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

(19) **World Intellectual Property Organization**
International Bureau

(43) **International Publication Date**
5 August 2010 (05.08.2010)

PCT

(10) **International Publication Number**
WO 2010/088030 A2

(51) **International Patent Classification**: *H04N 7/24* (2006.01)

(21) **International Application Number**: PCT/US2010/020784

(22) **International Filing Date**: 12 January 2010 (12.01.2010)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:
12/362,420 29 January 2009 (29.01.2009) US

(71) **Applicant** *(for all designated States except US)*: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) **Inventors**: **GU, Chuang**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **CHAN, Chun-Wei**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **CHEN, William**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **SPEARS, Stacey**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **HOLCOMB, Thomas W.**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **LIN, Chih-Lung**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **MEHROTRA, Sanjeev**; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM,

[Continued on next page]

(54) **Title**: MULTIPLE BIT RATE VIDEO ENCODING USING VARIABLE BIT RATE AND DYNAMIC RESOLUTION FOR ADAPTIVE VIDEO STREAMING

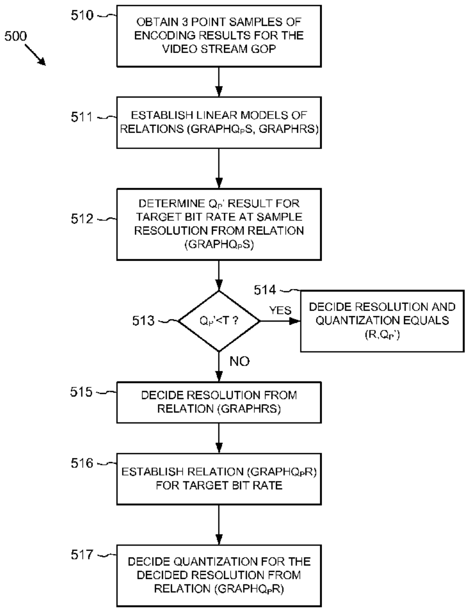
500
510
OBTAIN 3 POINT SAMPLES OF ENCODING RESULTS FOR THE VIDEO STREAM GOP
511
ESTABLISH LINEAR MODELS OF RELATIONS (GRAPHQ$_P$S, GRAPHRS)
512
DETERMINE Q$_P$' RESULT FOR TARGET BIT RATE AT SAMPLE RESOLUTION FROM RELATION (GRAPHQ$_P$S)
513
Q$_P$'<T ?
YES
514
DECIDE RESOLUTION AND QUANTIZATION EQUALS (R,Q$_P$')
NO
515
DECIDE RESOLUTION FROM RELATION (GRAPHRS)
516
ESTABLISH RELATION (GRAPHQ$_P$R) FOR TARGET BIT RATE
517
DECIDE QUANTIZATION FOR THE DECIDED RESOLUTION FROM RELATION (GRAPHQ$_P$R)

FIG. 5

(57) **Abstract**: A video encoding system encodes video streams for multiple bit rate video streaming using an approach that permits the encoded bit rate to vary subject to a peak bit rate and average bit rate constraints for higher quality streams, while a bottom bit rate stream is encoded to achieve a constant chunk rate. The video encoding system also dynamically decides an encoding resolution for segments of the multiple bit rate video streams that varies with video complexity so as to achieve a better visual experience for multiple bit rate streaming.

WO 2010/088030 A2

TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

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

MULTIPLE BIT RATE VIDEO ENCODING USING VARIABLE BIT RATE AND DYNAMIC RESOLUTION FOR ADAPTIVE VIDEO STREAMING

Background

[001] With the increasing popularity of playing streaming audio and video over networks such as the internet, there is a need for optimizing the data transferred from a server to a client such that the client's experience is maximized even if network conditions during playback are inconsistent. Optimizing the client's experience involves choosing a quality level for encoding the audio and video portions of the video playback such that the video can be transferred and reconstructed uninterrupted while preserving the quality of the video content.

[002] The quality level is generally dictated by the bit rate specified for the encoded audio or video portions of the input stream. A higher bit rate generally indicates that a larger amount of information about the original audio or video is encoded and retained, and therefore a more accurate reproduction of the original input audio or video will be presented during video playback. Conversely, a lower bit rate indicates that less information about the original input audio or video is encoded and retained, and thus a less accurate reproduction of the original audio or video will be presented during video playback.

[003] Generally, the bit rate is specified for encoding each of the audio and video based on several factors. The first factor is the network condition between the server and the client. A network connection that can transfer a high amount of data indicates that a higher bit rate can be specified for the input video that is subsequently transferred over the network connection. The second factor is the desired start-up latency. Start-up latency is the delay that a video playback tool experiences when first starting up due to the large amount of data that has to be received, processed, and buffered. The third factor is the tolerance to glitching. Glitching is when video playback has to stop because data is missing. In most cases any amount of start-up latency or glitching is intolerable, and it is therefore desirable to optimize the bit rate specified such that the start-up latency and the glitching are minimized or eliminated.

[004] Currently available commercial streaming media systems rely on multi bit rate (MBR) coding to perform coding rate control. In MBR coding, source video content is encoded into alternative bit streams at different coding rates and typically stored in the same media file at the server. This then allows the content to be streamed in segments or chunks at varying levels of quality corresponding to different coding rates according to the changing network conditions, typically using bit stream switching between segments.

[005] The currently available multi bit rate video streaming systems use a constant bit rate approach to encoding each alternative video stream. However, a typical video will generally include scenes having a wide variety of visual complexity. However, the constant bit rate approach can not efficiently encode video segments with different quality. The constant bit rate approach unnecessarily spends too many bits for encoding low complexity video segments, and conversely the high complexity scenes are allocated too few bits. Consequently, the constant bit rate approach to encoding the alternative streams results in video quality for internet streaming that is undesirable and inconsistent.

[006] The currently available multi bit rate video streaming systems also have a further requirement for the final display resolution to be fixed. By maintaining a fixed display resolution, the video streams at the multiple bit rates can all be decoded and scaled to this same final display resolution in order to achieve a glitch free video presentation. With the fixed display resolution, the various alternative video streams can have a wide range of bit rates from a few megabits per second to a few kilobits per second. One problem is to match an appropriate video resolution to each video stream bit rate. The currently available multi bit rate video streaming systems use a pre-defined encoding resolution, which again may not be well suited to the varying complexity of the video scenes. For low complexity video, the pre-defined resolution may be too small. For complex video, the pre-defined resolution may be too large.

Summary

[007] The following Detailed Description concerns techniques (implemented via methods, devices and systems) for multiple bit rate video encoding, which are intended to make better use of the available bits with each bit rate so as to achieve generally higher quality video.

[008] According to one technique described herein, a multiple bit rate video encoder encodes a plurality of video streams for multiple bit rate video streaming with an objective of providing a more consistent video quality. For encoding the highest bit rate video stream, the bit rate at which the stream is encoded is allowed to vary subject to certain constraints: a peak bit rate constraint and an average bit rate constraint. For a lowest bit rate stream, the multiple bit rate video encoder encodes the stream with a constant chunk (a given size group of pictures) rate approach. Video streams at intermediate bit rates are encoded at progressively decreasing variable bit rates (subject to decreasing peak and average bit rate constraints).

[009] According to a further technique described herein, the multiple bit rate video encoder also dynamically varies the video resolution of the streams. For each bit rate, the video encoder dynamically decides the resolution based on the video content of a scene (which may comprise one or more groups of pictures) in order to achieve better visual quality. The multiple bit rate video encoder selects a higher video resolution for groups of pictures that have less complex video content, whereas a lower resolution is assigned for groups of pictures that have higher complexity. This dynamic resolution approach allows the multiple bit rate video encoder to achieve a generally better video quality for a given bit rate.

[010] This Summary is provided to introduce a selection of concepts in a simplified form that is further described below in the Detailed Description. This summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter. Additional features and advantages of the invention will be made apparent from the following detailed description of embodiments that proceeds with reference to the accompanying drawings.

Brief Description Of The Drawings

[011] Figure 1 is a block diagram of a video streaming system that provides segmented streaming of video at variable bit rates.

[012] Figure 2 is a block diagram of a generalized implementation of a video encoder for encoding streams at varying bit rates for the video streaming system of Figure 1.

