US 20130114679A1

(19) United States
(12) Patent Application Publication
Wilkins et al.
(10) Pub. No.: US 2013/0114679 A1
(43) Pub. Date: May 9, 2013

(54) **METHOD AND APPARATUS FOR VIDEO CODING USING ADAPTIVE LOOP FILTER**

(71) Applicants: **Paul Wilkins**, Cambridge (GB); **James Bankoski**, North Greenbush, NY (US); **Yaowu Xu**, Clifton Park, NY (US)

(72) Inventors: **Paul Wilkins**, Cambridge (GB); **James Bankoski**, North Greenbush, NY (US); **Yaowu Xu**, Clifton Park, NY (US)

(21) Appl. No.: **13/688,608**

(22) Filed: **Nov. 29, 2012**

Related U.S. Application Data

(63) Continuation of application No. 12/329,070, filed on Dec. 5, 2008, now Pat. No. 8,326,075.

(60) Provisional application No. 61/096,147, filed on Sep. 11, 2008.

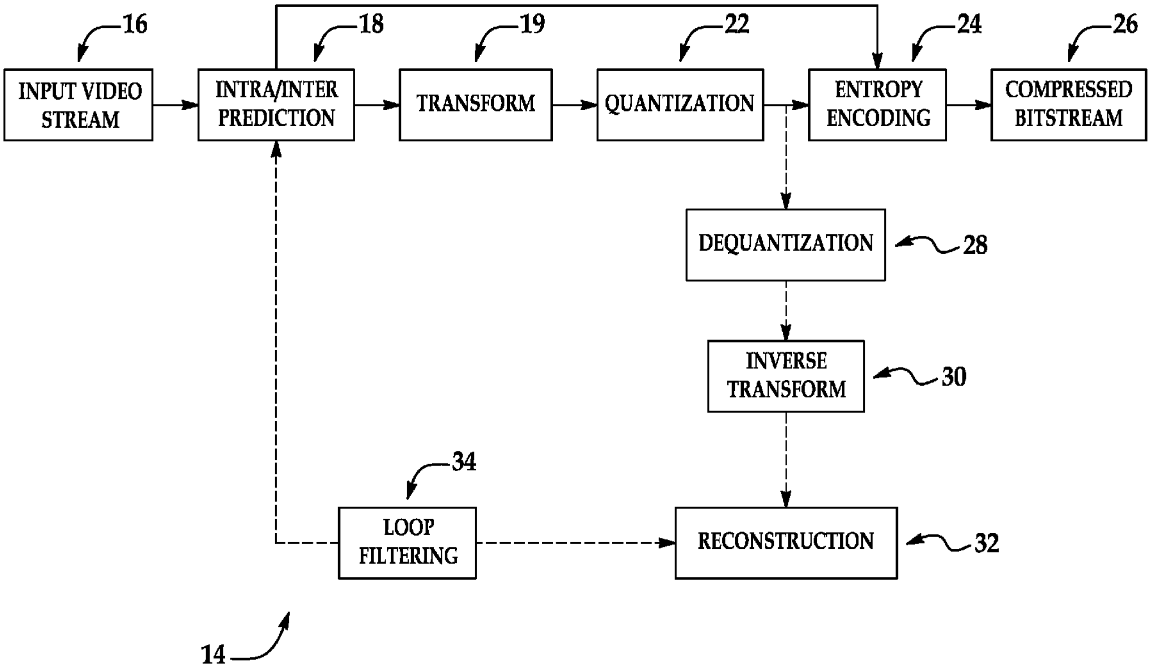
Publication Classification

(51) **Int. Cl.**
H04N 7/26 (2006.01)
H04N 7/32 (2006.01)

(52) **U.S. Cl.**
USPC **375/240.02**

(57) **ABSTRACT**

Methods for reducing blocking artifacts at the boundary between adjacent blocks reconstructed from a frame of compressed video information are disclosed herein. The frame includes a prediction stage parameter and a residual error attribute with respect to at least one of the blocks. One method includes reconstructing the at least one block based on the prediction stage parameter and the residual error attribute, categorizing the at least one reconstructed block into one of a plurality of categories based on the prediction stage parameter and the residual error attribute, identifying a filter strength value for the category in which the at least one reconstructed block is categorized based on at least one of the prediction stage parameter or the residual error attribute associated with that category, and filtering the boundary adjacent to the at least one reconstructed block using the identified filter strength value.



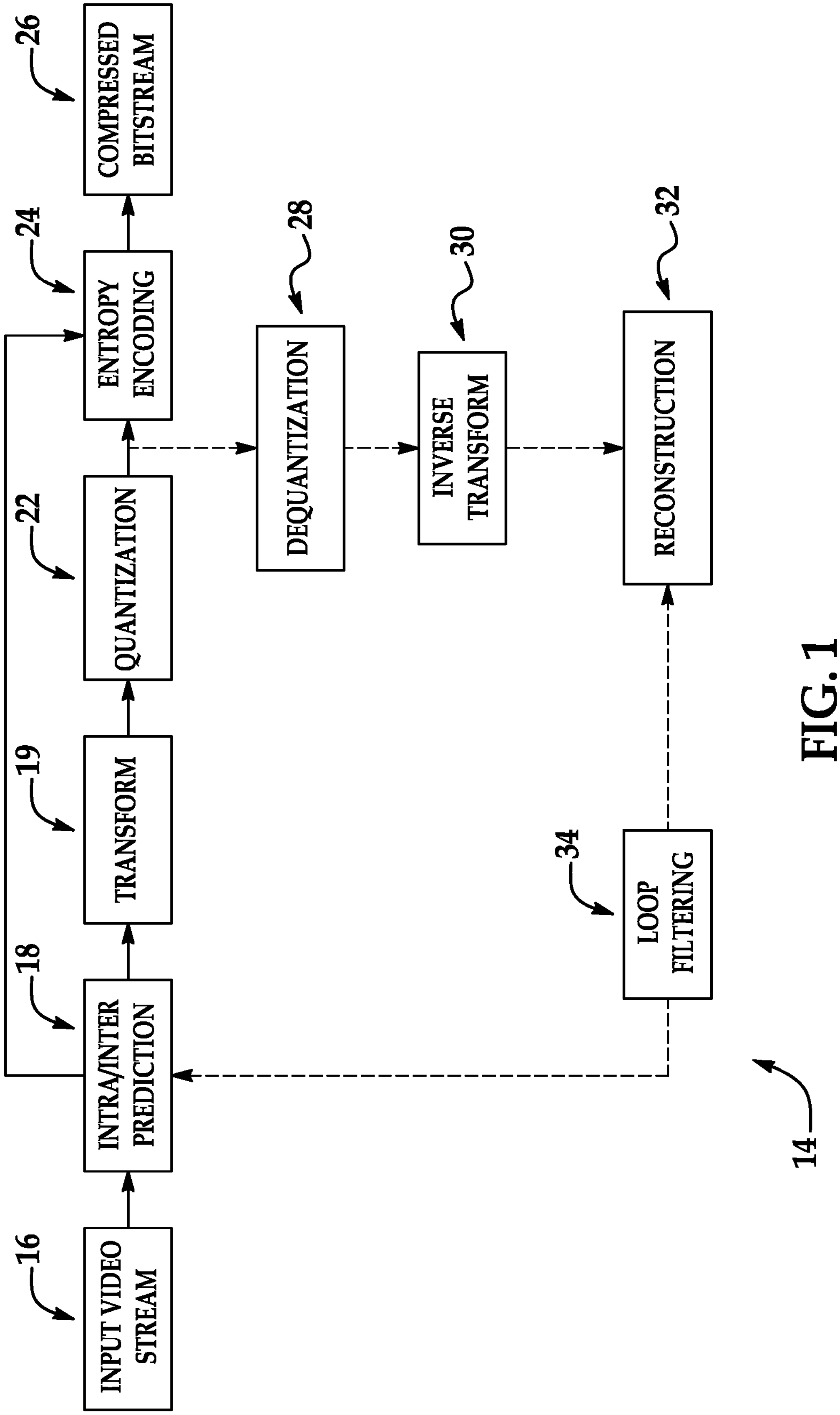

INPUT VIDEO STREAM
16
INTRA/INTER PREDICTION
18
TRANSFORM
19
QUANTIZATION
22
ENTROPY ENCODING
24
COMPRESSED BITSTREAM
26
DEQUANTIZATION
28
INVERSE TRANSFORM
30
RECONSTRUCTION
32
LOOP FILTERING
34
14

