US 20130279882A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2013/0279882 A1

Singer (43) Pub. Date: Oct. 24, 2013

(54) **CODING OF VIDEO AND AUDIO WITH INITIALIZATION FRAGMENTS**

(71) Applicant: **APPLE INC.**, Cupertino, CA (US)

(72) Inventor: **David W. Singer**, Fremont, CA (US)

(73) Assignee: **APPLE INC.**, Cupertino, CA (US)

(21) Appl. No.: **13/631,194**

(22) Filed: **Sep. 28, 2012**

Related U.S. Application Data

(60) Provisional application No. 61/637,068, filed on Apr. 23, 2012, provisional application No. 61/637,263, filed on Apr. 23, 2012.

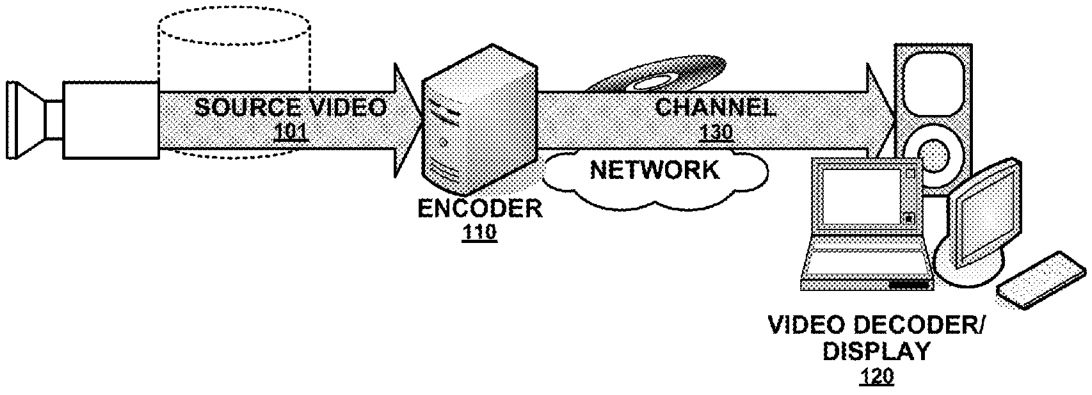
Publication Classification

(51) **Int. Cl.**
H04N 7/26 (2006.01)
H04N 9/80 (2006.01)

(52) **U.S. Cl.**
USPC **386/241**; 375/240.01; 375/240.25; 375/E07.026; 375/E07.027; 386/E05.003

(57) **ABSTRACT**

A new file format for coded video data is provided. A decoder may identify patterns in the coded video data in order to make the decoding process and/or display of data more efficient. Such patterns may be predefined and stored at the decoder, may be defined by each encoder and exchanged during terminal initialization, or may be transmitted and/or stored with the associated video data. Initialization information associated with the fragments of video data may also provide for carouseling initialization updates such that the initialization fragments may indicate either that the initialization information should be updated or that the decoder should be re-initialized. Additionally, media files or segments may be broken into fragments and each segment may have an index to provide for random access to the media data of the segment.



