APPLICATION ENHANCEMENT TRACKS

BACKGROUND

[0001] The present invention relates generally to digital video distribution and playback systems and in particular to digital video distribution and playback systems providing enhanced playback control.

[0002] The digital video revolution is evolving from a physical-media distribution model to electronic-media distribution models that utilize Content Delivery Networks (CDNs) and Consumer Grade Networks (CGNs – such as residential Internet and in-home networks) for delivery of content to devices. The utilization of the Advanced Video Coding (AVC/H.264) standard is prevalent in today's optical and broadcast industries, but the adoption of this standard at bit-rates suitable for CDN/CGN distribution has not yet materialized in a unified and open specification for resolutions including full-HD (1080p) video.

[0003] Digital video formats however are typically designed to efficiently support playback of content. Other common user functions are typically supported through increased player complexity (and therefore cost) or the performance of the other functions is compromised, limiting the quality of the user-experience.

[0004] For example, visual-search through digitally encoded multimedia files is typically performed by displaying only the key-frames (aka intra-frames) of the relevant video stream. The key-frames are displayed for a time corresponding to the speed of the visual search being performed by the user and some may be skipped when a high search speed is requested. Alternate methods may decode all or parts of the video stream at higher rates and display selective frames to visually increase the presentation speed. These methods for visual-search may deliver a poor visual experience to the user due to the coarseness and inconsistency of the temporal difference between displayed images. Complicating matters even more is that devices

operate differently depending on whether visual-search is performed in the forward or reverse direction. Finally, devices may require the video stream to be read at speeds that are multiple times higher than the standard rate required to playback the video in normal speed, challenging the device's subsystems.

[0005] Similarly, other typical functions performed or required to be performed during a playback session with a single or with multiple titles of content can often be limited in their ability to deliver a consistent, high-quality experience to the user.

SUMMARY

[0006] Generally, the present invention provides a specific set of operating points that have been devised in order to maximize compatibility across both personal computer (PC) and consumer electronics (CE) platforms, resulting in high quality video at data rates that are encoded at up to 40% lower rates than those of the H.264 Level 4 data rates, while still maintaining a good visual quality level.

[0007] In particular, the effects of the CDN/CGN compression settings on visual-search are provided along with a method and system that increases the user-experience beyond traditional visual-search on optical-media. The method and system offer smooth visual-search capability in both the forward and reverse directions, operating at speeds from 2x to 200x and beyond, implementable on both PCs and CE devices that access content from optical disks or electronic sources across CDNs/CGNs. When combined, both of these features provide a high quality user-experience for content targeted at delivery over many types of networks.

[0008] In various embodiments provided herein consistent visual behavior during visual-search is provided, which operates equally well in both forward and reverse search directions, while simultaneously and substantially reducing the demands on the device in delivering the high-quality experience.

[0009] In one embodiment, a method of encoding a media file for playing back is provided. The method comprises extracting a video track from an original media file, where content is encoded in the video track; using the encoded content to encode an application enhancement track, where encoding the application enhancement track includes discarding at least some of the content; and creating a media file including the content from the original media file encoded in a video track and an accompanying application enhancement track.

[0010] In another embodiment, method of decoding a media file for playing back comprises obtaining a media file containing compressed content and an accompanying application enhancement track which is a subset of the compressed content; playing back the compressed content; and decoding frames of the application enhancement track at a rate proportional to a visual-search speed and from a location determined by the portion of the compressed content most recently played back.

[0011] In yet another embodiment, a system for playback of a media file comprises a media server and a client processor. The media server is configured to generate at least one application enhancement track from an original media file, the at least one application enhancement track having at least one frame in common with the original media file and being substantially smaller in file size than the original media file. The client processor is in network communication with the media server and is configured to send requests for the original media file to the media server. The media server is also configured to transmit the requested original media file along with the at least one application enhancement track.