FIG. 1

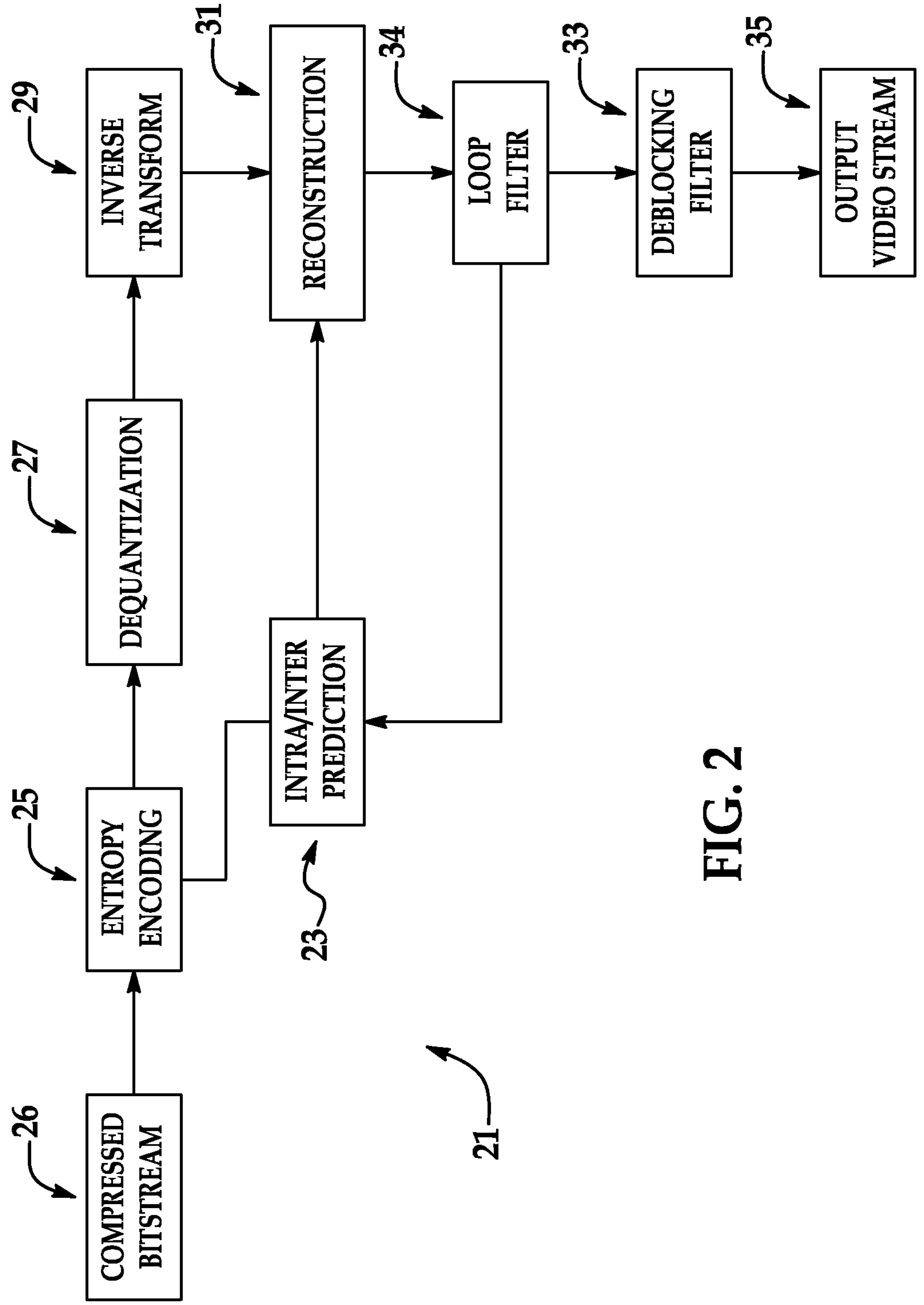

COMPRESSED BITSTREAM
26
ENTROPY ENCODING
25
DEQUANTIZATION
27
INVERSE TRANSFORM
29
INTRA/INTER PREDICTION
23
RECONSTRUCTION
31
LOOP FILTER
34
DEBLOCKING FILTER
33
OUTPUT VIDEO STREAM
35
21

FIG. 2

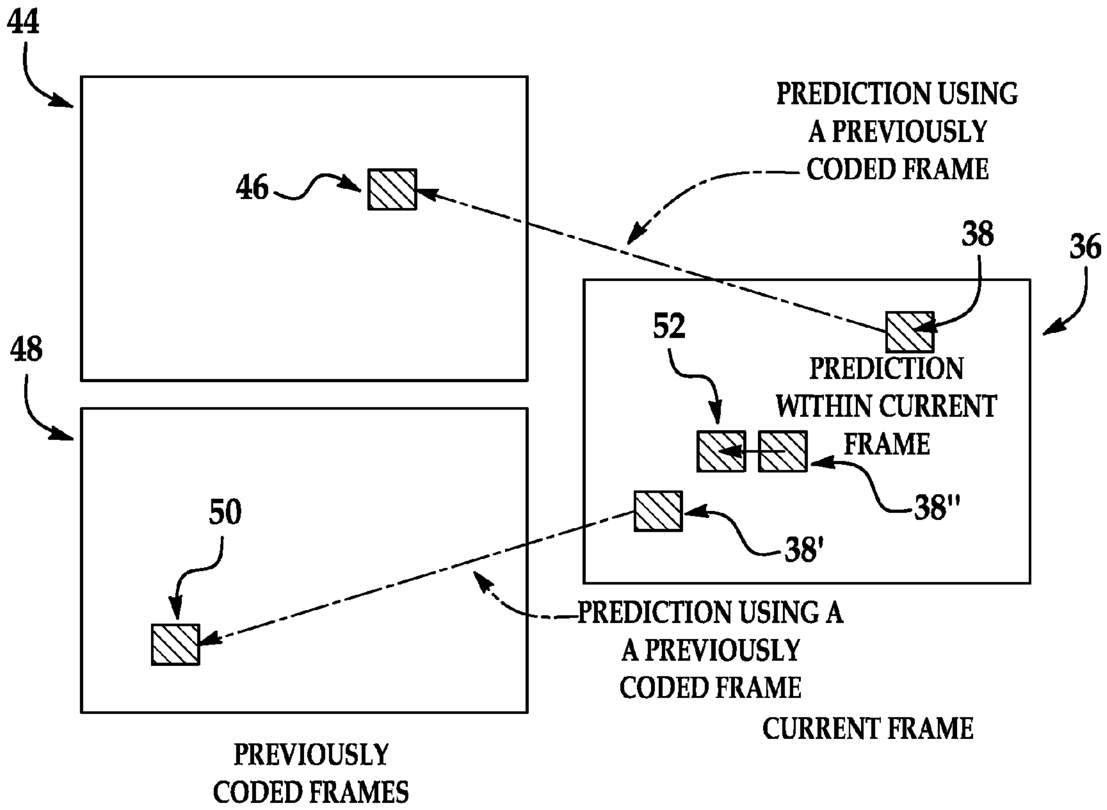

44
46
PREDICTION USING A PREVIOUSLY CODED FRAME
38
36
48
52
PREDICTION WITHIN CURRENT FRAME
38"
50
38'
PREDICTION USING A A PREVIOUSLY CODED FRAME
CURRENT FRAME
PREVIOUSLY CODED FRAMES