100

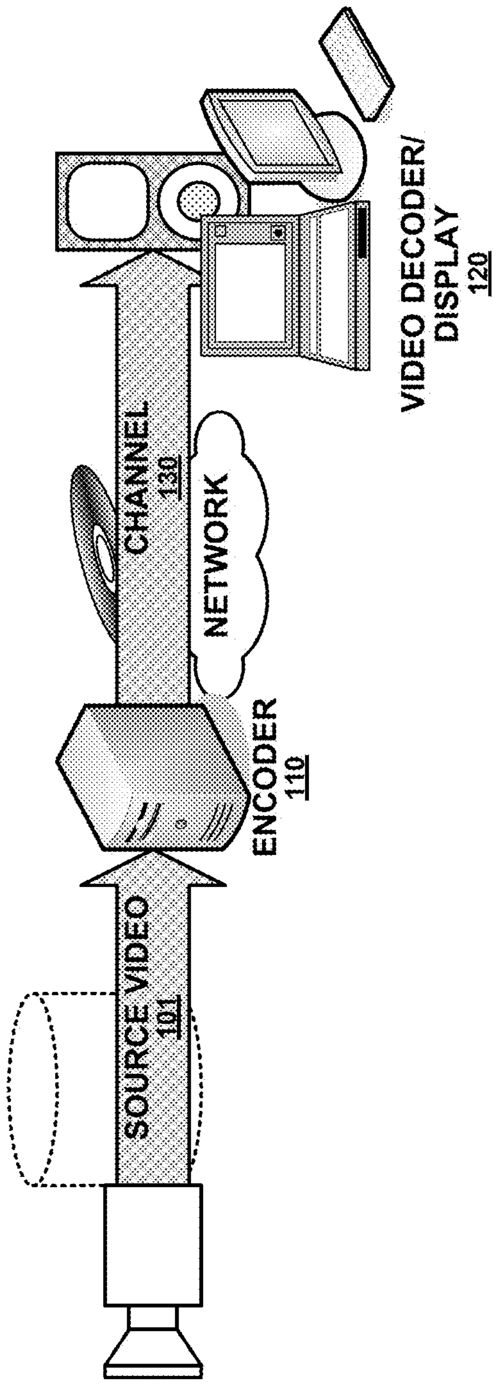

SOURCE VIDEO
101
ENCODER
110
CHANNEL
130
NETWORK
VIDEO DECODER/
DISPLAY
120

FIG. 1
100

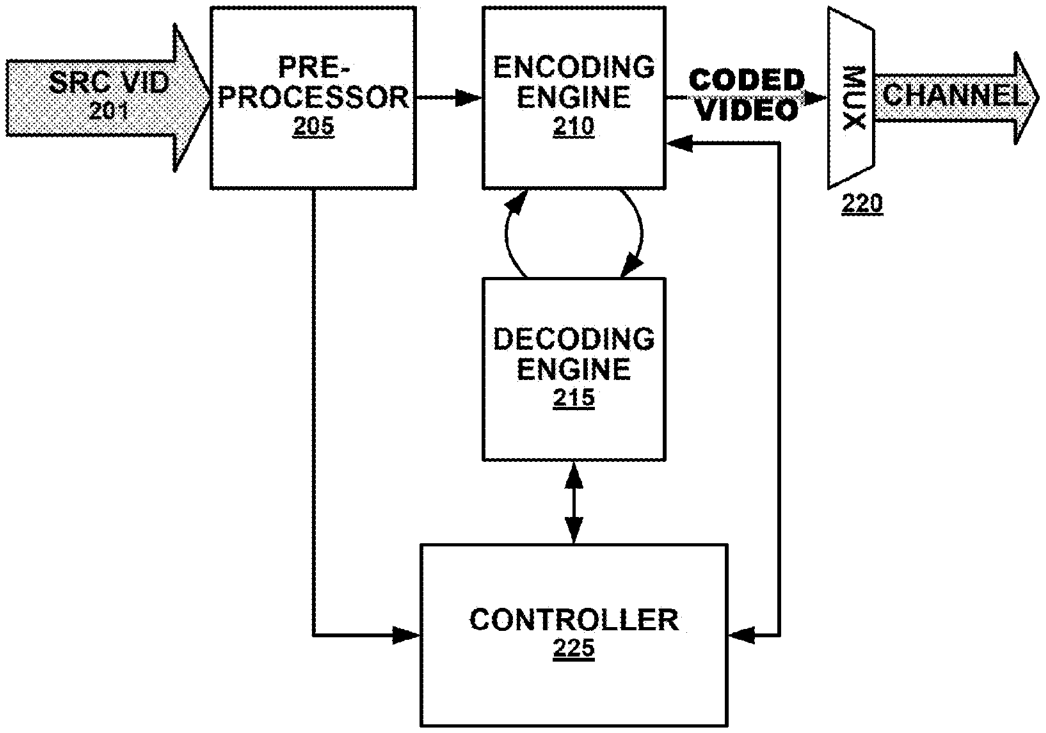

SRC VID
201
PRE-PROCESSOR
205
ENCODING ENGINE
210
CODED VIDEO
MUX
220
CHANNEL
DECODING ENGINE
215
CONTROLLER
225

FIG. 2
200

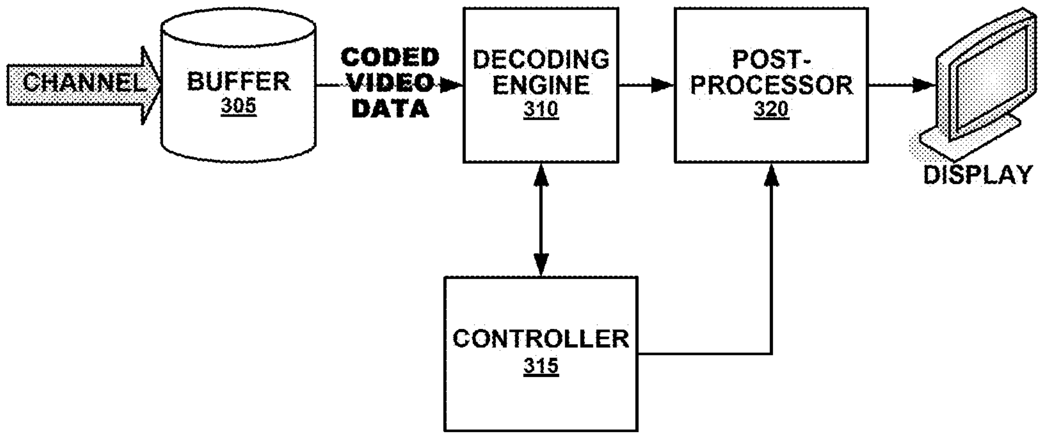

CHANNEL
BUFFER
305
CODED VIDEO DATA
DECODING ENGINE
310
POST-PROCESSOR
320
DISPLAY
CONTROLLER
315

FIG. 3
300

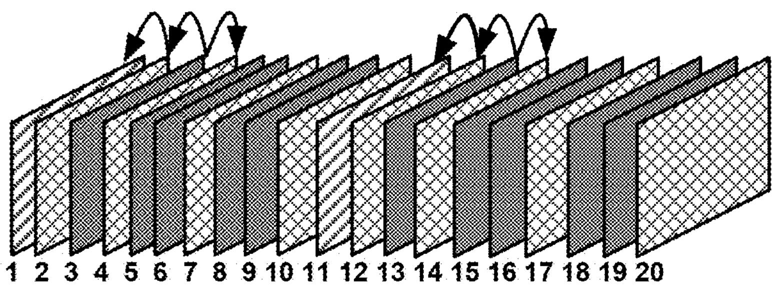


FIG. 4

400

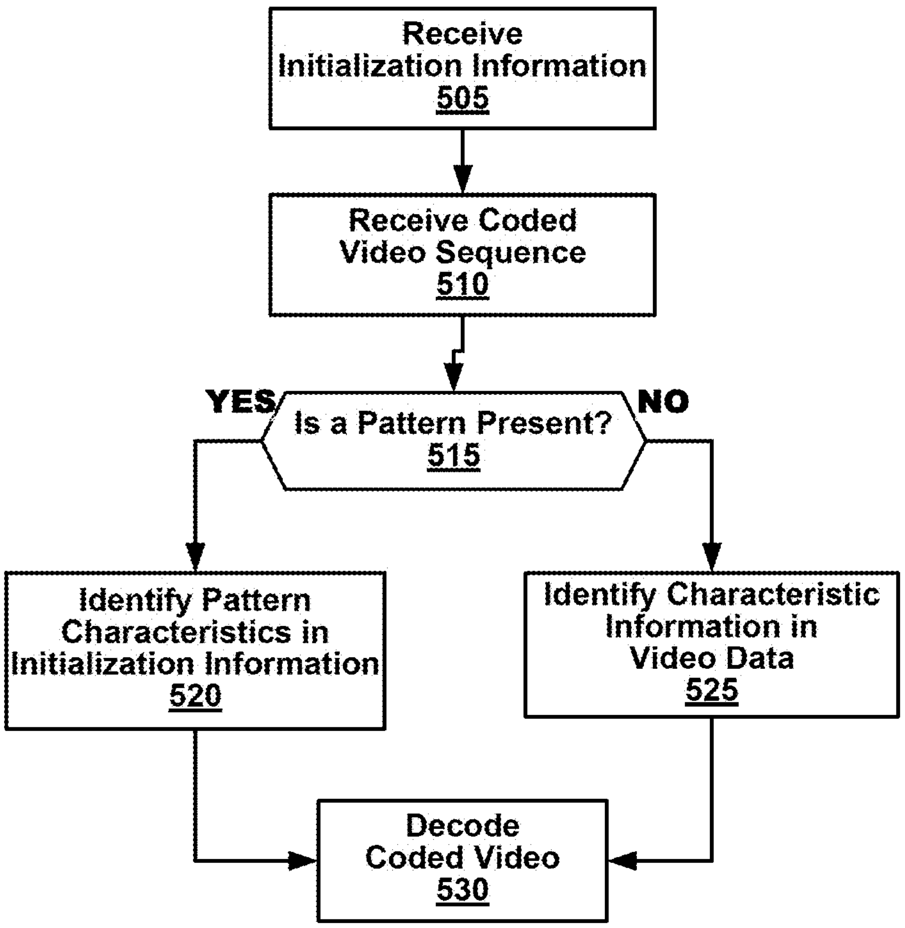

Receive Initialization Information 505
Receive Coded Video Sequence 510
YES
Is a Pattern Present? 515
