US 20090307258A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2009/0307258 A1

(43) Pub. Date: Dec. 10, 2009

Priyadarshi et al.

(54) MULTIMEDIA DISTRIBUTION AND PLAYBACK SYSTEMS AND METHODS USING ENHANCED METADATA STRUCTURES

(76) Inventors: **Shaiwal Priyadarshi**, San Diego, CA (US); **Kourosh Soroushian**, San Diego, CA (US); **Jason Braness**, San Diego, CA (US); **Loren Kirkby**, San Diego, CA (US)

Correspondence Address:
KAUTH , POMEROY , PECK & BAILEY ,LLP
2875 MICHELLE DRIVE, SUITE 110
IRVINE, CA 92606 (US)

(21) Appl. No.: 12/480,251

(22) Filed: Jun. 8, 2009

Related U.S. Application Data

(60) Provisional application No. 61/059,547, filed on Jun. 6, 2008, provisional application No. 61/109,476, filed on Oct. 29, 2008.

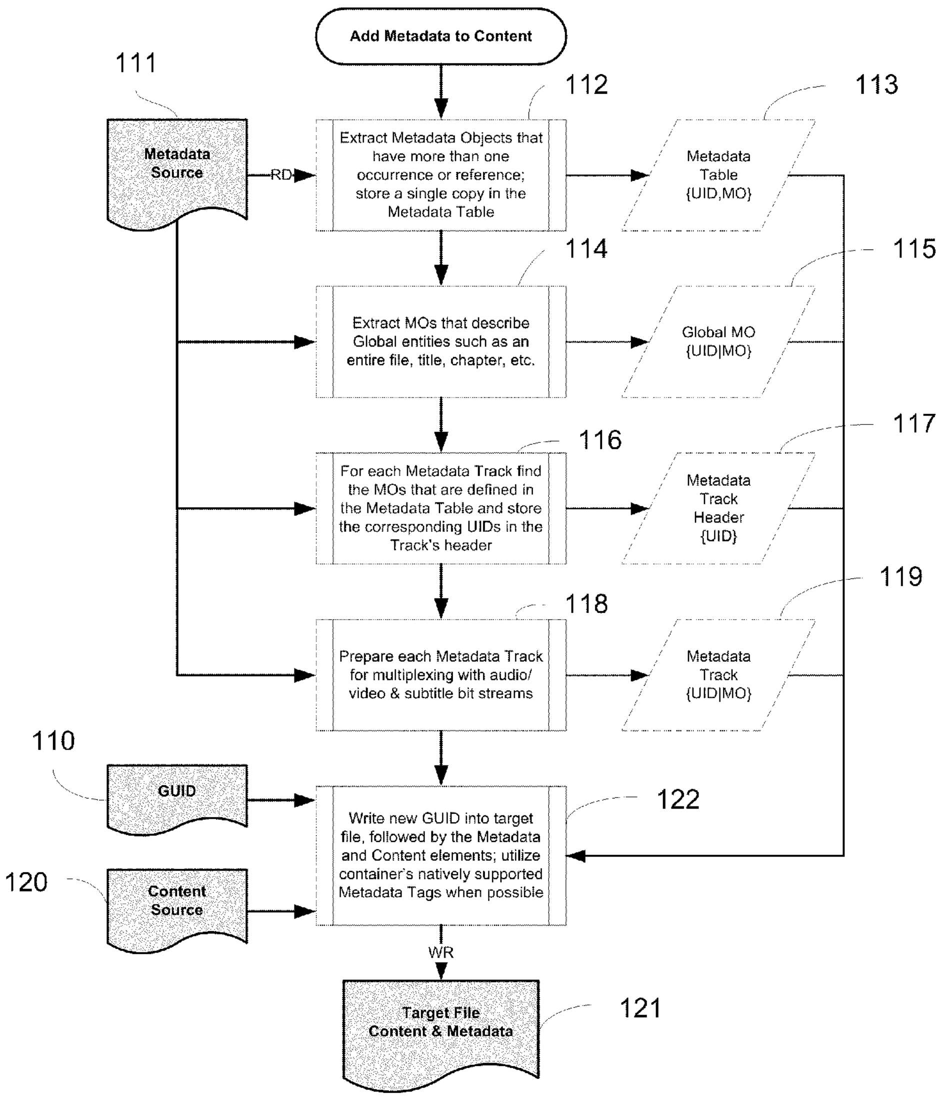
Publication Classification

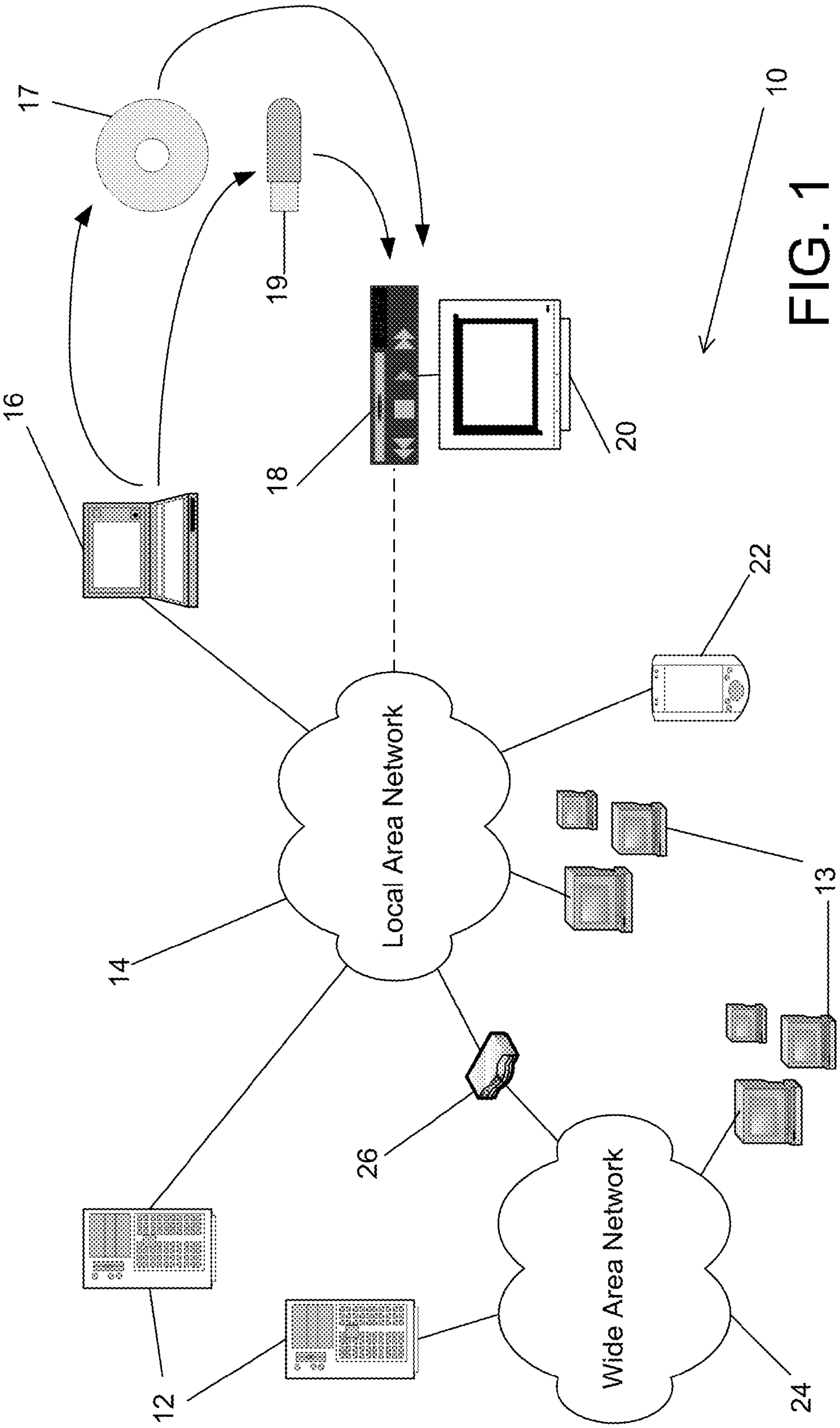
(51) Int. Cl.
G06F 17/30 (2006.01)

(52) U.S. Cl. .... **707/102**; 707/104.1; 707/103 R; 707/E17.009; 707/E17.055; 707/E17.01

(57) ABSTRACT

A metadata systems and methods are provided that enhance the playback features of multimedia files. A metadata structure is used that includes metadata tags and objects to allow access to various data typically not available to most playback devices.