FIG. 3

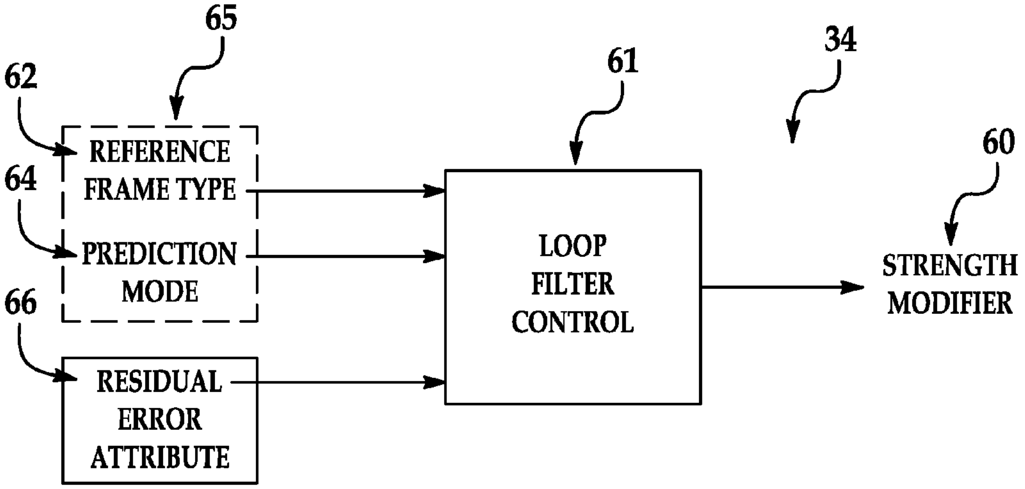

65
62
34
61
REFERENCE FRAME TYPE
64
60
PREDICTION MODE
LOOP FILTER CONTROL
STRENGTH MODIFIER
66
RESIDUAL ERROR ATTRIBUTE

FIG. 4

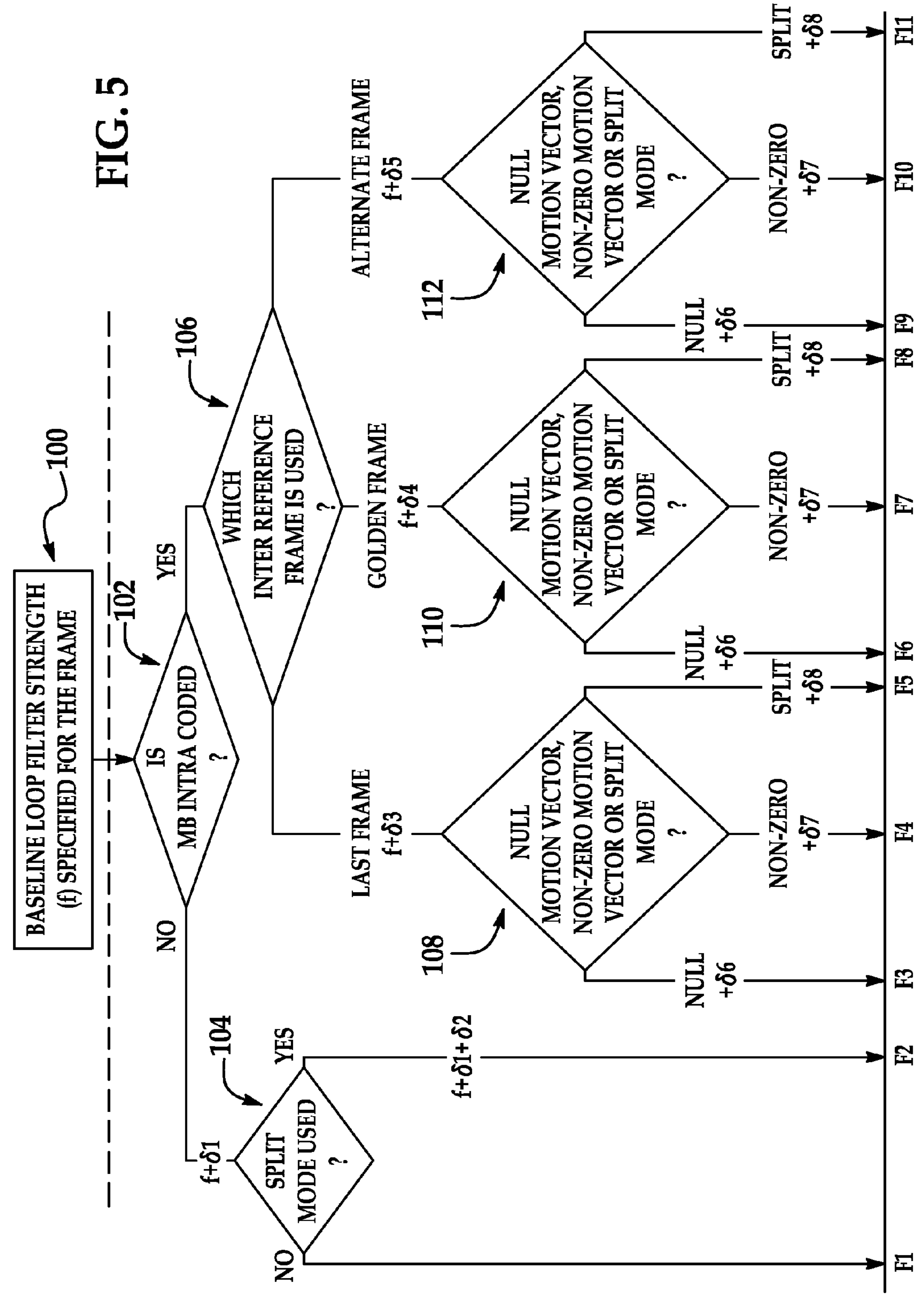

100
BASELINE LOOP FILTER STRENGTH (f) SPECIFIED FOR THE FRAME
102
IS MB INTRA CODED ?
YES
NO
f+δ1
104
SPLIT MODE USED ?
YES
NO
f+δ1+δ2
106
WHICH INTER REFERENCE FRAME IS USED ?
LAST FRAME f+δ3
GOLDEN FRAME f+δ4
ALTERNATE FRAME f+δ5
108
110
112
NULL MOTION VECTOR, NON-ZERO MOTION VECTOR OR SPLIT MODE ?
NULL +δ6
NON-ZERO +δ7
SPLIT +δ8
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11