[013] Figure 3 is a process flow chart for a multiple bit rate video encoding system that applies a variable bit rate approach and dynamic resolution approach to encoding video using the video encoder of Figure 2 for streaming by the system of Figure 1

[014] Figure 4 is a graph of a relation between resolution, quantization step size and coded size for an example stream for use in a three point sampling approach to dynamic resolution selection.

[015] Figure 5 is a process flow chart for a dynamic resolution decision by the multiple bit rate video encoding system.

[016] Figure 6 is a block diagram of a generalized operating environment in conjunction with which various described embodiments may be implemented.

Detailed Description

[017] The following detailed description concerns various techniques and systems for video encoding using variable bit rate and dynamic resolution to produce video streams at multiple bit rates for streaming. Although the techniques are described in the context of their application to a multiple bit rate streaming application, the techniques can be applied more broadly to other video encoding applications.

[018] The various techniques and tools described herein may be used independently. Some of the techniques and tools may be used in combination. Various techniques are described below with reference to flowcharts of processing acts. The various processing acts shown in the flowcharts may be consolidated into fewer acts or separated into more acts. For the sake of simplicity, the relation of acts shown in a particular flowchart to acts described elsewhere is often not shown. In many cases, the acts in a flowchart can be reordered.

I. Multi Bit Rate Video Streaming

[019] Figure 1 depicts a generalized block diagram of a system 100 for segmented streaming of multimedia content contained in an indexed video stream file. The indexed file generally divides video of a multimedia program into multiple streaming segments, and contains a number of compressed bit streams representing the video segments at various bit rates. Although the MBR video streams are described as separate coded streams, alternative implementations can have some or all of the MBR video streams encoded as one coded compressed video stream with multiple coding layers. In the system 100, a server 110 (e.g., a server computer system such as a standard HTTP server) provides multimedia content to a client 120 (e.g., a client computer system, such as a laptop or desktop computer, or another type of computing device, such as a PDA or mobile phone) via a network 130 (e.g., the Internet). In the system 100, the server 110 stores programs in an indexed file. The client 120 comprises client-side rate control software and/or hardware.

[020] In one specific example implementation, the server 110 is a standard HTTP server without any specialized streaming capability other than the ability to serve files. Because the server 110 does not support any specialized bit rate selection capability, the client 120 must perform all bit rate selection activities. In this implementation, the client 120 performs all bit rate selection activities. For example, the client 120 can perform rate control using the index information obtained from the server 110 (e.g., alone or in combination with other information, such as client buffer information, network bandwidth,

etc.). However, in other implementations, some or all of the rate-control functions can occur at the server.

[021] In general, the indexed file for multi bit rate streaming can be used by standard HTTP servers to serve multimedia content at multiple bit rates with bit rate selection (rate control) being performed client-side (e.g., exclusively client-side). Clients can perform rate control by first obtaining index information from the server describing the various bit rates available for streaming segments of a program. Based on the index information, and possibly other information (e.g., network bandwidth, buffer information, etc.), the client can decide which bit rate streaming segments to download from the server to provide a desired user experience (e.g., the best user experience possible based on the available bit rates and current network conditions).

[022] Other types of computing devices (e.g., other than traditional HTTP servers) can provide files using the indexed file. For example, a computing device (e.g., a personal computer, server computer, or special-purpose streaming media server) can use the indexed file layout to serve multimedia content using various file serving protocols (e.g., File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), Real Time Streaming Protocol (RTSP), MMS (Microsoft Media Services), etc.).

[023] In order to support bit rate switching, programs are divided into temporal chunks called streaming segments (self-contained units). The server stores each streaming segment at one or more bit rates (e.g., each streaming segment – bit rate combination is a separate streaming segment encoding). Each streaming segment includes one or more available bit rate encodings for a specific track (e.g., a specific audio track, such as an English audio track, or a specific video track) of a program. Clients then determine which bit rate, from the available bit rates (e.g., from the available streaming segment encodings), to download for each streaming segment. For example, a client may obtain a first streaming segment, for a video track, encoded at 250Kb/sec (kilo-bits per second) (from one or more available streaming segment encodings for the first streaming segment), a second streaming segment, for the video track, encoded at 500Kb/sec (from one or more available streaming segment encodings for the second streaming segment), and a third streaming segment, for the video track, encoded at 1Mb/sec (mega-bit per second) (from one or more available streaming segment encodings for the third streaming segment). In the illustrated streaming system 100, each streaming segment contained in the indexed file is encoded by a video encoder at a variable bit rate (VBR) and variable resolution, as described below.

II. Video Encoder Overview

[024] Figure 2 depicts one example of a video encoder 200 that can be used for encoding video for multi bit rate video streaming. The video encoder 200 has inputs 210, 220 for receiving "raw" (uncompressed) frames of video content and also previously calculated motion information for the video content. The video encoder then performs intra-frame coding of reference frames of the video content, and utilizes the motion information to perform inter-frame coding of the predicted frames of the video content. The encoding can be performed according to a known video encoding standard, such as Windows Media Video format, SMPTE 421-M format, MPEG-x format (e.g., MPEG-1, MPEG-2, or MPEG-4), H.26x format (e.g., H.261, H.262, H.263, or H.264), or other format. However, in the case of inter-frame coding, the video encoder can choose to use the pre-calculated motion information for the inter-frame coding of a predicted frame, rather than performing its own motion estimation for the frame. The video encoder encodes the video content into a compressed bitstream provided as output 230. The video encoder may also output the motion information that it used for inter-frame compression of the input video content as motion information output 240 (such as for encoding a lower bit rate video stream for the multiple bit rate video streaming).

[025] Figure 2 is a generalized block diagram showing one example suitable implementation of the video encoder 200 for producing multiple bit rate video streams using variable bit rate and variable resolution encoding for the multiple bit rate video streaming system 100. The video encoder 200 receives a sequence of video pictures (frames) as its raw video content input 210 and produces a compressed bit stream 230 as output.

[026] The video encoder 200 processes video pictures. The term "picture" generally refers to source, coded, or reconstructed image data. For progressive video, a picture is a progressive video frame. For interlaced video, a picture may refer to an interlaced video frame, the top field of the frame, or the bottom field of the frame, depending on context.

[027] The video encoder 200 compresses inter-coded, predicted pictures of the input video and intra-coded pictures of the input video. For the sake of presentation, Figure 2 shows a path for intra-coded frames through the encoder 200 and a path for inter-coded predicted frames. Many of the components of the video encoder 200 are used for compressing both intra-coded content and inter-coded, predicted content. The exact operations performed by those components can vary depending on the type of information being compressed.

[028] In general, within the video encoder 200, an inter-coded, predicted frame (as a picture) is represented in terms of prediction from previously reconstructed content (as one or more other pictures, which are typically referred to as reference pictures or anchors). For example, content at a given time is encoded as a progressive P-frame or B-frame, interlaced P-field or B-field, or interlaced P-frame or B-frame. Within the video encoder 200, a prediction residual is the difference between predicted information and corresponding intra-coded frames.

[029] The input video 110 content on the inter-path is encoded as a predicted picture based on motion information. If certain conditions are met, the video encoder 100 uses the pre-calculated motion information from input 120 (as illustrated by selection switch 256), which can be in the form of a set or sequence of motion vector for macroblocks or other sets of samples of the inter-path video picture with respect to one or more reference pictures. In general, the choice to use the pre-calculated motion information can be based on: first, the availability of pre-calculated motion information; and second, which and whether encoding parameters were changed from the previous calculation of the motion information and the parameters used for the current encoding of the video content. In one example, the video encoder will choose not to use the previously calculated motion information from input 130 if the motion information was calculated for encoding the video content with a different video resolution than that which the video encoder is current encoding.

[030] However, the video encoder 100 can instead choose (again illustrated by selection switch 256) to perform new motion estimation for the inter-path video content 110 with motion estimator 258. The motion estimator 258 estimates motion of macroblocks or other sets of samples of the video picture with respect to one or more reference pictures, which represent reconstructions of previously encoded video content frames. The picture store 264 buffers this reconstructed video content 266 as a reference picture or pictures. When multiple reference pictures are used, the multiple reference pictures can be from different temporal directions or the same temporal direction. The motion estimator 258 outputs motion information 260 such as motion vector information.