10
12
13
14
16
17
18
19
20
22
24
26
Local Area Network
Wide Area Network

FIG. 1

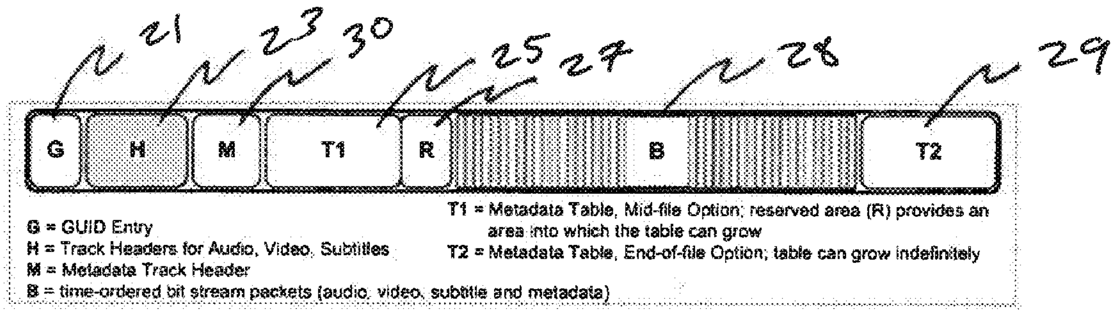


FIG. 2

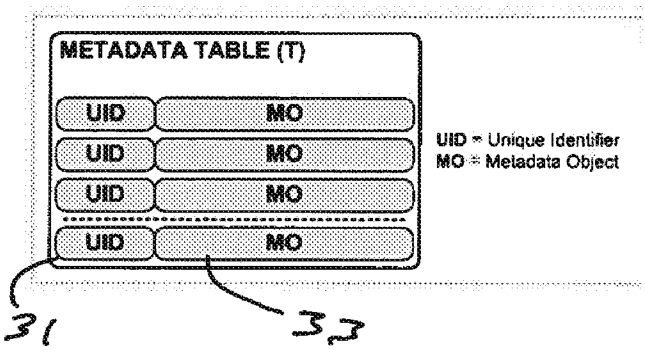


FIG. 3

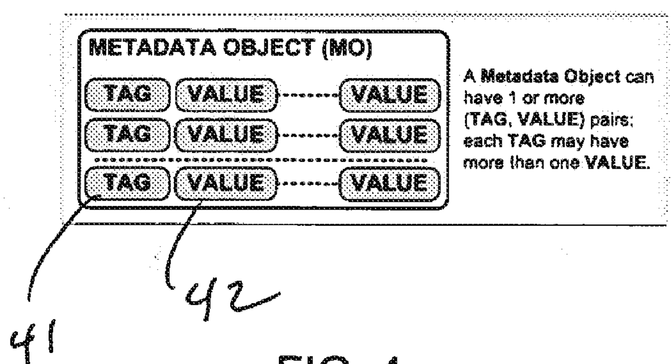


FIG. 4

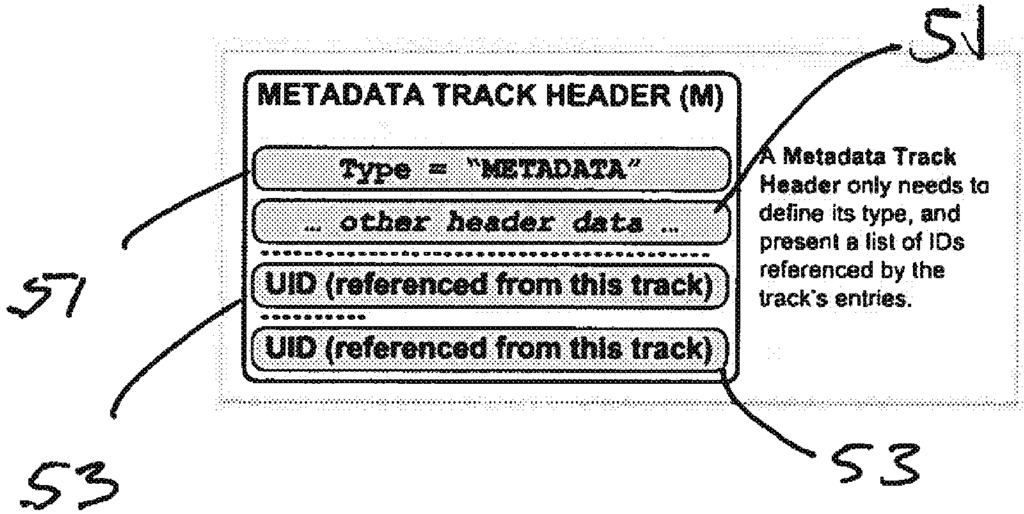

METADATA TRACK HEADER (M)
Type = "METADATA"
… other header data …
UID (referenced from this track)
UID (referenced from this track)
A Metadata Track Header only needs to define its type, and present a list of IDs referenced by the track's entries.
51
57
53
53

FIG. 5

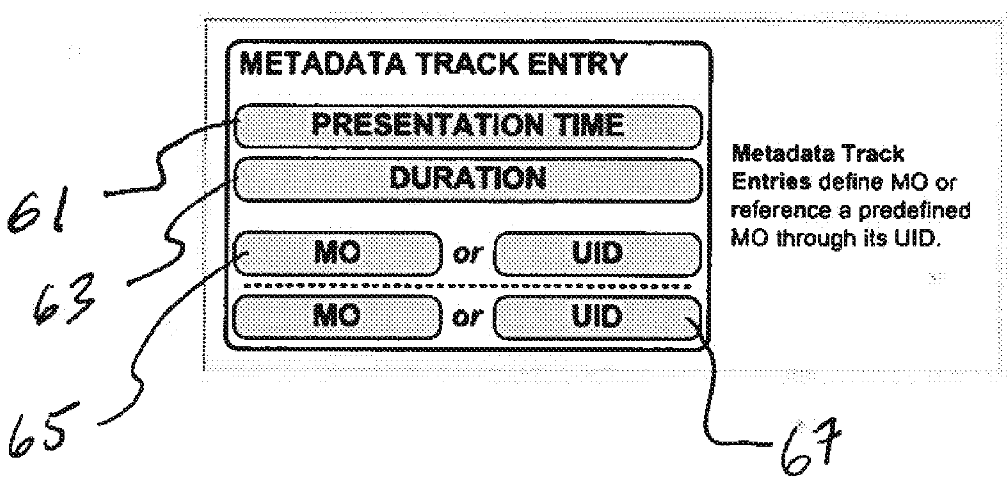
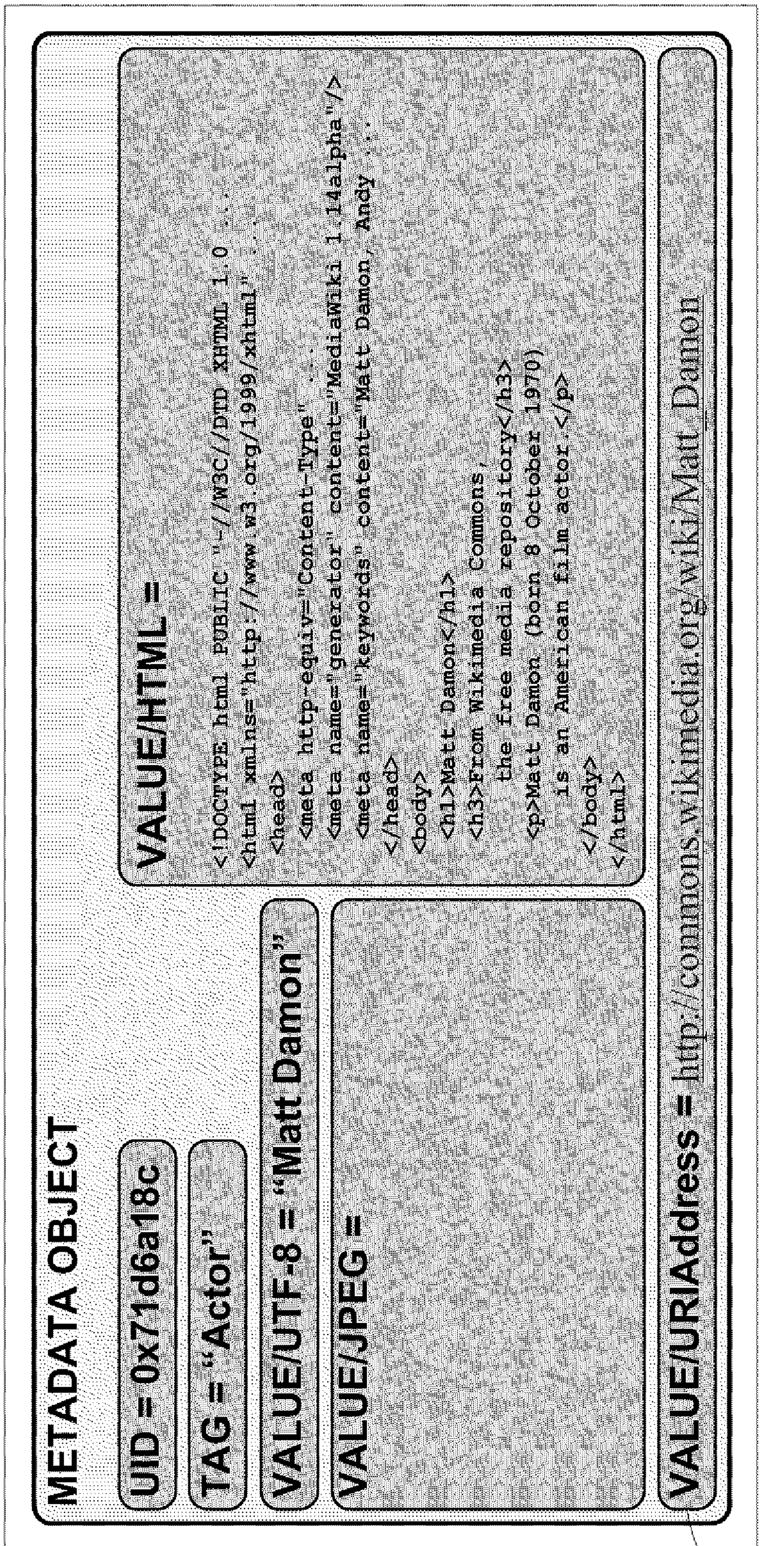

METADATA TRACK ENTRY
PRESENTATION TIME
DURATION
MO
or
UID
MO
or
UID
Metadata Track Entries define MO or reference a predefined MO through its UID.
61
63
65
67

FIG. 6


METADATA OBJECT
UID = 0x71d6a18c
TAG = "Actor"
VALUE/UTF-8 = "Matt Damon"
VALUE/JPEG =
VALUE/HTML =
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 . . .
<html xmlns="http://www.w3.org/1999/xhtml" . . .
<head>
<meta http-equiv="Content-Type" . . .
<meta name="generator" content="MediaWiki 1.14alpha"/>
<meta name="keywords" content="Matt Damon, Andy . . .
</head>
<body>
<h1>Matt Damon</h1>
<h3>From Wikimedia Commons,
the free media repository</h3>
<p>Matt Damon (born 8 October 1970)
is an American film actor.</p>
</body>
</html>
VALUE/URIAddress = http://commons.wikimedia.org/wiki/Matt_Damon