NO
Identify Pattern Characteristics in Initialization Information 520
Identify Characteristic Information in Video Data 525
Decode Coded Video 530

FIG. 5
500

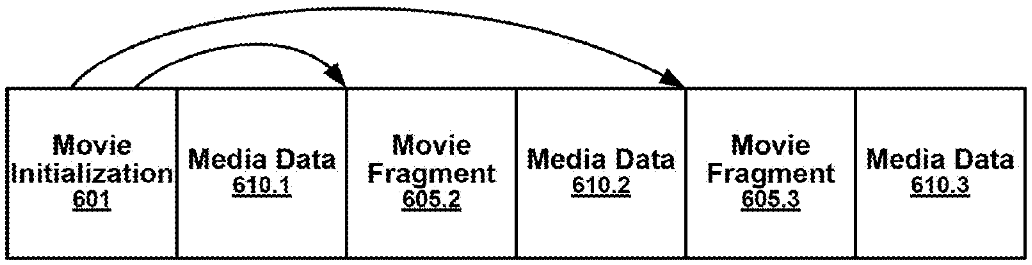

Movie Initialization 601
Media Data 610.1
Movie Fragment 605.2
Media Data 610.2
Movie Fragment 605.3
Media Data 610.3

FIG. 6
600

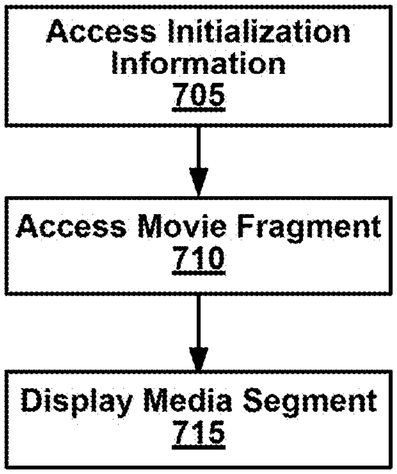

Access Initialization Information
705
Access Movie Fragment
710
Display Media Segment
715

FIG. 7
700

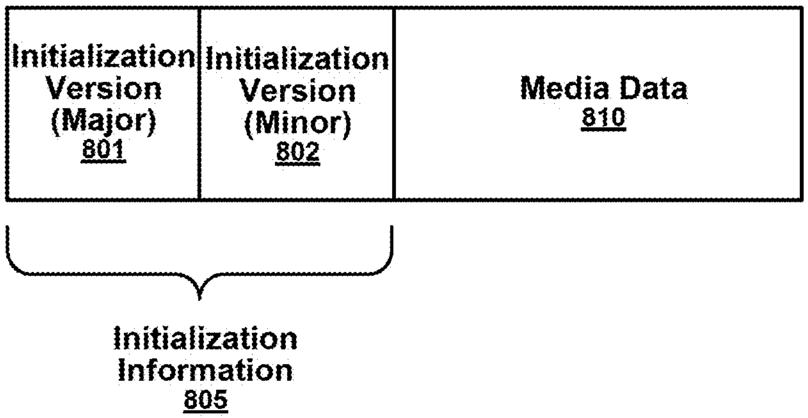

Initialization Version (Major) 801
Initialization Version (Minor) 802
Media Data 810
Initialization Information 805

