or 32, wherein the edge offset classification takes  
into account the sign of at least one edge offset of the plurality which is signalled in  
the bitstream.  
10

35. A method according to claim 25, wherein the at least one edge offset is predicted  
from a sign parameter N to use in a sign function for determining an index of the edge  
offset.

15 36. A method according to claim 25, wherein at least some edge offsets of the plurality  
the sign of which is signalled in the bitstream are not predicted.

37. A method according to claim 25, further comprising:  
applying an edge offset classification to pixels of the image in order to classify the  
20 pixels in four categories, each category being associated with a given edge offset, and  
wherein only the edge offsets associated with the first and last  
categories corresponding to the peak and valley are predicted.

38. A method according to claim 25, wherein the edge offsets are predicted for the luma  
25 component only.

39. A method according to claim 25, further comprising:  
applying an edge offset classification to pixels of the image in order to classify the  
pixels in four categories, each category being associated with a given edge offset, and  
30 wherein only the edge offsets associated with the first and last categories  
corresponding to the peak and valley, and the edge offsets for the luma component are  
predicted.

40. A method according to claim 25, wherein the at least one edge offset is predicted  
35 from a neighbouring edge offset value.

41. A method according to claim 25, wherein the at least one edge offset is predicted from a neighbouring edge offset value only if the neighbouring edge offset is determined using the same sign method.
- 5 42. A method of providing an edge offset to perform a sample adaptive offset (SAO) filtering on an image, the method comprising:  
obtaining a plurality of edge offsets to perform the SAO filtering;  
obtaining, for each edge offset, a sign of the edge offset;  
wherein the method further comprises signalling the sign of at least one edge  
10 offset of the plurality in the bitstream; and  
wherein obtaining a plurality of edge offsets comprises predicting at least one edge offset the sign of which is signalled in the bitstream.
- 15 43. A method of performing sample adaptive offset (SAO) filtering on an image comprising a plurality of image parts, the method comprising:  
obtaining a plurality of edge offsets to perform a SAO filtering;  
wherein obtaining a plurality of edge offsets comprises predicting at least one  
edge offset; and  
20 wherein a sign of the at least one edge offset is implicit and the minimum value of the edge offset is set to a default value different from zero.
44. A method of providing an edge offset to perform a sample adaptive offset (SAO) filtering on an image, the method comprising:  
25 computing a plurality of edge offsets to perform the SAO filtering;  
wherein computing a plurality of edge offsets comprises predicting at least one edge offset, and  
wherein a sign of the at least one edge offset is implicit and the minimum value of the edge offset is set to a default value different from zero.  
30
45. A method of encoding an image comprising performing sample adaptive offset (SAO) filtering using the method of any one of claims 11, 24, 42 or 44.
- 35 46. A method of decoding an image comprising performing sample adaptive offset (SAO) filtering using the method of any one of claims 1, 16, 25 or 43.

47. A device for performing sample adaptive offset (SAO) filtering on an image comprising a plurality of image parts, the device comprising a microprocessor configured for performing the steps of a method according to any one of claims 11, 24, 42 or 44.  
5
48. A device for performing sample adaptive offset (SAO) filtering on an image comprising a plurality of image parts, the device comprising a microprocessor configured for performing the steps of a method according to any one of claims 1, 16, 25 or 43.  
10
49. An encoder comprising the device of claim 46.
50. A decoder comprising the device of claim 47.  
15
51. A program which, when executed by a computer or processor, causes the computer or processor to carry out the method of any one of claims 1, 11, 16, 24, 25, 42, 43, 44.
- 20 52. A system comprising an encoder according to claim 49 and a decoder according to claim 50 configured to communicate through a communication network.

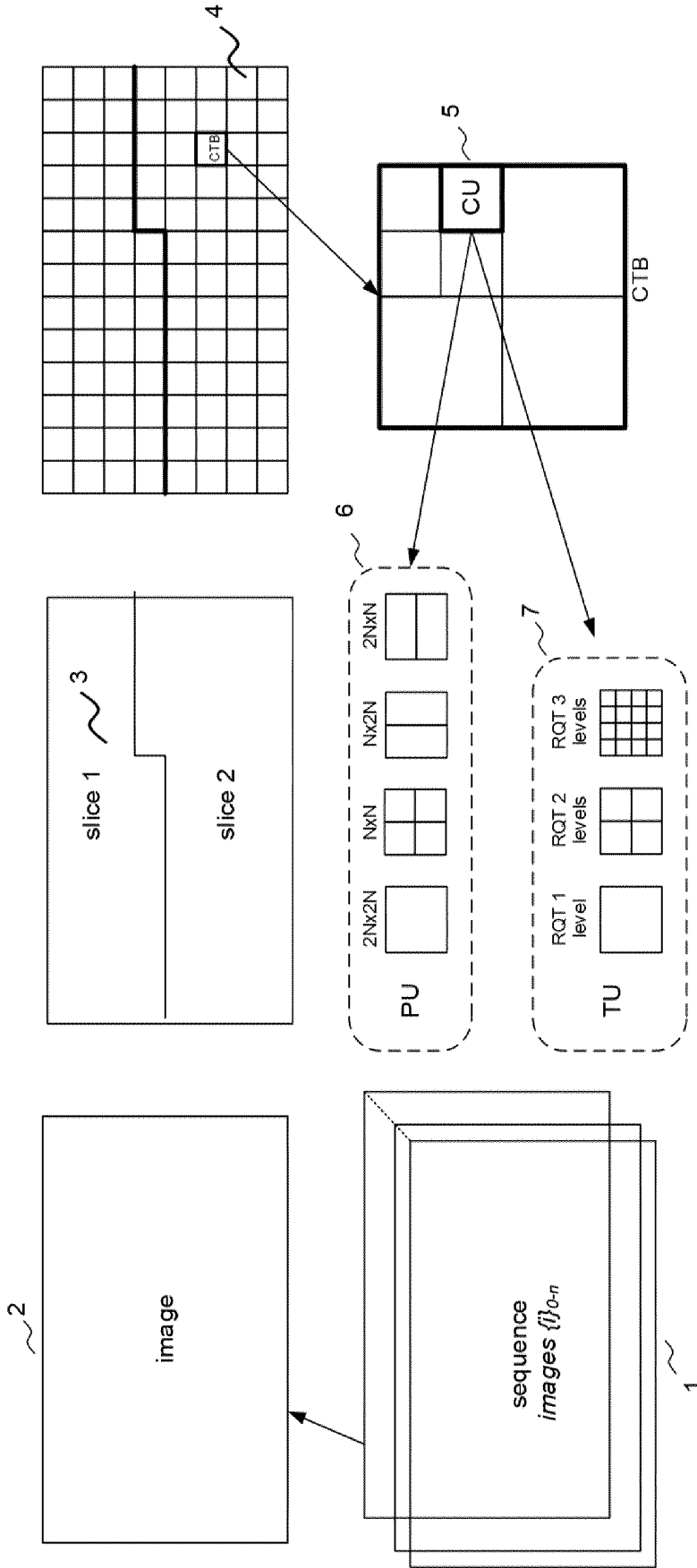


Figure 1

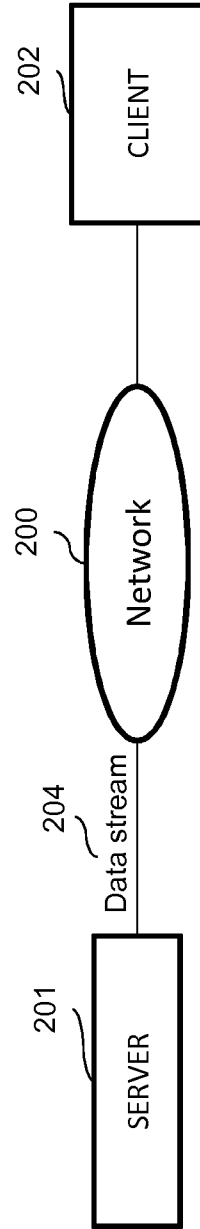


Figure 2

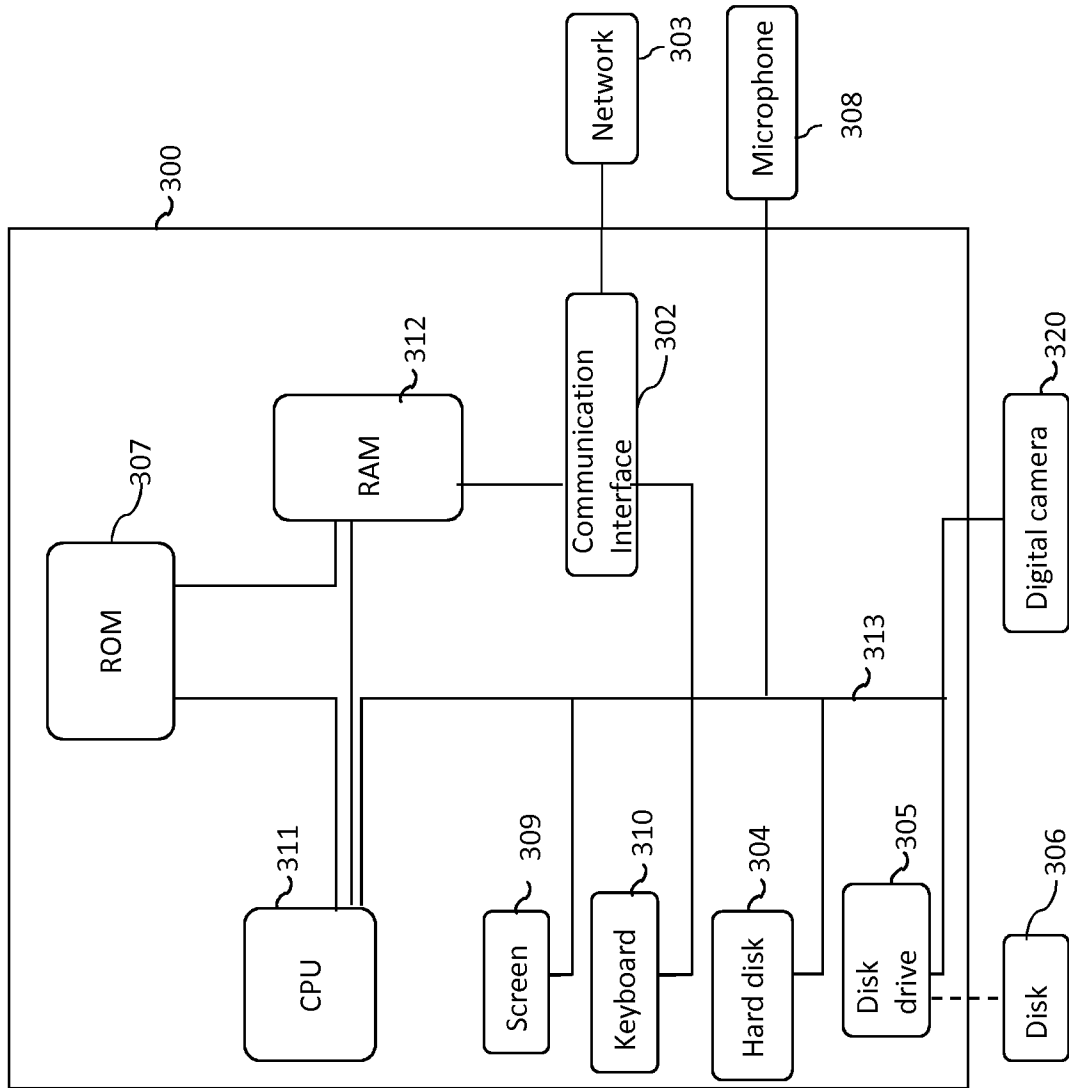


Figure 3

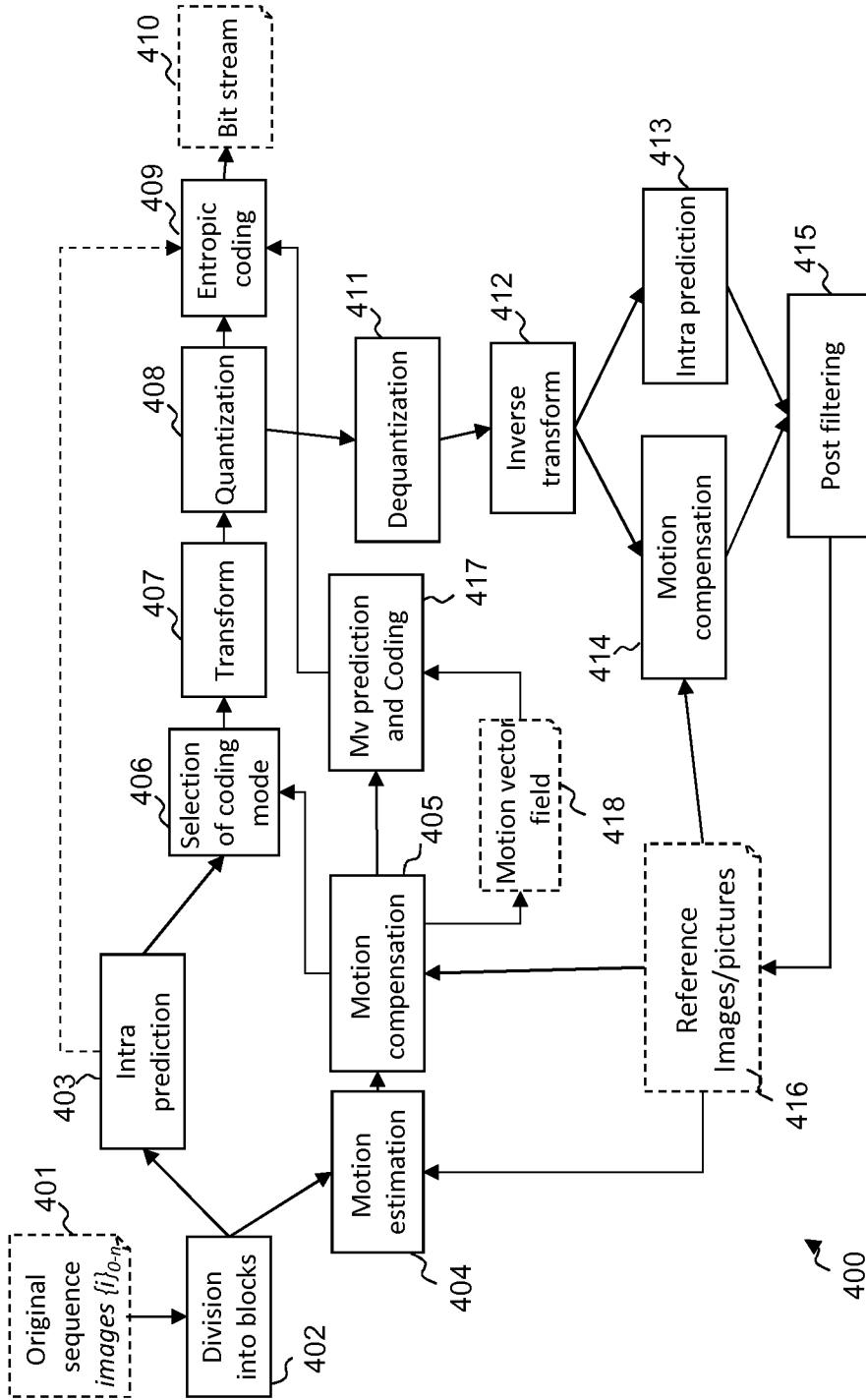
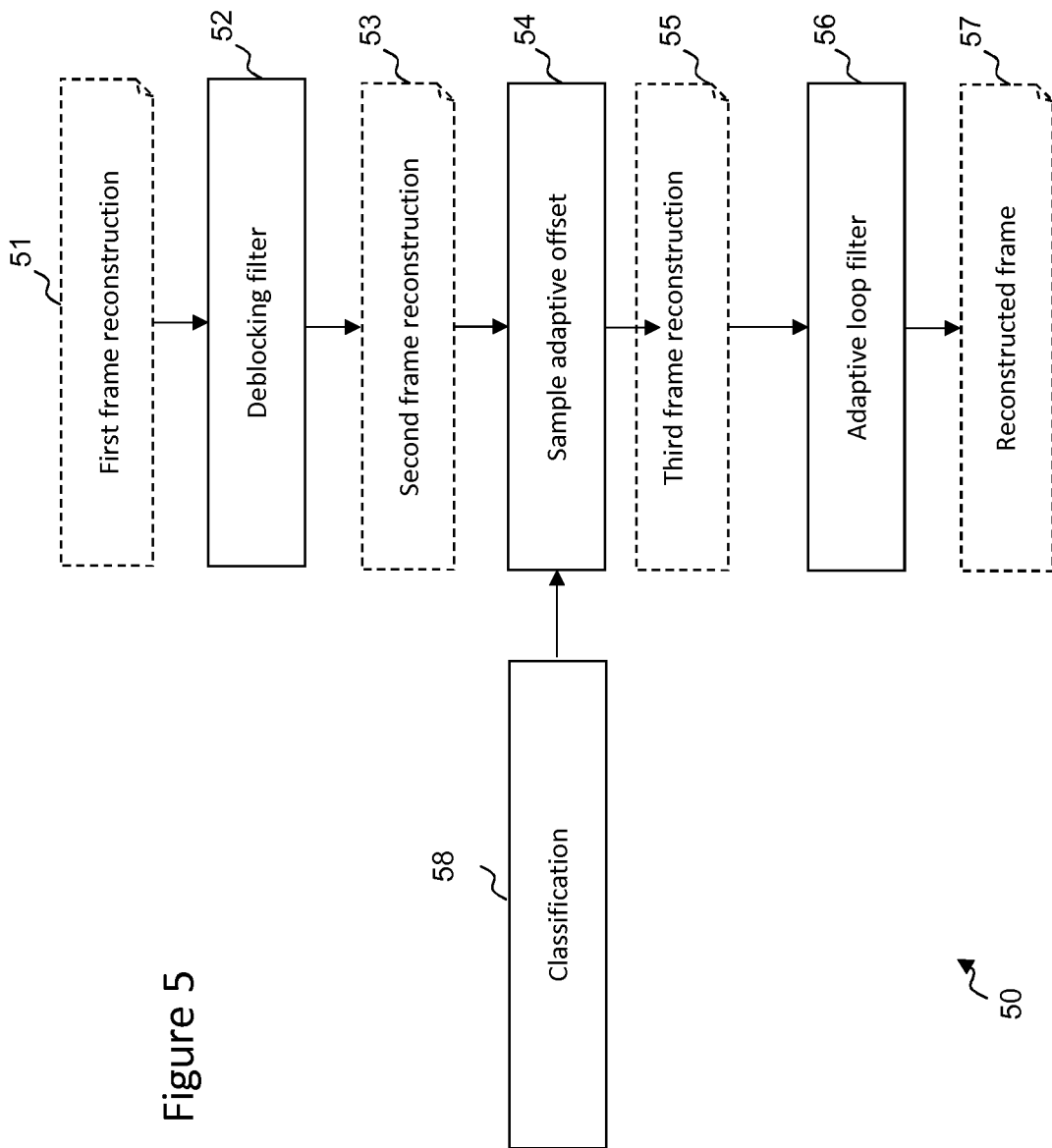