71

FIG. 7

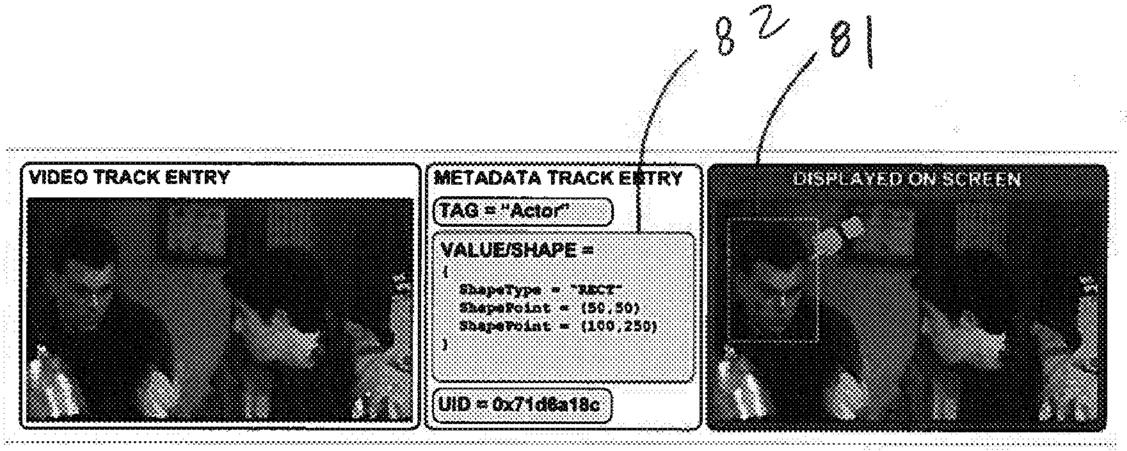

82
81
VIDEO TRACK ENTRY
METADATA TRACK ENTRY
TAG = "Actor"
VALUE/SHAPE =
{
ShapeType = "RECT"
ShapePoint = (50,50)
ShapePoint = (100,250)
}
UID = 0x71d6a18c
DISPLAYED ON SCREEN

FIG. 8

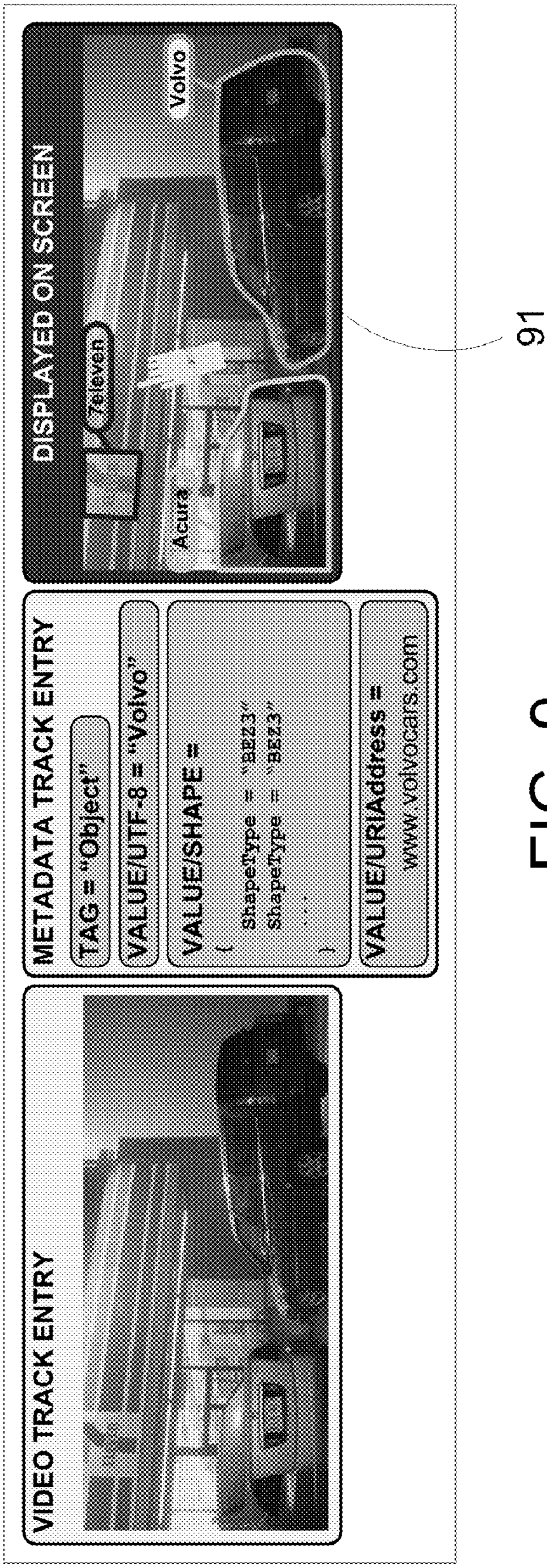

VIDEO TRACK ENTRY
METADATA TRACK ENTRY
TAG = “Object”
VALUE/UTF-8 = “Volvo”
VALUE/SHAPE =
{
ShapeType = “BEZ3”
ShapeType = “BEZ3”
...
}
VALUE/URIAddress =
www.volvocars.com
DISPLAYED ON SCREEN
Volvo
7eleven
Acura
91

FIG. 9

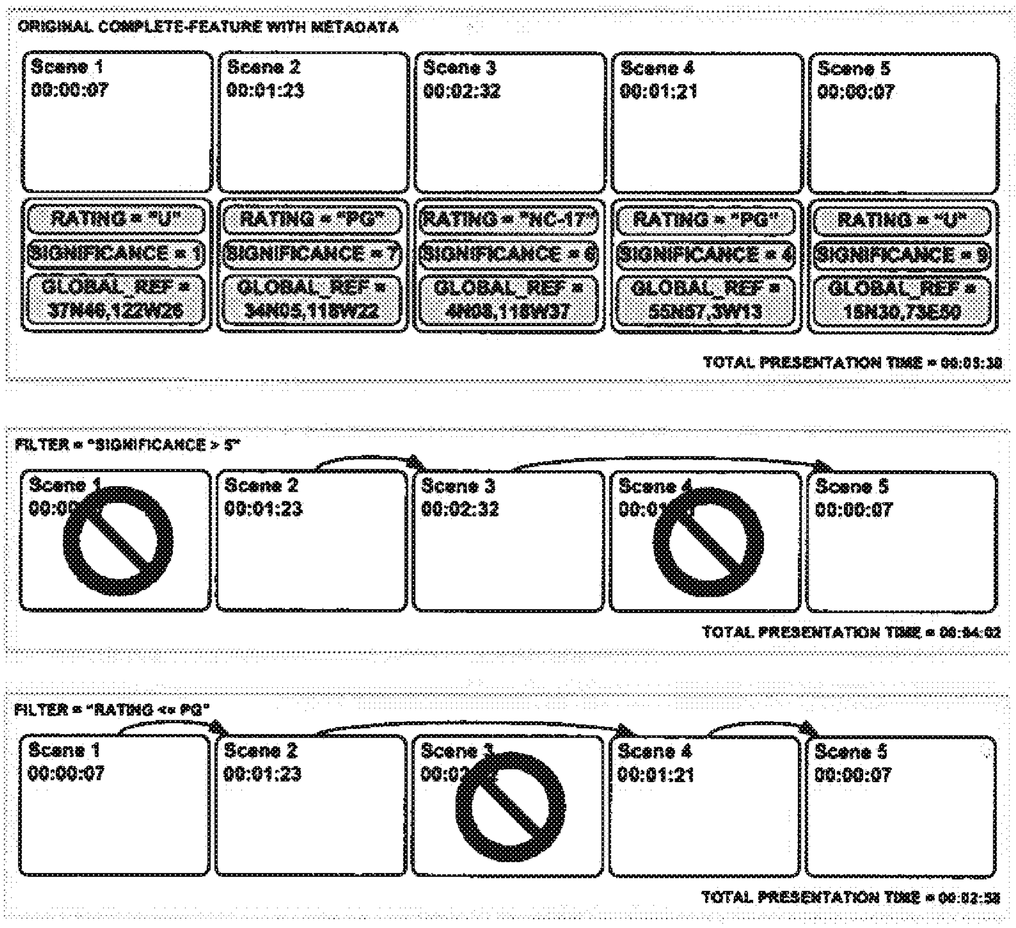

ORIGINAL COMPLETE-FEATURE WITH METADATA
Scene 1
00:00:07
Scene 2
00:01:23
Scene 3
00:02:32
Scene 4
00:01:21
Scene 5
00:00:07
RATING = "U"
SIGNIFICANCE = 1
GLOBAL_REF = 37N46,122W26
RATING = "PG"
SIGNIFICANCE = 7
GLOBAL_REF = 34N05,118W22
RATING = "NC-17"
SIGNIFICANCE = 6
GLOBAL_REF = 4N08,118W37
RATING = "PG"
SIGNIFICANCE = 4
GLOBAL_REF = 55N57,3W13
RATING = "U"
SIGNIFICANCE = 9
GLOBAL_REF = 15N30,73E50
TOTAL PRESENTATION TIME = 00:05:30
FILTER = "SIGNIFICANCE > 5"
Scene 1
Scene 2
00:01:23
Scene 3
00:02:32
Scene 4
Scene 5
00:00:07
TOTAL PRESENTATION TIME = 00:04:02
FILTER = "RATING <= PG"
Scene 1
00:00:07
Scene 2
00:01:23
Scene 3
Scene 4
00:01:21
Scene 5
00:00:07
TOTAL PRESENTATION TIME = 00:02:58