[0012] In one other embodiment, a media player comprises a user interface configured to receive user instructions and a playback engine configured to decode a media file containing content and an accompanying application enhancement track, which is a subset of the content. The playback engine is configured to commence playback of the content in response to a user instruction received via the user interface.

The playback engine is also configured to select portions of the application enhancement track to decode and playback in response to receipt of a user instruction received via the user interface and the portion of the content most recently played back by the playback engine.

[0013] The above-mentioned and other features of this invention and the manner of obtaining and using them will become more apparent, and will be best understood, by reference to the following description, taken in conjunction with the accompanying drawings. The drawings depict only typical embodiments of the invention and do not therefore limit its scope.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] FIG. 1 is a graphical representation of an exemplary frame distribution in one second of video at twenty four frames per second;

[0015] FIG. 2 is a graphical representation of an exemplary frame distribution in one second of video at twenty four frames per second;

[0016] FIG. 3 is a graphical representation of an exemplary frame distribution in one second of video at twenty four frames per second;

[0017] FIG. 4 is a graphical representation of an exemplary frame distribution in one second of video at twenty four frames per second;

[0018] FIG. 5 is a graphical representation of an exemplary frame distribution in a predetermined section of video content;

[0019] FIG. 6 is a graphical representation of an exemplary file distribution of video content and an application enhancement track in accordance with an embodiment of the invention;

[0020] FIG. 7 is a graphical representation of an exemplary file distribution of video content and an application enhancement track in accordance with an embodiment of the invention;

[0021] FIG. 8 is a graphical representation of an exemplary file distribution of video content and an application enhancement track in accordance with an embodiment of the invention;

[0022] FIG. 9 is a graphical representation of spatial scaling conversion of video content into an application enhancement track in accordance with an embodiment of the invention;

[0023] FIG. 10 is a graphical representation of temporal scaling conversion of video content into an application enhancement track in accordance with an embodiment of the invention;

[0024] FIG. 11 is a graphical representation of an exemplary frame distribution in two seconds of video of an application enhancement track in accordance with an embodiment of the invention;

[0025] FIG. 12 is a graphical representation of an exemplary frame distribution in two seconds of video of an application enhancement track in accordance with an embodiment of the invention;

[0026] FIG. 13 is a graphical representation of an exemplary frame distribution in one second of video at twenty four frames per second in accordance with an embodiment of the invention;

[0027] FIGs. 14A-B are graphical representations of an exemplary frame distribution of video;

[0028] FIG. 14C is a graphical representation of an exemplary frame distribution of video of an application enhancement track in accordance with an embodiment of the invention;

[0029] FIG. 15 is a graphical representation of an exemplary audio distribution of two channel audio at 96 KHz sample rate and 24 bit sample sizes;

[0030] FIG. 16 is a graphical representation of spatial scaling conversion of audio content into an application enhancement track in accordance with an embodiment of the invention;

[0031] FIG. 17 is a graphical representation of temporal scaling conversion of audio content into an application enhancement track in accordance with an embodiment of the invention;

[0032] FIG. 18 is a flow chart showing a process of encoding an application enhancement track along with the content in accordance with an embodiment of the invention;

[0033] FIG. 19 is a network diagram of an exemplary system of encoding, communicating and decoding an application enhancement track and the content in accordance with an embodiment of the invention; and

[0034] FIG. 20 is a flow chart showing a process of playing an application enhancement track along with the content in accordance with an embodiment of the invention.

DETAILED DESCRIPTION

[0035] Generally, a digital video playback system is provided to ensure smooth playback regardless of the playback speed or direction that allows for the delivery of a high-quality experience to the user while allowing for the reduction in the processing load placed on a device.

[0036] Digitally compressed video is typically encoded using algorithms such as those defined by the MPEG committee (e.g. MPEG-2, MPEG-4 Part 2 and MPEG-4 Part 10). These algorithms encode images from the source material in to sequences of "key-frames" and "delta-frames".

