(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 April 2014 (03.04.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/051712 A1

(51) International Patent Classification:
H04N 7/26 (2006.01) *H04N 7/36* (2006.01)

(21) International Application Number:
PCT/US2013/043884

(22) International Filing Date:
3 June 2013 (03.06.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/707,650 28 September 2012 (28.09.2012) US
13/725,940 21 December 2012 (21.12.2012) US
13/797,644 12 March 2013 (12.03.2013) US

(71) Applicant: **EUCLID DISCOVERIES, LLC** [US/US]; 30 Monument Square, Suite 212, Concord, MA 01742 (US).

(72) Inventors: **DEFOREST, Darin**; 30 Monument Square, Suite 212, Concord, MA 01742 (US). **PACE, Charles, P.**; 30 Monument Square, Suite 212, Concord, MA 01742 (US). **LEE, Nigel**; 30 Monument Square, Suite 212, Concord, MA 01742 (US). **PIZZORNI, Renato**; 30 Monument Square, Suite 212, Concord, MA 01742 (US).

(74) Agents: **WAKIMURA, Mary, Lou** et al.; Hamilton, Brook, Smith & Reynolds, P.C., 530 Virginia Rd., P.O. Box 9133, Concord, MA 01742-9133 (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, BN, 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, 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, PA, 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, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

(54) **Title**: STANDARDS-COMPLIANT MODEL-BASED VIDEO ENCODING AND DECODING

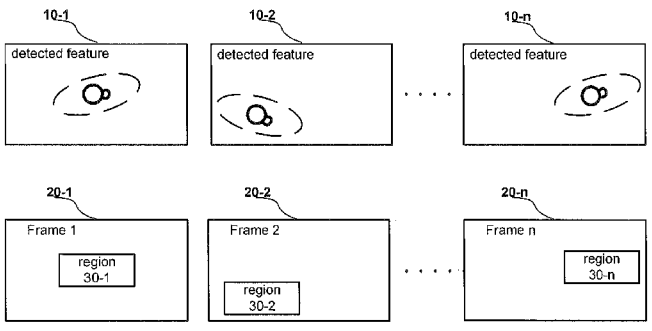

FIG. 1A

(57) **Abstract**: A model-based compression codec applies higher-level modeling to produce better predictions than can be found through conventional block-based motion estimation and compensation. Computer- vision-based feature and object detection algorithms identify regions of interest throughout the video datacube. The detected features and objects are modeled with a compact set of parameters, and similar feature/object instances are associated across frames. Associated features/objects are formed into tracks and related to specific blocks of video data to be encoded. The tracking information is used to produce model-based predictions for those blocks of data, enabling more efficient navigation of the prediction search space than is typically achievable through conventional motion estimation methods. A hybrid framework enables modeling of data at multiple fidelities and selects the appropriate level of modeling for each portion of video data. A compliant-stream version of the model-based compression codec uses the modeling information indirectly to improve compression while producing bitstreams that can be interpreted by standard decoders.

WO 2014/051712 A1

STANDARDS-COMPLIANT MODEL-BASED VIDEO ENCODING AND DECODING

RELATED APPLICATION(S)

[0001] This application claims the benefit of U.S. Provisional Application No. 61/707,650, filed on September 28, 2012. This application also claims priority to U.S. Utility Application Nos. 13/725,940, filed December 21, 2012 and 13/797,644, filed March 12, 2013. U.S. Application Nos. 13/725,940, filed December 21, 2012 and 13/797,644, filed March 12, 2013 are also continuations-in part of U.S. Patent Application No. 13/121,904, filed October 6, 2009, which is a U.S. National Stage of PCT/US2009/059653 filed October 6, 2009, which claims the benefit of U.S. Provisional Application No. 61/103,362, filed October 7, 2008. The '904 application is also a continuation-in part of U.S. Patent Application No. 12/522,322, filed January 4, 2008, which claims the benefit of U.S. Provisional Application No. 60/881,966, filed January 23, 2007, is related to U.S. Provisional Application No. 60/811,890, filed June 8, 2006, and is a continuation-in-part of U.S. Application No. 11/396,010, filed March 31, 2006, now U.S. Patent No. 7,457,472, which is a continuation-in-part of U.S. Application No. 11/336,366 filed January 20, 2006, now U.S. Patent No. 7,436,981, which is a continuation-in-part of U.S. Application No. 11/280,625 filed November 16, 2005, now U.S. Patent No. 7,457,435, which is a continuation-in-part of U.S. Application No. 11/230,686 filed September 20, 2005, now U.S. Patent No. 7,426,285, which is a continuation-in-part of U.S. Application No. 11/191,562 filed July 28, 2005, now U.S. Patent No. 7,158,680. U.S. Application No. 11/396,010 also claims priority to U.S. Provisional Application No. 60/667,532, filed March 31, 2005 and U.S. Provisional Application No. 60/670,951, filed April 13, 2005.

This present application is also related to U.S. Provisional Application No. 61/616,334, filed March 27, 2012, U.S. Provisional Application No. 61/650,363 filed May 22, 2012 and U.S. Application No. 13/772,230 filed February 20, 2013 which claims the benefit of the '334 and '363 Provisional Applications.

[0002] The entire teachings of the above applications are incorporated herein by reference.

BACKGROUND

[0003] Video compression can be considered the process of representing digital video data in a form that uses fewer bits when stored or transmitted. Video compression algorithms can achieve compression by exploiting redundancies and irrelevancies in the video data, whether spatial, temporal, or color-space. Video compression algorithms typically segment the video data into portions, such as groups of frames and groups of pels, to identify areas of redundancy within the video that can be represented with fewer bits than the original video data. When these redundancies in the data are reduced, greater compression can be achieved. An encoder can be used to transform the video data into an encoded format, while a decoder can be used to transform encoded video back into a form comparable to the original video data. The implementation of the encoder/decoder is referred to as a codec.

[0004] Standard encoders divide a given video frame into non-overlapping *coding units* or macroblocks (rectangular regions of contiguous pels) for encoding. The macroblocks are typically processed in a traversal order of left to right and top to bottom in the frame. Compression can be achieved when macroblocks are predicted and encoded using previously-coded data. The process of encoding macroblocks using spatially neighboring samples of previously-coded macroblocks within the same frame is referred to as intra-prediction. Intra-prediction attempts to exploit spatial redundancies in the data. The encoding of macroblocks using similar regions from previously-coded frames, together with a motion estimation model, is referred to as inter-prediction. Inter-prediction attempts to exploit temporal redundancies in the data.

[0005] The encoder may measure the difference between the data to be encoded and the prediction to generate a residual. The residual can provide the difference between a predicted macroblock and the original macroblock. The encoder can generate motion vector information that specifies, for example, the location of a macroblock in a reference frame relative to a macroblock that is being encoded or decoded. The predictions, motion vectors (for inter-prediction), residuals, and related data can be combined with other processes such as a spatial transform, a

quantizer, an entropy encoder, and a loop filter to create an efficient encoding of the video data. The residual that has been quantized and transformed can be processed and added back to the prediction, assembled into a decoded frame, and stored in a framestore. Details of such encoding techniques for video will be familiar to a person skilled in the art.

[0006] H.264/MPEG-4 Part 10 AVC (advanced video coding), hereafter referred to as H.264, is a codec standard for video compression that utilizes block-based motion estimation and compensation and achieves high quality video representation at relatively low bitrates. This standard is one of the encoding options used for Blu-ray disc creation and within major video distribution channels, including video streaming on the internet, video conferencing, cable television and direct-broadcast satellite television. The basic coding units for H.264 are 16x16 macroblocks. H.264 is the most recent widely-accepted standard in video compression.

[0007] The basic MPEG standard defines three types of frames (or pictures), based on how the macroblocks in the frame are encoded. An I-frame (intra-coded picture) is encoded using only data present in the frame itself. Generally, when the encoder receives video signal data, the encoder creates I frames first and segments the video frame data into macroblocks that are each encoded using intra-prediction. Thus, an I-frame consists of only intra-predicted macroblocks (or "intra macroblocks"). I-frames can be costly to encode, as the encoding is done without the benefit of information from previously-decoded frames. A P-frame (predicted picture) is encoded via forward prediction, using data from previously-decoded I-frames or P-frames, also known as *reference frames*. P-frames can contain either intra macroblocks or (forward-)predicted macroblocks. A B-frame (bi-predictive picture) is encoded via bidirectional prediction, using data from both previous and subsequent frames. B-frames can contain intra, (forward-)predicted, or bi-predicted macroblocks.

[0008] As noted above, conventional inter-prediction is based on block-based motion estimation and compensation (BBMEC). The BBMEC process searches for the best match between the target macroblock (the current macroblock being encoded) and similar-sized regions within previously-decoded reference frames. When a best match is found, the encoder may transmit a motion vector. The motion

vector may include a pointer to the best match's frame position as well as information regarding the difference between the best match and the corresponding target macroblock. One could conceivably perform exhaustive searches in this manner throughout the video "datacube" (height x width x frame index) to find the best possible matches for each macroblock, but exhaustive search is usually computationally prohibitive. As a result, the BBMEC search process is limited, both temporally in terms of reference frames searched and spatially in terms of neighboring regions searched. This means that "best possible" matches are not always found, especially with rapidly changing data.

[0009] A particular set of reference frames is termed a Group of Pictures (GOP). The GOP contains only the decoded pels within each reference frame and does not include information as to how the macroblocks or frames themselves were originally encoded (I-frame, B-frame or P-frame). Older video compression standards, such as MPEG-2, used one reference frame (the previous frame) to predict P-frames and two reference frames (one past, one future) to predict B-frames. The H.264 standard, by contrast, allows the use of multiple reference frames for P-frame and B-frame prediction. While the reference frames are typically temporally adjacent to the current frame, there is also accommodation for the specification of reference frames from outside the set of the temporally adjacent frames.

[0010] Conventional compression allows for the blending of multiple matches from multiple frames to predict regions of the current frame. The blending is often linear, or a log-scaled linear combination of the matches. One example of when this bi-prediction method is effective is when there is a fade from one image to another over time. The process of fading is a linear blending of two images, and the process can sometimes be effectively modeled using bi-prediction. Some past standard encoders such as the MPEG-2 interpolative mode allow for the interpolation of linear parameters to synthesize the bi-prediction model over many frames.

[0011] The H.264 standard also introduces additional encoding flexibility by dividing frames into spatially distinct regions of one or more contiguous macroblocks called slices. Each slice in a frame is encoded (and can thus be decoded) independently from other slices. I-slices, P-slices, and B-slices are then defined in a manner analogous to the frame types described above, and a frame can

consist of multiple slice types. Additionally, there is typically flexibility in how the encoder orders the processed slices, so a decoder can process slices in an arbitrary order as they arrive to the decoder.

[0012] Historically, model-based compression schemes have been proposed to avoid the limitations of BBMEC prediction. These model-based compression schemes (the most well-known of which is perhaps the MPEG-4 Part 7 standard) rely on the detection and tracking of objects or features in the video and a method for encoding those features/objects separately from the rest of the video frame. These model-based compression schemes, however, suffer from the challenge of segmenting video frames into object vs. non-object (feature vs. non-feature) regions. First, because objects can be of arbitrary size, their shapes need to be encoded in addition to their texture (color content). Second, the tracking of multiple moving objects can be difficult, and inaccurate tracking causes incorrect segmentation, usually resulting in poor compression performance. A third challenge is that not all video content is composed of objects or features, so there needs to be a fallback encoding scheme when objects/features are not present.

[0013] While the H.264 standard allows a codec to provide better quality video at lower file sizes than previous standards, such as MPEG-2 and MPEG-4 ASP (advanced simple profile), "conventional" compression codecs implementing the H.264 standard typically have struggled to keep up with the demand for greater video quality and resolution on memory-constrained devices, such as smartphones and other mobile devices, operating on limited-bandwidth networks. Video quality and resolution are often compromised to achieve adequate playback on these devices. Further, as video resolution increases, file sizes increase, making storage of videos on and off these devices a potential concern.

SUMMARY OF THE INVENTION

[0014] The present invention recognizes fundamental limitations in the inter-prediction process of conventional codecs and applies higher-level modeling to overcome those limitations and provide improved inter-prediction, while maintaining the same general processing flow and framework as conventional encoders.

[0015] In the present invention, higher-level modeling provides an efficient way of navigating more of the prediction search space (the video datacube) to produce better predictions than can be found through conventional block-based motion estimation and compensation. First, computer-vision-based feature and object detection algorithms identify regions of interest throughout the video datacube. The detection algorithm may be from the class of nonparametric feature detection algorithms. Next, the detected features and objects are modeled with a compact set of parameters, and similar feature/object instances are associated across frames. The invention then forms tracks out of the associated feature/objects, relates the tracks to specific blocks of video data to be encoded, and uses the tracking information to produce model-based predictions for those blocks of data.

[0016] In embodiments, the specific blocks of data to be encoded may be macroblocks. The formed tracks relate features to respective macroblocks.

[0017] Feature/object tracking provides additional context to the conventional encoding/decoding process. Additionally, the modeling of features/objects with a compact set of parameters enables information about the features/objects to be stored efficiently in memory, unlike reference frames, whose totality of pels are expensive to store. Thus, feature/object models can be used to search more of the video datacube, without requiring a prohibitive amount of additional computations or memory. The resulting model-based predictions are superior to conventional inter-predictions, because the model-based predictions are derived from more of the prediction search space.

[0018] In some embodiments, the compact set of parameters includes information about the features/objects and this set is stored in memory. For a feature, the respective parameters include a feature descriptor vector and a location of the feature. The respective parameters are generated when the respective feature is detected.

[0019] The model-based compression framework (MBCF) of the present invention avoids the segmentation problem encountered by previous model-based schemes. While the MBCF of the present invention also detects and tracks features/objects to identify important regions of the video frame to encode, it does not attempt to encode those feaures/objects explicitly. Rather, the features/objects

are related to nearby macroblocks, and it is the macroblocks that are encoded, as in "conventional" codecs. This implicit use of modeling information mitigates the segmentation problem in two ways: it keeps the sizes of the coding units (macroblocks) fixed (thus avoiding the need to encode object/feature shapes), and it lessens the impact of inaccurate tracking (since the tracking aids but does not dictate the motion estimation step). Additionally, the MBCF of the present invention applies modeling to video data at multiple fidelities, including a fallback option to conventional compression when features/objects are not present; this hybrid encoding scheme ensures that modeling information will only be used where needed and not incorrectly applied where it is not.

[0020] In an alternative embodiment, the MBCF may be modified so that the resulting bitstream of the encoder is H.264-compliant, meaning that the bitstream can be interpreted (decoded) by any standard H.264 decoder. The modifications in this standards-compliant MBCF (SC-MBCF) mostly involve simplification of processing options to fit entirely with the signal processing architecture of H.264. The most important of the modifications is the encoding of model-based motion vectors directly into the H.264-compliant bitstream, which incorporates modeling information in a way that is standards-compliant.

[0021] In further embodiments, the MBCF may be modified so that the resulting bitstream is compliant with any standard codec – including MPEG-2 and HEVC (H.265) – that employs block-based motion estimation followed by transform, quantization, and entropy encoding of residual signals. The steps to make the resulting bitstream compliant will vary depending on the standard codec, but the most important step will always be the encoding of model-based motion vectors directly into the compliant bitstream.

[0022] Computer-based methods, codecs and other computer systems and apparatus for processing video data may embody the foregoing principles of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0023] The foregoing will be apparent from the following more particular description of example embodiments of the invention, as illustrated in the

accompanying drawings in which like reference characters refer to the same parts throughout the different views. The drawings are not necessarily to scale, emphasis instead being placed upon illustrating embodiments of the present invention.

[0024] FIG. 1A is a block diagram depicting feature modeling according to an embodiment of the invention.

[0025] FIG. 1B is a block diagram depicting feature tracking according to an embodiment of the invention.

[0026] FIG. 1C is a block diagram illustrating the steps in relating features to nearby macroblocks and using the tracks of those features to generate good predictions for those macroblocks, according to an embodiment of the invention.

[0027] FIG. 2A is a schematic diagram illustrating the modeling of data at multiple fidelities to provide efficient encodings, according to an embodiment of the invention.

[0028] FIG. 2B is a block diagram illustrating the identification of objects through feature model correlation and aggregation, according to an embodiment of the invention.

[0029] FIG. 2C is a block diagram illustrating the identification of objects via aggregation of both nearby features and nearby macroblocks, according to an embodiment of the invention.

[0030] FIG. 3A is a schematic diagram of the configuration of an example transform- based codec according to an embodiment of the invention.