FIG. 10

FIG. 11

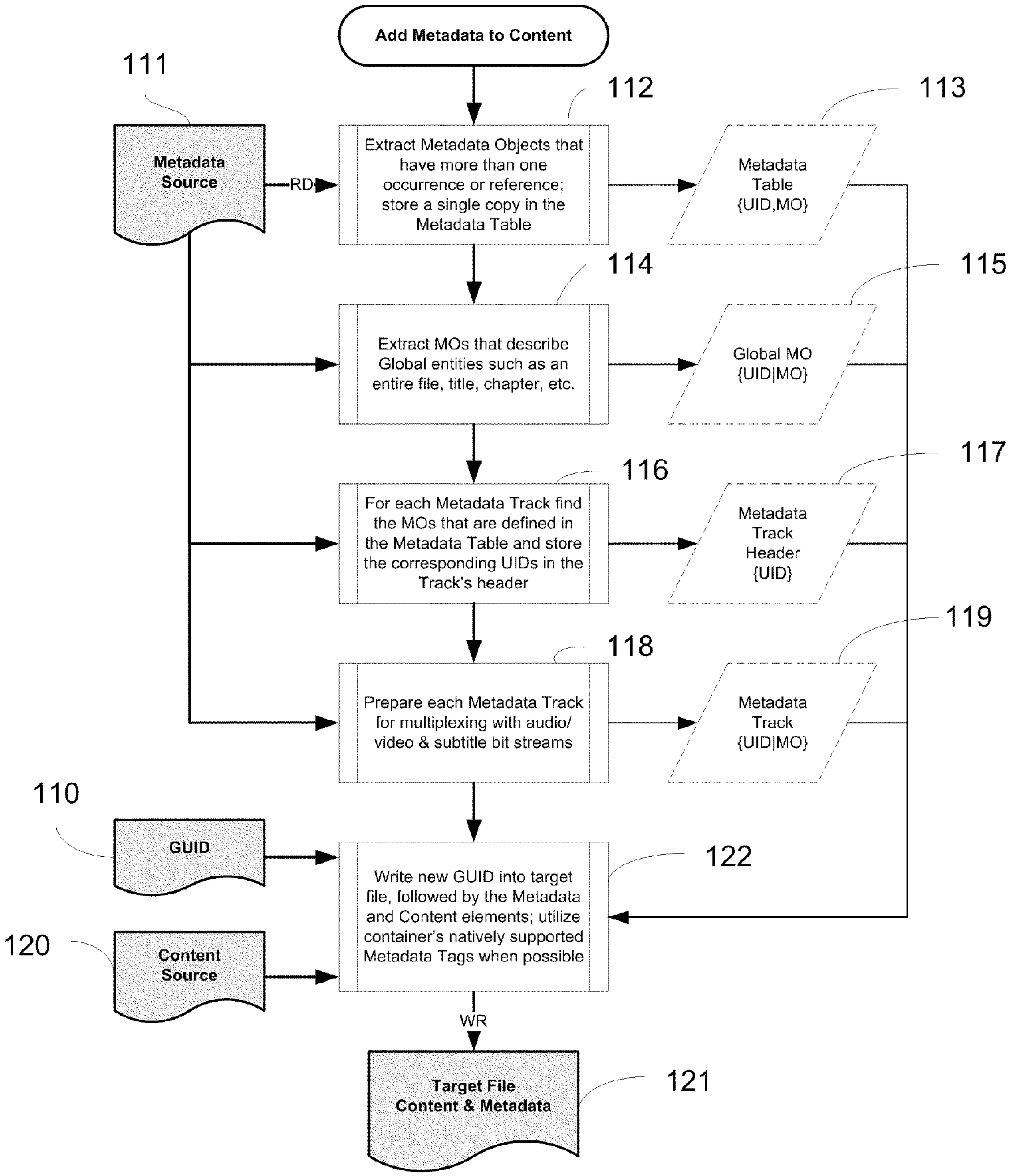
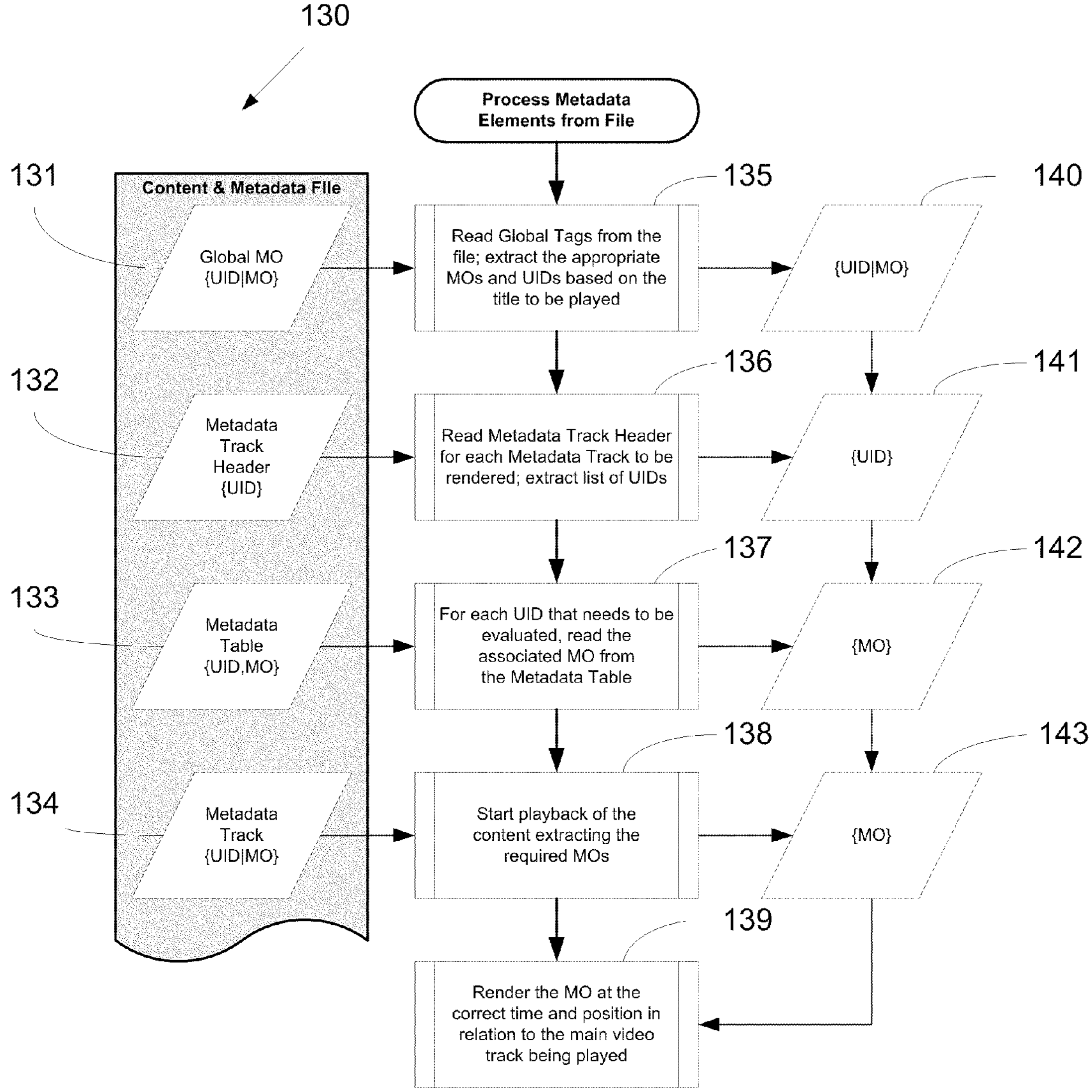

Add Metadata to Content
111
Metadata Source
RD
112
Extract Metadata Objects that have more than one occurrence or reference; store a single copy in the Metadata Table
113
Metadata Table {UID,MO}
114
Extract MOs that describe Global entities such as an entire file, title, chapter, etc.
115
Global MO {UID|MO}
116
For each Metadata Track find the MOs that are defined in the Metadata Table and store the corresponding UIDs in the Track's header
117
Metadata Track Header {UID}
118
Prepare each Metadata Track for multiplexing with audio/ video & subtitle bit streams
119
Metadata Track {UID|MO}
110
GUID
122
Write new GUID into target file, followed by the Metadata and Content elements; utilize container's natively supported Metadata Tags when possible
120
Content Source
WR
Target File Content & Metadata
121

FIG. 12



130
131
132
133
134
Content & Metadata File
Global MO {UID|MO}
Metadata Track Header {UID}
Metadata Table {UID,MO}
Metadata Track {UID|MO}
Process Metadata Elements from File
135
Read Global Tags from the file; extract the appropriate MOs and UIDs based on the title to be played
136
Read Metadata Track Header for each Metadata Track to be rendered; extract list of UIDs
137
For each UID that needs to be evaluated, read the associated MO from the Metadata Table
138
Start playback of the content extracting the required MOs
139
Render the MO at the correct time and position in relation to the main video track being played
140
{UID|MO}
141
{UID}
142
{MO}
143
{MO}

MULTIMEDIA DISTRIBUTION AND PLAYBACK SYSTEMS AND METHODS USING ENHANCED METADATA STRUCTURES

CROSS-REFERENCE To RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Patent Application Nos. 61/059,547, filed on Jun. 6, 2008, and 61/109,476, filed on Oct. 29, 2008, the entire disclosures of which are hereby incorporated by reference as if set in full herein.

BACKGROUND

[0002] Typical multimedia container formats offer practical and efficient methods of encapsulating standard multimedia data types such as audio, video and subtitles. The same efficiency, however, does not typically extend to metadata, especially in most consumer targeted multimedia container formats. Often the descriptive and interactive metadata associated with content is collectively placed in a distinct section of the same file, or stored in secondary files using proprietary formats. To date, practical implementations of metadata have been limited to simple descriptions of the video title, rarely extending to any direct associations with the actual scenes in the video. Moreover, in systems where secondary metadata files are employed, many challenges come to light when delivery occurs over the Internet due to factors such as the re-naming and re-grouping of files by caches between the publisher and the consumer.

[0003] In addition, to support the demands of Internet based video services, more and more metadata are being amassed in disparate systems to drive those services. The weakness with the methods currently employed by many of these Internet services is that the rich-experiences are only available through the hosted on-line service which, therefore, must be accessed through a web-browser. If the content can be downloaded from the provider's web-site, typically all of the metadata that enables the rich-experience cannot be. This has the effect of tying the content and the viewers to a PC-based experience rather than a home theater one, even when the home theater is the desired viewing environment. This limitation is a barrier for the wide-scale adoption of Internet distribution of TV, movies and other forms of multimedia by commercial content distribution networks. For large content providers and their customers to participate in an Internet based content distribution system, the signature experience of each provider must be able to migrate with the content to the viewer's home theater, in-car entertainment system and/or their mobile phone just as easily and vividly as it is viewable through their PC's web-browsers—regardless of whether the playback environment has an immediate, active connection to the Internet.