FIG. 5

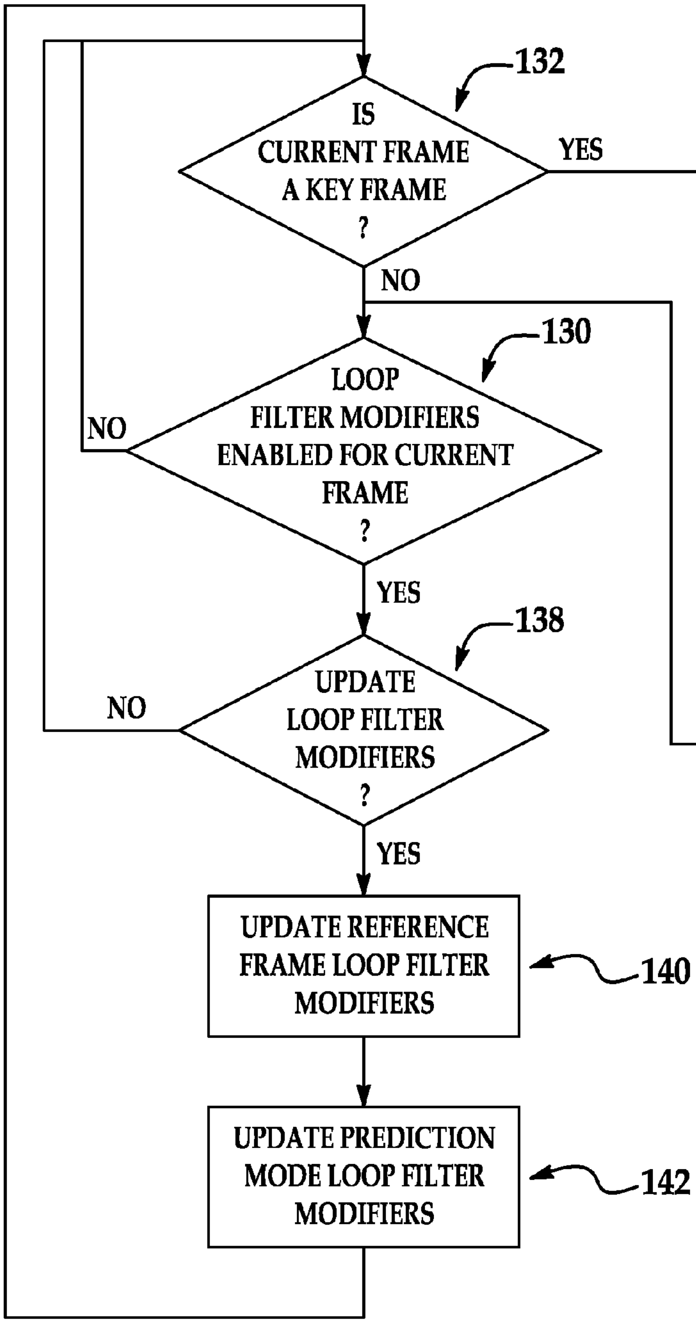

132
IS CURRENT FRAME A KEY FRAME ?
YES
NO
134
SET REFERENCE FRAME LOOP FILTER MODIFIERS TO DEFAULT
136
SET PREDICTION MODE LOOP FILTER MODIFIERS TO DEFAULT
130
LOOP FILTER MODIFIERS ENABLED FOR CURRENT FRAME ?
NO
YES
138
UPDATE LOOP FILTER MODIFIERS ?
NO
YES
UPDATE REFERENCE FRAME LOOP FILTER MODIFIERS
140
UPDATE PREDICTION MODE LOOP FILTER MODIFIERS
142

FIG. 6

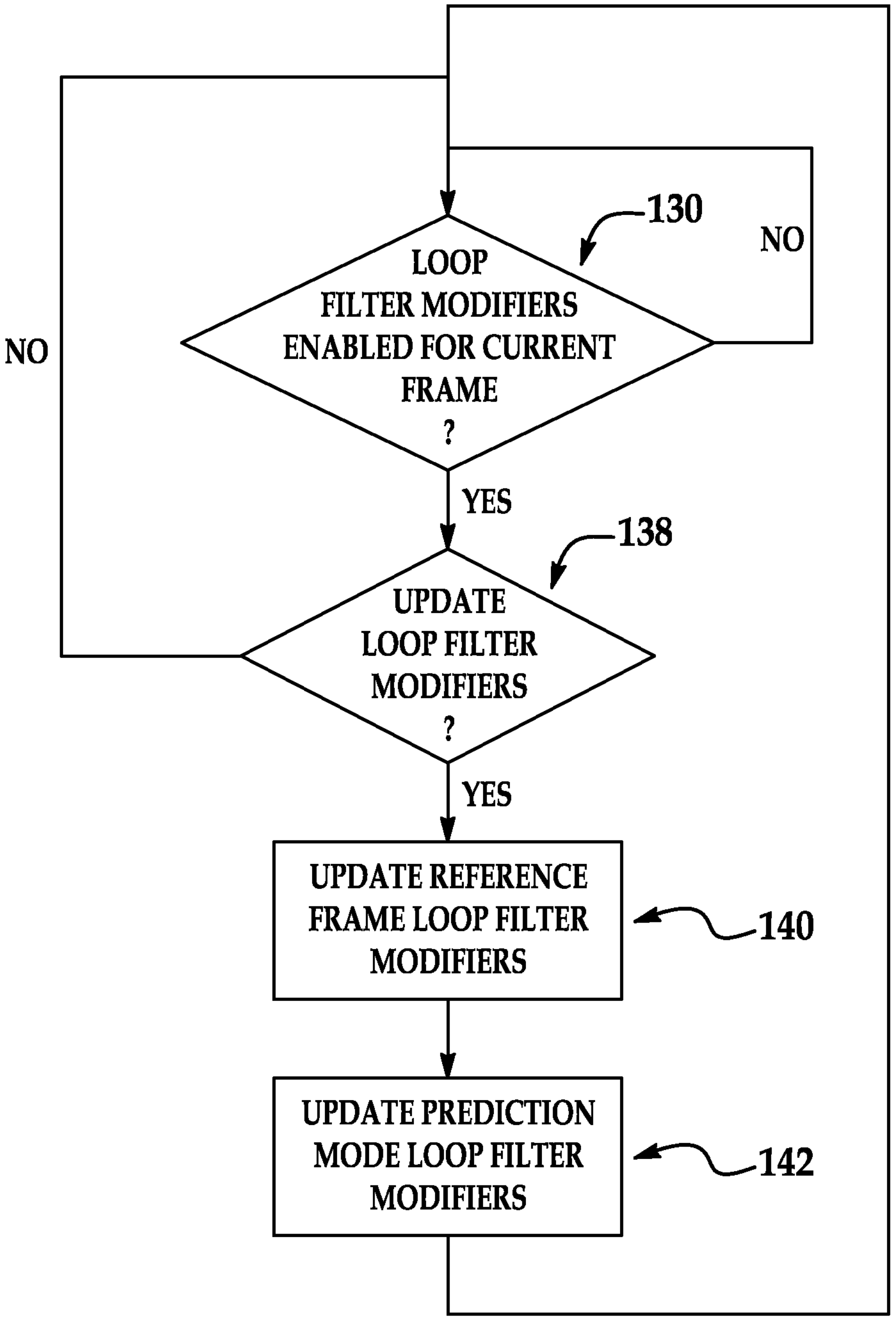

LOOP FILTER MODIFIERS ENABLED FOR CURRENT FRAME ?
130
NO
NO
YES
UPDATE LOOP FILTER MODIFIERS ?
138
YES
UPDATE REFERENCE FRAME LOOP FILTER MODIFIERS
140
UPDATE PREDICTION MODE LOOP FILTER MODIFIERS
142

FIG. 7

METHOD AND APPARATUS FOR VIDEO CODING USING ADAPTIVE LOOP FILTER

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation of U.S. patent application Ser. No. 12/329,070, filed Dec. 5, 2008 which claims priority to U.S. provisional patent application No. 61/096,147, filed Sep. 11, 2008, which is incorporated herein in its entirety by reference.

TECHNICAL FIELD

[0002] The present invention relates in general to video encoding and more particularly, video encoding using a loop filter.

BACKGROUND

[0003] An increasing number of applications today make use of digital video for various purposes including, for example, remote business meetings via video conferencing, high definition video entertainment, video advertisements, and sharing of user-generated videos. As technology is evolving, people have higher expectations for video quality and expect high resolution video with smooth playback at a high frame rate.

[0004] There can be many factors to consider when selecting a video coder for viewing digital video. Some applications may require excellent video quality where others may need to comply with various constraints including, for example, bandwidth or storage requirements. To permit higher quality transmission of video while limiting bandwidth consumption, a number of video compression schemes are noted including proprietary formats such as VPx (promulgated by On2 Technologies, Inc. of Clifton Park, N.Y.), H.264 standard promulgated by ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG), including present and future versions thereof. H.264 is also known as MPEG-4 Part 10 or MPEG-4 AVC (formally, ISO/IEC 14496-10).