[0031] FIG. 3B is a block diagram of an example decoder for intra-predicted macroblocks, according to an embodiment of the invention.

[0032] FIG. 3C is a block diagram of an example decoder for inter-predicted macroblocks, according to an embodiment of the invention.

[0033] FIG. 3D is a schematic diagram of the configuration of an example transform based codec employing feature-based prediction, according to an embodiment of the invention.

[0034] FIG. 4 is a block diagram of an example decoder within a feature-based prediction framework, according to an embodiment of the invention.

[0035] FIG. 5 is a block diagram illustrating the state isolation process of feature instances according to an embodiment of the present information.

[0036] FIG. 6 is a block diagram illustrating an overview of example cache architecture according to an embodiment of the invention.

[0037] FIG. 7A is a block diagram illustrating the processing involved in utilizing the local (short) cache data, according to an embodiment of the invention.

[0038] FIG. 7B is a block diagram illustrating the processing involved in utilizing the distant cache data, according to an embodiment of the invention.

[0039] FIG. 8A is a schematic diagram of a computer network environment in which embodiments are deployed.

[0040] FIG. 8B is a block diagram of the computer nodes in the network of FIG. 8A.

DETAILED DESCRIPTION

[0041] The teachings of all patents, published applications and references cited herein are incorporated by reference in their entirety. A description of example embodiments of the invention follows.

[0042] The invention can be applied to various standard encodings and coding units. In the following, unless otherwise noted, the terms "conventional" and "standard" (sometimes used together with "compression," "codecs," "encodings," or "encoders") will refer to H.264, and "macroblocks" will be referred to without loss of generality as the basic H.264 coding unit.

Feature-Based Modeling

Definition of features

[0043] Example elements of the invention may include video compression and decompression processes that can optimally represent digital video data when stored or transmitted. The processes may include or interface with a video compression/encoding algorithm(s) to exploit redundancies and irrelevancies in the video data, whether spatial, temporal, or spectral. This exploitation may be done through the use and retention of feature-based models/parameters. Moving forward, the terms "feature" and "object" are used interchangeably. Objects can be defined,

without loss of generality, as "large features." Both features and objects can be used to model the data.

[0044] Features are groups of pels in close proximity that exhibit data complexity. Data complexity can be detected via various criteria, as detailed below, but the ultimate characteristic of data complexity from a compression standpoint is "costly encoding," an indication that an encoding of the pels by conventional video compression exceeds a threshold that would be considered "efficient encoding." When conventional encoders allocate a disproportionate amount of bandwidth to certain regions (because conventional inter-frame search cannot find good matches for them within conventional reference frames), it becomes more likely that the region is "feature-rich" and that a feature model-based compression method will improve compression significantly in those regions.

Feature detection

[0045] FIG. 1A depicts a feature whose instances 10-1, 10-2,…,10-n have been detected in one or more frames of the video 20-1, 20-2,…,20-n. Typically, such a feature can be detected using several criteria based on both structural information derived from the pels and complexity criteria indicating that conventional compression utilizes a disproportionate amount of bandwidth to encode the feature region. Each feature instance can be further identified spatially in its frame 20-1, 20-2, …,20-n by a corresponding spatial extent or perimeter, shown in FIG. 1A as "regions" 30-1, 30-2, …, 30-n. These feature regions 30-1, 30-2, …, 30-n can be extracted, for instance, as simple rectangular regions of pel data. In one embodiment in the current invention, the feature regions are of size 16x16, the same size as H.264 macroblocks.

[0046] Many algorithms have been proposed in the literature for detecting features based on the structure of the pels themselves, including a class of nonparametric feature detection algorithms that are robust to different transformations of the pel data. For example, the scale invariant feature transform (SIFT) [Lowe, David, 2004, "Distinctive image features from scale-invariant keypoints," *Int. J. of Computer Vision*, 60(2):91-110] uses a convolution of a difference-of-Gaussian function with the image to detect blob-like features. The

speeded-up robust features (SURF) algorithm [Bay, Herbert et al., 2008, "SURF: Speeded up robust features," *Computer Vision and Image Understanding*, 110(3):346-359] uses the determinant of the Hessian operator, also to detect blob-like features. In one embodiment of the present invention, the SURF algorithm is used to detect features.

[0047] Other feature detection algorithms are designed to find specific types of features, such as faces. In another embodiment of the present invention, the Haar-like features are detected as part of frontal and profile face detectors [Viola, Paul and Jones, Michael, 2001, "Rapid object detection using a boosted cascade of simple features," *Proc. of the 2001 IEEE Conf. on Computer Vision and Pattern Recognition*, 1:511-518].

[0048] In another embodiment, discussed in full in U.S. Application No., 13/121,904, filed October 6, 2009, which is incorporated herein by reference in its entirety, features can be detected based on encoding complexity (bandwidth) encountered by a conventional encoder. Encoding complexity, for example, can be determined through analysis of the bandwidth (number of bits) required by conventional compression (e.g., H.264) to encode the regions in which features appear. Restated, different detection algorithms operate differently, but each are applied to the entire video sequence of frames over the entire video data in embodiments. For a non-limiting example, a first encoding pass with an H.264 encoder is made and creates a "bandwidth map." This in turn defines or otherwise determines where in each frame H.264 encoding costs are the highest.

[0049] Typically, conventional encoders such as H.264 partition video frames into uniform tiles (for example, 16x16 macroblocks and their subtiles) arranged in a non-overlapping pattern. In one embodiment, each tile can be analyzed as a potential feature, based on the relative bandwidth required by H.264 to encode the tile. For example, the bandwidth required to encode a tile via H.264 may be compared to a fixed threshold, and the tile can be declared a "feature" if the bandwidth exceeds the threshold. The threshold may be a preset value. The preset value may be stored in a database for easy access during feature detection. The threshold may be a value set as the average bandwidth amount allocated for previously encoded features . Likewise, the threshold may be a value set as the

median bandwidth amount allocated for previously encoded features. Alternatively, one could calculate cumulative distribution functions of the tile bandwidths across an entire frame (or an entire video) and declare as "features" any tile whose bandwidth is in the top percentiles of all tile bandwidths.

[0050] In another embodiment, video frames can be partitioned into overlapping tiles. The overlapping sampling may be offset so that the centers of the overlapping tiles occur at the intersection of every four underlying tiles' corners. This over-complete partitioning is meant to increase the likelihood that an initial sampling position will yield a detected feature. Other, possibly more complex, topological partitioning methods are also possible.

[0051] Small spatial regions detected as features can be analyzed to determine if they can be combined based on some coherency criteria into larger spatial regions. Spatial regions can vary in size from small groups of pels to larger areas that may correspond to actual objects or parts of objects. However, it is important to note that the detected features need not correspond to unique and separable entities such as objects and sub-objects. A single feature may contain elements of two or more objects or no object elements at all. For the current invention, the critical characteristic of a feature is that the set of pels comprising the feature can be efficiently compressed, relative to conventional methods, by feature model-based compression techniques.

[0052] Coherency criteria for combining small regions into larger regions may include: similarity of motion, similarity of appearance after motion compensation, and similarity of encoding complexity. Coherent motion may be discovered through higher-order motion models. In one embodiment, the translational motion for each individual small region can be integrated into an affine motion model that is able to approximate the motion model for each of the small regions. If the motion for a set of small regions can be integrated into aggregate models on a consistent basis, this implies a dependency among the regions that may indicate a coherency among the small regions that could be exploited through an aggregate feature model.

Feature model formation

[0053] After features have been detected in multiple frames of a video, it is important that multiple instances of the same feature be related together. This process is known as *feature association* and is the basis for feature tracking (determining the location of a particular feature over time), described below. To be effective, however, the feature association process must first define a *feature model* that can be used to discriminate similar feature instances from dissimilar ones.

[0054] In one embodiment, the feature pels themselves can be used to model a feature. Feature pel regions, which are two-dimensional, can be vectorized and similar features can be identified by minimizing mean-squared error (MSE) or maximizing inner products between different feature pel vectors. The problem with this is that feature pel vectors are sensitive to small changes in the feature, such as translation, rotation, scaling, and changing illumination of the feature. Features often change in these ways throughout a video, so using the feature pel vectors themselves to model and associate features requires some accounting for these changes. In one embodiment, the invention accounts for such feature changes in the simplest way, by applying standard motion estimation and compensation algorithms found in conventional codecs (e.g., H.264), which account for translational motion of features. In other embodiments, more complex techniques can be used to account for rotations, scalings, and illumination changes of features from frame to frame.

[0055] In an alternate embodiment, feature models are compact representations of the features themselves ("compact" meaning "of lower dimension than the original feature pels vectors") that are *invariant* (remain unchanged when transformations of a certain type are applied) to small rotations, translations, scalings, and possibly illumination changes of the feature – meaning that if the feature changes slightly from frame to frame, the feature model will remain relatively constant. A compact feature model of this type is often termed a "descriptor." In one embodiment of the current invention, for example, the SURF feature descriptor has length 64 (compared to the length-256 feature pel vectors) and is based on sums of Haar wavelet transform responses. In another embodiment, a color histogram with 5 bins is constructed from a colormap of the feature pels, and this 5-component histogram acts as the feature descriptor. In an alternate

embodiment, feature regions are transformed via 2-D DCT. The 2-D DCT coefficients are then summed over the upper triangular and lower triangular portions of the coefficient matrix. These sums then comprise an edge feature space and act as the feature descriptor.

[0056] When feature descriptors are used to model features, similar features can be identified by minimizing MSE or maximizing inner products between the feature descriptors (instead of between the feature pel vectors).

Feature association and tracking

[0057] Once features have been detected and modeled, the next step is to associate similar features over multiple frames. Each instance of a feature that appears in multiple frames is a sample of the appearance of that feature, and multiple feature instances that are associated across frames are considered to "belong" to the same feature. Once associated, multiple feature instances belonging to the same feature may be aggregated to form a feature track.

[0058] A *feature track* is defined as the (x,y) location of a feature as a function of frames in the video. One embodiment associates newly detected feature instances with previously tracked features (or, in the case of the first frame of the video, with previously detected features) as the basis for determining which features instances in the current frame are extensions of which previously-established feature tracks. The identification of a feature's instance in the current frame with a previously established feature track (or, in the case of the first video frame, with a previously detected feature) constitutes the tracking of the feature.

[0059] FIG. 1B demonstrates the use of a feature tracker 70 to track features 60-1, 60-2, …, 60-n. A feature detector 80 (for example, SIFT or SURF) is used to identify features in the current frame. Detected feature instances in the current frame 90 are matched to previously detected (or tracked) features 50. In one embodiment, prior to the association step, the set of candidate feature detections in the current frame can be sorted using an auto-correlation analysis (ACA) metric that measures feature strength based on an autocorrelation matrix of the feature, using derivative-of-Gaussian filters to compute the image gradients in the autocorrelation matrix, as found in the Harris-Stephens corner detection algorithm [Harris, Chris

and Mike Stephens, 1988, "A combined corner and edge detector," in *Proc. of the 4th Alvey Vision Conference*, pp. 147-151]. Feature instances with high ACA values are given priority as candidates for track extension. In one embodiment, feature instances lower in the ACA-sorted list are pruned from the set of candidate features if they are within a certain distance (e.g., one pel) of a feature instance higher in the list.

[0060] In different embodiments, feature descriptors (e.g., the SURF descriptor) or the feature pel vectors themselves may serve as the feature models. In one embodiment, previously-tracked features, depicted as regions 60-1, 60-2, …, 60-n in FIG. 1B, are tested one at a time for track extensions from among the newly detected features in the current frame 90. In one embodiment, the most recent feature instance for each feature track serves as a focal point (or "target feature") in the search for a track extension in the current frame. All candidate feature detections in the current frame within a certain distance (e.g., 16 pels) of the location of the target feature are tested, and the candidate having minimum MSE with the target feature is chosen as the extension of that feature track. In another embodiment, a candidate feature is disqualified from being a track extension if its MSE with the target feature is larger than some threshold.

[0061] In a further embodiment, if no candidate feature detection in the current frame qualifies for extension of a given feature track, a limited search for a matching region in the current frame is conducted using either the motion compensated prediction (MCP) algorithm within H.264 or a generic motion estimation and compensation (MEC) algorithm. Both MCP and MEC conduct a gradient descent search for a matching region in the current frame that minimizes MSE (and satisfies the MSE threshold) with respect to the target feature in the previous frame. If no matches can be found for the target feature in the current frame, either from the candidate feature detection or from the MCP/MEC search process, the corresponding feature track is declared "dead" or "terminated."

[0062] In a further embodiment, if two or more feature tracks have feature instances in the current frame that coincide by more than some threshold (for example, 70% overlap), all but one of the feature tracks are pruned, or dropped from

further consideration. The pruning process keeps the feature track that has the longest history and has the largest total ACA, summed over all feature instances.

[0063] The following combination of the above steps is henceforth referred to as the feature point analysis (FPA) tracker and serves as an embodiment of the invention: SURF feature detection, feature modeling (using SURF descriptors), ACA-based sorting of candidate features, and feature association and tracking via minimization of MSE from among candidate features, supplemented by MCP/MEC searching for track extensions.

[0064] In another embodiment of the invention, macroblocks in the video frame are thought of as features, registration of the features/macroblocks is done through the MCP engine found in H.264, and feature/macroblocks are associated using the inter-frame prediction metrics (such as sum of absolute transform differences [SATD]) of H.264; this combination is termed the macroblock cache (MBC) tracker. The MBC tracker is differentiated from standard inter-frame prediction because certain parameters are different (for example, search boundaries are disabled, so that the MBC tracker conducts a wider search for matches) and because certain aspects of the matching process are different. In a third embodiment, SURF detections are related to nearby macroblocks, and the macroblocks are associated and tracked using the MCP and inter-frame prediction engines of H.264; this combination is termed the SURF tracker.

Feature-Based Compression

[0065] Feature modeling (or data modeling in general) can be used to improve compression over standard codecs. Standard inter-frame prediction uses block-based motion estimation and compensation to find predictions for each coding unit (macroblock) from a limited search space in previously decoded reference frames. Exhaustive search for good predictions throughout all past reference frames is computationally prohibitive. By detecting and tracking features throughout the video, feature modeling provides a way of navigating the prediction search space to produce improved predictions without prohibitive computations. In the following, the terms “feature-based” and “model-based” are used interchangeably, as features are a specific type of model.

[0066] In one embodiment of the invention, feature tracks are used to relate features to macroblocks. The general steps for this are depicted in FIG. 1C. A given feature track indicates the location of a feature across frames, and there is an associated motion of that feature across frames. Using the location of the feature in the two most recent frames prior to the current frame, one can project the position of the feature in the current frame. This projected feature position then has an associated nearest macroblock, defined as the macroblock having greatest overlap with the projected feature position. This macroblock (now the target macroblock that is being encoded) has been associated to a specific feature track whose projected position in the current frame is nearby the macroblock (100 in FIG. 1C).

[0067] The next step is to calculate an offset 110 between the target macroblock and the projected feature position in the current frame. This offset can then be used to generate predictions for the target macroblock, using earlier feature instances in the associated feature's track. These earlier feature instances occupy either a local cache 120, comprised of recent reference frames where the feature appeared, or a distant cache 140, comprised of "older" reference frames 150 where the feature appeared. Predictions for the target macroblock can be generated by finding the regions in the reference frames with the same offsets (130, 160) from earlier feature instances as the offset between the target macroblock and the projected feature position in the current frame.

Generating model-based primary and secondary predictions

[0068] In one embodiment of the present invention, feature-based prediction is implemented as follows: (1) detect the features for each frame; (2) model the detected features; (3) associate features in different frames to create feature tracks; (4) use feature tracks to predict feature locations in the "current" frame being encoded; (5) associate macroblocks in the current frame that are nearby the predicted feature locations; (6) generate predictions for the macroblocks in Step 5 based on past locations along the feature tracks of their associated features.

[0069] In one embodiment, features are detected using the SURF algorithm and they are associated and tracked using the FPA algorithm, as detailed in the previous section. Once features have been detected, associated, and tracked, the feature

tracks can be used to associate each feature track with a nearest macroblock, as detailed above. It is possible for a single macroblock to be associated with multiple features, so one embodiment selects the feature having maximum overlap with the macroblock as the associated feature for that macroblock.