[0037] Key-frames contain all the data required to display a specific image from the source video. A delta-frame contains the difference data between one or more

previously decoded images and the image it encodes. In general, there is a 1:1 mapping of source images and encoded frames. However, the 1:1 mapping does not hold true when the video is encoded at a different frame rate relative to the source video sequence. Thus, to decode frame *F* of a video sequence, all the frames that form the basis of the difference values contained in *F* must first be decoded. Applied recursively, this decode method ultimately requires a key-frame to start the decode process, since it is not based on any previously decoded image. Hence, the very first frame generated by typical encoders is a key-frame.

[0038] Since key-frames are encoded independently of other frames, they require more space to be stored (or more bandwidth during transfer), and put generically, can be attributed with a higher cost than delta-frames. For the purpose of describing the current inventions, a cost ratio of key-frames (K) versus delta-frames (D) 12 is used in which the ratio of 3:1 has been selected from observation of a range of encoded video content.

[0039] FIG. 1 illustrates a simplified exemplary view of a minimum bandwidth approach to creating an encoded bitstream 20 (assuming that there are no scene changes), where all source images are encoded to delta-frames 21. The frame cost distribution in one second of video at 24 frames per second is low with the cost equaling the number of delta-frames multiplied by the delta-frame cost (D). However, this approach is not practical, since at least the very first frame must be encoded as a key-frame and since there needs to be other key-frames inserted at other locations in the file to facilitate trick-play, e.g., visual-search and random access functions. As shown here and in the other similar following figures, the decode order is from left to right with the order of placement of each associated frame also being from left to right.

[0040] FIG. 2 provides a view of the very first second of a video track, where it can be seen that the first encoded frame is a key-frame 31 followed by delta-frames 32.

Consequently, the cost of the first second of this track has increased by approximately 8% from the delta-frame only approach of FIG. 1. (Cost = 1*K + 23*D = 26).

[0041] FIG. 3 and FIG. 4 illustrate how the cost increases as the frequency of key-frames is increased. The tradeoff is that more frequent use of key-frames gives greater flexibility in enabling decoding to near a randomly chosen point in the bitstream. In FIG. 3, two key frames 41 are spaced between twenty two delta frames 42. (Cost = 2*K + 22*D = 28). In FIG. 4, the number of key frames 51 is increased and the key frames are spaced within a fewer number of delta frames 52. (Cost = 4*K + 20*D = 32).

[0042] To perform rapid visual search through an encoded video bitstream, the decoder should increase its rate of decoding to match the speed requested by the user. However, this is not always practical due to performance bottlenecks in typical systems imposed by components such as disk I/O, memory I/O and the processor itself. Furthermore, reverse decode may be cumbersome, due to the nature of the encoding method which is optimized for playback in forward chronological order. Therefore most systems may ultimately rely on schemes such as dropping the processing of certain delta-frames or processing only key-frames in an order and at a rate determined by the direction and speed of visual search being conducted by the user.

[0043] Based on at least the above-noted factors, many video formats, such as DVD, ensure that there are key-frames regularly inserted throughout the duration of the multimedia content. In fact, the DVD format requires a key-frame to be inserted approximately every 600 ms throughout the content. This regular key-frame insertion delivers good visual-search functionality but at the cost of significantly increasing the file-size due to the frequent use of key-frames. Schemes such as these essentially employ encoding the content into segments with the key-frame/delta-frame distribution similar to those illustrated by FIGs. 2, 3 and 4.

[0044] Multiple efficient methods and systems to encode video content are described below in accordance with embodiments of the invention. One such encoding method

and system insert key-frames at least every 10 seconds, 4 seconds for H.264 content, in the absence of scene changes, and additionally at scene-change locations. This ensures efficient encoding of typical produced content. For example, compared to the DVD method of encoding (approx. 2 key-frames per second), methods in accordance with embodiments of the invention yield much smaller file sizes.

