



(51) International Patent Classification:

G06F 17/30 (2006.01)

(21) International Application Number:

PCT/US2017/013829

(22) International Filing Date:

17 January 2017 (17.01.2017)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/279,616	15 January 2016 (15.01.2016)	US
62/446,650	16 January 2017 (16.01.2017)	US

(72) Inventors; and

(71) Applicants : BAO, Sheng [US/US]; 915 Quarry Dr., Akron, OH 44307 (US). LIU, Yang [US/US]; 410 Wyant Rd, Akron, OH 44313 (US).

(74) Agent: ASSAR, Ali; Cooper Legal Group, LLC, 6505 Rockside Road, Suite 330, Independence, OH 44131 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

[Continued on nextpage]

(54) Title: SEARCHING, SUPPLEMENTING AND NAVIGATING MEDIA

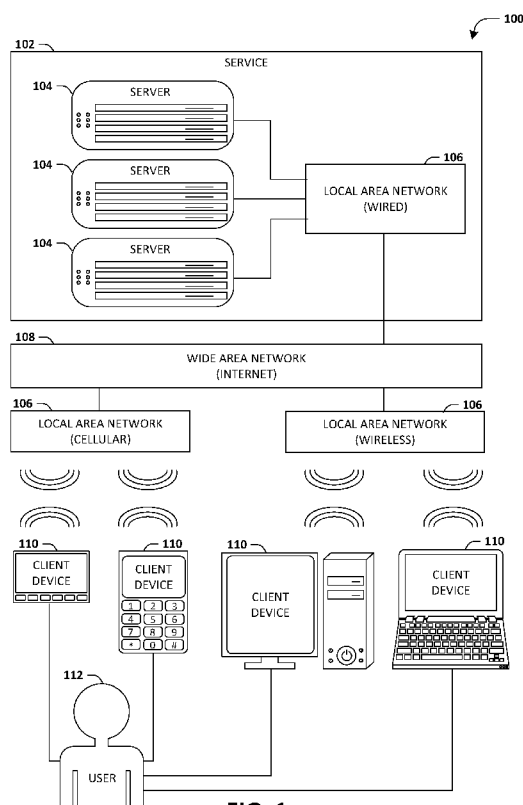


FIG. 1

(57) Abstract: One or more computing devices, systems, and/or methods for searching, supplementing and/or navigating media are provided. For example, a query for media may be used to identify results and provide the results based upon temporal properties of the results. In another example, media may be segmented into portions based upon time-associated text information of the media, and each portion of the media may be supplemented with content selected based upon a context of the portion. In another example, an area of a video may be selected based upon image analysis of the video, and the video may be supplemented with content at the area. In another example, a video may be supplemented with content, and properties of the content may be adjusted based upon image analysis of the video. In another example, media may be navigated through at different rates of advancement.



Published:

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

— before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))

SEARCHING, SUPPLEMENTING AND NAVIGATING MEDIA

RELATED APPLICATION

[0001] This application claims priority to U.S. Provisional Application No. 62/279,616, titled "TEXT-ENABLED MULTIMEDIA SYSTEM" and filed on January 15, 2016, and U.S. Provisional Application No. 62/446,650, titled "SEARCHING, SUPPLEMENTING AND/OR NAVIGATING MEDIA" and filed on January 16, 2017, both of which are incorporated herein by reference.

BACKGROUND

[0002] Many devices, such as mobile phones, tablets, laptops, mp4 players and/or desktop computers, provide media output by playing media files. A media file may be identified from a plurality of media files based upon a name of the media file. Media output may be navigated with an interface used to access different parts of the corresponding media file. Content may be played before or after the media output.

SUMMARY

[0003] In accordance with the present disclosure, one or more computing devices and/or methods for searching media (e.g., a movie) are provided. In an example, a query, comprising a first term, for the media may be received. A first result and a second result may be identified in time-associated information (e.g., a transcript) of the media based upon a determination that the first result comprises a first match of the first term and the second result comprises a second match of the first term. The first result and the second result may be provided (e.g., for presentation) based upon a first temporal property of the first match of the first term in the first result and a second temporal property of the second match of the first term in the second result.

[0004] In accordance with the present disclosure, one or more computing devices and/or methods for supplementing media (e.g., a movie) with content are provided. In an example, the media may be segmented into a first portion (e.g., a first scene of the movie) and a second portion (e.g., a second scene of the movie) based upon time-associated text information (e.g., a transcript) of the media. The time-associated text information of the media may be analyzed to determine a first context (e.g.,

entertainment) for the first portion and a second context (e.g., business) for the second portion. A first content (e.g., an advertisement for an entertainment-related product) may be selected from a plurality of contents for the first portion based upon the first context and a second content (e.g., an advertisement for a business-related product) may be selected from the plurality of contents for the second portion based upon the second context. The first portion of the media may be supplemented (e.g., overlaid, interrupted, etc.) with the first content and the second portion of the media may be supplemented (e.g., overlaid, interrupted, etc.) with the second content.

[0005] In accordance with the present disclosure, one or more computing devices and/or methods for supplementing a video with content are provided. In an example, a first content may be selected from a plurality of contents for the video. A first area (e.g., a top area, a bottom area, a side area, etc.) may be selected from a plurality of areas in the video based upon image analysis of the video. The video may be supplemented (e.g., overlaid) with the first content at the first area.

[0006] In accordance with the present disclosure, one or more computing devices and/or methods for supplementing a video with content are provided. In an example, a first content may be selected from a plurality of contents for the video. The video may be supplemented with the first content. One or more properties (e.g., color, transparency, size, duration, etc.) of the first content may be adjusted based upon image analysis of the video.

[0007] In accordance with the present disclosure, one or more computing devices and/or methods for generating a representation of a performance are provided. In an example, a request to implement a (e.g., karaoke) performance with a first user and a second user may be received. A determination may be made the first user is associated with a first type of participation (e.g., singing) in the performance. A determination may be made that the second user is associated with a second type of participation (e.g., dancing) in the performance. A first content (e.g., a first version of a song) may be selected from a plurality of contents for the first user based upon the first type of participation and a second content (e.g., a second version of the song) may be selected from the plurality of contents for the second user based upon the second type of participation. The first content may be provided to the first user and the second content may be provided to the second user. A first signal (e.g., comprising audio of the first user singing) may be received from the first user in association with the performance

and a second signal (e.g., comprising video of the second user dancing) may be received from the second user in association with the performance. A representation of the performance may be generated based upon a combination of the first signal, the second signal, the first content and the second content.

[0008] In accordance with the present disclosure, one or more computing devices and/or methods for navigating through media (e.g., a video) are provided. In an example, a request to move (e.g., drag) a control along a first axis from a first portion of the first axis to a second portion of the first axis may be received. Responsive to determining that the control is being moved along the first axis within the first portion (e.g., between the second minute and fourth minute of the video), the media may be navigated through at a first rate of advancement (e.g. a first temporal resolution) based upon a first feature of the first portion. Responsive to determining that the control is being moved along the first axis within the second portion (e.g., between the eighth minute and tenth minute of the video), the media may be navigated through at a second rate of advancement (e.g. a second temporal resolution) based upon a second feature of the second portion. The first rate of advancement may be different than the second rate of advancement.

DESCRIPTION OF THE DRAWINGS

[0009] While the techniques presented herein may be embodied in alternative forms, the particular embodiments illustrated in the drawings are only a few examples that are supplemental of the description provided herein. These embodiments are not to be interpreted in a limiting manner, such as limiting the claims appended hereto.

[001 0] Fig. 1 is an illustration of a scenario involving various examples of networks that may connect servers and clients.

[001 1] Fig. 2 is an illustration of a scenario involving an example configuration of a server that may utilize and/or implement at least a portion of the techniques presented herein.

[001 2] Fig. 3 is an illustration of a scenario involving an example configuration of a client that may utilize and/or implement at least a portion of the techniques presented herein.

[0013] Fig. 4A is a flow chart illustrating an example method for searching media.

[0014] Fig. 4B is a component block diagram illustrating an example system for searching media.

[0015] Fig. 5A is a flow chart illustrating an example method for supplementing media with content.

[0016] Fig. 5B is a component block diagram illustrating an example system for supplementing media with content.

[0017] Fig. 5C is a flow chart illustrating an example method for supplementing a video with content.

[0018] Fig. 5D is a flow chart illustrating an example method for supplementing a video with content.

[0019] Fig. 6A is a flow chart illustrating an example method for generating a representation of a performance.

[0020] Fig. 6B is a component block diagram illustrating an example system for generating a representation of a performance.

[0021] Fig. 7A is a flow chart illustrating an example method for navigating through media.

[0022] Fig. 7B is a component block diagram illustrating an example system for navigating through media.

[0023] Fig. 8 is an illustration of a scenario featuring an example non-transitory machine readable medium in accordance with one or more of the provisions set forth herein.

[0024] Fig. 9 is an illustration of a disclosed embodiment.

[0025] Fig. 10 is an illustration of a disclosed embodiment.

[0026] Fig. 11 is an illustration of a disclosed embodiment.

[0027] Fig. 12 is an illustration of a disclosed embodiment.

[0028] Fig. 13 is an illustration of a disclosed embodiment.

[0029] Fig. 14 is an illustration of a disclosed embodiment.

[0030] Fig. 15 is an illustration of a disclosed embodiment.

- [0031] Fig. 16 is an illustration of a disclosed embodiment.
- [0032] Fig. 17 is an illustration of a disclosed embodiment.
- [0033] Fig. 18 is an illustration of a disclosed embodiment.
- [0034] Fig. 19 is an illustration of a disclosed embodiment.
- [0035] Fig. 20 is an illustration of a disclosed embodiment.
- [0036] Fig. 21 is an illustration of a disclosed embodiment.
- [0037] Fig. 22 is an illustration of a disclosed embodiment.
- [0038] Fig. 23 is an illustration of a disclosed embodiment.
- [0039] Fig. 24 is an illustration of a disclosed embodiment.
- [0040] Fig. 25 is an illustration of a disclosed embodiment.
- [0041] Fig. 26 is an illustration of a disclosed embodiment.
- [0042] Fig. 27 is an illustration of a disclosed embodiment.
- [0043] Fig. 28 is an illustration of a disclosed embodiment.
- [0044] Fig. 29 is an illustration of a disclosed embodiment.
- [0045] Fig. 30 is an illustration of a disclosed embodiment.
- [0046] Fig. 31 is an illustration of a disclosed embodiment.
- [0047] Fig. 32 is an illustration of a disclosed embodiment.
- [0048] Fig. 33 is an illustration of a disclosed embodiment.
- [0049] Fig. 34 is an illustration of a disclosed embodiment.
- [0050] Fig. 35 is an illustration of a disclosed embodiment.
- [0051] Fig. 36 is an illustration of a disclosed embodiment.
- [0052] Fig. 37 is an illustration of a disclosed embodiment.
- [0053] Fig. 38 is an illustration of a disclosed embodiment.
- [0054] Fig. 39 is an illustration of a disclosed embodiment.
- [0055] Fig. 40 is an illustration of a disclosed embodiment.
- [0056] Fig. 41 is an illustration of a disclosed embodiment.

- [0057] Fig. 42 is an illustration of a disclosed embodiment.
- [0058] Fig. 43 is an illustration of a disclosed embodiment.
- [0059] Fig. 44 is an illustration of a disclosed embodiment.
- [0060] Fig. 45 is an illustration of a disclosed embodiment.
- [0061] Fig. 46 is an illustration of a disclosed embodiment.
- [0062] Fig. 47 is an illustration of a disclosed embodiment.
- [0063] Fig. 48 is an illustration of a disclosed embodiment.
- [0064] Fig. 49 is an illustration of a disclosed embodiment.
- [0065] Fig. 50 is an illustration of a disclosed embodiment.
- [0066] Fig. 51 is an illustration of a disclosed embodiment.
- [0067] Fig. 52 is an illustration of a disclosed embodiment.
- [0068] Fig. 53 is an illustration of a disclosed embodiment.
- [0069] Fig. 54 is an illustration of a disclosed embodiment.
- [0070] Fig. 55 is an illustration of a disclosed embodiment.
- [0071] Fig. 56 is an illustration of a disclosed embodiment.
- [0072] Fig. 57 is an illustration of a disclosed embodiment.
- [0073] Fig. 58 is an illustration of a disclosed embodiment.
- [0074] Fig. 59 is an illustration of a disclosed embodiment.
- [0075] Fig. 62 is an illustration of a disclosed embodiment.
- [0076] Fig. 63 is an illustration of a disclosed embodiment.
- [0077] Fig. 64 is an illustration of a disclosed embodiment.
- [0078] Fig. 65 is an illustration of a disclosed embodiment.
- [0079] Fig. 66 is an illustration of a disclosed embodiment.
- [0080] Fig. 67 is an illustration of a disclosed embodiment.
- [0081] Fig. 68 is an illustration of a disclosed embodiment.
- [0082] Fig. 69 is an illustration of a disclosed embodiment.

[0083] Fig. 70 is an illustration of a disclosed embodiment.

[0084] Fig. 71 is an illustration of a disclosed embodiment.

[0085] Fig. 72 is an illustration of a disclosed embodiment.

[0086] Fig. 73 is an illustration of a disclosed embodiment.

DETAILED DESCRIPTION

[0087] Subject matter will now be described more fully hereinafter with reference to the accompanying drawings, which form a part hereof, and which show, by way of illustration, specific example embodiments. This description is not intended as an extensive or detailed discussion of known concepts. Details that are known generally to those of ordinary skill in the relevant art may have been omitted, or may be handled in summary fashion.

[0088] The following subject matter may be embodied in one or more different forms, such as methods, devices, components, and/or systems. Accordingly, this subject matter is not intended to be construed as limited to any example embodiments set forth herein. Rather, example embodiments are provided merely to be illustrative. Such embodiments may, for example, take the form of hardware, software, firmware or any combination thereof.

[0089] Computing Scenario

[0090] The following provides a discussion of some types of computing scenarios in which the disclosed subject matter may be utilized and/or implemented.

[0091] Networking

[0092] Fig. 1 is an interaction diagram of a scenario 100 illustrating a service 102 provided by a set of servers 104 to a set of client devices 110 via various types of networks. The servers 104 and/or client devices 110 may be capable of transmitting, receiving, processing, and/or storing many types of signals, such as in memory as physical memory states.

[0093] The servers 104 of the service 102 may be internally connected via a local area network 106 (LAN), such as a wired network where network adapters on the respective servers 104 are interconnected via cables, such as coaxial and/or fiber optic cabling, for example, and may be connected in various topologies, such as buses, token rings, meshes, and/or trees, for example. The servers 104 may utilize one or more physical networking protocols, such as Ethernet and/or Fiber Channel, and/or logical networking protocols, such as variants of an Internet Protocol (IP), a Transmission Control Protocol (TCP), and/or a User Datagram Protocol (UDP). The servers 104 may be interconnected directly, or through one or more other networking devices, such as routers, switches, and/or repeaters.

[0094] The local area network 106 may be organized according to one or more network architectures, such as server/client, peer-to-peer, and/or mesh architectures, and/or one or more roles, such as administrative servers, authentication servers, security monitor servers, data stores for objects such as files and databases, business logic servers, time synchronization servers, and/or front-end servers providing a user-facing interface for the service 102.

[0095] The local area network 106 may include, e.g., analog telephone lines, such as a twisted wire pair, a coaxial cable, full or fractional digital lines including T1, T2, T3, or T4 type lines, Integrated Services Digital Networks (ISDNs), Digital Subscriber Lines (DSLs), wireless links including satellite links, or other communication links or channels, such as may be known to those skilled in the art.

[0096] Likewise, the local area network 106 may comprise one or more sub-networks, such as may employ differing architectures, may be compliant or compatible with differing protocols and/or may interoperate within the local area network 106. Additionally, one or more local area networks 106 may be interconnected; e.g., a router may provide a link between otherwise separate and independent local area networks 106.

[0097] The local area network 106 of the service 102 may be connected to a wide area network 108 (WAN) that allows the service 102 to exchange data with other services 102 and/or client devices 110. The wide area network 108 may encompass various combinations of devices with varying levels of distribution and exposure,

such as a public wide-area network, such as the Internet, and/or a private network, such as a virtual private network (VPN) of a distributed enterprise.

[0098] The service 102 may be accessed via the wide area network 108 by a user 112 of one or more client devices 110, such as a portable media player, such as an electronic text reader, an audio device, or a portable gaming, exercise, or navigation device; a portable communication device, such as a camera, a phone, a wearable or a text chatting device; a workstation; and/or a laptop form factor computer.

[0099] One or more client devices 110 may communicate with the service 102 via various connections to the wide area network 108. For example, one or more client devices 110 may comprise a cellular communicator and may communicate with the service 102 by connecting to the wide area network 108 via a wireless local area network 106 provided by a cellular provider. In another example, one or more client devices 110 may communicate with the service 102 by connecting to the wide area network 108 via a wireless local area network 106 provided by a location such as the user's home or workplace, such as a WiFi (Institute of Electrical and Electronics Engineers (IEEE) Standard 802.11) network or a Bluetooth (IEEE Standard 802.15.1) personal area network. Thus, the servers 104 and the client devices 110 may communicate over various types of networks. Other types of networks that may be accessed by the servers 104 and/or client devices 110 include mass storage, such as network attached storage (NAS), a storage area network (SAN), and/or other forms of computer or machine readable media.

[001 00] Server Configuration

[001 01] Fig. 2 presents a schematic architecture diagram 200 of a server 104 that may utilize at least a portion of the techniques provided herein. Such a server 104 may vary widely in configuration or capabilities, alone or in conjunction with other servers, in order to provide a service such as the service 102.

[001 02] The server 104 may comprise one or more processors 210 that process instructions. The one or more processors 210 may optionally include a plurality of cores; one or more coprocessors, such as a mathematics coprocessor or an integrated graphical processing unit (GPU); and/or one or more layers of local cache memory. The server 104 may comprise memory 202 storing various forms of applications, such

as an operating system 204; one or more server applications 206, such as a hypertext transport protocol (HTTP) server, a file transfer protocol (FTP) server, or a simple mail transport protocol (SMTP) server; and/or various forms of data, such as a database 208 or a file system. The server 104 may comprise one or more peripheral components, such as a wired and/or wireless network adapter 214 connectible to a local area network and/or wide area network; one or more storage components 216, such as a hard disk drive, a solid-state storage device (SSD), a flash memory device, and/or a magnetic and/or optical disk reader.

[001 03] The server 104 may comprise a mainboard featuring one or more communication buses 212 that interconnect the processor 210, the memory 202, and various peripherals, using one or more bus technologies, such as a variant of a serial or parallel AT Attachment (ATA) bus protocol; a Uniform Serial Bus (USB) protocol; and/or Small Computer System Interface (SCI) bus protocol.

[001 04] A communication bus 212 may interconnect the server 104 with at least one other server (e.g., in a multibus scenario). Other components that may be included with the server 104 (though not shown in the schematic diagram 200 of Fig. 2) include a display; a display adapter, such as a graphical processing unit (GPU); input peripherals, such as a keyboard and/or mouse; and a flash memory device that may store a basic input/output system (BIOS) routine that facilitates booting the server 104 to a state of readiness.