Figure 4



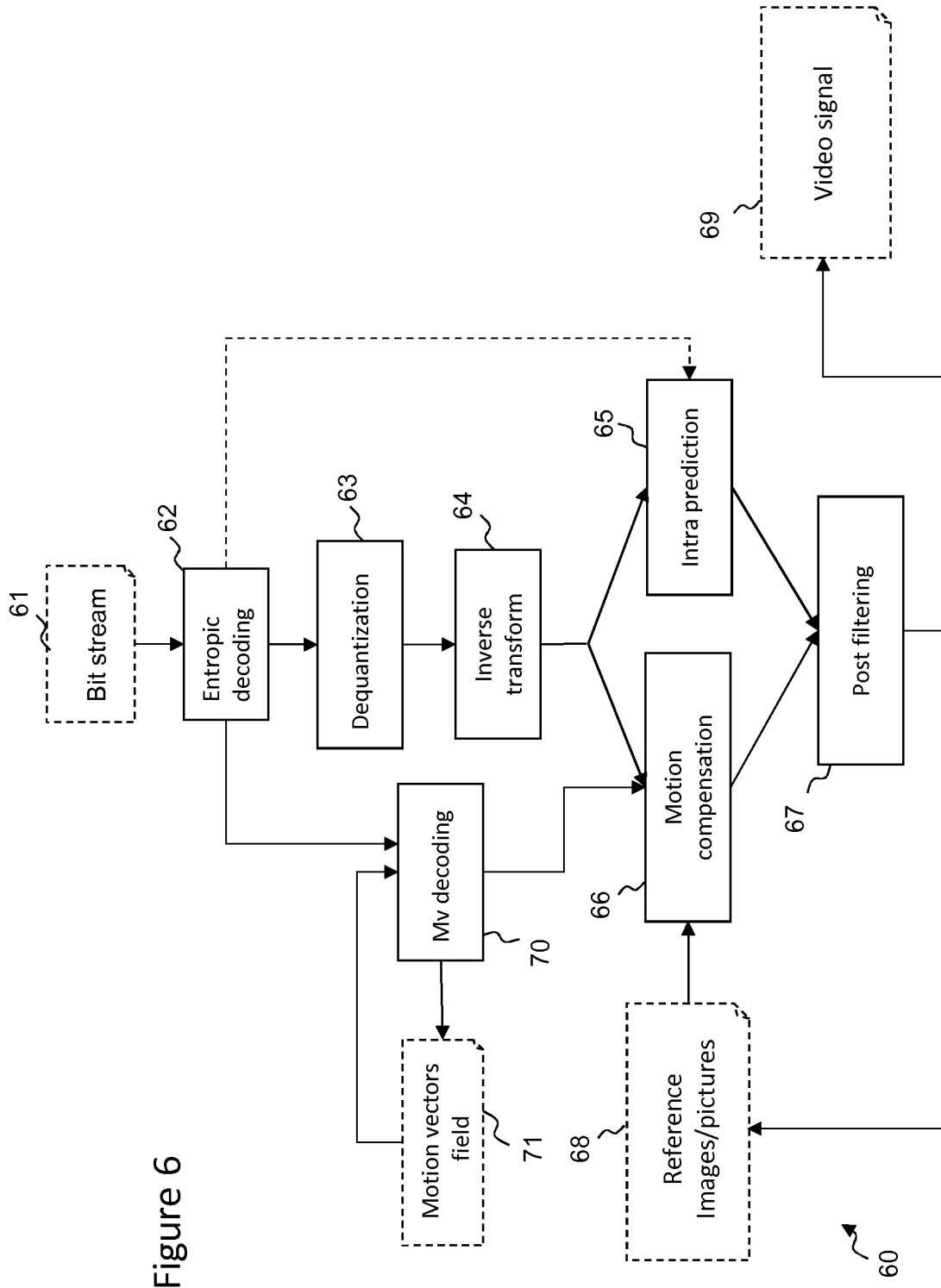


Figure 6

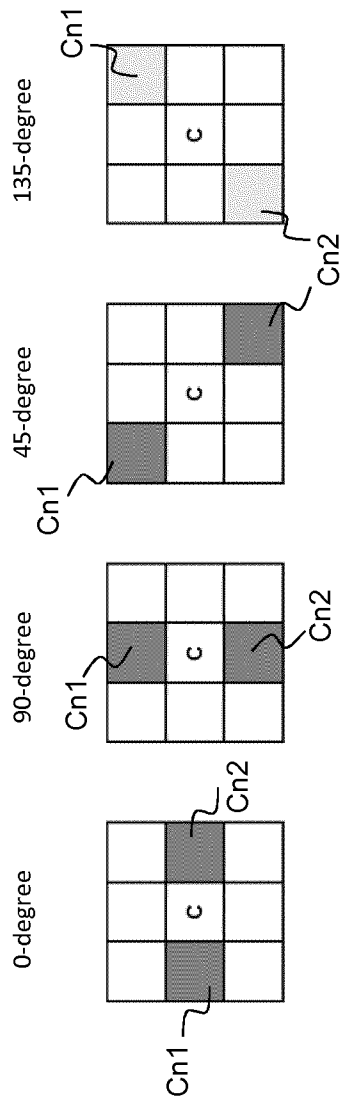


Figure 7A


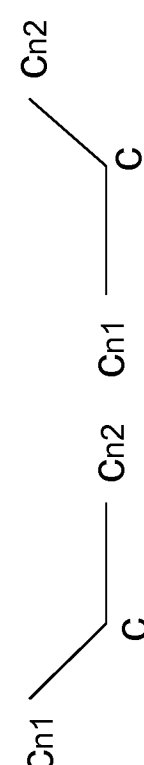
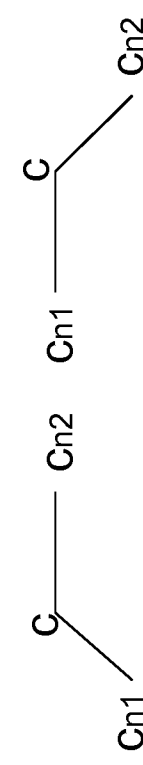
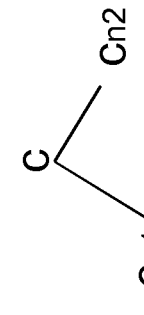
Edge Index	Conditions	Signal type	Offset
0	C<2 neighboring pixels		+ 01
1	C<1 neighbor and C==1 neighbor		+ 02
3	C>1 neighbor and C==1 neighbor		- 03
4	C>2 neighboring pixels		- 04
2	None of the above		

Figure 7B

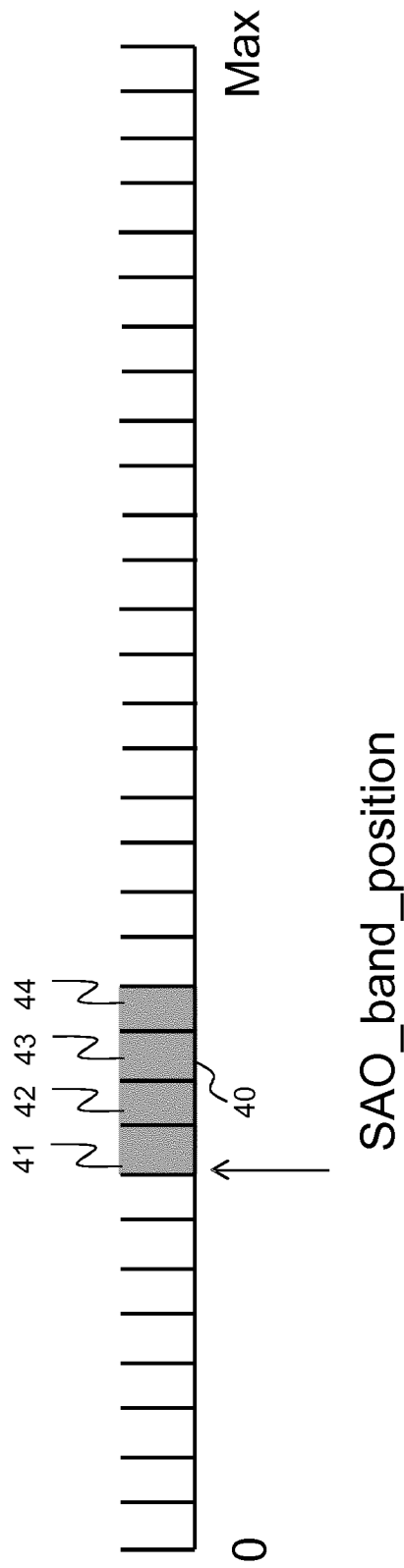


Figure 8

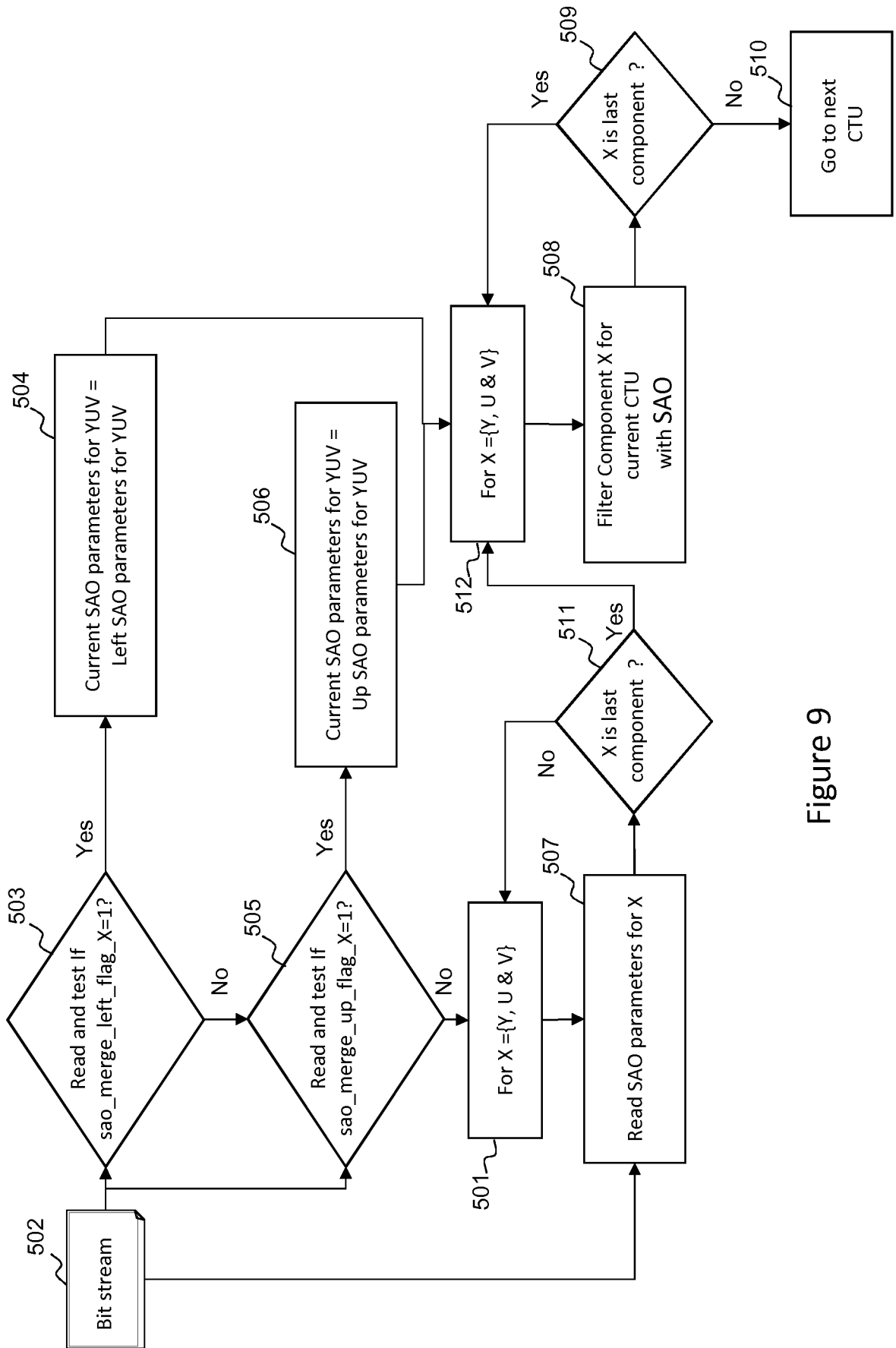
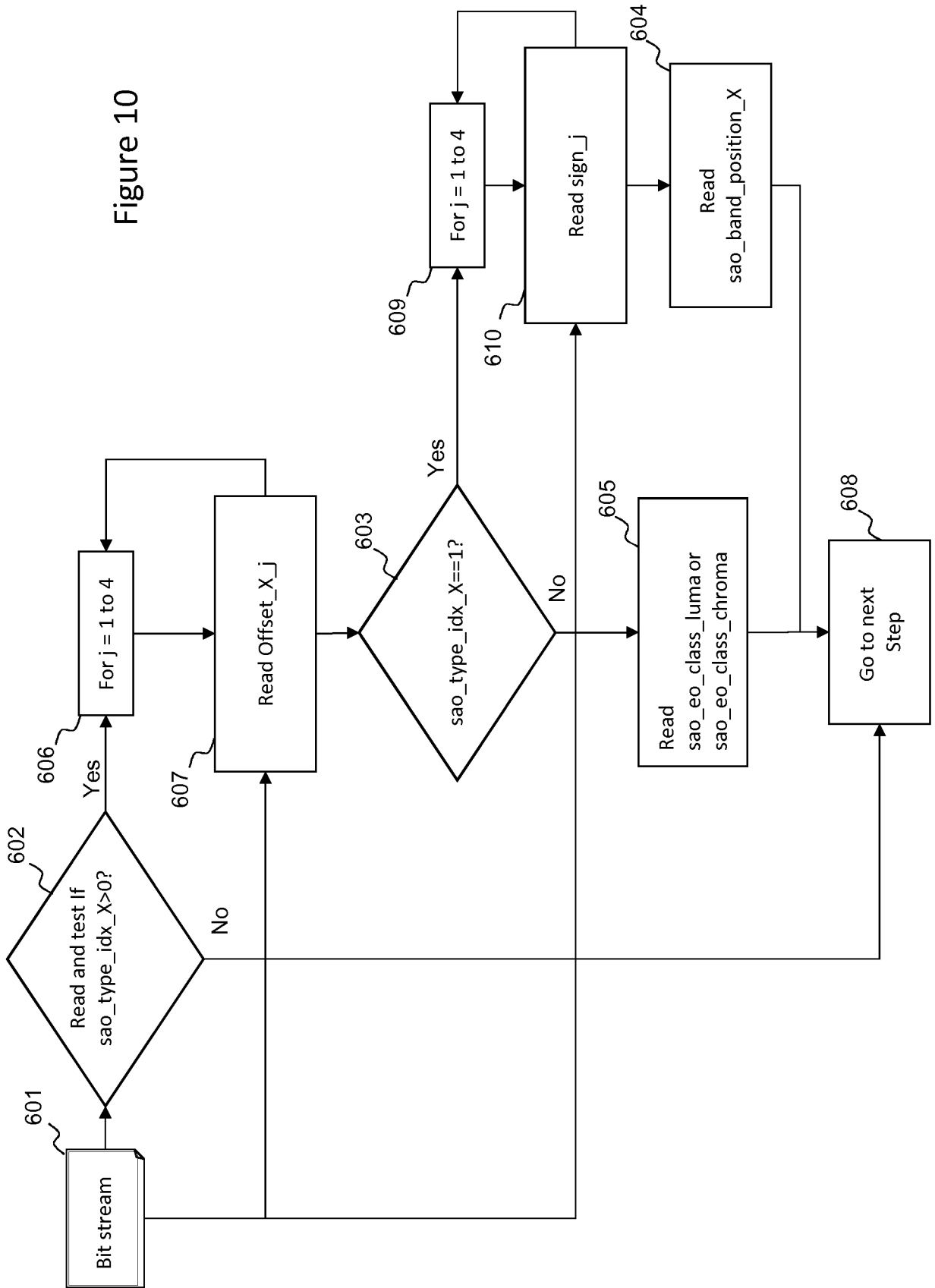