[0005] Many video coding techniques use block based prediction and quantized block transforms. With block based prediction, a reconstructed frame buffer can be used to predict subsequent frames. The use of block based prediction and quantized block transforms can give rise to discontinuities along block boundaries. These discontinuities (commonly referred to as blocking artifacts) can be visually disturbing and can reduce the effectiveness of the reference frame as a predictor for subsequent frames. These discontinuities can be reduced by the application of a loop filter. The loop filter can be applied to the reconstructed frame buffers. Some conventional loop filters apply different filtering strengths to different block boundaries. For example, some compression systems vary the strength of the loop filter based on, for example, whether the block has been inter-coded or intra-coded. Other compression systems apply a filter strength based on, for example, the difference between the extent of the discontinuity and threshold level. Further, for example, some compression systems may vary the strength of the loop filter by computing a difference value illumination change of a block compared to its neighboring block.

SUMMARY

[0006] Disclosed herein are embodiments of methods and apparatuses for coding video information.

[0007] One aspect of the disclosed embodiments is a method for reducing blocking artifacts at the boundary between adjacent blocks reconstructed from a frame of compressed video information that includes a prediction stage parameter and a residual error attribute with respect to at least one of the blocks. The method includes reconstructing the at least one block based on the prediction stage parameter and the residual error attribute and categorizing the at least one reconstructed block into one of a plurality of categories based on the prediction stage parameter and the residual error attribute. The method also includes identifying a filter strength value for the category in which the at least one reconstructed block is categorized based on at least one of the prediction stage parameter or the residual error attribute associated with that category. Further, the method includes filtering the boundary adjacent to the at least one reconstructed block using the identified filter strength value.

[0008] Another aspect of the disclosed embodiments is a method for decoding frames of compressed video information. Each frame includes a plurality of blocks having a prediction stage parameter and a residual error attribute. The method includes categorizing the blocks of at least one of the frames into categories based on both the prediction stage parameter and the residual error attribute associated with each given block and determining a filter strength value for each of the categories. The method also includes filtering the boundary adjacent to at least one of the blocks using the filter strength value assigned to the category in which that block is categorized.

[0009] Another aspect of the disclosed embodiments is an apparatus for reducing blocking artifacts at the boundary between adjacent blocks reconstructed from a frame of compressed video information that includes a prediction stage parameter and residual error attribute with respect to at least one of the blocks. The apparatus includes means for reconstructing the at least one block based on the prediction stage parameter and the residual error attribute and means for categorizing the at least one reconstructed block into one of a plurality of categories based on the prediction stage parameter and the residual error attribute. The apparatus also includes means for identifying a filter strength value for the category in which the at least one reconstructed block is categorized based on at least one of the prediction stage parameter or the residual error attribute associated with that category. Further, the apparatus includes means for filtering the boundary adjacent to the at least one reconstructed block using the identified filter strength value.

[0010] These and other embodiments of the invention are described in additional detail hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] The description herein makes reference to the accompanying drawings wherein like reference numerals refer to like parts throughout the several views.

[0012] FIG. **1** is a block diagram of a video compression system in accordance with one embodiment of the present invention.

[0013] FIG. **2** is a block diagram of a video decompression system in accordance with one embodiment of the present invention.

[0014] FIG. **3** is a schematic diagram of intra-prediction and inter-prediction modes used in the video compression and decompression systems of FIGS. **1** and **2**.

[0015] FIG. **4** is a block diagram of a loop filter control used to compute a strength modifier used in the video compression system of FIG. **1**.

[0016] FIG. **5** is a flowchart diagram of a method of selecting the strength modifier of FIG. **4**.

[0017] FIG. **6** is a flowchart diagram of a method of updating loop filtering video data used in the video compression system of FIG. **1**.

[0018] FIG. **7** is a flowchart diagram of another method of updating loop filtering video data used in the video compression system of FIG. **1**.

DETAILED DESCRIPTION

[0019] Disclosed herein are embodiments of an adaptive loop filter that remove or reduce blocking artifacts. Further, disclosed herein are embodiments of an adaptive loop filter that either remove or reduce blocking artifacts using less overhead data and/or reduce computational complexity.

[0020] FIG. **1** is a block diagram of a video encoder **14** using an adaptive loop filter **34** according to one embodiment of the present invention.

[0021] In the disclosed embodiments, block-based video compression operates on fixed-shaped groups of neighboring pixels, called a macroblock. In general, each frame of video can be divided into macroblocks, where each macroblock consists of a plurality of smaller-sized blocks. These pixel groups within the macroblocks and blocks can be compared with either data found in the current frame or in other frames in order to formulate motion data and error signals. In this embodiment, each macroblock can be a group of 16×16 pixels. In other embodiments, macroblocks can also be any other suitable size.

[0022] Although the description of embodiments of the adaptive loop filter innovations are described in the context of the VP8 video coding format, alternative embodiments of the present invention can be implemented in the context of other video coding formats. Further, the embodiments are not limited to any specific video coding standard or format.

[0023] To remove discontinuities at block boundaries, loop filtering can be applied to reconstructed frames during a reconstruction path. As explained in more detail below, the choice of loop filter and the strength of the loop filter can have a significant effect on image quality. A filter that is too strong may cause blurring and loss of detail. A filter that it is too weak may not adequately suppress discontinuities between adjacent blocks.

[0024] Referring to FIG. **1**, to encode an input video stream **16**, encoder **14** performs the following functions in a forward path (shown by the solid connection lines) to produce an encoded bitstream **26**: intra/inter prediction **18**, transform **19**, quantization **22** and entropy encoding **24**. Encoder **14** also includes a reconstruction path (shown by the dotted connection lines) to reconstruct a frame for encoding of further macroblocks. Encoder **14** performs the following functions in the reconstruction path: dequantization **28**, inverse transformation **30**, reconstruction **32** and loop filtering **34**. Other structural variations of encoder **14** can be used to encode bitstream **26**.

[0025] Referring to FIG. **1**, when input video stream **16** is presented for encoding, each frame within input video stream **16** can be processed in units of macroblocks. At intra/inter prediction stage **18**, each macroblock can be encoded using either intra prediction or inter prediction mode. In either case, a prediction macroblock can be formed based on a reconstructed frame. In the case of intra-prediction, for example, a prediction macroblock can be formed from samples in the current frame that have been previously encoded and reconstructed. In the case of inter-prediction, for example, a prediction macroblock can be formed from one or more previous or future frames (i.e. reference frames) that have already been encoded and reconstructed. Further, alternate embodiments can encode a macroblock by using some combination of both intra prediction and inter prediction.

[0026] Next, still referring to FIG. **1**, the prediction macroblock can be subtracted from the current macroblock to produce a residual macroblock (residual). Transform stage **19** transform codes the residual and quantization stage **22** quantizes the residual to provide a set of quantized transformed coefficients. The quantized transformed coefficients can be then entropy coded by entropy encoding stage **24**. The entropy-coded coefficients, together with the information required to decode the macroblock, such as the type of prediction mode used, motion vectors and quantizer value, can be outputted to compressed bitstream **26**.

[0027] The reconstruction path in FIG. **1** can be present to permit that both the encoder and the decoder use the same reference frames required to decode the macroblocks. The reconstruction path, similar to functions that take place during the decoding process, which are discussed in more detail below, includes dequantizing the transformed coefficients by dequantization stage **28** and inverse transforming the coefficients by inverse transform stage **30** to produce a derivative residual macroblock (derivative residual). At the reconstruction stage **32**, the prediction macroblock can be added to the derivative residual to create a reconstructed macroblock. The adaptive loop filter **34** can be applied to the reconstructed macroblock to reduce blocking artifacts.

[0028] Referring to FIG. **2**, in accordance with one embodiment, to decode compressed bitstream **26**, a decoder **21** having a structure similar to the reconstruction path of the encoder **14** discussed previously performs the following functions to produce an output video stream **35**: entropy decoding **25**, dequantization **27**, inverse transformation **29**, intra/inter prediction **23**, reconstruction **31**, adaptive loop filter **34** and deblocking filtering **33**. Other structural variations of decoder **21** can be used to decode compressed bitstream **26**.