[001 05] The server 104 may operate in various physical enclosures, such as a desktop or tower, and/or may be integrated with a display as an "all-in-one" device. The server 104 may be mounted horizontally and/or in a cabinet or rack, and/or may simply comprise an interconnected set of components. The server 104 may comprise a dedicated and/or shared power supply 218 that supplies and/or regulates power for the other components. The server 104 may provide power to and/or receive power from another server and/or other devices. The server 104 may comprise a shared and/or dedicated climate control unit 220 that regulates climate properties, such as temperature, humidity, and/or airflow. Servers 104 may be configured and/or adapted to utilize at least a portion of the techniques presented herein.

[001 06] Client Device Configuration

[001 07] Fig. 3 presents a schematic architecture diagram 300 of a client device 110 whereupon at least a portion of the techniques presented herein may be implemented. Such a client device 110 may vary widely in configuration or capabilities, in order to provide one or more functionality to a user such as the user 112. The client device 110 may serve the user in one or more roles, such as a workstation, kiosk, media player, gaming device, and/or appliance.

[001 08] The client device 110 may comprise one or more processors 310 that process instructions. The one or more processors 310 may optionally include a plurality of cores; one or more coprocessors, such as a mathematics coprocessor or an integrated graphical processing unit (GPU); and/or one or more layers of local cache memory. The client device 110 may comprise memory 301 storing various forms of applications, such as an operating system 303; one or more user applications 302, such as document applications, media applications, file and/or data access applications, communication applications such as web browsers and/or email clients, utilities, and/or games; and/or drivers for various peripherals.

[001 09] The client device 110 may comprise one or more peripheral components, such as a wired and/or wireless network adapter 306 connectible to a local area network and/or wide area network; one or more output components, such as a display 308 coupled with a display adapter (optionally including a graphical processing unit (GPU)), a sound adapter coupled with a speaker, and/or a printer; input devices for receiving input from the user, such as a keyboard 311, a mouse, a microphone, a camera, and/or a touch-sensitive component of the display 308; and/or environmental sensors, such as a global positioning system (GPS) receiver 319 that detects the location, velocity, and/or acceleration of the client device 110, a compass, accelerometer, and/or gyroscope that detects a physical orientation of the client device 110.

[001 10] Other components that may optionally be included with the client device 110 (though not shown in the schematic architecture diagram 300 of Fig. 3) include one or more storage components, such as a hard disk drive, a solid-state storage device (SSD), a flash memory device, and/or a magnetic and/or optical disk reader; and/or a flash memory device that may store a basic input/output system (BIOS) routine that facilitates booting the client device 110 to a state of readiness; and

a climate control unit that regulates climate properties, such as temperature, humidity, and airflow.

[001 11] The client device 110 may be provided in one or more form factors, such as a desktop or tower workstation; an "all-in-one" device integrated with a display 308; a laptop, tablet, convertible tablet, or palmtop device; a wearable device mountable in a headset, eyeglass, earpiece, and/or wristwatch, and/or integrated with an article of clothing; and/or a component of a piece of furniture, such as a tabletop, and/or of another device, such as a vehicle or residence.

[001 12] The client device 110 may comprise a dedicated and/or shared power supply 318 that supplies and/or regulates power for other components, and/or a battery 304 that stores power for use while the client device 110 is not connected to a power source via the power supply 318. The client device 110 may provide power to and/or receive power from other client devices.

[001 13] The client device 110 may include one or more servers that may locally serve the client device 110 and/or other client devices of the user 112 and/or other individuals. For example, a locally installed webserver may provide web content in response to locally submitted web requests. Many such client devices 110 may be configured and/or adapted to utilize at least a portion of the techniques presented herein.

[001 14] The client device 110 may comprise a mainboard featuring one or more communication buses 312 that interconnect the processor 310, the memory 301, and various peripherals, using one or more bus technologies, such as a variant of a serial or parallel AT Attachment (ATA) bus protocol; the Uniform Serial Bus (USB) protocol; and/or the Small Computer System Interface (SCI) bus protocol.

[001 15] In some scenarios, as a user 112 interacts with a software application on a client device 110, such as an instant messenger and/or electronic mail application), descriptive content in the form of signals or stored physical states within memory, such as an email address, instant messenger identifier, phone number, postal address, message content, date, and/or time) may be identified. Descriptive content may be stored, typically along with contextual content. For example, the source of a phone number, such as a communication received from another user via an instant messenger application) may be stored as contextual content associated with the phone

number. Contextual content, therefore, may identify circumstances surrounding receipt of a phone number, such as the date or time that the phone number was received), and may be associated with descriptive content. Contextual content, may, for example, be used to subsequently search for associated descriptive content. For example, a search for phone numbers received from specific individuals, received via an instant messenger application or at a given date or time, may be initiated.

[001 16] Presented Techniques

[001 17] One or more computing devices and/or techniques for searching media, supplementing media with content, supplementing a video with content, generating a representation of a performance and/or navigating through media are provided. For example, a server, such as that of an online media content publisher, may serve to host media received from a user of the server such that the hosted media may be accessed by a plurality of users. The media may be difficult to find from amongst many (e.g., hundreds, thousands, millions, etc.) of other media. If the media has a length exceeding a threshold (e.g., 1 hour), identifying a portion of the media (e.g., a scene) that is of interest to a viewer may be difficult. For example, the media may have to be identified based upon a file name, a title and/or a description of the media. It may be appreciated that a scene in media may not be described in the title, the file name, the description, etc. Viewing and/or listening to the media in its entirety may consume a significant amount of time and resources of the server and the user. Fast forwarding and/or rewinding through the media using some techniques may also consume a significant amount of time and resources while still possibly resulting in the viewer overlooking the portion of the media of that is of interest. Supplementing the media using some techniques may interrupt, distract from and/or otherwise interfere with a desired experience of the viewer. Thus, in accordance with one or more of the techniques presented herein, media may be searched, supplemented and/or navigated through in a manner that is efficient, convenient, effective and/or timely.

[001 18] An embodiment of searching media (e.g., a video) is illustrated by an example method 400 of Fig. 4A. A user, such as user Jill, (e.g., and/or a device associated with the user) may access and/or interact with a website, an application, etc. that provides a platform for searching the media using a server (e.g., of the website, the

application, etc.). The server may host uploaded media, and the website may provide access to view and/or hear the uploaded media to an audience. Accordingly, at 404, a query, comprising at least a first term, for the media may be received (e.g., by the server and/or from the user).

[001 19] The media may comprise video, audio, an image and/or a document. For example, the media may comprise a video file, a movie, a television show, an audiobook, a podcast, radio, a soundtrack, a song recording, a voice memo, a voicemail, virtual reality content, augmented reality content, videochat streaming, financial data, stock indexes, security prices, financial transaction data, financial statements, a balance sheet, an income statement, a statement of changes in equity, a cash flow statement, physiological data, medical data, medical sensor data, vital sign data, electrophysiological data, medical images, medical lab analysis data, industrial data, security data and/or military data.

[001 20] At 406, a first result may be identified in time-associated information (e.g., a transcript) of the media based upon a determination that the first result comprises a first match of the first term, and/or a second result may be identified in the time-associated information of the media based upon a determination that the second result comprises a second match of the first term. The time-associated information may comprise text information of the media, such as a transcript, a comment, a user annotation or a review associated with the media. The time-associated information may alternatively and/or additionally comprise non-text information of the media, such as audio (e.g., a soundtrack), an image (e.g., a frame), etc. One or more portions, terms, etc. of the time-associated information may be associated with one or more timestamps.

[001 21] It may be appreciated that the matches may comprise exact matches and/or non-exact matches between a term of the query and the time-associated information of the media.

[001 22] The identifying of the first result (e.g., and/or one or more other results) may be performed using a brute-force search, a satisfiability check, a temporal sliding window, clustering, unsupervised machine learning, supervised machine learning, reinforcement learning, deep learning and/or pre-indexing.

[001 23] In an example, the media may be transcribed (e.g., before 406, before 404 and/or after 404) to generate a transcript of the media, which may be at least some of the time-associated information. The transcript may include text representative of recognized speech (e.g., spoken in audio of the media), music (e.g., played in the audio of the media) or other sounds (e.g., background noise, such as of cars, movement, applause, etc.). The transcript may include text representative of recognized objects (e.g., a computer), persons (e.g., the President) or other images (e.g., of a location, scenery, weather, etc.).

[001 24] The transcript may be in a first language (e.g., English), and may be translated to generate a second transcript of the media in a second language (e.g., German), which may be at least some of the time-associated information.

[001 25] One or more landmarks (e.g., entities, trademarks, names, brands, key phrases, indications of emphasis, etc.), tags, summaries and/or cross-references may be identified and extracted (e.g., before 406, before 404 and/or after 404) from the time-associated information (e.g., and stored in an index). The identifying of the first result (e.g., and/or one or more other results) may be performed by searching the index (e.g., prior to searching a larger portion (e.g., entirety) of the time-associated information).

[001 26] At 408, the first result may be provided (e.g., for presentation) based upon a first temporal property of the first match of the first term in the first result and the second result may be provided (e.g., for presentation) based upon a second temporal property of the second match of the first term in the second result.

[001 27] For example, the first result and the second result may be provided (e.g., for presentation) responsive to determining that the first temporal property and the second temporal property (e.g., individually, in combination, etc.) exceed a threshold temporal property. Alternatively and/or additionally, the first result may be provided (e.g., for presentation) in association with a higher rank than the second result based upon a comparison of the first temporal property and the second temporal property (e.g., the first result may be ranked higher based upon the first temporal property being greater than, less than, before and/or after the second temporal property).

[001 28] In an example, the query may comprise the first term and a second term. In the example, the first result may be identified based upon the determination that the first result comprises the first match of the first term and a determination that the first result comprises a first match of the second term, and the second result may be identified based upon the determination that the second result comprises the second match of the first term and a determination that the second result comprises a second match of the second term. In the example, the first result may be provided (e.g., for presentation) based upon the first temporal property of the first match of the first term in the first result and a third temporal property of the first match of the second term in the first result, and the second result may be provided (e.g., for presentation) based upon the second temporal property of the second match of the first term in the second result and a fourth temporal property of the second match of the second term in the second result.

[001 29] A first temporal distance of the first result may be determined based upon the first temporal property and the third temporal property. For example, the first temporal distance may correspond to a difference between a first timestamp of the first match of the first term in the first result and a second timestamp of the first match of the second term in the first result. A second temporal distance of the second result may be determined based upon the second temporal property and the fourth temporal property. For example, the second temporal distance may correspond to a difference between a third timestamp of the second match of the first term in the second result and a fourth timestamp of the second match of the second term in the second result.

[001 30] In an example, the first result and the second result may be provided (e.g., for presentation) responsive to determining that the first temporal distance (e.g., 5 seconds) and the second temporal distance (e.g., 10 seconds) are less than a threshold temporal distance (e.g., 12 seconds). For example, results with an excessively large temporal distance between terms may be assumed as being unlikely to be the result sought by the user, and thus excluded from presentation to the user. The threshold temporal distance may be determined based upon the query. For example, the query may comprise a value (e.g., 12 seconds) specifying the threshold temporal distance, or the value may be estimated based upon one or more non-numerical aspects of the query.

[001 31] In an example, the first result may be ranked (e.g., and correspondingly presented) higher than the second result responsive to determining that the first temporal distance of the first result is less than the second temporal distance of the second result. For example, a result with little temporal distance between terms may be determined to be more likely to be the result sought by the user than a result with a large temporal distance between terms.

[001 32] The providing of the first result and the second result may comprise providing the results for presentation (e.g., by providing instructions to a device of the user to display the results), or providing the results to another website, application, etc. for further processing.

[001 33] In an example, the media may comprise a soundtrack (e.g., a song), and the time-associated information of the media may comprise lyrics of the soundtrack. The first result and/or the second result (e.g., in response to being selected) may be used to provide a karaoke presentation of the soundtrack (e.g., from a timestamp of the first result and/or the second result).

[001 34] In an example, a list of index keys corresponding to the query may be generated. The list of index keys may comprise index keys corresponding to each result and associated with a corresponding portion (e.g., timestamp, segment, etc.) of the media. For example, the list of index keys may comprise a first index key corresponding to the first result and associated with a first portion of the media, and a second index key corresponding to the second result and associated with a second portion of the media. When the first index key is selected, access may be provided to the first portion of the media. For example, one or more frames of the media corresponding to a first timestamp of the first portion may be displayed, the media may be played from a first timestamp of the first portion and/or a portion of time-associated information (e.g., a transcript) corresponding to the first timestamp may be displayed. When the second index key is selected, access may be provided to the second portion of the media. For example, one or more frames of the media corresponding to a second timestamp of the second portion may be displayed, the media may be played from the second timestamp of the second portion and/or a portion of time-associated information (e.g., a transcript) corresponding to the second timestamp may be displayed. In this manner, a user may identify a portion of the media to view, edit, comment upon, etc. It should be appreciated that a user may

identify a portion of first media from a plurality of media to view, edit, comment upon, etc.

[001 35] It may be appreciated that method 400 may be implemented in contexts other than a search performed by a user. For example, a content allocator may submit the query to identify portions of the media with context matching a context of one or more content, and to supplement the portions of the media with the one or more content.

[00136] Fig. 4B illustrates an example of a system 450 for searching media. Query component 410 may receive an indication of a media to search (e.g., the movie "Classic Ancient Warrior Movie"), and a query with one or more terms to find in the media (e.g., the terms "gold gate"). Search component 412 may search through time-associated information of the media (e.g., a transcript of the movie) to identify results where the terms of the query are found. For example, the search component 412 may find a first result comprising a first portion of the transcript of the movie comprising text with the first term of the query "gold" and the second term of the query "gate," and a second portion of the transcript of the movie comprising text with the first term of the query "gold" and the second term of the query "gate."

[001 37] Temporal property determination component 414 may determine one or more temporal properties of each of the results. For example, a first temporal distance (e.g., 10 seconds) between a match (e.g., same, similar, relevant, etc.) of the first term in the first result and a match (e.g., same, similar, relevant, etc.) of the second term in the first result may be measured (e.g., based upon a timestamp associated with the match of the first term in the first result, a timestamp associated with the match of the second term in the first result, and/or based upon an estimate based upon a number of words, characters, syllables, etc. between the matches). A second temporal distance (e.g., 5 seconds) between a match (e.g., same, similar, relevant, etc.) of the first term in the second result and a match (e.g., same, similar, relevant, etc.) of the second term in the second result may be measured.

[00138] Ranking and presentation component 416 may rank the first result and the second result based upon temporal properties of each of the results. For example, the second result may be ranked higher than the first result responsive to determining that the second temporal distance (e.g., 5 seconds) of the second result is less than the

first temporal distance (e.g., 10 seconds) of the first result. It should be appreciated that the ranking based on temporal distance may be given higher priority than other ranking considerations (e.g., such as how many words are between each of the matches). The ranking and presentation component 416 may present the ranked results with an excerpt of at least a portion of the time-associated information (e.g., transcript) corresponding to each result, a link to access the portion of the time-associated information corresponding to each result and/or a link to access (e.g., view, play, hear, etc.) the portion of the media corresponding to each result.

[00139] An embodiment of supplementing media (e.g., a video) with content is illustrated by an example method 500 of Fig. 5A. A website, an application, etc. may provide a platform for supplementing the media with content (e.g., advertisements) using a server (e.g., of the website, the application, etc.). The media may comprise video, audio, an image, a document, virtual reality content and/or augmented reality content. The server may host uploaded media, and the website may provide access to view and/or hear the uploaded media to an audience. Accordingly, at 502, the media may be segmented into a first portion and a second portion based upon time-associated text information of the media. For example, the segmenting may be performed based upon identification of scene changes, topic changes, location changes, etc. in the time-associated text information of the media, and/or may be performed based upon timestamps.

[00140] The first portion and the second portion may be of different, similar or equal length. It may be appreciated that the media may be segmented into any number of portions, such as three, four, five, or five hundred, and that each of the portions may be of different, similar or equal length. The number of portions for the media to be segmented into may be determined based upon a (e.g., default or user defined) desired length of each portion, or based upon a number of portions that is determined to be appropriate for the media based upon the time-associated text information.

[00141] At 504, the time-associated text information of the media may be analyzed to determine a first context for the first portion and a second context for the second portion. For example, a first portion of the time-associated text information that corresponds to the first portion of the media may be analyzed to determine the first context, which may be a first topic, a first theme, a first location, a first object, a

first person, etc., and/or a second portion of the time-associated text information that corresponds to the second portion of the media may be analyzed to determine the second context, which may be a second topic, a second theme, a second location, a second object, a second person, etc.

[001 42] In an example, the media may be transcribed (e.g., before 504, before 502 and/or after 502) to generate a transcript of the media, which may be at least some of the time-associated text information.

[001 43] At 506, a first content may be selected from a plurality of contents for the first portion based upon the first context, and a second content may be selected from the plurality of contents for the second portion based upon the second context. For example, a first advertisement for a first product may be determined to be relevant to the first portion of the media, and may thus be selected for the first portion of the media, while a second advertisement for a second product may be determined to be relevant to the second portion of the media, and may thus be selected for the second portion of the media.

[001 44] In an example, the first content may be selected for the first portion responsive to determining that the first portion is content-compatible (e.g., based upon the first context) and/or the second content may be selected for the second portion responsive to determining that the second portion is content-compatible (e.g., based upon the second context). The time-associated text information of the media may be analyzed to determine a third context for a third portion of the media. In the example, the third portion may (e.g., selectively) not be supplemented with content responsive to determining that the third portion is content-incompatible (e.g., based upon the third context). Content-compatibility versus incompatibility may correspond to a likelihood of content being received without (versus with) negative reaction and/or without exceeding a level of distraction. For example, the first portion may be determined to be content-compatible due to the first context being indicative of a racing scene, and the second portion may be determined to be content-compatible due to the second context being indicative of a funny scene, but the third portion may be determined to be content-incompatible due to the third context being indicative of a funeral scene.

[001 45] At 508, the first portion of the media may be supplemented with the first content, and the second portion of the media may be supplemented with the second content. For example, the first content may be overlaid upon and/or displayed concurrently with the first portion of the media while the second content may be overlaid upon and/or displayed concurrently with the second portion of the media. In another example, the first content may be played before, after or in between frames of the first portion, while the second content may be played before, after or in between frames of the second portion.

[001 46] In an example, a first timestamp may be selected from a plurality of timestamps in the first portion based upon a first match between the first content and a portion of the time-associated text information associated with the first timestamp. A second timestamp may be selected from a plurality of timestamps in the second portion based upon a second match between the second content and a portion of the time-associated text information associated with the second timestamp. For example, a timestamp corresponding to a part of the media determined to be best situated, least disruptive and/or most relevant for each content may be found in each portion. In the example, the first portion of the media may be supplemented with the first content at the first timestamp, and the second portion of the media may be supplemented with the second content at the second timestamp.

