



- (51) **International Patent Classification:**
H04N 7/26 (2006.01) *H04N 7/36* (2006.01)
- (21) **International Application Number:**
PCT/US2012/031579
- (22) **International Filing Date:**
30 March 2012 (30.03.2012)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
13/081,662 7 April 2011 (07.04.2011) US
- (71) **Applicant (for all designated States except US):**
GOOGLE INC. [US/US]; 1600 Amphitheatre Parkway,
Mountain View, California 48033 (US).
- (72) **Inventors; and**
- (75) **Inventors/Applicants (for US only):** **BANKOSKI, James**
[US/US]; c/o Google Inc., 1600 Amphitheatre Parkway,
Mountain View, California 94043 (US). **WILKINS, Paul**
[GB/GB]; c/o Google Inc., 1600 Amphitheatre Parkway,
Mountain View, California 94043 (US). **XU, Yaowu**
[US/US]; c/o Google Inc., 1600 Amphitheatre Parkway,
Mountain View, California 94043 (US).
- (74) **Agents:** **BASILE, Andrew R., Jr.** et al.; Young Basile
Hanlon & Macfarlane, P.C., 3001 West Big Beaver Road,
Suite. 624, Troy, Michigan 48084 (US).
- (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, BR, BW, BY, BZ,
CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO,
DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN,
HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR,
KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME,
MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ,
OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, 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, SZ, TZ,
UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, 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, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) **Title:** ENCODING AND DECODING MOTION VIA IMAGE SEGMENTATION

(57) **Abstract:** A system and method for encoding and decoding a video image having a plurality of frames is disclosed. The method includes a previously reconstructed frame that is segmented to create an alternate reference frame used to encode a current frame. The segments of the previously reconstructed frame are assigned motion parameters, which are used to compile the alternate reference frame. The current frame is encoded and the segmentation parameters and motion parameters are transmitted in the bitstream. The segmentation parameters and motion parameters can be used by a decoder to recreate the alternate reference frame in order to decode the encoded current frame.

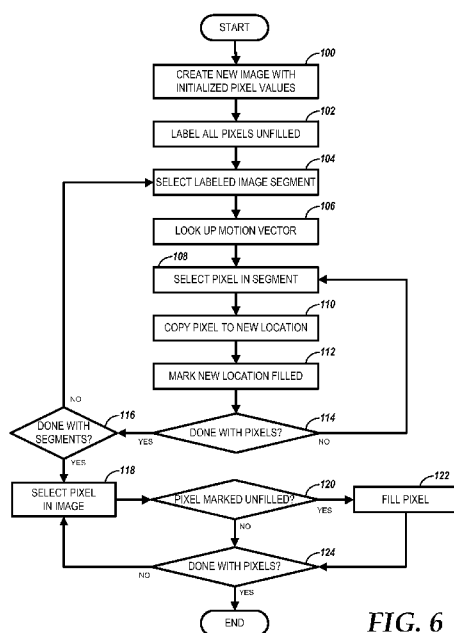


FIG. 6



Published:

— *with international search report (Art. 21(3))*

ENCODING AND DECODING MOTION VIA IMAGE SEGMENTATION

TECHNICAL FIELD

[0001] The present invention relates in general to video encoding and decoding.

BACKGROUND

[0002] 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, users have higher expectations for video quality and expect high resolution video even when transmitted over communications channels having limited bandwidth.

[0003] To permit higher quality transmission of video while limiting bandwidth consumption, a number of video compression schemes are noted including formats such as VPx, promulgated by Google Inc. of Mountain View, California, and H.264, a 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).

[0004] These compression schemes may use prediction techniques to minimize the amount of data required to transmit video information by using, for example, the spatial and temporal coherences in video information. Prediction techniques can allow for multiple past transmitted frames and future frames to be used as potential reference frame predictors for macroblocks in a frame.

SUMMARY

[0005] Systems, apparatuses and methods for encoding and decoding a video signal are disclosed. In accordance with one aspect of the disclosed embodiments, a method for encoding a current frame of a video signal having a plurality of frames, each frame having a plurality of pixels, is disclosed. One such method comprises segmenting a previously reconstructed frame into at least one image segment using a segmentation parameter, each image segment having a segment identifier, determining a motion parameter for each image segment using a processor, creating an alternate reference frame

from the previously reconstructed frame using the motion parameter for at least certain of the image segments, and encoding the current frame using the alternate reference frame for inclusion in a bitstream.

[0006] An apparatus for encoding a current frame of a video signal having a plurality of frames, each frame having a plurality of pixels, includes in one exemplary embodiment means for segmenting a previously reconstructed frame into at least one image segment using a segmentation parameter, each image segment having a segment identifier, means for determining a motion parameter for each image segment, means for creating an alternate reference frame from the previously reconstructed frame using the motion parameter for each image segment, and means for encoding the current frame using the alternate reference frame for inclusion in a bitstream.

[0007] One method for decoding a video signal having a plurality of frames taught herein, where each frame has a plurality of pixels, comprises identifying a current frame to be decoded from a bitstream using a processor, the current frame having an associated previously reconstructed frame, a segmentation parameter, a plurality of segment identifiers and a motion parameter associated with each segment identifier, segmenting the previously reconstructed frame into a plurality of image segments using the segmentation parameter, each image segment associated with one of the segment identifiers, creating an alternate reference frame from the previously reconstructed frame using the motion parameter of at least certain of the image segments, and decoding the current frame using the alternate reference frame.

[0008] Variations of these methods and apparatuses and details of other aspects and embodiments are described in additional detail hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The description herein makes reference to the accompanying drawings wherein like reference numerals refer to like parts throughout the several views, and wherein:

[0010] FIG. 1 is a diagram of a video bitstream;

[0011] FIG. 2 is a block diagram of a video compression system in accordance with one embodiment;

[0012] FIG. 3 is a block diagram of a video decompression system in accordance with another embodiment;

[0013] FIG. 4 is a flow chart of a method of encoding with a new alternate reference frame synthesized using segmentation and motion compensation as implemented in the encoder of FIG. 2;

[0014] FIG. 5 is a flow chart of a method of decoding with a new alternate reference frame synthesized using segmentation and motion compensation as implemented in the decoder of FIG. 3;

[0015] FIG. 6 is a flow chart of a method of creating a new alternate reference frame using segments and motion vectors as shown in FIGS. 4 and 5;

[0016] FIG. 7 is a schematic diagram illustrating exemplary frames of a video bitstream as used and created by the methods of FIGS. 4-6; and

[0017] FIG. 8 is a block diagram of computer systems implementing the video compression system and video decompression system described herein.

DETAILED DESCRIPTION

[0018] FIG. 1 is a diagram a typical video bitstream 10 to be encoded and decoded. Video coding formats, such as VP8 or H.264, provide a defined hierarchy of layers for video stream 10. Video stream 10 includes a video sequence 12. At the next level, video sequence 12 consists of a number of adjacent frames 14, which can then be further subdivided into a single frame 16. At the next level, frame 16 can be divided into a series of blocks or macroblocks 18, which can contain data corresponding to, for example, a 16x16 block of displayed pixels in frame 16. Each macroblock can contain luminance and chrominance data for the corresponding pixels. Macroblocks 18 can also be of any other suitable size such as 16x8 pixel groups or 8x16 pixel groups. Herein, unless otherwise stated, the terms macroblocks and blocks are used interchangeably.

[0019] FIG. 2 is a block diagram of a video compression system in accordance with one embodiment. An encoder 20 encodes an input video stream 10. Encoder 20 has the following stages to perform the various functions in a forward path (shown by the solid connection lines) to produce an encoded or a compressed bitstream 24: an intra/inter prediction stage 26, a transform stage 28, a quantization stage 30 and an entropy encoding stage 32. Encoder 20 also includes a reconstruction path (shown by the dotted connection lines) to reconstruct a frame for encoding of further macroblocks. Encoder 20 has the following stages to perform the various functions in the reconstruction path: a dequantization stage 34, an inverse transform stage 36, a reconstruction stage 37 and a

loop filtering stage 38. Other structural variations of encoder 20 can be used to encode input video stream 10.