FIG. 8
800

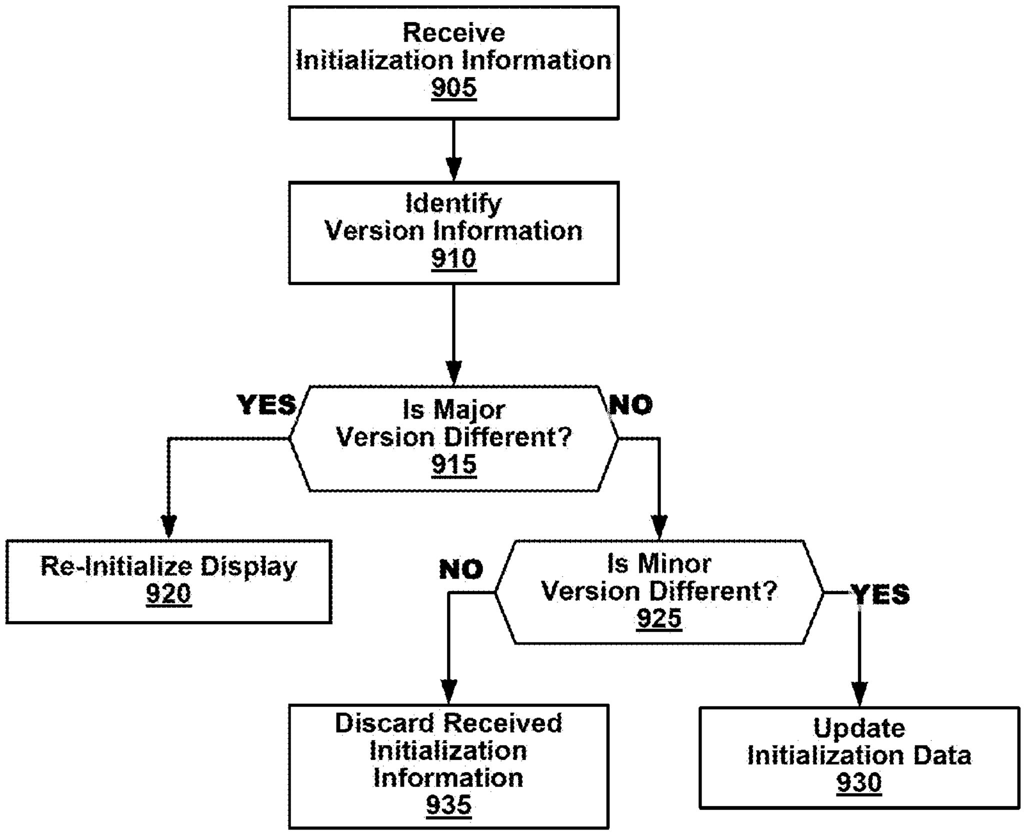

Receive
Initialization Information
905
Identify
Version Information
910
YES
Is Major
Version Different?
915
NO
Re-Initialize Display
920
NO
Is Minor
Version Different?
925
YES
Discard Received
Initialization
Information
935
Update
Initialization Data
930

FIG. 9
900

CODING OF VIDEO AND AUDIO WITH INITIALIZATION FRAGMENTS

CROSS-REFERENCE TO RELATED APPLICATION

[0001] The present application claims the benefit of co-pending U.S. provisional application Ser. No. 61/637,068, filed Apr. 23, 2012, entitled, “CODING OF VIDEO AND AUDIO”, and provisional application Ser. No. 61/637,263, filed Apr. 23, 2012, entitled, “A NEW MPEG FILE FORMAT” the disclosures of which are incorporated herein by reference in their entirety.

BACKGROUND

[0002] Aspects of the present invention relate generally to the field of video processing, and more specifically to the encoding and decoding of video data.

[0003] In video coding systems, an encoder may code a source video sequence into a coded representation that has a smaller bit rate than does the source video and thereby achieve data compression. Using predictive coding techniques, some portions of a video stream may be coded independently (intra-coded I-frames) and some other portions may be coded with reference to other portions (inter-coded frames, e.g., P-frames or B-frames). Such coding often involves exploiting redundancy in the video data via temporal or spatial prediction, quantization of residuals and entropy coding. Previously coded frames, also known as reference frames, may be temporarily stored by the encoder for future use in inter-frame coding. Thus a reference frame cache stores frame data that may represent sources of prediction for later-received frames input to the video coding system. The resulting compressed data (bitstream) may be transmitted to a decoding system via a channel. To recover the video data, the bitstream may be decompressed at a decoder by inverting the coding processes performed by the encoder, yielding a recovered decoded video sequence.

[0004] When coded video data is decoded after having been retrieved from a channel, the recovered video sequence replicates but is not an exact duplicate of the source video. Moreover, video coding techniques may vary based on variable external constraints, such as bit rate budgets, resource limitations at a video encoder and/or a video decoder or the display sizes that are supported by the video coding systems. In many coding applications, there is a continuing need to maximize bandwidth conservation. When video data is coded for consumer applications, such as portable media players and software media players, the video data often is coded at data rates of approximately 8-12 Mbits/sec and sometimes 4 MBits/sec from source video of 1280×720 pixels/frame, up to 30 frames/sec.

[0005] In many systems, bandwidth is rising but latency is typically limited to a large extent by the speed signals travel, and therefore remains consistent or may even be rising due to buffer delays. Furthermore, many common file formats were not designed for modern media delivery techniques, notably streaming and one-to-many distribution (broadcast, multicast, application layer multicast or peer-to-peer distribution). Conventionally, bandwidth efficiency has been achieved when round-trip delay cost is amortized over larger data objects, such as segments. HTTP streaming is one example of an environment that addresses these issues but the format for such media in that instance is dependent on the delivery system.