[001 47] In an example, method 500 may be implemented in the context of a movie theater, where the media may be a movie. For example, the first portion of the movie may be supplemented with the first content, the second portion of the movie may be supplemented with the second content, and the portions of the movie may be provided for presentation in the movie theater. It may be appreciated that costs of accessing the movie theater may be reduced or eliminated by supplementing the movie with various, relevant (e.g., and sponsored) content.

[001 48] In an example, method 500 may be implemented in the context of television programming, where the media may be (e.g., live) television. For example, the first portion of the television may be supplemented with the first content, the second portion of the television may be supplemented with the second content, and the portions of the television may be provided for presentation (e.g., broadcast) to an audience. It may be appreciated that costs of accessing the television may be reduced or eliminated by (e.g., dynamically) supplementing the television with various,

relevant (e.g., and sponsored) content, and that the content supplemented may be more relevant and/or less invasive/interruptive (e.g., by being displayed concurrently with or overlaid on television programming) than existing television programming.

[001 49] In an example, method 500 may be implemented in the context of educational material, where the media may be an educational lecture. For example, the first portion of the educational lecture may be supplemented with the first content, the second portion of the educational lecture may be supplemented with the second content, and the portions of the educational lecture may be provided for presentation to a student. It may be appreciated that costs of getting an education may be reduced or eliminated by (e.g., dynamically) supplementing the educational lecture with various, relevant (e.g., and sponsored) content. Presentation of the content may be implemented in an education-friendly manner. For example, the content may be presented at one or more times determined to be less likely to distract the student from learning and/or content selected by the student may not be accessed and/or presented until after one or more portions of the educational lecture are completed.

[001 50] Fig. 5B illustrates an example of a system 525 for supplementing media with content. Segmenter 510 may segment media (e.g., a video) into one or more portions, such as a first portion and a second portion. Context analyzer 512 may (e.g., in parallel) analyze time-associated text information of the media, the first portion and/or the second portion to determine a first context 514 associated with the first portion and a second context 516 associated with the second portion. For example, the first context 514 may indicate that the first portion of the media pertains to sports, while the second context 516 may indicate that the second portion of the media pertains to cars.

[00151] Content selector 518 may (e.g., in parallel) select a first content from a plurality of contents based upon the first context 514 and/or a second content from a plurality of contents based upon the second context 516. For example, a first sponsored message pertaining to sports may be selected based upon the first context 514, while a second sponsored message pertaining to cars may be selected based upon the second context 516.

[00152] Assembler 520 may assemble the first portion of the media, the first content, the second portion of the media and/or the second content to generate a

supplemented media 522 comprising a combination of the first portion supplemented with the first content and the second portion supplemented with the second portion. For example, the first portion of the media, the first content, the second portion of the media and the second content may be merged, concatenated and/or otherwise combined with one another (e.g., temporally, and/or spatially, and/or other ways of combination), and the supplemented media 522 comprising the combination of the first portion of the media, the first content, the second portion of the media and/or the second content may be generated.

[001 53] An embodiment of supplementing a video with content is illustrated by an example method 550 of Fig. 5C. A website, an application, etc. may provide a platform for supplementing the video with content (e.g., advertisements) using a server (e.g., of the website, the application, etc.). The server may host uploaded videos, and the website may provide access to view and/or hear the uploaded videos to an audience. Accordingly, at 524, a first content may be selected (e.g., based upon context, etc.) from a plurality of contents for the video.

[00154] At 526, a first area (e.g., and one or more additional areas) may be selected from a plurality of areas in the video based upon image analysis of the video. For example, image analysis may be performed upon one or more frames of the video to identify the first area (e.g., as a location suitable for placing supplemental content). It should be appreciated that 524 may happen before, after or concurrently with 526.

[001 55] In an example, the first area may be selected responsive to determining, based upon the image analysis, that the first area has a focus below a focus threshold. For example, a determination may be made that areas that are more out-of-focus are less likely to be important and/or result in disruption, inconvenience, etc. if they are overlaid by content.

[001 56] In an example, the first area may be selected responsive to determining, based upon the image analysis, that the first area comprises a first image feature. For example, a determination may be made that an area of a video representative of a side of a truck may be an appropriate location to overlay content (e.g., associated with a truck).

[001 57] In an example, the first area may be selected responsive to determining, based upon the image analysis, that the first area does not comprise a representation of

a face. For example, a determination may be made that areas that display a face (e.g., of a person, animal, character, etc.) are likely to be important and/or result in disruption, inconvenience, etc. if they are overlaid by content.

[001 58] In an example, the first area may be selected responsive to determining, based upon the image analysis, that the first area has a texture below a texture threshold. For example, a determination may be made that areas that have a low level of texture are less likely to be important and/or result in disruption, inconvenience, etc. if they are overlaid by content. In another example, the first area may be selected responsive to determining, based upon the image analysis, that the first area has a texture above a texture threshold or within a range of texture.

[001 59] In an example, the first area may be selected responsive to determining, based upon the image analysis, that the first area has motion below a motion threshold. For example, a determination may be made that areas that have a low level of motion are less likely to be important and/or result in disruption, inconvenience, etc. if they are overlaid by content. In another example the first area may be selected responsive to determining, based upon the image analysis, that the first area has motion above a motion threshold or within a range of motion.

[001 60] At 528, the video may be supplemented with the first content at the first area (e.g., and the one or more additional areas). The first area may be a frame, a combination of frames, and/or a region (e.g., top, bottom, side, etc.) of one or more frames of the video. In an example, the video may be supplemented with the first content at the first area through image overlay (e.g. overlaying advertisement).

[001 61] In one example, method 550 may be implemented in the context of a movie theater, where the media may be a movie. In another example, method 550 may be implemented in the context of television programming, where the media may be (e.g., live) television. In yet another example, method 550 may be implemented in the context of educational material, where the media may be an educational lecture.

[001 62] An embodiment of supplementing a video with content is illustrated by an example method 575 of Fig. 5D. A website, an application, etc. may provide a platform for supplementing the video with content (e.g., advertisements) using a server (e.g., of the website, the application, etc.). The server may host uploaded videos, and the website may provide access to view and/or hear the uploaded videos to an audience.

Accordingly, at 530, a first content may be selected (e.g., based upon context, etc.) from a plurality of contents for the video.

[00163] At 532, the video may be supplemented with the first content. For example, the first content may be overlaid and/or displayed concurrently with the video. In another example, the first content may be played before, after or in between frames of the video.

[001 64] At 534, one or more properties of the first content may be adjusted based upon image analysis of the video. For example, the first content may be displayed across one or a plurality of frames of the video, and may be (e.g., dynamically) adjusted across the plurality of frames based upon the image analysis of each frame.

[00165] In an example, a color of the first content is adjusted (e.g., between a first frame and a second frame of the video) based upon the image analysis. For example, the first content may be adjusted from a first color to a second color (e.g., on the first frame, and/or to a third color on the second frame, and/or the first content may be left the first color on a third frame).

[00166] In an example, a transparency of the first content is adjusted (e.g., between a first frame and a second frame of the video) based upon the image analysis. For example, the first content may be adjusted from a first transparency to a second transparency (e.g., on the first frame, and/or to a transparency color on the second frame, and/or the first content may be left the first transparency on the third frame).

[001 67] In an example, a size of the first content is adjusted (e.g., between a first frame and a second frame of the video) based upon the image analysis. For example, the first content may be adjusted from a first size to a second size (e.g., on the first frame, and/or to a third size on the second frame, and/or the first content may be left the first size on the third frame).

[001 68] In an example, a duration of the first content is adjusted (e.g., between a first frame and a second frame of the video) based upon the image analysis.

[00169] An embodiment of generating a representation of a (e.g., karaoke) performance is illustrated by an example method 600 of Fig. 6A. A website, an application, etc. may provide a platform for generating a representation of a performance using a server (e.g., of the website, the application, etc.). The server may

host performances, and the website may provide access to view and/or hear the performances (e.g., live, recorded, etc.) to an audience. Accordingly, at 604, a request may be received to implement a performance with a first user and a second user (e.g., and one or more other users). The first user and the second user (e.g., and the one or more other users) may be at one or more (e.g., different) geographical locations.

[00170] At 606, a determination may be made that the first user is associated with a first type of participation in the performance. For example, the first user (e.g., or another user) may request that the first user participate in the performance with the first type of participation, the first user may (e.g., randomly) be assigned the first type of participation (e.g., from amongst a plurality of types of participation), and/or the first user may be assigned the first type of participation based upon one or more scores, one or more past performances and/or one or more games (e.g., played against the second user and/or one or more other users before the performance).

[00171] At 608, a determination may be made that the second user is associated with a second type of participation in the performance. For example, the second user (e.g., or another user) may request that the second user participate in the performance with the second type of participation, the second user may (e.g., randomly) be assigned the second type of participation (e.g., from amongst a plurality of types of participation), and/or the second user may be assigned the second type of participation based upon one or more scores, one or more past performances and/or one or more games (e.g., played against the first user and/or one or more other users before the performance). The second type of participation may be different than or the same as the first type of participation. Types of participation may correspond to singing, dancing and/or playing one or more instruments.

[00172] At 610, a first content may be selected from a plurality of contents for the first user based upon the first type of participation, and a second content may be selected from the plurality of contents for the second user based upon the second type of participation. For example, the first content may comprise a first version of a soundtrack (e.g., associated with the first type of participation- e.g., a version of the soundtrack for singing) and the second content may comprise a second version of the soundtrack (e.g., associated with the second type of participation- e.g., a version of the soundtrack for dancing). In an example, the first content may be the same as the second content.

[00173] At 612, the first content may be provided to the first user, and the second content may be provided to the second user. For example, the server may send the first content to a device of the first user via a first (e.g., network) connection, and/or may send the second content to a device of the second user via a second (e.g., network) connection. In another example, a local computer or karaoke machine can supply both first content and second content in a merged way to both the first user and the second user who are at the same geographical location.

[00174] At 614, a first signal may be received from the first user in association with the performance, and a second signal may be received from the second user in association with the performance. For example, the first signal may comprise an acoustic signal comprising a representation of the first user singing and/or the second signal may comprise a visual signal comprising a representation of the second user dancing. The server may receive the first signal from the device of the first user via the first (e.g., network) connection (e.g., or a third connection different than the first connection) and/or may receive the second signal from the device of the second user via the second (e.g., network) connection (e.g., or a fourth connection different than the second connection). It should be appreciated that the server can be a local computer or karaoke machine, and the first connection and second connection can be local wired/wireless connections.

[00175] At 616, a representation of the performance may be generated based upon a combination of the first signal, the second signal, the first content and/or the second content. For example, a video of the performance playing the soundtrack combined with audio of singing by the first user and images of dancing by the second user may be generated and/or provided for display to the first user, the second user, an audience, one or more judges, etc.

[00176] Fig. 6B illustrates an example of a system 650 for generating a representation of a (e.g., karaoke) performance. Performance creation component 618 may create a performance responsive to a request from one or more users, such as a first user, Jack, and a second user, Jill, and/or randomly (e.g., for one or more users accessing a common application, service, etc.). The request may include a name for the performers of the performance, such as a band name, for example. Participation manager 620 may determine that the first user, Jack, is associated with a first type of participation 622, such as singing, and that the second user, Jill, is associated with a

second type of participation 624, such as dancing. Participation manager 620 may also determine whether Jack and Jill are at the same geographical location or different geographical locations.

[001 77] Content selector 626 may (e.g., in parallel) select a first content from a plurality of contents based upon the first type of participation 622 and a second content from a plurality of contents based upon the second type of participation 624. For example, a first version of a soundtrack customized for singing may be selected based upon the first type of participation 622, while a second version of the (same) soundtrack customized for dancing may be selected based upon the second type of participation 624. It may be appreciated that in some examples, the first content and the second content may be a same content (e.g. when Jack and Jill are at the same geographical location), while in other examples, the first content and the second content may be different soundtracks, images, videos, and/or different types of media.

[00178] The first content may be provided to the first user 628, Jack, while the second content may be provided to the second user 630, Jill. For example, the first version of the soundtrack may be played to Jack, while the second version of the soundtrack may be played to Jill. A first signal 632, such as a (e.g., audio) signal of Jack singing along with the first version of the soundtrack, may be received from the first user 628. A second signal 634, such as a (e.g., video) signal of Jill dancing to the second version of the soundtrack, may be received from the second user 630.

[001 79] Assembler 636 may assemble the first signal 632, the second signal 634, the first content and/or the second content (e.g., and/or third content comprising a third version of the soundtrack) to generate a representation of the performance 638. The representation of the performance 638 may comprise, for example, a video displaying Jack singing the soundtrack, Jill dancing to the soundtrack (e.g., and another user, Jane, playing an instrument in the soundtrack) and the soundtrack being played in the background. For example, the first signal 632, the second signal 634, the first content and/or the second content may be merged, concatenated and/or otherwise combined with one another, and the representation of the performance 638 comprising the combination of the first signal 632, the second signal 634, the first content and/or the second content may be generated.

[001 80] An embodiment of navigating through media (e.g., a video) is illustrated by an example method 700 of Fig. 7A. A user, such as user Jill, (e.g., and/or a device associated with the user) may access and/or interact with a website, an application, etc. that provides a platform for searching the media using a server or a local computer (e.g., of the website, the application, etc.). The server may host uploaded media, and the website may provide access to view and/or hear the uploaded media to an audience. It should be appreciated that the media may be hosted locally or on the networked server. Accordingly, at 704, a request to move (e.g., drag) a control along a first axis from a first portion of the first axis to a second portion of the first axis may be received (e.g., by the server and/or from the user). The media may comprise video, audio, an image, a document and/or an application interface.

[001 81] At 706, responsive to determining that the control is being moved (e.g., dragged) along the first axis within the first portion of the first axis, the media may be navigated through at a first rate of advancement based upon a first feature of the first portion of the media. For example, the media (e.g., video) may be forwarded or rewinded at a first rate (e.g., frames per second).

[001 82] At 708, responsive to determining that the control is being moved (e.g., dragged) along the first axis within the second portion of the first axis, the media may be navigated through at a second rate of advancement based upon a second feature of the second portion of the media. For example, the media (e.g., video) may be forwarded or rewinded at a second rate (e.g., frames per second). The first rate of advancement may be different than (e.g., or the same as) the second rate of advancement. For example, portions of the media determined to be more (e.g., above a threshold) important, popular, exciting, etc. may be navigated through at a slower rate of advancement than portions of the media determined to be less (e.g., below the threshold) important, popular, exciting, etc.

[001 83] In another example, the axis itself can be represented nonlinearly in the graphic user interface, with the first pixel distance between the first location to the second location representing a first rate of advancement, and the second pixel distance between the second location to the third location representing a second rate of advancement, wherein the first rate of advancement may be different than (e.g., or the same as) the second rate of advancement (e.g. the user jumps from location A to another location B with more precision between location A and C wherein location B is between

location A and location C, but jumps from location B to another location D with (e.g., relatively) less precision between locations B and E (e.g., in comparison to between A and C) wherein location D is between location B and location E).

[001 84] In an example, the first axis may correspond to (e.g., a spectrum of) time. For example, a first point on the axis may correspond to a first timestamp of the media, a second point on the axis may correspond to a second timestamp (e.g., after the first timestamp) of the media, etc. In an example, a rate of advancement may comprise temporal resolution.

[001 85] In an example, the media that is navigated through may comprise a list of contacts (e.g., of the user). The first feature may comprise a frequency of contact between the user and a first contact in the list of contacts, the second feature may comprise a frequency of contact between the user and a second contact in the list of contacts, etc.

[001 86] In an example, the media that is navigated through may comprise a list of messages (e.g., of the user). The list of messages may comprise text messages, instant messages, email messages, etc. The first feature may comprise a feature of a first message in the list of messages, the second feature may comprise a feature of a second message in the list of messages, etc. A feature of a message may comprise a contact frequency between a receiver and sender of the message, whether the receiver is a TO recipient or a CC recipient of the message, an importance of the message, a timestamp of the message, a length of the message, a domain name of the sender, a subject of the message, a signature of the sender, whether the message is an initial message, whether the message is a reply message, whether the message is a forwarded message, a number of recipients of the message, user data, user history, how frequent the user replied to previous messages from the sender, how soon the user replied to the previous messages of the sender after the user saw the messages, the length of the previous messages between the receiver and the sender.

[00187] In an example, the first feature and/or the second feature may be determined based upon information of the media. For example, the first feature and/or the second feature may be determined based upon text, an image, audio, comments, tags, titles, a transcript, cross-references, data analytics from a plurality of users, recommendations, reviews and/or user history associated with the media.

[001 88] In an example, the first feature may correspond to a first distance of (e.g., a first instance of) a focus point from the first portion of the first axis and/or the second feature may correspond to a second distance of (e.g., a second instance of) the focus point from the second portion of the first axis. For example, if the first instance of the focus point is a first distance (e.g., 2 inches on the computer display) away from the first portion of the first axis (e.g., along a second axis different than (e.g., perpendicular to) the first axis) and the second instance of the focus point is a second distance (e.g., less than the first distance) (e.g., 1 inch on the computer display) from the second portion of the first axis (e.g., along the second axis), the first rate of advancement may be greater than the second rate of advancement.

[00189] In an example, a representation of the moving of the control along a representation of the first axis may be provided for presentation, for example, as part of a playback interface.

[001 90] Fig. 7B illustrates an example of a system 750 for navigating through media. An interface 755 may be displayed on a device of a user. The interface 755 may, in some examples, display an application, such as a media (e.g., video) player, on the interface, which may include a media display portion 702 within which a media may be played and/or a media control bar 704 as part of a playback interface. In some examples, the interface 755 may further display information about a source of the media, a control that when selected enables sharing the media, one or more other recommended media and/or a media upload button, which may be selected by the user to upload one or more videos to a server associated with the application.

[001 91] It may be appreciated that while the control is moved along a first axis (e.g., the media control bar 704), the media in the display portion 702 and/or in a preview box (e.g., displayed overlaying the display portion 702 and/or the media control bar 704) may be updated to reflect the movement of the control and/or to present (e.g., display) which portion and/or frame of the media would be played if the control was released at that instant. While the control is being moved, the rate of movement (e.g., the updating of the media displayed) may be different (e.g., faster) than a normal rate of playing the media (e.g., to enable the user to identify a desired part of the media without having to view and/or listen to the media from the beginning).

[001 92] In a first instance 710 of the interface 755, a control at a first location 706 in the media control bar 704 may be moved (e.g., dragged) to a second location 708. Responsive to determining that the second location 708 is within a first portion 708 of the media, the updating of the media displayed may be at a first rate of advancement (e.g., 3x).

[001 93] In a second instance 712 of the interface 755, the control at the first location 706 in the media control bar 704 may be moved (e.g., dragged) to a third location 712. Responsive to determining that the third location 712 is within a second portion 714 of the media, the updating of the media displayed may be at a second rate of advancement (e.g., 10x).