[0045] The distribution of key-frames 61 and delta-frames 62 in content 60, encoded in accordance with an embodiment of the invention is illustrated in FIG. 5. However, with such a potentially low frequency and non-deterministic placement of key-frames, trick-play performance can be severely degraded, based on the performance metrics previously described. As explained above, the addition of extra key-frames at more frequent, regular intervals would provide the ability for applications such as visual-search to be performed with consistent, high quality behavior. However, introducing more key-frames also increases the cost of any portion of video encoding. Cost also depends on the time bracket chosen. For example, a system with 6 key-frames per second would provide a smoother visual-search capability (in systems which rely only on key-frame decoding), but could increase the cost of each second of video by 50%. This high cost is prohibitive for most wide-scale applications.

[0046] In FIG. 5, it should be noted that the section of video shown is greater than one second as shown in previous figures to better illustrate the frame distribution for this case. As such, the key-frames 61 and delta-frames 62 appear smaller than other previously shown frames. It should also be noted that the figures are not necessarily to scale, but are merely conceptual illustrations to facilitate the description of the various embodiments of the invention.

[0047] Another functional area where most devices provide a compromised user-experience is when providing the user with a list of multiple content files. Many devices simply show the text of the file name and may show a static graphical icon for each piece of content. A higher-quality experience in such cases for example could be to

provide animated icons which show all or some of the content's video sequences. However, this type of capability would require a playback engine capable of decoding many pieces of content simultaneously, which is typically not feasible.

Application Enhancement Tracks

[0048] Application Enhancement Tracks (AETs) are media tracks that are encoded in a manner that improves the performance of one or more applications, or application features. Generally, AETs are re-encoded versions of one or more tracks of the source content. AETs for the most part contain significantly less data than the original file, and the data is present in a form that can be accessed easily to enhance the user-experience when one of the enhanced applications or features is being run. The data in an AET is, generally, designed such that the utilizing application does not need to achieve performance greater than that required to process the content.

[0049] Hence, an AET designed to allow "6x" visual-search through a title originally encoded at a frame-rate of 30 frames per second (fps) could be generated by re-encoding the original video to a rate of 5 fps. Thus, even when the AET was utilized to deliver "6x" visual-search, the device would experience a load of less than or equal to "1x" of that required to decode the original video. To further enhance this example, the original video could be spatially scaled down in resolution to 50% of its original size also in each of the vertical and horizontal dimensions (leading to a factor of four reduction in the number of pixels per frame); in this case the device could perform "24x" visual-search without requiring more than "1x" original video decode performance.

[0050] The AETs can be recorded directly in to the same file as the content they offer enhancements for, or they can be stored in a separate file. When stored within the same file as the content they enhance, the enhancement track data can be interleaved with the content, or may be located in one or more contiguous blocks anywhere in the file. FIGs. 6-8 illustrate some of the different methods by which the enhancement track

data can be associated with the content it enhances. Any one or any combination or all methods may be utilized to store the AETs. In FIG. 6, the content 71 is in one file and the associated AET 72 is in a separate file. In FIG. 7, the AET 82 follows or precedes the content 81. Content 91 is embedded with portions of the AET 92 or the content 91 and the AET 92 are weaved together to generate the combined content AET file 93.

Video AETs

[0051] As an example, Video AETs designed to improve the performance of visual-search and/or the displaying of dynamic icons by a preview engine can be created by optionally scaling the content to different spatial and temporal resolutions, and then re-encoding this scaled version into a data stream with a regular pattern of key-frames and, in several embodiments, delta-frames.

[0052] FIG. 9 illustrates the spatial scaling of the video source 101 to a suitable size for the enhancement track 102. The amount of scaling could increase or decrease the spatial resolution from the source video, and may alter the aspect ratio of the source video. Spatial scaling could also be achieved by adding or cropping the original image to the size required in the AET. FIG. 10 illustrates temporal scaling of the video source 111 to a reduced rate for the enhancement track 112. However, the amount of scaling applied could increase or decrease the temporal resolution from the source. Spatial scaling may be performed before temporal scaling and vice versa.