Figure 9

Figure 10



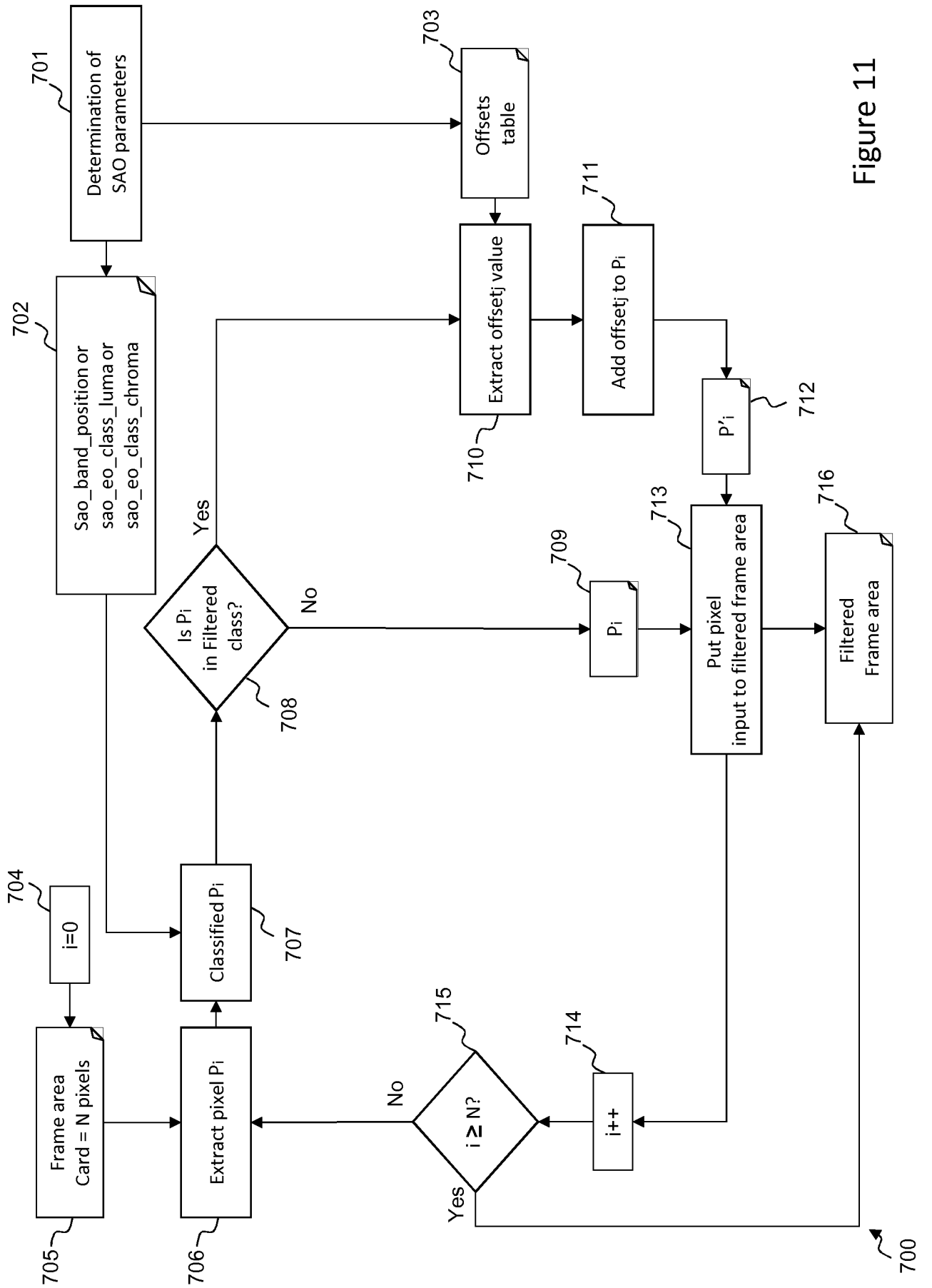


Figure 11

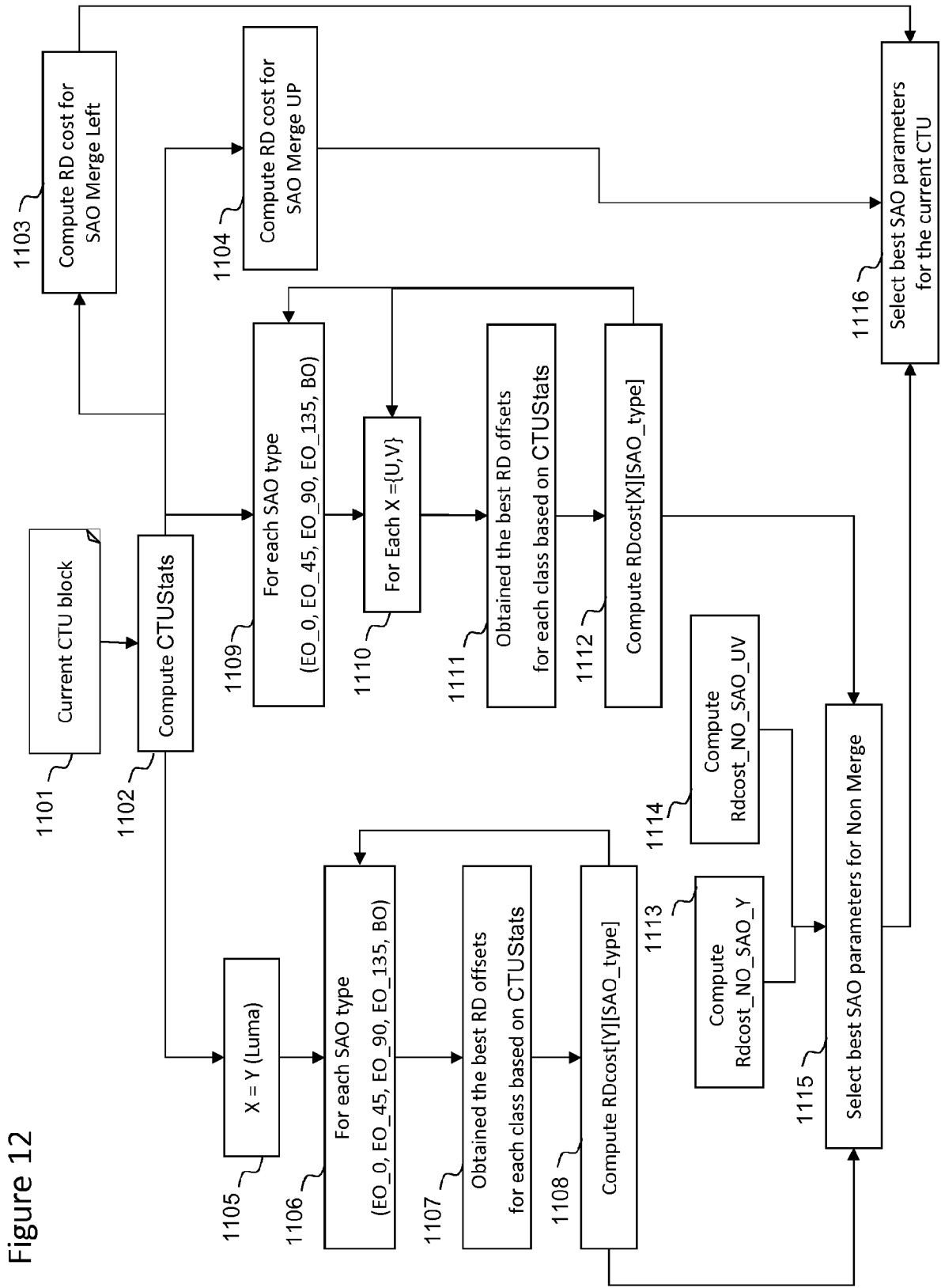


Figure 12

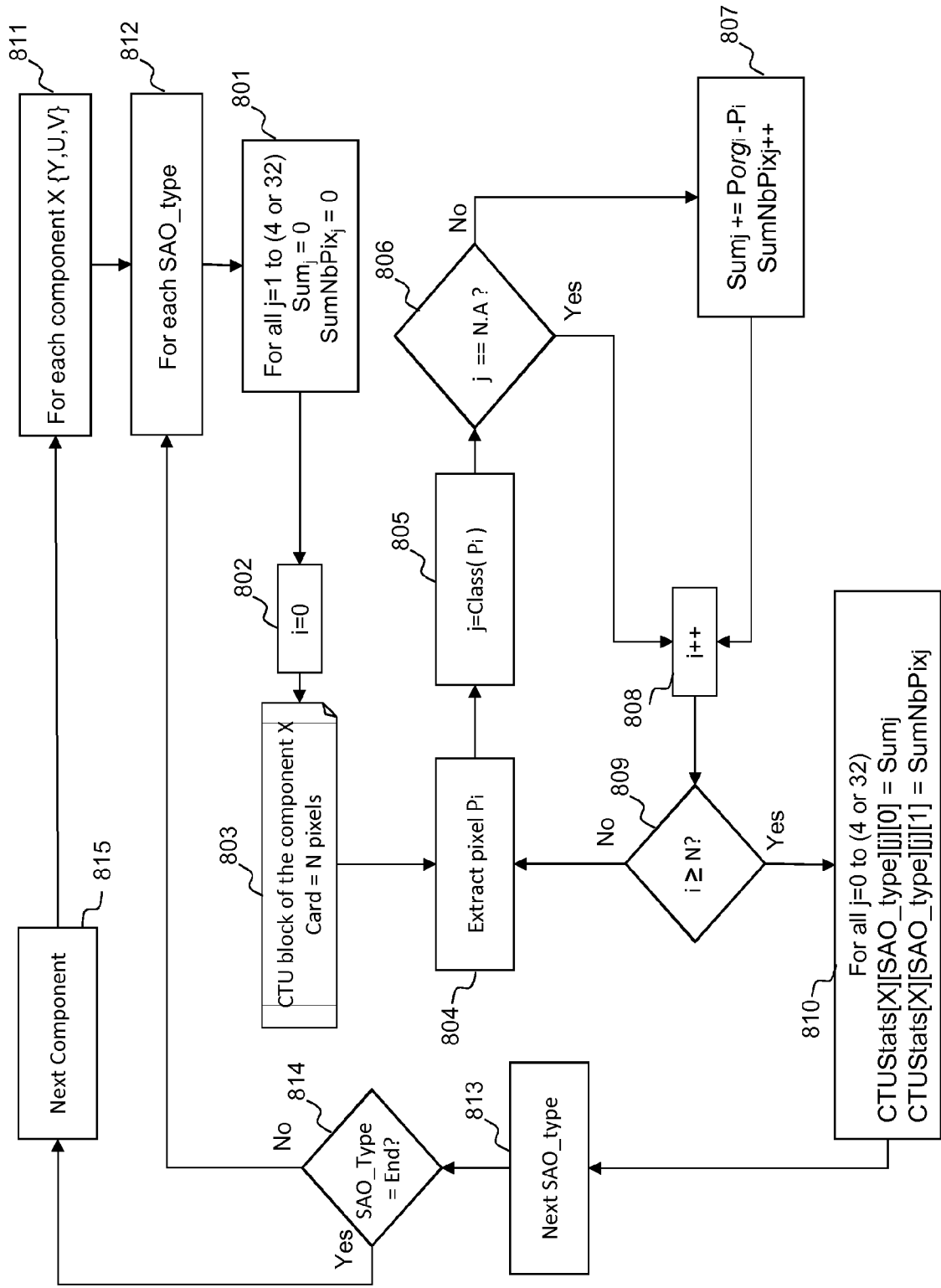


Figure 13

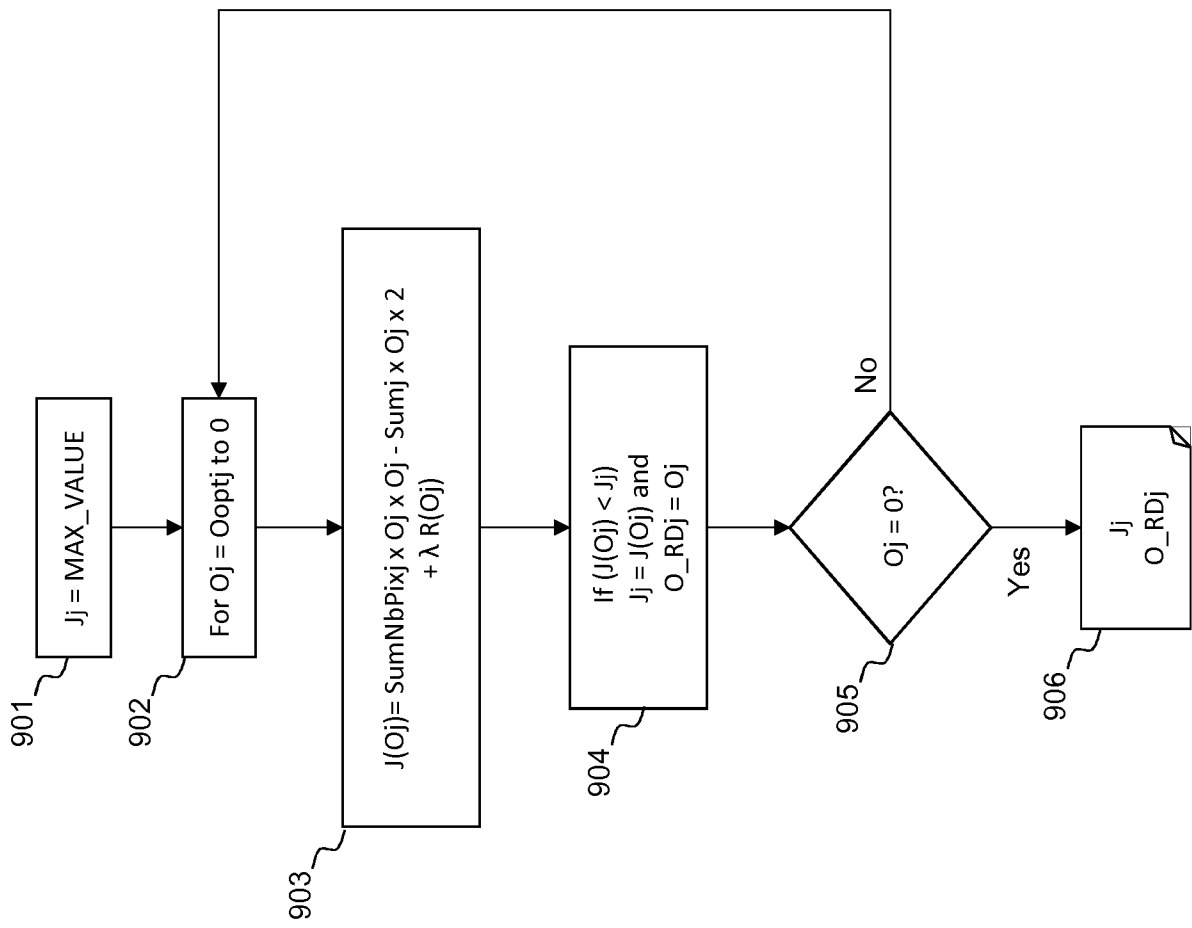