[031] The motion compensator 262 applies motion vectors to certain reconstructed video content 266 (stored as reference picture(s)) when forming a motion-compensated current picture 268. The difference (if any) between a block of the motion-compensated picture 268 and corresponding block of the original inter-path video picture is the prediction residual 270 for the block. During later reconstruction of the inter-path video frame (e.g., at a video decoder), reconstructed prediction residuals are added to the motion compensated residual video 268 to obtain reconstructed content closer to the original inter-path video 256. In lossy compression, however, some information is still lost from the original inter-path video. Alternatively, a motion estimator and motion compensator apply another type of motion estimation/compensation.

[032] A frequency transformer 280 converts spatial domain video information into frequency domain (*i.e.*, spectral, transform) data. For block-based video content, the frequency transformer 280 applies a DCT, variant of DCT, or other forward block transform to blocks of the samples or prediction residual data, producing blocks of frequency transform coefficients. The frequency transformer 280 may apply an 8x8, 8x4, 4x8, 4x4 or other size frequency transform.

[033] A quantizer 282 then quantizes the blocks of transform coefficients. The quantizer 282 applies non-uniform, scalar quantization to the spectral data with a step size that varies spatially on a picture-by-picture basis, macroblock-by-macroblock basis or other basis. Additionally, in some cases the quantizer varies quantization across color channels of the inter-layer residual video picture. The quantizer 282 can also apply another type of quantization, for example, a uniform or adaptive quantization for at least some spectral data coefficients, or directly quantizes spatial domain data in an encoder system that does not use frequency transformations.

[034] When reconstructed video content is needed for subsequent motion estimation/compensation of an inter-path video picture, an inverse quantizer 290 performs inverse quantization on the quantized spectral data coefficients. An inverse frequency transformer 292 performs an inverse frequency transform, producing blocks of reconstructed prediction residuals (for predicted inter-path residual video content) or samples (for intra-path residual video content). If the residual video content 256 was motion-compensation predicted, the reconstructed prediction residuals are added to the motion-compensated predictors 268 to form the reconstructed residual video. The picture store 264 buffers the reconstructed residual video for use in subsequent motion-compensated prediction.

[035] The entropy coder 284 compresses the output of the quantizer 282 as well as certain side information (*e.g.*, quantization parameter values) Typical entropy coding techniques include arithmetic coding, differential coding, Huffman coding, run length coding, LZ coding, dictionary coding, and combinations of the above. The entropy coder

284 typically uses different coding techniques for different kinds of information, and can choose from among multiple code tables within a particular coding technique.

[036] When the video encoder 240 performs intra-compression of the intra-path video content, the encoder intra-compresses it as an intra-coded picture, without motion compensation. The video 256 is provided directly to the frequency transformer 280, quantizer 282, and entropy coder 284 and output as encoded video. A reconstructed version of the intra-coded video can be buffered for use in subsequent motion compensation of other inter-path video.

[037] A controller 294 receives inputs from various modules such as the motion estimator 258, frequency transformer 280, quantizer 282, inverse quantizer 290, and entropy coder 284. The controller 294 evaluates intermediate results during encoding, for example, setting quantization step sizes and performing rate-distortion analysis. The controller 294 works with other modules to set and change coding parameters during encoding. When the controller 294 evaluates different coding parameter choices, the controller 294 may iteratively perform certain stages to evaluate different parameter settings, or the controller 294 may jointly evaluate different coding parameters. The tree of coding parameter decisions to be evaluated, and the timing of corresponding encoding, depends on implementation. In some embodiments, the controller 294 also receives input from an encoding session wizard interface, other encoder application interface, or other source to designate video to be encoded using specific rules.

III. Variable Bit Rate Encoding of MBR Streams

[038] For the multiple bit rate video streaming system 100 (Figure 1, a multiple bit rate video encoding system separately encodes the input video as a set of compressed video streams with successively decreasing overall bit rates. Although described herein as encoding separate individual MBR video streams, an alternative implementation of the MBR video streaming system and encoding system can encode one or more of the MBR video streams as a compressed bitstream having multiple separable coding layers. The multiple bit rate video encoding system includes an MBR encoding engine (not shown) that drives the video encoder 200 (Figure 2) to encode the input video with varying encoding parameters according to a multiple bit rate encoding process (as shown in Figure 3) that implements a variable bit rate and dynamic resolution approach described in this and the following section. The MBR encoding engine can provides a user interface or console for receiving user input to configure parameters for the MBR video stream encoding (or alternatively an application programming interface to receive such input from a caller application), such as the number of streams, and other parameters mentioned below.

[039] In contrast to other currently available multiple bit rate video streaming systems (which use a constant bit rate approach to encoding the multiple video streams), the MBR encoding system for the multiple bit rate video stream system 100 aims at providing a constant or consistent quality for each video stream. For the top MBR video stream (generally having highest overall bit rate), the video encoder 200 encodes the video stream with a varying bit rate constrained to fall under a specified peak bit rate while satisfying a specified average bit rate. For the bottom MBR stream (generally having the lowest bit rate of the set), the video encoder uses a constant chunk rate approach. In the context of the multiple bit rate video streaming system, the term chunk refers to a group of pictures (GOP) into which the video stream are segmented, and define the level of granularity at which the video streaming system may switch playing individual segments between video streams. The constant chunk rate approach enables the video streaming system to guarantee predictability of streaming, in that when the lowest bit rate or quality video stream is streamed, the client will receive the chunk amount of pictures at the constant rate so as to maintain minimum quality continuous playing of the video.

[040] In between the lowest and highest overall bit rate streams, the video encoder encodes one or more intermediate video streams also using variable bit rates of coding within the constraints of a peak bit rate and average bit rate that aim to maintain a constant video quality. The peak and average bit rate constraints of the intermediate video streams can be specified to decrease progressively in a proportional, logarithmic or other decreasing manner. For example, the average bit rate of the intermediate stream can decrease proportionally to be ¾, ½, and ¼ that of the average bit rate constraint of the highest bit rate video stream. In this way, the video streaming system 100 is able to provide an instant start and swift video switching from a guaranteed low constant chunk rate up to a highest quality variable rate bit stream. The peak and average bit rates, as well as the constant chunk rate are encoding parameters that can be configured by the user. These parameters can be configured explicitly by the user, or calculated by the MBR encoding system engine based on more generalized parameters input by the user. For example, the MBR encoding engine can have an automatic mode where the user (or a caller application) simply specifies the minimal and maximal target bit rates and a number of video streams or layers. The engine in this automatic mode then calculates all the intermediate bit rate constraints (peak and average) in a uniform, logarithmic or other distribution space.

[041] With reference now to Figure 3, the MBR encoding system encodes the set of MBR video streams with a process 300 that uses a two pass encoding approach. This process includes an analysis pass and an encoding pass. The goal of the analysis pass is to find the scene complexity of the video content based on the encoding configurations as well as the input video source material itself. Once this information is extracted in the analysis pass, the following encoding pass then generates the set of MBR video streams.

[042] The MBR encoding process 300 begins with an initialization step 310. In this step, the MBR encoding process determines the parameters for the encoding from user input, including number of MBR video streams, peak and average bit rate constraints for the streams, and the constant chunk rate of the lowest quality MBR video stream, and segment parameters, among others.

[043] The analysis pass of the MBR encoding process 300 includes actions 311-314. In the analysis pass, the MBR encoding engine analyzes the input source video frame by frame. The analysis includes a number of different tasks including scene change detection, segmenting a video sequence between scene change boundaries into group of picture segments, and video frame complexity measurements. Based on the scene change detection, the MBR encoding engine marks boundaries at which scene changes occur during the video. Between marked boundaries of a video sequence (sequence mark-in and sequence mark-out positions), the MBR encoding process 300 further determines a total number of group of pictures segments in which to divide the video sequence within user-specified constraints (such as a specified average GOP length and maximum allowed GOP length within a scene) and sets boundaries of each group of pictures. Once the GOP boundaries are defined, the total numbers of frames within each GOP is calculated by the MBR encoding engine. The MBR encoding engine also calculates a set of three texture measurements per frame of each group of pictures, which are used in the variable resolution encoding described in the next section. The three texture measurements include a frame global texture, frame horizontal texture and frame vertical texture measurement. The MBR engine writes these analysis pass results (the scene and GOP boundaries, and the texture measurements) into a log file, as indicated at action 314.