[0006] Accordingly, there is a need in the art for a new file format designed to simplify and modernize existing file formats while still re-using existing formats as much as possible, to allow for easy conversion between delivery and storage, and optimization for media delivery.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The foregoing and other aspects of various embodiments of the present invention will be apparent through examination of the following detailed description thereof in conjunction with the accompanying drawing figures in which similar reference numbers are used to indicate functionally similar elements.

[0008] FIG. **1** is a simplified block diagram illustrating components of an exemplary video coding system according to an embodiment of the present invention.

[0009] FIG. **2** is a simplified block diagram illustrating components of an exemplary encoder according to an embodiment of the present invention.

[0010] FIG. **3** is a simplified block diagram illustrating components of an exemplary decoder according to an embodiment of the present invention.

[0011] FIG. **4** illustrates an exemplary fragment having a pattern according to an embodiment of the present invention.

[0012] FIG. **5** is a simplified flow diagram illustrating an exemplary method for selecting pattern based defaults for decoding a coded video sequence.

[0013] FIG. **6** illustrates an exemplary video file having a plurality of movie fragments indexed by the initialization information according to an embodiment of the present invention.

[0014] FIG. **7** is a simplified flow diagram illustrating an exemplary method for randomly accessing a portion of a media file according to an embodiment of the present invention.

[0015] FIG. **8** illustrates an exemplary fragment of coded video data having carouseling initialization information according to an embodiment of the present invention.

[0016] FIG. **9** is a simplified flow diagram illustrating an exemplary method for identifying initialization update information in a stream of video data according to an embodiment of the present invention.

DETAILED DESCRIPTION

[0017] A new file format is presented, designed to simplify and modernize existing file formats. Conventional file types, including regular MP4 files may be automatically convertible into a file of the presented format. Data objects defined herein are incremental, capable of carrying a single stream or a package, re-packageable, and so on. A presentation can be represented as a packaged single media file, or as a plurality of tracks, or the un-timed data items such that the contents of the package may be stored separately. Presentations can also be fragmented in time, and the fragments collected into separate segments. According to an embodiment, a media fragment may contain a section of the timeline, including the media data whereas an initialization fragment may contain initialization information that may or may not be followed by media data. A segment of data may start with an initialization fragment.

[0018] FIG. **1** is a simplified block diagram illustrating components of an exemplary video coding system **100** according to an embodiment of the present invention. As shown in FIG. **1**, an exemplary video coding system may include an encoder system **110** and a decoder system **120** provided in communication via a channel **130**. The encoder system **110** may accept a source video **101** and may code the source video **101** as coded video, which typically has a much lower bit rate than the source video **101**. The encoder system **110** may output the coded video data to the channel **130**, which may be a storage device, such as an optical, magnetic or electrical storage device, or a communication channel formed by computer network or a communication network.

[0019] The terminals may exchange information as part of an initial handshake, for example information detailing the capabilities of each terminal. Each terminal may include an initialization table containing default parameters for encoding and decoding. The information exchanged during the handshake may then identify the coding format and default initialization table.

[0020] The decoder system **120** may have initialization information that may include coding patterns in the coded video data. For example, for time based coding techniques, the media data may consist of fragments that each may have various time-based characteristics. For example, each frame may have an associated time length, coding order and display order. A fragment may contain a sequence of frames for which the associated characteristics follow a predefined pattern. The pattern information may be exchanged during the handshake process or may otherwise be stored or transmitted with the coded video data. In some instances, the initialization information may define the default setup of data references, sample entries, etc. For a fragment of video data, associated initialization information may include tables that define such coding patterns. When such patterns are known, they can be indexed into. These patterns include sample size pattern, duration, timing, re-ordering, sample group membership, and other fragment characteristics that may be used to decode and display the coded video data.

[0021] The decoder system **120** may retrieve the coded video data from the channel **130**, invert the coding operations performed by the encoder system **110** and output decoded video data to an associated display device. The decoder system **120** may access to initialization information associated with the retrieved coded video data. The initialization information may facilitate the decoding and/or display of the recovered media data.

[0022] According to an embodiment, the coding system **100** may include terminals that communicate via a network. The terminals each may capture video data locally and code the video data for transmission to another terminal via the network. Each terminal may receive the coded video data of the other terminal from the network, decode the coded data and display the recovered video data. Video terminals may include personal computers (both desktop and laptop computers), tablet computers, handheld computing devices, computer servers, media players and/or dedicated video conferencing equipment. As shown in FIG. **1**, a pair of terminals are represented by the encoder system **110** and the decoder system **120**. As shown, the coding system **100** supports video coding and decoding in one direction only. However, according to an embodiment, bidirectional communication may be achieved with an encoder and a decoder implemented at each terminal.

[0023] FIG. **2** is a simplified block diagram illustrating components of an exemplary encoder **200** according to an embodiment of the present invention. As shown in FIG. **2**, the encoder **200** may include a pre-processor **205**, a coding engine **210**, a decoding engine **215**, a multiplexer **220**, and a controller **225**. The encoder **200** may receive an input source video sequence **201** from a video source such as a camera or storage device. The pre-processor **205** may process the input source video sequence **201** as a series of frames and condition the source video for more efficient compression. For example, the image content of an input source video sequence may be evaluated to determine an appropriate coding mode for each frame. The pre-processor **205** may additionally perform video processing operations on the frames including filtering operations such as de-noising filtering, bilateral filtering or other kinds of processing operations that improve efficiency of coding operations performed by the encoder **200**.