[0053] FIGs. 11-12 illustrate examples of the order and distribution of key-frames and delta-frames in Video AETs 120,130 where the AETs contain both key-frames 121,131 and delta-frames 122,132. In these examples, FIG. 11 illustrates a frame order and distribution for an enhanced forward visual search in accordance with an embodiment of the invention and FIG. 12 illustrates a frame order and distribution for an enhanced reverse visual search in accordance with another embodiment of the

invention. Frames 123,133 marked as "sk" indicate that the corresponding frame from the original source content has been skipped.

[0054] In these examples, during visual-search each frame in the AET is decoded and displayed at a rate proportional to the visual-search speed requested by the user. When the decode and display rate required by the visual-search speed exceeds the device's capabilities, the device can change mode, for example, from processing all frames to processing as many key-frames as required or pre-selected to support the required search speed. Since the key-frames have been placed at regular intervals throughout the AET, the visual performance of skipping from key-frame to key-frame will be consistent.

[0055] In a similar scheme as illustrated in FIGs. 11 and 12, each delta-frame could be predicted from a single key-frame (for example, the nearest regularly placed key-frame). In this method of encoding, an AET could be created with the property of needing no more than two decode cycles to display any frame in the AET. This scheme offers a refinement to the key-frame to key-frame skipping solution when performing rapid visual-searches, and allows for an intermediate state of speed where all key-frames and some delta-frames are decoded and displayed. Such an encoding method would also support bi-directional visual-searches.

[0056] Referring again to FIG. 12, the visual-search AET that is designed to specifically support reverse visual-search is generated by encoding the source video in reverse chronological order, while placing the encoded frames in the chronological order that they occur in the original, compressed or encoded content. This method allows reverse trick-play to be performed by reading the AET in the reverse-order of placement in the file, which is also the correct order for decoding. This method best allows the interleaving of content media track chunks with the chunks of the reverse visual-search AET.

[0057] FIG. 13 illustrates a single visual-search AET 140 that is encoded to support both forward and reverse search. This method uses only key-frames 141 and no delta-frames. Since key-frames contain absolute data, there is no implicit order required to decode them, therefore facilitating reverse or forward visual-search. Further examples of AETs that only include key-frames in accordance with embodiments of the invention are discussed below with reference to FIG. 19.

[0058] FIGs. 14A-C further illustrate the visual-search AET 150 encoded to support both forward and reverse search. In FIG. 14A, a sample video 150 encoded at 25 frames-per-second is provided that includes an intra-frame type (I Frame) 151, a bi-directionally predicted frame (B Frame) 152 and a uni-directionally predicted frame (P Frame) 153. As shown, the distance between the Intra frames is not uniform, which is shown more clearly in FIG. 14B. The temporal distance between the intra frames and the bit-rates for those intra frames vary greatly across the different frames. Both of these have been resolved with the addition of the visual-search AET 150 (FIG. 14C). The bit-rates across the intra-frames are more uniform, and the temporal distance is also consistent. Both of these factors will lead to a better user experience during smooth forward and reverse operations, in both optical as well as streaming playback scenarios.

[0059] Such Video AETs may also contain information in the track's data that relate the encoded video frames with the corresponding frame or time-period in the original title. This information may be encoded as time-stamps relative to the time-line of the content, or a file-position offset, or any other mechanism.

[0060] The illustrated AETs can also enhance the content-preview experience by virtue of the same properties exploited for visual-search. In typical content-preview modes, a reduced resolution version of the content is displayed along with other information that could be used to identify the content, such as the file name, duration, etc.

[0061] By using a Video AET, such as that illustrated in FIG. 13, a content preview engine could read a portion or all of the AET data for each title to be previewed, and then decode and display the multiple AETs on one screen. Since AETs are designed to have a fraction of the processing needs of the main content, this multi-title animated preview becomes possible without having to increase the performance of the device.