Figure 14

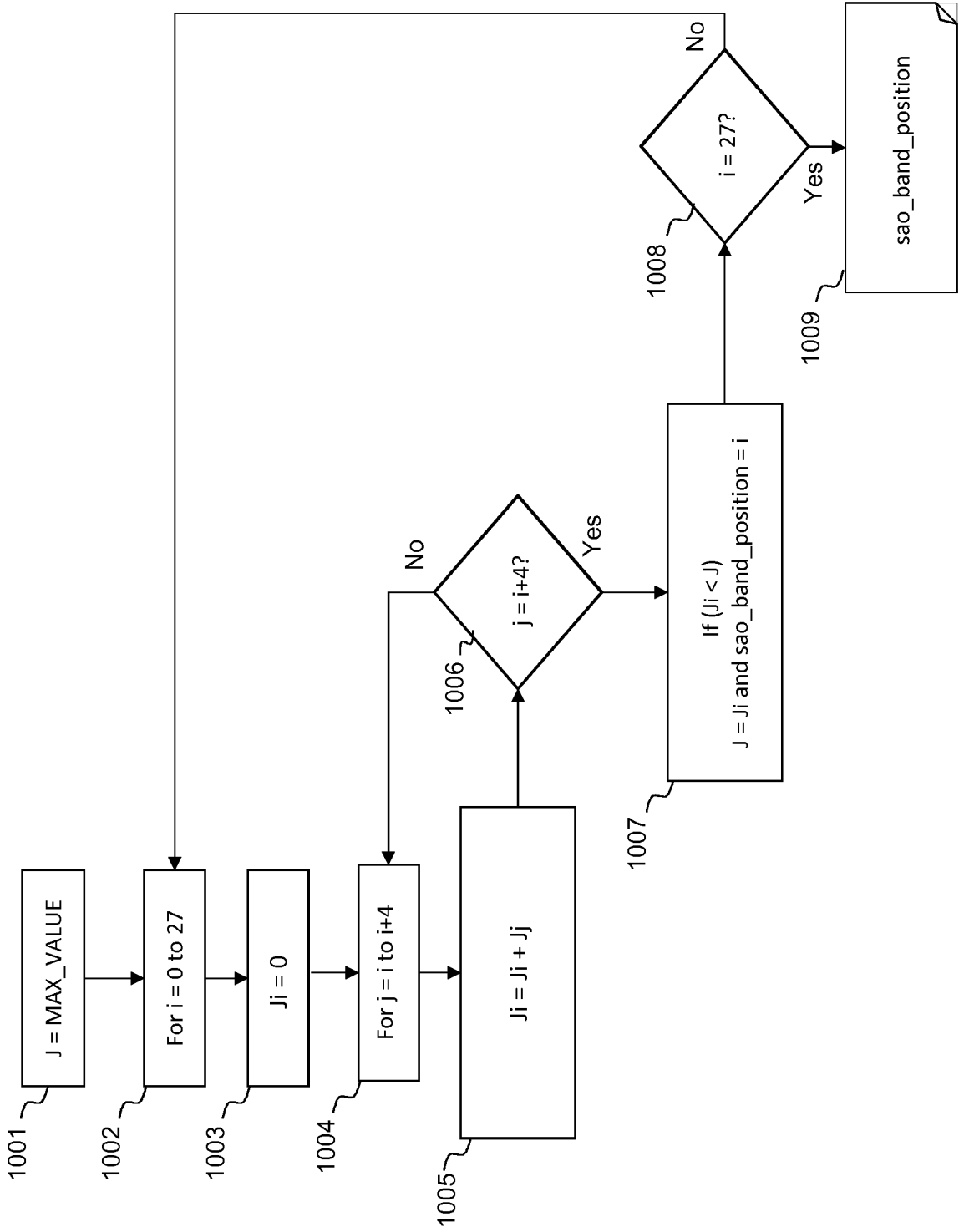


Figure 15

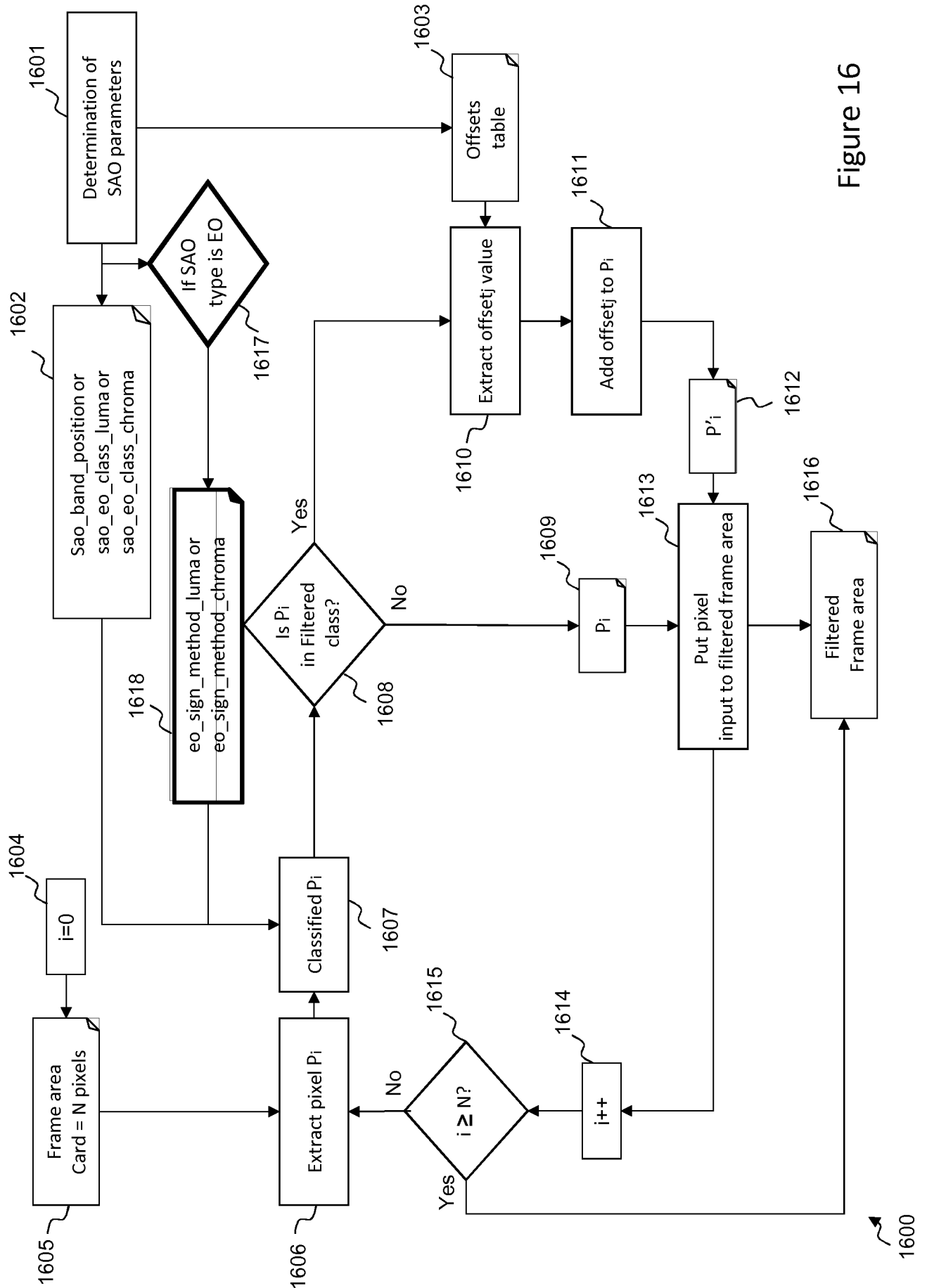


Figure 16

Figure 17

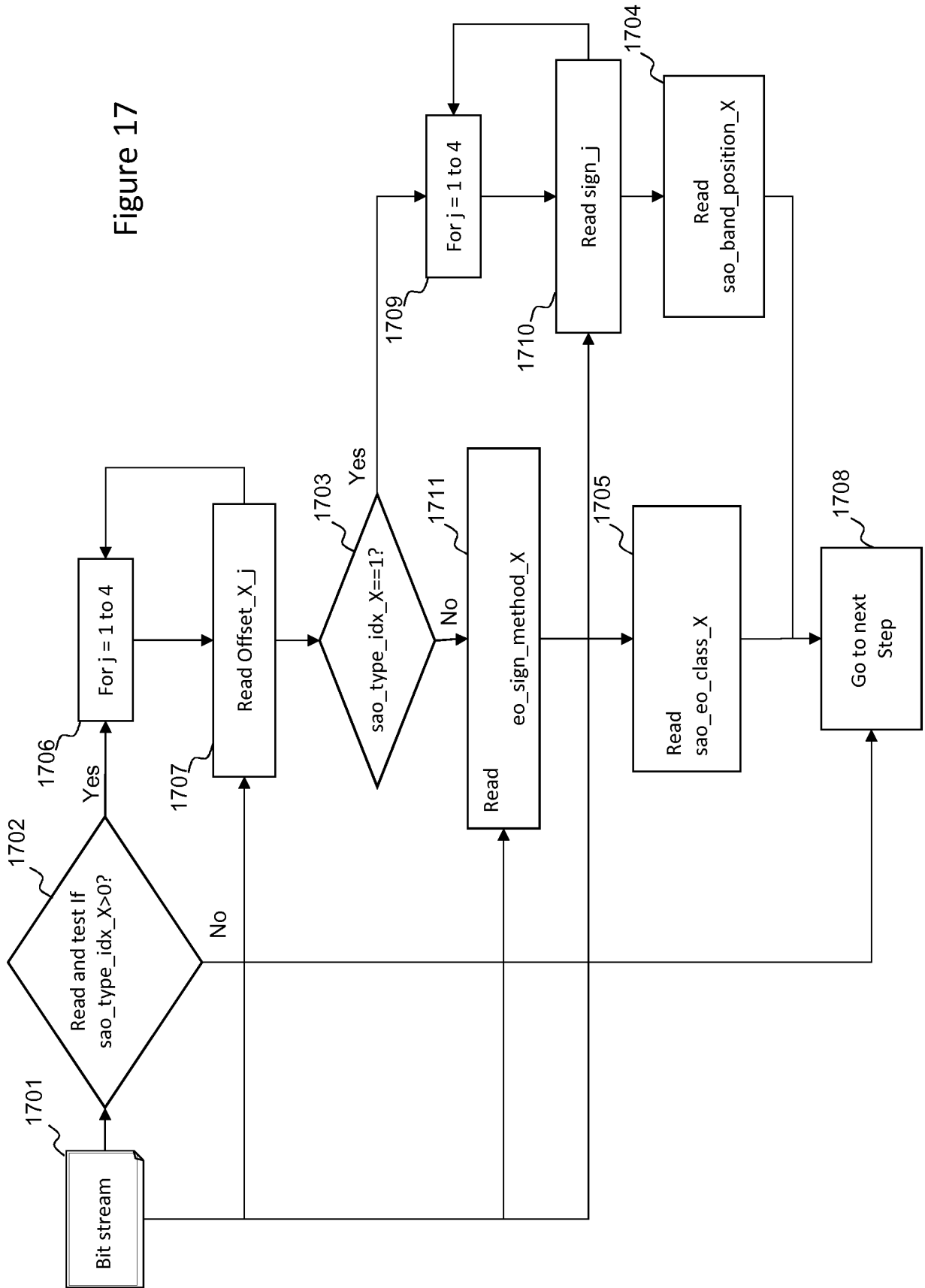
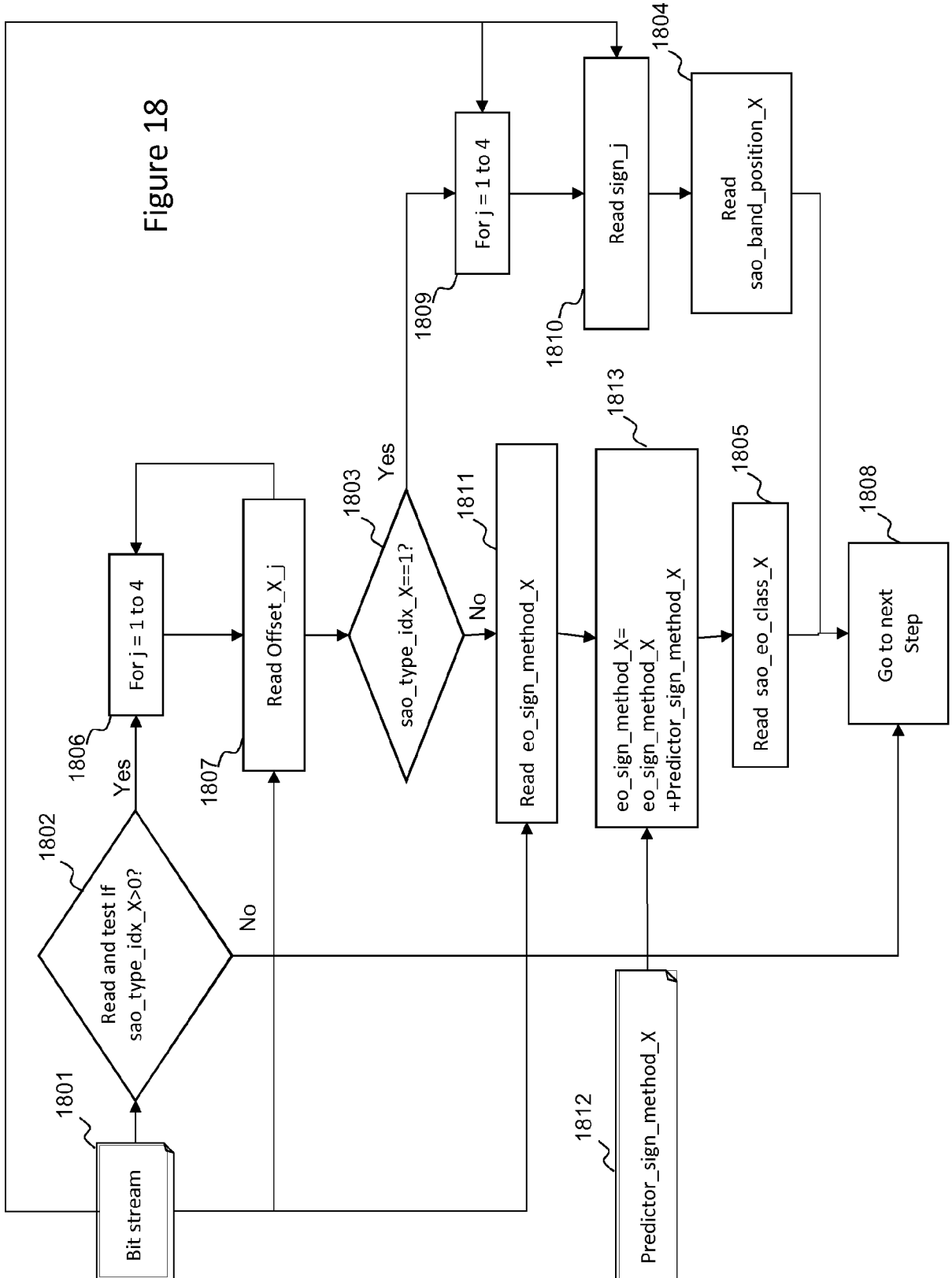