[0024] The coding engine **210** may receive the processed video data from the pre-processor **205** and generate compressed video. The coding engine **210** may operate according to a predetermined multi-stage protocol, such as H.263, H.264, or MPEG-2. The coded video data, therefore, may conform to a syntax specified by the protocol being used. The coding engine may additionally select from or be assigned one of a variety of coding modes to code the video data, where each different coding mode yields a different level of compression, depending upon the content of the source video. For example, the coding engine **210** may parse source video frames according to regular arrays of pixel data (e.g., 8×8 or 16×16 blocks), called "pixel blocks" herein, and may code the pixel blocks according to block prediction and calculation of prediction residuals, quantization and entropy coding.

[0025] The encoder **200** may further include a decode engine **215** that decodes the coded pixel blocks output from the coding engine **210** by reversing the coding operations performed therein. The decoding engine **215** may generate the same decoded replica of the source video data that a decoder system will generate, which can be used as a basis for predictive coding techniques performed by the coding engine **210**. The decoding engine **215** may access the reference frame cache to retrieve reference data for decoding and to store decoded frame data that may represent sources of prediction for later-received frames input to the video coding system.

[0026] The coded frames or pixel blocks may then be output from the coding engine **210** and stored by the MUX **220** where they may be combined into a common bit stream to be delivered by the transmission channel to a decoder, terminal, or data storage. In an embodiment, the encoder **200** may transmit initialization information with the coded frames for a fragment of video data in logical channels established by the governing protocol for out-of-band data. As one example, used by the H.264 protocol, the encoder **200** may transmit accumulated statistics in a supplemental enhancement information (SEI) channel specified by H.264. In such an embodiment, the MUX **220** represents processes to introduce the initialization information in a logical channel corresponding to the SEI channel. When the present invention is to be used with protocols that do not specify such out-of-band channels, the MUX **220** may establish a separate logical channel for the noise parameters within the output channel.

[0027] During encoding, the controller **225** may monitor the operations of the preprocessor **205**, the operations of the coding engine **210**, the coded video data, and/or the recovered video data to identify patterns of certain characteristics. For example, patterns for the segment size, duration, timing, and re-ordering of video data may be identified. According to an embodiment, patterns may be limited to a specific segment size of the coded video data. The controller **225** may collect this information in an initialization table for each segment. The initialization information may then be transmitted on the channel with the associated coded video data. According to an embodiment, the controller **225** may control certain aspects of the coding engine, for example by setting coding parameters or segmenting the source video data, to ensure appropriate patterns are utilized.

[0028] FIG. **3** is a simplified block diagram illustrating components of an exemplary decoder **300** according to an embodiment of the present invention. As shown in FIG. **3**, the decoder **300** may include a buffer **305** to receive and store the coded channel data and to separate the coded video data from the initialization information, a decoding engine **310** to receive coded video data and inverse coding processes performed by an encoder, a controller **315** to identify the characteristics of the coded video data and select a decoding mode for the coded video data, and a post-processor **320** that further processes the decoded video to prepare it for display.

[0029] The decoder **300** may receive initialization information from the channel. For example, in a supplemental enhancement information (SEI) channel specified by H.264. In such an embodiment, the buffer **305** represents processes to separate the noise parameters from a logical channel corresponding to the SEI channel. However, when the present invention is to be used with protocols that do not specify such out-of-band channels, the buffer **305** may separate the noise parameters from the encoded video data by utilizing a logical channel within the input channel.

[0030] Initialization information may be utilized by the controller **315** to set certain parameters for the decoding engine **310** or to otherwise prepare the video data for display. For example, decoding parameters may be set for based on the known coding modes for each frame according to a predefined pattern. Initialization information may be stored at the controller **315** or in a separate memory device (not shown) accessible by the controller **315**.

[0031] Post-processing operations may include filtering, de-interlacing, scaling or performing other processing operations on the decompressed sequence that may improve the quality of the video displayed with the post-processor. The processed video data may be displayed on a screen or other display or may be stored in a storage device for later use. The initialization information may be utilized to index the recovered video data and facilitate random access playback of the media.

[0032] FIG. **4** illustrates an exemplary fragment having a pattern according to an embodiment of the present invention. As shown, the fragment includes a plurality of frames, the first frame (1) may be encoded as an I-frame, and a plurality of subsequent frames (2-10) may be coded as B- or P-frames. Then a subsequent frame (11) may be coded as an I frame and the plurality of subsequent frames (12-30) may be encoded in a similar pattern of B- and P-frames.

[0033] A long pattern may have sub-parts that cover common short patterns. A short pattern may regularly repeat. A fragment longer than a pattern may keep looping with the same pattern until the end of the fragment. If sequence-specific values are used such that the fragment does not follow a default or otherwise known pattern, then a table defining the pattern values may be transmitted from the encoder to the decoder with the fragment. If no information is transmitted, an implied pattern may be used, based on an initial offset into the default pattern and the length in the fragment.

[0034] The initialization information associated with the fragment may set a pattern for multiple characteristics. Patterns may be known for segment size, duration, timing, re-ordering, group membership, etc. For example, each pattern may be defined to have a fixed length. The pattern length may be shorter or longer than the total segment count in a fragment. The fragment may additionally indicate an initial offset into a specified pattern. The pattern may then be repeated as needed, to cover the entire fragment.

[0035] FIG. **5** is a simplified flow diagram illustrating an exemplary method **500** for selecting pattern based defaults for decoding a coded video sequence. As shown in FIG. **5**, an initialization table may be received at the decoder associated with a received coded video sequence (blocks **505**, **510**). The initialization table may define the default values of the patterns in the sequence. As previously noted, the initialization table may be transmitted to the decoder during an initial handshake process between terminals, or may be transmitted with each associated fragment of video data. A fragment having initialization information within the fragment may be known as an initialization fragment. The initialization information may define the patterns for the media data associated with the fragment or may define the patterns for a plurality of media fragments, including all the fragments in an associated segment or file.