[0070] Given a target macroblock (the current macroblock being encoded), its associated feature, and the feature track for that feature, a *primary* prediction for the target macroblock can be generated. Data pels for the primary prediction comes from the most recent frame (prior to the current frame) where the feature appears, henceforth referred to as the *key frame*. The primary prediction is generated after selecting a motion model and a pel sampling scheme. In one embodiment of the present invention, the motion model can be either "0th order," which assumes that the feature is stationary between the key frame and the current frame, or "1st order," which assumes that feature motion is linear between the 2nd-most recent reference frame, the key frame, and the current frame. In either case, the motion of the feature is applied (in the backwards temporal direction) to the associated macroblock in the current frame to obtain the prediction for the macroblock in the key frame. In one embodiment of the present invention, the pel sampling scheme can be either "direct," in which motion vectors are rounded to the nearest integer and pels for the primary prediction are taken directly from the key frame, or "indirect," in which the interpolation scheme from conventional compression such as H.264 is used to derive a motion-compensated primary prediction. Thus, the present invention can have four different types of primary prediction, depending on the motion model (0^{th} or 1^{st} order) and the sampling scheme (direct or indirect).

[0071] In an alternative embodiment, data pels for the primary prediction do not have to come from the key frame (the most recent frame prior to the current frame where the feature occurs) but can be taken from any previous reference frame stored in the reference frame buffer. In this case, the primary prediction can still be calculated via 0^{th} or 1^{st} order motion models and through direct or indirect sampling schemes. In the case of the 1^{st} order motion model, linear motion is assumed between the current frame, the key frame, and the past reference frame.

[0072] Primary prediction can be refined by modeling local deformations through the process of subtiling. In the subtiling process, different motion vectors are

calculated for different local regions of the macroblock. In one embodiment, subtiling can be done by dividing the 16×16 macroblock into two 8×16 regions, two 16×8 regions, four 8×8 quadrants, or even smaller partitions (4×8, 8×4, 4×4), and calculating motion vectors for each local region separately. In another embodiment, subtiling can be carried out in the Y/U/V color space domain by calculating predictions for the Y, U, and V color channels (or various partitions of them) separately.

[0073] In addition to the primary prediction for the target macroblock, one can also generate secondary predictions based on positions of the associated feature in reference frames prior to the key frame. In one embodiment, the offset from the target macroblock to the (projected) position of the associated feature in the current frame represents a motion vector that can be used to find secondary predictions from the feature's position in past reference frames. In this way, a large number of secondary predictions can be generated (one for each frame where the feature has appeared previously) for a given target macroblock that has an associated feature. In one embodiment, the number of secondary predictions can be limited by restricting the search to some reasonable number of past reference frames (for example, 25).

Composite predictions

[0074] Once primary and secondary predictions have been generated for a target macroblock, the overall reconstruction of the target macroblock can be computed based on these predictions. In one embodiment, following conventional codecs, the reconstruction is based on the primary prediction only, henceforth referred to as primary-only (PO) reconstruction.

[0075] In another embodiment, the reconstruction is based on a composite prediction that sums the key prediction and a weighted version of one of the secondary predictions. This algorithm, henceforth referred to as PCA-Lite (PCA-L), involves the following steps:

[0076] 1. Create the vectorized (1-D) versions of the target macroblock and primary prediction. These can then be denoted as the target vector t and primary vector p.

[0077] 2. Subtract the primary vector from the target vector to compute a residual vector r.

[0078] 3. Vectorize the set of secondary predictions to form vectors s_i (Without loss of generality, assume that these secondary vectors have unit norm.) Then subtract the primary vector from all the secondary vectors to form the primary-subtracted set, s_i-p. This has the approximate effect of projecting off the primary vector from the secondary vectors.

[0079] 4. For each secondary vector, calculate a weighting $c = r^T (s_i - p)$.

[0080] 5. For each secondary vector, calculate the composite prediction as $\hat{t} = p + c \cdot (s_i - p)$.

[0081] In general, the steps in the PCA-Lite algorithm approximate the operations in the well-known orthogonal matching pursuit algorithm [Pati, 1993], with the composite prediction meant to have non-redundant contributions from the primary and secondary predictions. In another embodiment, the PCA-Lite algorithm described above is modified so that the primary vector in Steps 3-5 above is replaced by the mean of the primary and the secondary vector. This modified algorithm is henceforth referred to as PCA-Lite-Mean.

[0082] The PCA-Lite algorithm provides a different type of composite prediction than the bi-prediction algorithms found in some standard codecs (and described in the "Background" section above). Standard bi-prediction algorithms employ a blending of multiple predictions based on temporal distance of the reference frames for the individual predictions to the current frame. By contrast, PCA-Lite blends multiple predictions into a composite prediction based on the *contents* of the individual predictions.

[0083] In another embodiment, the coefficients for the PCA-Lite algorithm can be computed over subtiles of a macroblock instead of over the entire macroblock. The benefit of this is similar to the benefit described above for calculating motion vectors over subtiles of the macroblock: calculating "local" coefficients over a subtile is potentially more "accurate" than calculating "global" coefficients over an entire macroblock. To perform the PCA-Lite coefficient calculation in subtile space, the target vector t, primary vector p, and secondary vectors s_i are divided into subtiles (either region-based partitions such as 16×8, 8×16, 8×8, and smaller

regions; or color-based partitions such as Y/U/V color channels) and Steps 1-5 above are repeated for each subtile. Thus, a larger number of coefficients are calculated (one for each subtile) and needed to be encoded; this is a tradeoff for the higher accuracy produced by the local coefficient calculation.

[0084] Note that the formation of composite predictions as described above does not require feature-based modeling; composite predictions can be formed from any set of multiple predictions for a given target macroblock. Feature-based modeling, however, provides a naturally-associated set of multiple predictions for a given target macroblock, and composite predictions provide an efficient way to combine the information from those multiple predictions.

Multiple Fidelity Data Modeling

[0085] The current invention provides the ability to model the data at multiple fidelities for the purpose of model-based compression. One embodiment of this is illustrated in FIG. 2A, which displays four levels of modeling. These four levels are summarized in the following table and discussed in more detail below.

	Size	Grid-Aligned	Can Span Multiple MBs	H.264 Motion Vector Predictors
Macroblocks	16x16	Yes	No	Yes
Macroblocks as Features	16x16	Yes	No	Yes
Features	16x16	No	Yes	Sometimes
Objects	Up to Frame Size	No	Yes	No

[0086] The bottom level 200 in FIG. 2A is termed the "Macroblock" (MB) level and represents conventional compression partitioning frames into non-overlapping macroblocks, tiles of size 16x16, or a limited set of subtiles. Conventional compression (e.g., H.264) essentially employs no modeling; instead, it uses block-based motion estimation and compensation (BBMEC) to find predictions 212 for

each tile from a limited search space in previously decoded reference frames. At the decoder, the predictions 212 are combined with residual encodings of the macroblocks (or subtiles) to synthesize 210 a reconstruction of the original data.

[0087] The second level 202 in FIG. 2A is termed the "Macroblocks as Features" (MBF) level and represents compression based on the MBC tracker described above and represented at 216 in Fig 2A. Here, macroblocks (or subtiles of macroblocks) are treated as features, through recursive application of conventional BBMEC searches through previously encoded frames. The first application of BBMEC is identical to that of the MB level, finding a conventional prediction for the target macroblock from the most recent reference frame in 216. The second application of BBMEC, however, finds a conventional prediction for the first prediction by searching in the second-most-recent frame in 216. Repeated application of BBMEC through progressively older frames in 216 creates a "track" for the target macroblock, even though the latter has not been identified as a feature per se. The MBC track produces a model 214 that generates a prediction 212 that is combined with residual encodings of the macroblocks (or subtiles) to synthesize 210 a reconstruction of the original data at the decoder.

[0088] The third level 204 in FIG. 2A is termed the "Features" level and represents feature-based compression as described above. To review, features are detected and tracked independent of the macroblock grid, but features are associated with overlapping macroblocks and feature tracks are used to navigate previously-decoded reference frames 216 to find better matches for those overlapping macroblocks. If multiple features overlap a given target macroblock, the feature with greatest overlap is selected to model that target macroblock at 214. In an alternate embodiment, the codec could encode and decode the features directly, without relating the features to macroblocks, and process the "non-feature" background separately using, for example, MB-level conventional compression. The feature-based model 214 generates a prediction 212 that is combined with residual encodings of the associated macroblocks (or subtiles) to synthesize 210 a reconstruction of the original data at the decoder.

[0089] The top level 206 in FIG. 2A is termed the "Objects" level and represents object-based compression. Objects are essentially large features that may encompass

multiple macroblocks and may represent something that has physical meaning (e.g., a face, a ball, or a cellphone) or complex phenomena 208. Object modeling is often parametric, where it is anticipated that an object will be of a certain type (e.g., a face), so that specialized basis functions can be used for the modeling 214. When objects encompass or overlap multiple macroblocks, a single motion vector 212 can be calculated for all of the macroblocks associated with the object 216, which can result in savings both in terms of computations and encoding size. The object-based model 214 generates a prediction 212 that is combined with residual encodings of the associated macroblocks (or subtiles) to synthesize 210 a reconstruction of the original data at the decoder.

[0090] In an alternate embodiment, objects may also be identified by correlating and aggregating nearby feature models 214. FIG. 2B is a block diagram illustrating this type of nonparametric or empirical object detection via feature model aggregation. A particular type of object 220 is detected by identifying which features have characteristics of that object type, or display "object bias" 222. Then, it is determined whether the set of features in 222 display a rigidity of the model states 224, a tendency over time for the features and their states to be correlated. If the individual feature models are determined to be correlated (in which case an object detection is determined 226), then a composite appearance model with accompanying parameters 228 and a composite deformation model with accompanying parameters 230 can be formed. The formation of composite appearance and deformation models evokes a natural parameter reduction 232 from the collective individual appearance and deformation models.

[0091] FIG. 2C illustrates a third embodiment of the "Objects" level 206 in FIG. 2A, employing both parametric and nonparametric object-based modeling. A parametrically modeled object is detected 240. The detected object 240 may be processed to determine if there are any overlapping features 250. The set of overlapping features may then be tested 260 to determine whether they can be aggregated as above. If aggregation of the overlapping features fails, then the process reverts to testing the macroblocks overlapping the detected object 240, to determine whether they can be effectively aggregated 270 to share a common motion vector, as noted above.

[0092] A multiple-fidelity processing architecture may use any combination of levels 200, 202, 204, 206 to achieve the most advantageous processing. In one embodiment, all levels in FIG. 2A are examined in a "competition" to determine which levels produce the best (smallest) encodings for each macroblock to be encoded. More details on how this "competition" is conducted follow below.

[0093] In another embodiment, the levels in FIG. 2A could be examined sequentially, from bottom (simplest) to top (most complex). If a lower-level solution is deemed satisfactory, higher-level solutions do not have to be examined. Metrics for determining whether a given solution can be deemed "good enough" are described in more detail below.

Model-Based Compression Codec

Standard codec processing

[0094] The encoding process may convert video data into a compressed, or encoded, format. Likewise, the decompression process, or decoding process, may convert compressed video back into an uncompressed, or raw, format. The video compression and decompression processes may be implemented as an encoder/decoder pair commonly referred to as a codec.

[0095] FIG. 3A is a block diagram of a standard encoder 312. The encoder in FIG. 3A may be implemented in a software or hardware environment, or combination thereof. Components of the example encoder may be implemented as executable code stored on a storage medium, such as one of those shown in FIGs. 8A and 8B, and configured for execution by one or more of processors 820. The encoder 312 may include any combination of components, including, but not limited to, an intra-prediction module 314, an inter-prediction module 316, a transform module 324, a quantization module 326, an entropy encoding module 328 and a loop filter 334. The inter prediction module 316 may include a motion compensation module 318, frame storage module 320, and motion estimation module 322. The encoder 312 may further include an inverse quantization module 330, and an inverse transform module 332. The function of each of the components of the encoder 312 shown in FIG. 3A is well known to one of ordinary skill in the art.

[0096] The entropy coding algorithm 328 in FIG. 3A may be based on a probability distribution that measures the likelihood of different values of quantized transform coefficients. The encoding size of the current coding unit (e.g., macroblock) depends on the current encoding state (values of different quantities to be encoded) and the relative conformance of the state to the probability distribution. Any changes to this encoding state, as detailed below, may impact encoding sizes of coding units in subsequent frames. To fully optimize an encoding of a video, an exhaustive search may be conducted of all the possible paths on which the video can be encoded (i.e., all possible encoding states), but this is computationally prohibitive. In one embodiment of the current invention, the encoder 312 is configured to focus on the current (target) macroblock, so that optimization is applied locally, rather than considering a larger scope, (e.g., over a slice, a frame, or a set of frames).

[0097] FIGs. 3B and 3C are block diagrams of a standard decoder 340 providing decoding of intra-predicted data 336 and decoding of inter-predicted data 338, respectively. The decoder 340 may be implemented in a software or hardware environment, or combination thereof. Referring to FIGs. 3A, 3B, and 3C, the encoder 312 typically receives the video input 310 from an internal or external source, encodes the data, and stores the encoded data in the decoder cache/buffer 348. The decoder 340 retrieves the encoded data from the cache/buffer 348 for decoding and transmission. The decoder may obtain access to the decoded data from any available means, such as a system bus or network interface. The decoder 340 can be configured to decode the video data to decompress the predicted frames and key frames (generally at 210 in FIG. 2A). The cache/buffer 348 can receive the data related to the compressed video sequence/bitstream and make information available to the entropy decoder 346. The entropy decoder 346 processes the bitstream to generate estimates of quantized transform coefficients for the intra-prediction in FIG. 3A or the residual signal in FIG. 3B. The inverse quantizer 344 performs a rescaling operation to produce estimated transform coefficients, and the inverse transform 342 is then applied to the estimated transform coefficients to create a synthesis of the intra-prediction of the original video data pels in FIG. 3A or of the residual signal in FIG. 3B. In FIG. 3B, the synthesized residual signal is

added back to the inter-prediction of the target macroblock to generate the full reconstruction of the target macroblock. The inter-prediction module 350 replicates at the decoder the inter-prediction generated by the encoder, making use of motion estimation 356 and motion compensation 354 applied to reference frames contained in the framestore 352. The decoder's inter-prediction module 350 mirrors the encoder's inter-prediction module 316 in FIG. 3A, with its components of motion estimation 322, motion compensation 318, and framestore 320.

Hybrid codec implementing model-based prediction

[0098] FIG. 3D is a diagram of an example encoder according to an embodiment of the invention that implements model-based prediction, the framework for which is henceforth referred to as a model-based compression framework (MBCF). At 362, the MBCF encoder 360 can be configured to encode a current (target) frame. At 364, each macroblock in the frame can be encoded, such that, at 366, a standard H.264 encoding process is used to define a base (first) encoding that yields an H.264 encoding solution. In one preferred embodiment, the encoder 366 is an H.264 encoder capable of encoding a Group of Pictures (set of reference frames). Further, the H.264 encoder preferably is configurable so that it can apply different methods to encode pels within each frame, i.e., intra-frame and inter-frame prediction, with inter-frame prediction able to search multiple reference frames for good matches for the macroblock being encoded. Preferably, the error between the original macroblock data and the prediction is transformed, quantized, and entropy-encoded.

[0099] Preferably, the encoder 360 utilizes the CABAC entropy encoding algorithm at 382 to provide a context-sensitive, adaptive mechanism for context modeling. The context modeling may be applied to a binarized sequence of the syntactical elements of the video data such as block types, motion vectors, and quantized coefficients, with the binarization process using predefined mechanisms. Each element is then coded using either adaptive or fixed probability models. Context values can be used for appropriate adaptations of the probability models.

[00100] While standard H.264 encoders encode motion vectors differentially with respect to neighboring, previously-decoded motion vectors, the MBCF encodes motion vectors differentially with respect to a "global" motion vector derived from

the tracker (whether FPA, MBC, SURF or other tracker known in the art?). One of the benefits of running a tracker is that this global motion vector is available as a by-product.

Competition Mode

[00101] In Fig. 3D, at 368, the H.264 macroblock encoding is analyzed. At 368, if the H.264 encoding of the macroblock is judged to be "efficient," then the H.264 solution is deemed to be close to ideal, no further analysis is performed, and the H.264 encoding solution is accepted for the target macroblock. In one embodiment, efficiency of the H.264 encoding can be judged by comparing the H.264 encoding size (in bits) to a threshold, which can be derived from percentile statistics from previously encoded videos or from earlier in the same video. In another embodiment, efficiency of the H.264 encoding can be judged by determining whether an H.264 encoder has declared the target macroblock a "skip" macroblock, in which the data in and around the target macroblock is uniform enough that the target macroblock essentially requires no additional encoding.