Audio and Other Media AETs

[0062] In a similar fashion to Video AETs, other AETs can be created by following the same principles as those employed in creating the Video AETs. Such media tracks suitable for AETs include audio and subtitle tracks. Example methods for creating Audio AETs 170, 175, from an audio track are illustrated in FIGs. 16 and 17. FIG. 15 is an illustration of the audio track 160. In the illustrated embodiments, the audio track is sampled and then the samples are combined in AETs to enable playback of the samples in conjunction with the video AETs.

Generating AETs

[0063] In the case when a user wishes to add one or more AETs to a piece of content, then the process of creating and storing ("generating") the AET(s) will take time and processing power to perform. If this time and processing requirement are imposed on the user at the time when the user wishes to utilize the content, then this would not constitute a good user-experience since the user would be forced to wait before continuing with whatever operation that was initiated.

[0064] In one embodiment, the need to wait for the AET generation process can be removed or reduced by performing AET generation in parallel with another operation such as downloading, burning to disk, or first playback. Another embodiment would be to integrate the AET into a "background task" that executes when the user's computer is not being actively used by the user, thus allowing a user's personal catalogue of content to be processed while the user performs other applications with the computer.

[0065] In FIG. 18, one embodiment of performing AET generation is provided. The content is sub-sampled to a lower resolution (181) which in turn this lower resolution content is sub-sampled to a lower frame rate (183). The low-resolution, low-frame-rate is encoded as a video AET as a key-frame only bit-stream (185). The resulting data is included with the content (187), e.g., either written/embedded into the content or appended to the end of the content.

[0066] For example, through experimental testing, by taking 25% of the spatial data (pixels) and 21% of the temporal data (frames) from a source, nearly 95% of the original data can be discarded. The resulting frames are all encoded as key-frames, which are known to be extremely inefficient. However, since the source being encoded is only 5% the data volume of the original, it is has been discovered that the video AET file can be anywhere from a few percent to 10% the size of the original content (a general rule of thumb of 7.5% can be used in the estimation of the encoded video AET file-size).

[0067] The following example as determined through experimental testing is also provided to at least illustrate the visual search enhancement relative to the file size. A "normally encoded" (i.e., one or more key-frames per second) movie (i.e., 23.976 fps) of resolution 1920x816 has a file-size of about 10 GB. The movie is re-mastered with a key-frame rate of one or more key-frames every 4 seconds, reducing the file-size to 8.23 GB, i.e., a 17.7% reduction. The content is then sub-sampled to a resolution of 480x272 and a frame-rate of 5 fps (25% and 21% respectively) to generate an AET source. The AET source is then encoded as key-frames only, resulting in an AET file-size of about 618 MB. The combined file-size of the "best-encoding" with "visual-search enhancement" is 8.85 GB. This is a saving of 1.15 GB from the original file-size and includes improved visual-search performance. In addition, advanced video players and media managers can use the AET to show animated content previews. In this case, a device could perform up to "40x" visual-search (in either the forward or reverse time-line) without requiring more than "1x" original video system performance. Higher

speeds of visual-search can be achieved by skipping key-frames as needed to keep the system performance within the limits of the device (or software) performing the visual-search.

[0068] A playback system in accordance with an embodiment of the invention is shown in FIG. 19. The progressive playback system 190 includes a media server 192 connected to a network 194. Media files are stored on the media server 194 and can be accessed by devices configured with a client application. The media files are encoded content files coupled to or embedded with application enhancement tracks. In the illustrated embodiment, devices that access media files on the media server include a personal computer 196, a consumer electronics device such as a set top box 198 connected to a playback device such as a television 200, and a portable device such as a personal digital assistant 202 or a mobile phone handset. The devices and the media server 192 can communicate over a network 194 that is connected to the Internet 204 via a gateway 206. In other embodiments, the media server 192 and the devices communicate over the Internet.