[0036] Then for fragments not having an associated initialization table (block **515**), the decoder may identify a pattern and characteristic information associated with the identified pattern. A need for the pattern identification may be signaled by the lack of a table associated with the segment. For known patterns, the predefined defaults may then be used. For frames not a part of any predefined pattern, the characteristic values may be included in the coded video data (block **525**).

[0037] For fragments having an associated initialization table (block **515**), the decoder may then identify the characteristic values for the frames in the sequence (block **520**). A pattern will have a corresponding set of defaults that may be the same for every frame following the pattern. Then the decoder may decode the coded video using the characteristic values previously identified (block **530**).

[0038] FIG. **6** illustrates an exemplary video file **600** having a plurality of movie fragments **605** indexed by the initialization information **601** according to an embodiment of the present invention. Each movie fragment **605.1**-N may have a defined start time and duration, and if stored contiguously, have a known starting byte and size. Movie fragments may be implemented such that the initialization table may be used to access each fragment in a file. Once a fragment **605.1**-N is accessed, the associated media data **610.1**-N may be known and available for display or playback. A table, and marking on the fragments, may then allow for random access of a portion of media data without requiring the decoding unit to parse the entire movie for playback.

[0039] FIG. **7** is a simplified flow diagram illustrating an exemplary method **700** for randomly accessing a portion of a media file according to an embodiment of the present invention. As shown in FIG. **7**, to display a portion of a media file, a decoder may access the received initialization information for the file (block **705**). As previously noted, the initialization information may include characteristic information for the file and include a table that identifies the size or duration of each of a plurality of fragments within the file. According to an embodiment, the initialization information may be transmitted to the decoder with the file, with each associated fragment, or may be exchanged in an initial handshake process between terminals. The initialization table associated with a file or fragment may then be stored therewith. In order to access a requested fragment, the controller may parse the table to identify the start of the relevant media file and any characteristics that carry from the initialization information throughout the file and are applicable to the requested fragment.

[0040] The decoder may then access the initialization information associated with the fragment information (block **710**). As previously noted, the fragment information may include characteristic information for the fragment. The requested media may then be displayed using the characteristic information accessed in the initialization information to identify the start of the fragment and any other necessary information for appropriate display (block **715**).

[0041] FIG. **8** illustrates an exemplary fragment of coded video data **800** having carouseling initialization information according to an embodiment of the present invention. Initialization information **805** for media data **810** may be periodically dumped or updated, for example, when streaming data. A conventional decoder displaying the media data **810** will then reinitialize the settings and characteristics for the media data each time an initialization fragment is received. As shown in FIG. **8**, initialization information having carouseling version data may indicate whether re-initialization is required, thereby avoiding unnecessary re-initialization.

[0042] As shown in FIG. **8**, the received initialization information **805** may give a major **801** and minor **802** version number, documenting whether re-initialization is needed when a new initialization packet is encountered or just an update is required. Both sync and non-sync initialization fragments may be included in the initialization information **805**. For example, the initialization version (major) **801** may indicate whether a complete re-initialization is required, for example, if the streaming information is switching between different codecs. Whereas the initialization version (minor) **802** may indicate whether an update of the initialization information is required.

[0043] According to an embodiment, the initialization information **805** for the fragment may contain an edit list such that if an update is to be applied to the whole media the edit list may be replaced in a carouseled version of the initialization segment. If a new edit list maps the media data to the same presentation times, the edit list indicates a minor update to the initialization segment. Otherwise, it indicates a major (re-initialization) update.

[0044] For example, for pair of initialization fragments **805** received at the decoder, if the two versions **801**, **802** are identical in both fragments, the initialization information is identical and the latter received initialization fragment is a resend of the known initialization fragment. If the major version **801** is identical, but the minor version **802** has changed between the two received fragments, then the later received fragment is a compatible update. For example, a fragment may be compatible if it includes additional metadata that applies to the whole duration of the presentation but does not require a re-initialization. However, if the major version **801** has changed from a first received fragment to a latter received fragment, then the latter received fragment contains new initialization information and requires a re-initialization.

[0045] According to another embodiment, a fragment **800** that contains an initialization fragment **805** can be marked as a random access point in the index for the file. Then for a minor version **802** change, the update initialization fragment may contain the difference between the original initialization fragment and the update fragment. Then each subsequent segment may be either an independent (I) segment when the major version **801** indicates a re-initialization or a predictively (P) coded segment when the minor version **802** identifies changes to be made to the data of the I segment data.

[0046] FIG. **9** is a simplified flow diagram illustrating an exemplary method **900** for identifying initialization update information in a stream of video data according to an embodiment of the present invention. As shown in FIG. **9**, a decoder may receive initialization information for data being streamed to the decoder (block **905**). The initialization information may contain version information, major and minor, that identifies the version of the data (block **910**). The version of the received initialization data may then be compared to the version of the currently utilized initialization data (block **915**).

[0047] If the version information includes a major identification and a minor identification, and the major identification for the received initialization data is different than the major identification for the currently utilized initialization data, the decoder should be reinitialized using the received initialization data (block **920**).

[0048] However, if the major versions are the same, but the minor versions are different, the received initialization data indicates that a change to the initialization information should occur (block **925**). Such changes may include updating the edit table or replacing other information in the currently utilized initialization data by the information in the received initialization data (block **930**). If the major and minor versions on both the received information data and the currently utilized information data are the same, the received information may be discarded (block **935**).