[00102] At 368, if the H.264 macroblock solution is not considered efficient, then additional analysis is performed, and the encoder enters Competition Mode 380. In this mode, several different predictions are generated for the target macroblock, based on multiple models 378. The models 378 are created from the identification of features 376 detected and tracked in prior frames 374. Note that as each new frame 362 is processed (encoded and then decoded and placed into framestore), the feature models need to be updated to account for new feature detections and associated feature track extensions in the new frame 362. The model-based solutions 382 are ranked based on their encoding sizes 384, along with the H.264 solution acquired previously. Because of its flexibility to encode a given macroblock using either a base encoding (the H.264 solution) or a model-based encoding, the present invention is termed a hybrid codec.

[00103] For example, in Competition Mode, an H.264 encoding is generated for the target macroblock to compare its compression efficiency (ability to encode data with a small number of bits) relative to other modes. Then for each encoding algorithm used in Competition Mode, the following steps are executed: (1) generate

a prediction based on the codec mode/algorithm used; (2) subtract the prediction from the target macroblock to generate a residual signal; (3) transform the residual (target minus prediction) using an approximation of a 2-D block-based DCT; (4) encode the transform coefficients using an entropy encoder.

[00104] In some respects, the baseline H.264 (inter-frame) prediction can be thought of as based on a relatively simple, limited model (H.264 is one of the algorithms used in Competition Mode). However, the predictions of the encoder 360 can be based on more complex models, which are either feature-based or object-based, and the corresponding tracking of those models. If a macroblock exhibiting data complexity is detected, the encoder 360 operates under the assumption that feature-based compression can do a better job than conventional compression.

Use of feature-based predictions in Competition Mode

[00105] As noted above, for each target macroblock, the MBCF encoder makes an initial determination as to whether the H.264 solution (prediction) is efficient ("good enough") for that macroblock. If the answer is negative, Competition Mode is entered.

[00106] In FIG. 3D for Competition Mode 380, the "entries" into the competition are determined by the various processing choices for feature-based prediction described above. Each entry comprises a different prediction for the target macroblock. Full description of the invention's feature-based prediction requires specification of the following processing choices:

- tracker type (FPA, MBC, SURF)
- motion model for primary prediction (0^{th} or 1^{st} order)
- sampling scheme for primary prediction (direct or indirect)
- subtiling scheme for motion vector calculation (no subtiling, local regions, color channels)
- reconstruction algorithm (PO or PCA-L)
- subtiling scheme for PCA-L coefficient calculation (no subtiling, local regions, color channels)
- reference frame for primary prediction (PO or PCA-L)
- reference frames for secondary prediction (for PCA-L).

[00107] The solution search space for a given target macroblock is comprised of all of the invention's feature-based predictions represented above, plus the H.264 solution (the "best" inter-frame prediction from H.264). In one embodiment, Competition Mode includes all possible combinations of processing choices noted above (tracker type, motion model and sampling scheme for primary prediction, subtiling scheme, and reconstruction algorithms). In another embodiment, the processing choices in Competition Mode are configurable and can be limited to a reasonable subset of possible processing combinations to save computations.

[00108] In an alternative embodiment, the MBCF may be modified so that the resulting bitstream of the encoder is H.264-compliant, meaning that the bitstream can be interpreted (decoded) by any standard H.264 decoder. In this standards-compliant MBCF (SC-MBCF), the processing options available to the Competition Mode are limited to those whose encodings can be interpreted within a standard H.264 bitstream. The available processing options in the SC-MBCF are:

- tracker type (FPA, MBC, SURF, or other known tracker)
- motion model for primary prediction (1^{st} order only)
- sampling scheme for primary prediction (direct or indirect)
- subtiling for motion vector calculation (local regions, color channels)
- reconstruction algorithm (PO only)
- reference frame for primary prediction.

[00109] In particular, standard H.264 decoders cannot interpret the additional coefficients required by the PCA-Lite algorithm variations, so the primary-only (PO) algorithm is the sole reconstruction algorithm available. For the (nonstandard) MBCF, the CABAC context for entropy encoded must be modified to accommodate the additional PCA-Lite coefficients, among other quantities; for the SC-MBCF, no such accommodation is necessary and standard H.264 CABAC context are used.

[00110] Potential solutions for the competition are evaluated one at a time by following the four steps noted previously: (1) generate the prediction; (2) subtract the prediction from the target macroblock to generate a residual signal; (3) transform the residual; (4) encode the transform coefficients using an entropy encoder. In FIG. 3D the output of the last step, 382 is a number of bits associated with a given solution 384. After each solution is evaluated, the encoder is rolled back to its state

prior to that evaluation, so that the next solution can be evaluated. In one embodiment, after all solutions have been evaluated, a "winner" for the competition is chosen 370 by selecting the one with smallest encoding size. The winning solution is then sent to the encoder once more 372 as the final encoding for the target macroblock. As noted above, this winning solution is a locally-optimum solution, as it is optimum for the target macroblock only. In an alternate embodiment, the selection of the optimal solution is hedged against larger scale encoding tradeoffs that include, but are not limited to, context intra-frame prediction feedback and residual error effects in future frames.

[00111] Information pertaining to the winning solution is saved into the encoding stream 386 and transmitted/stored for future decoding. This information may include, but is not limited to, the processing choices noted above for feature-based prediction (e.g., tracker type, primary prediction, subtiling scheme, reconstruction algorithm, etc.).

[00112] In some cases, the encoder 360 may determine that the target macroblock is not efficiently coded by H.264, but there is also no detected feature that overlaps with that macroblock. In this case, the encoder uses H.264 anyway to encode the macroblock as a last resort. In an alternate embodiment, the tracks from the feature tracker can be extended to generate a pseudo-feature that can overlap the macroblock and thus produce a feature-based prediction.

[00113] In one embodiment, movement among the four levels in Fig. 2A is governed by Competition Mode.

Decoding using feature-based predictions

[00114] FIG. 4 is a diagram of an example decoder according to an embodiment of the invention implementing model-based prediction within the Assignee's EuclidVision codec. The decoder 400 decodes the encoded video bitstream to synthesize an approximation of the input video frame that generated the frame encoding 402. The frame encoding 402 includes a set of parameters used by the decoder 400 to reconstruct its corresponding video frame 418.

[00115] The decoder 400 traverses each frame with the same slice ordering used by the encoder, and the decoder traverses each slice with the same macroblock

ordering used by the encoder. For each macroblock 404, the decoder follows the same process as the encoder, determining 406 whether to decode the macroblock conventionally 408 or whether to decode the macroblock utilizing feature models and parameters at 416. If a macroblock was encoded via the invention's model-based prediction (within its model-based compression framework [MBCF]), the decoder 400 extracts whatever feature information (feature tracks, feature reference frames [GOP], feature motion vectors) is needed to reproduce the prediction for that solution 418. The decoder updates feature models (410, 412, 414) during the decoding so they are synchronized with the encoder feature state for the particular frame/slice/macroblock that is being processed. The need to run the feature detector 410 and tracker 414 at the decoder is non-standard but necessary to re-create the tracker-based global motion vectors for differential encoding of motion vectors.

[00116] In an alternative embodiment, within the standards-compliant MBCF (SC-MBCF), feature information is not used directly to encode model-based predictions. Instead, feature information identifies particular motion vectors and corresponding regions for primary prediction, and the motion vectors are encoded directly (or differentially with respect to neighboring motion vectors, as in standard H.264 encoders) into the bitstream. In this case, the decoder 400 never needs to extract additional feature information 416 but is always able to decode the macroblock conventionally at 408. Thus, in the SC-MBCF, the decoders are standard H.264 decoders that do not run feature detection and tracking.

[00117] Note that, because of memory limitations, conventional codecs do not typically retain the entire prediction context for decoded frames in the framestore 352 and cache 348 of FIG. 3C, but only the frames (pels) themselves. By contrast, the invention extends the prediction context stored in the framestore 352 and cache 348 of FIG. 3C by prioritizing retention of feature-based models and parameters.

[00118] The full set of parameters that describe a feature model is known as the state of the feature, and this state must be isolated to retain feature models effectively. FIG. 5 is a block diagram illustrating the state isolation process 500 of feature instances according to an embodiment of the present invention. This state isolation information can be associated with a target macroblock and include parameters associated with relevant feature instances 502 that can be of assistance in

the encoding of that target macroblock. The state isolation information can be also used to interpolate predicted features in future video frames. Each respective feature instance has an associated GOP 504. Each GOP includes respective state information regarding, for example, respective boundary information. The respective state isolation information of a feature instance may further include state information about any relevant associated objects, their respective slice parameters 506, and their respective entropy state 508. In this way, the state information provides instructions regarding the boundaries of GOP/slice/entropy parameters of feature instances and their corresponding extensions into new states and state contexts. The state information 506, 508 can be used to predict and interpolate the state of a predicted feature in future frames.

[00119] Together, the macroblock data (pels) and state isolation information from associated features form an extended prediction context. Extended contexts from multiple feature instances and their previously decoded neighbors may be combined. The extended prediction context for the encoder 312 in FIG. 3A and decoder 340 in FIGs. 3B and 3C may include, but is not limited to: (1) one or more macroblocks, (2) one or more neighboring macroblocks, (3) slice information, (4) reference frames [GOP], (5) one or more feature instances, (6) object/texture information.

Cache Organization and Access of Feature Model Information

[00120] During the process of generating feature models, it is often the case that multiple instances of a specific feature are found in a given video. In this case, the feature model information can be stored or cached efficiently by organizing the model information prior to caching. This technique can be applied to both parametric and nonparametric model-based compression schemes.

[00121] In FIG. 3C, for example, if it is determined that the use of feature-based modeling prediction context information improves compression efficiency, the cache 348 (including the framestore 352) can be configured to include feature-based modeling prediction context information. Attempts to access uncached feature-based prediction context data can generate overhead that degrades the system's responsiveness and determinism. This overhead can be minimized by caching, ahead of time, the preprocessed feature-based encoding prediction context. Doing

this provides a means by which much of the repetition of accessing data related to the feature-based prediction context can be avoided.

[00122] The encoder 312/decoder 340 (FIGs. 3A, 3C) can be configured using, for example, a cache that is adapted to increase the execution speed and efficiency of video processing. The performance of the video processing may depend upon the ability to store, in the cache, feature-based encoding prediction data such that it is nearby in the cache to the associated encoded video data, even if that encoded video data is not spatially close to the frame(s) from which the feature-based encoding prediction data was originally derived. Cache proximity is associated with the access latency, operational delay, and transmission times for the data. For example, if the feature data from a multitude of frames is contained in a small amount of physical memory and accessed in that form, this is much more efficient than accessing the frames from which those features were derived on a persistent storage device. The encoder 312/decoder 340 (FIGs. 3A, 3C) may include a configurator that stores the prediction data in the cache in such a way to ensure that, when a macroblock or frame is decoded, the feature-based prediction context information is easily accessible from the cache/buffer/framestore.

[00123] Certain embodiments of the present invention can extend the cache by first defining two categories of feature correlation in the previously decoded frames, namely local and non-local previously decoded data for the cache. The local cache can be a set of previously decoded frames that are accessible in batches, or groups of frames, but the particular frames that constitute those groups are determined by detected features. The local cache is driven by features detected in the current frame. The local cache is used to a greater extent when there are relatively few "strong" feature models (models having a long history) for the current frame/macroblock. The local cache processing is based on batch motion compensated prediction, and groups of frames are stored in reference frame buffers. FIG. 6 is a block diagram illustrating an overview of example cache architecture 610-1 according to an embodiment of the invention. The cache access architecture 610-1 includes the decision processes 610 for local cache access 612 (616, 618, 620, 622, and 624) and distant cache access 614 (626, 628, 630, and 632). If the features

are mostly local 612 (for example, there are few strong feature models for the current frame/macroblock), then local cache processing 618 is provided.

[00124] FIG. 7A is a block diagram illustrating the processing involved in utilizing the local (short) cache data 734. The local cache can be a set of previously decoded frames that are accessible in batches, or groups of frames, but the particular frames that constitute those groups are determined by detected features. The local cache 734 in FIG. 7A groups only "short history" features 736, those whose tracks only comprise a small number of frames. The aggregate set of frames encompassed by the short history features determines a joint frameset 738 for those features. Frames in the joint frameset 738 may be prioritized 740 based on the complexity of the feature tracks in the respective frames. In one embodiment, complexity may be determined by the encoding cost of the features from a base encoding process such as H.264. Referring to FIGs. 3B, 3C, 6, and 7A, the local cache may be stored in the framestore 352 or in the cache buffer 348. The locally cached frames are utilized at 620. A GOP/batch 742 based on detected feature instances can then be formed at 622. The GOP/batch based on detected feature instances can be tested at 624 as reference frames 744 for the motion compensation prediction process. Motion compensated prediction done in this way can be said to be "biased" toward feature tracking information, because the reference frames for the motion estimation are the frames with previously-detected feature instances. At 746, additional rollback capabilities are provided to test the applicability of the residual modeling within the GOP/batch, slice, and entropy state. In this way, reference frames that are remote in the video frame sequence to the current frame being encoded can be evaluated more efficiently.

[00125] Thus, certain embodiments of the invention are able to apply analysis to past frames to determine the frames that will have the highest probability of providing matches for the current frame. Additionally, the number of reference frames can be much greater than the typical one-to-sixteen reference frame maximum found in conventional compression. Depending on system resources, the reference frames may number up to the limit of system memory, assuming that there are a sufficient number of useful matches in those frames. Further, the intermediate

form of the data generated by the present invention can reduce the required amount of memory for storing the same number of reference frames.

[00126] When the features have an extensive history 626 in FIG. 6, features are located in storage that is mostly in the non-local/distant cache. The non-local cache is based on two different cache access methods, frame and retained. The frame access of the non-local cache accesses frames directly to create feature models that are then utilized to encode the current frame. The retained mode does not access the previously decoded data directly, but rather utilizes feature models that have been retained as data derived from those previously decoded frames (the feature model and the parameters of the instances of the feature model in those frames) and thereby can be used to synthesize that same data. At 628, the models for the feature instances are accessed. At 630, the reference frames are accessed, and at 632 the combination of optimal reference frames and models are marked for use. Criteria for optimality are based on intermediate feature information for the feature models in each reference frame, including feature strength and feature bandwidth.

[00127] The distant cache 614 can be any previously decoded data (or encoded data) that is preferably accessible in the decoder state. The cache may include, for example, reference frames/GOPs, which are generally a number of frames that precede the current frame being encoded. The decoder cache allows for other combinations of previously decoded frames to be available for decoding the current frame.

[00128] FIG. 7B is a block diagram illustrating the processing involved in utilizing the distant cache data. The distant (non-local) cache 748 illustrates the longer range cache architecture. The distant cache is initialized from the local cache 750 in response to a determination 752 that the detected features have an extensive history (many reoccurrences). . The process then determines which retention mode 754 is used. The two modes of the non-local cache are the retained 760 and non-retained 756. The non-retained 756 is a conventional motion compensated prediction process augmented with predictions based on feature models (similar to the usage of implicit modeling for the hybrid codec described above). The non-retained mode 756 thus accesses 758 reference frames to obtain working predictions. The retained mode is similar to the non-retained mode, but it uses predictions that come explicitly

from the feature model itself 762, 766. The retained model necessarily limits the prediction searches to that data for which the feature model is able to synthesize the feature that it models. Further, the feature model may contain the instance parameterizations for the feature's instances in prior frames, which would be equivalent to the pels contained in those prior frames. The interpolation of the function describing those parameters is also used to provide predictions to the motion compensation prediction process to facilitate frame synthesis 764.

Digital Processing Environment and Communication Network

[00129] Example implementations of the present invention may be implemented in a software, firmware, or hardware environment. In an embodiment, FIG. 8A illustrates one such environment. Client computer(s)/devices 810 and a cloud 812 (or server computer or cluster thereof) provide processing, storage, and input/output devices executing application programs and the like. Client computer(s)/devices 810 can also be linked through communications network 816 to other computing devices, including other client devices/processes 810 and server computer(s) 812. Communications network 816 can be part of a remote access network, a global network (e.g., the Internet), a worldwide collection of computers, Local area or Wide area networks, and gateways that currently use respective protocols (TCP/IP, Bluetooth, etc.) to communicate with one another. Other electronic device/computer network architectures are suitable.