[044] For the encoding pass (actions 315-324), the MBR engine applies the results of the analysis pass to encode the MBR video streams using the video encoder 200 (Figure 2). The MBR engine causes the video encoder 200 to encode each segment (action 317) for all the MBR streams (action 318). For each segment of an MBR video stream, the MBR encoding engine controls the encoding parameters of the video encoder 200 in an attempt to hit user specified targets for an average bit rate. If the source video content is too complex to be encoded at the targeted bit rate, the MBR encoding engine starts to increase the quantization step size in order to achieve a better visual quality. The top or best quality video stream is encoded subject to a maximal peak bit rate constraint meeting the user specified MBR encoding parameters. For the bottom or lowest quality video stream, the MBR engine controls the video encoder to produce an encoding of the video stream to meet the constant chunk rate constraint, which helps guarantee client side predictability for playing the streamed video. In intermediate video streams, the MBR engine causes the video encoder to encode the source video with encoding parameters to produce the video stream a variable bit rate falling within maximal peak and average bit rate constraints for the respective intermediate stream.

[045] As a result of the encoding pass, the MBR engine then outputs compressed video bit streams for the set of MBR streams that are produced using the video encoder, as well as a log file. With the variable bit rate approach of this MBR encoding process 300, the MBR engine produces a set of MBR video streams that decreases evenly from a top to bottom quality stream for each GOP. With this set of MBR video streams, the MBR system 100 (Figure 1) can deliver a desired constant or consistent visual quality for the video sequence according to the available connection bandwidth.

IV. Variable Resolution Encoding of MBR Streams

[046] The MBR encoding engine also applies a technique that dynamically varies resolution of encoding for each of the MBR video streams. For each video stream ranging from the top to bottom of the MBR video streams, the MBR encoding engine dynamically decides the resolution for encoding each video GOP to produce a better visual quality. For each video stream, the MBR encoding engine assigns a higher resolution to a low complexity GOP (or segment), while a more complex GOP (or segment) is assigned a lower resolution of encoding.

[047] In the example implementation, the MBR encoding engine applies the decision to dynamically resize each GOP at scene boundaries of the video. This avoids introducing any undesirable visual effects that resizing video resolution in the middle of a video scene might produce. For example, in a scene featuring a “talking head,” varying the video resolution mid-scene could introduce a noticeable popping or pulsing as the detail edges and features in the scene sharpen or soften along with the resolution change. Accordingly, the MBR encoding engine performs the below described process for the GOP or GOPs of a scene (e.g., for the first GOP after a scene change boundary identified in the analysis phase described above).

[048] In one example implementation of the dynamic resolution encoding, the MBR encoding engine uses a three-point sampling approach to make the dynamic resolution decision. Each sampling point represents the result (in terms of actual encoded bit rate or size) from encoding the GOP using three different pairs of video resolution and quantization step sizes. With these three sampling point results, the MBR engine establishes a model of the relation between resolution, quantization step size and coded size, which relation is illustrated graphically in Figure 4. From this dynamically extracted model for the video sequence, the MBR encoding engine can then decide a resolution for each targeted bit rate of the MBR video streams. In alternative implementations, the MBR encoding engine can use more sampling points to establish the model. However, the three sampling point approach is found to be sufficient to establish the model while remaining most practical for purposes of encoding speed.

[049] Figure 5 illustrates a process 500 performed by the MBR video encoding engine for making the dynamic resolution decision for a GOP of each video stream. The process 500 begins with obtaining the three sample points of encoding results (action 510). The MBR encoding engine controls the video encoder 200 to encode the segment or GOP with three parameter combinations for resolution and quantization step size. The initial sample resolution may be chosen based on a desired display resolution for the video. The initial sample quantization step size can be chosen depending on the particular codec standard used by the video encoder. For example, in the case of the SMPTE 421-M video codec, the initial sample quantization step size may be chosen to be 4. In the case of that the video encoder uses the H.264 standard, an appropriate initial sample quantization step size may be 28. However, other initial sample quantization step sizes and resolutions can instead be selected.

[050] In the illustrated model, the MBR video encoding engine performs the encoding for an initial sample resolution and quantization step size parameter pair (R, Q_p), as well as at one fourth of the initial sample resolution (i.e., (R/4, Q_p)) and at twice the initial sample quantization step size (i.e., (R, Q_p*2)). Alternatively, other parameter pairs for the sample points can be used, such as at half resolution, four times the quantization step size, etc. The MBR video encoding engine observes the encoded bit sizes (S_1, S_2, and S_3) that result from encoding the GOP of the video stream with the three resolution and quantization step size parameter pairs.

[051] In a next action 511, the MBR engine establishes two linear models: one for the relation between quantization step size and encoded size (labeled GraphQ_pS in the diagram of Figure 4), and one for the relation between resolution and encoded size (GraphRS). The relation between quantization step size and encoded size is determined by the encoded sizes that result from the two sample points where the quantization step size is varied while the resolution is held constant, while conversely the relation between resolution and encoded size is determined from the two sample points which vary the resolution while quantization step size remains constant.

[052] At action 512, the MBR engine uses the relation of encoded size to quantization step size to find the quantization step size that yields the encoded size corresponding to the desired bit rate. This is the modeled result quantization step size (labeled Q_p') at the full sampling resolution R that should yield the target bit rate for the GOP of the video stream.

[053] The MBR engine then compares the modeled result quantization step size to an empirically determined threshold (determined from experiments measuring video texture over a wide range of video content). If the modeled result quantization step size is smaller than the threshold, then the MBR engine decides to use the full sample resolution and modeled result quantization step size, i.e., (R, Qp') at action 514.

[054] More specifically, the MBR engine determines the appropriate quantization step threshold based on the per frame texture measurements made during the analysis phase (discussed above) for the input video content. The MBR engine calculates the texture measurements for the GOP by averaging the frame texture measurements for all frames in the GOP. This produces GOP global texture, GOP horizontal texture and GOP vertical texture measurements. Of these, the GOP global texture measurement determines the quantization step size threshold that controls when to resize video resolution. From experimental results over a broad range of video content (including sports, television, movies, etc.), it has been determined that a quantization step size threshold of Qp equal to 12 (for video encoding with the SMPTE 421M standard) is suitable for video with a typical GOP global texture measurement. In other words, if the modeled result quantization step size Qp‘ is over 12, then the MBR encoder should resize to a lower video resolution in order to encode at a lower Qp. However, in an example implementation, the MBR encoder can further vary the quantization step size threshold for resizing depending on the overall global texture measurement for the video. The MBR encoder has established a linear relationship between global texture and the quantization step size threshold for resizing. For video having a low overall global texture, a lower quantization step size threshold is expected. This allows the MBR encoder to be more aggressive in resizing down the video resolution of video content having a lot of smooth regions (for which resizing to a lower resolution would tend not to produce artifacts). Whereas, for video with high global texture, the MBR encoder expects a higher quantization step size threshold for resizing. Such higher threshold makes the MBR encoder more careful in resizing down video resolution of frames that have a lot of detail, so as to avoid smoothing of detailed regions of those frames. In alternative implementations, the quantization step size threshold can be established at other quantization step sizes, such as for use with other video encoding standard, or to achieve a desired degree of aggressiveness/caution in resizing the video resolution.

[055] On the other hand at action 515, if the modeled result is larger than the threshold defined by the video texture, the MBR engine instead uses the relation between encoded size and resolution (GraphRS) to find a modeled result resolution (R') that yields the encoded size corresponding to the target bit rate of the video stream.

[056] The MBR engine further uses the GOP average horizontal and vertical texture measurements to control how much to resize the video resolution in each direction. The MBR engine calculates a ratio of the GOP horizontal and vertical texture measurements. Once it is determined to resize the resolution (action 514), the MBR engine calculates a particular resize amount according to the GraphRS relation. For example, the MBR engine may determine to resize by half the initial resolution. The MBR engine then determines how to distribute the resize amount in the vertical and horizontal directions based on the ratio of GOP horizontal and vertical texture measurement. In particular, if there is a large discrepancy or delta between horizontal and vertical texture measurements (i.e., the ratio is non-unity), the MBR engine distributes the resizing to apply more resizing in the lower detail direction than is applied to the higher detail direction. For example, when the ratio is two, then the MBR engine would resize in the vertical direction twice as much as the horizontal direction. Otherwise, if the delta between the horizontal and vertical texture measurements for the GOP is low (the ratio is near unity), then the MBR engine resizes the resolution equally between the directions.

[057] The MBR engine at action 516 then uses the relations between quantization step size and encoded size (GraphQpS) and between resolution and encoded size (Graph RS) as well as the target bit rate of the respective video stream to establish a relation (GraphQpR shown at top left of Figure 4) between resolution and quantization step size for the particular target bit rate.