Figure 18



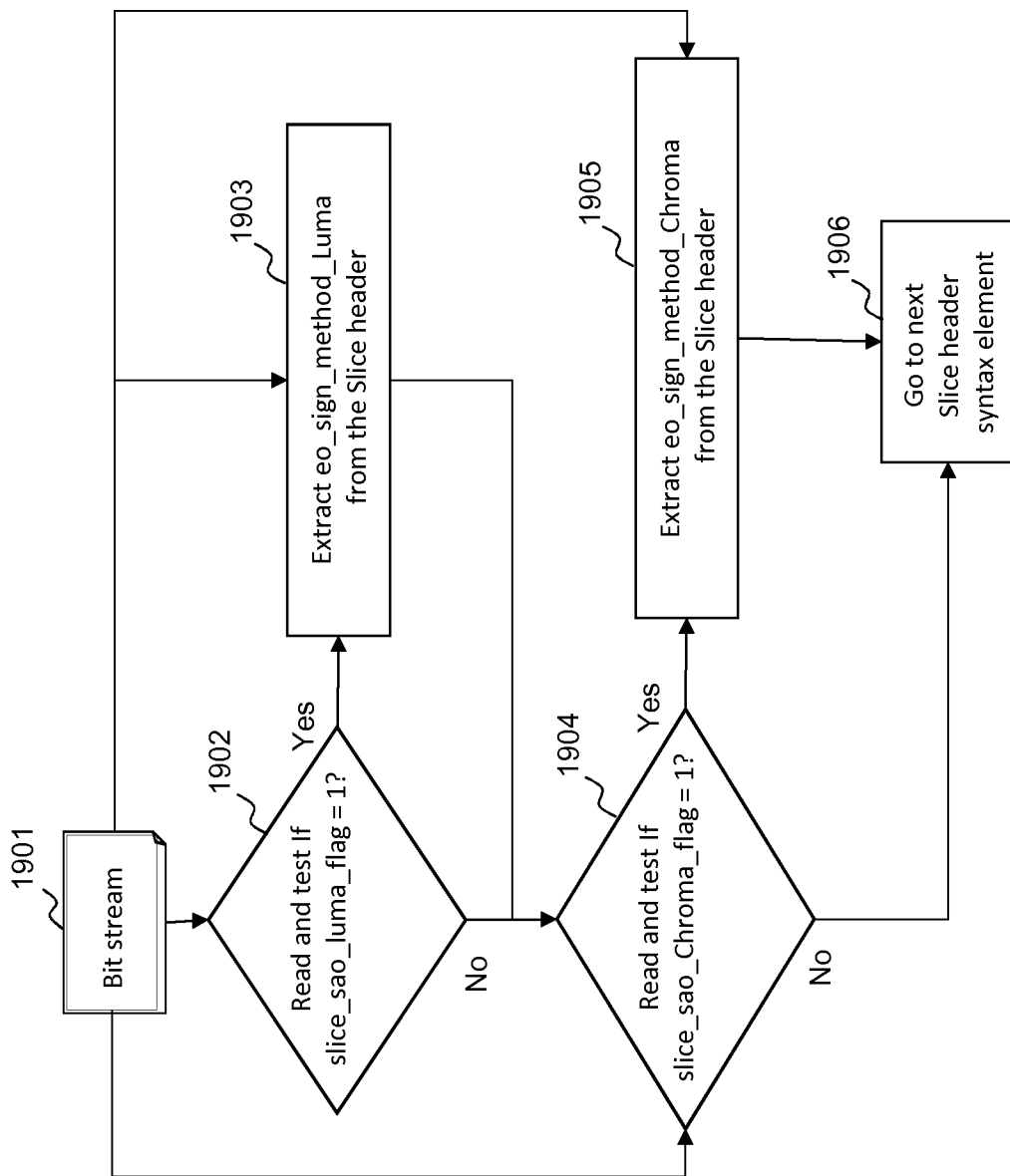


Figure 19

20/40

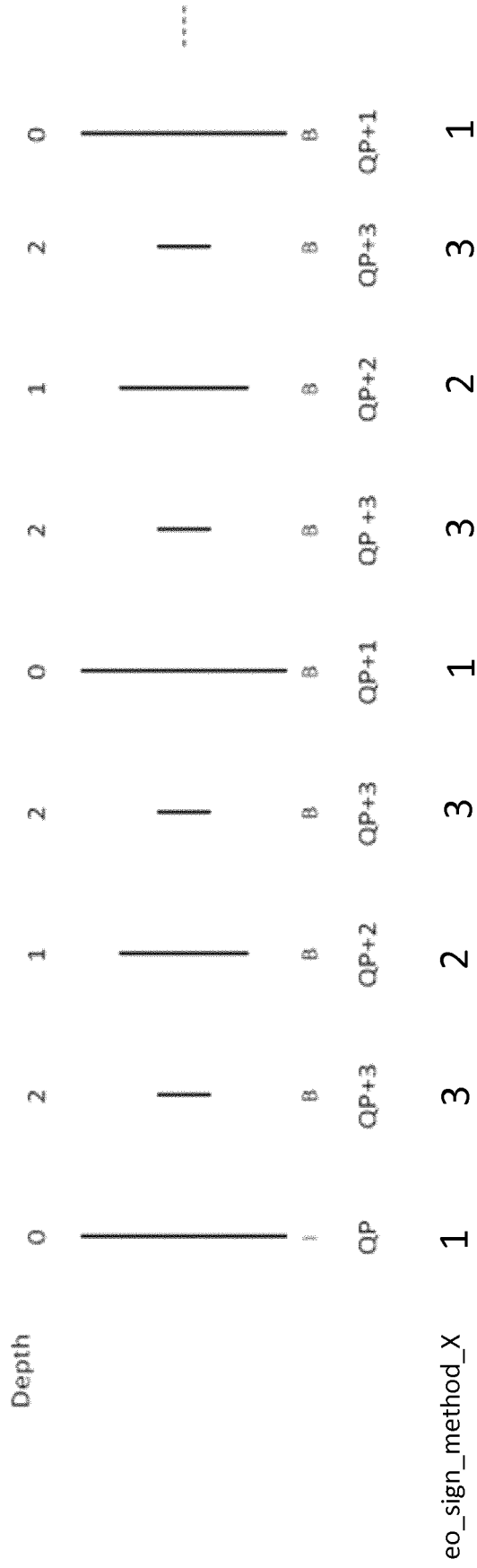


Figure 20

21/40

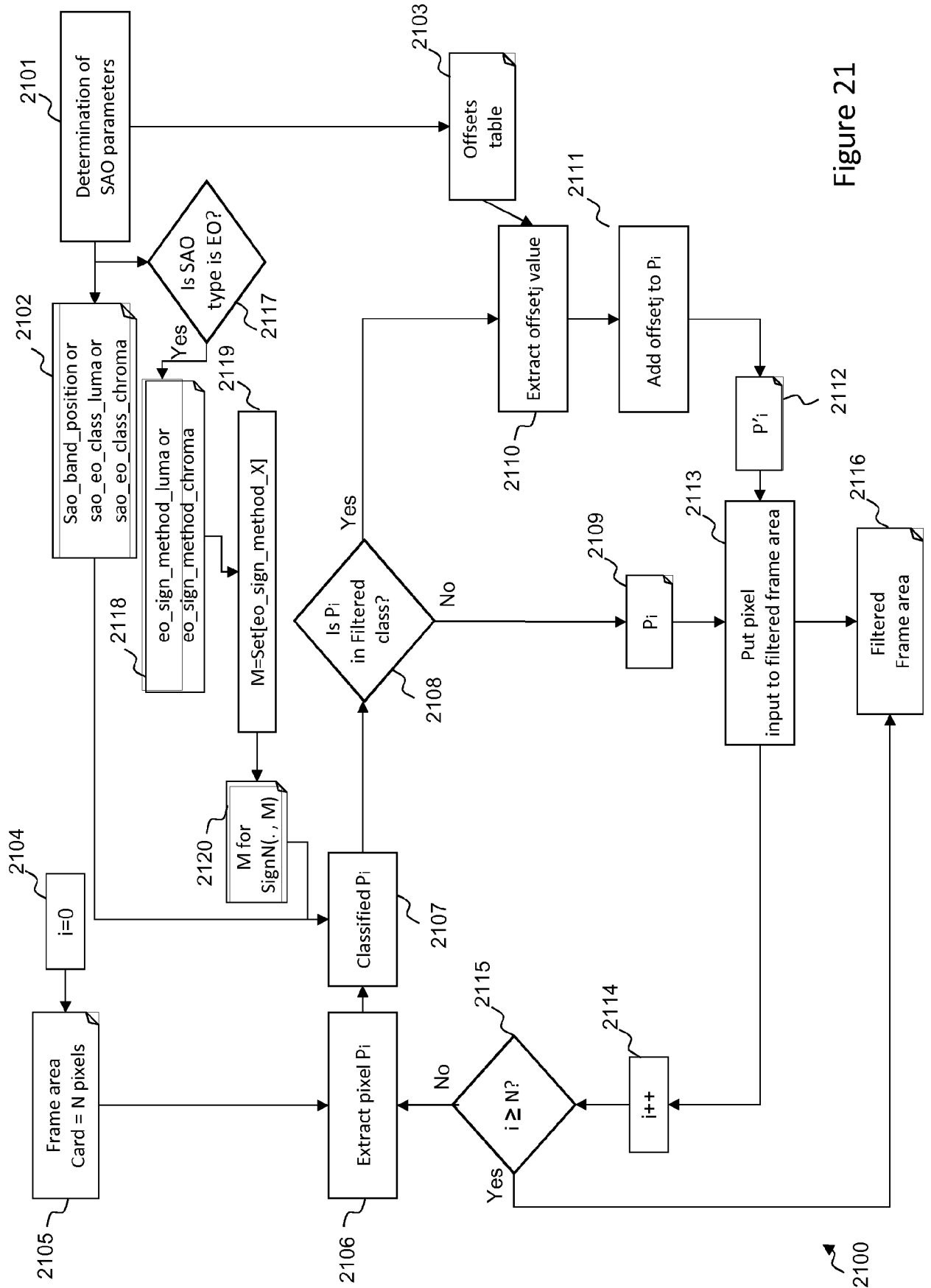
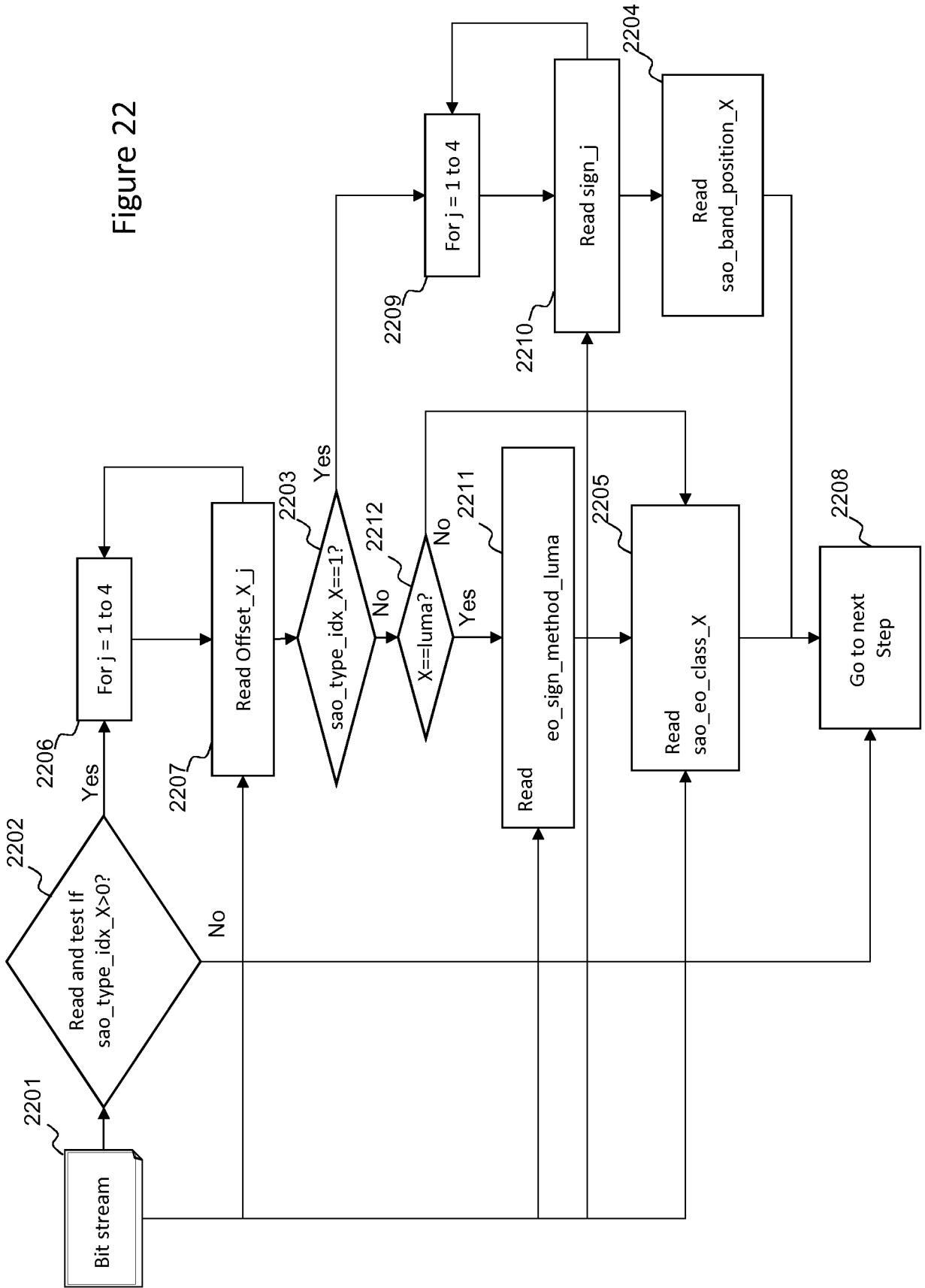