[00130] FIG. 8B is a diagram of the internal structure of a computer/computing node (e.g., client processor/device 810 or server computers 812) in the processing environment of FIG. 8A. Each computer 810, 812 contains a system bus 834, where a bus is a set of actual or virtual hardware lines used for data transfer among the components of a computer or processing system. Bus 834 is essentially a shared conduit that connects different elements of a computer system (e.g., processor, disk storage, memory, input/output ports, etc.) that enables the transfer of information between the elements. Attached to system bus 834 is an I/O device interface 818 for connecting various input and output devices (e.g., keyboard, mouse, displays, printers, speakers, etc.) to the computer 810, 812. Network interface 822 allows the computer to connect to various other devices attached to a network (for example the

network illustrated at 816 of FIG. 8A). Memory 830 provides volatile storage for computer software instructions 824 and data 828 used to implement an embodiment of the present invention (e.g., codec, video encoder/decoder code). Disk storage 832 provides non-volatile storage for computer software instructions 824 (equivalently, "OS program" 826) and data 828 used to implement an embodiment of the present invention; it can also be used to store the video in compressed format for long-term storage. Central processor unit 820 is also attached to system bus 834 and provides for the execution of computer instructions. Note that throughout the present text, “computer software instructions” and "OS program” are equivalent.

[00131] In one embodiment, the processor routines 824 and data 828 are a computer program product (generally referenced 824), including a computer readable medium capable of being stored on a storage device 828, which provides at least a portion of the software instructions for the invention system. The computer program product 824 can be installed by any suitable software installation procedure, as is well known in the art. In another embodiment, at least a portion of the software instructions may also be downloaded over a cable, communication, and/or wireless connection. In other embodiments, the invention programs are a computer program propagated signal product 814 (in Fig 8A) embodied on a propagated signal on a propagation medium (e.g., a radio wave, an infrared wave, a laser wave, a sound wave, or an electrical wave propagated over a global network such as the Internet, or other network(s)). Such carrier media or signals provide at least a portion of the software instructions for the present invention routines/program 824, 826.

[00132] In alternate embodiments, the propagated signal is an analog carrier wave or digital signal carried on the propagated medium. For example, the propagated signal may be a digitized signal propagated over a global network (e.g., the Internet), a telecommunications network, or other network. In one embodiment, the propagated signal is transmitted over the propagation medium over a period of time, such as the instructions for a software application sent in packets over a network over a period of milliseconds, seconds, minutes, or longer. In another embodiment, the computer readable medium of computer program product 824 is a propagation medium that the computer system 810 may receive and read, such as by receiving

the propagation medium and identifying a propagated signal embodied in the propagation medium, as described above for computer program propagated signal product.

Digital rights management

[00133] In some embodiments, the models of the present invention can be used as a way to control access to the encoded digital video. For example, without the relevant models, a user would not be able to playback the video file. An example implementation of this approach is discussed in U.S. Application No. 12/522,357, filed January 4, 2008, the entire teachings of which are incorporated by reference. The models can be used to "lock" the video or be used as a key to access the video data. The playback operation for the coded video data can depend on the models. This approach makes the encoded video data unreadable without access to the models.

[00134] By controlling access to the models, access to playback of the content can be controlled. This scheme can provide a user-friendly, developer-friendly, and efficient solution to restricting access to video content.

[00135] Additionally, the models can progressively unlock the content. With a certain version of the models, an encoding might only decode to a certain level; then with progressively more complete models, the whole video would be unlocked. Initial unlocking might enable thumbnails of the video to be unlocked, giving the user the capability of determining if they want the full video. A user that wants a standard definition version would procure the next incremental version of the models. Further, the user needing high definition or cinema quality would download yet more complete versions of the models. The models are coded in such a way as to facilitate a progressive realization of the video quality commensurate with encoding size and quality, without redundancy.

Flexible macroblock ordering and scalable video coding

[00136] To improve the encoding process and produce compression benefits, example embodiments of the invention may extend conventional encoding/decoding processes. In one embodiment, the present invention may be applied with flexible

macroblock ordering (FMO) and scalable video coding (SVC), which are themselves extensions to the basic H.264 standard.

[00137] FMO allocates macroblocks in a coded frame to one of several types of slice groups. The allocation is determined by a macroblock allocation map, and macroblocks within a slice group do not have to be contiguous. FMO can be useful for error resilience, because slice groups are decoded independently: if one slice group is lost during transmission of the bitstream, the macroblocks in that slice group can be reconstructed from neighboring macroblocks in other slices. In one embodiment of the current invention, feature-based compression can be integrated into the "foreground and background" macroblock allocation map type in an FMO implementation. Macroblocks associated with features comprise foreground slice groups, and all other macroblocks (those not associated with features) comprise background slice groups.

[00138] SVC provides multiple encodings of video data at different bitrates. A base layer is encoded at a low bitrate, and one or more enhancement layers are encoded at higher bitrates. Decoding of the SVC bitstreams can involve just the base layer (for low bitrate/low quality applications) or some or all of the enhancement layers as well (for higher bitrate/quality applications). Because the substreams of the SVC bitstream are themselves valid bitstreams, the use of SVC provides increased flexibility in different application scenarios, including decoding of the SVC bitstream by multiple devices (at different qualities, depending on device capabilities) and decoding in environments with varying channel throughput, such as Internet streaming.

[00139] There are three common types of scalability in SVC processing: temporal, spatial, and quality. In one embodiment of the current invention, feature-based compression can be integrated into a quality scalability implementation by including the primary feature-based predictions in the base layer (see the section above on model-based primary and secondary predictions). The coded frames in the base layer can then serve as reference frames for coding in the enhancement layer, where secondary feature-based predictions can be used. In this way, information from feature-based predictions can be added incrementally to the encoding, instead of all at once. In an alternate embodiment, all feature-based predictions (primary

and secondary) can be moved to enhancement layers, with only conventional predictions used in the base layer.

[00140] It should be noted that although the figures described herein illustrate example data/execution paths and components, one skilled in the art would understand that the operation, arrangement, and flow of data to/from those respective components can vary depending on the implementation and the type of video data being compressed. Therefore, any arrangement of data modules/data paths can be used.

[00141] While this invention has been particularly shown and described with references to example embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the scope of the invention encompassed by the appended claims.

CLAIMS

What is claimed is:

1. A method for processing video data, comprising:

detecting at least one of a feature and an object in the region of interest using a detection algorithm in at least one frame;

modeling the detected at least one of the feature and the object using a set of parameters;

associating any instances of the at least one of the feature and the object across frames;

forming at least one track of the associated instances;

relating the at least one track to at least one specific block of video data to be encoded; and

producing a model-based prediction for the at least one specific block of video data using the related track information, said producing including storing the model-based prediction as standards-compliant encoded/decoded video data.

2. The method of Claim 1 wherein the detection algorithm is of a class of nonparametric feature detection algorithms.

3. The method of claim 1, wherein the set of parameters includes information about the at least one of the feature and the object and is stored in memory.

4. The method of claim 3, wherein the respective parameter of the respective feature includes a feature descriptor vector and a location of the respective feature.

5. The method of claim 4, wherein the respective parameter is generated when the respective feature is detected.

6. The method of claim 1, wherein the at least one specific block of video data is a macroblock, the at least one track relating features to the macroblock.

7. A method for processing video data, comprising:

detecting at least one of a feature and an object in the region of interest;

modeling the at least one of the feature and the object using a set of parameters;

associating any instances of the at least one of the feature and the object across frames;

forming at least one matrix of the associated instances;

relating the at least one matrix to at least one specific block of video data to be encoded; and

producing a model-based prediction for the at least one specific block of video data using the related matrix information, said producing storing the model-based prediction as standards-compliant encoded/decoded video data.

8. The method of claim 7, wherein the set of parameters includes information about the at least one of the feature and the object and is stored in memory.

9. The method of claim 8, wherein the respective parameter of the respective feature includes a feature descriptor vector and a location of the respective feature.

10. The method of claim 9, wherein the respective parameter is generated when the respective feature is detected.

11. The method of claim 7, further comprising:

summarizing the at least one matrix using at least one subspace of a certain vector space as a parametric model of the associated at least one of the feature and the object.

12. A codec for processing video data, comprising:

a feature-based detector configured to identify instances of a feature in at least two video frames, where each identified feature instance includes a plurality of

pixels exhibiting data complexity relative to other pixels in the one or more video frames;

a modeler operatively coupled to the feature based detector and configured to create feature-based models modeling correspondence of the feature instances in two or more video frames; and

a cache configured to prioritize use of the feature-based models if it is determined that a standards-compliant encoding of the feature instances using the feature-based models provides improved compression efficiency relative to a standards-compliant encoding of the feature instances using a first video encoding process.

13. The codec of claim 12, wherein the data complexity is determined when an encoding of the pixels by a conventional video compression technique exceeds a predetermined threshold.

14. The codec of claim 12, wherein the data complexity is determined when a bandwidth amount allocated to encode the feature by conventional video compression technique exceeds a predetermined threshold.

15. The codec of claim 14, wherein the predetermined threshold is at least one of: a preset value, a preset value stored in a database, a value set as the average bandwidth amount allocated for previously encoded features, and a value set as the median bandwidth amount allocated for previously encoded features.

16. The codec of claim 12, wherein the first video encoding process includes a motion compensation prediction process.

17. The codec of claim 12, wherein the prioritization of use is determined by comparison of encoding costs for each potential solution within Competition Mode, a potential solution comprising a tracker, a primary prediction motion model, a primary prediction sampling scheme, a subtiling scheme, a reconstruction algorithm, and (possibly) a secondary prediction scheme.

18. The codec of claim 17, wherein the prioritization of use of the feature-based modeling initiates a use of that data complexity level of the feature instance as the threshold value, such that if a future feature instance exhibits the same or more data complexity level as the threshold value then the encoder automatically determines to initiate and use feature-based compression on the future feature instance.

19. The codec of claim 12, wherein the feature detector utilizes one of an FPA tracker, an MBC tracker, and a SURF tracker.

20. A codec for processing video data, comprising:

a feature-based detector to identify an instance of a feature in at least two video frames, an identified feature instance including a plurality of pixels exhibiting data complexity relative to other pixels in at least one of the at least two video frames;

a modeler operatively coupled to the feature-based detector, wherein the modeler creates a feature-based model modeling correspondence of the respective identified feature instance in the at least two video frames; and

a memory, wherein for a plurality of the feature-based models, the memory prioritizes standards compliant use of a respective feature-based model if an improved compression efficiency of the identified feature instance is determined.

21. The codec of claim 20, wherein the improved compression efficiency of the identified feature instance is determined by comparing the compression efficiency of the identified feature relative to one of: a standards compliant encoding of the feature instance using a first video encoding process and a predetermined compression efficiency value stored in a database.

22. A method for processing video data, comprising:

modeling a feature by vectorizing at least one of a feature pel and a feature descriptor;

identifying similar features by at least one of (a) minimizing means-squared error (MSE) and (b) maximizing inner products between different feature pel vectors or feature descriptors; and

applying a standard motion estimation and compensation algorithm to account for translational motion of the feature, resulting in standards-compliant encoded/decoded video data.

23. A method for processing video data, comprising:

implementing a model-based prediction by configuring a codec to encode a target frame;

encoding a macroblock in the target frame using a conventional encoding process;

analyzing the macroblock encoding, wherein the conventional encoding of the macroblock is deemed to be at least one of efficient and inefficient;

wherein if the conventional encoding is deemed inefficient, the encoder is analyzed by generating several predictions for the macroblock based on multiple models, and

wherein the evaluation of the several predictions of the macroblock are based on an encoding size; and

ranking the predictions of the macroblock with the conventionally encoded macroblock.

24. The method of claim 23, wherein the conventional encoding of the macroblock is efficient if an encoding size is less than a predetermined threshold size.

25. The method of claim 23, wherein the conventional encoding of the macroblock is efficient if the target macroblock is a skip macroblock.

26. The method of claim 23, wherein the conventional encoding of the macroblock is inefficient if the encoding size is larger than a threshold.

27. The method of claim 23, wherein if the conventional encoding of the macroblock is deemed inefficient, Competition Mode encodings for the macroblock are generated to compare their relative compression efficiencies.

28. The method of claim 27, wherein the encoding algorithm for Competition Mode includes:

subtracting the prediction from the macroblock to generate a residual signal;

transforming the residual signal using an approximation of a 2-D block-based DCT; and

encoding transform coefficients using an entropy encoder.

29. The method of claim 23 wherein the encoder being analyzed by generating several predictions includes generating a composite prediction that sums a primary prediction and a weighted version of a secondary prediction.

30. A method for processing video data, comprising:

modeling data at multiple fidelities for a model-based compression, the multiple fidelities including at least one of a macroblock level, a macroblock as feature level, a feature level, and an object level,

wherein the macroblock level uses a block-based motion estimation and compensation (BBMEC) application to find predictions for each tile from a limited search space in previously decoded reference frames,

wherein the macroblock as feature level (i) uses a first BBMEC application identical to the macroblock level to find a first prediction for a target macroblock from a most-recent reference frame, (ii) uses a second BBMEC application to find a second prediction for the first prediction by searching in a second-most-recent frame, and (iii) creates a track for the target macroblock by applying BBMEC applications through progressively older frames,

wherein the feature level detects and tracks features independent of the macroblock grid and associates the features with overlapping macroblocks such that feature tracks are used to navigate previously-decoded reference frames to find better matches for the overlapping macroblocks; and where multiple features overlap

a given target macroblock, the feature with greatest overlap is selected to model that target macroblock, and

wherein the object level an object encompasses or overlaps multiple macroblocks, a single motion vector can be calculated for all of the macroblocks associated with the object to result in computation and encoding size savings.

31. The method of claim 30, wherein the multiple fidelities are examined sequentially.

32. The method of claim 30, wherein the multiple fidelities are examined in competition mode.

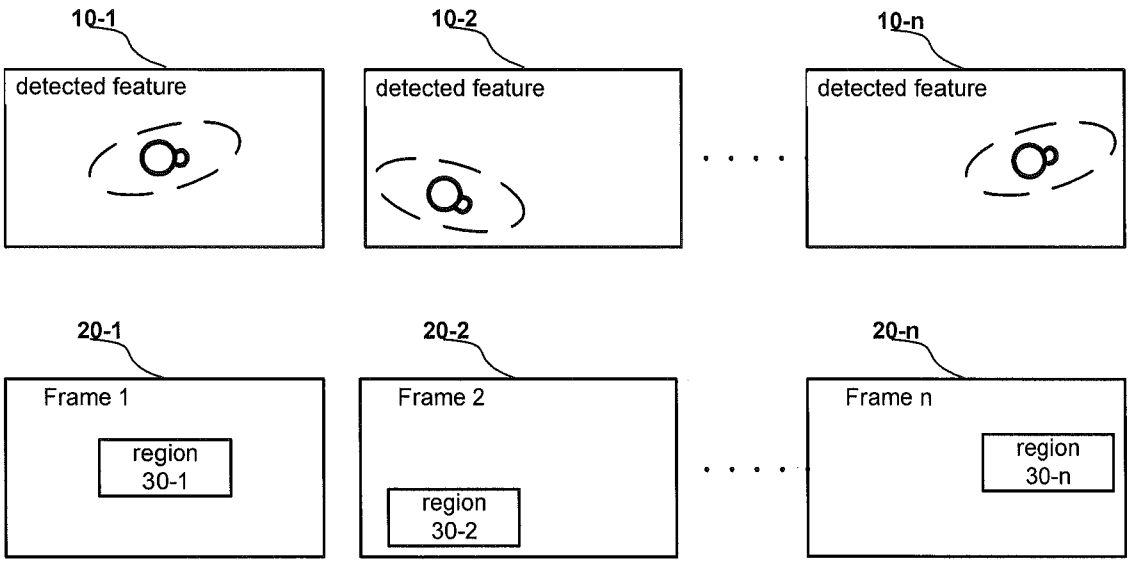
10-1
detected feature
10-2
detected feature
10-n
detected feature
20-1
Frame 1
region
30-1
20-2
Frame 2
region
30-2
20-n
Frame n
region
30-n