[0020] When input video stream 10 is presented for encoding, each frame 16 within input video stream 10 is processed in units of macroblocks. At intra/inter prediction stage 26, each macroblock can be encoded using either intra-frame prediction (i.e., within a single frame) or inter-frame prediction (i.e. from frame to frame). In either case, a prediction macroblock can be formed. In the case of intra-prediction, 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, a prediction macroblock can be formed from samples in one or more previously constructed reference frames as described in additional detail herein.

[0021] Next, still referring to FIG. 2, the prediction macroblock can be subtracted from the current macroblock at stage 26 to produce a residual macroblock (residual). Transform stage 28 transforms the residual into transform coefficients in, for example, the frequency domain, and quantization stage 30 converts the transform coefficients into discrete quantum values, which are referred to as quantized transform coefficients or quantization levels. The quantized transform coefficients are then entropy encoded by entropy encoding stage 32. The entropy-encoded coefficients, together with the information required to decode the macroblock, such as the type of prediction used, motion vectors, and quantizer value, are then output to compressed bitstream 24.

[0022] The reconstruction path in FIG. 2 is present to ensure that both encoder 20 and a decoder 42 (described below) use the same reference frames to decode compressed bitstream 24. The reconstruction path performs functions that are similar to functions that take place during the decoding process that are discussed in more detail below, including dequantizing the quantized transform coefficients at dequantization stage 34 and inverse transforming the dequantized transform coefficients at an inverse transform stage 36 in order to produce a derivative residual macroblock (derivative residual). At reconstruction stage 37, the prediction macroblock that was predicted at intra/inter prediction stage 26 can be added to the derivative residual to create a reconstructed macroblock. A loop filter 38 can then be applied to the reconstructed macroblock to reduce distortion such as blocking artifacts.

[0023] Other variations of encoder 20 can be used to encode compressed bitstream 24. For example, a non-transform based encoder can quantize the residual signal directly without transform stage 28. In another embodiment, an encoder may have

quantization stage 30 and dequantization stage 34 combined into a single stage. In yet another embodiment, a temporal filter may be included with the intra/inter prediction stage 24.

[0024] The encoding process shown in FIG. 2 can include two iterations or “passes” of processing the video data. The first pass can be carried out by encoder 20 using an encoding process that is less computationally intensive, and that gathers and stores information about input video stream 10 for use in the second pass. In the second pass, encoder 20 uses this information to optimize final encoding of compressed bitstream 24. For example, encoder 20 may use this information to select parameters for encoding, locating key-frames and selecting coding modes used to encode macroblocks 18, and allocating the number of bits to each frame. The output of the second pass can be final compressed bitstream 24.

[0025] FIG. 3 is a block diagram of a video decompression system or decoder 42 to decode compressed bitstream 24. Decoder 42, similar to the reconstruction path of the encoder 20 discussed previously, includes the following stages to perform various functions to produce an output video stream 44 from compressed bitstream 24: an entropy decoding stage 46, a dequantization stage 48, an inverse transform stage 50, an intra/inter prediction stage 52, a reconstruction stage 54, a loop filter stage 56 and a deblocking filtering stage 58. Other structural variations of decoder 42 can be used to decode compressed bitstream 24.

[0026] When compressed bitstream 24 is presented for decoding, the data elements within compressed bitstream 24 can be decoded by entropy decoding stage 46 (using, for example, Context Adaptive Binary Arithmetic Decoding) to produce a set of quantized transform coefficients. Dequantization stage 48 dequantizes the quantized transform coefficients, and inverse transform stage 50 inverse transforms the dequantized transform coefficients to produce a derivative residual that can be identical to that created by the reconstruction stage in the encoder 20. Using header information decoded from the compressed bitstream 24, decoder 42 can use intra/inter prediction stage 52 to create the same prediction macroblock as was created in encoder 20. At the reconstruction stage 54, the prediction macroblock can be added to the derivative residual to create a reconstructed macroblock. The loop filter 56 can be applied to the reconstructed macroblock to reduce blocking artifacts. Deblocking filter 58 can be applied to the reconstructed macroblock to reduce blocking distortion, and the result is output as output video stream 44.

[0027] Other variations of decoder 42 can be used to decode compressed bitstream 24. For example, a decoder may produce output video stream 44 without deblocking filtering stage 58.

[0028] Referring again to encoder 20, video encoding methods compress video signals by using lossless or lossy compression algorithms to compress each frame or blocks of each frame of a series of frames. As can be implied from the description above, intra-frame coding refers to encoding a frame using data from that frame, while inter-frame coding refers to predictive encoding schemes such as schemes that comprise encoding a frame based on other so-called “reference” frames. For example, video signals often exhibit temporal redundancy in which frames near each other in the temporal sequence of frames have at least portions that match or at least partially match each other. Encoders can take advantage of this temporal redundancy to reduce the size of encoded data by encoding a frame in terms of the difference between the current frame and one or more reference frames.

[0029] As described briefly above, many video coding algorithms first partition each picture into macroblocks. Then, each macroblock can be coded using some form of predictive coding method. Some video coding standards use different types of predicted macroblocks in their coding. In one scenario, a macroblock may be one of three types: 1) Intra (I) macroblock that uses no information from other pictures in its coding; 2) Unidirectionally Predicted (P) macroblock that uses information from one preceding picture; and 3) Bidirectionally Predicted (B) macroblock that uses information from at least one past picture and at least one future picture.

[0030] To facilitate higher quality compressed video, it is helpful to have a matching reference frame in order to have the smallest difference to encode, which generally results in a more compact encoding. Reference frames are based on past frames, future frames, or an intra-frame so that the encoder can find a matching block to use in the predictive process as shown in, for example, U.S. Application Publication No. 2005/0286629. Reference frames can also be based on synthesized or constructed frames that are not shown to the end user after decoding, as shown in, for example, in U.S. Application Publication No. 2010/0061461 A1. Such constructed reference frames are referred to herein as “alternate reference frames.” The alternate reference frames are constructed from macroblocks, which are referred to herein as “alternate reference blocks.”

[0031] Another method of creating and using an alternate reference frame is described herein. FIG. 4 is a flow chart of a method of encoding with a new alternate reference frame synthesized using segmentation and motion compensation as implemented in the encoder of FIG. 2. First, a reconstructed frame is selected and segmented (60). The reconstructed frame is obtained from the reconstruction path, and typically would be output from the loop filter 38. The reconstructed frame may be segmented into two or more segments of at least one pixel using a variety of segmentation parameters. For example, the reconstructed frame could be segmented using color segmentation or motion segmentation.

[0032] Color segmentation divides a frame into segments based upon the colors presented in the captured image. More specifically, when a camera captures images, the original color, typically RGB, for each pixel is converted and stored as a signal including separate luminance (or luma) and chrominance (or chroma) values. That is, each pixel is represented by a luma value Y and by two values collectively forming the chroma value, a chroma value U and a chroma value V . Luma values Y of the image represent the achromatic portion of an image (that is, the image without color), while the chroma values U and V are sometimes called color difference components and generally represent the color of the image. Most commonly, each luma value Y is a weighted sum of gamma-compressed red, green and blue components ($R'G'B'$) represented by Y' , and chroma values U and V are respectively equal to $B'-Y'$ and $R'-Y'$ represented by Cb and Cr . Here, the actual color format is not critical or limited and can be any color format or colorspace.

[0033] Encoder 20 and decoder 42 code macroblocks 18 of frame 16 using image formats that rely on chroma, chrominance or chromatic sampling or subsampling. That is, because human vision is more sensitive to black and white differences than to chromatic differences, a lower sampling rate (fewer bits) can be used to represent the chroma values of pixels of each frame 16 than to represent the luma values of those pixels. By example, encoder 20 and decoder 42 can code macroblocks 18 using 4:2:0 YUV chroma subsampling. This scheme halves the amount of data as compared to an image with no chroma subsampling as only one chroma value for each of chroma values U and V is sampled for every four luma values Y . This essentially means that while there are two chroma values for each group of four image pixels, there are four luma values.