[058] At action 517, the MBR engine then uses the relation (GraphQpR) established in action 516 to find a modeled result of the quantization step size (Qp') for the modeled result resolution R' decided at action 515. The MBR engine then decides to encode this GOP of this video stream at the modeled result quantization step size and resolution (R', Qp').

[059] By use of this dynamic resolution approach, the MBR encoding system is able to assign a larger encoding resolution to less complex video segments (or GOP), which maintains more visual detail. On the other hand, more complex video segments (or GOP) are assigned a smaller resolution that reduces visual artifacts. This approach has been found to provide a better visual experience for multiple bit rate streaming.

V. Representative Computing Environment

[060] Figure 6 illustrates a generalized example of a suitable computing environment 600 in which described embodiments, techniques, and technologies may be implemented. The computing environment 600 is not intended to suggest any limitation as to scope of use or functionality of the technology, as the technology may be implemented in diverse general-purpose or special-purpose computing environments. For example, the disclosed technology may be implemented with other computer system configurations, including hand held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The disclosed technology may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

[061] With reference to Figure 6, the computing environment 600 includes at least one central processing unit 610 and memory 620. For the multi core motion information precalculation discussed above, the computer includes a multi core CPU having plural CPU cores. In Figure 6, this most basic configuration 630 is included within a dashed line. The central processing unit 610 executes computer-executable instructions and may be a real or a virtual processor. In a multi-processing system, multiple processing units execute computer-executable instructions to increase processing power and as such, multiple processors can be running simultaneously. The memory 620 may be volatile memory (e.g., registers, cache, RAM), non-volatile memory (e.g., ROM, EEPROM, flash memory, etc.), or some combination of the two. The memory 620 stores software 680 that can, for example, implement the technologies described herein. A computing environment may have additional features. For example, the computing environment 600 includes storage 640, one or more input devices 650, one or more output devices 660, and one or more communication connections 670. An interconnection mechanism (not shown) such as a bus, a controller, or a network, interconnects the components of the computing environment 600. Typically, operating system software (not shown) provides an operating environment for other software executing in the computing environment 600, and coordinates activities of the components of the computing environment 600.

[062] The storage 640 may be removable or non-removable, and includes magnetic disks, magnetic tapes or cassettes, CD-ROMs, CD-RWs, DVDs, or any other medium which can be used to store information and which can be accessed within the computing environment 600. The storage 640 stores instructions for the software 680, which can implement technologies described herein.

[063] The input device(s) 650 may be a touch input device, such as a keyboard, keypad, mouse, pen, or trackball, a voice input device, a scanning device, or another device, that provides input to the computing environment 600. For audio, the input device(s) 650 may be a sound card or similar device that accepts audio input in analog or digital form, or a CD-ROM reader that provides audio samples to the computing environment 600. The output device(s) 660 may be a display, printer, speaker, CD-writer, or another device that provides output from the computing environment 600.

[064] The communication connection(s) 670 enable communication over a communication medium (e.g., a connecting network) to another computing entity. The communication medium conveys information such as computer-executable instructions, compressed graphics information, or other data in a modulated data signal.

[065] Computer-readable media are any available media that can be accessed within a computing environment 600. By way of example, and not limitation, with the computing environment 600, computer-readable media include memory 620, storage 640, the communication medium, and combinations of any of the above. As should be readily understood, the term computer-readable storage media includes the media for data storage such as memory 620 and storage 640, and not simply transmission media such as modulated data signals.

[066] Any of the methods described herein can be performed via one or more computer-readable media (e.g., storage or other tangible media) comprising (e.g., having or storing) computer-executable instructions for performing (e.g., causing a computing device, audio and/or video processing device, or computer to perform) such methods. Operation can be fully automatic, semi-automatic, or involve manual intervention.

[067] Having described and illustrated the principles of our innovations in the detailed description and accompanying drawings, it will be recognized that the various embodiments can be modified in arrangement and detail without departing from such principles. It should be understood that the programs, processes, or methods described herein are not related or limited to any particular type of computing environment, unless indicated otherwise. Various types of general purpose or specialized computing environments may be used with or perform operations in accordance with the teachings described herein. Elements of embodiments shown in software may be implemented in hardware and vice versa.

[068] In view of the many possible embodiments to which the principles of our invention may be applied, we claim as our invention all such embodiments as may come within the scope and spirit of the following claims and equivalents thereto.

We claim:

1. A method of processing input video for variable bit rate encoding of segmented compressed video streams for multi bit rate video streaming, the method comprising:

receiving input of raw video content (210) to be encoded for multi bit rate video streaming;

encoding (315-324) the raw video content in segments comprising groups of pictures as a plurality of compressed video streams ranging in video quality from a top video stream to a bottom video stream, wherein said encoding comprises:

for the top video stream, encoding the raw video content at a variable bit rate subject to peak bit rate and average bit rate constraints;

for the bottom video stream, encoding the raw video content at a constant chunk rate, wherein the constant chunk rate is a constant number of pictures per unit time; and

producing an encoded set of the video streams (230) segmented into the groups of pictures.

2. The method of claim 1 wherein said encoding the top video stream comprises encoding with a variable bit rate approach targeting a designated average bit rate and constrained to remain under a peak bit rate.

3. The method of claim 2 wherein said encoding the raw video content further comprises, for one or more intermediate video streams, also encoding the raw video content at a variable bit rate subject to peak bit rate and average bit rate constraints, wherein the peak bit rate and average bit rate constraints of the top video stream and intermediate video streams decrease progressively.

4. The method of claim 3 further comprising:

receiving input designating maximum and minimum bit rates, as well as a number of compressed video streams to be encoded for multi bit rate video streaming;

selecting peak bit rate and average bit rate for the top video stream and intermediate video streams to progressively decrease.

5. The method of claim 1 further comprising:

selectively streaming the segments of the encoded video streams over a network to a client video playing device, wherein the streamed segments are selected from among the encoded video streams according to client-side rate control decisions based on network conditions.

6. A method of processing input video for dynamic resolution encoding of segmented compressed video streams for multi bit rate video streaming, the method comprising:

receiving input of raw video content (210) to be encoded for multi bit rate video streaming;

encoding (315-324) the raw video content in segments comprising groups of pictures as a plurality of compressed video streams ranging in video quality from a top video stream to a bottom video stream, wherein said encoding comprises:

for a group of pictures of a video stream to be encoded, obtaining (510) at least three encoding samples for varying resolution and quantization step size;

establishing (511) a model relating resolution, quantization step size and resulting encoded size for the group of pictures;

performing a texture measurement on the group of pictures;

determining (512) a quantization threshold for resizing resolution based on the texture measurement;

dynamically deciding (514, 515, 517) a video resolution and quantization step size for encoding the group of pictures based on the model and quantization threshold for resizing; and

encoding the group of pictures with the dynamically decided video resolution and quantization step size.

7. The method of claim 6 further comprises:

detecting scene changes in the raw video content; and

wherein said dynamically deciding video resolution comprises applying dynamic video resolution changes at scene change boundaries between groups of pictures.

8. The method of claim 6 wherein said establishing the model comprises:

establishing a linear model for a relation of quantization step size to encoded size for a given video resolution based on at least two out of the at least three encoding samples of the group of pictures where the encoded size is sampled at the given video resolution and varying quantization step size.

9. The method of claim 8 wherein said dynamically deciding the video resolution comprises:

determining a quantization step size that yields an encoded size corresponding to a desired bit rate for the group of pictures when encoding at the given video resolution according to the established linear model relating quantization step size to encoded size for the given video resolution;

comparing the determined quantization step size to the quantization threshold for resizing resolution;

if the comparison is not indicative of resizing resolution, deciding to encode the group of pictures using the given video resolution and determined quantization step size; and

otherwise, deciding to resize the video resolution for encoding the group of pictures.

10. The method of claim 9 wherein said establishing the model further comprises:

establishing a linear model for a relation of video resolution to encoded size for a given quantization step size based on at least two out of the at least three encoding samples of the group of pictures where the encoded size is sampled at the given quantization step size and varying video resolution.

11. The method of claim 10 wherein said dynamically deciding the video resolution comprises:

in the case where the video resolution for encoding the group of pictures is to be resized, determining a resized video resolution that yields an encoded size corresponding to a desired bit rate for the group of pictures according to the established linear model relating video resolution to encoded size for the given quantization step size.