FIG. 1A

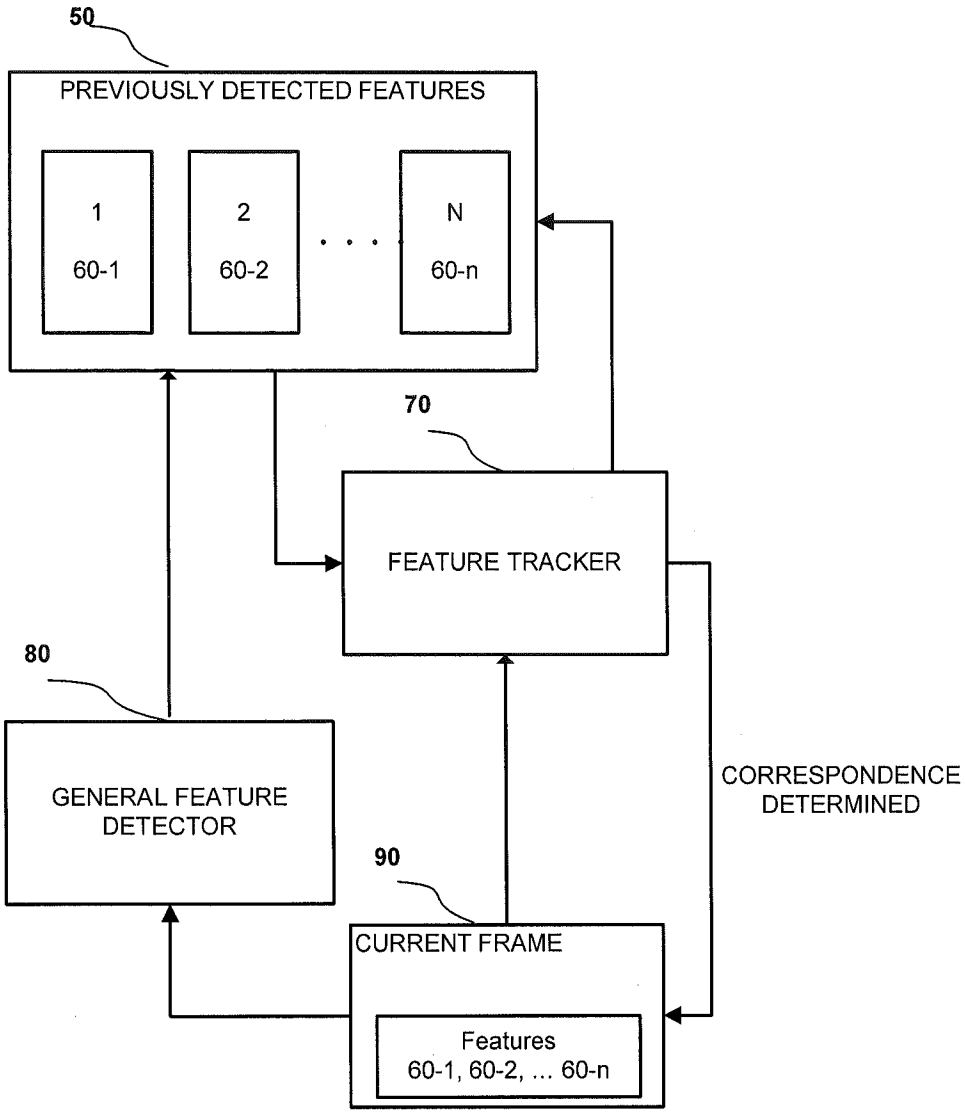
50
PREVIOUSLY DETECTED FEATURES
1
60-1
2
60-2
N
60-n
70
FEATURE TRACKER
80
GENERAL FEATURE DETECTOR
CORRESPONDENCE DETERMINED
90
CURRENT FRAME
Features
60-1, 60-2, ... 60-n

FIG. 1B

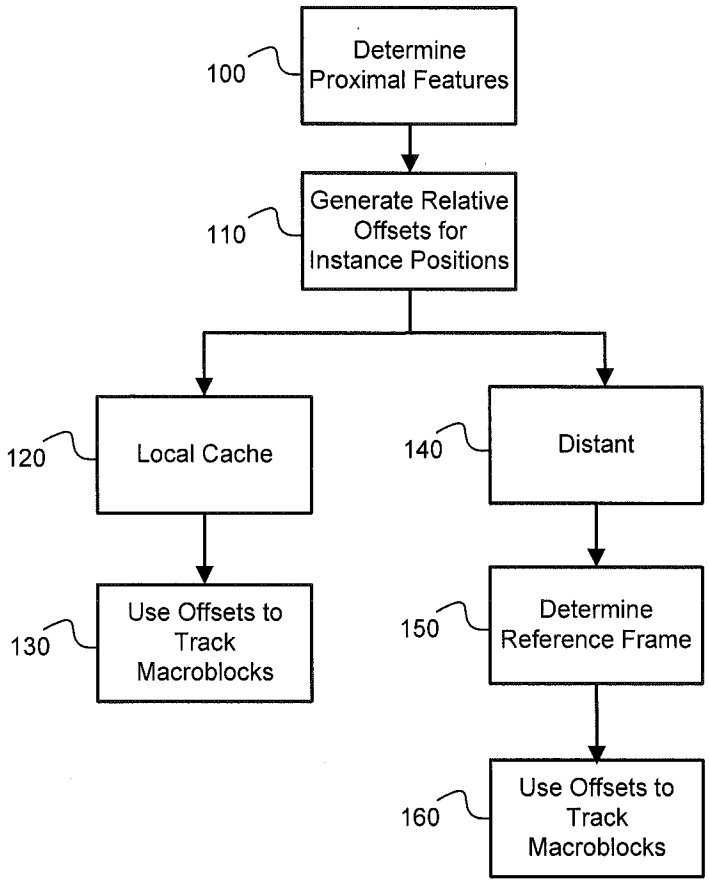
100
Determine Proximal Features
110
Generate Relative Offsets for Instance Positions
120
Local Cache
130
Use Offsets to Track Macroblocks
140
Distant
150
Determine Reference Frame
160
Use Offsets to Track Macroblocks

FIG. 1C

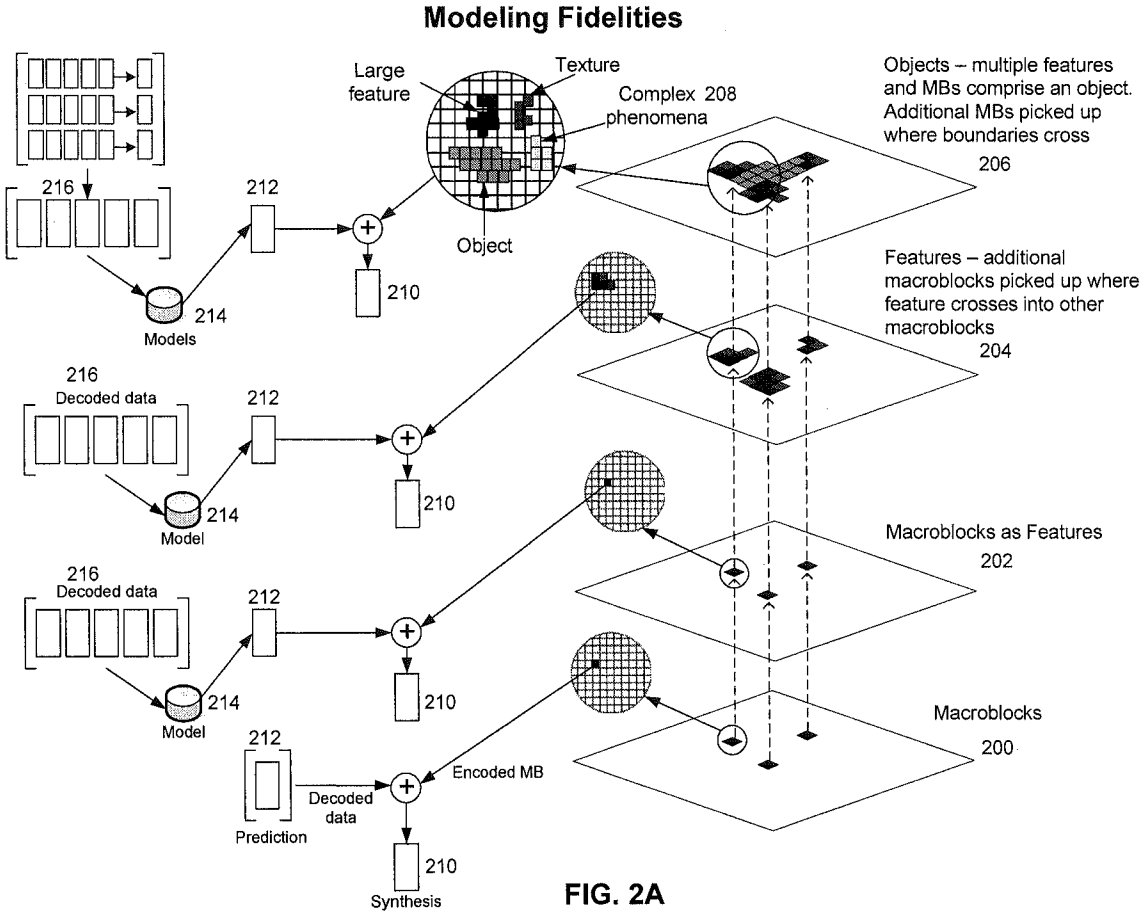
Modeling Fidelities
Large feature
Texture
Complex 208 phenomena
Object
Objects – multiple features and MBs comprise an object. Additional MBs picked up where boundaries cross
206
Features – additional macroblocks picked up where feature crosses into other macroblocks
204
Macroblocks as Features
202
Macroblocks
200
216
212
210
214
Models
Decoded data
Model
Prediction
Encoded MB
Synthesis

FIG. 2A

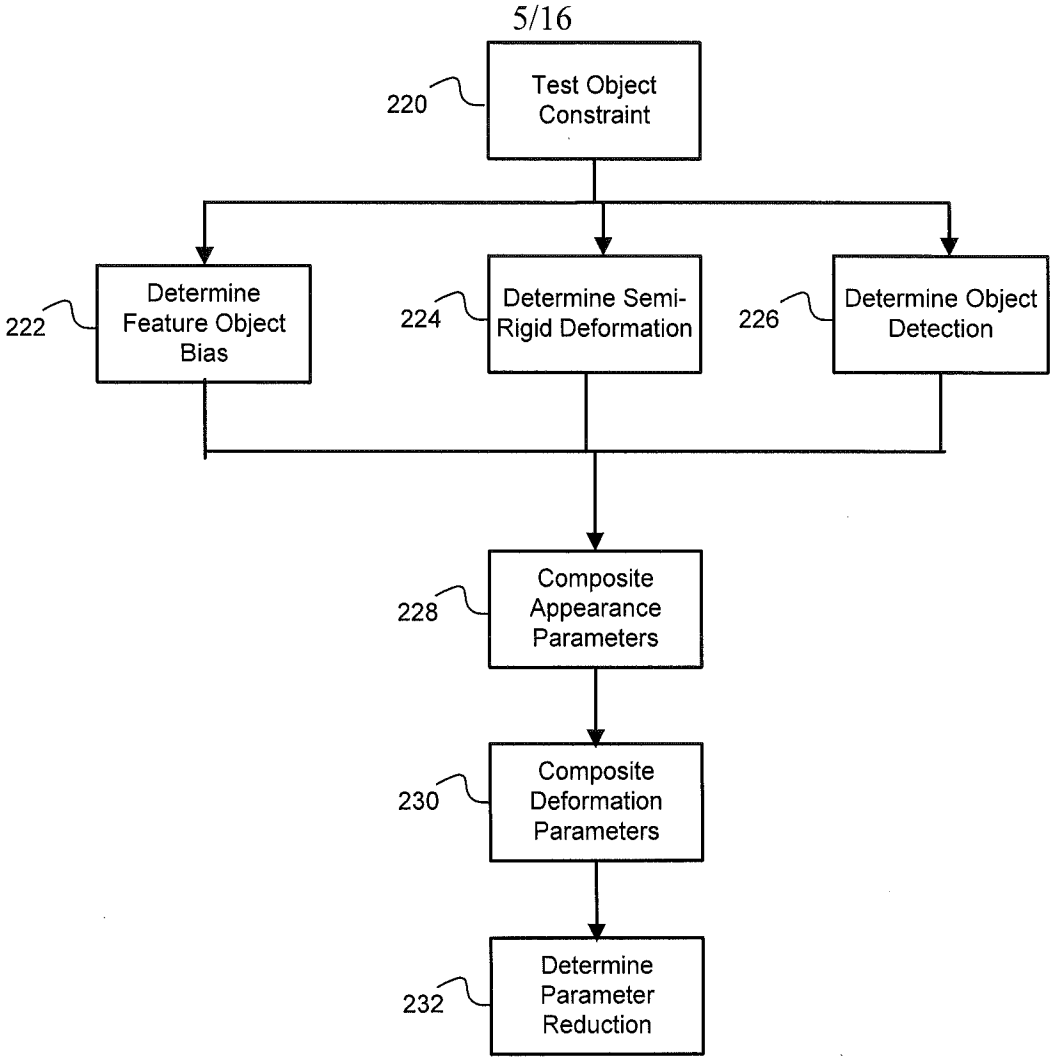
220
Test Object Constraint
222
Determine Feature Object Bias
224
Determine Semi-Rigid Deformation
226
Determine Object Detection
228
Composite Appearance Parameters
230
Composite Deformation Parameters
232
Determine Parameter Reduction

FIG. 2B

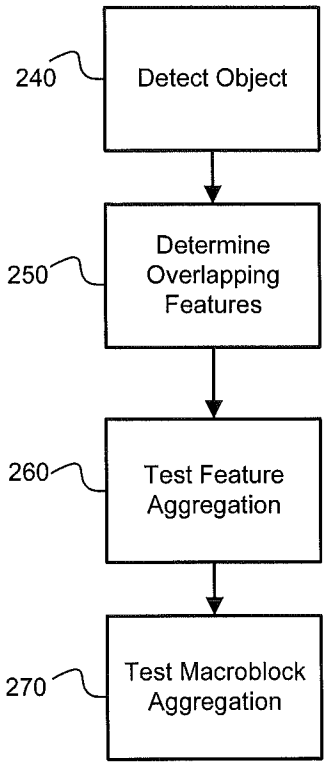
240
Detect Object
250
Determine Overlapping Features
260
Test Feature Aggregation
270
Test Macroblock Aggregation

FIG. 2C

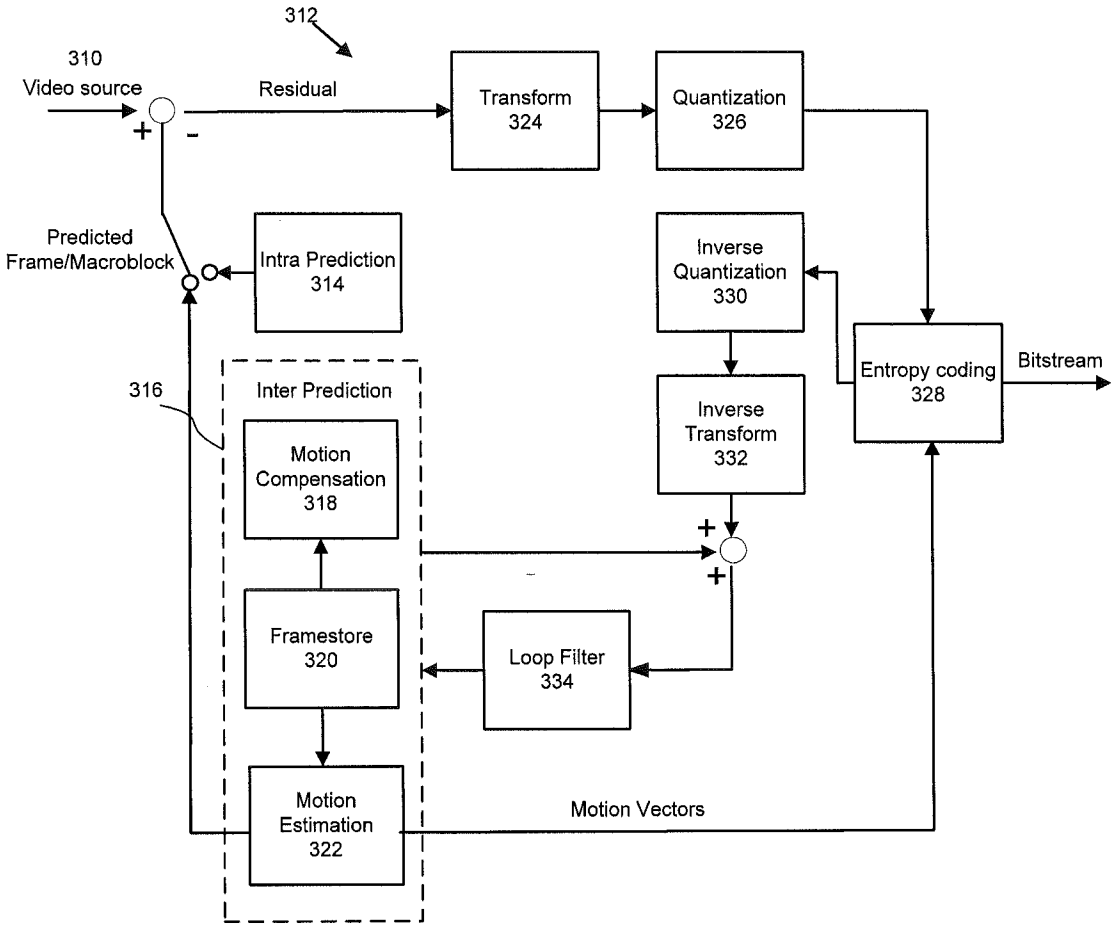
312
310
Video source
Residual
+
-
Transform
324
Quantization
326
Predicted
Frame/Macroblock
Intra Prediction
314
Inverse
Quantization
330
Entropy coding
328
Bitstream
316
Inter Prediction
Motion
Compensation
318
Inverse
Transform
332
+
+
Framestore
320
Loop Filter
334
Motion
Estimation
322
Motion Vectors