[0034] As described above, when input video stream 10 is presented for encoding, each frame 16 within input video stream 10 is processed into units of macroblocks. At

intra/inter prediction stage 26 of encoder 20, each macroblock can be encoded using either an intra prediction or inter prediction mode. The predictions for luma values Y and for chroma values U and V of pixels of the macroblock can be calculated independently of each other. Further, and in regard to the predictions for chroma values U and V, a prediction mode can treat these values identically (i.e., use the same relative addressing and arithmetic). In certain embodiments, predictions for chroma values U and V may be different. In the examples herein, pixels are considered with regard to their combined color value YUV.

[0035] Returning now to FIG. 4, a frame can be segmented (60) by segmenting individual macroblocks or by segmenting the frame in its entirety. As one example, segmenting can be performed by quad-tree decomposition (or deconstruction). One implementation of quad-tree decomposition initially subdivides the image into four equal-size blocks and then selectively subdivides the smaller blocks in a like manner if the smaller blocks have more than one different color value YUV for pixels in that block. That is, each block is subdivided into groups of four until each set of blocks represents pixels having a same color value YUV.

[0036] Note that while each set of blocks is said to represent pixels having the same color value YUV, in actuality the pixels in a set have values Y, U and V that are not necessarily the same. Instead, the blocks are sub-divided by defining an acceptable variance for each of the values Y, U and V for the pixels that form each set of blocks. In this manner, each set of blocks can be represented by three representative values for luma and chroma values. The representative values can be average values using the luma and chroma values of the individual pixels comprising a set. Alternatively, the representative values could be some other combination of the respective luma and chroma values of the pixels comprising that set. As one example, each of the representative values for luma and chroma could be average values of only certain of the pixel values, such as those that fall within a defined range about a median value.

[0037] In the example above, partitioning continues until the number of values YUV in a set is equal to one (that is, all the pixel values Y, U, V are within a maximum variance). The frame and/or each macroblock could instead be partitioned into sets of pixels based on another desirable threshold. For example, partitioning could occur until a fixed number of sets is defined.

[0038] Segmenting a frame (60) is not limited to quad-tree decomposition and can be implemented by binary-tree or octree decomposition in a similar manner. Also, these

methods of segmenting a frame can be combined such that part of the segmentation is performed by one decomposition method while another part of the segmentation is performed by another decomposition method.

[0039] Another method of segmenting a frame (60) involves k-means clustering techniques, or similar, to produce either the smallest set of entries that ensures the reconstruction error is below a defined threshold, or a fixed number of entries. As known to those skilled in the art, k-means clustering is a method that can be used to partition n values into k clusters where each value belongs in the cluster with the nearest mean. In this case, for example, the number k can be predetermined for either a macroblock 18 or frame 16 such that a fixed number of clusters results.

[0040] An alternative implementation could first use a k-means segmentation to split the entire frame (irrespective of block boundaries) into N regions and then use binary-tree or quad-tree decomposition to encode region membership. Decomposition is stopped when all pixels in a sub-region have the same region membership, which is then encoded as a leaf node.

[0041] While the descriptions above implement segmentation using color, in another embodiment the frame could be segmented (60) by motion segmentation. Motion segmentation allows for division of a frame by identifying as segments various components that are independently moving in an image. For example, the image deconstruction or k-means segmentation described above could be implemented using the magnitude and direction of a motion vector as the variables for each pixel to be considered. Motion segmentation would require at least two adjacent reconstructed frames to establish the movement needed for segmentation. Other types of segmentation are also possible and may be used, such as expectation-maximization clustering techniques.

[0042] Segmentation may result in any number of different segments, each segment having any number of pixels. The methods disclosed herein may be used with any shape or size segment. The number of segments is based on accuracy versus the amount of data to be encoded (i.e., the bit cost). More segments require more bits for encoding, but the fewer the number of segments, generally the less accurate the reconstruction of the segmented images. Known techniques for comparing the bit cost of various encoding options can be incorporated with the techniques described herein to determine a desired number of segments.

[0043] Referring back to FIG. 4, once the segments have been determined (60), they are labeled with a segment identifier for processing (62). The segment identifier can be, for instance, a sequentially generated number created in the order of segment creation. Once the segments are labeled (62), a segment is selected for motion estimation processing (64). In this example where the motion parameter is a motion vector, a motion vector search is performed on the selected segment (66). A motion vector can be found by any motion estimation method such as full search or diamond search. One useful method is described in, for example, Hamosfakidis et al., "A Novel Hexagonal Search Algorithm for Fast Block Matching Motion Estimation," EURASIP Journal on Applied Signal Processing 2002:6, 595-600. Other methods are available and will be apparent to those skilled in the art.

[0044] Although this example describes motion estimation processing as searching for a motion vector (66), and the following steps describe using the motion vector as a motion parameter for the creation of the new alternate reference frame, the process described herein is not limited to using motion vectors to describe the movement of each segment. That is, segments may undergo movement other than no movement (zero motion vector) or simple translation (a motion vector having a non-zero magnitude and direction). Use of other motion parameters such as those associated with a zoom, a rotation or a shear is possible. Obtaining these motion parameters can be handled using an object based affine transform such as known to those skilled in the art. Examples of motion of a segment are described with respect to FIG. 7 hereinafter.

[0045] Once the motion vector for the selected segment is determined (66), the encoder 20 determines if the currently selected segment is the last available segment (68). If not, the next available segment is selected (64) and its motion vector is determined (66). Once a motion vector has been determined for each segment (68), a new alternate reference frame is created (70) using the segments and motion vectors. An exemplary process for creating the alternate reference frame is described in more detail later with respect to FIG. 6.

[0046] Once the alternate reference frame is created, it is then available in the intra/inter prediction stage 26 for encoding frames (72). The identification of segments and their motion vectors are transmitted in compressed bitstream 24 (74) for use by the decoder 42 to recreate the alternate reference frame for decoding those blocks/frames encoded by encoder 20 with reference to the alternate reference frame. In one embodiment, it is desirable to transmit the segment and its motion parameter, such as the

motion vector, only if that segment has greater than a minimum number of pixels, such as 20. Such segments may also optionally be omitted when creating the alternate reference frame.

[0047] As mentioned, each motion parameter (e.g., a motion vector) transmitted is associated with a segment. For example, the association can be made by transmitting the segment identifier along with each motion vector. Additionally, an identifier of the segmentation parameters and method used to segment the frame may be transmitted to the decoder 42 so that the decoder 42 can recreate the same segments. Alternately, the segmentation parameters and method may be pre-negotiated between the encoder 20 and decoder 42 so that they need not be transmitted each time in step 74.

[0048] FIG. 5 is a flow chart of a method of decoding with a new alternate reference frame synthesized using segmentation and motion compensation as implemented in the decoder of FIG. 3. First, a frame to be decoded is identified (80). In this example, the frame to be decoded was encoded by the encoder 20 using the alternate reference frame described with reference to FIG. 4. Once the frame is identified, the segmentation parameters used to segment the frame by the encoder 20 are determined (82). As described above, the segmentation parameters may be transmitted in the compressed bitstream 24 along with the frame to be decoded or may be pre-negotiated between the encoder 20 and decoder 42.

[0049] Once the segmentation parameters and method are determined, decoder 42 will then segment the frame to be decoded (84). The alternate reference frame will be created using motion vectors and segment labels encoded into the compressed bitstream 24 (86). An exemplary process for creating the alternate reference frame is described in more detail later with respect to FIG. 6. Once the alternate reference frame has been created, the frame to be decoded will be decoded using the new alternate reference frame (88).

[0050] Now, with respect to FIG. 6, a method of creating a new alternate reference frame using segments and motion vectors is described. The method uses segments and motion vectors as provided by either the encoder 20 or decoder 42 as described with respect to FIGS. 4 and 5 above. First, a frame is initialized with initial pixel values (100). The initial pixel values may be any desirable default value, such as zero. Each pixel is initially marked as unfilled (102).