[0004] The requirements for a metadata system that can be applied to multimedia files are complex as the files may include a combination of video, audio and subtitle tracks. Furthermore, some multimedia formats, such as DVD, require the playback of the video presentation to follow an authored path, such as the displaying of copyright notices, trailers, chapter menus, etc. In the model of physical distribution of DVDs and Blu-ray Discs (BDs), direct associations between the authored presentation order and the multimedia files is maintained by the physical properties of the disc.

SUMMARY

[0005] Generally, digital video distribution and playback systems and methods that provide an enriched and versatile metadata structure are provided.

[0006] In one embodiment, a method of playing back metadata content stored in a media file comprises providing a media file to a playback device. The media file has at least one metadata object and an association with content data in which the metadata object references at least one facet of the content data. The method further comprises decoding the content data by the playback device, displaying content on a display screen from the decoded content data, and decoding the at least one metadata object based on the displayed content by the playback device.

[0007] In another embodiment, a system for playback of a media file comprises a media server and a client processor. The media server is configured to locate media files with each media file having an immutable global identifier. The client processor is in network communication with the media server and is configured to send requests for a media file to the media server. The media server is also configured to locate and transmit the requested media file based on the global identifier and the client processor further comprises a playback engine configured to decode a metadata track within the transmitted media file. The metadata track refers to content data in the transmitted media file.

[0008] In yet another embodiment, a method of creating a media file having metadata information comprises supplying a source of metadata information to an encoder; supplying a source of content to the encoder; generating a metadata object from the supplied metadata information by the encoder, the generated metadata object referencing at least one portion of the supplied content and integrating the metadata object with the supplied content to form a media file by the encoder.

[0009] In one embodiment, a single unique identifier is used to refer to a repeatedly referenced metadata object.

[0010] 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

[0011] FIG. **1** is a semi-schematic diagram of networked and local-file playback systems in accordance with embodiments of the invention.

[0012] FIG. **2** is a graphical representation of metadata structure within a multimedia file in accordance with an embodiment of the invention.

[0013] FIG. **3** is a graphical representation of metadata table in accordance with an embodiment of the invention.

[0014] FIG. **4** is a graphical representation of metadata object in accordance with an embodiment of the invention.

[0015] FIG. **5** is a graphical representation of metadata track header in accordance with an embodiment of the invention.

[0016] FIG. 6 is a graphical representation of metadata track entry in accordance with an embodiment of the invention.

[0017] FIG. 7 is a graphical representation of metadata object in accordance with an embodiment of the invention.

[0018] FIG. 8 is a graphical representation of metadata track entry relative to a display and a video track entry in accordance with an embodiment of the invention.

[0019] FIG. 9 is a graphical representation of a metadata track entry relative to a display and a video track entry in accordance with an embodiment of the invention.

[0020] FIG. 10 is a graphical representation of metadata structure within a multimedia file in accordance with an embodiment of the invention.

[0021] FIG. 11 is a flowchart of a process encoding a multimedia file to include metadata information in accordance with an embodiment of the invention.

[0022] FIG. 12 is a flowchart of a process decoding a multimedia file having metadata information in accordance with an embodiment of the invention.

DETAILED DESCRIPTION

[0023] Generally, a rich metadata structure for multimedia files is provided that increases the scope of metadata tags and fundamentally enhances the capabilities of media-managers and players on both personal computer and consumer electronic (CE) platforms. In one embodiment, multimedia metadata systems and methods are provided that enable associations to be maintained by immutable logical properties that remain robust to changes to the mutable logical properties of the data, such as the file names and paths. The systems and methods allow description of audio and video, as well as subtitles and any other types of presentation data tracks within a file. The content can be contained in a single file, or be distributed across a multi-segment range of files. In addition, many embodiments of the invention support both the DVD experience of authored presentations, as well as the Internet-based dynamic and collaborative experience. In several embodiments, the metadata system maintains the same level of experience across PC and embedded platforms, regardless of whether the player has a live Internet connection or not. In a number of embodiments, the metadata can be associated with the content, regardless of whether the metadata is stored in-file with the content, or in another file. In addition, the metadata can describe a variety of entities and properties of the content including the entire file, each feature in the file, each chapter of the features, segments or segments of those chapters and even spatial regions of the video.

[0024] Metadata frameworks in accordance with embodiments of the invention utilize three items of support from the containers that incorporate it. First, the ability to store a Globally Unique Identifier (GUID) or Universally Unique Identifier (UUID) for the file. Second, the ability to store a table of metadata tags, values and UIDs using common and new data types and lastly, the ability to store a new type of multimedia track, with a non-standard data type, a "metadata bit stream track". It should however be appreciated that one or more of the items, e.g., the first and third items, can be optional items to be used in more advanced cases or devices. Use of one or more metadata bit stream tracks enable metadata to be available in close proximity within the file to the content that the metadata describes, as well as delivering the metadata only when/if it is needed.

[0025] The metadata table enables the efficient, singular, storage of metadata multiply referenced by metadata tags contained in the Media File, including those in any contained metadata bit stream track. The use of a GUID allows the content of a Media File, including any metadata and metadata bit stream tracks, to change without breaking references made to it from other, previously authored, Media Files.

[0026] The metadata format system extends the scope of metadata tags (or just "tags") from traditional coarse-grain descriptors to medium and fine-grained descriptors of subsections of the file and individual video frames. The system introduces some new data types for the tag values that enable the demarcation and outlining of spatial regions of interest and support linking with internal and external objects. The system also increases the robustness of and options for, content distribution across the Internet. The system can be utilized by applications to enable them to, for example, regulate playback of content in many ways, allow annotation of the content by groups of viewers, and/or redirect playback to remote content, such as adverts or special-features, hosted on web-sites. These extensive enhancements to conventional metadata tagging opens-up options for application functionality and enables the creation of a wide set of portable, rich, commercial and non-commercial content based services.

[0027] In one embodiment, the metadata structure is largely in-line with those of MPEG-7 and the data-types similar to those defined in the SMPTE Metadata Dictionary. In doing so, it is straightforward to provide translation services between these professional audiovisual metadata standards and the simpler consumer oriented format.

[0028] Referring now to FIG. 1, playback systems in accordance with embodiments of the invention are shown. The playback system 10 includes media servers 12 and metadata servers 13 connected to a LAN (e.g., a home network) or a WAN (e.g., the Internet) 14. Media files are stored on the media servers 12 and metadata resource databases stored on the metadata servers and can be accessed by devices configured with a client application. In the illustrated embodiment, devices that access media files on the media servers and metadata on the metadata servers include a personal computer 16, a consumer electronics device such as a Media File Player 18 connected to a visualization device such as a television 20, and a portable device such as a personal digital assistant 22 or a mobile phone handset. The devices and the servers 12 can communicate over a LAN 14 that is connected to the WAN 24 via a gateway 26. In other embodiments, the servers 12, 13 and the devices communicate over a WAN (such as the Internet).

[0029] In some embodiments, the Media File Player 18 is directly connected to a LAN and can directly access a WAN. In some embodiments, the Media File Player is not directly connected to a network and plays files that have been copied onto Optical Disks 17, USB thumb-drives 19 or other direct-access physical media. In such embodiments, the software that copied the media files to the physical media (e.g., running on a computer 16) copies the media files from the media servers and the metadata from the metadata servers to the physical media. When copying the metadata from the metadata servers, the copying software can translate metadata values that refer online resources to the location of the resource on the local media.

[0030] The devices are configured with client applications that read portions of media files from the media servers 12 or physical media 17, 19 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. When the media file includes an index, a device configured with a client application in accordance with an embodiment of the invention can use the index to determine the location of various portions of the media. Therefore, the index can be used to provide a user with "trick play" functions. When a user provides a "trick play" instruction, the device uses the index to determine the portion or portions of the media file that are required in order to execute the "trick play" function and reads those portions from the server or physical media.

[0031] In a number of embodiments, the client application requests portions of the media file from media servers 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.

Incorporation of Metadata within Media Files

[0032] In several embodiments, a media track for the incorporation of metadata throughout the duration of the content is provided. With having metadata tracks embedded into a file, many options open-up for the utility of these tracks. It should be noted that a metadata track is different from video, audio and subtitle tracks in typical media files. For example, the metadata track refers to the other tracks and thus is only relevant through the use of the other tracks. In other words, the metadata track would appear as pieces of detached information without the context of the other tracks. Additionally, the metadata track can refer to other information, i.e., metadata information. This for example allows the information to be stored and/or extracted from another source or location within the media file or referenced multiple times but stored in a single or just a few locations. These differences and other such distinguishing and additional features are described in greater detail below.

[0033] FIG. **2** illustrates the metadata structure within the context of a typical multimedia file. GUIDs **21** are used to identify files, with a new GUID created for every new file (ideally, even for a copy of a file, since a copy is a new file). Once defined at file creation time, the GUID should be regarded as an immutable property of that file. Even if the file's contents change its GUID should remain the same. This rule allows a "main title" to be authored with a reference to a "placeholder" file via the GUID contained in the placeholder file; the "placeholder" file's contents could change to accommodate a new movie trailer or advertisement, or other form of content. Whenever the main title makes a reference to the "placeholder" file's GUID, it would receive the latest video encoded within the "placeholder"; hence the viewer's copy of the "main title" need never change physically, though it would always link to "current" content.