[0029] When compressed bitstream **26** is presented for decoding, the data elements can be entropy decoded by entropy decoding stage **25** to produce a set of quantized coefficients. Dequantization stage **27** dequantizes and inverse transform stage **29** inverse transforms the coefficients to produce a derivative residual that is identical to that created by the reconstruction stage in the encoder **14**. Using header information decoded from the compressed bitstream **26**, at intra/inter prediction stage **23**, decoder **21** creates the same prediction macroblock as was created in encoder **14**. At the reconstruction stage **31**, the prediction macroblock can be added to the derivative residual to create a reconstructed macroblock. The adaptive loop filter **34** can be applied to the reconstructed macroblock to reduce blocking artifacts. A deblocking filter **33** can be applied to the reconstructed macroblock to further reduce blocking distortion and the result can be outputted to output video stream **35**.

[0030] Although the description of embodiments of the adaptive loop filter innovations are described with reference to adaptive loop filter **34** in the encoder, the described filtering techniques are also implemented in adaptive loop filter **34** in the decoder. Reference to adaptive loop filter **34** in the decoder has been omitted throughout the disclosure only to aid in understanding of the invention. However, the filtering innovations are not limited to adaptive loop filter **34** in the encoder and can be applied to adaptive loop filter **34** in the decoder or any other unit incorporating filtering techniques.

[0031] FIG. **3** shows reference frames **44**, **48** and a current frame **36** that is currently being encoded or decoded. As discussed previously, each frame can be processed in units of macroblocks and at intra/inter prediction stage **18**, each macroblock can be encoded using either intra prediction, inter prediction or some combination of inter prediction and intra prediction. For example, a current macroblock **38** is being encoded or decoded using inter prediction from a macroblock **46** from previously coded reference frame **44**. Similarly, a current macroblock **38'** is being encoded or decoded using inter prediction from a macroblock **50** from previously encoded reference frame **48**. Also, for example, a current macroblock **38"** is being encoded or decoded using intra prediction from a macroblock **52** within current frame **36**.

[0032] Blocking artifacts can be created during the encoding process and can originate from, for example, intra/inter prediction stage **18**, transform stage **19** or quantization stage **22**. Since some conventional filters make filter strength dependent on block boundaries, computational processing can be complex and time-consuming.

[0033] FIG. **4** is a block diagram illustrating a loop filter control **61** of adaptive loop filter **34** in one embodiment of the present invention. According to one embodiment, loop filter control **61** determines strength modifier **60** based on block attributes. Block attributes are based on existing encoded information about a block or information that is passed to the decoder to assist in properly decoding the bitstream.

[0034] Block attributes can include a prediction stage parameter **65** and a residual error attribute **66**. Prediction stage parameter **65** can include a reference frame type **62** and a type of prediction mode **64**. As discussed in more detail below, strength modifier **60** alters the levels of thresholds in adaptive loop filter **34**.

[0035] Reference frame type **62** can be determined by, similar to the illustration in FIG. **3**, whether intra mode or inter frame mode coding is used when constructing prediction blocks. If intra mode predictive coding is used, reference frame type **62** can be intra-frame (i.e. the current frame). When using an intra-frame, the prediction block can be formed, as discussed previously, from samples in the current frame that have been previously encoded and reconstructed.

[0036] If inter mode predictive coding is used, inter-frames can be used as a basis for formulating the prediction block. When using inter-frames, the prediction block can be formed, for example, from one or more previous frames, future frames or some combination thereof that have already been encoded and reconstructed. Accordingly, when using inter-frames, reference frame type **62** may include, for example, a last frame, a golden frame or an alternate reference frame. The last frame can be the previously encoded frame before the current frame. The golden frame can be a past frame chosen arbitrarily from the distant past to use as a predictor for subsequent frames. The alternate reference frame may include any frame that is not the last frame or the golden frame. For example, the alternate reference can be a past frame, a future frame, or a constructed reference frame. Further, for example, the constructed reference may be the reference frame as disclosed in patent application titled "System and Method for Video Encoding Using Constructed Reference Frame" that is assigned to the assignee of the present invention, is filed concurrently herewith and which is hereby incorporated by reference in its entirety.

[0037] Type of prediction mode **64** can be determined, similar to reference frame type **62**, by whether intra mode or inter frame mode coding is used when constructing prediction blocks (as illustrated in FIG. **3**). If intra mode predictive coding is used, two types of intra-coding can be supported which are denoted as non-split mode and split mode. If inter mode predictive coding is used, two types of inter-coding can be supported, which are denoted as non-split mode and split mode.

[0038] If inter mode predictive coding is used with non-split mode, residual error attribute **66** can be determined by whether the resulting motion vector is null or is non-zero.

[0039] As discussed previously, a macroblock can be an array of 16×16 luminance pixels. In intra-coding, each macroblock can be further split into, for example, 4×4 luminance samples referred to as 4×4 sub-blocks. Accordingly, a macroblock can be made of 16 4×4 sub-blocks. This means that a prediction block may be formed for either a macroblock (i.e. non-split mode) or each of the 16 4×4 sub-blocks (i.e. split mode). Other sub-block sizes are also available such as 16×8, 8×16, and 8×8. Although the description of embodiments for intra-coding is described with reference to 4×4 sub-block split mode, any other sub-block size can be used with split mode, and the description of the embodiments is not limited to a 4×4 sub-block.

[0040] In intra-coding, non-split mode results in prediction of the whole 16×16 macroblock whereas split mode leads to separately predicting each 4×4 sub-block.

[0041] For intra-coding non-split mode, for example, one of four prediction modes can be utilized to reference neighboring pixel samples of previously-coded blocks which are to the left and/or above the 16×16 block to be predicted. The four selectable prediction modes may be vertical prediction, horizontal prediction, DC prediction and plane prediction.

[0042] For intra-coding split mode, for example, one of nine prediction modes can be utilized to reference neighboring pixel samples of previously-coded blocks which are to the left and/or above the 4×4 sub-block to be predicted. The nine selectable prediction modes may be vertical prediction, horizontal prediction, DC prediction, diagonal down-left prediction, diagonal down-right prediction, vertical-right prediction, horizontal-down prediction, vertical-left prediction and horizontal-up prediction.

[0043] In inter-coding, non-split mode results in calculating one or motion vectors based on displacing an area of a corresponding reference frame for prediction of the whole 16×16 macroblock. Alternatively, split mode results in calculating a motion vector based on displacing an area of a corresponding reference frame for prediction of a partition of the 16×16 macroblock. The 16×16 macroblock may be split into partitions of 16×8, 8×16, 8×8 or 4×4 each with its own motion vector. Other partition sizes are also available.

[0044] A motion vector can be calculated for each whole macroblock or each separate partition. In particular, motion compensation predicts pixel values of the macroblock (or the corresponding partition within the macroblock) from a translation of the reference frame. The motion vector for each macroblock or partition may either be null, which indicates there has been no change in motion or non-zero, which indicates there has been a change in motion.

[0045] Although the description of embodiments describes how adaptive loop filter **34** applies a different strength modifier **60** based on the prediction stage parameter **65** and residual error attribute **64**, any other loop filter attribute may be varied such as the filter type, filter coefficients, and filter taps, and the description of the embodiments is not limited to varying strength modifier **60**.

[0046] FIG. **5** is a flowchart showing the operation of loop filter control **61** from FIG. **4** according to one embodiment of the present invention. Referring to FIG. **5**, at block **100**, a baseline loop filter strength f can be selected for the frame that defines the behavior of adaptive loop filter **34**. Accordingly, baseline filter strength f will be specified at the frame level in the encoded bitstream. By specifying the baseline filter strength fat the frame level, overhead can be reduced since very few bits can be used to specify a single baseline filter value f for a whole frame. However, even though only one baseline filter strength f can be specified for the frame, filtering quality is not compromised since a filter strength value/modifier **60** alters the levels of thresholds in adaptive loop filter **34**, as discussed below.