[0051] A first segment is selected (104) thereafter, and the motion vector related to the segment is looked up (106). The motion vector may be associated with the segment by way of a segment identifier as described previously.

[0052] A pixel in the segment is then selected (108). The new location of the pixel is calculated using the previously selected motion vector. Alternatively, the pixel may have more than one new location, depending on the type of motion parameter associated with the segment. Once the at least one new location is determined, the pixel is copied to that at least one new location (110).

[0053] Typically, the pixel will be copied to the new location even if that pixel location already has been filled with a pixel from another segment. But alternatively, the new pixel may be averaged or by some other means combined with the pixel that was previously copied. The new location is then marked as a filled location (112). If there are additional pixels in the current segment to be processed (114), the next pixel is selected (108), the new pixel is copied to its new location (110), and the new located in marked as filled (112). This process repeats until all pixels in the current segment are moved by the segment's motion vector (114).

[0054] If additional segments need to be processed (116), a new segment is selected (104) so as to move its pixels in accordance with the motion vector for that new segment. Otherwise, a first pixel from the alternate reference frame is selected (118). A query is then made as to determine whether the pixel has been filled (120). If the pixel is not marked as filled, the pixel is filled (122). The pixel may be filled by, for example, determining the value of a nearby filled pixel and copying that pixel value to the currently selected unfilled pixel. Alternatively, the pixel may be filled by calculating an average value of nearby filled pixels. The pixel may be filled by copying a pixel from a reconstructed frame. Other methods of filling unfilled pixels are available and may be used.

[0055] Once the unfilled pixel has been filled (122) or if the current pixel was already filled (120), the method determines whether there are additional pixels to process (124). If so, control returns to select the next available pixel (118). Otherwise, the method ends as the new alternate reference frame is completed.

[0056] FIG. 7 is a schematic diagram illustrating exemplary frames of a video bitstream as used and created by the methods of FIGS. 4-6. FIG. 7 shows an exemplary application of color segmentation. The figure includes a reconstructed frame X 150. The reconstructed frame X 150 can include macroblocks, such as macroblock 152.

Reconstructed frame X contains a circular object 154. In this example, circular object 154 contains color values distinct from the remainder of the reconstructed frame X 150.

[0057] Circular object 154 occupies space within a macroblock cluster 156. Each macroblock in macroblock cluster 156 may include one or more pixels comprising circular object 154. For example, macroblocks 156a-d contain a varying percentage of pixels that are a part of circular object 154. For example, while macroblock 156d is entirely inclusive of circular object 154 pixels, only a small portion of macroblock 156c includes circular object 154 pixels.

[0058] The application of reconstructed frame X 150 to the new alternate reference frame encoding method described in FIG. 4 will now be described. As applied to the method, reconstructed frame X 150 is analogous to the reconstructed frame of step 60. The circular object 154 portrays an exemplary segment that could, for example, be created in the segmentation process of steps 60 and 62. The exemplary segment, in this case, is defined by the color of circular object 154.

[0059] The segment is defined at a granularity of size less than a macroblock. For example, the entire macroblock 156c is not included in the segment defined by circular object 154, since circular object 154 only has a small number of pixels within macroblock 156c.

[0060] The segment of circular object 154 can be used to predict other frames, such as original frame Y 160. Original frame Y 160 also includes macroblocks such as macroblock 162. Original frame Y 160 includes a second circular object 164. In this example, circular object 164 is the same size, shape, and color as circular object 154, but is in a different location within the frame. The relative location is illustrated by the dotted circle indicating the position of circular object 154 with respect to original frame Y 160 (154').

[0061] To utilize reconstructed frame X 150 as a basis for a new alternate reference frame 170 for original frame Y 160, a motion vector 166 must be determined for the exemplary segment. The motion vector 166 is determined as discussed in the motion vector search step 66. The new alternate reference frame 170 is created using the reconstructed frame X 150, the exemplary segment corresponding to the circular object 154 and the motion vector 166. The process of creating the new reference frame is as defined in step 70 and in the method of FIG. 6.

[0062] The exemplary segment is moved pixel by pixel into the new alternate reference frame 170, creating a second circular segment predictor 176. The new alternate

reference frame 170 is composed of macroblocks such as macroblock 172. Even though the second circular object predictor 176 is within a macroblock cluster 174, only the pixels needed to create the second circular object predictor 176 are moved. Every pixel within macroblock cluster 174 is not necessarily moved to create the new reference frame.

[0063] The segment of circular object 154 can also be used to predict another frame, such as original frame Z 180. Original frame Z 180 also includes macroblocks such as macroblock 182. Original frame Z 180 includes a third circular object 184 that is the same shape and color as circular object 156, but is in a different location within the frame and is a larger size. Essentially, this reflects the existence of movement of third circular object 184 indicating both a zoom and a simple translation. The relative location is illustrated by the dotted circle indicating the position of circular object 154 with respect to original frame Z 180 (154'').

[0064] The third circular object 184 illustrates the application of more than one motion parameter to the segment of circular object 154 to create an alternate reference frame 200. Alternate reference frame 200 includes blocks such as macroblock 202. Motion vector 186 illustrates the transposition of the position of the circular object 156 to the position of circular object 184. The new position of circular object 154 after applying motion vector 186 is shown by the dotted circle location 188. The arrows 190 depict a zoom-type operation on circular object 154 at location 188 that would be represented by other motion parameters. These multiple motion parameters would be determined, for example, in the motion vector search (66) of FIG. 4.

[0065] The exemplary frames shown in FIG. 7 are included purely for illustrative purposes to demonstrate one application of the methods described above. Numerous other applications, including those with much more complex frames and segments are possible. In addition, variations of the methods disclosed above may be used. These variations may have more or less steps or may perform steps in different orders. For example, the segment labeling (62) of FIG. 4 may not be performed, or elements of segment labeling (62) may be performed within another step, such as segmenting the reconstruction frame (60). Another exemplary modification may be to perform the motion vector search (66) for each segment in parallel instead of sequentially (64, 66 and 68) as shown in FIG. 4.

[0066] The operation of encoding can be performed in many different ways and can produce a variety of encoded data formats. The above-described embodiments of

encoding or decoding may illustrate some exemplary encoding techniques. However, in general, encoding and decoding are understood to include any transformation or any other change of data whatsoever.

[0067] Encoder 20 and/or decoder 42 (and the algorithms, methods, instructions etc. stored thereon and/or executed thereby) can be implemented in whole or in part by one or more processors which can include computers, servers, or any other computing device or system capable of manipulating or processing information now-existing or hereafter developed including optical processors, quantum processors and/or molecular processors. Suitable processors also include, for example, general purpose processors, special purpose processors, IP cores, ASICS, programmable logic arrays, programmable logic controllers, microcode, firmware, microcontrollers, microprocessors, digital signal processors, memory, or any combination of the foregoing. In the claims, the term “processor” should be understood as including any the foregoing, either singly or in combination. The terms “signal” and “data” are used interchangeably.

[0068] Encoder 20 and/or decoder 42 also include a memory, which can be connected to the processor through, for example, a memory bus. The memory may be read only memory or random access memory (RAM) although any other type of storage device can be used. Generally, the processor receives program instructions and data from the memory, which can be used by the processor for performing the instructions. The memory can be in the same unit as the processor or located in a separate unit that is coupled to the processor.

[0069] For example, encoder 20 can be implemented using a general purpose processor with a computer program that, when executed, carries out any of the respective methods, algorithms and/or instructions described herein. FIG. 8 illustrates one suitable implementation in which encoder 20 is implemented in a general purpose computer including a central processing unit (CPU) 212 and random access memory (RAM) 214. Decoder 42 is implemented using a general purpose computer including a central processing unit (CPU) 216 and random access memory (RAM) 218. In addition or alternatively, for example, a special purpose processor can be utilized which can contain specialized hardware for carrying out any of the methods, algorithms and/or instructions described herein. Portions of encoder 20 or decoder 42 do not necessarily have to be implemented in the same manner. Thus, for example, intra/inter prediction stage 26 can be implemented in software whereas transform stage 28 can be implemented in hardware. Portions of encoder 20 or portions of decoder 42 may also be distributed across multiple

processors on the same machine or different machines or across a network such as a local area network, wide area network or the Internet.