[0034] Track headers **23** for audio, video, subtitles, etc. along with a metadata track header **30** follows the GUID **21** entry. In one embodiment, a metadata table **25** and/or metadata table **29** follows the metadata track header **30**. Metadata references in accordance with this invention can refer to metadata objects previously stored within one or more of the metadata tables. In this way, frequently referenced metadata objects can be referred to efficiently through the single metadata object residing in a table, rather than repeating the declaration of the object whenever it is needed. As metadata can change quite frequently, a metadata table is written into the file at a position where changes to the table's size do not require the entire multimedia file to be remuxed and rewritten. For example, the first metadata table **25** is a mid-file option and has a reserved area **27** that provides an open area to allow growth of the first metadata table if needed. The second metadata table **29** is an end-of-file option that allows the growth of the metadata table to be indefinite or unrestricted. Thus, as shown in FIG. **2**, the tables could be stored as the very last element in a file, or could be embedded within a file with an immediately following reserved area (R)into which the table could grow. In some embodiments, it can be useful to utilize both areas for a distributed table. In some embodiments there may not be a reserved area (R), in which case changes to the size of the metadata table may require a rewrite of the entire file. One embodiment of a metadata table is shown in FIG. **3** and sets forth a set of metadata tags and values, a metadata object (MO) **33**, assigned to an unique identifier (UID) **31** that is unique within the set of all metadata UIDs in the file. FIG. **3** is discussed further below.

[0035] Time-ordered bit stream packets [audio, video, subtitle and metadata] **28** are usually located in files that also contain metadata tags and metadata tables. However, it should be appreciated that the packets or portions thereof could be in separate files, e.g., a video file and a metadata file, automatically linked to each other through the use of the GUIDs or manually linked through user specification. For example, a first GUID could reference the metadata file and a second GUID could reference the video file. Within the metadata file, the second GUID would be included to link the video file to the metadata file. Likewise, the video file would include the first GUID to link it to the metadata file. As such, a multimedia presentation can be a collection of files linked by GUIDs. Additionally, by linking the files with GUIDs, if a playback device first attempts to playback the metadata file when it intended to playback the video file, the playback device can retrieve all the associated files as desired to playback the intended multimedia presentation. In one embodiment, the GUID is not included or not used when all the presentation components, e.g., the metadata and audiovisual data, is placed in the same file.

[0036] Typically, the bit stream data in (B) is differentiated by a track ID, and each track's data type is defined by a Track Header **23**. Audio, video and subtitle tracks are well-defined data types for multimedia files, as are the track header **23** for these data types. The metadata track (metadata packets within the bit stream data **28**) places the definition or references of the metadata objects close to the associated audio, video and subtitle packets. For example, if a car appears on screen at nine minutes and nineteen seconds into a presentation and metadata tags that detail the car's make, model and associated web-site are to be presented to the viewer at the same time, then the metadata tags are placed in a packet in the (B) bit stream element, physically near the audio/video packets with the same presentation time. Alternatively, the metadata could have been placed in a separate entity unrelated to the nearest bit stream (B) element. In such, typical, cases, such a list of metadata objects and presentation times would be analyzed before playback, and the objects retained in memory until it was time for them to be displayed. Such an alternative scheme could, therefore, slow start-up performance and may require a potentially large amount of memory to be available in a playback system.

Metadata Tables

[0037] Referring again to FIG. **3**, a unique ID **31** (UID, as opposed to a Globally Unique ID) allows the referencing of the defined set of metadata **33** at a future point by using only the UID, rather than by an entire declaration (re-declaration) of the tags and values. The metadata table as such provides the definition of a UID-based look-up table of metadata tags and values; a minimum size for a file's metadata, by defining multiple referenced metadata objects (MO) only one time; and a minimally invasive manner to change the properties of an object that is referenced at multiple points.

Metadata Objects

[0038] In FIG. **4**, one embodiment of a metadata object (MO) is shown. The MO may be overloaded with multiple tags **41** and values **42**, with each value of a different data type. As such, the metadata object can have one or more [tag, value] pairs and each tag can have one or more associated values. This ability to associate multiple tags and values gives the metadata systems a wide-range of options to describe aspects of the audiovisual presentation. As a result, versatility is provided by allowing the systems that may be able to understand, handle and display complex metadata to utilize all the metadata information, whereas other systems that may only have implemented the mandatory, basic, tags and data types (native to their format) are still able to utilize the basic metadata information. As such, the tags defined by possibly overloaded values allow metadata to be defined in a scalable manner, where all players will be able to retrieve and offer at least one value for a tag, and more complex and capable players will be able to offer many different values for a tag. Hence, multimedia files can be authored with complex applications described through rich metadata to be utilized by high-end players, and these same files will still function to an approximate degree on much simpler players, such as low-cost CE devices.

Metadata Tracks

[0039] In one embodiment, the metadata track header (M) **30** is a data type for packets in the bit stream **28**, and is illustrated in FIG. **5**. In addition to the various implementation-specific definitions required of headers **51** (such as a "Type" parameter), each metadata track header also lists the UIDs **53** from the table metadata table that the track's entries reference. At a minimum, the metadata track header only provides its type and a list of unique identifier UIDs referenced by the track's entries. This allows a playback system to locate and store only the metadata objects from the metadata table that would be required to display that particular track fully. In many embodiments, the metadata objects are declared in-stream in the track entry, if desired by for example the publisher, and thereby bypassing the metadata tables.

Metadata Contained within Media Files

[0040] Portions of metadata that are interleaved in the audio and video data, (i.e., bit stream **28**) in accordance with an embodiment of the invention are shown in FIG. **6**. The metadata track entry provides a presentation time **61**, duration **63** and one or more metadata objects **65** or associated unique identifiers **67**, (i.e., references to predefined metadata objects). In one example, specific tracks can be written that contain metadata for different purposes, e.g. a track could incorporate all the director's annotated, multimedia commentary presentation regarding the shooting of a movie, while another track could contain the lead actress's multimedia presentation on her character with links to further details on web-sites. Viewers would have the option of enabling or disabling as many tracks as they choose in order to see exactly the types of data in which they are interested. As such, commentary from each viewer of the movie could be incorporated into separate tracks and during playback, only the current viewer's selected user tracks (for example, the viewer's friends' tracks) could be enabled and rendered for the viewer.

Data Types for Rich Tags and Rich Experiences

[0041] Typing metadata data values in a highly portable manner can result in metadata that is compatible with many different applications and devices. In various embodiments, the metadata types utilized by the metadata structure are found typically in a standard container format, e.g., the MKV format. The incorporation of standard data types enables devices with limited capabilities to simply decode the standard data and more advanced devices to access the full array of metadata, when the advanced data types are incorporated. The metadata types are thus recognized by the advanced "rich featured" applications and devices and thereby allow such applications to access the metadata information and present the metadata information appropriately. Non-advanced applications and devices may or may not recognize the metadata types and ignore the information. In other words, if the device recognizes that there is metadata information that its capability is limited to present, the device skips the metadata information, instead of attempting to display the metadata information that may cause the application to shutdown or stop the playback of the content.

[0042] Accordingly, metadata systems in accordance with a number of embodiments of the invention incorporate standard data types, such as 32-bit integers and UTF-8 encoded text strings. The metadata systems can also introduce some extra data types that can provide support for rich applications. These data types enable a richer variety of metadata to be contained within a file and can include portable formats for images, sounds, URIs, formatted text and Region of Interest (ROI) demarcations.

[0043] Additionally, in a number of embodiments, the richer metadata is stored using standards based formats that are commonly supported in consumer electronics devices. For images and sounds, widely supported compression formats can be utilized such as JPEG for images and MP3 for sound. Both of these formats are well established in practically all DVD, BD and PC platforms and are available from many vendors for many other platforms. Since both formats allow storage of data in the compressed domain they offer a simple yet scalable solution to store images and sounds, despite their general utilization for data that they may not have originally been designed to represent. For example, the JPEG standard is primarily targeted at compressing "natural" images, although in the context of the present invention its use is extended to allow storage of CGI images, that are typically better suited to storage in the GIF format, for example.

[0044] It should be appreciated that some multimedia formats, e.g., MKV, have practically no requirement to order data in the file according to any scheme to allow files to be created in the way that best suits the file writer. However, the onus is on the playback device to correctly retrieve all elements of the file needed to playback the content. However, this can burden the player to perform a significant amount of initial processing before it can playback the content, such as repetitively searching through lists. As previously described and described below in greater detail the metadata structure in accordance with various embodiments, such as using a single unique identifier to refer to a repeatedly referenced metadata object, can significantly reduce the processing burden on the player.

Embedding Dynamic Objects Using Metadata

[0045] Referring now to FIG. **7**, incorporation of URIs **71** into the metadata structure in accordance with an embodiment of the invention is shown. The incorporated URIs, e.g., addresses and/or client arguments, allow for the invocation of local or remotely stored dynamic objects. For example, the URIAddress of "http://commons.wikimedia.org/wiki/Matt_Damon" could be defined in an "Actor" tag for any scene in which the Hollywood actor Matt Damon appears. The contents of the referenced page can be created dynamically, based on the server serving the page. Hence, although the tag value may be set once and engraved into master discs, its invocation on a player would allow the most up-to-date information to be displayed, directly off the Internet. To support playback devices that do not have an Internet connection, or where the connection is temporarily unavailable (e.g. during a flight), the tag could be overloaded with another value of HTML type; this value would contain a copy of the HTML that was current at the time of writing.

[0046] The URI information in one embodiment indicates server and client components. In this manner a playback device can interpret the location or resources used to display or provide the metadata information. As such, an remote address, e.g., a server address, can be provided in the URI to indicate a remote server and/or database that hosts the associated object. Likewise, a local address, e.g., a client address, can be provided to indicate a local client and/or database to retrieve the requested information. For example, a selection of an actor or another indicator can cause the playback device to seek the information remotely (e.g., via the Internet) if indicated by a server address by the metadata information or locally (e.g., via a local drive or local area network) if indicated by a local address.

[0047] Furthermore, to facilitate specifying client-side processing rules a secondary string can also be defined that contains parameters and values that are intended to be processed by the playback device rather than the remote server. In some embodiments the metadata tag "URIClientArgs" contains the set of client-side parameters and values.