[001 94] In yet another example, the user can click on the locations on the first axis of the media control to jump between locations on the first axis. The first axis may be represented nonlinearly, wherein the first rate of advancement between the first location and second location may be different than (e.g., or the same as) the second rate of advancement between the second location and third location. The user can click to jump to a fourth location that is between the first location and second location at the first rate of advancement, and/or click to jump to a fifth location that is between the second location and third location at the second rate of advancement.

[001 95] In some examples, at least some of the disclosed subject matter may be implemented on a client (e.g., a device of a user), and in some examples, at least some of the disclosed subject matter may be implemented on a server (e.g., hosting a service accessible via a network, such as the Internet).

[001 96] Fig. 8 is an illustration of a scenario 800 involving an example non-transitory machine readable medium 802. The non-transitory machine readable medium 802 may comprise processor-executable instructions 812 that when executed by a processor 816 cause performance (e.g., by the processor 816) of at least some of the provisions herein. The non-transitory machine readable medium 802 may comprise a memory semiconductor (e.g., a semiconductor utilizing static random access memory (SRAM), dynamic random access memory (DRAM), and/or synchronous dynamic random access memory (SDRAM) technologies), a platter of a hard disk drive, a flash memory device, or a magnetic or optical disc (such as a compact disc (CD), digital versatile disc (DVD), or floppy disk). The example non-

transitory machine readable medium 802 stores computer-readable data 804 that, when subjected to reading 806 by a reader 810 of a device 808 (e.g., a read head of a hard disk drive, or a read operation invoked on a solid-state storage device), express the processor-executable instructions 812. In some embodiments, the processor-executable instructions 812, when executed, cause performance and/or implementation of an embodiment 814, such as at least some of the example method 400 of Fig. 4A, the example method 500 of Fig. 5A, the example method 550 of Fig. 5C, the example method 575 of Fig. 5D, the example method 600 of Fig. 6A and/or the example method 700 of Fig. 7A, for example, and/or at least some of the example system 450 of Fig. 4B, the example system 525 of Fig. 5B, the example system 650 of Fig. 6B and/or the example system 750 of Fig. 7B, for example.

[001 97] Further embodiments

[001 98] Short summary and table of contents:

[001 99] The disclosed subject matter solves these problems:

[00200] A) Transcript-based video search and navigation (search video, navigate in video based on transcripts)

[00201] B) Transcript-based contextual advertisement (display relevant ad based on transcripts), automatic selection of regions of ad-overlay, and automatic adjustment of overlaying advertisement

[00202] C) Multimodal karaoke: (how to integrate singing, dancing, instrument playing together, also has a transcript search capability based on lyrics)

[00203] D) Non-linear video navigation: (how to navigate through video manually in a precise and easy way, especially on mobile devices.)

[00204] Table of Contents:

[00205] 1. Introduction

[00206] 2. Search with Time-associated text: methods and algorithms

[00207] 3. Videomark technology: conveniently manage and navigate through contents

- [00208] 4. Time-associated text search and media navigation control (aka. Search and play)
- [00209] 5. Phone and media chat recording/voicemail search (aka. Search in voicemail and video chat)
- [00210] 6. Audiobooks/Podcasts/Radio search and management
- [0021 1] 7. Integrated Publishing: text and media (aka. Navigate and search in text and media data conveniently)
- [00212] 8. Other time-associated data
- [00213] 9. Enhanced media enjoyment, composition and editing
- [00214] 10. Recommendation systems based on Time-associated text information and media search
- [00215] 11. Dynamic Contextual Advertisement based on time-associated information, automatic selection of regions of ad-overlay, and automatic adjustment of overlaying advertisement
- [00216] 12. Text-/audio-gated media coding rate
- [00217] 13. Virtual reality (VR) and augmented reality (AR) applications
- [00218] 14. Voice memo/voice recording software with search based on Time-associated text information
- [00219] 15. Multimodal karaoke:
- [00220] 16. Non-linear video navigation
- [00221] 17. Hardware implementation
- [00222] 1. Introduction
- [00223] Media search may be done with queries in fields such as title. The Time-associated text information such as transcripts, legends, captions, lyrics, subtitles may not be used in existing search technologies such as search engines. Furthermore, the current media players, either online or offline, either hardware-implemented or software-implemented, do not provide a convenient way to locate specific words or phrases in the Time-associated text information such as transcripts, for accurate localization and media navigation. For instance, if the word "computer"

is mentioned in the speaker's speech in the video but not in the title and playlists names, the current search will not work. Current web search may be limited to searching video and audio search with title or descriptions, not the content of the media works themselves.

[00224] Time-associated text information, such as transcripts, legends, captions, subtitles and annotations can provide important information for content searching in media. In one embodiment of disclosed subject matter, the search can be done purely based on Time-associated text information. For instance, with the disclosed subject matter, the search engine can search different videos based on the presence of the query in the transcripts of the videos. The search engine will subsequently return the videos where the query words have presences in the video transcripts. In another embodiment, the search engine can ask user to input a combination of queries in different fields, where at least one of the said queries is for the field search of transcript. For instance, a query "happy" in the field of "title" and a query "lab" in the field of "transcript" may be inputted by the user for a combined search for desirable videos. The engine will subsequently return the results of relevant videos (or segments of videos, playlists, channels, etc.) based on the search, where the 2 queries are found in the title and transcript, respectively. The results may be ranked based the relevance or other attributes such as time for uploading. In yet another embodiment, other text fields which might not be necessarily directly corresponds to the transcript of the audio, such as descriptions, comments, and tags, may also be included in the search. For instance, a query "happy" in the field of "tag" and a query "lab" in the field of "transcript" can be inputted by the user. The search engine will subsequently return results of relevant videos (or segments of videos, playlist, channels, etc.) based on the search, where either the first query is found in the tag or the second query is found in the transcript, respectively. The search engine will have an algorithm to rank the relevance of results, which will be described in the later part of the disclosed subject matter. In yet another embodiment, the user stores his/her family videos on local harddisk drive and a computer program scans over all those videos for the user to quick find out his/her family videos dated years back by searching.

[00225] Media referred to in the disclosed subject matter uses a combination of different content forms. Media may include a combination of text, audio, still image,

animation, video, or interactive content forms. It should be appreciated that the disclosed subject matter is applicable to any media, such as files online or local, as well as YouTube, MP3, Quicktime, AAC, radio, DVDs, Blu-Ray, TV, virtual reality, Netflix, Amazon Video, HBO Now, or any content-distribution service, in physical media, in the cloud or in cyberspace. It should be further appreciated that the media are intended for use on a computer (including smartphones, tablets, etc.) or on any other hardware such as DVD players, Karaoke players, video streaming sticks, or TV set-top boxes. It should be further appreciated that many hardware implementation can contain a computing unit. One of such hardware-implemented examples is a Karaoke player that has an embedded system/computer usually using a micro-controller.

[00226] It should be appreciated that all media, such as video, audio, games or any other format that carries Time-associated text information will benefit from the disclosed subject matter. In the disclosed subject matter, Time-associated text information is defined as any text information that is associated with the sound, speech or any other information-carrying signals. We can further expand the text information to be any time-associated information that may be represented using text, such as the annotation of a movie (although it is not audio-associated, it is time-dependent) or an object that can be expressed in natural language (e.g., "Taipei 101" in the movie Lucy).

[00227] With Time-associated text information such as transcripts (including subtitles, closed captions, lyrics, annotations, or equivalents of aforementioned) of media entities available, when users search for a plurality of queries, the search can go into Time-associated text and provide matching results in the Time-associated text which carries additional information related to the audio/video, unlike current search where only text outside of the media entities, such as title, descriptions, or tags are used. It should be appreciated that the aforementioned search method in audio-related text can also be expanded to all kinds of media content search, including but not limited to search functions built into or outside of content distribution websites, such as YouTube.com, Netflix.com, Amazon.com Video, HBO on Demand, Apple Store, Google Play, etc.

[00228] Definition of software: In the disclosed subject matter, software refers to programs or codes that can run a computer, or other electronic devices. In one

aspect, the software can be implemented as a program that can run on different operating systems or without operating system. Note that an operating system itself is a program, and it may support to the disclosed subject matter (e.g., via a built-in function or user interface) to certain extent. In another aspect, the software can be implemented at the hardware level, such as using a field-programmable gate array (FPGA) or specific electronic circuits.

[00229] Definition of timestamps: In the disclosed subject matter, a timestamp refers to a specific point on the time axis of the media. For instance, a timestamp when the word "hello" is said can be expressed in absolute time, such as 5 minutes 55 seconds (counted from the beginning of the video). Likewise, other ways to encode timestamp is also possible.

[00230] 2. Search with Time-associated text: methods and algorithms

[00231] The basic processing pipeline for the Time-associated text search is illustrated below. The software will take a plurality of user inputted queries, match queries with Time-associated text information in media on text domain (e.g., transcript), audio-domain (e.g., the sound waveforms), and visual-domain (e.g., video frames), and return the results to user through a list or GUI-powered (graphic user interface powered) visualization. The user can select the result of interest (such as a video or a video segment) for viewing.

[00232] Figure 9: Flowchart of search with Time-associated text

[00233] 2. 1 Submitting queries

[00234] Each query consists of parts known as terms. Each term can form a query, e.g., two words can be a query while each word is a term. A term can also be a relationship on terms. For example, "not "happy"" can be a term meaning the word "happy" shall not appear in search results, where not is treated as a keyword for a logical relationship and the word "happy" itself is also a term. In the disclosed subject matter, we allow time-related terms in the search query and they can be composed (e.g., binary search using AND and OR) with any existing search terms in a search query (e.g., words in a query cannot be more than 5 seconds apart). A term doesn't have to be in the form of text or formatted string. It can be media too, e.g., an audio clip (as in Apple Siri or Microsoft Cortana) or an image (as in Wolfram Mathematica syntax where images can be operands of functions).

[00235] A query is searched against a reference. Similar to terms, a reference does not have to be limited to text, but a media work of multiple media, e.g., a video that comes with transcripts in two languages. A result of searching for a query against a reference is called a match. A match is collection of terms and their timestamps such that the query is fully or partially satisfied. A query can be searched against one reference or multiple references, returning one or more matches.

[00236] The user can specify searching different terms at different media field. For example, the user can specify to search one word in the title and a two-word phrase in the closed caption of a media work. A field can be time-advancing with playing of the media, e.g., English transcript, Chinese transcript, left-channel audio, right-channel audio, video frames, etc.; or a field can be time-independent, e.g., the title, the description, etc. A field could also mean a constraint to apply onto terms, for example, we can have a field called "must have exact words" or "not including the following words". The field can include human annotation, from the content creator, the viewers and distributors, such as Amazon X-ray, providing trivia and backgrounds about the storyline and actors.

[00237] Treating language as a field allows the user to search for the Time-associated text in specific languages. For instance, if user input the query "hola" in the field "transcript" and query "Spanish" in the field "Ho-la", the software will return the results from clips with corresponding Spanish words of "Hola" being said. Similarly, dialect such as Cantonese or Hakka can also be specified in the search field. This function re-quires the identification of the language in media entities. This problem is also known as language identification. In one aspect, the language detection can be done in audio domain. In another aspect, the language detection can be done in text domain, e.g., using n-gram classification models. In yet another aspect, the language detection can be done based on tags from users or the transcripts.

[00238] The input method may be based on text input, voice input, body gesture, brain-computer interface, or any other input methods. Keyboards (physical or software), mouse, voice, touchpads, microphones, RGB-D camera (e.g., Microsoft Kinect), audio-based controller (e.g., Amazon Alexa or Dot), or other input devices may be used. The input can be done on one computer (e.g., a mobile smartphone or an tablet computer) while the result is shown on another computer (e.g., a Google

Chromecast). The input device might also be far away from the output device (e.g., a phone at Mountain View for input and a tablet computer at Redmond for output).

[00239] In one embodiment, the software will allow user to simply input a plurality of words as the query, and the software to determine whether the words inputted are individual terms or a plurality of the words are phrases where a compound word should be considered as a term. Detecting a compound word has been well studied in NLP, in problems such as collocation. One approach is to use item names in Wikipedia, e.g., Illinois Wikifier.

[00240] Terms-and-connectors search: Users are empowered to search in multiple terms and connect them using different connectors in the query. In one embodiment, the syntax of terms-and-connectors, including but not limited to, the following forms. Phrases in double quotes enclose a term in double quotes, "like this". Double quotes can define a single search term that contains spaces, e.g., "holly dolly" where the space is quoted as a character, differs much from holly dolly where the space is interpreted as a logical AND. Boolean connectors can be used to connect terms, such as (blue OR red) AND green, which differs from: blue OR (red AND green). Terms can be excluded by prefixing a hyphen or dash (-), which is "logical not". For example, payment card -"credit card" finds all matches with "payment" and "card", but not "credit card" yet would include "credit" separately. A wildcard character * can match any extra character-string, can prefix or suffix a word or string. For example, "*like" will match "childlike" or "dream-like". Spelling relaxation occurs by suffixing a tilde (~) like this-, with results like "thus" and "thins". This is also called search~ish. It should be appreciated that the aforementioned terms-and-connectors search method can be defined with other reserved words. It should be further appreciated that the terms-and-connectors search can be used in combination with different fields of search such as the field "transcript", "title", "tag" where each field specify where the corresponding terms will be searched for. For example, a user could inquiry to find all occurrences where the word "happy" appears in closed caption while the word "lab" appears in title.

[00241] Because the search terms may be time-related, we allow users to specify a new kind of fields: temporal constraints, which are relationships for timestamps. Via time-constrained search introduced by the disclosed subject matter, a user can specify temporal constraints in a query, or, search results can be presented to the users with temporal consideration, e.g., temporal distances (expressed in time) among query terms

in each match can be involved in ranking. This provide additional advantage compared to the capability of search techniques in prior arts. The temporal distance in the disclosed subject matter refer to the interval between event A and event B, where the said event can be a word, a phrase, a sentence in the timed transcript, a scene in the movie (e.g., fighting), or plot related events (e.g., Tom Cruise enters the scene). For example, if the word "happy" is said at a timestamp of 0 hour, 0 minutes 20 seconds and the word "lab" is said at a timestamp of 0 hour, 2 minutes 30 seconds, the temporal distance between "happy" and "lab" is 0 hour, 2 minutes 10 seconds. Thus, temporal constraints can be built based on temporal distance. For instance, we can specify a temporal constraint of "less than 30 seconds" between the words "happy" and "lab".

[00242] In one embodiment, the media is a video clip accompanied by a transcript where each word is associated with a timestamp while the said query is a phrase. The user can find the timestamp of matches in which 80% of the words in the query are at most 5 seconds apart, e.g., find all timestamps where at least 4 out of the 5 words in "happy lab good best award employee" are mentioned no more than 5 seconds apart. In another embodiment, the matches are ranked based on the proximity of temporal distances between all query terms in each match. In this case, if several successive words are too far away temporally from each other, they are not considered a match. For instance, when the queries "happy" and "lab" are inputted, only co-occurrence of both "happy" and "lab" in the transcript, along with a short temporal distance between the 2 words (e.g., < 10 seconds) will constitute a match. In another embodiment, the distance between words can be a combination (e.g., weighted sum, etc.) of temporal, lexical and semantic distances. Lexical distance can be defined as how many words in between, how many syllables in between, or even how many sentences in between, or their distances on the parsing tree, etc. for instance, in the sentence "this is a powerful patent", the lexical distance between "this" and "patent" is 3 words. The semantic distance can be defined in many ways. In one embodiment, the semantic distance can be defined as the average pair-wise vector distance between any two words, except stop/background words, 5 words up stream and 5 words down stream near the match. For example, if the query is to find all of the 2 words "happy" and "lab", and one match is in the phrase "Google Venture gives Happy Lab a \$20 Million valuation", the semantic distance will be the average vector distance of any pairs of the words in "google", "venture", "million", "valuation". The vector distance is defined as

the cosine of the angle between the two vectors representing the two words. Methods to generate vector representation of words include but are not limited to, Google word2vec, Stanford NLP Lab's Glove, etc.

[00243] In the disclosed subject matter, we provide powerful ways to users to express various temporal constraints via our proprietary TymeTravel syntax for developers and users to compose and send queries to a local device or a web server in their applications. In the disclosed subject matter, "TymeTravel syntax" refers to the method and syntax for ending time-related query. In one embodiment, the user can specify the approximate time range of each term, e.g., 'happy' around 3:00 AND 'lab' at 5:30 after 'happy', or "happy' no more than 30 seconds before 'lab'. The user can even specify the timestamp of which term to be returned, or they could generally specify using words, including but not limited to, "center", "leftmost", and "latest". For example, if the user searches for two words "happy" and "lab" and matches of "happy" and "lab" are found at 0.5 second and 0.6 seconds, the timestamp of 0.5 second will be returned if the user specifies to return the earliest timestamp among the two.

[00244] In another embodiment, the time constraints can be represented in a syntax other than natural languages. For example, "the word 'happy' no earlier than 30 seconds before the word 'lab' " can be expressed as $\text{time}(\text{happy}) - \text{time}(\text{lab}) < 30\text{s}$, or "the word 'happy' around 3 minutes 00 seconds and the word 'lab' after 5 minutes 30 seconds" can be expressed as $\text{happy} \sim 3:00 \text{ AND } \text{lab} @ 5:30+$. Some reserved word can be used to allow the user to specify constraints on the terms, such as "all-apart 5s" meaning that all words in the query must be 5 seconds apart temporally. Our syntax also allows temporal constraints to be expressed with respect to the entire duration of the media, e.g., $\text{happy} @ 25\%$ - means that the word "happy" needs to appear at the first quarter of the video (within the first 25% temporal duration). In another embodiment, a mixture of computer syntax style and natural language style is also allowed. It should be appreciated that the temporal query can be sent via HTTP, HTTP/s, or other Internet protocol, with or without encryption, through POST, GET or any other HTTP/s methods to transmit the temporal query. It should be also appreciated that the temporal query can be sent with the text query.

[00245] TymeTravel beyond end user - The use of TymeTravel syntax does not have to be limited to end users. Any information can be converted into/from TymeTravel syntax and be used by any part of the software system. For example, a

client program uses TymeTravel syntax to send the queries to a server. In another example, the user fills in different terms into a webform and then the program converts the inputs of the webform into TymeTravel syntax and send to a server. In another example, based on data, the program can automatically (without an explicit time-related search request from the user) synthesize a query in TymeTravel and send to a server. In yet another example, the TymeTravel syntax is used to exchange data/query/info between two different programs running on the same computer. In yet another example, the TymeTravel syntax is used to exchange information between a program and the APIs/libraries that it calls.

[00246] Varieties of TymeTravel Syntax

[00247] It should be appreciated that the TymeTravel Syntax is not a fixed grammar. It should be treated as a general grammar that allows users to specify time-related constraint in query terms.