12. The method of claim 11 wherein, in the case where the video resolution for encoding the group of pictures is to be resized, said dynamically deciding the quantization step size comprises:

establishing a linear model relating quantization step size to resolution at the desired bit rate for the group of pictures based on the at least three encoding samples; and

deciding the quantization step size for the resized video resolution according to the linear model relating quantization step size to resolution.

13. A video processing system for encoding compressed video streams for multiple bit rate video streaming, the system comprising:

a memory (620) for storing raw video content to be encoded;

a computer processing means (610) for encoding (315-324) the raw video content using a variable bit rate and dynamic resolution approach in segments comprising groups of pictures as a plurality of compressed video streams, wherein said encoding by the computer processing means comprises:

analyzing the raw video content to determine scene change boundaries in the video content, divide the video content between said scene change boundaries in the segments comprising groups of pictures, and determine a measurement of video complexity of the segments;

dynamically deciding (500, 514, 515, 517) encoding parameters for variable bit rate encoding of the raw video content of each segment into one or more higher video streams each having a target average bit rate and a peak bit rate constraint that yield a progression of decreasing coded bit rates of the video streams, and for encoding the raw video content of each segment into a bottom video stream at a constant chunk rate, wherein the encoding parameters comprise at least video resolution; and

encoding the higher and bottom video streams using the dynamically decided encoding parameters.

14. The video processing system of claim 13 wherein said dynamically deciding the video resolution for a video stream segment by the computer processing means comprises:

obtaining at least three samples of encoding a segment for varying resolution and quantization step size;

establishing a model of a relation between quantization step size, video resolution and encoded size; and

deciding the video resolution for the segment based on the established model and measurement of video complexity of the segment.

15. The video processing system of claim 13 further comprising said computer processing means automatically selecting the peak bit rate and average bit rate for the upper video streams to have a desired numeric distribution between designated maximum and minimum bit rates for a designated number of the video streams.

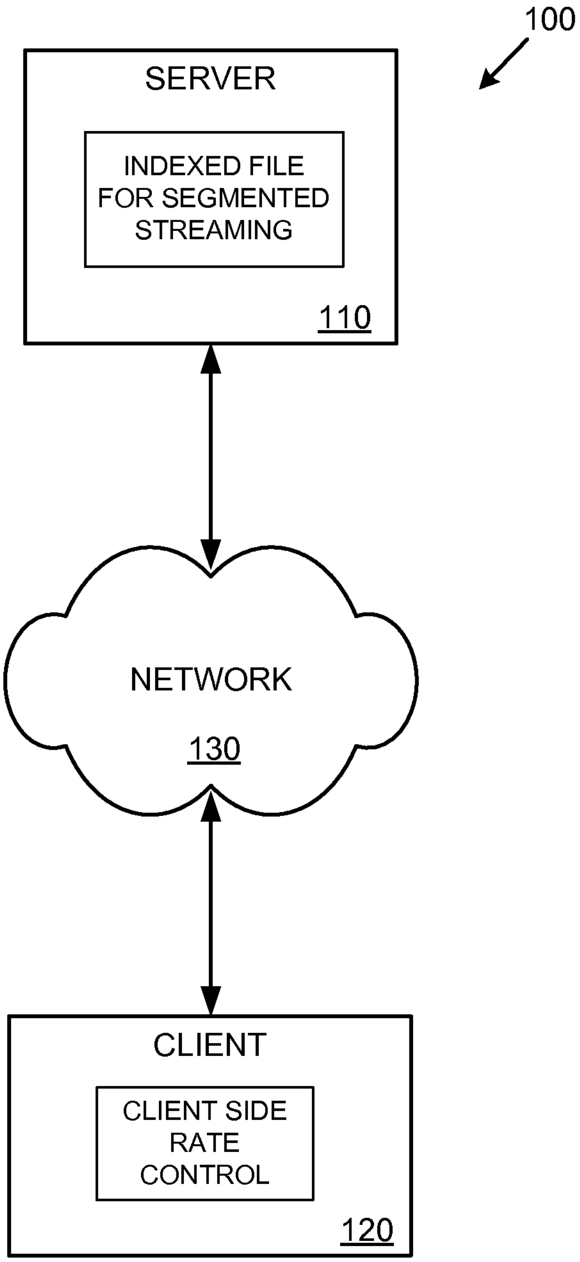
100
SERVER
INDEXED FILE FOR SEGMENTED STREAMING
110
NETWORK
130
CLIENT
CLIENT SIDE RATE CONTROL
120

FIG. 1

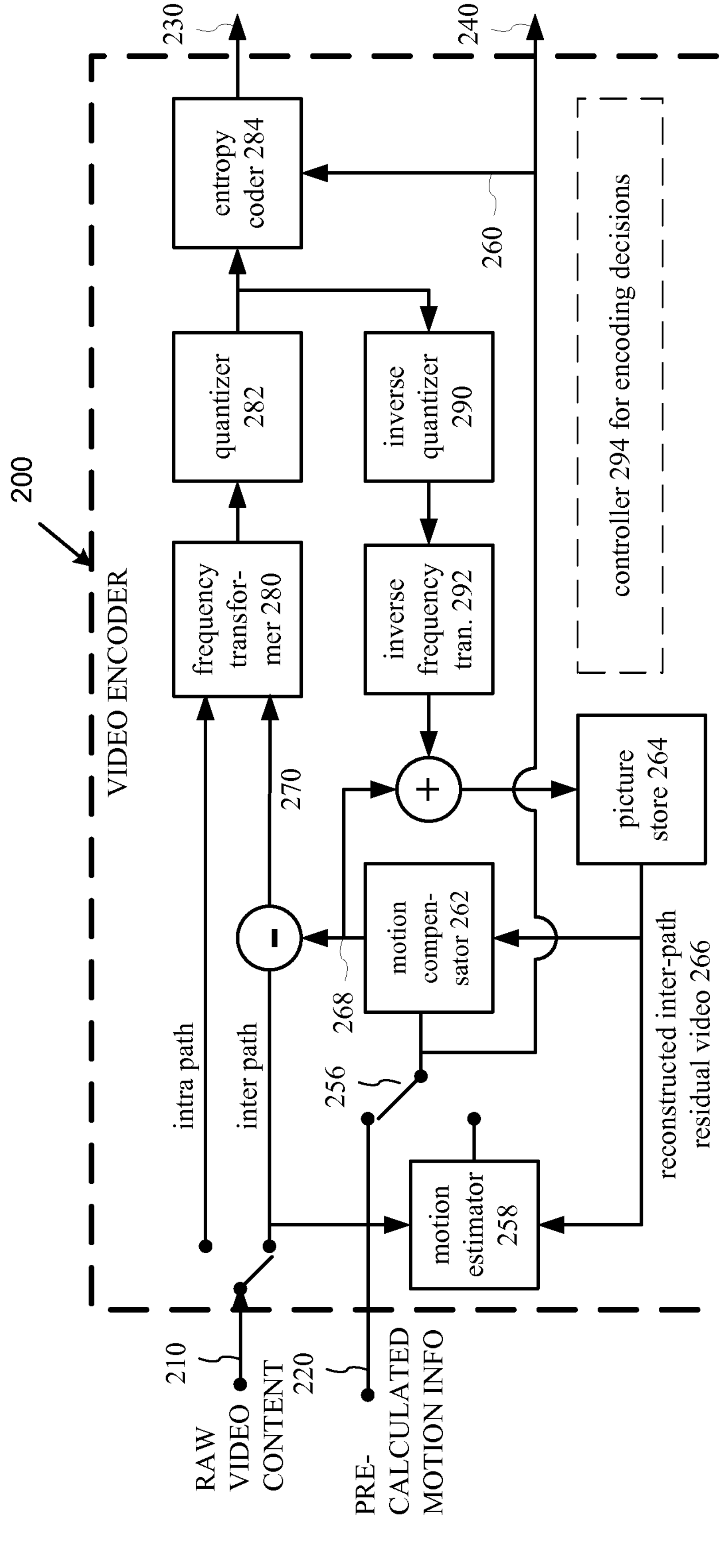
200
VIDEO ENCODER
RAW VIDEO CONTENT
210
PRE-CALCULATED MOTION INFO
220
intra path
inter path
270
frequency transformer 280
quantizer 282
entropy coder 284
230
240
260
inverse quantizer 290
inverse frequency tran. 292
+
-
motion compensator 262
268
256
motion estimator 258
picture store 264
reconstructed inter-path residual video 266
controller 294 for encoding decisions