Figure 21

2100

Figure 22



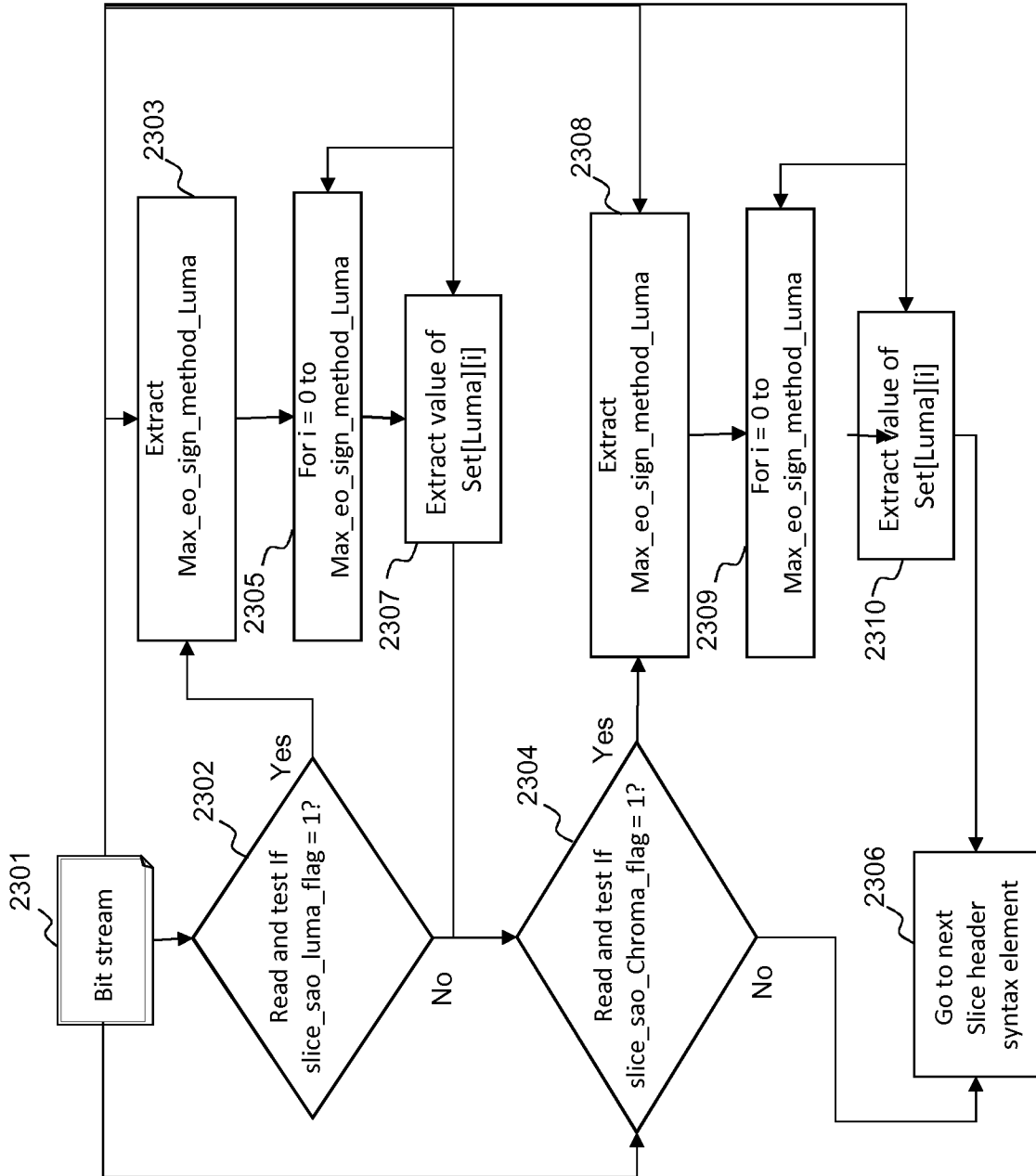


Figure 23

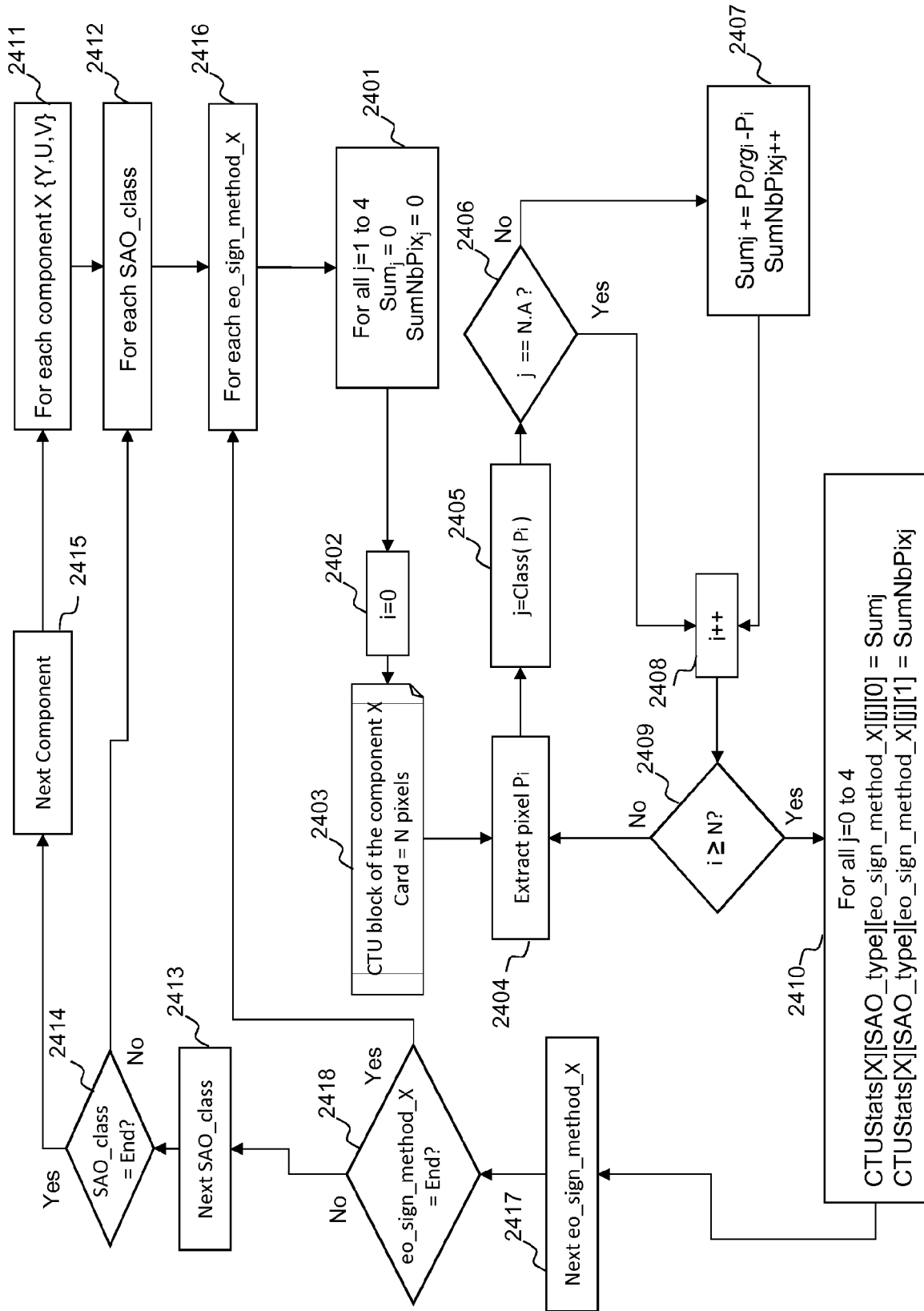


Figure 24

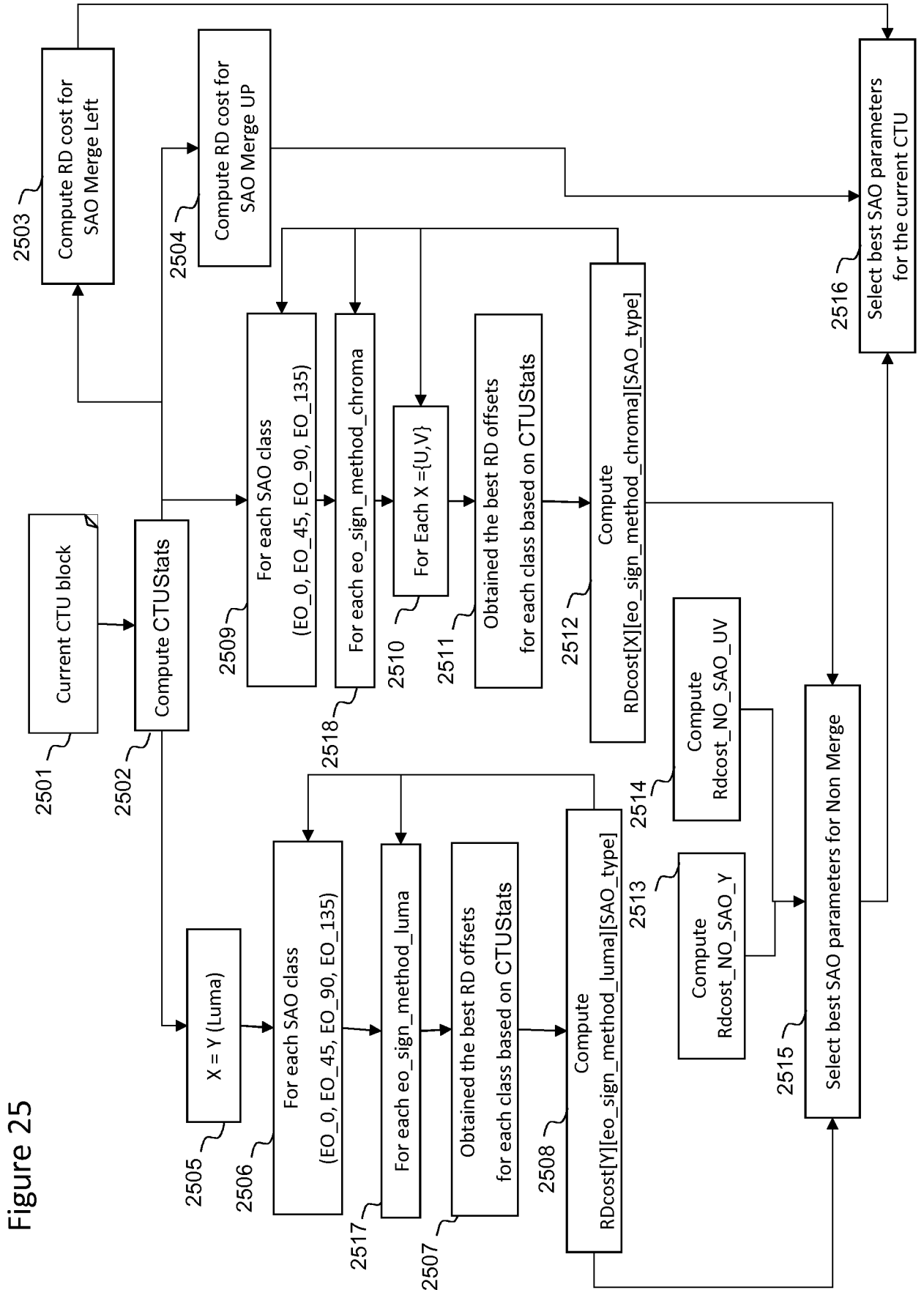
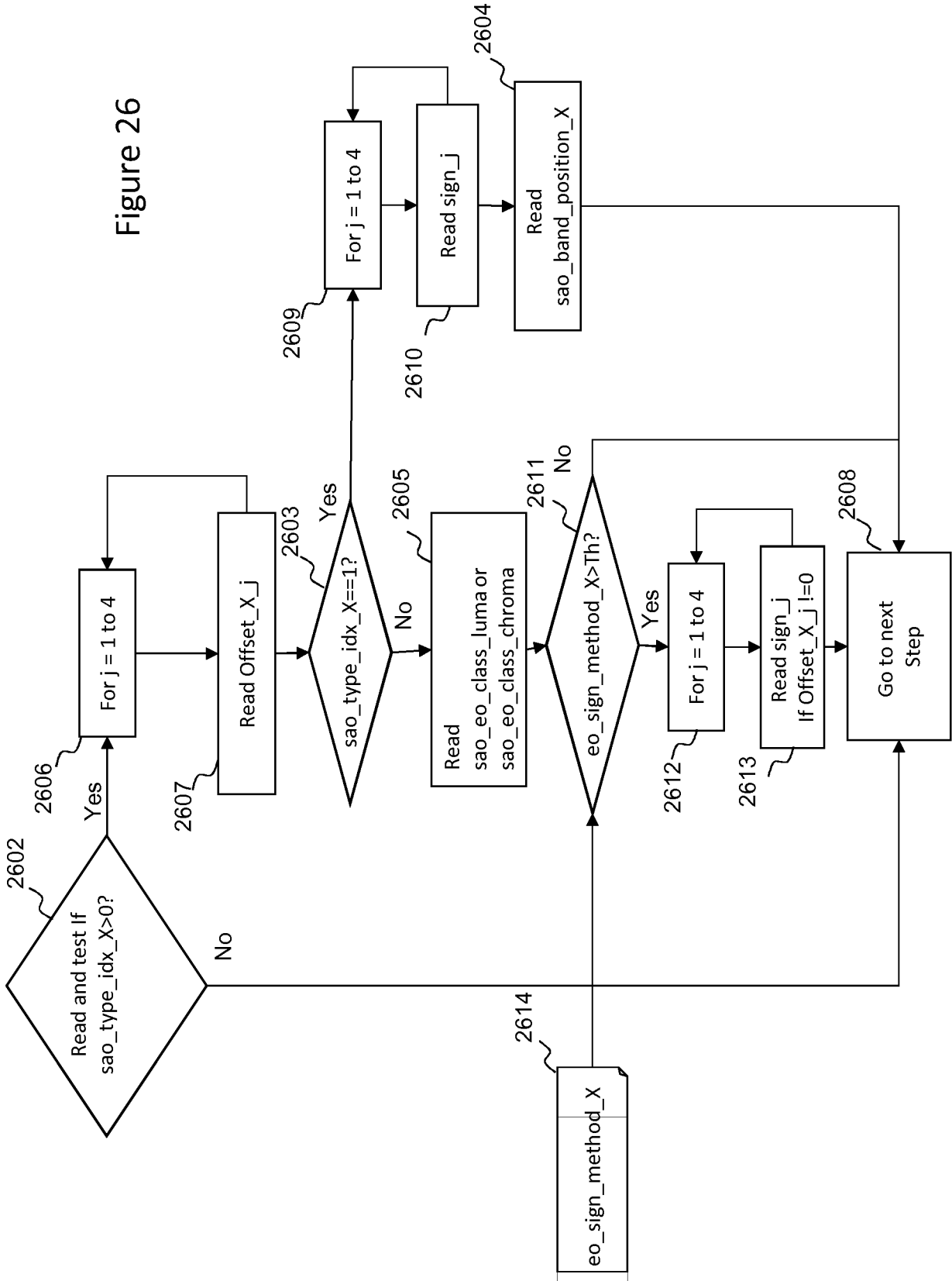


Figure 25

Figure 26



Edge Index	Conditions	Offset
0	$c < a - N \ \&\& \ c < b - N$	01
1	$(c < a - N \ \&\& \  c-b  \leq N) \ \parallel \ ( c - a  \leq N \ \&\& \ c < b - N)$	02
3	$(c > a + N \ \&\& \  c-b  \leq N) \ \parallel \ ( c - a  \leq N \ \&\& \ c > b + N)$	03
4	$c > a + N \ \&\& \ c > b + N$	04
2	None of the above	

Figure 27A

Edge Index	Conditions	Offset
0	$c < a - N \ \&\& \ c < b - N$	01
1	$(c < a - N \ \&\& \  c-b  \leq N) \ \parallel \ ( c - a  \leq N \ \&\& \ c < b - N)$	+ 02
3	$(c > a + N \ \&\& \  c-b  \leq N) \ \parallel \ ( c - a  \leq N \ \&\& \ c > b + N)$	- 03
4	$c > a + N \ \&\& \ c > b + N$	04
2	None of the above	