[0047] To adjust strength modifier **60** at the macroblock level, delta values **1-8** can be encoded in the bitstream. These delta values are, for example, added to baseline filter strength f. Other suitable procedures for combining baseline filter strength f and strength modifier **60** are also available. Delta values may also be incremental values or percentage increase/decrease values or the like. Delta values may also be positive, negative or zero. Application of the deltas according to the flowchart of FIG. **5** gives rise to 11 different strength modifiers **60** identified as F**1**-F**11**.

[0048] At decision block **102**, control **61** determines whether the current macroblock being reconstructed has been intra-coded.

[0049] If the current macroblock has been intra-coded, delta **1** can be added to baseline filter strength f. Referring back to FIG. **4**, in this case, reference frame type **62** is an intra-frame. Then, control **61** moves to decision block **104**.

[0050] At decision block **104**, control **61** determines whether intra-coding split mode is being used. If intra-coding split mode is being used, delta **2** can be added to delta **1** and baseline filter strength f to yield strength modifier F2. Referring back to FIG. **4**, in this case, prediction mode **64** is intra-coding split mode.

[0051] If intra-coding split mode is not being used (i.e. non-split mode), only delta **1** can be added to baseline filter strength f to yield strength modifier F**1**. Referring back to FIG. **4**, in this case, prediction mode **64** is intra-coding non-split mode.

[0052] If the current macroblock has not been intra-coded, control **61** moves to decision block **106** to determine the type of inter-coded reference frame used. If the last frame is used, delta **3** can be added to baseline filter strength f. Referring back to FIG. **4**, in this case, reference frame type **62** is the last frame. Then, control **61** moves to decision block **108**.

[0053] If a golden frame is used, delta **4** can be added to baseline filter strength f. Referring back to FIG. **4**, in this case, reference frame type **62** is the golden frame. Then, control **61** moves to decision block **110**.

[0054] If an alternate frame is used, delta **5** can be added to baseline filter strength f. Referring back to FIG. **4**, in this case, reference frame type **62** is an alternate frame. Then, control **61** moves to decision block **112**.

[0055] As discussed previously, if the last frame is used, control **61** determines prediction mode **64** at decision block **108**. If inter-coding split mode is being used, delta **8** can be added to baseline filter strength f and delta **3** to yield strength modifier F**5**. Referring back to FIG. **4**, in this case, prediction mode **64** is inter-coding split mode.

[0056] If inter-coding split mode is not being used, control **61** determines whether the calculated motion vector is null or non-zero. If the motion vector is null, delta **6** can be added to baseline filter strength f and delta **3** to yield strength modifier F**3**. Referring back to FIG. **4**, in this case, prediction mode **64** is inter-coding non-split mode, and residual error attribute **66** is a null motion vector. If the motion vector is non-zero, delta **7** can be added to baseline filter strength f and delta **3** to yield strength modifier F**4**. Referring back to FIG. **4**, in this case, prediction mode **64** is inter-coding non-split mode, and residual error attribute **66** is a non-zero motion vector.

[0057] As discussed previously, if a golden frame is used, control **61** determines prediction mode **64** at decision block **110**. If inter-coding split mode is being used, delta **8** can be added to baseline filter strength f and delta **4** to yield strength modifier F**8**. Referring back to FIG. **4**, in this case, prediction mode **64** is inter-coding split mode.

[0058] If inter-coding split mode is not being used with the golden frame, control **61** determines whether the calculated motion vector is null or non-zero. If the motion vector is null, delta **6** can be added to baseline filter strength f and delta **4** to yield strength modifier F**6**. Referring back to FIG. **4**, in this case, prediction mode **64** is inter-coding non-split mode, and residual error attribute **66** is a null motion vector. If the motion vector is non-zero, delta **7** can be added to baseline filter strength f and delta **4** to yield strength modifier F**7**. Referring back to FIG. **4**, in this case, prediction mode **64** is inter-coding non-split mode, and residual error attribute **66** is a non-zero motion vector.

[0059] As discussed previously, if an alternate frame is used, control **61** determines prediction mode **64** at decision block **112**. If inter-coding split mode is being used, delta **8** can be added to baseline filter strength f and delta **5** to yield strength modifier F**11**. Referring back to FIG. **4**, in this case, prediction mode **64** is inter-coding split mode.

[0060] If inter-coding split mode is not being used with the alternate frame, control **61** determines whether the calculated motion vector is null or non-zero. If the motion vector is null, delta **6** can be added to baseline filter strength f and delta **5** to yield strength modifier F**9**. Referring back to FIG. **4**, in this case, prediction mode **64** is inter-coding non-split mode, and residual error attribute **66** is a null motion vector. If the motion vector is non-zero, delta **7** can be added to baseline filter strength f and delta **5** to yield strength modifier F**10**. Referring back to FIG. **4**, in this case, prediction mode **64** is inter-coding non-split mode, and residual error attribute **66** is a non-zero motion vector.

[0061] Generally, different levels of strength modifier **60** are applied to blocking artifacts the extent of which are more or less likely to be present depending on reference frame type **62** and prediction mode **64**. As illustrated in FIG. **5**, for example, a different strength modifier **60** can be applied to intra-coded macroblocks rather than inter-coded macroblocks (i.e. F**1**-F**2** vs. F**3**-F**11**). Further, a different strength modifier **60** can be applied to intra-coded non-split mode macroblocks rather than intra-coded split-mode macroblocks (i.e. F**1** vs. F**2**). Other suitable schemes to adjust strength modifier **60** are also available.

[0062] FIG. **6** is a flowchart of a method of updating loop filtering video data according to one embodiment of the present invention. The loop filtering video data can include both the reference frame loop filter modifiers as well as the prediction mode loop filter modifiers. The reference frame loop filter modifiers can include the delta values for the reference frame type **62**. The prediction mode loop filter modifiers can include the delta values for both prediction mode **64** and the residual error attribute **66**.

[0063] Referring again to FIG. **6**, at decision block **132**, adaptive loop filter **34** determines whether the current frame is a frame coded without reference to any other frame except itself (commonly referred to as a key frame). If the current frame is a key frame, adaptive loop filter **34** moves to block **134** to set reference frame loop filter modifiers to default values. Then, at block **136**, adaptive loop filter **34** sets prediction mode loop filter modifiers to default values. Once the values have been set to default, adaptive loop filter moves to decision block **130** to determine whether a filter condition is enabled at the frame level.

[0064] If current frame is a key frame and the values have been set to default or if the current frame is not a key frame, adaptive loop filter **34** moves to decision block **130** to determine whether a filter condition is enabled at the frame level. Adaptive loop filter **34** can determine whether loop filter modifiers are enabled through a single bit, a byte, a flag or the like.

[0065] If loop filter modifiers are not enabled (i.e. a single loop filter condition has been detected), loop filtering in adaptive loop filter **34** stage can be skipped for the current frame. In other words, a single loop filter strength can be applied to all the blocks within the frame. A single loop filter strength can also include not applying a loop filter for any part of the frame.

[0066] Once loop filtering has been skipped for the current frame, adaptive loop filter will return to decision block **130** to determine whether loop filter modifiers have been enabled for the next frame. Adaptive loop filter **34** may choose to skip loop filtering based on one or more characteristics of the residual error signal, reference frame type **62**, prediction mode **64** or some combination thereof Other suitable factors to skip loop filtering in adaptive loop filter **34** are also available.

[0067] For example, loop filtering may be skipped when there is no AC component of the residual macroblock in transform stage **19** and where the macroblock is inter-coded with a null motion vector. Skipping loop filtering in this instance will prevent repeated loop filtering over several frames in regions of the image where there is no motion. Accordingly, blurring will be reduced and less computations will be involved reducing the overall computational complexity.

[0068] Still referring to FIG. **6**, if loop filter modifiers are enabled, adaptive loop filter **34** moves to decision block **138** to determine whether a loop filter strength value condition has been detected. More specifically, at decision block **138**, adaptive loop filter **34** determines whether there have been any updates to the loop filter modifiers that have been encoded in the current frame. Adaptive loop filter **34** can determine whether loop filter modifiers are to be updated through a single bit, a byte, a flag or the like.