[00248] Allow me to explain it using an analogy. Existing advance search tools (e.g., google advanced search https://www.google.com/advanced_search) provide syntax for users to specify the relationship between terms. For example "happy + lab", "happy & lab", and "happy AND lab" could all mean that both the term "happy" and the term "lab" should appear in the search result. But the syntaxes are different here due to the different operator/connector used (i.e., +, & and AND).

[00249] Similarly, in TymeTravel, the syntax does not have to limited to one form or one set of keywords. There can be unlimited possibilities to specify time-related query terms. For example, "happy"@25%+)and at("happy", after 1/4) could both mean that the word "happy" needs to appear in the last three quarters of the video. There are unlimited different key/command/control characters/words, and unlimited syntaxes to use them (including orders) to specify a time-related query.

[00250] Various syntax rules may be used to specify time-related terms, constraints or relationships.

[00251] Query beyond end user

[00252] The query does not have to be initiated by end users. Any information can be converted into/from query and be used by any part of the software system. For example, the user fills in different terms into a web-form and then the program converts

the inputs of the webform into a query and send to a server. In another example, based on data, the program can automatically (without an explicit time-related search request from the user) synthesize a query and send to a server. In yet another example, a query can be exchanged between two different programs running on the same computer. In yet another example, a query can be exchanged between a program and the APIs/libraries that it calls, where the APIs/libraries can be local or remote.

[00253] 2.2 Finding matches

[00254] As mentioned previously, the search does not have to be limited to Time-associated text information but all kinds of information, embedded in the media or outside of media, at all kinds of media forms, related with time or not. For example, an object recognized in a movie at a timestamp can be converted into a text describing it (eg. a picture of a chair can be recognized and converted to text information "a chair").

[00255] Text-based search/matching: In the embodiments of the disclosed subject matter where audio information are transcribed into text in this disclosed subject matter, many string matching algorithms can be employed to find the matches for a query, including but not limited to, Naive string searching algorithm, Rabin-Karp algorithm, Finite-state automaton search, Knuth-Morris-Pratt algorithm, Boyer-Moore algorithm, dynamic programming-based string alignment, Bitap algorithm, Aho-Corasick algorithm, Commentz-Walter algorithm, etc. Text-based matching results can be ranked based on the distances between the query and the matches, using distance metrics, including but not limited to, Hamming distance, editing distance, Levenshtein distance, Jaro-Winkler distance, most frequent k words, S0rensen-Dice coefficient, Jaccard similarity, Tversky index, Jensen-Shannon divergence, etc. Typos will be tolerated with spell correction suggestions. Text-based match can be expanded to include word related to the query terms. For example, when searching for "inn", other words "hotel" and "resort" will also be included. A common case in handling natural languages is that one word, phrase, or clause/sentence can match multiple words, phrases, or clauses/sentences. This is due to the ambiguity of natural languages. In one aspect, for a query not having exact match in the transcript, its equivalent counterpart could be in the transcript and will be returned. Furthermore, the temporal distance concept previously described can also be used for text-based matching.

[00256] In one embodiment, text match can be searched using text alignment algorithms studied in NLP. Query terms, individually, as a partial group or as a whole, can be aligned against the time-associated text. Methods and tools for text alignment include but are not limited to, Smith-waterman algorithm, Jacana-align, Semi-Markov Phrase-based Alignment, IBM models, MANLI aligner, etc.

[00257] Audio-based search/matching: In another embodiment, the search matching will be done in the audio domain rather than the text domain. The matching algorithm will operate based on audio signals directly. The query from the user, can be an audio clip, either from the user's audio input or an artificial audio clip converted/synthesized from text input. The audio query can be directly aligned/searched against the audio of the media stream and the playing will begin from the timestamp of the beginning of the matching. Aligning audios can be done by Hidden Markov Models (HMM), Dynamic Time Warping (DTW), Neural Networks, Deep Learning Models, etc. In one embodiment, the audio input or text input may be translated by the software from one language to another one for the search. For example, "Hola" in Spanish may be translated to "Hello" in English for the search. The match for audio can be done in time domain, frequency domain, time-frequency joint domain, or the combination of thereof, via any necessary transforms, e.g., Fourier Transform, Short-term Fourier Transform, Wavelet transform.

[00258] Image-based search/matching: In yet another embodiment, the search matching will be performed in the image domain. The matching algorithm will operate not based on text but based on images and video frames. The query of the user, can be a plurality of video clips or a plurality of images, either uploaded from existing data or taken at the time of search. The image query can be directly aligned/searched against the video frames of the media stream and the playing will begin from the timestamp of the beginning of the matching. Searching for images can be done by Object recognition algorithms, including but not limited to, Histogram of Oriented Gradients (HOG), Aggregated Channel Features (ACF), Viola-Jones algorithms, SURF, etc. Ranking can be done based on the matching scores.

[00259] Hybrid search: Instead of making use of the text, audio or video independently for searching and matching, in another embodiment we propose a media joint search and data mining scheme. When a user specifies the query, they can define some terms in text and some other terms in video. For example, "Steve Jobs" in the text

query field and the "iMac G3" in the video/image query field can be specified by a user. The text searching algorithm will find matches for "Steve Jobs" in text while the image recognition algorithm will detect objects of iMac G3 in the video stream. Users can set different priority levels for different types of search terms. User feedbacks or tagging can also be used to teach the computers.

[00260] In order to efficiently find results for queries that involve time on multi-channel/-modality/-field media/data, we propose several algorithms here. Note that the algorithms below assumes that a valid result has all elements in the query satisfied. As an option, the system should allows partial matching of the query , e.g., if ranking score significantly increases after dropping some terms or if failed to find any match that completely satisfy the query. Note that specific algorithms for searching terms in a particular types of media, e.g., searching phrases in a text body, are discussed in a separate section as follows.

[00261] 2.2.1 Algorithm 1 (Brutal force)

[00262] In this algorithm, we first discretize the time throughout the duration of the media (e.g., 0.1 second per step), and then check the query at each discretized timestamp. For search terms, we check whether there is any match at each discretized timestamp, respectively. For instance, if the query of "happy" is inputted, the software will look for a match for "happy" at each discretized timestamp. For terms that have temporal relationships, such as term 1 and term 2 in the query being 3 seconds apart, the algorithm will look forward and/or backward along time axis to check the existence of terms and whether their temporal relationship is satisfied.

[00263] 2.2.2 Algorithm 2 (find each term first and then check the satisfactory of query at timestamps of terms)

[00264] This algorithm comprises of two steps. In Step 1, the software finds matches for each query term in each one field. By "field", we refers to an organization of data that follows time axis, e.g., the transcript in one language or a stock price of one trade symbol, such as transcript in Chinese or stock price of Apple inc.. In Step 2, we check whether the query is satisfied at each timestamp where at least one term is matched. Term herein refers keywords, audio waveforms, images, videos or other items composing the query for the search. If matched, and the user specifies how the return time should be extracted, e.g., "the center timestamp of all terms", the specific

timestamps will be returned. If matched but the user does not give a preference on the return timestamp, the timestamp of the earliest term in the query may be returned as the default option.

[00265] There are multiple ways to check the satisfiability of the query in Step 2. In one embodiment (the most straightforward but time-consuming way), the combinations of all timestamps, each of which matches at least one term, and terms associated with those timestamps are enumerated and checked against the query. This approach is illustrated using the example below where the query is finding two words "happy" and "lab" in the transcript and the temporal distance between the matches of the two words must be under 5 seconds.

[00266] Figure 10: An example of matching multiple terms involving time

[00267] Because this exhaustive approach is time consuming, in the disclosed subject matter another embodiment is described (slide temporal window approach). In one embodiment, a sliding temporal window is used and the query is checked within each incremental temporal position of the sliding window along the time axis. For example, if the user wants to search for co-occurrences of the two words "happy" and "lab" that are no more than 5 seconds apart, we establish a +/- 5-second sliding temporal window for each occurrences of "happy" and "lab" and check whether both words appear in each position of the slide window. The sliding temporal window is established based on the time constraints inputted by the user or by default. In addition to the example just mentioned, here is another example: the temporal constraint is, using the "Tymetravel" syntax introduced above, "time(happy)-time(lab)<30". In this example, the sliding temporal window will be at least 30 seconds in temporal width to allow checking whether the word "lab" appears within 30 seconds after the timestamp of the word "happy". In one aspect, if the user does not specify any time constraint, a default value for the sliding window will be used, such as 10 seconds in temporal width, because in most cases of media enjoyment people care about a short event. In many cases, a plurality of matches will be found, and each match corresponds to a temporal position of the sliding window that satisfies the query. If the two sliding temporal windows have majority overlap (including subset/superset relationship) on terms, they are merged and treated as one sliding window. In one example, the majority overlap is defined as over 60% overlap. The example above is illustrated in the figure below.

[00268] Figure 11: Sliding-window based search

[00269] In yet another embodiment, instead of establishing a sliding temporal window, we first cluster timestamps of matching terms and then check the satisfaction of the query at each cluster. Clustering is a well-studied problem in machine learning. Approaches for clustering include but are not limited to connectivity-based clustering (e.g., single-linkage clustering, complete linkage clustering, average linkage clustering, etc.), centroid-based clustering (e.g., k-means clustering, etc.), distribution based clustering (e.g., expectation-maximization algorithm, etc.) and density-based clustering (e.g., DBSCAN, OPTICS, etc.). The clustering algorithm can consider temporal constraints and temporal distances in the query to avoid splitting two timestamps that can satisfy the query into more than one clusters. For example, if one temporal constraint is that $\text{time(happy)} - \text{time(lab)} < 30$, then we must merge two consecutive clusters containing "happy" and "lab". An example illustrating this embodiment is given below.

[00270] Figure 12: Cluster-based match finding.

[00271] The algorithm steps for sliding window and cluster are also shown below in Figure 13.

[00272] Figure 13: Left: Algorithm steps for using sliding window to check matches along time. Right: Algorithm steps for using clustering for checking matches along time.

[00273] In yet another embodiment, we use specially crafted data structures to speed up the search. In one aspect, a matrix is created for every timestamp of a matched query term, where each row corresponds to a query term and each column corresponds to a timestamp. The timestamps are sorted into ascending or descending order. Column by column, we first check the satisfaction of non-temporal terms at each timestamps, labeling those satisfied as 0's and those not as 1's to form a binary vector. Then with this binary vector, we check the temporal constraints at timestamps that are labeled as 1's. This embodiment is further explained in Section 2.2.5.

[00274] In all embodiments described for "Algorithm 2", the query satisfaction check can be done via encoding each query into a constraint satisfiability problem. Each search term will be represented as a variable.

[00275] In one aspect, if the condition "any of these words" is specified by the user, each term for the condition "any of these words" will be translated into powerset of all terms that must be included at least in the query. For example, the term "any of these words: happy lab" will be translated into 3 expressions: "happy", "lab" and "happy, lab". Each condition "all these words" or "exact word or phrase" term will be translated into expressions each of which is one term there. "None of these words" or logic operators will be translated according to logical expressions of the variables representing those terms. The Tymetravel syntax described in the disclosed subject matter has already discussed how to translate temporal constraints into mathematical expressions, e.g., happy@0.5+ is translated into a mathematical inequality representing a temporal constraint $\text{time}(\text{happy}) > 0.5$ and each of temporal constraints will be represented using one Boolean logical variable (eg. true vs false). The query will be encoded as mathematical constraints over those variables, representing temporal or non-temporal constraints. The constraint satisfiability problem can be solved using multiple programming methods, including but not limited to, Boolean Satisfiability Problem, Constraint Programming, Constraint Logic Programming, Answer Set Programming, etc. Part of or all timestamps that a query can be satisfied will be returned as results. As mentioned earlier, should multiple terms occur at different timestamps, the earliest time will be given to the user by default unless the user specify a different one.

[00276] Figure 14: The 3 embodiments of Algorithm 2. In practice, the combination of all search algorithms can be used. The preprocessing includes transcribing, translation, parsing, etc.

[00277] 2.2.3 Algorithm 3 (Pre-index)

[00278] The two aforementioned algorithms require just-in-time search which might be time consuming and repetitive. Another approach which is time saving, is pre-building an index for various queries. An index is a mapping from queries to their corresponding results. The queries for building index can be generated by computers automatically or from real queries that users input. By using index, the results can be acquired without doing the search again. An index is usually stored in a data structure called hash table. Using index to speed up data fetching has been widely used in searches, such as web search, data base search, etc.

[00279] In one embodiment, the disclosed subject matter enable the software to build and update indexes for queries for Time-associated text information, time-associated information, and other information. The queries can be of different lengths using various operators.

[00280] In one aspect, the indexes for temporal queries can be build and updated. Then for each non-temporal query, we enumerate the combinational of basic temporal constraints, e.g., temporal distances between terms, time elapses of terms with respect to the beginning, end and middle of the media. Combinations of different non-temporal queries and temporal constraints on them form the temporal queries that users may enter and hence are indexed. The media database used to build the index can be acquired in many ways. For content hosting websites, such as YouTube, Vimeo, Netflix, Hulu, they can just build index for all media contents. This applies to all cases of this disclosed subject matter where the content is local (even for those websites, contents are considered local to them because the data is stored in their storage, although the storage may be distributed over computer networks, e.g., Network File System, Lustre File System, etc.), including but not limited to, voicemail boxes, private media stored in private storage (e.g., all family videos on a mobile hard drive), media contents of textbooks or MOOC (Massive Open Online Course, e.g., Coursera or Udacity). If the media entity is not in the storage that belongs to or specially authorized to access by computing system which runs the indexing algorithm, then the computing system can crawl the contents by following links pointing to media entity. For example, the indexer can find a YouTube video as long as it can have the weblink to it. In one embodiment, the indexer program can generate queries to media content providers to obtain the weblink to all entities belong to or are distributed through that provider. In another embodiment, the indexer program can crawl by following the menus on the content provider's website, e.g., following menus of movies provided by Amazon Prime Video. In yet another embodiment, the indexer program can leverage search engines by submitting queries to them (such as Google Video search) and following links in the search results. It should be appreciated that a weblink does not necessarily mean a URL specified in HTTP protocol or URI specified by Android operating system. It can be any form of identification to point to a media entity.

[00281] The index can be pre-built and updated on a plurality of levels. In one aspect, the index can be built for a particular user (eg. Tom's files on all his computers,

tablets and smartphones). In another aspect, the index can be built for a particular group of users (eg. all the family members including the father, mother and kids; all students and teachers involved in a class). In yet another aspect, the index can be built and updated for the entire search engine, or a video content provider. In yet another aspect, the index can be built and updated for a local device (eg. the karaoke machine; a computer; a smartphone). In yet another aspect, the index can be built and updated in a cloud-based architecture. It should be appreciated that different versions of index can be stored, and recovery of earlier version index is possible, if needed.

[00282] 2.2.4 Algorithm 4 (One match on one term each time)

[00283] If the user just needs to find only one result without needing to have all the match results, aforementioned algorithms can be revised to accommodate this need. In one embodiment, the search software begins with only one query term and gradually add other query terms. When the first match of the first term is found, the second term will be added into search and the query satisfiability will be checked (e.g., within the sliding temporal window). New term will be added gradually (e.g., one at each time, or two at each time, or a plurality at each time) until the query is satisfied or until the query is dissatisfied. If the query is dissatisfied, the algorithm will begin from the first term at the next match occurrence. The search result, even for those dissatisfied/failed search, could be logged for ranking. In order to speed up the search, terms can be ranked based on computational time-complexity to search them (e.g., text searching is easier than recognizing objects in video frames), likelihood that a match for this term can be found (e.g., there could be more matches for the word "government" than "telescope" in a video of presidential debate), and other factors. The pseudocode below shows how to find exactly one match. Note that the pseudocode below does not use a data structure to log what combinations of matches have been searched. Hence it is slow, especially when we want to reuse it to search for multiple matches.

[00284] Input: Q (the list of terms in the query), C (the constraint equivalent to the query)

[00285] R := empty list

[00286] for each term T in Q (may rank them into order)

[00287] for each timestamp M of the term T

```

[00288]             if adding M still satisfies the constraint C,
[00289]                 Append M to R
[00290]             break // go to add the next term; else, check next
match of the term
[00291]         end if
[00292]     end for
[00293]     if R satisfies the query (or part of it but is sufficient)
[00294]         return R
[00295]     end if
[00296] end for
[00297] return R

```

[00298] In another aspect, a data structure, such as a tree, can be used to keep track of the search process. Such search process can be optimized using heuristics methods, including but not limited to, A* algorithm, unit propagation, DPLL, etc. The algorithms can also be used to find all matches, i.e., traverse all nodes on the search tree.

[00299] Suppose the query is "'happy' and 'lab' and 'time(happy)-time(lab) <5'" meaning that both the word "happy" and "lab" have to appear in a match and the time interval between "happy" and "lab" must be under 5 seconds. We will demonstrate how to use this tree-based search to exhaustively find all matches, as shown in the figure below. The algorithm will first pick one term, say "happy" and find its first match which is t_1 timestamp at 2.1 seconds. That is the left branch at "happy" level on the search tree. Since the term "happy" at timestamp t_1 does not violate/dissatisfy the query, the algorithm create one more level on the search tree for the next term "lab". The first occurrence of "lab" is found at time $t_2=3.2$ seconds, the leftmost branch at "lab" level on the search tree. Now, the query is fully satisfied and hence the first match is returned. Should the user just needs one match, this is the end. But since in this example all matches are needed, the algorithm will continue. It searches for the next occurrence of "lab" and finds a match at $t_4=11.6$. But since $t_4-t_1= 9.5$ violates the query, this result is discarded - marked by a void arrow on the illustration. Then the 3rd match of "lab" at

$t_5=18.7$ is checked. Again, the query is not satisfied because the temporal constraint $\text{time(happy)}-\text{time(lab)}<5$ is violated. Now, since all occurrences for "lab" have been checked, the algorithm trace back to search for the cases where "happy"@ t_2 , i.e., the right branch of the "happy" level on the search tree.

[00300] Figure 15: An illustration of query matching using a search tree. The matches for query terms are visualized on time axis while the search tree is given below the time axis. On the search tree, unsatisfied combinations of timestamps are in void arrows while satisfied in solid arrows. Solid lines represent nodes visited in the search process while dash lines for future visits. The illustration shows the search tree after checking "happy" @ t_2 and "lab" @ t_3 .

[00301] Many heuristics can be used to speed up the search process discussed in the disclosed subject matter. For example, the search tree can be pruned. In the demonstration above, the checking for "happy" @ t_1 and "lab" @ t_5 are not necessary after we learned that "happy" @ t_1 and "lab" @ t_4 dissatisfies the query - because t_5 is greater than t_4 and thus if t_4 cannot satisfy the temporal constraint, t_5 cannot satisfy the temporal constraint, either. In the example above, the trace back is only one level up on the tree. More intelligent methods can be used to prune the search tree or prioritize the branches on the search tree, including but not limited to, conflict-driven clause learning (originated from solving Boolean Satisfiability problems), cut-set based, smart backtrack, A* search, etc.