[0070] Encoder 20 and decoder 42 can, for example, be implemented in a wide variety of configurations, including for example on servers in a video conference system. Alternatively, encoder 20 can be implemented on a server and decoder 42 can be implemented on a device separate from the server, such as a hand-held communications device such as a cell phone. In this instance, encoder 20 can compress content and transmit the compressed content to the communications device, using the Internet for example, as shown in FIG. 8. In turn, the communications device can decode the content for playback. Alternatively, the communications device can decode content stored locally on the device (i.e. no transmission is necessary). Other suitable encoders and/or decoders are available. For example, decoder 42 can be on a personal computer rather than a portable communications device.

[0071] The operations of encoder 20 or decoder 42 (and the algorithms, methods, instructions etc. stored thereon and/or executed thereby) can be realized in hardware, software or any combination thereof. All or a portion of embodiments of the present invention can take the form of a computer program product accessible from, for example, a computer-usable or computer-readable medium. A computer-usable or computer-readable medium can be any device that can, for example tangibly contain, store, communicate, and/or transport the program for use by or in connection with any processor. The medium can be, for example, an electronic, magnetic, optical, electromagnetic, or a semiconductor device. Other suitable mediums are also available.

[0072] The above-described embodiments have been described in order to allow easy understanding of the present invention and do not limit the present invention. On the contrary, the invention is intended to cover various modifications and equivalent arrangements included within the scope of the appended claims, which scope is to be accorded the broadest interpretation so as to encompass all such modifications and equivalent structure as is permitted under the law.

What is claimed is:

1. A method for decoding a video signal having a plurality of frames, each frame having a plurality of pixels, the method comprising:

identifying a current frame to be decoded from a bitstream using a processor, the current frame having an associated previously reconstructed frame, a segmentation parameter, a plurality of segment identifiers and a motion parameter associated with each segment identifier;

segmenting the previously reconstructed frame into a plurality of image segments using the segmentation parameter, each image segment associated with one of the segment identifiers;

creating an alternate reference frame from the previously reconstructed frame using the motion parameter of at least certain of the image segments; and

decoding the current frame using the alternate reference frame.

2. The method of claim 1 wherein the segmentation parameter is at least one of a color or a motion vector of each pixel in the previously reconstructed frame.

3. The method of claim 1 or claim 2 wherein creating the alternate reference frame comprises:

calculating a new position for each pixel of each image segment using the motion parameter of that respective image segment;

comparing the new position to a range of positions of the alternate reference frame;

when the new position is within the range of positions of the alternate reference frame, copying a value associated with the pixel in the new position in the alternate reference frame;

selecting an unfilled pixel of the alternate reference frame that has no value copied to it after values for all pixels of the previously reconstructed frame having the new position within the range of positions of the alternate reference frame are copied into the new position in the alternate reference frame;

determining a desired value for the unfilled pixel; and

copying the desired value to the unfilled pixel.

4. The method of claim 1 or claim 2 wherein the previously reconstructed frame is one of a golden frame, an original alternate reference frame, or a last frame decoded.

5. The method of claim 1 or claim 2 wherein creating the alternate reference frame comprises:

selecting an image segment of the previously reconstructed frame;
calculating a new position for each pixel of the image segment using the motion parameter of that image segment;

comparing the new position to a range of positions of the alternate reference frame; and

when the new position is within the range of positions of the alternate reference frame, copying a value associated with the pixel in the new position in the alternate reference frame.

6. The method of claim 5 wherein creating the alternate reference frame comprises:

performing each of selecting the image segment, calculating the new position, comparing the new position and, when the new position is within the range of positions, copying the value for each image segment of the previously reconstructed frame.

7. The method of claim 6 wherein creating the alternate reference frame comprises:

selecting an unfilled pixel of the alternate reference frame that has no value copied to it after values for all pixels of the previously reconstructed frame having the new position within the range of positions of the alternate reference frame are copied into the new position in the alternate reference frame;

determining a desired value for the unfilled pixel; and

copying the desired value to the unfilled pixel.

8. The method of claim 1 or claim 2 wherein the motion parameter for each image segment is at least one of a motion vector or a parameter reflecting at least one of a zoom, a rotation or a shear of the image segment.

9. A method for encoding a current frame of a video signal having a plurality of frames, each frame having a plurality of pixels, the method comprising:
- segmenting a previously reconstructed frame into at least one image segment using a segmentation parameter, each image segment having a segment identifier;
 - determining a motion parameter for each image segment using a processor;
 - creating an alternate reference frame from the previously reconstructed frame using the motion parameter for at least certain of the image segments; and
 - encoding the current frame using the alternate reference frame.
10. The method of claim 9, further comprising:
- identifying an association between the current frame and:
 - the previously reconstructed frame,
 - the motion vector for each segment, and
 - the segmentation parameter.
11. The method of claim 10 wherein identifying an association comprises including at least one of the identified associations in the encoded output.
12. The method of claim 9 or claim 10 wherein segmenting the previously reconstructed frame comprises segmenting the previously reconstructed frame using at least one of:
- k-means clustering, binary tree deconstruction, quad-tree deconstruction, or oct-tree deconstruction; and the segmentation parameter comprises at least one of color or a motion vector for each pixel of the previously reconstructed frame.
13. The method of claim 9 or claim 10 wherein creating the alternate reference frame from the previously reconstructed frame excludes those image segments and their respective motion parameters having fewer than a predetermined number of pixels.
14. An apparatus for encoding a current frame of a video signal having a plurality of frames, each frame having a plurality of pixels, the apparatus comprising:

means for segmenting a previously reconstructed frame into at least one image segment using a segmentation parameter, each image segment having a segment identifier;

means for determining a motion parameter for each image segment;

means for creating an alternate reference frame from the previously reconstructed frame using the motion parameter for at least certain of the image segments; and

means for encoding the current frame using the alternate reference frame.