[0069] If there are no updates to loop filter modifiers, adaptive loop filter **34** uses the preset loop filter modifiers from the previous frame to apply to the current frame. Once the previous values have been applied, adaptive loop filter will return to decision block **130** to determine whether loop filter modifiers have been enabled for the next frame.

[0070] If there are updates to loop filter modifiers, adaptive loop filter **34** will update the preset values of reference frame loop filter modifiers at block **140**. Then, adaptive loop filter **34** will move to block **142** to update the preset values of prediction mode loop filter modifiers. Once the values have been updated, adaptive loop filter **34** will return to decision block **130** to determine whether loop filter modifiers have been enabled for the next frame.

[0071] Referring back to FIG. **5**, reference frame loop filter modifiers and prediction mode loop filter modifiers can be delta values **1-8** applied at each of the junctions of the flowcharts. Specifically, delta values **1**, **3**, **4** and **5** can be reference frame loop filter modifiers corresponding to reference frame type **62**, delta values **2** and **8** can be prediction mode loop filter modifiers corresponding to prediction mode **64** and delta values **6** and **7** can be prediction mode loop filter modifiers corresponding to residual error attribute **66**. Each of these delta values can be updated in adaptive loop filter **34** using the method shown in the flowchart of FIG. **6**

[0072] FIG. **7** is a flowchart of a method of updating loop filtering video data according to one embodiment of the present invention. FIG. **7** is similar to the flowchart of FIG. **6** except that adaptive loop filter **34** does not determine whether the current frame is a key frame. Accordingly, if loop filter modifiers are enabled and there are updates to those loop filter modifiers, all frames may update the reference frame loop filter modifiers and prediction mode loop filter modifiers.

[0073] Exemplary pseudo code for implementing the steps of the method in FIG. **7** is shown in Table 1.

TABLE 1

```
// Indicate if Loop modifiers enabled
WriteBit( LoopModifiersEnabled );
if ( LoopModifiersEnabled )
{
  // Are any updates needed
  WriteBit( UpdateLoopModifiers );
  if (UpdateLoopModifiers)
  {
    // Reference frame loop filter modifiers
    for ( i = 0; i < REF_MODIFIERS; i++ )
    {
      Data = RefLfModifiers[i];
      if ( Data )
      {
        Onyx_WriteBit(1);
        // Sign bit
        if ( Data > 0 )
          Onyx_WriteBit(0);
        else
        {
          Onyx_WriteBit(1);
          Data = -Data;
        }
        // 6 bit magnitude
        WriteLiteral( (Data & 0x3F), 6 );
      }
      else
        Onyx_WriteBit(0);
    }
```

TABLE 1-continued

```
    // Prediction mode loop filter modifiers
    for ( i = 0; i < PREDICTION_MODE_MODIFIERS; i++ )
    {
      Data = PredictionModeModifiers[i];
      if ( Data )
      {
        Onyx_WriteBit(1);
        // Sign bit
        if ( Data > 0 )
          Onyx_WriteBit(0);
        else
        {
          Onyx_WriteBit(1);
          Data = -Data;
        }
        // 6 bit magnitude
        WriteLiteral( (Data & 0x3F), 6 );
      }
      else
        Onyx_WriteBit(0);
    }
  }
}
```

[0074] The aforementioned pseudo code is not intended to be limited to any particular programming language and implementation thereof. It will be appreciated that a variety of programming languages and implementations thereof may be used to implement the teachings of embodiments of the invention as described herein.

[0075] Referring to FIGS. 6 and 7 and the aforementioned pseudo code, embodiments of the present invention decrease the amount of overhead in the bitstream. For example, determining whether loop filter modifiers are enabled can be accomplished through a single bit. Further, for example, determining whether updates to loop filter modifiers are enabled can also be accomplished through a single bit.

[0076] While the invention has been described in connection with certain embodiments, it is to be understood that the invention is not to be limited to the disclosed embodiments but, on the contrary, is intended to cover various modifications and equivalent arrangements included within the spirit and scope of the appended claims, which scope is to be accorded the broadest interpretation so as to encompass all such modifications and equivalent structures as is permitted under the law.

What is claimed is:

1. A method for reducing blocking artifacts at the boundary between adjacent blocks reconstructed from a frame of compressed video information that includes a prediction stage parameter and a residual error attribute with respect to at least one of the blocks, the method comprising:

reconstructing the at least one block based on the prediction stage parameter and the residual error attribute;

categorizing the at least one reconstructed block into one of a plurality of categories based on the prediction stage parameter and the residual error attribute;

identifying a filter strength value for the category in which the at least one reconstructed block is categorized based on at least one of the prediction stage parameter or the residual error attribute associated with that category; and

filtering the boundary adjacent to the at least one reconstructed block using the identified filter strength value.

2. The method of claim **1**, wherein the prediction stage parameter includes at least one of a reference frame type and a prediction mode.

3. The method of claim **2**, wherein the reference frame type is one of intra-frame, last frame, golden frame and alternate reference frame.

4. The method of claim **2**, wherein the prediction mode is one of non-split mode and split mode.

5. The method of claim **1**, wherein the residual error attribute is one of a null motion vector and a non-zero motion vector.

6. The method of claim **1**, wherein identifying the filter strength value further comprises:

identifying one or more filter modifiers for the category in which the at least one reconstructed block is categorized based on at least one of the prediction stage parameter or the residual error attribute associated with that category; and

computing the filter strength value based on a baseline filter strength and the one or more filter modifiers.

7. The method of claim **6**, wherein the one or more filter modifiers are identified at the frame level.

8. The method of claim **7**, further comprising:

determining whether the frame being decoded is a key frame; and wherein at least some of the filter modifiers are set to a default value if the frame being decoded is a key frame.

9. A method for decoding frames of compressed video information, each frame including a plurality of blocks having a prediction stage parameter and a residual error attribute, the method comprising:

categorizing the blocks of at least one of the frames into categories based on both the prediction stage parameter and the residual error attribute associated with each given block;

determining a filter strength value for each of the categories;

filtering the boundary adjacent to at least one of the blocks using the filter strength value assigned to the category in which that block is categorized.

10. The method of claim **9**, wherein the prediction stage parameter includes at least one of a reference frame type or a prediction mode.

11. The method of claim **10**, wherein the reference frame type is one of intra-frame, last frame, golden frame and alternate reference frame.

12. The method of claim **10**, wherein the prediction mode is one of non-split mode and split mode.

13. The method of claim **10**, wherein the residual error attribute is one of a null motion vector and a non-zero motion vector.

14. The method of claim **9**, wherein determining the filter strength value includes:

adding, for each of the categories, one or more filter modifiers to a baseline filter strength value.

15. The method of claim **14**, wherein the one or more filter modifiers are determined based on delta values specified in at least one of the frames.

16. The method of claim **14**, further comprising:

determining whether the frame being decoded is a key frame; and wherein at least some of the filter modifiers are set to a default value if the frame being decoded is a key frame.

17. An apparatus for reducing blocking artifacts at the boundary between adjacent blocks reconstructed from a frame of compressed video information that includes a prediction stage parameter and residual error attribute with respect to at least one of the blocks, the apparatus comprising:

means for reconstructing the at least one block based on the prediction stage parameter and the residual error attribute;

means for categorizing the at least one reconstructed block into one of a plurality of categories based on the prediction stage parameter and the residual error attribute;

means for identifying a filter strength value for the category in which the at least one reconstructed block is categorized based on at least one of the prediction stage parameter or the residual error attribute associated with that category; and

means for filtering the boundary adjacent to the at least one reconstructed block using the identified filter strength value.

18. The apparatus of claim **17**, wherein the prediction stage parameter includes at least one of a reference frame type and a prediction mode.

19. The apparatus of claim **18**, wherein the reference frame type is one of intra-frame, last frame, golden frame and alternate reference frame and the prediction mode is one of non-split mode and split mode.

20. The apparatus of claim **17**, wherein the filter strength value is identified at the frame level.

* * * * *