[00302] It should be appreciated that the temporal constraints can be user-defined or by default. For example, if user input "happy lab" without specifying the temporal constraints, the computer will use default setting (e.g. temporal constraint: time interval between the first keyword and second keyword is less than 10 seconds)

[00303] 2.2.4 An example

[00304] Here is an example using a combination of 2 of the embodiments of Algorithm 2.

[00305] Transcript of the media: "Let's take a look at how a volcano erupts. In Italy, over 300 volcano eruptions happen each year."

[00306] Query: 2 words out of the words "Italy", "volcano" and "erupt\V'eruption", and the temporal distance between the 2 words are at most 2 seconds apart.

[00307] Without losing generosity, we assume that each syllable lasts 0.4 second and the pause between two consecutive words is 0.2 second.

[00308] In Step 1, we run the search of each term (a match of each term is in bold font) and extract the timestamp associated with each word (in parenthesis):

[00309] Italy: Let's take a look at how a volcano erupts. In Italy (8.5s), over 300 volcano eruptions happen each year.

[00310] volcano: Let's take a look at how a volcano (5.9s) erupts. In Italy, over 300 volcano (9.9s) eruptions happen each year.

[0031 1] erupt/eruption: Let's take a look at how a volcano erupts (7.3s). In Italy, over 300 volcano eruptions (11.3s) happen each year.

[0031 2] For the sake of convenience, we store the result in a binary matrix where each row corresponds to a term while each column corresponds to the time for each term. The transpose of the matrix in Figure 16 also works.

[00313] In Step 2, we find matches that satisfy terms with or without rules connecting them. In one embodiment, we first go over the timestamps, each of which has at least one term matched, e.g., 8.5, 5.9, 9.9, 7.3, and 11.3 in the example above. Then we run clustering on the timestamps that at least one term is matched. The clustering condition is that no two timestamps in one cluster is more than 2 seconds apart. Hence we form 3 clusters: volcano (5.9s) + erupts (7.3s), Italy (8.5s) + volcano (9.9s), volcano (9.9s) + eruption (11.3s). The last two clusters have large enough overlaps in both time and terms and therefore are merged as one cluster. Lastly, we check whether non-temporal constraints are satisfied in each cluster. Here we use one method we mentioned earlier that we check column by column. Given the query is "2 words out of the words "Italy", "volcano" and "erupt\V'eruption", and the temporal distance between the 2 words are at most 2 seconds apart.", each cluster satisfies the query. Because the user does not specify what time in each cluster to return, the earliest time of each cluster is returned, i.e., 5.9s and 8.5s. As an alternative, without breaking sentences, the time of the first word belonging to each of the 2 cluster is returned, i.e.,

0.0s for "Let's" and 8.3 for "In". Finally, the two clusters can be ranked using some embodiments to be discussed as follows. For example, the 2nd cluster contains all 3 query terms while the first only has 1. If the only ranking criterion is coverage of query terms, then the 2nd cluster will be ranked as the 1st.

[00314] In the aforementioned example, smallest discretized timestamp interval is 0.1 second. It should be appreciated that the other values can also be applied as the smallest discretized timestamp interval, such as 1 second.

[00315] 2.3 Ranking results

[00316] Ranking search results has been intensively studied in the area of computer science and information systems. Here we list some approaches, some of which are our new disclosed subject matters while others are well known search result ranking methods.

[00317] In one embodiment, the results of the search will be ranked. The flowchart including ranking is shown below:

[00318] Figure 17: Flowchart of search with Time-associated text information with ranking method

[00319] A query could return many results and they need to be ranked to present to users. Existing text search rank text search results and this topic has been well-studied in areas such as information retrieval in traditional search such as website search. All those reported algorithms can be applied for our applications in ranking search results.

[00320] However, existing algorithms are for matching one type of media, e.g., text, video, image, only. With the methods in the disclosed subject matter we can rank matches on all types of media, equally or discriminatingly, e.g., different weights for matches found on different types of media. Later we will discuss how to calculate the ranking score using our method.

[00321] With our method, we can treat all entities such as media as text entities. Consequently, combining our method with existing methods, the ranking of media can be performed with enhanced accuracy. The ranking algorithms that can be used with our methods include, but are not limited to, Inverse Term Frequency (IDF), Term Frequency - Inverse Term Frequency (TF - IDF), Okapi BM25, cosine similarity between TF-IDF of two results, PageRank, HIST algorithm, etc.

[00322] Because now timestamps are associated with the text, and media-based search (e.g., search by matching an voice recording from the user in the audio track or by matching a picture uploaded by user in the video stream) is allowed in conjunction with text-based search, we propose to incorporate information related to those factors into search result ranking.

[00323] In one embodiment of the disclosed subject matter, the ranking is based on score, denoted as the ranking score. The final ranking score is a function combining the values of factors in various ways, or the combination of those ways, including but not limited to summation, subtraction, multiplication, division, exponent, logarithm, sigmoid, sine, cosine, softmax, etc. FMethods used in existing ranking algorithms may also be used, solely or as part of (including joint use) the ranking function. In one aspect, the ranking methods reported in prior arts for search (eg. pagerank, inverse Term Frequency, TF - IDF, etc) can be used to generate a primitive ranking score, which can be used as part of input for calculating the final ranking score.

[00324] It should be appreciated that the ranking function does not necessarily have to be expressed analytically, and it could be a numerical transformation obtained or stored in many ways, including but not limited to, weighted sum of those factors, artificial neural networks (including neural networks for deep learning), support vector machines with or without kernel functions, or ensembled (e.g., via boosting or bagging, specialized methods, such as random forest for decision trees) versions of them or combinations of them. The function can be one transform, or a plurality of combination thereof. The computing of the final ranking score comprises of existing ranking algorithms (e.g., existing algorithms can be used to generate the primitive ranking score) and our new methods (partly based on the power of searching on time-associated media).

[00325] The flowchart of two embodiments are given below.

[00326] Figure 18.

[00327] 2.3.1 Additional factors to consider when ranking

[00328] The temporal distances between terms in a query (e.g., all words in a multi-word search, or a search querying one word in audio track and the other word in video track, respectively, of the same media object) can be included in calculating the ranking score. The temporal distance can be defined as a function of many factors

carrying temporal information. In one aspect, the temporal distance can be defined as the time difference between two timestamps, or the natural exponent of the time difference between two timestamps. Different distance measures can be used when calculating the time difference. In one embodiment, the ranking score is reciprocally propositional to the variance of all timestamps of elements on temporal distance which is defined as the variance of timestamp differences. For example, the query is "happy lab search" and that they have to be no more than 5 seconds apart is the time constraint. Let 3.6 seconds, 4.0 seconds, and 5.0 seconds be the timestamps of the 3 words found as one result in the media. The average timestamps of them are $(3.6+4.0+5.0)/3=4.2$ seconds. Then the variance of them are $[(3.6-4.2)^2+(4.0-4.2)^2+(5.0-4.2)^2] = 1.04$. Then the primitive ranking score calculated using various ranking methods can be divided by 1.04 to take the temporal distances into consideration. In one aspect, the temporal distance can serve as an input for calculating the primitive ranking score; in another aspect, the temporal distance can serve as an input for calculating the final ranking score directly.

[00329] In another embodiment, the ranking score is weighted by the sigmoid of temporal distance, which is defined by the geometric average of all timestamps to their mean. The temporal distance can reflect other factors, too. In another aspect, the temporal distance between two matching words will be doubled if they appear in two different lines or sentences of the transcript (e.g., we may define a penalty term for having words in different lines or sentences). In this case the line is defined as the temporal distance greater than an interval (e.g., $> 2s$). It takes into account the fact that most people usually pay attention to words that are temporally close to each other when watching videos. In yet another embodiment, the temporal distance between two query words are amplified by their distance on the parsing tree, where the parsing tree distance is defined as the number of branches (a branch is also known as an edge in the graph theory) along the shortest path between the two words on the parsing tree.

[00330] In another aspect, the temporal distance is signed with positive and negative. For example, in a string "happy lab" the temporal distance from "happy" to "lab" (denoted as positive) could have different signs than the temporal distance from "lab" to "happy" (denoted as negative). Similarly, when a plurality of words are inputted by the user as the query, signed temporal distances can be calculated between each term pair. The (term1, term2) pair will have the opposite sign from the (term2,

teiml) pair, if the first the term of the pair is defined as the leading term. It should be appreciated that the absolute value may also be used in addition to the signed values for calculate the ranking scores.

[00331] We also consider the confidence level of each match in the media search of the disclosed subject matter and build an overall confidence score based on the said confidence levels. The confidence level is a general term to describe the likelihood that one query element (or a combination of elements; e.g., a keyword) matches one element (or a combination of elements) in the search target, e.g., how close a word in the user-inputted query matches a word in the transcript or how confident the system is about recognizing an actor in a scene. The confidence level can depend on many factors, such as vectorized word distances for word-to-word matches, object matching score for figure-to-video-frame matches. In one embodiment, for word-to-word distance, the confidence level is the vector distance, including but not limited to, cosine of the angle and the Euclidean distance, between the two vectors representing the two words. The vector for a word can be obtained using methods such as Google Word2Vec or Stanford Glove. For recognizing objects in images or video frames, methods such as HOG or ACF can be used to compute a score about how good or reliable the match is.

[00332] Not only can the confidence level itself be a factor in calculating the final ranking score, but also it can be used to affect other factors previously described in the disclosed subject matter, including the temporal distance. In one embodiment, the confidence levels of all matches in one result are passed into a function, and the output of the function, denoted as the overall confidence score of this result, will be used as one input for the function to calculate the final ranking score. This process can be represented by the flowchart shown below.

[00333] 1. user input query

[00334] ↓

[00335] 2. find matches and calculate a plurality of confidence levels

[00336] ↓

[00337] 3. calculate final ranking score using confidence levels and other factors.

[00338] For example, a match contains 3 words, "happy lab search" in one sliding window for the query "happy experimental find", where the confidence levels

are 100% for "happy"->"happy", 80% for "lab"-> "experimental", and 90% for "search"-> "find". The overall confidence score may be defined as the average of them, thus, 90% for this 3-word match. Furthermore, the confidence levels can be phrase-based or N-gram based (eg. with "happy lab" as one term), rather than unigram-based. In one aspect, the overall confidence score can serve as an input for calculating the primitive ranking score; in another aspect, the overall confidence score can serve as a input for calculating the temporal distance; in yet another aspect, the overall confidence score can serve as one of the inputs for calculating the final ranking score.

[00339] In another embodiment, the confidence levels of all matches will be passed into the function to calculate final ranking score directly. Given a match, its confidence level is a function of factors of the matches and terms in the matches. Those factors include but are not limited to media type, length of media, body of the match (e.g., on producer-provided subtitle, speech-to-text converted caption, audio or video), orders of terms in a query in the match (e.g., finding "happy lab" where term "happy" is before term "lab"), source of the media (e.g., if an element involves searching a word in the user annotation, the confidence level for such a match is linked to the credibility and history of the annotator). In one embodiment, the confidence levels of matches can be weighted to calculate either final ranking score or overall confidence level, based on the types of media and the length of media. For example, the algorithm can have more confidence for a text match found in the lyrics of a short music TV (MTV, Karaoke, etc.) than an object recognition match found in the a user uploaded lengthy and blurry surveillance video. In yet another embodiment, the user could set the search result to rely on video frame matches more than on text matches, i.e., video matches are given higher confidence weight than text matches. In yet another embodiment, the user wants to find words matching a speech-to-text (STT) converted caption. The confidence level for matching a word in query (inputted as text) and a word obtained via transcribing the media is the multiplication/product of the vector distance between the vectors representing the two words while the confidence level of the speech-to-text conversion for the second word. . For example, let's say we want to find the confidence level of matching the word "sun" in the query and the word "solar" which is transcribed from a video clip. Let the vector distance between "sun" and "solar" be 0.9 and the confidence level of correctly transcribing "solar" is 0.85, then the confidence level for matching the two words is $0.9 \times 0.85 = 0.765$. An extreme case for confidence level is when the

search is on professional annotation of the media, such as the X-Ray (TM) labels made by Amazon.com when streaming timestamp Videos. For professional annotation, the confidence level should be significantly high, e.g., set as 100%.

[00340] Order or time of occurrence is also a factor that should be considered when ranking results and for calculating the final ranking score. Through this disclosed subject matter, especially the TymeTravel syntax, users are allowed to specify temporal constraints in the query (e.g., happy@25%+, meaning that the word "happy" appearing in the first quarter of the media), while the search results may match the temporal constraints to different extent which can be ranked. In addition to allowing certain flexibility of discrepancies between results and queries, the present disclosed subject matter also uses the discrepancies between results and queries as a factor for calculating final ranking score, which can be modeled as a function. Different temporal discrepancies (whether and how the temporal constraints are satisfied by search results) can be used to calculate the final ranking score, and different types of temporal discrepancies should affect the final ranking score to different extent. Users can specify (e.g., by ordering) the tolerance to different temporal discrepancies. In one embodiment, the final ranking score will be obtained by multiplying the primitive ranking score with the sigmoid of number of temporal discrepancies in the result. In another embodiment, different types of discrepancies have different weights in calculating the final ranking score. For discrepancies on global temporal constraints, e.g., happy@25+, the weight is 2 while the weight for discrepancies on local temporal constraint, e.g., time(happy) - time(lab) < 40, is 1. In this setting, if two results are identical but one has a discrepancy on global temporal constraint (weight=2) while the other has a discrepancy on local temporal constraint (weight=1), the former result with discrepancies on global temporal constraints will be ranked lower than the later result, as the global temporal constraint discrepancy results in a heavier penalty term due to its heavier weight. In this case the mathematical expression is: final ranking score = primitive ranking score - weight* discrepancy of global temporal constraint - weight* discrepancy of local temporal constraint.

[00341] A flowchart describing this process is shown below:

[00342] 1. calculate primitive ranking score

[00343] ↓

[00344] 2. assign weights for a plurality of time constraints

[00345] ↓

[00346] 3. calculate a plurality of temporal discrepancies

[00347] ↓

[00348] 4. calculate final ranking score using primitive ranking score, weights of time constraints and temporal discrepancies.

[00349] In another embodiment, the order or time of occurrences of keywords can be used to calculate the final ranking score. In one aspect, if there are two occurrences of the word "happy" found at the first quarter of the media, with one at 1/8 and the other one at 1/6 temporal distance from the beginning timestamp of the media, the user could configure the search engine described in the disclosed subject matter to calculate the final ranking score with the temporal distance being considered. For instance, if the user prefer the earlier occurrence of the query keyword, he may check "the earlier the better" checkbox being supplied by the software GUI, and in this case the occurrence at 1/6 will be ranked higher. Exemplified equations are shown below:

[00350] "The earlier the better" for query keywords to appear:

[00351] Final ranking score = primitive ranking score / temporal distance

[00352] "The later the better" for query keywords to appear:

[00353] Final ranking score = primitive ranking score * temporal distance

[00354] A flowchart of aforementioned embodiment is given as follows:

[00355] 1. input query terms

[00356] 4

[00357] 2. specify user preference on temporal distance of query terms

[00358] 4

[00359] 3. calculate temporal distances of search match results

[00360] 4

[00361] 4. Calculate primitive ranking score

[00362] 4

[00363] 5. Calculate final ranking score based on primitive ranking score, temporal distances of search match results and user preference on temporal distance of query terms

[00364] 2.3.2 Ranking update based on user feedback

[00365] The ranking results, parameters (including weights) for ranking factors, and equations for calculating final ranking score, can be refined based on user feedback, machine learning, crowd sourcing and/or user-interaction with the system. To customize and reorder the ranking, the updating process can be scoped for different scenarios. Possible scenarios comprise a genre of media work (e.g., in speeches, the text is more important than video) , a piece of media work (e.g., one chapter of an audio book or one scene of a movie), different fields of a media work (e.g., two queries using the same terms to search in the same movie but one on text and the other on audio can have different rankings), a user, a group of users (e.g., friend circle on social media), etc., or a combination thereof (e.g., customized ranking for a group of users on all dramas produced during 1990s). The feedback can be from one user, a group of user, a plurality of databases, and recent trends, etc.

[00366] User feedback can be extracted in multiple ways and various types of information can be collected. In one aspect, the most common way is to log which result the user clicks. In one embodiment, when the user clicks a search result, he/she votes for that result. The most voted result should pop up. Indeed, after a long run, the ranking will be tuned more and more toward user votes. For example, if sufficient amount of users click the second result instead of the top result, the second result will be popped to top. More explicit way of soliciting user feedback is asking questions (e.g., "should this result be moved up or down?") or asking the user to manually reorder the search result. In another embodiment, the order that a user clicks the results and the temporal interval between clicks are logged. The earlier a link is clicked, the higher it should be ranked. However, if the duration between two consecutive clicks is too short, that indicates the former result may not be the correct one and the user returned to the search result to find the next candidate, which can be taken into account for updating the ranking.

[00367] Once we collect the user feedback, there are many algorithms that can be used, in off-line and/or on-line fashions. Methods in learning to rank or machine-

learned ranking (MLR) can be applied. In one aspect, the software will apply methods comprising pointwise approaches (e.g., RankBoost, McRank), listwise approaches (e.g., AdaRank, ListMLE), pairwise approaches (e.g., GBlend, LambdaMART), or combination thereof (e.g., IntervalRank using pairwise and listwise approaches). In one embodiment, the ranking update problem can be modeled as a regression problem in the context of machine learning, e.g., finding a mapping from factors mentioned above to user votes. In another embodiment, the ranking update problem can be solved through a classification problem in the context of machine learning, e.g., classifying whether the rank of each result is over-ranked or under-ranked. It should be appreciated that the classifier can be binary or multi-class. It should be further appreciated that the machine learning algorithm can be supervised learning or unsupervised learning.

[00368] In one embodiment, the results of very low ranking scores may not be returned/displayed to users.

[00369] A representative flowchart including ranking update based on user feedback is shown below:

[00370] Figure 19

[00371] 2.4 Transcribing text

[00372] Time-associated text information, such as transcripts, legends, captions, lyrics, and annotations may or may not be already available for search. In the cases that the Time-associated text information is unavailable, speech-to-text conversion can be employed to extract the transcript. The speech-to-text conversion or automatic speech recognition (ASR) method comprises an algorithm/model, including but not limited to, Hidden Markov Models (HMM), Dynamic Time Warping (DTW), Neural Networks, Deep Learning Models. The basic pipeline including text generation from audio is shown below.

[00373] Note that we mean a broader meaning of transcribing: turning any text embedded in the media into text, such as turning characters on the street in a movie to text, turning the famous "A long time ago in galaxy" opening scene of Star Wars into text, or turning a background music into the text "Beethoven Symphony 9".

[00374] Figure 20: Flowchart of search with Time-associated text including transcribing

[00375] 2.5 Translation

[00376] Enabling translation of the either the queries and/or the reference (e.g., a transcript) has several benefits, e.g., the user does not have to search in the native language of the media entity, or they can watch the media entity with captions in a different language.