[0069] The devices are configured with client applications that can request all or portions of media files from the media server 192 for playing. The client application can be implemented in software, in firmware, in hardware or in a combination of the above. In many embodiments, the device plays media from downloaded media files. In several embodiments, the device provides one or more outputs that enable another device to play the media. In one example, when the media file includes one or more application enhancement tracks, a device configured with a client application in accordance with an embodiment of the invention can use the AETs to provide a user with trick-play functions. When a user provides a trick-play instruction, the device uses the AETs to execute the trick-play function. In a number of embodiments, the client application requests all or portions of the media file using a transport protocol that allows for downloading of specific byte ranges within the media file. One such protocol is the

HTTP 1.1 protocol published by The Internet Society or BitTorrent available from www.bittorrent.org. In other embodiments, other protocols and/or mechanisms can be used to obtain specific portions of the media file from the media server.

[0070] In FIG. 20, one embodiment of utilizing the application enhancement tracks is shown. A media file is received or retrieved from, for example, a media server (211). The media file is examined to extract the application enhancement tracks (212). The user interface of the media player allows the user to interact with the media file, e.g., provide trick-play requests (213). For example, the user can indicate a desire to fast forward through the media content. As such, the media player accesses the AET and traverses the AET in a forward direction. In one embodiment, the media player determines the time and position of the content being played and locates the corresponding time and position of the AET relative to the content (214). In many embodiments, the corresponding position is located using timestamps within the AET. In a number of embodiments, the corresponding position is located utilizing an index. The media player from this location sequentially decodes or plays the frames in the AET until a user request to stop (215). Through 2x, 4x, etc. fast forward user requests, the speed in which the AET is decoded or displayed can also be varied by the user. Rewind requests operate in the same manner but in a direction opposite of the forward requests. At a user "play" request, the media player determines the time and position of the AET relative to the content and from this location sequentially decodes the frames in the content until another user request is received.

[0071] In many embodiments, locating a frame with a timestamp corresponding to a frame within an AET can involve locating a key-frame with the closest timestamp preceding the desired timestamp within an index contained within the multimedia file, decoding the key-frame and decoding the difference frames between the key-frame and the difference frame with the desired timestamp. At which point, the presentation can commence playing using the higher resolution content. In many embodiments, other

techniques are used to seamlessly transition from viewing low resolution content in an AET during a trick-play mode and the higher resolution tracks that contain the full content within the multimedia file.

[0072] In the case where timestamps are not present in the media file, e.g., audio video interleaved (AVI) files, locating the start point to play the higher resolution content is based on the position of the AET within the media file. In one embodiment, a timestamp although not present in the AET or content is derived from the frame count and the associated frame rate. Using this derived timestamp, a frame within the high resolution content that corresponds to the AET frame or closest AET frame and vice versa can be located.

[0073] Generally, application enhancement tracks are derived from the main content that they are associated with. They are typically encoded to aid the performance of one or more functions related to the content, such as visual-search, or content-preview, and can be stored in the same file as the main content, or in one or more separate files. AETs provide many factors of improved performance while incurring only a slight increase to the cost associated with the main content (storage, transfer speed, etc.). In fact, since the AET is a fractional cost of the main content, only that track may be needed to perform certain functions and can therefore reduce the overall cost of viewing, or otherwise interacting with, the content.

[0074] An AET is not tied to any single type of media data or encoding standard, and is in fact equally applicable to many widely used video standards (MPEG-2, MPEG-4 Part 2 and H.264) as well as widely available media formats (DivX, DVD, Blu-ray Disc, HD-DVD).

[0075] In several embodiments that implement the methods and systems described above, scalable speeds of visual-search can be conducted in both the forward and reverse directions while incurring additional file size costs of only 5% relative to the size

of the main content file. Furthermore, these same tracks can be utilized for content-preview animations.

[0076] Finally, it should be understood that while preferred embodiments of the foregoing invention have been set forth for purposes of illustration, the foregoing description should not be deemed a limitation of the invention herein. Accordingly, various modifications, adaptations and alternatives may occur to one skilled in the art without departing from the spirit and scope of the present invention.