FIG. 2

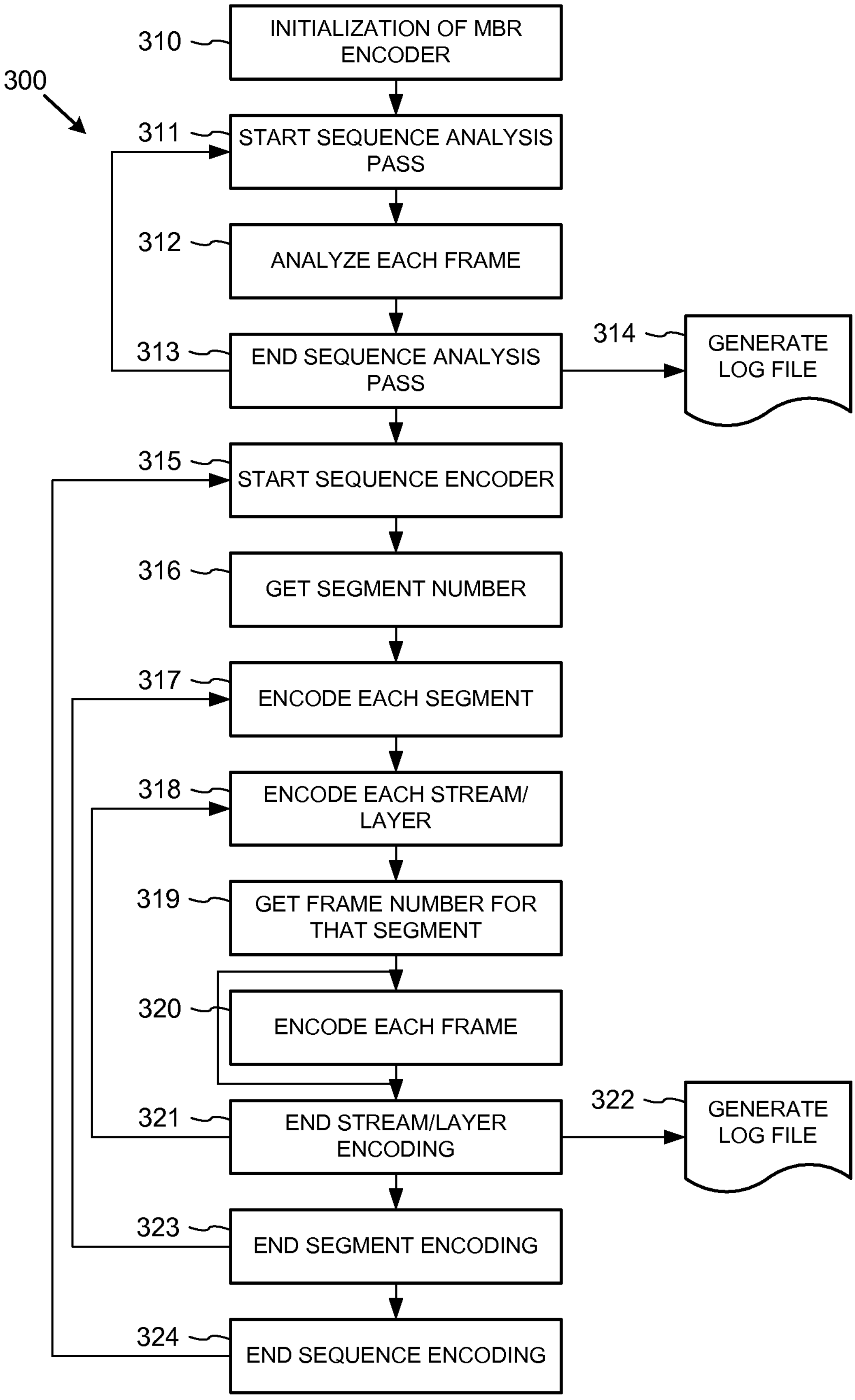
300
310
INITIALIZATION OF MBR ENCODER
311
START SEQUENCE ANALYSIS PASS
312
ANALYZE EACH FRAME
313
END SEQUENCE ANALYSIS PASS
314
GENERATE LOG FILE
315
START SEQUENCE ENCODER
316
GET SEGMENT NUMBER
317
ENCODE EACH SEGMENT
318
ENCODE EACH STREAM/ LAYER
319
GET FRAME NUMBER FOR THAT SEGMENT
320
ENCODE EACH FRAME
321
END STREAM/LAYER ENCODING
322
GENERATE LOG FILE
323
END SEGMENT ENCODING
324
END SEQUENCE ENCODING

FIG. 3

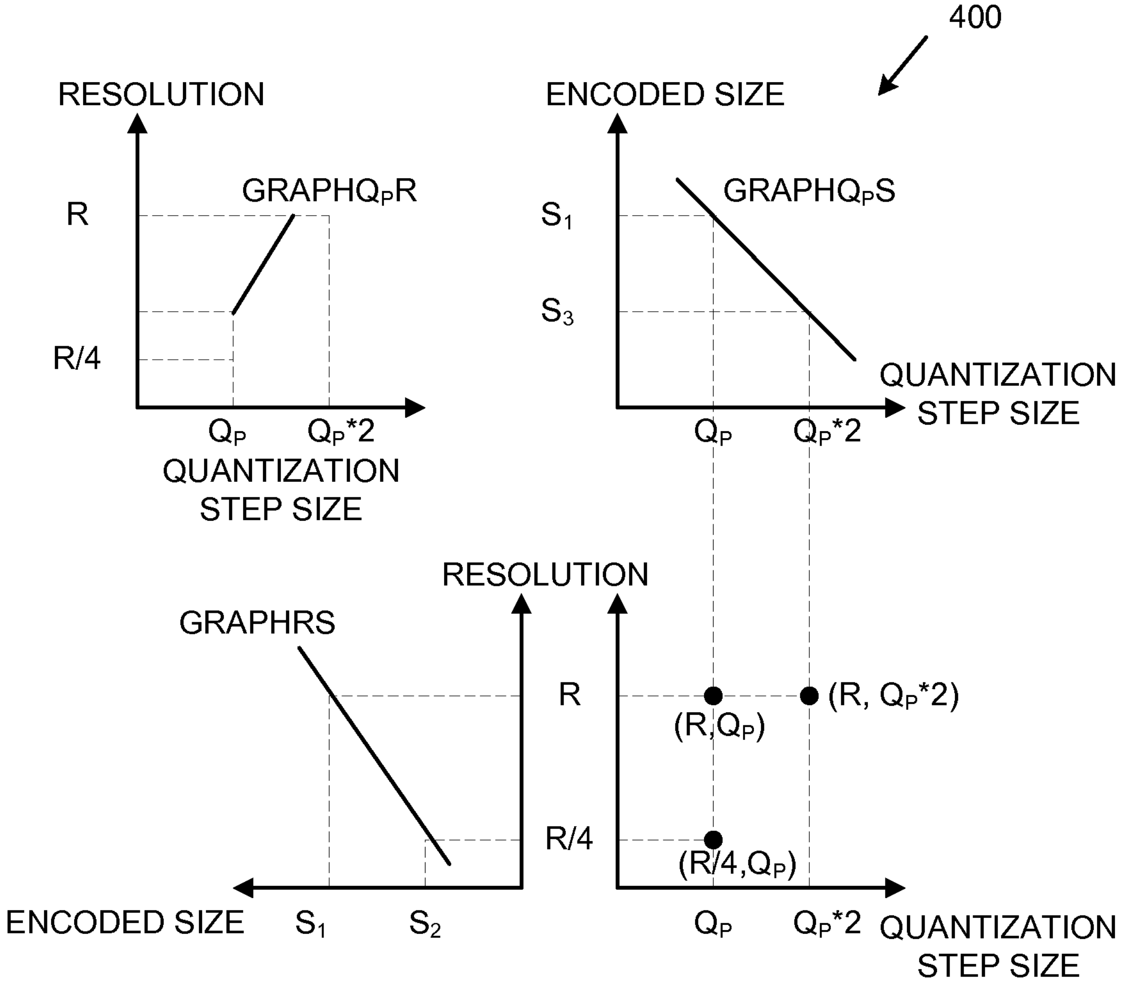
400
RESOLUTION
R
GRAPHQ$_P$R
R/4
Q$_P$
Q$_P$*2
QUANTIZATION STEP SIZE
ENCODED SIZE
S$_1$
GRAPHQ$_P$S
S$_3$
QUANTIZATION STEP SIZE
Q$_P$
Q$_P$*2
RESOLUTION
GRAPHRS
R
R/4
ENCODED SIZE
S$_1$
S$_2$
(R,Q$_P$)
(R, Q$_P$*2)
(R/4,Q$_P$)
Q$_P$
Q$_P$*2
QUANTIZATION STEP SIZE