[0049] As discussed above, FIGS. **1**, **2**, and **3** illustrate functional block diagrams of terminals. In implementation, the terminals may be embodied as hardware systems, in which case, the illustrated blocks may correspond to circuit sub-systems. Alternatively, the terminals may be embodied as software systems, in which case, the blocks illustrated may correspond to program modules within software programs. In yet another embodiment, the terminals may be hybrid systems involving both hardware circuit systems and software programs. Moreover, not all of the functional blocks described herein need be provided or need be provided as separate units. For example, although FIG. **2** illustrates the components of an exemplary encoder, such as the pre-processor **205** and coding engine **210**, as separate units. In one or more embodiments, some components may be integrated. Such implementation details are immaterial to the operation of the present invention unless otherwise noted above. Similarly, the encoding, decoding and post-processing operations described with relation to FIGS. **5**, **7**, and **9** may be performed continuously as data is input into the encoder/decoder. The order of the steps as described above does not limit the order of operations.

[0050] Some embodiments may be implemented, for example, using a non-transitory computer-readable storage medium or article which may store an instruction or a set of instructions that, if executed by a processor, may cause the processor to perform a method in accordance with the disclosed embodiments. The exemplary methods and computer program instructions may be embodied on a non-transitory machine readable storage medium. In addition, a server or database server may include machine readable media configured to store machine executable program instructions. The features of the embodiments of the present invention may be implemented in hardware, software, firmware, or a combination thereof and utilized in systems, subsystems, components or subcomponents thereof. The "machine readable storage media" may include any medium that can store information. Examples of a machine readable storage medium include electronic circuits, semiconductor memory device, ROM, flash memory, erasable ROM (EROM), floppy diskette, CD-ROM, optical disk, hard disk, fiber optic medium, or any electromagnetic or optical storage device.

[0051] While the invention has been described in detail above with reference to some embodiments, variations within the scope and spirit of the invention will be apparent to those of ordinary skill in the art. Thus, the invention should be considered as limited only by the scope of the appended claims.

We claim:

1. A video coding method, comprising:

for a sequence of video frames, coding the sequence of frames;

identifying a pattern in a characteristic of the frames; and

transmitting the pattern on a channel with the sequence of frames.

2. The method of claim **1**, wherein said characteristic includes a coding mode for each frame.

3. The method of claim **1**, wherein said characteristic includes a coding order for the sequence of frames.

4. The method of claim **1**, wherein said characteristic includes a display order for the sequence of frames.

5. The method of claim **1**, wherein said characteristic includes a character sync for each frame in the sequence of frames.

6. The method of claim **1**, wherein said characteristic includes a display duration for each frame in the sequence of frames.

7. The method of claim **1**, wherein said pattern has a length shorter than the sequence of frames.

8. The method of claim **7**, wherein the pattern repeats within the sequence of frames.

9. The method of claim **1**, wherein the pattern comprises a plurality of patterns shorter than the sequence of frames.

10. The method of claim **1**, wherein said pattern has a length greater than the sequence of frames.

11. The method of claim **1**, wherein said identifying further comprises: matching the sequence of frames to a pattern in a default pattern table.

12. The method of claim **11**, further comprising: exchanging the default pattern table during a handshake procedure between an encoder and a decoder

13. The method of claim **12**, wherein said transmitting further comprises including an indication that the default pattern table contains the characteristic pattern.

14. The method of claim **1**, further comprising: defining the characteristic pattern with an initialization table.

15. The method of claim **14**, wherein said transmitting further comprises: transmitting the initialization table on the channel with the sequence of frames.

16. A method for decoding a sequence of coded frames comprising:

identifying a pattern of frames in the sequence of frames;

matching the pattern of frames to a pattern defined in an initialization table; and

decoding frames in the pattern of frames using values defined in the initialization table for the pattern.

17. The method of claim **16**, further comprising: exchanging the initialization table during a handshake procedure between an encoder and a decoder.

18. The method of claim **16**, further comprising: receiving the initialization table on the channel with the sequence of frames.

19. The method of claim **16**, further comprising: receiving a pattern identifier on the channel with the sequence of frames.

20. A method for accessing a media segment comprising:

accessing an initialization block for the media segment, the initialization block identifying a plurality of movie fragments in the media segment; and

with the identification provided in the initialization block, accessing a fragment of the media segment.

21. The method of claim **20**, wherein said initialization block further defines at least one characteristic of the media segment.

22. The method of claim **21**, wherein said fragment of the media segment includes a change to be applied to the at least one characteristic defined in the initialization block.

23. The method of claim **20**, wherein said fragment further defines at least one characteristic of media data in the fragment.

24. The method of claim **23**, wherein said characteristic is a size of the fragment.

25. The method of claim **23**, wherein said characteristic is a duration of the media data.

26. A method for decoding streaming media data, comprising:

receiving at a decoder a first initialization fragment for the media data, the first fragment having a first major version number and a first minor version number;

receiving at a decoder a second initialization fragment for the media data, the second fragment having a second major version number and a second minor version number;

if the first major version number and the second major version number are different, re-initializing the decoder with the second fragment;

if the first major version number and the second major version number are the same, and the first minor version number and second minor version number are different, updating initialization data at the decoder with information provided in the second fragment; and

if the first major version number and the second major version number are the same, and the first minor version number and second minor version number are the same, discarding the second fragment.

27. The method of claim **26**, wherein said updating initialization data further comprises:

applying a change to the initialization data of the first fragment according to the information provided in the second fragment, wherein the information provided in the second fragment includes a difference between the initialization data for the first fragment and the updated information.

* * * * *