Conversion of HTML for Display Using on Simple Devices

[0048] Formatted text of the referenced page is typically supported through HTML V4.01 (or more recent version) which specifies a complex language for the markup of text and images. However, this complexity can also be an inhibitor to wide-scale support on CE devices. Hence, to simplify the implementation to realize use of the metadata information in full, CE devices do not have to process all aspects of HTML V4.01. For example, a CE device could reduce its implementation complexity by rendering text using a single embedded compacted font, as described in US Patent Application entitled Systems and Methods for font file optimization for multimedia files, filed on Jun. 6, 2009, thereby reducing the need for the device to have access to as many fonts as a typical web-browser; this method could be used when encoding any text as metadata for any tag. Furthermore, the playback device could limit native support for images to the JPEG format rather than all image formats supported by typical web-browsers. This allows a more complex "metadata server" to translate complex HTML into simplified versions where all the potentially non-supported features of HTML can be translated into a combination of JPEG and "simplified" HTML. This scheme guarantees that practically any translated page is viewable on any platform that incorporates a metadata player that embodies aspects of this patent application. Hence, metadata services can take complex HTML directly off the Internet and translate them into a simplified version for inclusion within the metadata value, offering a much richer mechanism for tag evaluation.

Regions of Interest

[0049] Metadata structures in accordance with a number of embodiments of the invention also include a set of data types for the definition of Regions of Interest (ROI), which can be used to visually highlight objects in the decoded video track and connect those objects with metadata tags. This "visual tagging" allows the development of many rich applications that can link the visual objects in the content with static or dynamic values. For example, in movies where the supporting roles are played by lesser-known actors, the scenes with the supporting actors could have an "ROI" line drawn around each actor's face, with a URI defined to connect the outline back to that actor's page on a suitable web-site, or local file-system resource (see discussion above with respect to using metadata to link to dynamic objects). In FIG. **8**, an example using a basic rectangle shape **81** is shown and in FIG. **9** a more complex shape **91**.

[0050] To implement ROIs in a portable manner, one mandatory data type and several optional types are defined. The first data type **82** is that of a "Bounding Area," which is intended to define a shape, e.g., a rectangle that fully encloses the object to be connected with a metadata tag. This object is simple in its definition and is intended to be simple to implement. It can have one or more tags associated with it, and like all other metadata tags, each tag could be overloaded with multiple values. In a number of embodiments, the ability to decode a basic ROI shape is supported on a large number and variety of devices and provides a baseline level of support. Therefore, media files that define ROIs can first define the ROI using the basic shape and can also more precisely define the ROI using more complex shapes supported by a smaller subset of devices.

[0051] A variety of increasingly complex shapes are also provided to allow the drawing of more complex and accurate outlines of arbitrarily shaped objects. These extended and optional types include but are not limited to shapes that can define: rectangles, ellipses, Bezier curves and multi-segment polygons. Each of these objects is defined using a minimum set of variables required in one embodiment. For example, a rectangle is defined by the coordinates of its opposite corners; ellipses by their center and a corner of its major and minor axis; and a quadratic Bezier curve by its two end-points and the control-point. The metadata structure allows each playback device to implement as many of these shapes as they can; using whatever algorithm is best suited to the platform.

[0052] A variety of different algorithms for drawing each of these shapes on PCs and embedded processors to determine the feasibility of including these complex shapes in the metadata structure have been explored. Results have shown that even low-end 200 MHz RISC processors are capable of drawing thousands of lines per second without hardware assist, and this result can be immediately translated to each of the complex shapes that can be drawn through a series of straight lines.

[0053] This set of one mandatory and multiple optionally implemented data types for ROI definitions allows for a very high degree of portability of the object demarcation system. For example, very simplistic players that cannot implement any of the "higher" drawing primitives can still provide very useful functionality through the "Bounding Area" object, by drawing a simple square or circle in the center of the Bounding Area. In extension, a complex player could utilize any further complex shapes to accurately trace the outline of a vehicle and link that object back to the web-site for that car, or if the device was personalized, to a dealer local to the viewer, as shown for example in FIG. **9**.

Tags for Rich Applications

[0054] In various embodiments, to enable the creation of other dynamic and rich applications, a set of identifiers or tags are provided to mark and record the data utilized to implement such applications. The tags provide an indication of the metadata information associated with the tag and thereby allows the playback device to decode and utilize the metadata information through established rules defined for the information. The tag also eases searching and filtering to locate specific metadata information.

[0055] The following are examples of these tags. For example, a COMMENT tag allows the association of a Unicode text string with a media file object, or a time-period, or an object in the presentation. A DISPLAY_ORIGIN tag indicates a rectangular 4:3 aspect ratio crop to be applied to a 16:9 (or 2.35:1, or other aspect ratio) video when displaying on a 4:3 display. DISPLAY_SETTINGS is a data structure that can be used to alter display characteristics during playback. DIVX_UID is a data structure that can be used to uniquely identify the file, each video track or each audio track. GLOBAL_REFERENCE is a data structure for the recording of GPS (or other)coordinates, height, direction, time, etc. OBJECT is a data structure for the description of a non-living entity in a scene. RATING is used to indicate the MPAA (or equivalent) rating assigned to an entire title, scene or frame of data. It can also be assigned globally to a track and individually to specific metadata objects; i.e. its purpose is contextual. RECORDER SETTINGS can be used to store values from an electronic recording device such as a video camera, DVR, etc.; typically the data is an EXIF table. SIGNIFICANCE is used to indicate the relevance of scenes to the overall storyline of a title. VIEWED_SEGMENTS allows the tracking of the time-periods that have been watched; each tag contains a counter also, indicating how many times that portion has been watched.

[0056] In some multimedia formats, tags are descriptions associated with other objects, e.g., editions, chapters, or tracks. As such, they do not have an identity of their own and, therefore, there is no way to independently refer to the value of a tag. In one embodiment, however, a tag can be defined with an identity, e.g., a TagUID, and a reference made to an identified tag through the a reference to the tag's identity, e.g., TagUIDReference. These extensions allow a single object to be defined once and then referenced as needed from other metadata definitions. The extensions are also useful in that file size can be reduced by providing references to handle metadata that is multiply-dispersed throughout a file. Also, the modification of a multiply-dispersed tag can be reduced to a single modification. In one embodiment, where a metadata track incorporates one or more references to an identified tag, the list of such references is placed within the metadata track's header, e.g., inside a TracksTagsList table.

[0057] In one embodiment, instead of extending the list of defined data types for a file format, existing tag names are extended to include the definition of the actual data type of the value that is stored inside a structure encoded into the file using one of the format's natively supported data types. For example, when TagName="TITLE", it can be assumed that the data type of the tag value will be a string and that the TagString is present with an UTF-8 encoding of the content's title. However, the base TagName is extended by adding a forward-slash "/" and then a set of characters that uniquely specify the data type of the value. For instance, if the cover art for a title is to be stored as metadata, and the cover art is in the JPEG format, then the tag "TITLE" will be extended with "/JPEG" and the TagBinary value will hold the binary data of the image. Also, the extensions used in one embodiment closely match [if not accurately match] the file extensions given to files of the same type. This allows for the development of advanced metadata viewers that can utilize a system's native data type handlers, by invoking the handler based on the TagName extension.

Applications Enabled by Rich Metadata

[0058] The following descriptions of rich applications provides some exemplary use-cases for the metadata structure. These are a few example applications and there are many more uses derivable from the correct and full utilization of the metadata structure provided.

Authored Versions

[0059] The metadata structure allows different versions of content to be authored into a single file, with each version being selectable by the user. This functionality is most analogous to a multi-DVD set containing individual DVDs of the "Studio Cut," "Director's Cut" and "Unrated" versions of a movie. In such cases, each version of the content contains many scenes in common with the other versions and some scenes that are unique to each version; the difference between each version is only in the set and order of scenes. This feature can also be used by people wishing to make, publish and exchange "community versions" of content—a feature that became very popular with the HD-DVD user-base. Each "community version" could be encoded by a small amount of metadata, rather than megabytes of bit stream data. Such an efficient way of recording each user's version makes the exchange of these versions feasible by email and web-site downloads.

Dynamic Versions

[0060] Dynamic versions of content can be presented by the Media File player based on metadata present in the Media File. Some different ways of creating dynamic versions are listed below. However, each method of creating a dynamic version still requires that the selected version be timed correctly so that a viewer can determine some basic time-related aspects of the version they choose, such as the total playback time for their version and the current playback position in that version.

[0061] To allow accurate timing information to be generated from a dynamic version many embodiments utilize the following clarifications for time-related variables, which are applicable to the structures of the MKV file format, as well as other file containers that utilize similar file segmentation methodologies.

Timed-Segment Being Described	Interpretation of ChapterTime* Values
none-same Segment	The time-codes define the beginning and end times of this Chapter relative to the start of the highest-priority video track they are associated with. The end time-code must be larger than the beginning.
external Segment defined by ChapterSegmentUID [ChapterSegmentChapterUID should not be defined]	The time-codes define the beginning and end times of this Chapter relative to the start of the highest-priority video track they are associated with in the identified Segment. If both time-codes are 0, then the defined Chapter is redirecting to the entire length of the external Segment. Otherwise, the end time-code must be larger than the beginning and they encode the portion of the Segment to be played.
external Edition defined by ChapterSegmentEditionUID [ChapterSegmentUID must also be defined]	The time-codes define the beginning and end times of this Chapter relative to the start of the highest-priority video track they are associated with in the identified Edition of the identified Segment. Both begin and end time-codes are ignored, and the defined Chapter is redirecting to the entire length of the external Edition.
external Chapter defined by ChapterSegmentChapterUID [ChapterSegmentUID and ChapterSegmentEditionUID must also be defined]	The time-codes define the beginning and end times of this Chapter relative to the start of the highest-priority video track they are associated with in the identified Chapter of the Edition of the external Segment. If both time-codes are 0, then the defined Chapter is redirecting to the entire length of the external Chapter. Otherwise, the end time-code must be larger than the beginning and they encode the portion of the Chapter to be played.

[0062] In another embodiment, as shown in FIG. **10**, playback could also be controlled by metadata track entries. For example, in the case where the "SIGNIFICANCE" tag has been used to mark the "importance" of each scene in a movie, the movie can be "intelligently" reduced in total duration by displaying only the most important scenes. Hence, a single encoding of a 3 hour movie could be viewed within 2.5 hours, or 1.5 hours depending on the amount of time the viewer had to watch the movie (for example during a short flight). Similarly, the metadata "RATING" tag could be used to store the age-appropriateness of each scene in the content, and only the scenes at or below a certain appropriateness would be shown (a feature that would allow many families to watch a new movie together). Practically any metadata tag could be used to view "dynamic versions" of one or more pieces of content. The filter for the dynamic version (or dynamic mash-up) could be based on whether a scene contains the viewer's favorite actor, or the GPS location of the shoot, etc.