[00377] Both the query and the native transcripts can be translated. However, translating the native transcripts might be more accurate as it can take advantage of the context. Machine translation approaches, including rule-based machine translation (RBMT) or statistical machine translation (SMT) can be used. RBMT systems include direct systems that uses dictionaries to map input to output, transfer RBMT systems that employ morphological and syntactical analysis and interlingual RBMT systems that employ abstract meanings of words. SMT systems can work at word, phrase or sentence level. The translation can be done hierarchically according to syntax. Hybrid systems may use both RBMT and SMT systems.

[00378] The translation can be done at different levels, including word, phrase, clause, sentence and paragraph levels, respectively. Because the order of words of the same sentence may differ in different languages, translating at sentence level may have an added advantage. In one aspect, for each sentence in the new caption, it will be displayed from the time at the beginning of the sentence in native language and to the time at the end of the sentence in native language. It is also possible to chop the translated sentence in many parts and displayed sequentially during that time interval. To determine when to display which word, we can employ automated audio and transcript alignment algorithms, such as finite state transducer (FST) or time-encoded transfer. Therefore, the translated transcripts can be obtained for searching.

[00379] Should the query and the media entity not have perfect/identical match in the time-associated text information such as transcripts, which may be a common case because of the ambiguity of natural languages, more advanced matching algorithms will be used as to be discussed later in this disclosed subject matter disclosure. The pipeline of searching with build-in translation is illustrated below.

[00380] Figure 21: Flowchart of search with Time-associated text information with build-in translation

[00381] 2.6 Miscellaneous features

[00382] Multi-channel sound: Many media entities contain more than one audio channels, for purposes such as stereo effects. We propose several approaches to solve this problem. In one aspect, the software will search the query in the transcript(s) of one or more channel(s), and show the result individually or collectively. An alternative approach is to first form a consensus from any, some, or all channels, and then run the search discussed above on the consensus of them. The consensus can be formed in many ways, in audio, text, or jointly using the information from more than one form. Using audio signal, the consensus can be form in many ways, including but not limited to, averaging the signals from different channels, detecting the phase shift between channels and then align the signals, etc. In text domain, the problem of text alignment is well studied, methods including the IBM method, Hidden Markov Model (HMM), Competitive Linking Algorithm, Dynamic Time Warpping (DTW), Giza++, etc.

[00383] Safeguard: In one embodiment, the software can be used to screen for inappropriate words or sex-related words in the video. Inappropriate text or media contents will be detected using the matching algorithms mentioned above. Then a "clean version" will be generated with those inappropriate content removed. For example, place a high-pitch noise at timestamps where bad words are found. As another approach, make a new copy with the time durations containing bad words removed.

[00384] Error checking: Ideally, Time-associated text information (e.g., a transcript) and pre-existing (e.g., closed caption provided by the producer) transcripts should match. If not, computer vision/audio analysis will try to align the media data with transcripts within nearby frames to understand and correct. Text and media data can be aligned using many methods. For example, text and speech can be aligned using, including but not limited to, comparing in audio domain by converting text to speech (such as SyncTS system), universal phone models, finite state machine methods, time-encoded method, dynamic programming, etc. If consensus between data of some domains (e.g., text and audio) cannot be reached, alignment with more domains (images, other data sources) can be the tier breaker.

[00385] Music/audio management: In software/services such as iTunes, Google Music, Amazon Music and karaoke software, the music may also be arranged for searching based on the matching query with lyrics. As such, songs with the same words, phrases or sentences can be grouped together. For instance, when a query "hotel" is inputted into the system, all songs with "hotel" in the lyrics will all show up under the

search and can be thus categorized together. Furthermore, search will enable synonyms matching or fuzzy search if desirable. For instance, "inn" and "hotel" may be matched together under synonyms. This way, the "theme", "context" of the music or videos can be grouped together.

[00386] Presentations the results: Search results can be presented to the users in multiple ways, including but not limited to, the original media entities, the most interesting (ranked by landmarks, which will also be discussed in later sections of this disclosure) segments (linear or nonlinear) of media, a montage of certain segments of media, text results of any form and lengths, or the combination of them. Ranks and scores of multiple matches will also be presented to the user, numerically or graphically, or the combination. The confidence level and ranking scores may also be presented to the user as the search results.

[00387] 3. The Videomark Technology: conveniently manage and navigate through contents

[00388] In the disclosed subject matter, we describe a technology that can help users to conveniently manage and navigate through their media contents. Currently, the media management and navigation system only offers limited functionalities: for instance, the currently available management systems are only based on files, rather than time-associated information. For instance, using conventional system a playlist can be created which contains a sequence of videos to be played. In this case, the playlist is based on different video files, ignoring the time-associated information.

[00389] With the "Videomark" technology described in the disclosed subject matter, the user can navigate and manage the contents within the same video based on time-associated information and/or audio-associated information. For instance, the software can create a new type of data that stores shortcuts

[00390] mapping relationship, which maps from an index key to a timestamp in the media. The index key can be a key for text, images, audio, videos, other media or a combination thereof. In one embodiment, the index key is similar to elements in a "table of contents" that is used for text contents, such as a PhD thesis. In one embodiment of the disclosed subject matter, the index key will map a text key to a timestamp in a video. A plurality of index fields can form a table of content for a video. For instance, a table of content for the TV series "Friends" episode 1 can be formed, as follows:

[00391] Figure 22. Index keys presented to user based on the Videomark technology

[00392] In the table of content shown above, each text key is mapping to one unique timestamp or a plurality of timestamps. When user click one text key, such as "Rachel decided to stay with Monica", the software will play from the corresponding timestamp associated with the event that shows that Rachel decided to stay with Monica.

[00393] It should be appreciated that the index keys can be generated either manually by the user or automatically by the software. It should be further appreciated that the index keys can be generated by input a search query such as "Ross and Rachel" into the software and the software will automatically generate the index key.

[00394] It should be further appreciated the video do not necessarily have to be segmented into video segments. Instead, the index keys can just be the beginning and end timestamps for each of the video duration desirable, without the video segmentation process.

[00395] In another embodiment, the table of content has fields that maps to timestamps in a plurality of videos. For instance, in the table of content shown below, the key "8. Rachel is going out with Paulo" maps to a timestamp in another episode (stored as a separate video file), when Rachel is going out with Paulo. Consequently, user can conveniently manage and navigate through the content based on information embedded in the media, and jump to the corresponding timestamp within the right video file for media playing easily.

[00396] Figure 23. Index keys presented to user based on the Videomark technology. Multiple files are indexed.

[00397] In yet another embodiment, the table of contents presented to the user contain text keys with a hierarchical structure. For instance, an exemplified table of content is shown below, where 2 level of text keys are presented. The key "4. 1 Rachel checked out Monica's apartment" and "4.2 Monica mentioned that she got the apartment from her grandma" are keys hierarchically under the key "4. Rachel decided to stay with Monica". By clicking on " 4.1 Rachel checked out Monica's apartment", the software will jump to the corresponding timestamp to play from the segment that Rachel checked out Monica's apartment.

[00398] Figure 24. Index keys presented to user based on the Videomark technology. Multiple files are indexed. Hierarchical structure is included.

[00399] In yet another embodiment, the index key can contain images as an representation for displaying to user. An example is shown below. In this case, the first key field contain an image, which may reminds the user of the plot. The image key field map to a timestamp that is associated with the image. In one aspect of the embodiment, the timestamp is a timestamp which is temporally proximal to the timestamp when the image is shown in the video. In another aspect of the embodiment, the image is user specified and the timestamp is also user-specified.

[00400] Figure 25. Index keys presented to user based on the Videomark technology. Multiple files are indexed. Hierarchical structure is included. The index key can include images, animations or video clips.

[00401] In a further embodiment, the index key is a combination of image and text, as shown below:

[00402] Figure 26. Index keys presented to user based on the Videomark technology. Multiple files are indexed. Hierarchical structure is included. The index key can include images, animations or video clips.

[00403] It should be appreciated that the index keys can also be in formats other than text or images, when they are presented to the user. The index keys can be in the forms of video, animation such as GIF format, or audio clips. It should be further appreciated that the index key can be a combination of video, animation, audio clips, text and images. In one aspect of the embodiment, the index key can be a GIF animation, which help the user to understand the context. In another aspect of the embodiment, the index key can be a video clip. When the user place the mouse cursor on the video clip, the video clip will play (muted or with sound), so the user can preview the index key to determine whether it is what he/she want.

[00404] It should be further appreciated that an index key can map to a plurality of timestamps. In one embodiment, when the user click on one such index key, the user will be presented with a selection from a plurality of timestamps in a form of menu. The user can therefore select from the selection menu selections for the desirable timestamp. In another aspect of the embodiment, when the user click on an higher level index key such as the "Friends meet each other in the Central Perk coffee shop ", the

user may be redirected to the lower level table of content enumerating multiple timestamps associated with the index key, as shown below. The user can therefore make selection for the desirable timestamp. In another aspect, the different level of index key list can be expanded or collapsed, as needed. An example given below. When the user input query "friends meet Central Perk", the software may return

[00405] Figure 27. Index keys presented to user based on the Videomark technology. when the user click on one such index key, the user will be presented with a selection from a plurality of timestamps in a form of menu.

[00406] In another embodiment, an index key maps to a timestamp in a plurality of audio files, such as voice memos, audio recordings, podcasts, music files or karaoke soundtracks. In one embodiment, the disclosed subject matter allow the user to manage and navigate within the voice memos, podcasts and audio recordings. In another embodiment, the disclosed subject matter allow user to manage and navigate within the music files. As such, the user can jump to the desirable timestamps within songs. In yet another embodiment, the disclosed subject matter embodies a karaoke player software that allows the user to navigate in the karaoke soundtracks to the timestamps where certain lines of lyrics are said, thereby skipping the irrelevant parts of the soundtracks.

[00407] It should be appreciated educational contents such as video lectures or audio lectures will benefit greatly from the disclosed subject matter by using the aforementioned methods. The index keys can map to the timestamps in video lectures or audio lectures. In one embodiment, the students can also be video or audio recordings as their homework to a problem; the teacher can navigate through the answers in the form of video/audio easily using the method in the disclosed subject matter.

[00408] In yet another embodiment, some of the index keys can map to a text block in a text file such as pdf file. In another embodiment, the index key can also map to a website. In another embodiment, the index key can also map to a hashtag or tweet. In another embodiment, the index key can also map to an image. An example is shown below. In this case, when user click on "1. Law of reflection", the user will be redirected to the paragraph in an optics textbook (eg. as a pdf file) which discusses the law of reflection. When the user click on "2. Demonstration of Law of reflection" or "4. Demonstration of Law of refraction", the user will be redirected to the corresponding timestamps in the video, as previously described. When the user click on "5. Discussion

of geometric optics", the user will be directed a website where the geometric optics are discussed. When the user click on "6. Further discussion of geometric optics", the user will be redirected to the corresponding timestamp in the audio clip. When the user click on "Summary graph of geometric optics", the user will be shown the image of a summary graph about optics.

[00409] Figure 28. Index keys presented to user based on the Videomark technology. Some of the index keys can map to a text block in a text file

[00410] It should be appreciated that the software may provide a graphic user interface that present the user the relevant information, so that the user does not have to jump between more than 1 software for using text, video and audio contents. A representative user interface is illustrated below. The panel for index keys will present the index keys such as table of contents, as previously discussed. The Panel for Presentation of Results will present the videos (play from the timestamp of choice) or present the text content (eg. particular paragraphs in the textbook) to user. As such, in one embodiment the software has an embedded media player for playing audio/video, and a text editor/reader for present the text content. It should be appreciated that a website can also be presented in the panel for presentation of results.

[00411] Figure 29. GUI presented to user based on the Videomark technology.

[00412] In another embodiment, the graphic user interface of the software is shown below. There is an additional panel for showing advertisements, recommendation or social media components. For instance, the software can show user an recommendation "since you like ABCD, you might want to try EFGH". Advertisement may also be shown here, in text, photo, animation, video or combination thereof. Shopping choices can also be presented here, partly based on user's choice of index key and/or user history. Social media can also be integrated or interfaced with the software: for instance, the user can share part of the content to his/her friends via social media (eg. "i just watched XYZ")

[00413] Figure 30. GUI presented to user based on the Videomark technology. Additional panel for showing advertisements, recommendation or social media components is included.

[00414] A representative flowchart for generating and presenting index keys to user is shown in Figure 31.

[00415] In another embodiment, the software will generate the list of index keys based on user input or search query. In one aspect of the embodiment, the index keys may be the exact search results generated by the software discussed in section 2 of the disclosed subject matter (with corresponding timestamps); in another aspect of the embodiment, the index keys may be entities based on search results with user input. The software will integrated the aforementioned search functionality with the media management functionality. Thus, the software can help the user to create a personalized table of content based on the search query. It should be appreciated that the list of index keys can be either generated in a just-in-time fashion, or to be previously calculated, indexed and stored to speed up the query speed. A representative flowchart is shown below:

[00416] Figure 32. flowchart for processes including user query, index key generations and presentation of relevant results

[00417] For instance, with the show "Friends" the user may search for the term ""Rachel" AND "Ross"" as the query, the software will match the search query with the videos of "Friends", based on transcript. The results will be ranked and presented to the user as a list of index keys, in an orderly fashion (eg. starting from season 1 episode 1 to Season 10 last episode, sequentially). A result as shown below may be presented to the user, as follows.

[00418] Figure 33.

[00419] As user click on each of the index keys, the software will play the video from the corresponding timestamp, so that the user can easily navigate through the show and watch the show in a convenient way. It should be appreciated that movies, TV shows, video lectures, audio clips, voice memos, games, radio recordings, podcasts can all benefit the disclosed subject matter, in a similar way.

[00420] In one embodiment, the software will create a spoiler-free version of list of index keys. The spoiler-free version will omit some detailed plots so that the user will not have spoiler of the media to be watched. For instance, with the show "Friends" the user may search for the term ""Rachel" AND "Ross"" as the query, the software will generate the following spoiler-free version of list of index keys. In another aspect of the embodiment, the user can click on an icon called "switch to full version list", and the user will be redirected to the version of list of index keys containing spoilers, as

shown previously. It should be appreciated that warning to user such as "Are you sure you want to switch to the full version of list containing spoilers?" can be displayed to user to solicit user confirmation. In one aspect of the embodiment, the spoiler-free version can be the default choice, to avoid revealing and spoilers. In another aspect, the software will show the user the full version by default if the watch history is tracked and suggested that the user has watched the show,

[00421] Figure 34. spoiler-free version of list of index keys

[00422] In yet another embodiment, the software will generate a short video based on the user query and list of index keys. In one aspect, the short video is similar to a summary video or trailer. For instance, with the show "Friends" the user may search for the term ""Rachel" AND "Ross"" as the query, the software will generate a short video containing short video segments associated with the index keys. In one aspect of the embodiment, each video segment associated with the index key may be a video starting from the timestamp of the index key and lasting for a duration of time of choice (e.g. 5-12 minutes after the timestamp). In another aspect of the embodiment, each video segment associated with the index key may have a variable duration, and the duration is determined by other segmentation techniques based on artificial intelligence. As such, the user can view the shorter video compilation instead of the whole TV shows. Upon different user query, different video compilation can be generated. It should be appreciated for each index key, a plurality of video segments (greater than 1) may be generated, based on user-preference or default settings. A representative flowchart of this process is shown below:

[00423] 1. User input query

[00424] 4

[00425] 2. Match query with Time-associated text information and rank results

[00426] 4

[00427] 3. Generate video segments and a list of index keys associated with video segments based on search results

[00428] 4

[00429] 4. Present the list of index keys to user

[00430] 4

- [00431] 5. (optional) Collect user feedback
- [00432] 4
- [00433] 6. (optional) Set duration for video segments associated with each index key
- [00434] 4
- [00435] 7. Generate the video compilation based on the index keys
- [00436] Flowchart for generating short video
- [00437] In yet another embodiment, the list of index keys is presented to the user in the form of a glossary. In one aspect of the embodiment, each element of the glossary maps to a plurality of timestamps in media such as videos or audio clips. In another aspect of the embodiment, some elements of the glossary map to text blocks in textbooks, and other element of the glossary maps to a plurality of timestamps in media such as videos or audio clips. In another aspect of the embodiment, some elements of the glossary map to websites or hashtags. It should be appreciated that the glossary can have a plurality of hierarchical levels of index keys. The lists of index keys can be expanded or collapsed as needed, for user visualization. In one aspect of the embodiment, user can enter search query as a new element to be added to the pre-configured glossary. It should be appreciated that the glossary feature described here will benefit both educational application and non-educational application.
- [00438] In another embodiment, the aforementioned methods can be used in medical fields, such as medical data search, recording, management and navigation. For medical records and data, the most common forms of data are timed readings (such as blood pressure at given date), timed text (e.g. physician's summary for a visit at certain date), images (e.g. CT, MRI, ultrasound data at a given date), videos, etc. As such, the methods in the disclosed subject matter can be applied to medical field. In another aspect of the embodiment, a medical training software can be embodied. For instance, the medical training software can navigate the user between video lectures, video for diagnostics, surgical demo videos, textbooks, websites, images, using the methods previously described in the disclosed subject matter.
- [00439] In another embodiment, the aforementioned methods can be applied to management of user-generated videos, such as videos taken on the phone or videos

created in the social media application (e.g. Snapchat etc.). Consequently, when the user query the software, videomark technology will arrange and present the results to the user. For instance, if the query "Mike" is inputted, the software will generate a list of index keys based on search results for "Mike", similar to the examples previously discussed. Thus the user can manage and edit the video contents conveniently. In one embodiment, the list of index keys generated by one user can be shared to other users via social media applications such as Snapchat, Twitter, WeChat or Facebook. In another embodiment, the videos being searched with the queries are media messages containing video (e.g. Snapchat or WeChat) and the list of index keys based on user query is build in the social media application. Consequently, the user can easily manage video messages in the social media application such as Snapchat. In another embodiment, the software will search in the user-defined library, such as videos in local storage, video-containing messages, MMS, videos in cloud storage, or a combination thereof. It should be appreciated that dating social media application can also benefit from the disclosed subject matter.

[00440] In another embodiment, the aforementioned methods can be used to managing adult videos. It should be appreciated that the adult videos can be user generated or be created by adult websites. For instance, when the query "take off" is inputted, the list of index keys will be generated based on the said query. The previously discussed "generating short video" feature can also be applied to adult video contents. The software can automatically generate a short summary video integrating video segments, based on user input/query.

[00441] In yet another embodiment, the media entity can be a game video or recording (e.g., in the game's own format). The videomarks hence can be events related with and specific to the game. For example, in the video game WarCraft III, the videomarks may include all times that a hero dies. When the user click a videomark, the game video or replay will jump to place where a hero dies. The game does not have to a video game. For example, videomarks can be all touchdown moments in one season of NFL. When the user traverses all videomarks, he/she will enjoy all touchdown moments in this season.

[00442] 3.1 Video/Audio Editing

[00443] Currently, the video/audio editing software is not very convenient to use. For instance, for one to segment a short clip from a longer video, one has to manually search on the time axis to define the starting and ending points of the segment. Also, it is sometimes very hard to quickly locate the right recorded event (e.g. if we look for the events when "Edison" is said in footage, it is not very convenient to search for it manually).