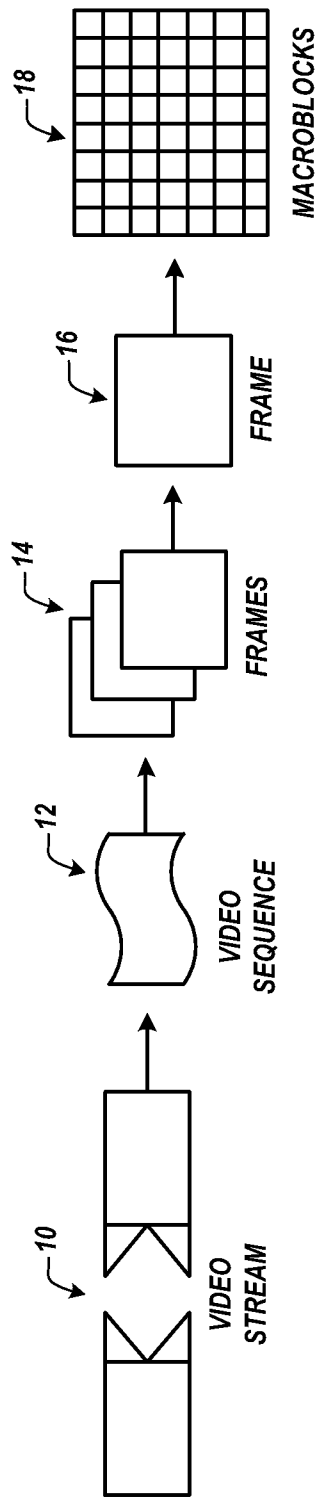


FIG. 1

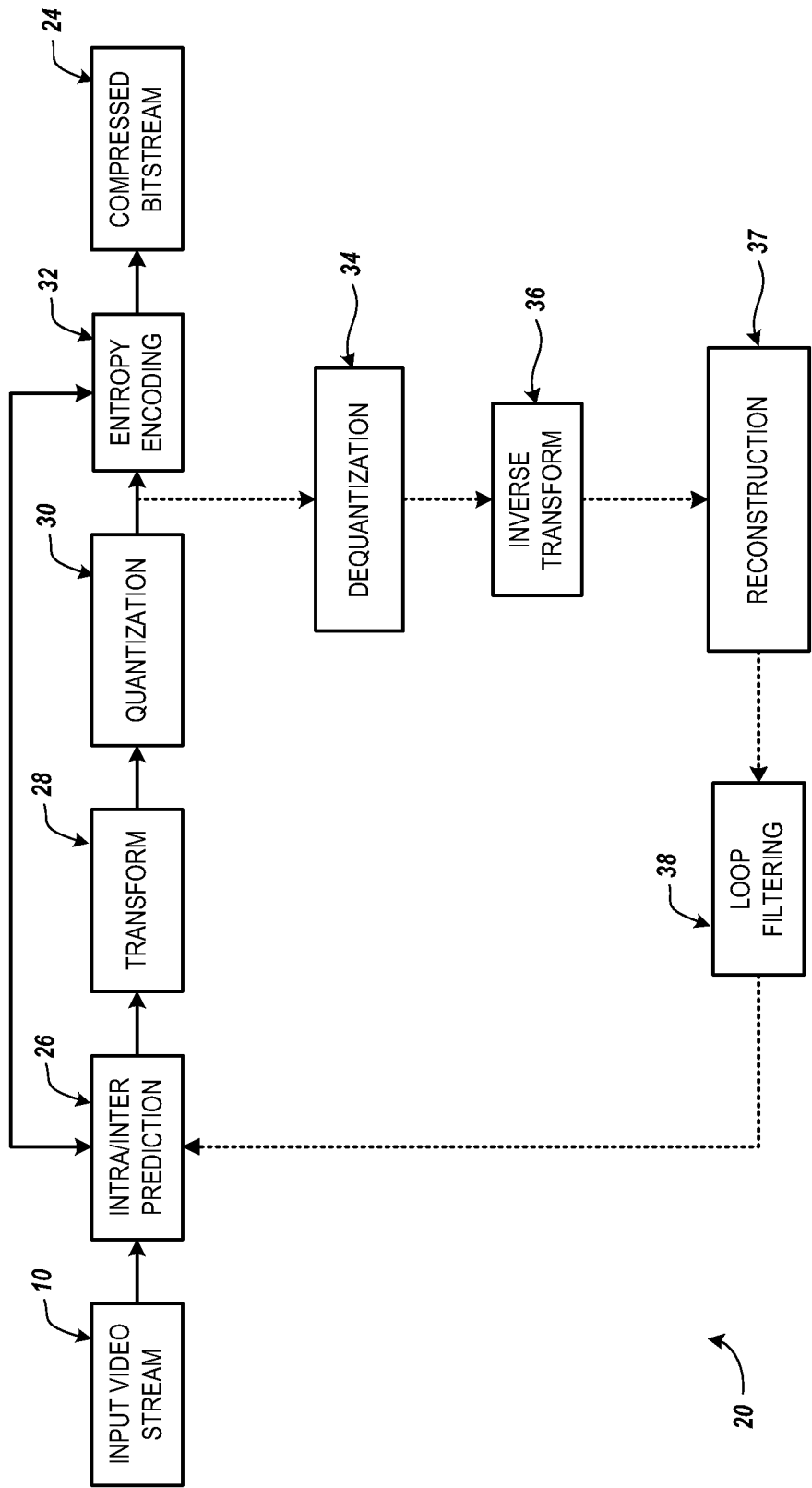


FIG. 2