FIG. 3A

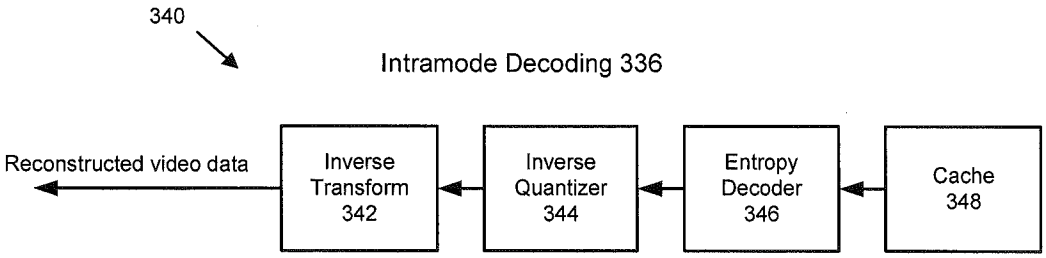
340
Intramode Decoding 336
Reconstructed video data
Inverse Transform 342
Inverse Quantizer 344
Entropy Decoder 346
Cache 348

FIG. 3B

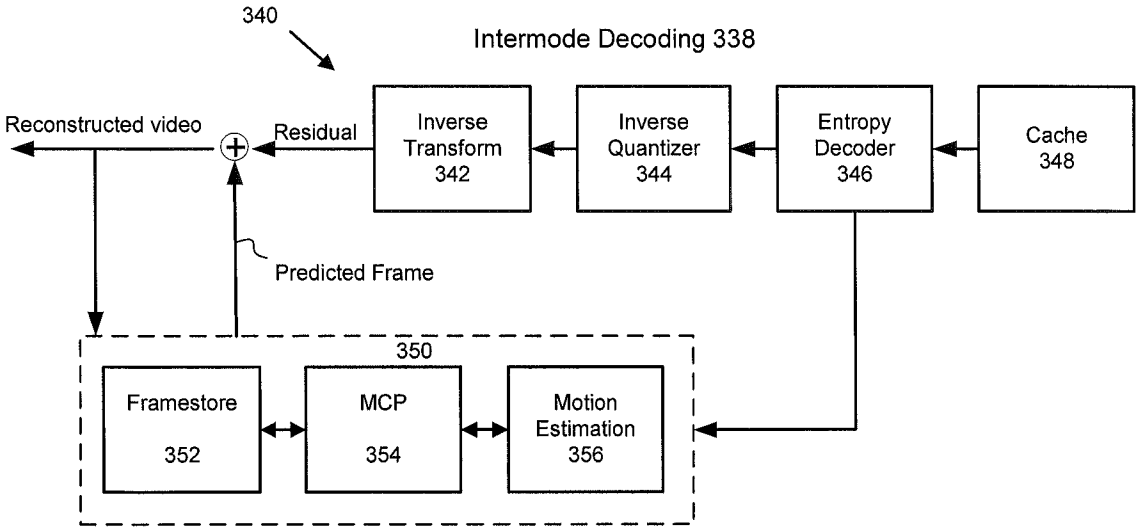
340
Intermode Decoding 338
Reconstructed video
Residual
Inverse Transform 342
Inverse Quantizer 344
Entropy Decoder 346
Cache 348
Predicted Frame
350
Framestore 352
MCP 354
Motion Estimation 356

FIG. 3C

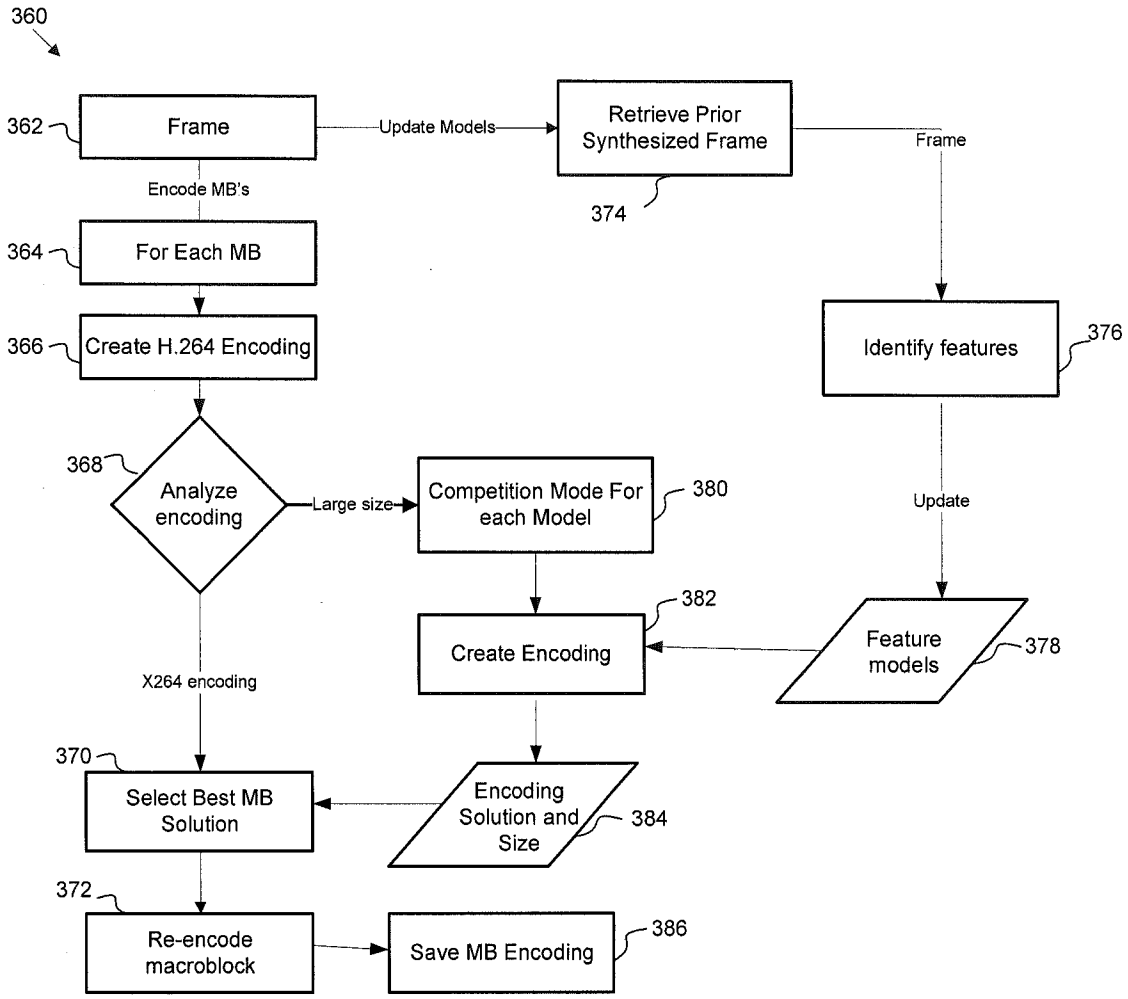
360
362
Frame
Update Models
Retrieve Prior Synthesized Frame
374
Frame
Encode MB's
364
For Each MB
366
Create H.264 Encoding
Identify features
376
368
Analyze encoding
Large size
Competition Mode For each Model
380
Update
382
Create Encoding
Feature models
378
X264 encoding
370
Select Best MB Solution
Encoding Solution and Size
384
372
Re-encode macroblock
Save MB Encoding
386

FIG. 3D

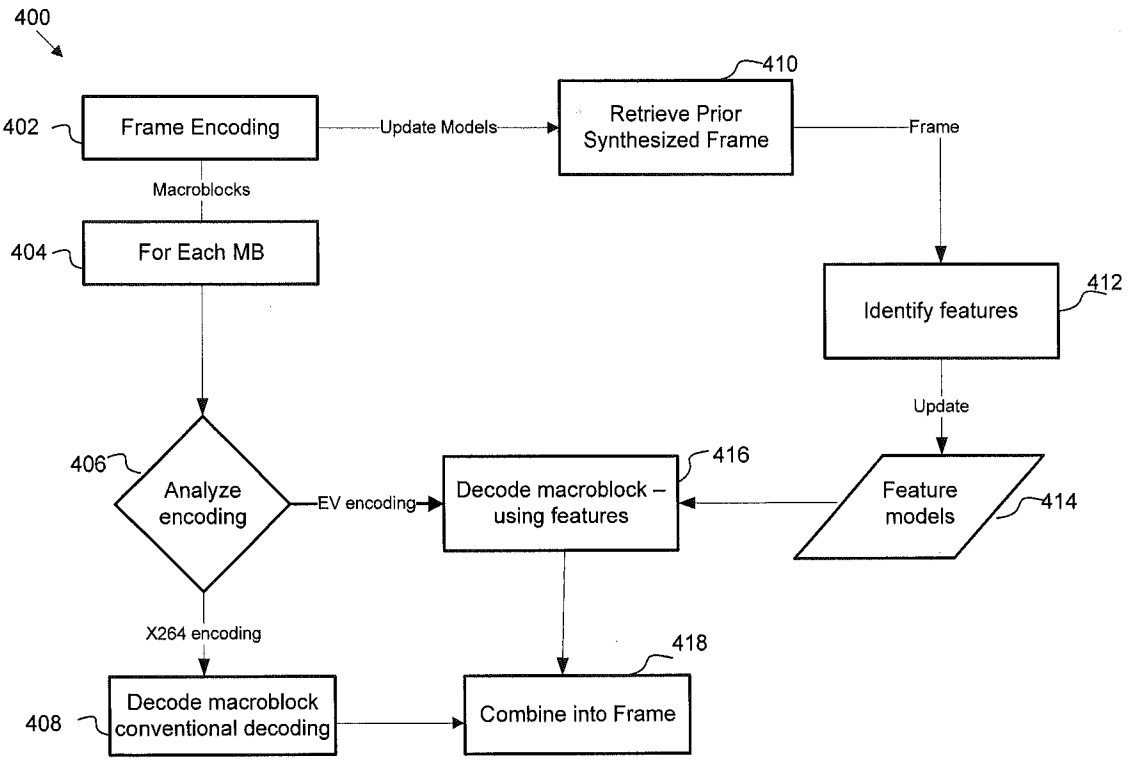
400
402
Frame Encoding
Update Models
410
Retrieve Prior Synthesized Frame
Frame
Macroblocks
404
For Each MB
412
Identify features
Update
406
Analyze encoding
EV encoding
416
Decode macroblock – using features
Feature models
414
X264 encoding
418
408
Decode macroblock conventional decoding
Combine into Frame

FIG. 4

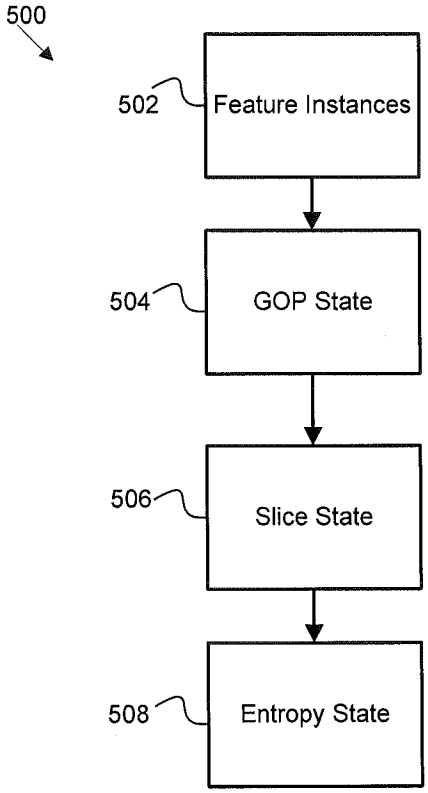
500
502
Feature Instances
504
GOP State
506
Slice State
508
Entropy State

FIG. 5

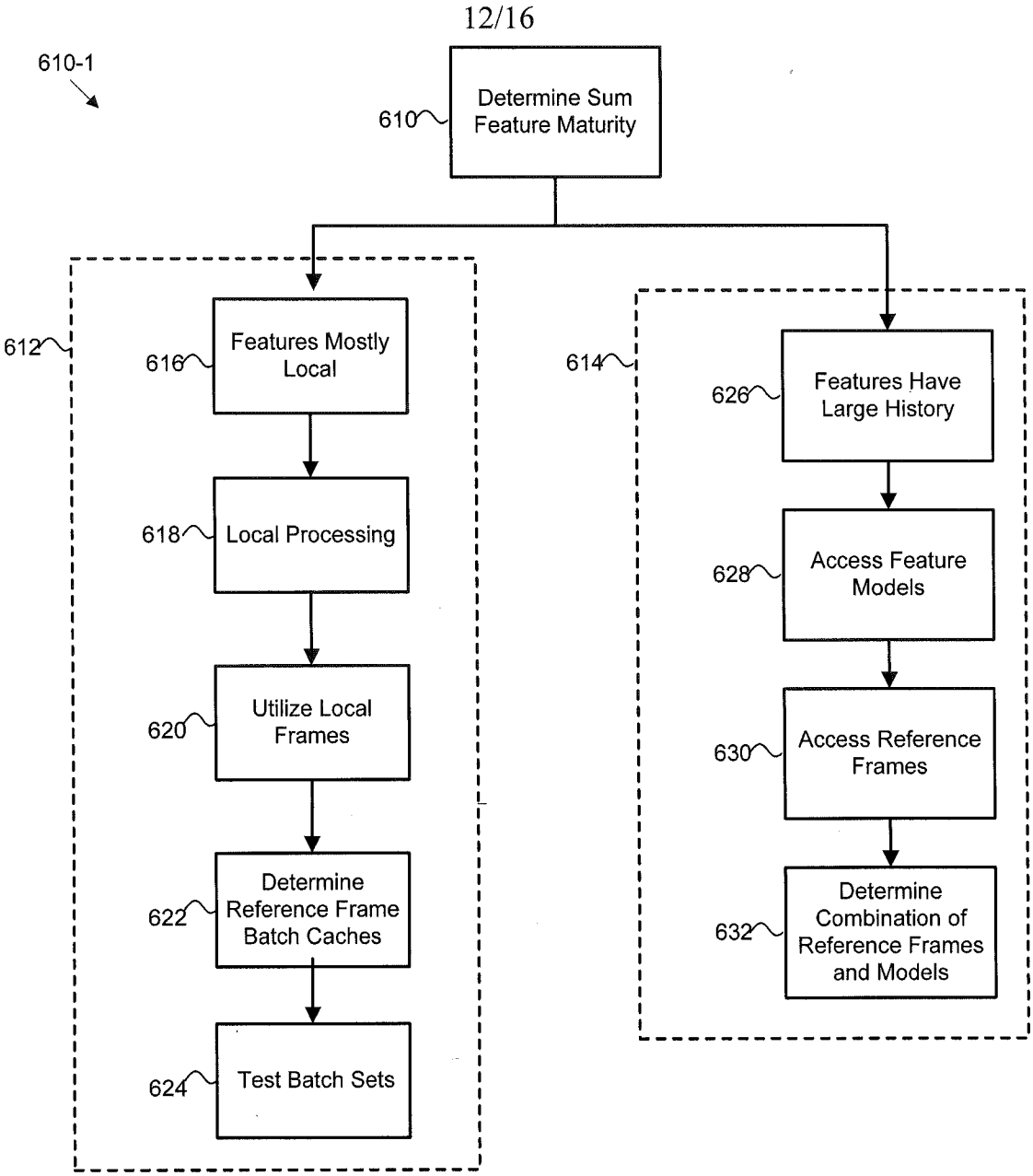
610-1
610
Determine Sum Feature Maturity
612
616
Features Mostly Local
618
Local Processing
620
Utilize Local Frames
622
Determine Reference Frame Batch Caches
624
Test Batch Sets
614
626
Features Have Large History
628
Access Feature Models
630
Access Reference Frames
632
Determine Combination of Reference Frames and Models