Playback Redirection

[0063] Playback redirection allows the ability to create different "content collections" from a set of content. For example, if a user has home-videos titled "Vacation 2006," "Vacation 2007" and "Vacation 2008," and each of these have chapters titled "Day 1," "Day 2," etc.—then a single "Day 1" collection could be created that redirects playback to each "Day 1" chapter of each year's vacation video. This redirection file would be small, and if another collection, say "Day 2", was required, then it could be created, without having to do any re-encoding of the original titles. Another example of this is to simply utilize the redirection to achieve the DVD experience of trailers, movie, and extras. And yet another example would be to utilize redirection to link to adverts that are dynamically set by a service provider.

User-Entered Comments

[0064] User-entered comments in various embodiments could be stored in a metadata track unique to a user. Various properties of the user could also be entered as metadata associated with that user's metadata track header, such as the user's age, GPS location, native language, etc., as well as properties related to the comments, such as age-appropriateness, language, etc. These metadata values could then be used as the basis of filters at playback time to ensure that the viewer only viewed comments in languages they've enabled, and only if the comments' age-appropriate flags are less than or equal to the viewer's age, etc.

Advanced Media Management & Search

[0065] Media managers usually collate metadata about the content they manage and store that information in their own databases. However, this can lead to a non-portable experience for users that are increasingly moving their content between devices. The metadata structure provided lends itself well to the task of keeping the user's experience portable by allowing incorporation of experience data [such as "COMMENT" and "VIEWED_SEGMENTS"] in to the file. Furthermore, with its ease of implementation, even lower-end devices should be able to update appropriate metadata fields in order to maintain experiential data.

Launching Other Applications

[0066] One or more applications can be launched or started from the identification or decoding of metadata information. As such, the metadata information can specify a specific application or file type utilized by a specific or default application that is launched upon activation and decoding of the metadata information, such as the data type extension of a tag name. A playback device in one embodiment can be more compact and/or less complex as not requiring the application and other similar applications to be integrated into the playback device.

Integrated Advertising

[0067] Referring back to FIG. **9**, embedding advertising information into media files can also be provided, where the advertising is related to objects within the video track, or words in the audio or subtitle track. This placement of the adverts could become the payment method of the content, assuming that viewer's are required to view the "advertising" metadata track. As shown for example in FIG. **9**, three brands are outlined, highlighted and annotated with ROI metadata that provides direct links to those brands' Internet properties.

Creating Media Files Including Metadata Tracks

[0068] Referring now to FIG. **11**, a method of creating a media file including multimedia content and associated metadata information is provided. Utilizing a metadata source **111**, metadata objects are extracted **112** to generate a metadata table **113**. In one embodiment, the metadata source is provided via a user interface such as a video authoring application or system. In other embodiments, the metadata source is provided through a formatted file, such as a XML file. The metadata objects that are extracted from the source are objects that are instantiated multiple times and/or are referenced multiple times. The metadata table **113** stores a single copy of the metadata object and associates the copy to a universal identifier.

[0069] Global metadata objects **115** are also extracted **114** from the metadata source **111**. The global metadata objects describe general or global metadata entities, such as an entire file, title, chapter, etc. Utilizing the metadata table **113**, the metadata track header **117** is created or populated **116**. In one particular embodiment, the metadata track includes a list of universal identifiers. The universal identifiers correspond to the associated metadata objects that will be called for in each metadata track. The metadata track(s) are prepared for multiplexing with audio, video and subtitle bit streams **118**. The metadata track(s) **119** include the universal identifier along with the associated metadata object.

[0070] Each metadata track is coupled with a global universal identifier **110** and the content source **120** to create a target or complete media file **121**. In one embodiment, the global universal identifier is written first in the complete media file **122**. Metadata objects and content elements follow **122**. As previously noted, to maintain at least portability, the container's natively supported metadata tags are utilized. The media file **121** is stored to be accessible later based on a user request and/or sent immediately or shortly thereafter to satisfy a previous or pending user request.

Decoding Media Files Including Metadata Tracks

[0071] In FIG. **12**, a method of decoding the media file including multimedia content and associated metadata information is shown. The media file **130** includes content and metadata information. Metadata information in one embodiment includes global metadata objects **131**, a metadata track header **132**, metadata table(s) **133** and/or a metadata track **134**. From the metadata information, the appropriate universal identifiers and metadata object(s) are created to be played at the appropriate time **135**. In one embodiment, from the global metadata objects, global tags are read and the metadata objects and universal identifiers **140** are extracted based on the title to be played **135**.

[0072] The metadata track header **132** is read for each metadata track to be rendered **136**. In one embodiment, a list of universal identifiers **141** is extracted. Similarly, for each universal identifier requiring evaluation, the associated metadata object **142** is read **137** from the metadata table **133** to generate one or more metadata objects. Playback of the content is started **138** and additional metadata objects **143** are extracted. The metadata objects **142** and **143** are rendered and/or displayed **139**. In one embodiment, the displayed metadata objects are triggered or result from user interaction with the displayed content via a user playback interface. Also, based on the time and position of the main video track, associated metadata objects relative to the main video can also be displayed. The process continues until the user stops playback or otherwise terminates the playback and/or decoding of the media file.

[0073] Referring also again to FIG. **11**, the global universal identifier **110** may also be utilized in the decoding or playback process of FIG. **12**. The GUID for example is used to locate metadata objects that would be found in another media file. By not utilizing filename conventions that vary widely, the GUID removes this limitation and allows a constant or reliable indicator to locate the desired metadata object. In one embodiment, if the GUID referenced in the metadata track is in the media file, the playback device or engine would search through its local content library for the referenced to media file. If the referenced media file is not found, the playback device can request the file from a media server.

[0074] While the above description contains many specific embodiments of the invention, these should not be construed as limitations on the scope of the invention, but rather as an example of one embodiment thereof. Accordingly, the scope of the invention should be determined not by the embodiments illustrated, but by the appended claims and their equivalents.

What is claimed is:

1. A method of playing back metadata content stored in a media file, comprising:

providing a media file to a playback device, the media file having at least one metadata object and an association with content data, the metadata object referencing at least one facet of the content data;

decoding the content data by the playback device;

displaying content on a display screen from the decoded content data; and

decoding the at least one metadata object based on the displayed content by the playback device.

2. The method of claim **1** wherein the at least one metadata object has at least one metadata tag associated with at least one metadata value.

3. The method of claim **2** wherein the media file further comprises at least one metadata table, the at least one metadata table having at least one identifier and at least one metadata object.

4. The method of claim **3** wherein the at least one metadata table is positioned at the end of the media file.

5. The method of claim **3** wherein the at least one metadata table includes a reserved space allowing the at least one metadata table to grow without having to re-calculate and re-write all the elements of the file.

6. The method of claim **1** wherein the media file further comprises at least one metadata track incorporated throughout the content data and the track includes a metadata object, or a reference to a metadata object in a metadata table.

7. The method of claim **2** wherein the media file further comprises a global identifier that is static for the media file.

8. The method of claim **2** wherein the content data comprises video, audio, or subtitle frames and the at least one metadata tag references at least one facet of at least one video, audio or subtitle frame and further comprising rendering text using a single embedded compacted font to match text for subtitle frames or the at least one metadata tag.

9. The method of claim **2** wherein the at least one metadata tag includes an extension providing a further description of the tag.

10. The method of claim **6** wherein the at least one metadata track is associated with at least one content track and at least one metadata track header.

11. The method of claim **10** further comprising decoding the at least one metadata track based on a user playback instruction.

12. The method of claim **2** further comprising assigning the at least one metadata tag to a portion of the content data by associating at least one identifier to the portion of the content data and the at least one metadata tag.

13. The method of claim **1** wherein the at least one metadata object describes a region of interest relative to the displayed content.

14. The method of claim **2** wherein the at least one metadata value refers to remote content and further comprising converting the remote content to a localized content.

15. The method of claim **2** further comprising launching other applications based on the decoded metadata object.

16. The method of claim **2** further comprising extracting the at least one metadata object prior to displaying content and controlling the displaying of content based on the extracted at least one metadata object.

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

a media server configured to locate media files, each media file having an immutable global identifier; and

a client processor in network communication with the media server and configured to send requests for a media file to the media server, the media server configured to locate and transmit the requested media file based on the global identifier and the client processor further comprises a playback engine configured to decode a metadata track within the transmitted media file, the metadata track referring to content data in the transmitted media file.

18. The system of claim **17** wherein the playback engine decodes the metadata track by accessing a metadata table having an identifier and metadata objects.

19. The system of claim **17** wherein the playback engine displays metadata information based on the decoded metadata track.

20. A method of creating a media file having metadata information, the method comprising:

supplying a source of metadata information to an encoder;

supplying a source of content to an encoder;

generating a metadata object from the supplied metadata information by the encoder, the generated metadata object referencing at least one portion of the supplied content; and

integrating the metadata object with the supplied content to form a media file by the encoder.

21. The method of claim **20** wherein the content comprises video, audio and subtitle information.

22. The method of claim **20** further comprising generating a metadata table referencing the metadata object generated from the supplied metadata information.

23. The method of claim **20** wherein the metadata object comprises a presentation time and a duration.

24. The method of claim **20** further comprising extracting multiple referenced metadata objects within the metadata information source and storing a copy of the referenced metadata object in a metadata table.

25. The method of claim **24** further comprising extracting global metadata objects within the metadata information source and storing a copy of each global metadata object along with an associated identifier.

26. The method of claim **25** further comprising generating a metadata track and metadata track header including the associated identifiers.

27. The method of claim **20** further comprising integrating the metadata table into the media file.

28. The method of claim **27** further comprising integrating a global unique identifier into the media file.

* * * * *