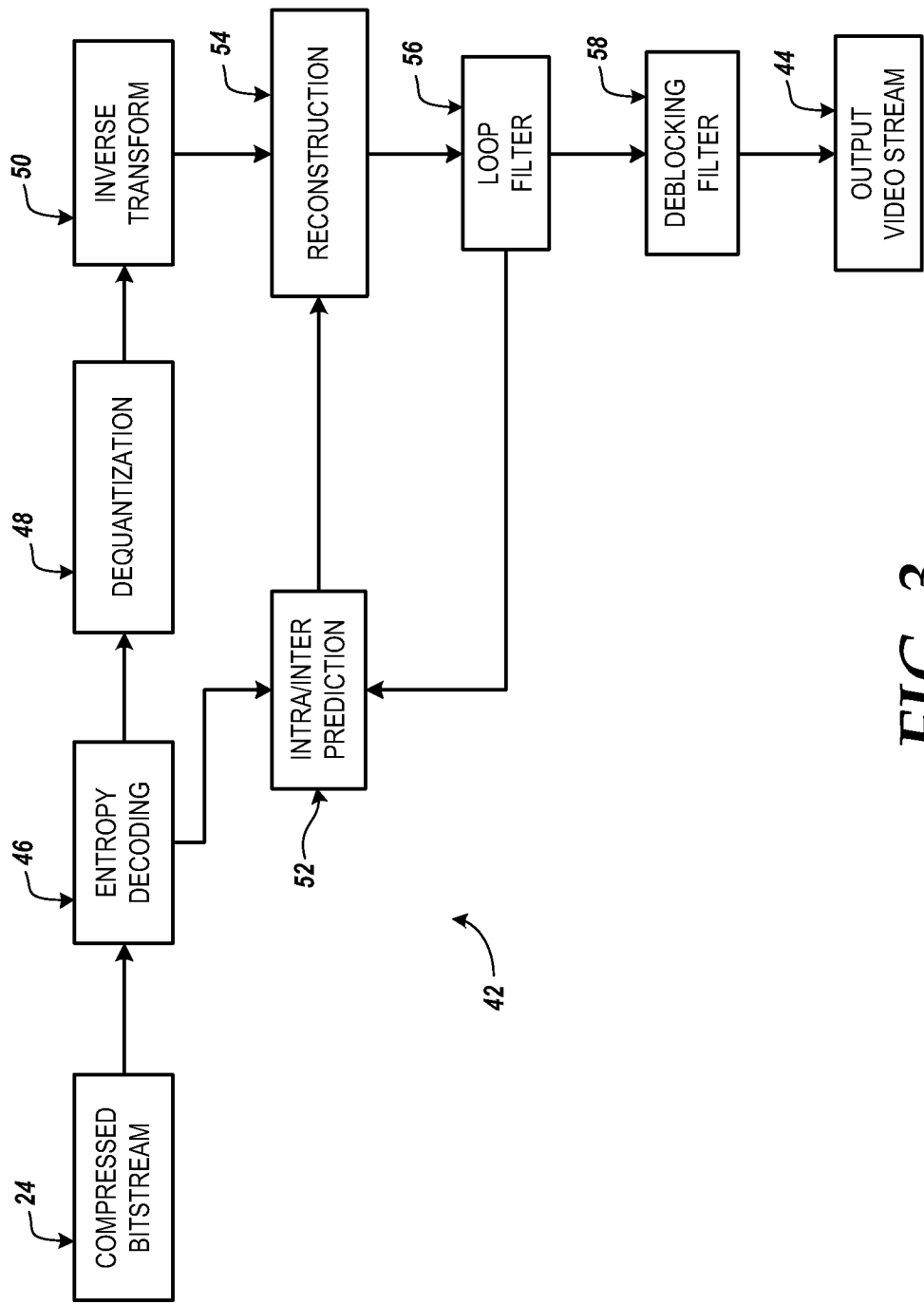
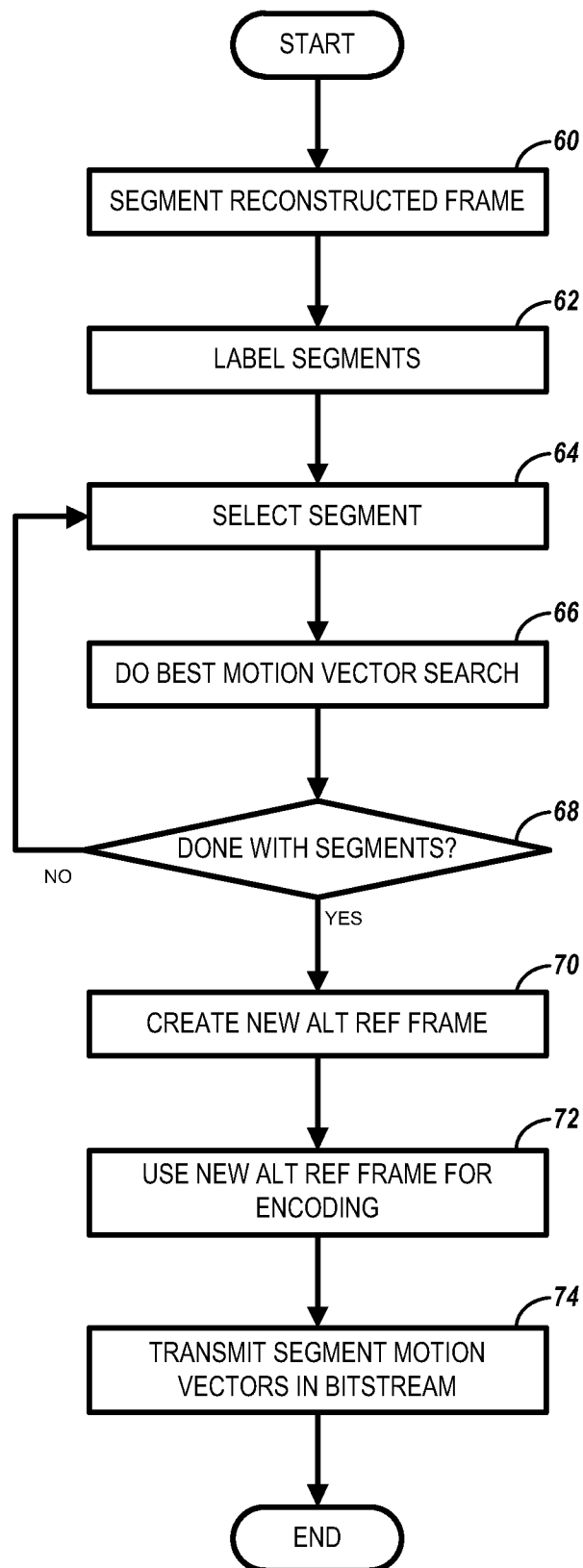
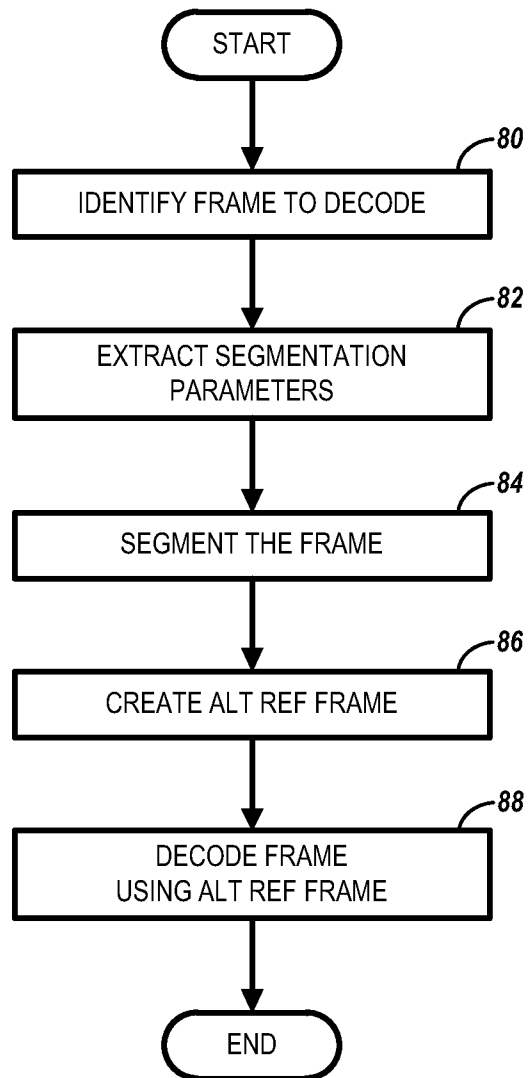