Fig. 6

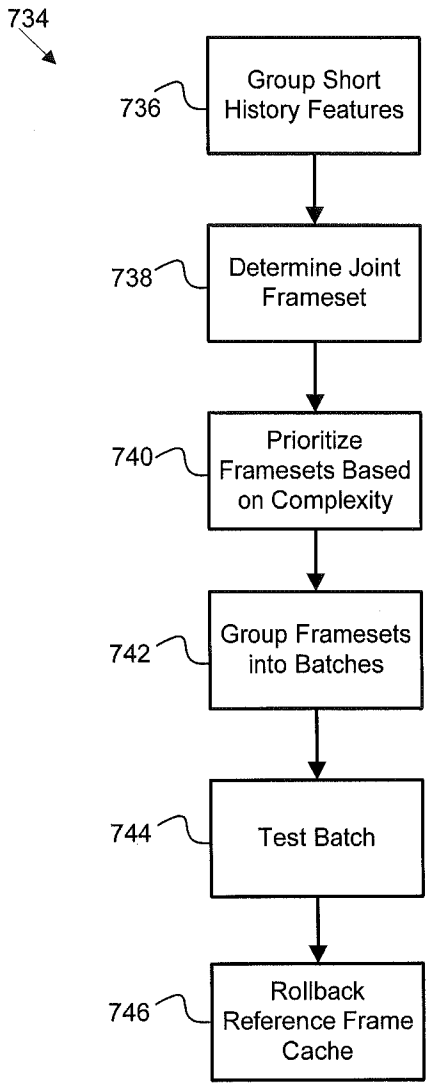
734
736
Group Short History Features
738
Determine Joint Frameset
740
Prioritize Framesets Based on Complexity
742
Group Framesets into Batches
744
Test Batch
746
Rollback Reference Frame Cache

FIG. 7A

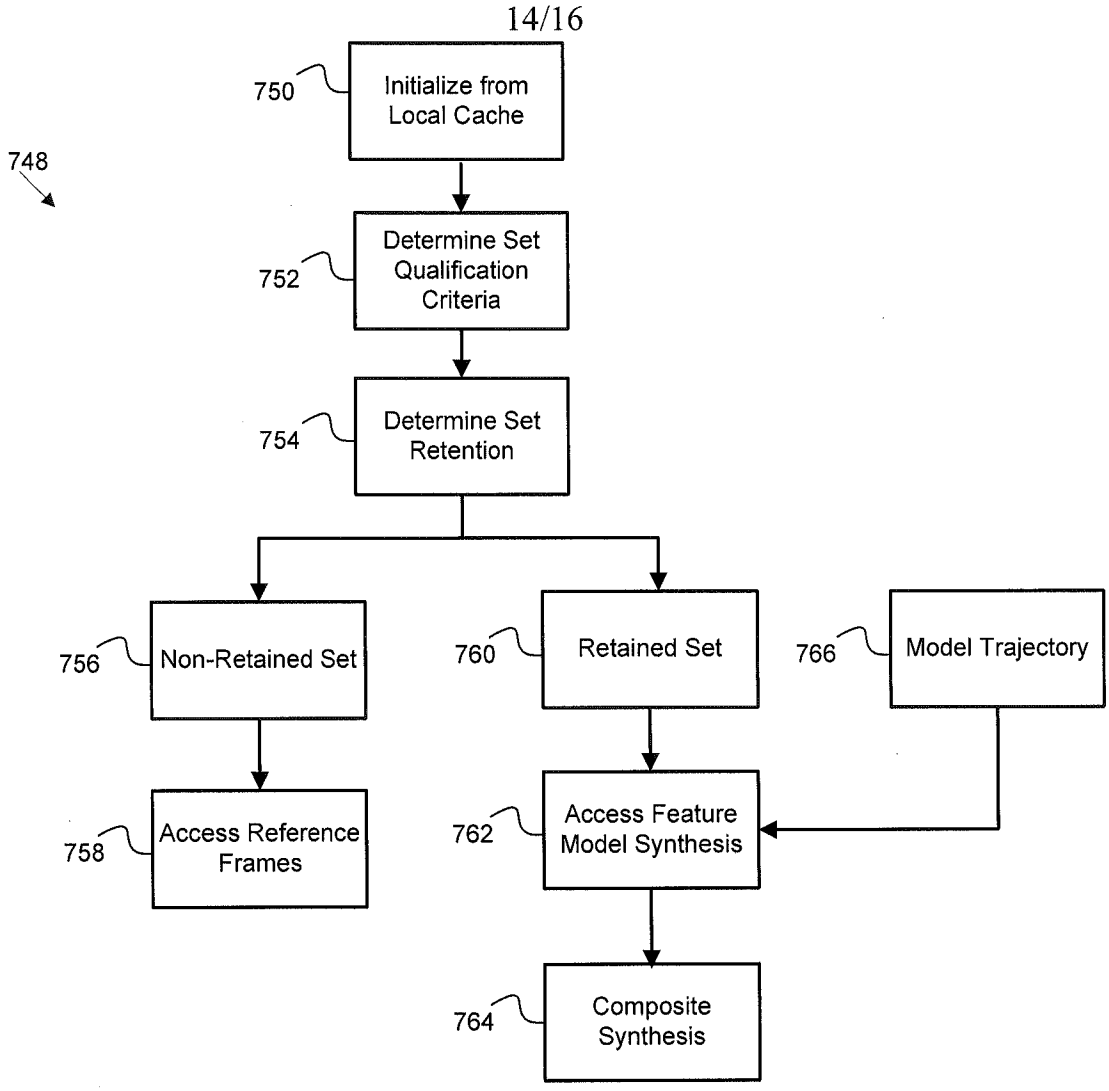
748
750
Initialize from Local Cache
752
Determine Set Qualification Criteria
754
Determine Set Retention
756
Non-Retained Set
758
Access Reference Frames
760
Retained Set
762
Access Feature Model Synthesis
764
Composite Synthesis
766
Model Trajectory

FIG. 7B

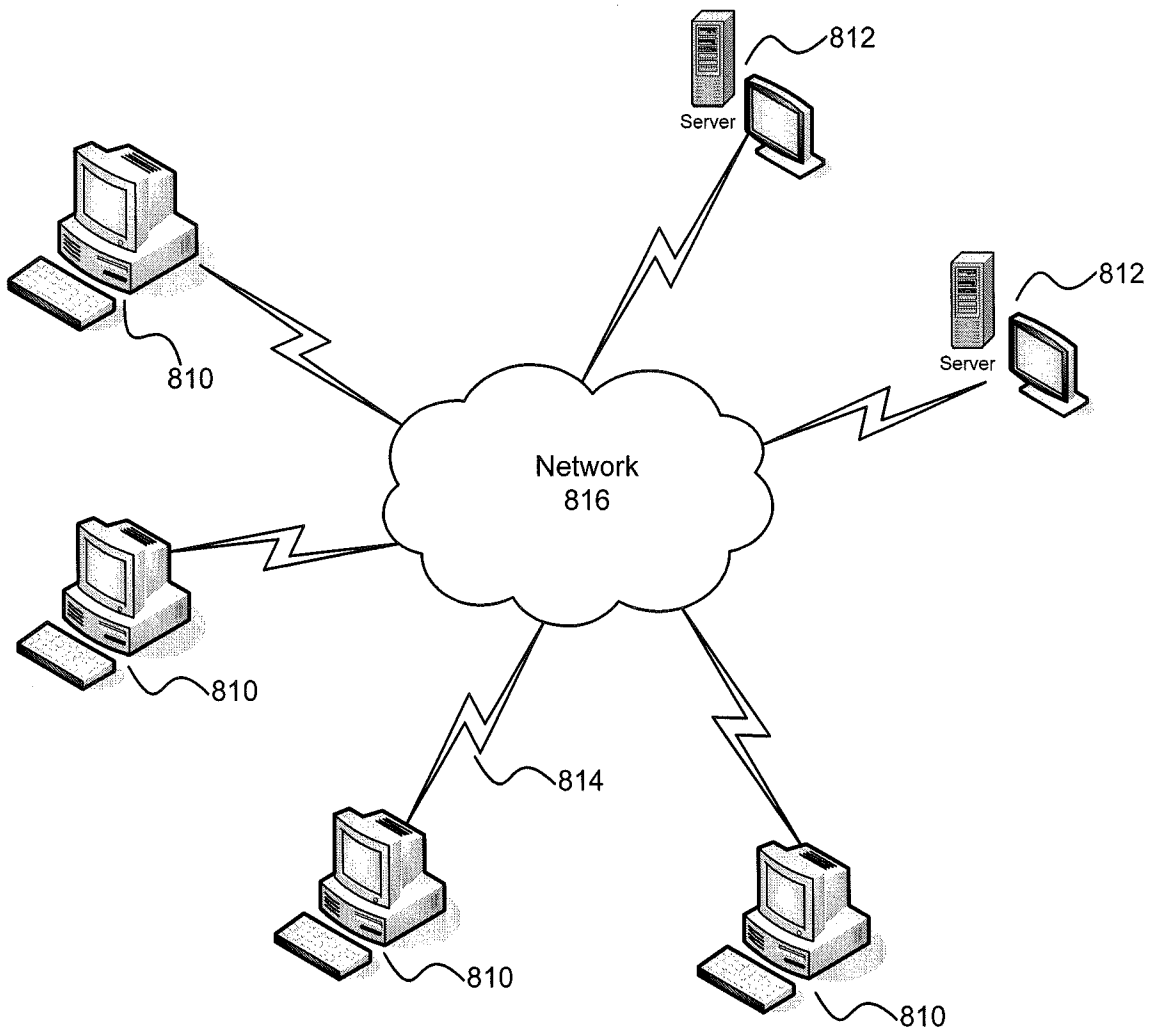

FIG. 8A

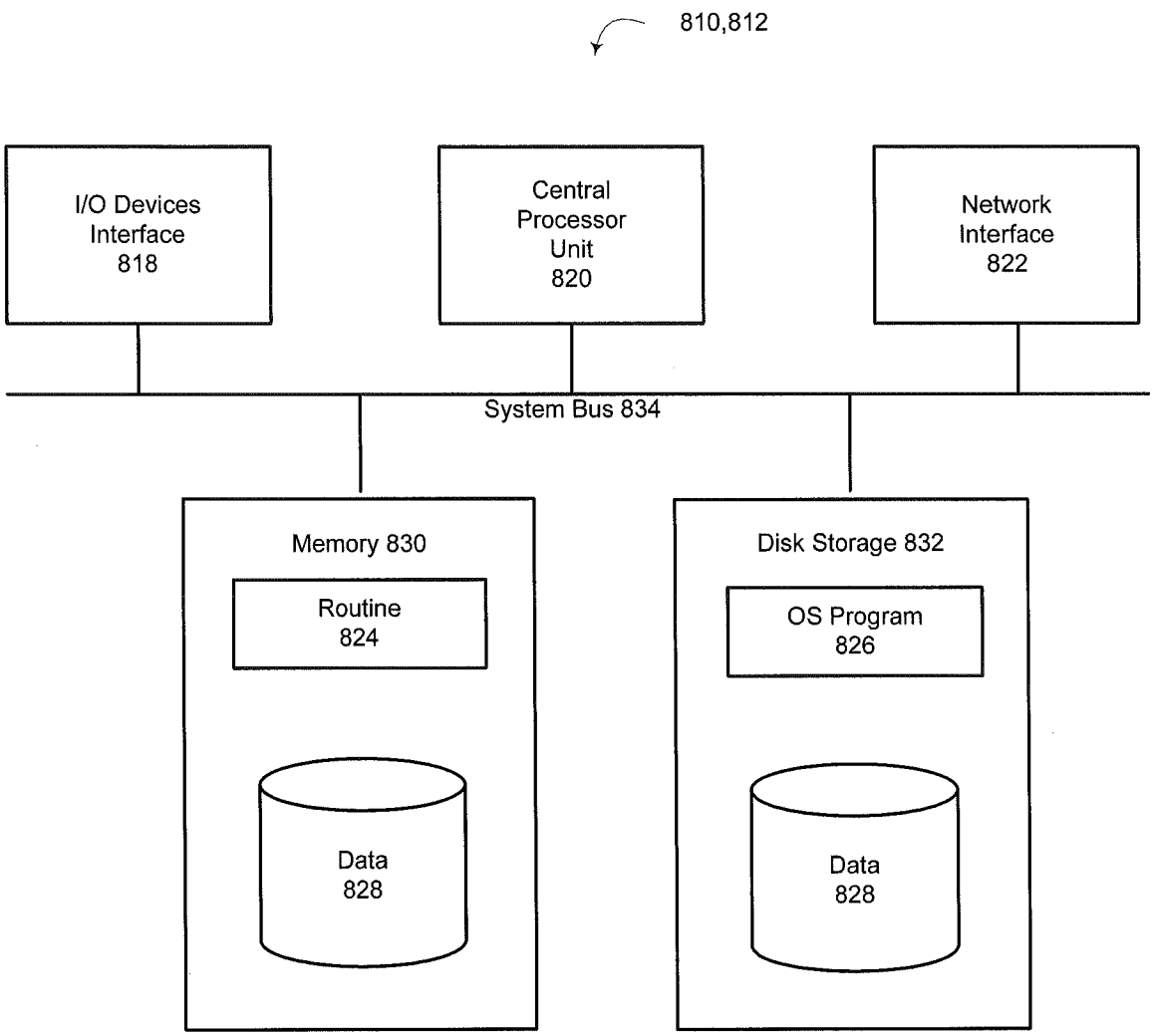
810,812
I/O Devices Interface 818
Central Processor Unit 820
Network Interface 822
System Bus 834
Memory 830
Routine 824
Data 828
Disk Storage 832
OS Program 826
Data 828

Fig. 8B

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2013/043884

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)
G06K H04N

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)
EPO-Internal, COMPENDEX, INSPEC, IBM-TDB, 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	WO 2008/091483 A2 (EUCLID DISCOVERIES LLC [US]; PACE CHARLES P [US]) 31 July 2008 (2008-07-31) the whole document	1-11
X	US 2011/058609 A1 (CHAUDHURY SANTANU [IN] ET AL) 10 March 2011 (2011-03-10)	1,3-5
A	paragraph [0033] - paragraph [0059]; figure 3	2,6-11
X	US 6 738 424 B1 (ALLMEN MARK [US] ET AL) 18 May 2004 (2004-05-18) column 14, line 32 - line 52; figure 4(a) column 20, line 66 - column 21, line 60	1-11

-/--

[X] Further documents are listed in the continuation of Box C.

[X] 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	Date of mailing of the international search report
22 July 2013	16/10/2013

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Georgiou, Georgia

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2013/043884

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EBRAHIMI T ET AL: "MPEG-4 natural video coding - An overview", SIGNAL PROCESSING. IMAGE COMMUNICATION, ELSEVIER SCIENCE PUBLISHERS, AMSTERDAM, NL, vol. 15, no. 4-5, 1 January 2000 (2000-01-01), pages 365-385, XP027357196, ISSN: 0923-5965 [retrieved on 2000-01-01] section 3. Shape coding tools -----	1-11

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-11

   Processing video data comprising detecting features, modeling features, tracking features across frames, relating the tracked features to blocks of data to be coded and producing a model based prediction of video data for encoding.

   ---

2. claims: 12-21

   Prioritizing use of feature based correspondence models if they provide improved compression efficiency.

   ---

3. claim: 22

   Identifying similar features and accommodate small changes in said features by motion estimation and motion compensation.

   ---

4. claims: 23-29

   Comparing feature based video coding to conventional video coding in terms of compression efficiency

   ---

5. claims: 30-32

   Modeling data at multiple fidelities for model based compression.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2013/043884

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

1-11

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2013/043884

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2008091483 A2	31-07-2008	CA 2676219 A1	31-07-2008
		CN 101939991 A	05-01-2011
		EP 2130381 A2	09-12-2009
		JP 2010526455 A	29-07-2010
		TW 200838316 A	16-09-2008
		US 2010008424 A1	14-01-2010
		US 2013083854 A1	04-04-2013
		WO 2008091483 A2	31-07-2008
US 2011058609 A1	10-03-2011	NONE	
US 6738424 B1	18-05-2004	NONE	