Figure 27B

Figure 28

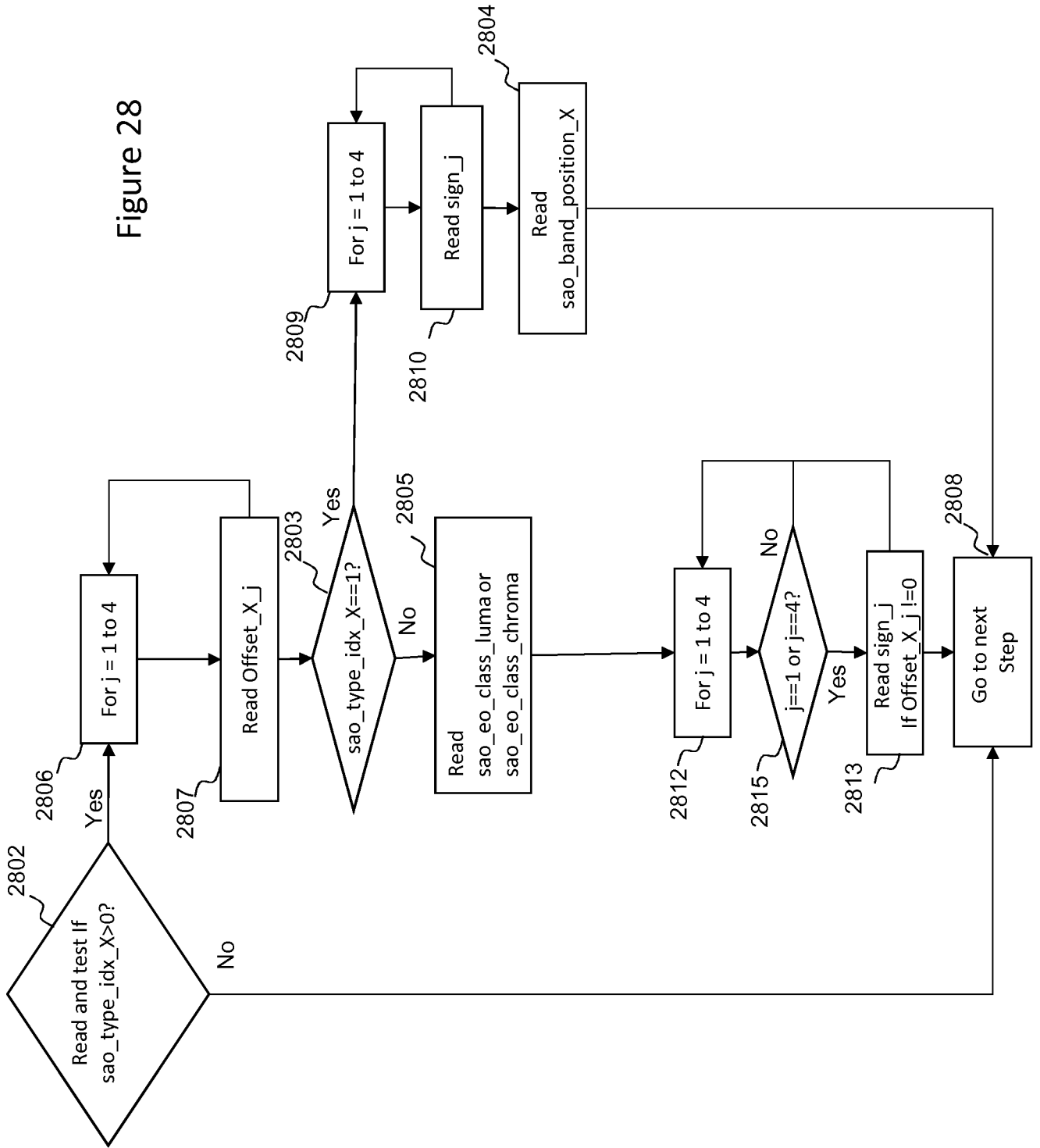


Figure 29

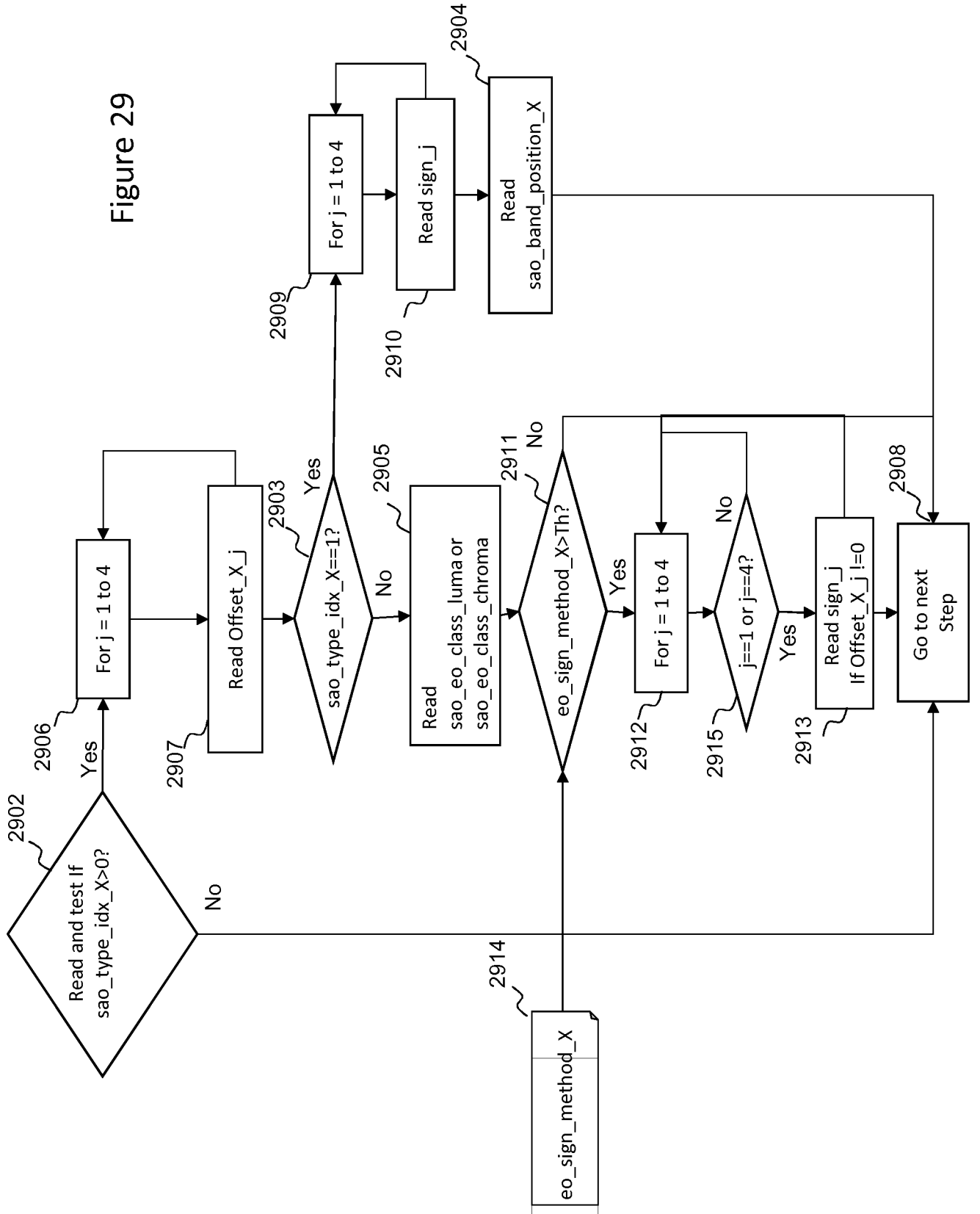


Figure 30

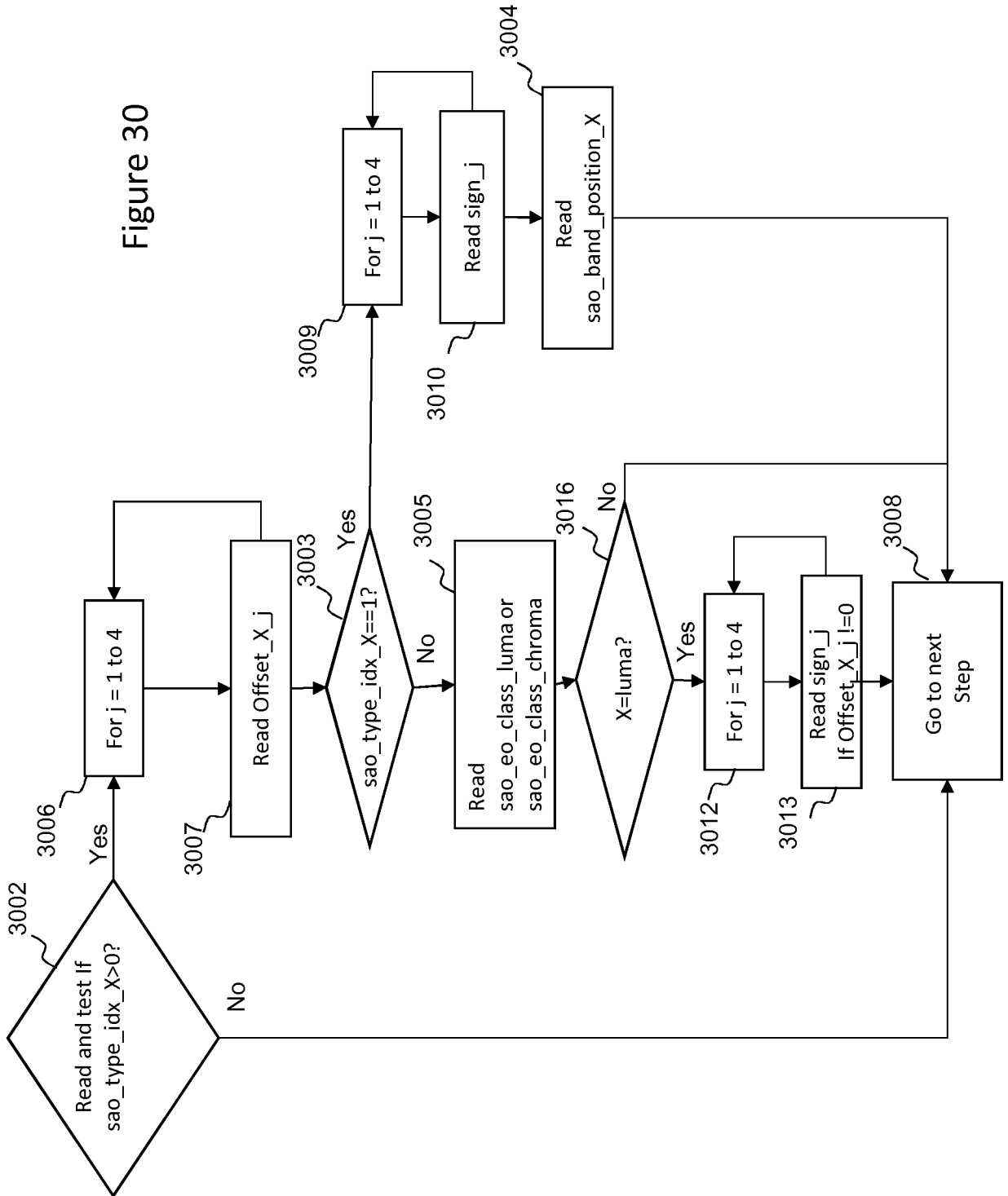


Figure 31

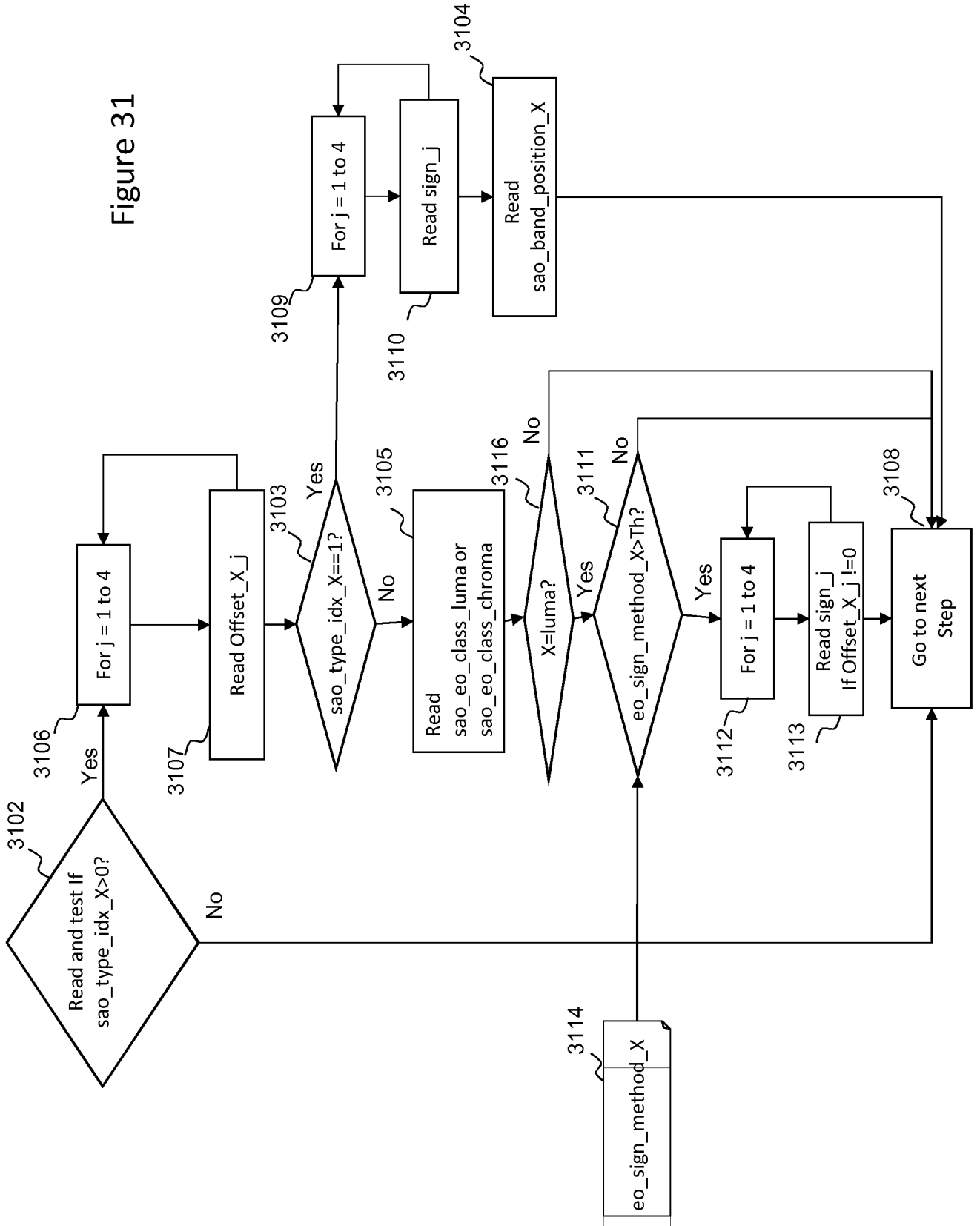


Figure 32

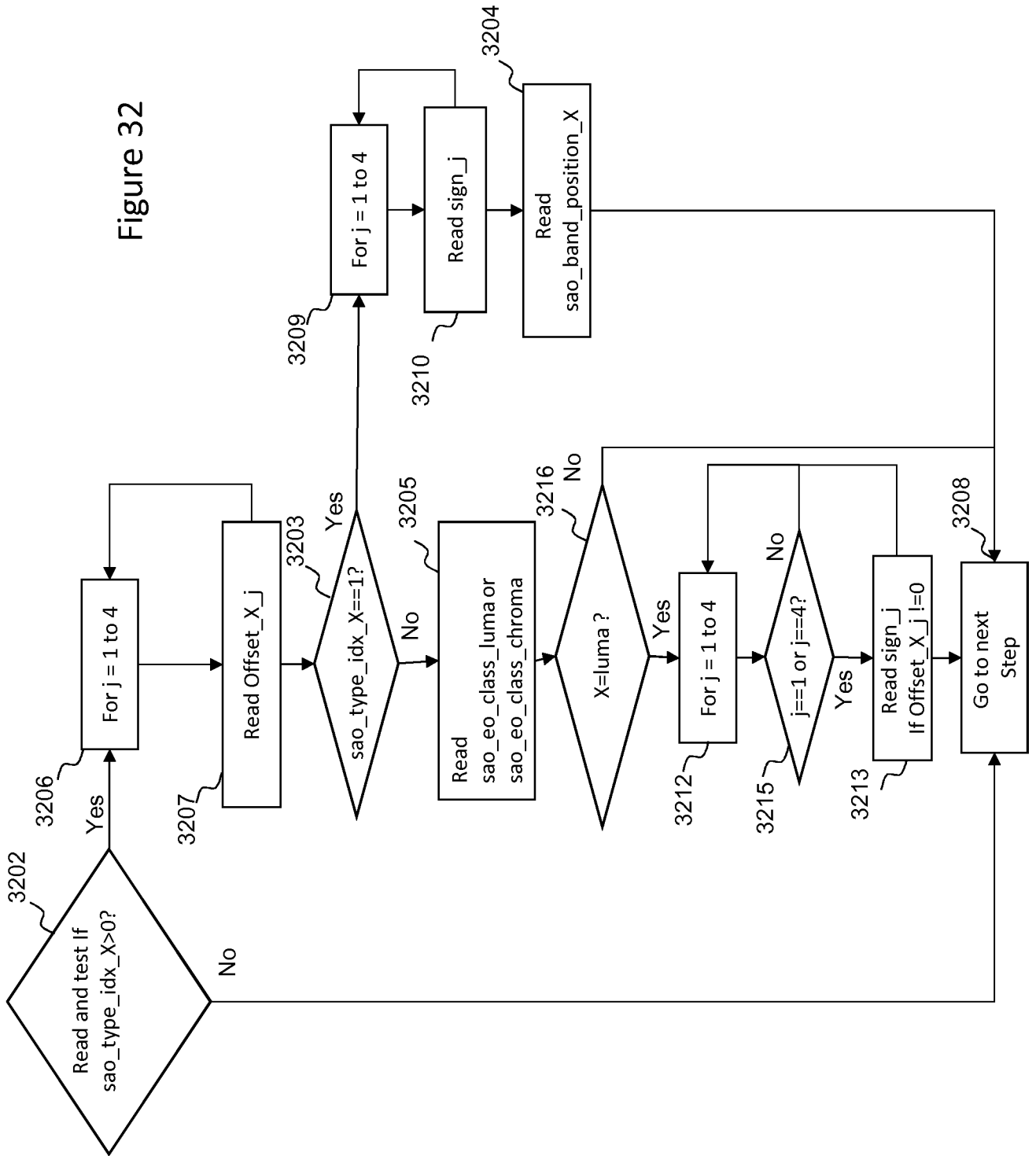


Figure 33

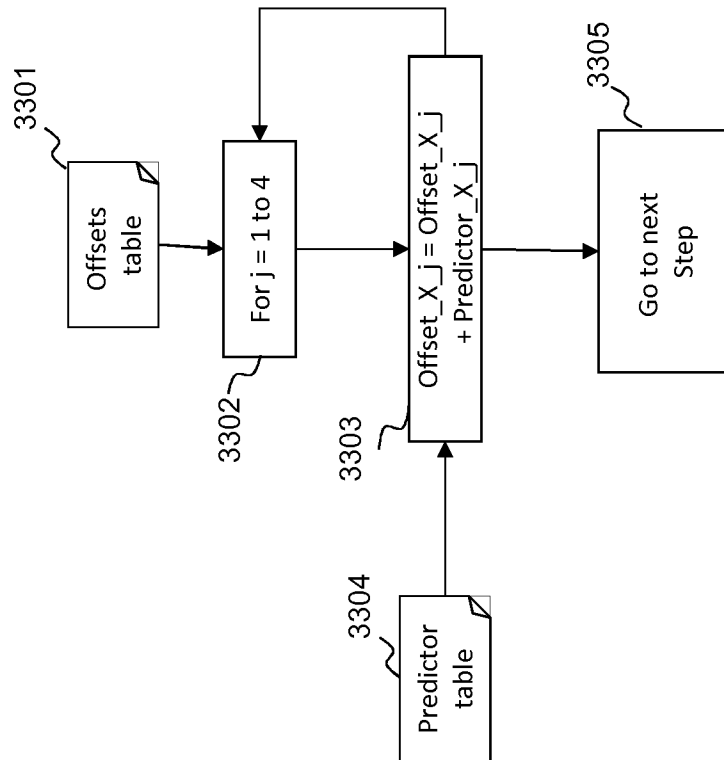