WHAT IS CLAIMED IS:

1. A method of encoding a media file for playing back, comprising:

extracting a video track from an original media file, where content is encoded in the video track;

using the encoded content to encode an application enhancement track, where encoding the application enhancement track includes discarding at least some of the content; and

creating a media file including the content from the original media file encoded in a video track and an accompanying application enhancement track.

2. The method of claim 1 wherein information is discarded from the content by sub-sampling the content.

3. The method of claim 2 wherein sub-sampling the content further comprises reducing a resolution scale of the content by a particular factor to generate the application enhancement track.

4. The method of claim 2 wherein sub-sampling the content further comprises reducing a frame rate of the content by a particular factor to generate the application enhancement track.

5. The method of claim 4 wherein creating the media file further comprises embedding portions of the accompanying application enhancement track throughout the video track encoding the content.

6. The method of claim 4 wherein creating the media file further comprises incorporating the accompanying application enhancement track as a block within the media file.

7. The method of claim 1 wherein the application enhancement track contains only video.

8. The method of claim 1 wherein the application enhancement track contains key frames spaced at specific time intervals between each key-frame.

9. The method of claim 1, wherein creating a media file including the content from the original media file and the accompanying application enhancement track further comprises:

re-encoding the content to maximize compression regardless of a size or frequency of time periods between key frames to create a compressed video track possessing the same resolution and frame rate as the video track; and

creating a media file including the compressed video track and the accompanying application enhancement track;

wherein the media file including the compressed video track and the application enhancement track is smaller than the original media file.

10. The method of claim 1 wherein the content is compressed content.

11. A method of decoding a media file for playing back, comprising:

obtaining a media file containing compressed content and an accompanying application enhancement track which is a subset of the compressed content;

playing back the compressed content; and

decoding frames of the application enhancement track at a rate proportional to a visual-search speed and from a location determined by the portion of the compressed content most recently played back.

12. The method of claim 11 wherein the application enhancement track includes only key-frames.

13. The method of claim 11 further comprising at least one delta frame placed within a constant time difference to adjacent frames.

14. The method of claim 11 wherein only key-frames are decoded when a predetermined visual-search speed is exceeded.

15. The method of claim 11 wherein a select number of key-frames are skipped along with any associated delta frames when a predetermined visual-search speed is exceeded.

16. The method of claim 11 wherein the application enhancement track is part of the media file and is the compressed content at a reduced frame rate and at a reduced resolution.

17. A system for playback of a media file, comprising:

a media server configured to generate at least one application enhancement track from an original media file, the at least one application enhancement track having at least one frame in common with the original media file and being substantially smaller in file size than the original media file; and

a client processor in network communication with the media server and configured to send requests for the original media file to the media server, the media

server configured to transmit the requested original media file along with the at least one application enhancement track.

18. The system of claim 17, wherein the client processor further comprises a playback engine configured to decode the application enhancement track and to determine portions of the application enhancement track required to comply with user playback instructions.

19. The system of claim 18, wherein the application enhancement track contains only key frames.

20. A media player comprising:

a user interface configured to receive user instructions; and

a playback engine configured to decode a media file containing content and an accompanying application enhancement track, which is a subset of the content;

wherein the playback engine is configured to commence playback of the content in response to a user instruction received via the user interface;

wherein the playback engine is configured to select portions of the application enhancement track to decode and playback in response to receipt of a user instruction received via the user interface and the portion of the content most recently played back by the playback engine.

21. The media player of claim 20 wherein the playback engine is configured to commence playback of the content in response to receipt of a user instruction received via the user interface and from a location determined based upon the portion of the application enhancement track most recently played back by the playback engine.

22. The media player of claim 21 wherein the application enhancement track is part of the media file and is a sequence of key frames time stamped with the time stamps of the corresponding frames of the content.

23. The media player of claim 22 wherein the key frames are lower resolution relative to a resolution of the corresponding frames of the content.