FIG. 4

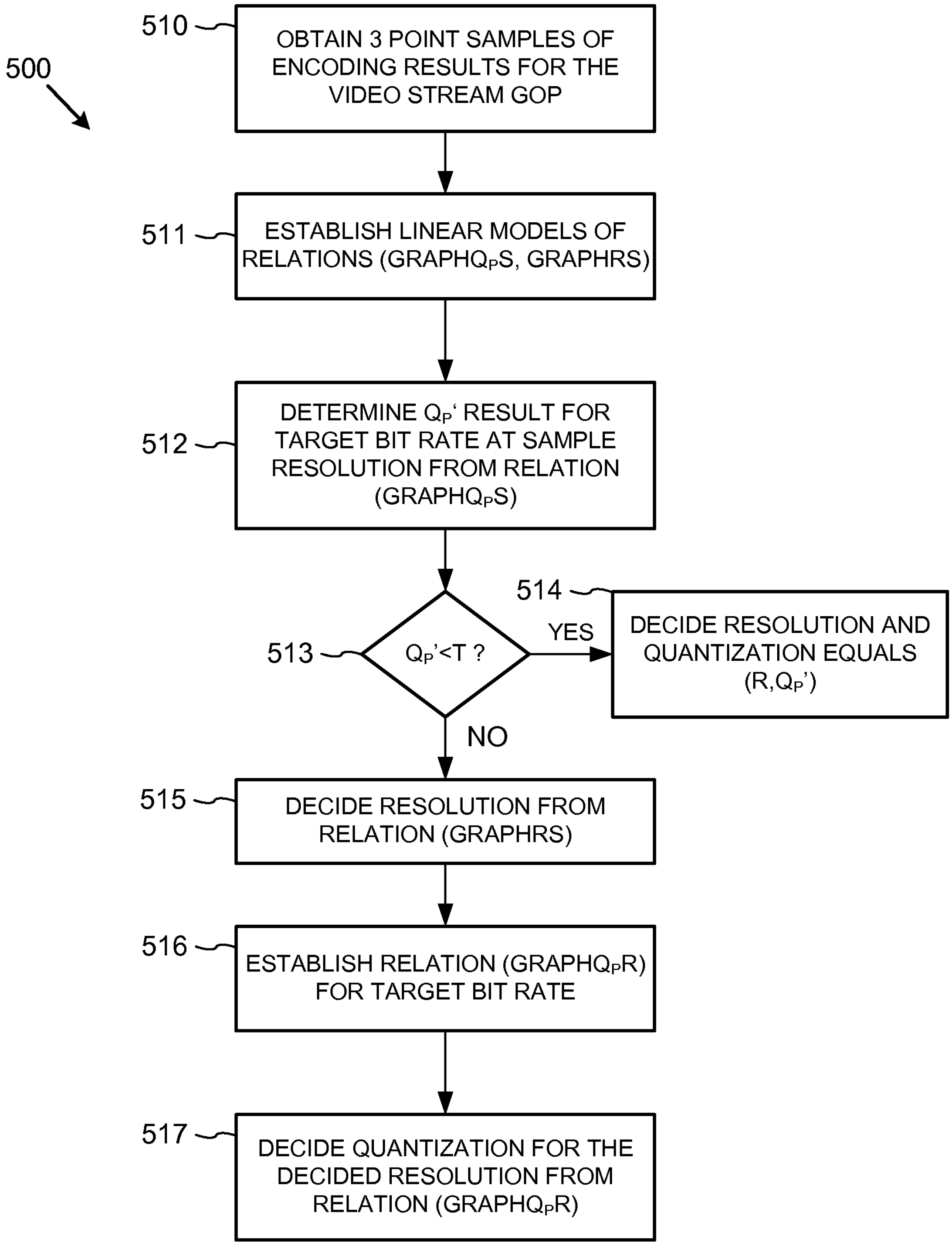
500
510
OBTAIN 3 POINT SAMPLES OF ENCODING RESULTS FOR THE VIDEO STREAM GOP
511
ESTABLISH LINEAR MODELS OF RELATIONS (GRAPHQ$_P$S, GRAPHRS)
512
DETERMINE Q$_P$' RESULT FOR TARGET BIT RATE AT SAMPLE RESOLUTION FROM RELATION (GRAPHQ$_P$S)
513
Q$_P$'<T ?
YES
514
DECIDE RESOLUTION AND QUANTIZATION EQUALS (R,Q$_P$')
NO
515
DECIDE RESOLUTION FROM RELATION (GRAPHRS)
516
ESTABLISH RELATION (GRAPHQ$_P$R) FOR TARGET BIT RATE
517
DECIDE QUANTIZATION FOR THE DECIDED RESOLUTION FROM RELATION (GRAPHQ$_P$R)

FIG. 5

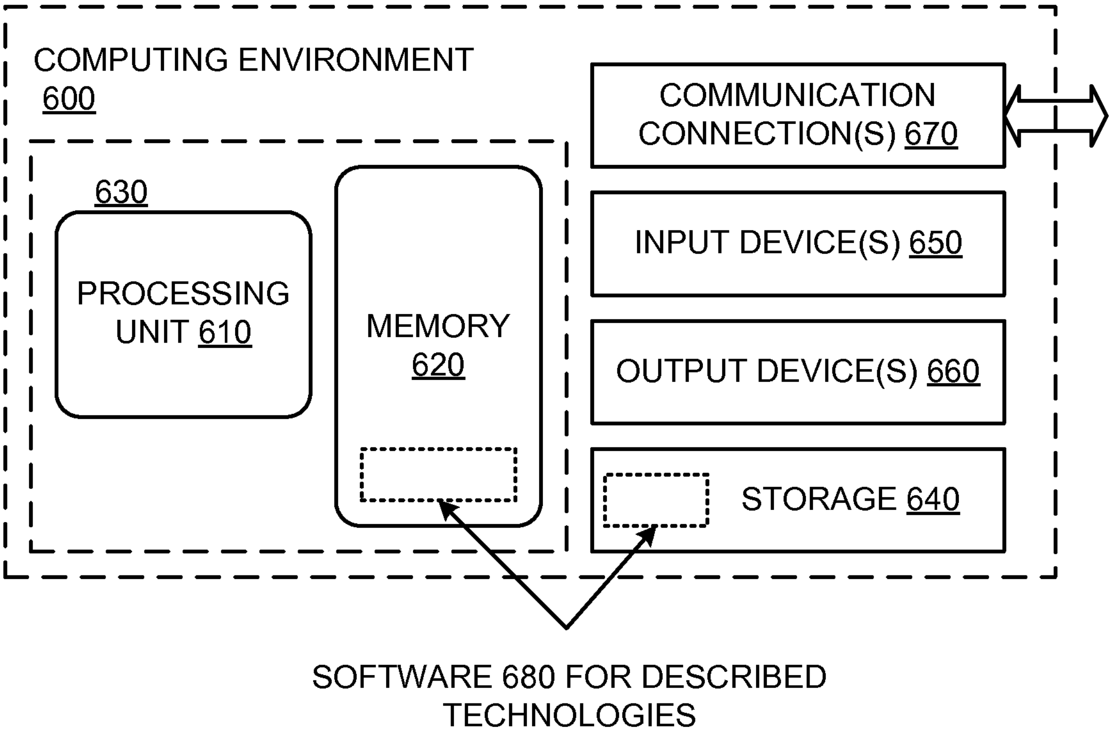
COMPUTING ENVIRONMENT 600
630
PROCESSING UNIT 610
MEMORY 620
COMMUNICATION CONNECTION(S) 670
INPUT DEVICE(S) 650
OUTPUT DEVICE(S) 660
STORAGE 640
SOFTWARE 680 FOR DESCRIBED TECHNOLOGIES

FIG. 6