Figure 34

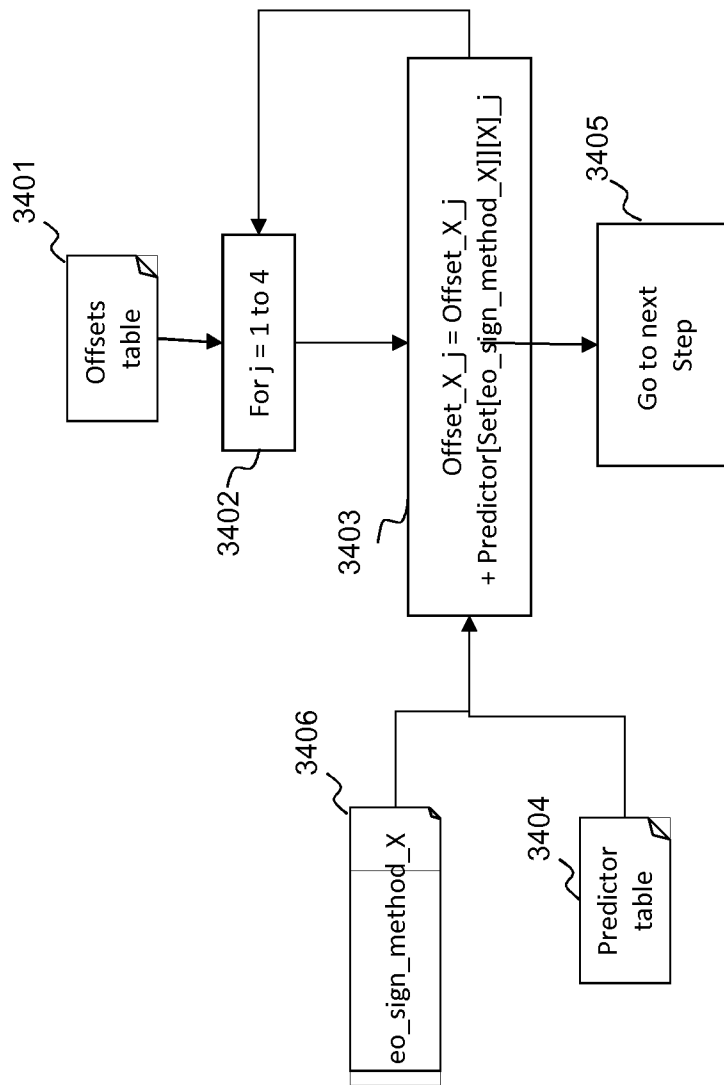


Figure 35

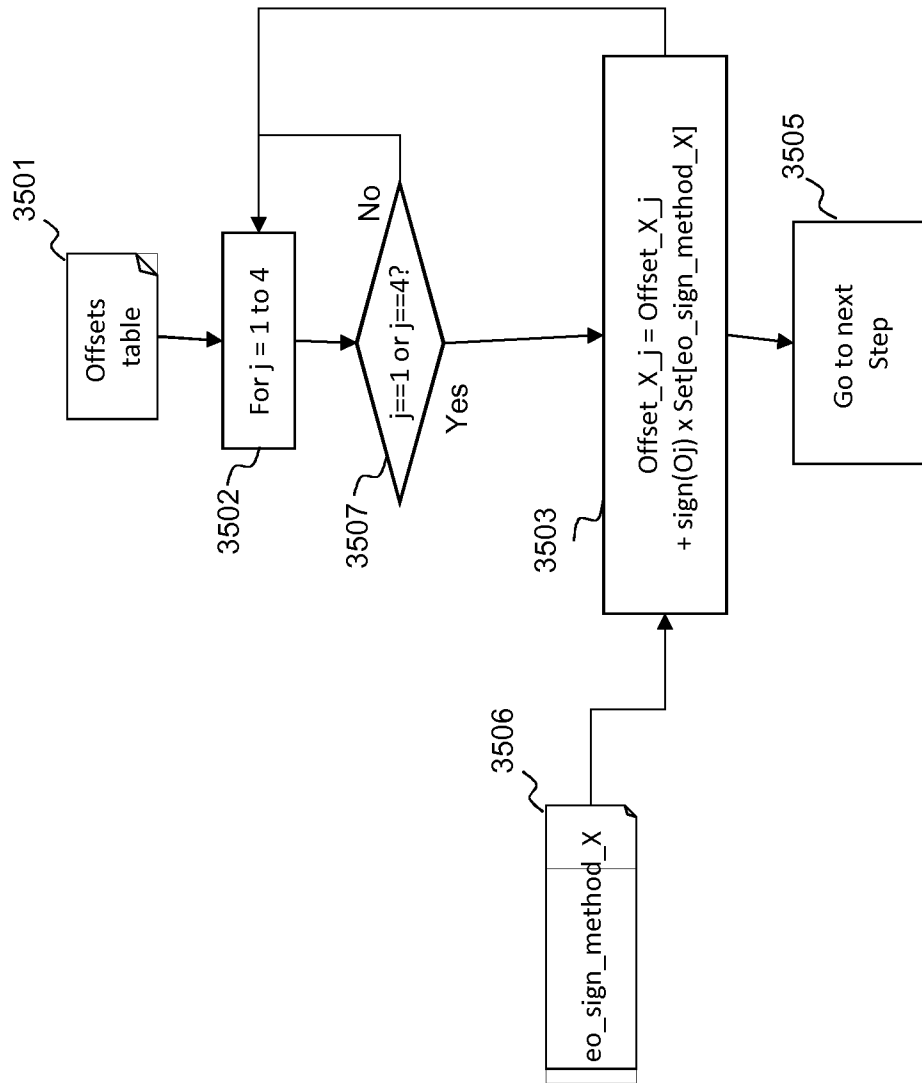


Figure 36

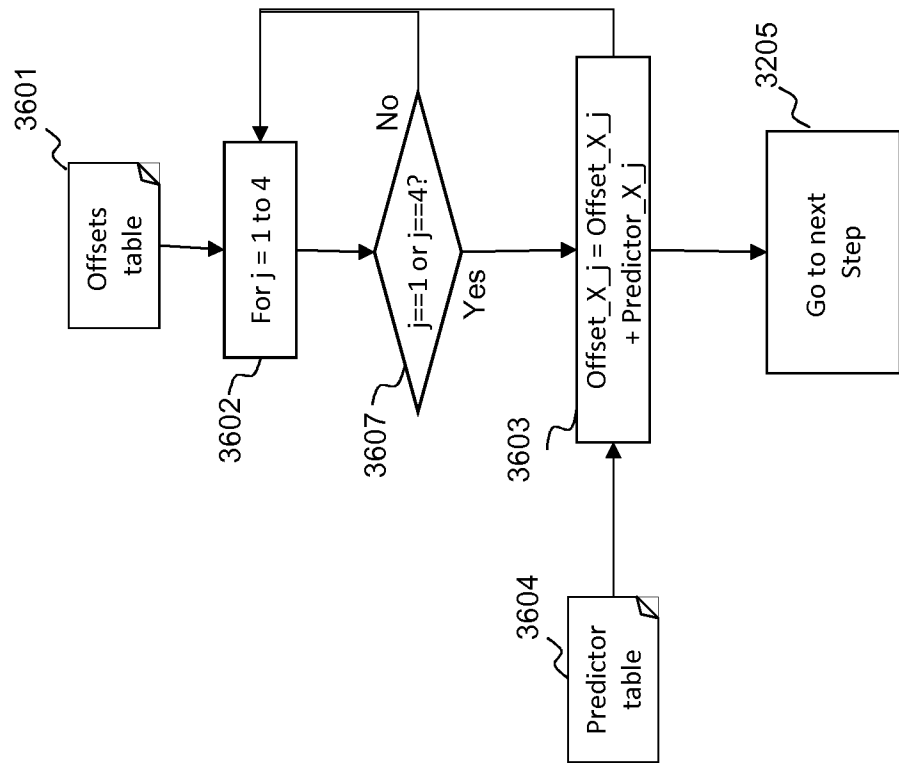


Figure 37

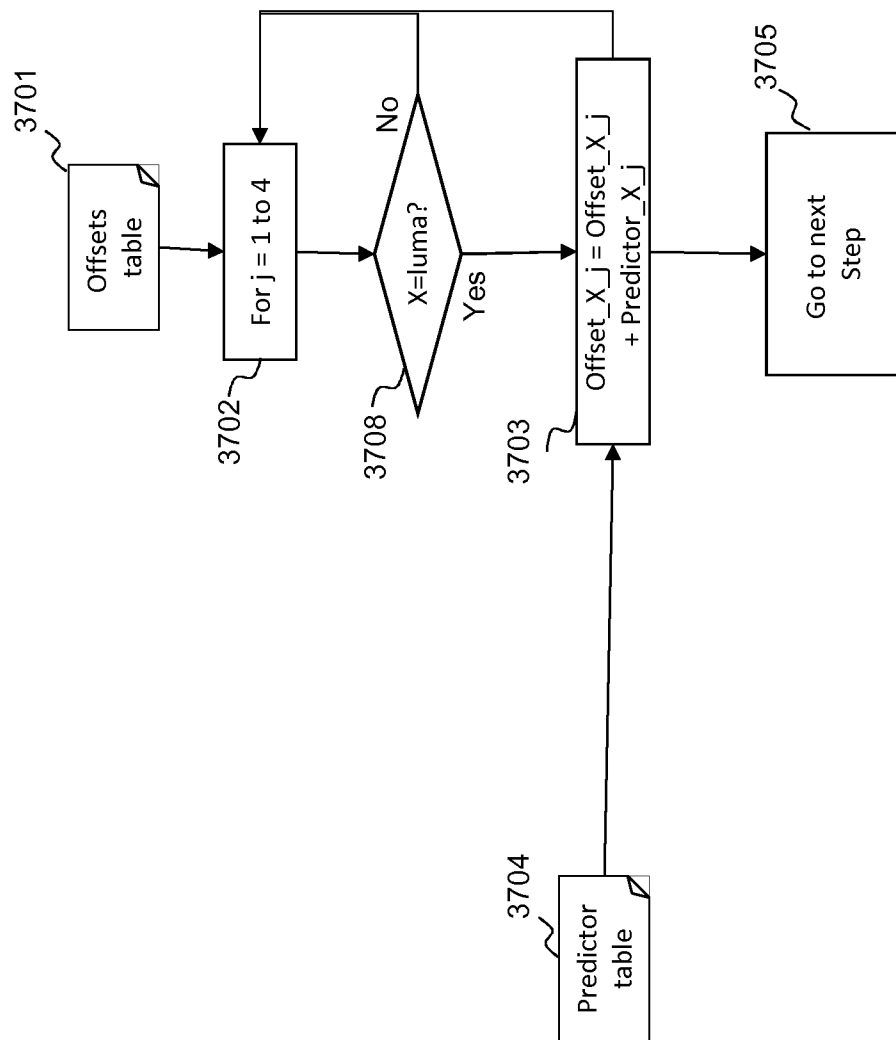
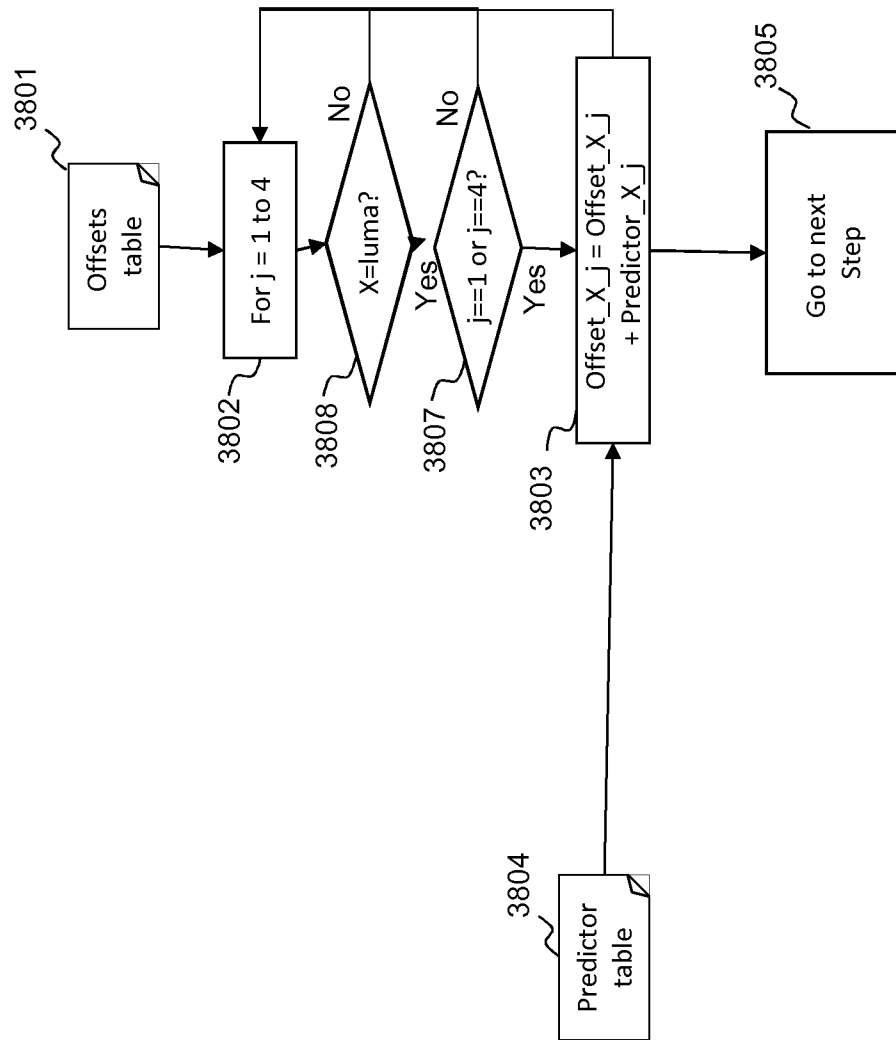


Figure 38



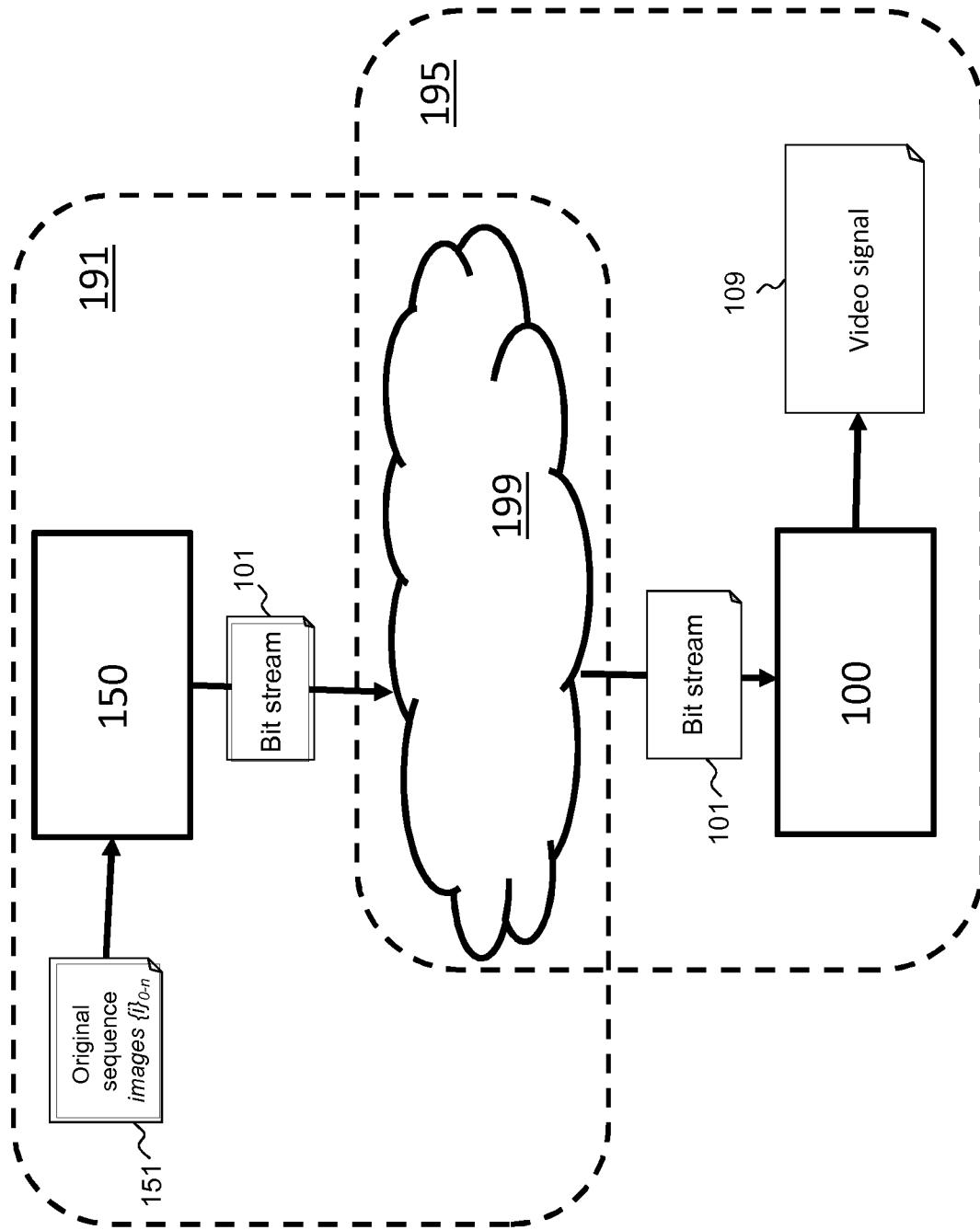


Fig. 39