FIG. 3

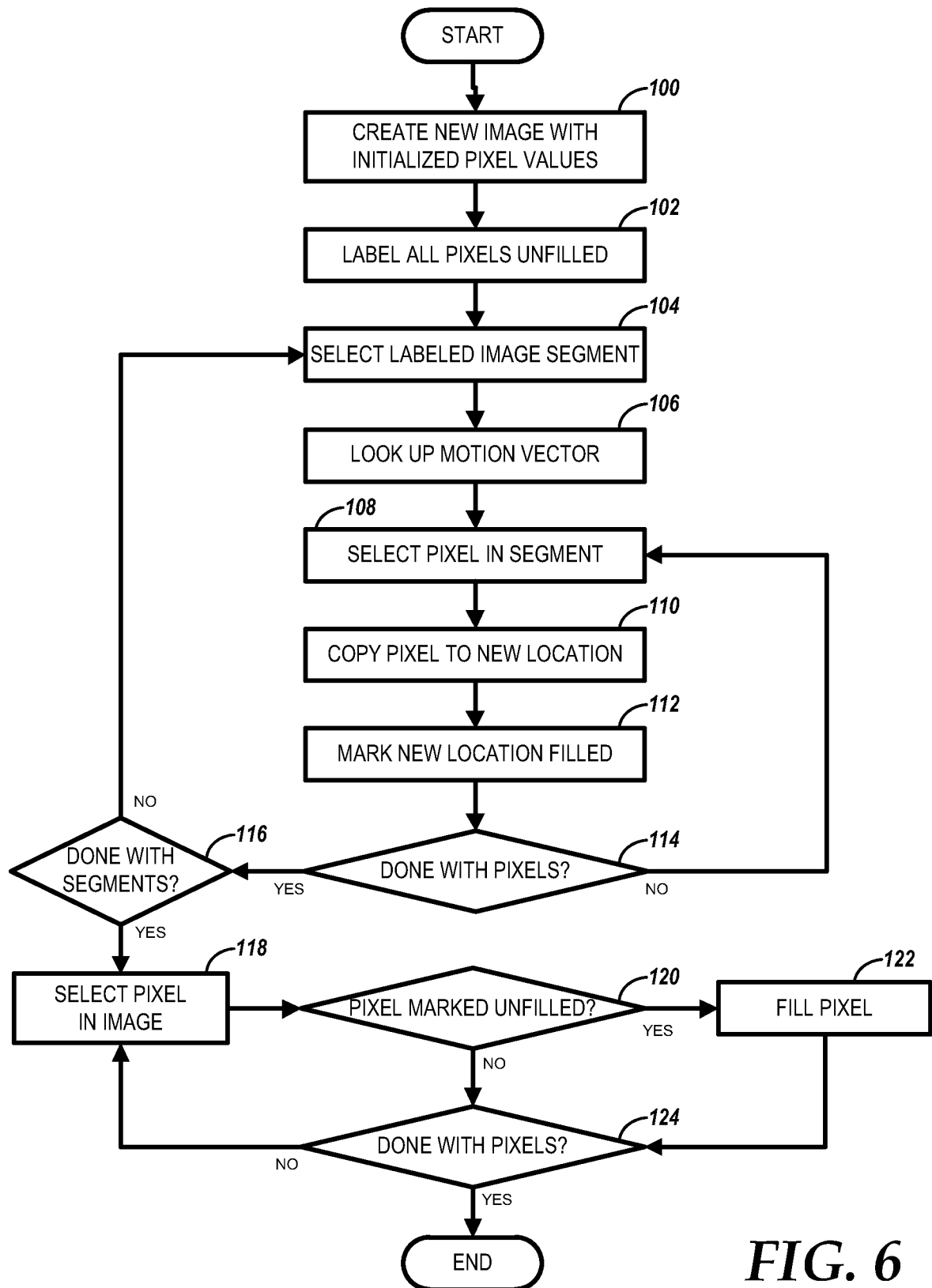
4/8

**FIG. 4**

5/8

**FIG. 5**

6/8

**FIG. 6**

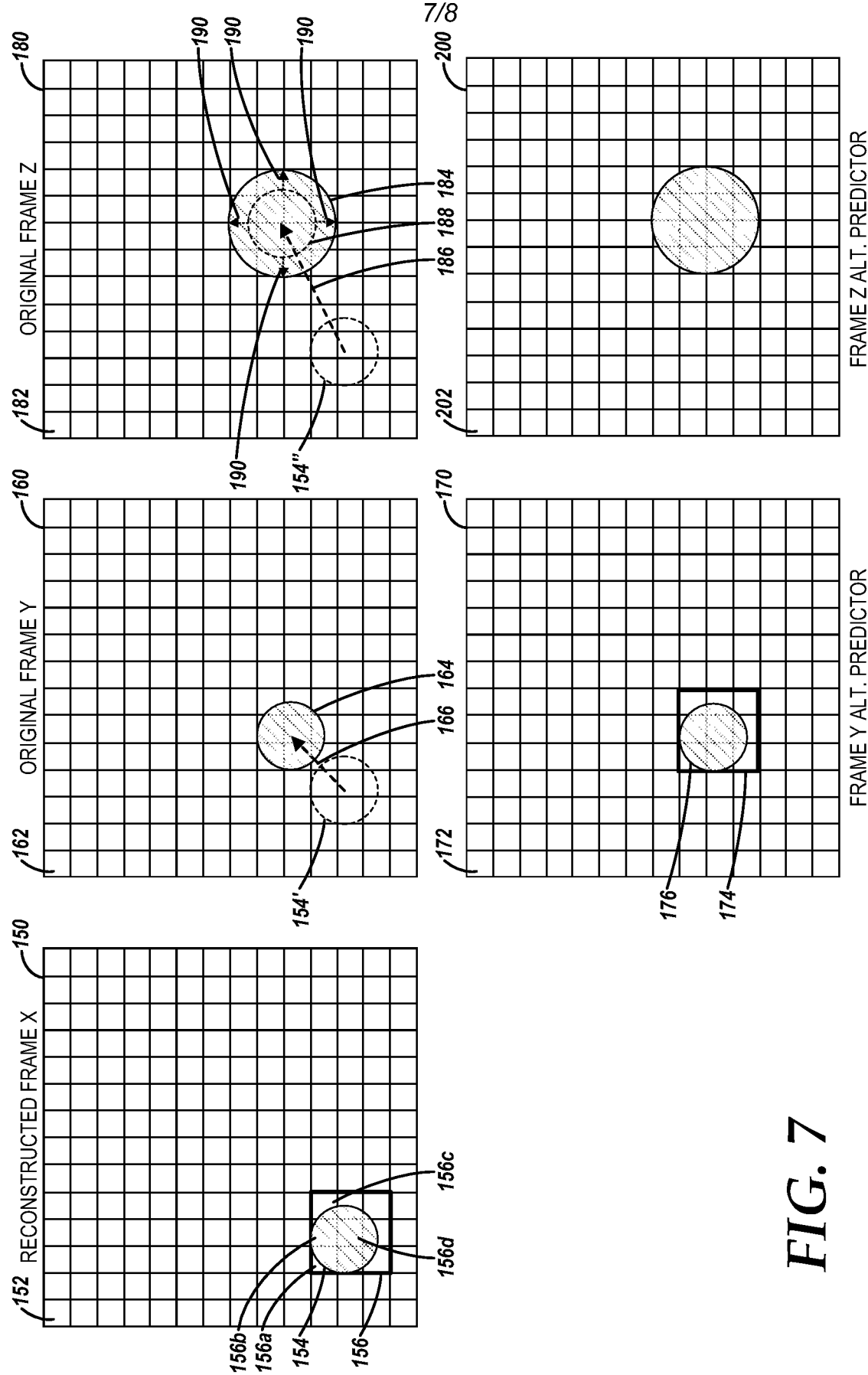
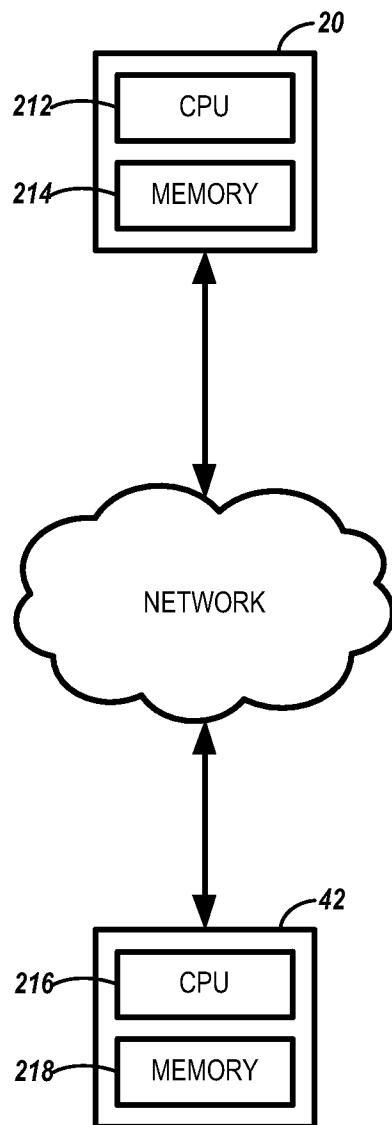


FIG. 7

8/8

**FIG. 8**

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2012/031579

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N7/26 H04N7/36
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04N G06T

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

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

EP0-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2003/123545 A1 (PRAKASH ADITYO [US] ET AL) 3 July 2003 (2003-07-03) paragraphs [0032], [0057], [0077] - [0093], [0126] - [0136] -----	1-14
X	WO 2006/078115 A1 (SAMSUNG ELECTRONICS CO LTD [KR]; CHA SANG-CHANG [KR]; HAN WOO-JIN [KR]) 27 July 2006 (2006-07-27) paragraphs [0014], [0015], [0043], [0048] - [0056] figure 12 -----	1,3-11, 14
A	----- -/--	2,12,13

☒ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

21 June 2012

Date of mailing of the international search report

29/06/2012

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Montoneri, Fabio

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2012/031579

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JOAO CARREIRA ET AL: "Constrained parametric min-cuts for automatic object segmentation", 2010 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 13-18 JUNE 2010, SAN FRANCISCO, CA, USA, IEEE, PISCATAWAY, NJ, USA, 13 June 2010 (2010-06-13), pages 3241-3248, XP031725892, ISBN: 978-1-4244-6984-0 the whole document	2,12,13
A	----- US 2010/061461 A1 (BANKOSKI JAMES [US] ET AL) 11 March 2010 (2010-03-11) cited in the application paragraphs [0006], [0007], [0035] - [0037]	1-14
A	----- US 2010/061444 A1 (WILKINS PAUL [GB] ET AL) 11 March 2010 (2010-03-11) paragraphs [0006], [0034] - [0045] -----	1-14

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2012/031579

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2003123545 A1	03-07-2003	NONE	
WO 2006078115 A1	27-07-2006	NONE	
US 2010061461 A1	11-03-2010	CA 2736886 A1 CN 102150429 A EP 2327212 A2 JP 2012502590 A KR 20110059853 A US 2010061461 A1 WO 2010030732 A2	18-03-2010 10-08-2011 01-06-2011 26-01-2012 07-06-2011 11-03-2010 18-03-2010
US 2010061444 A1	11-03-2010	CA 2736890 A1 CN 102150428 A EP 2324637 A2 JP 2012502593 A KR 20110056388 A US 2010061444 A1 WO 2010030761 A2	18-03-2010 10-08-2011 25-05-2011 26-01-2012 27-05-2011 11-03-2010 18-03-2010