[00444] With Videomark technology and search methods previously described, video editing software can be developed with build in search capability. In one embodiment, the software can search for the event based on Time-associated text information. For instance, when the word "Edison" is inputted into the software as the query, the software will search, rank and return the results matching "Edison" to user, when the word "Edison" is said in the video. The search and rank process is similar to the methods previously described in the disclosed subject matter. By clicking on the result, the software will take the user to the timestamp associated with the query (eg. when the word "Edison" is said in the transcript). As such, the user can easily define the starting point or end point of a video segment, without manually searching in the videos and dragging back and forth on the time axis. It should be appreciated that video editing functionalities of computer science can be enabled in the media editing software, such as divide, combine, segment, special effect, slow motion, fast motion, picture in picture, montage, trimming, splicing, cutting, arranging clips across the timeline, color manipulation, titling, visual effects, mixing audio synchronized with the video image sequence. It should be further appreciated that conventional method of manually editing the media across the time axis is also supported by the media editing software described in the disclosed subject matter. A representative flowchart of this embodiment is shown below.

[00445] Figure 35. flowchart including video editing functionalities.

[00446] In another embodiment, the search results are be presented to the user using the "Videomark technology" described in the disclosed subject matter. The user will be able to drag and drop the index keys, or realign the index keys along the timeline/time-axis for media editing, with GUI. A representative flowchart is shown below

[00447] 1. User input query

- [00448] 4
- [00449] 2.Match query with Time-associated text information and rank results
- [00450] 4
- [00451] 3. Generate a list of index keys based on search results
- [00452] 4
- [00453] 4. Present the list of index keys to user
- [00454] 4
- [00455] 5. (optional) Collect user feedback
- [00456] 4
- [00457] 6. User select an index key or a plurality of index keys
- [00458] ↓
- [00459] 7. Show user the corresponding timestamps to which the index keys are mapped to
- [00460] ↓
- [00461] 8. media editing
- [00462] ↓
- [00463] 9. Present edited media to user
- [00464] 4
- [00465] 12. (Optional) Return to step 1
- [00466] In yet another embodiment, reserved words can be defined for automatic or semi-automatic segmentation of media. For instance, words such as "3, 2, 1, action" and "cut" can be defined as starting point and end point, respectively. As such, the occurrences when "3, 2, 1, action" and the associated timestamps will be automatically defined as the starting points of the segment, and the occurrences when "cut" and the associated timestamps will be automatically defined as the end points of the segment. Consequently, various comment/production terminologies can be incorporated into the software as reserved words.

[00467] It should be further appreciated that the aforementioned methods can be applied to editing music and podcasts, including music soundtrack and music video.

[00468] It should be appreciated that the media editing software embodied by the disclosed subject matter can run on any computing devices, such as desktops, laptops, smartphones, tablet computers, smart watches, smart wearable devices. It should be further appreciated that camcorder, cameras, webcams, voice recorder with computing power can also run the media editing software. It should be appreciated that the media editing software can run locally, on the cloud or on a local/cloud hybrid environment. It should be appreciated that the media editing software can be integrated with video streaming websites/providers and social media software applications.

[00469] 3.2 Video selfie and video based social media

[00470] In one embodiment, the software can enable video selfie (eg. take a video of oneself) capturing, processing, management and navigation.

[00471] With Videomark technology, search methods and video editing methods previously described, a video selfie software can be developed with build-in search capability. In one embodiment, the software can search for the event based on Time-associated text information. For instance, when the word "Edison" is inputted into the software as the query, the software will search, rank and return the results matching "Edison" to user, when the word "Edison" is said in the video. The search and rank process is similar to the methods previously described in the disclosed subject matter. By clicking on the result, the software will take the user to the timestamp associated with the query (eg. when the word "Edison" is said in the transcript). As such, the user can easily define the starting point or end point of a video segment, without manually searching in the videos and dragging back and forth on the time axis. It should be appreciated that video editing functionalities of computer science can be enabled in the media editing software, such as divide, combine, segment, special effect, slow motion, fast motion, picture in picture, montage, trimming, splicing, cutting, arranging clips across the time axis, color manipulation, titling, visual effects, mixing audio synchronized with the video image sequence. It should be further appreciated that conventional way of manually editing the media across the timeline is also supported by the media editing software described in the disclosed subject matter.

[00472] In another embodiment, the search results are be presented to the user using the "Videomark technology" previously described in the disclosed subject matter. The user will be able to drag and drop the index keys, or realign the index keys along the timeline/time-axis for media editing of selfie videos.

[00473] In one embodiment, the video selfie software can facilitate batch processing capability to handle videos. Because a video contains a collection of image frames, the processing of video is not as easy as the processing of individual selfie images. In one aspect of the embodiment, the processing can be done on one representative frame, and the settings will be applied to other image frames automatically.

[00474] In one embodiment, the software comprises object recognition, object tracking and face recognition algorithms. Moreover, region of interests (ROI) can also be user defined (for instance, the user may draw a square via the GUI to define the ROI manually). Furthermore, the features such as human or dog can also be identified using segmentation techniques. The software will detect the faces and track them across the frames. Consequently, similar settings can be applied to processing of each face across the frames.

[00475] Faces can be identified by face recognition algorithms. Algorithms for facial recognition include but are not limited to, eigenfaces, fisherfaces, local binary patterns histogram. Many open source software libraries has functions for facial recognition off-the-shelf, including OpenCV.

[00476] In one embodiment, if different number of faces shows up in different frames (eg. frame 1-100 has Jim, Mike, Lily, Frame 101-130 has Mike and Lily, Frame 131-200 has Mike, Lily and Lucy), the software may show 3 selected frames to the user (eg. a frame selected from frame 1-100, a frame selected from frame 101-130, a frame selected from frame 131-200, respectively). In another aspect of the embodiment, a plurality of frames will be synthesized based on the characteristics of the all frames. For example, a weighted algorithm can be implemented to generate a "representative" image for frame 1-100, frame 101-130, and frame 131-200, respectively. One possible method is to perform a special region based ensemble averaging (averaging for each face with affine transformation). Another method is to digitally synthesize an image

with the faces of Jim, Mike, Lucy and Lily with their faces having typical intensity values. Thus, the user only has to edit one image frame instead of a plurality of frames.

[00477] The software will track the features and regions of interests across different frames. In one aspect of this embodiment, faces will be tracked. The software can use Object recognition algorithms comprises: Edge detection, Primal sketch, Recognition by parts, Appearance-based methods, Edge matching, Divide-and-Conquer search, Greyscale matching, Gradient matching, Histograms of receptive field responses, Large modelbases, Feature-based methods, Interpretation trees, Hypothesize and test, Pose consistency, Pose clustering, Invariance, Geometric hashing, Scale-invariant feature transform (SIFT), Speeded Up Robust Features (SURF), Bag of words representations, 3D cues, Artificial neural networks and Deep Learning, Context, Explicit and implicit 3D object models, Fast indexing, Global scene representations, Gradient histograms, Intraclass transfer learning, Leveraging internet data, Reflectance

[00478] Shading, Template matching, Texture, Topic models, Unsupervised learning, supervised learning, Window-based detection, Deformable Part Model, Bingham distribution, etc.

[00479] In one embodiment, the software to map the regions of interest such as the faces in other image frames to the representative frame. Given that the user has already specified global and/or local operation/processing of the image, where global operation applies to the whole image frame while local operation applies only to a plurality of pixel neighborhoods, the software will map the ROIs in other frames to the ROI in the representative frame. For instance, a one-to-one correspondence will be established between different regions within the face of Lily in the representative image frame and the different regions within the face of Lily in other frames. Consequently, the global operations and local operation that user specified on the representative frame can be applied to other frames. It should be appreciated that the operation on other frames may not be identical that on the representative frames, to accommodate the differences between individual frames. The flowchart is shown below

[00480] 1. edit representative frame with global and/or local operations/processing as the reference editing method

[00481] ↓

[00482] 2. Map the region of interests in the representative frame to the ROI in other frames; establish correspondences between plurality of regions in the he representative frame to plurality of regions in the ROI in other frames

[00483] ↓

[00484] 3. Calculate the transformed global and local operations on individual frames based on the mapping and the reference editing method

[00485] ↓

[00486] 4. Present the edited video or image frames to user

[00487] ↓

[00488] 5. (optional) Collect user feedback

[00489] 4

[00490] 6. (Optional) Return to step 1

[00491] Flowchart of processing pipeline using mapping

[00492] In another embodiment, the ROI such as the faces are divided into a plurality of regions. The software calculates the histograms with each regions in the ROI of the processed representative image frame (after the global and/or local operations applied). Subsequently, the software will also calculate the he histograms within each regions in the ROI of the other image frames. The software will process the other image frames so that the histograms of regions in the ROI will be more similar to that of the processed representative image frame. It should be appreciated that the values and/or distribution of the histogram will be used for the calculation of operations for other image frames.

[00493] In another embodiment, the user may process one image with global and local operation. In one example, the user is process a selfie photo so that she looks better on that image. The software may apply the aforementioned methods to batch process other photos of the user, so that she looks better on other photos without requiring her to go through editing each photo individually.

[00494] In another embodiment, the software will record user history of his/her photo/video editing activities and automatically generate a template for editing photo with individualized settings. Machine learning algorithms may be applied to calculate

the setting based on user history. In yet another embodiment, the software will rely on crowd sourcing and record the editing activities of a plurality of users. The software will use machine learning algorithm such as deep learning to generate suggested template settings to the user based on machine learning, for photo/video editing. It should be appreciated that the software may use both local user history and crowd sourcing/ deep learning results to generate the optimized settings.

[00495] 4. Time-associated text search and media navigation control:

[00496] When playing media entities such as videos and audios, especially those recording speeches, teaching, presentation or movies, a user often need to jump to a particular location where a particular word, phrase or sentence is mentioned in the media stream. However, currently, users cannot do such thing automatically but have to manually move forward/backward to search using their eyes, ears and memories. For instance, the user may use a time cursor (aka. seekbar, progress bar) to advance to rewind the video to the desirable timestamp, when the said queries (e.g., a particular word, phrase or sentence) occurs. Often time, it is inconvenient and may takes several iterations, especially when the user does not know where the query (e.g., a particular word, phrase or sentence) happens in the video.

[00497] For a new lengthy media entity that is unfamiliar to the user, it is even more difficult to locate the relevant segments where the query word is mentioned. For example, if the user knows that the word "toothbrush" is mentioned in a 1-hour video, but the user does not know where the word "toothbrush" was mentioned, it is hard to locate the occurrences of the said word without watching the video or find it through trials and errors.

[00498] In the disclosed subject matter we introduce a solution by searching for queries in the media using the transcript, or any Time-associated text content of the said media, to provide the user the capability to jump to any locations when the said queries was mentioned. The query may be a plurality of particular words, phrases or sentences, etc, in any language. The user may type, say or use other methods (such as those mentioned in Section 2.1 earlier) to input the said queries to the computer. For example, when the user says "Pythagorean theorem" as a query in a video of geometry class recording video, the user will jump to the particular timestamp when the teacher mentions the phrase "Pythagorean theorem". The video segment may be presented to

the user. Upon user command, the video may start to play at a timestamp when the said query word was uttered. It should be appreciated that a plurality of results that match the queries may be returned to the user. The software will jump to the desirable timestamp based on the user preference. For instance, the user may opt to jump to the first timestamp where the query match is found, or alternatively the user may manually select the timestamp to jump to based on the results returned by the software. As mentioned earlier, the search does not have to be limited to Time-associated text but all kinds of information, at all kinds of media forms, related with time or not.

[00499] In one embodiment, the search in the disclosed subject matter is a query search based on transcripts associated with the media of interest. The transcript-based query search can be performed using 3 steps illustrated below. The time-associated transcript may be provided by the content provider, e.g., close captioning or subtitles from movies. Upon user input of a plurality of queries and query searching based on transcripts, the software will jump to a particular timestamp where the said queries are matched with the transcripts. It should be appreciated that if a plurality of matching results are found, the software can jump to a particular timestamp selected from the matching results, based on the user preference. In one aspect, a default setting can be implemented such that default mode of operation of the software is to jump to the earliest timestamp; in another aspect the default setting the software can return a plurality of results to the user and the user can manually select the timestamp to jump to.

[00500] Figure 36: Flowchart of media navigation/playing control

[00501] In another example, a query of "happy lab" may be inputted into the software, and software will determine the a plurality of locations/timestamps where the said query occurs in the text information. Consequently, the user can locate the timestamps within the media, where the said timestamps are matched with the query. The association algorithm may comprise a method selected from the group consisting of temporally mapping to, temporally being prior to, or temporally being after. The time constraint between the said timestamp and said query may be user-defined or automatic. For instance, the user can input the "happy lab" in the search field of "transcript", and input the "5 seconds after" in the field "association method"; the software will return all timestamps where the said timestamps occurs 5 seconds after the corresponding query in the transcript. This way, the user can determine the correct timestamp. In

another aspect, the user can input the "happy lab" in the search field of "transcript", and input the "in the first half of video" in the field "association method"; the software will return all timestamps where the said timestamps occurs 5 seconds after another timestamp when the said query occurs (e.g., were said by the presenter) in the transcript. In another aspect, the timestamp associated with the word/phrases/sentences in the search query (association algorithm is previously described) is extracted and the media stream then starts playing from the said timestamp. A confirmation may be promoted to the user before jump to the new time of playing. The user can use various methods to submit the query.

[00502] The basic pipeline including transcribing audio into text is shown below in Figure 30. This way, the software will search in the video transcript for the query, find the matching results and jump to the timestamp when the said query occurs. In another aspect, with the user definition of association method (e.g., "5 seconds prior to"), the software will search in the video transcript for the query, find the matching results and jump to the timestamp 5 seconds prior to when the said query occurs.

[00503] Figure 37: Flowchart of media navigation/playing control using searching in audio-generated text

[00504] The basic pipeline including language translation is shown below :

[00505] Figure 38: Flowchart of media navigation/playing control with Time-associated text including translation

[00506] All methods, e.g., inputting and submitting query, matching, ranking, and safeguarding, mentioned in Section 2, can be used here. Multiple results can be returned to the user with timestamps, previews (in all media, text, video snapshot, etc.), ranking scores. The search can be on mixed fields. The user has the freedom to select from one of the results.

[00507] In one embodiment, a plurality of the search results can be represented by a GIF animation to the user, which help the user to understand the context. In another aspect of the embodiment, a plurality of the search results can be represented by a video clip to the user. When the user place the mouse cursor on the video clip, the video clip will play (muted or with sound), so the user can preview the index key to determine whether it is what he/she want.

[00508] A representative graphic user interface containing the search results is shown in Fig. 39.

[00509] Figure 39. An exemplified GUI for the software. The user put in the query, and view 4 search results. When the cursor is moved on to the video clip window (eg. Timestamp 1), the clip will automatically play in the clip window for preview. When user click on any of the windows, the video will play full screen from that particular timestamp.

[00510] End point search: in yet another embodiment, text-based search will determine and ending point of the clip to be played (part of the video). The search and matching methods are essentially similar to that in aforementioned embodiments. Different from the starting point, the end point is being specified. The pipeline is illustrated below.

[00511] Figure 40: Flowchart of media navigation/playing control with end point search

[00512] In yet another embodiment, at least 2 queries will be inputted into the software, with the first query representing the starting point for the clip, and the second query representing the end point of the clip to be searched for. For instance, the user may specify a query "happy" in the field "starting word", and specify a query "lab" in the field "ending word". The software will return a plurality of video clips with the corresponding starting word and ending word. Furthermore, the user may specify additional information such as the length of clips as another field for the search to narrow down possible clips. The "length of segment" field can either be a user defined interval in the form of "less than 7 minutes", "greater than 7 minutes" or "3 to 5 minutes", or representing with signs such as "< 7m", "> 7 m" and "~= 7 m", respectively. The user may also manually select the clips of their desire based on the search results. The user can also use a composition of the "starting word" and "end word" in one query, such as "starting with happy and end with lab.", which can be inputted through voice command known in the field of computer industry or text command.

[00513] Graphic user interface: A exemplified GUI of the software is shown below. In this example, "the bRaln" is submitted by the user as the query, and a list of matching results are returned. Both timestamps and the transcripts containing the

matches are shown to the user. The user can select any of the match results and software will play the video from the timestamp selected.

[00514] Figure 41. Graphic user interface for the software: an example

[00515] 5. Phone and media chat recording/voicemail search

[00516] Current voice box technologies rely on voice recording or transcripts to present the contents to user. However, there is no search technology available to users for finding the voicemails rapidly based on contents or keywords. Nor is there any analysis on the voicemail or any media recordings. Hence, we propose text analysis, including but not limited to, searching, categorization, spam filtering, semantic analysis, topic modeling, sentimental analysis, on the audio-generated text from phone or media chat recordings or voicemails.

[00517] It should be appreciated that the voice box described in the disclosed subject matter can be made available for a variety of communication methods, such as telephone, mobile phone, online audio/video chat (e.g. Skype video messages, Google Hangouts, WhatsApp, WeChat voice message), Voice over IP (voice being transmitted over the Internet or any kind of packet-switching networks such as Bluetooth scatternet), voice mail messages, answering machines, etc. It should be further appreciated that this disclosed subject matter should not be limited to voice or audio calls, but also messaged-based communication, such as text messages.

[00518] In one embodiment, our disclosed subject matter allows users to rapidly locate the voicemails containing the query of interest inputted by user. The basic processing pipeline for the Time-associated text search in voicemails is illustrated below. The software will take a plurality of user inputted queries, match queries with Time-associated text information in media based on the text domain to locate the relevant voicemails (e.g., based on transcripts), and return the results of relevant voicemails to user. The user may choose to play the voicemails based on search results. Either the whole voicemail or a segment of recording within the said voicemail containing the said query word can be presented to the user. It should be appreciated that the query can be a plurality of words, phrases or sentences. Either exact match or fuzzy match methods previously described in the disclosed subject matter may be used. The search is not limited only to the transcript of the voicemail, but also all types of text information, such as the caller name or phone number. In one aspect, a new number

not recognized by the current phonebook will trigger a search on the internet for such as number, and relevant caller information will be fetched for searching and ranking purpose. Ranking can be done with different user preference. In one aspect, the ranking can be done based on the date and time for calling; in another aspect, the ranking can be done based on caller id (recognized number or new number; the group where the number belongs to: eg. family, friends, etc).

[00519] Figure 42: Flowchart for searching text query in text information in voicemails

[00520] In another embodiment, upon completion of the aforementioned search process, the software will play the relevant voicemails beginning from the timestamps associated with the queries. For instance, if the query "happy lab" is inputted, the software will match the query with the voicemails based on the Time-associated text such as the transcript, and play the media from the points matching the query. In another example, a query of "happy lab" may be inputted into the software, and software will determine the a plurality of locations where the said query occurs in the transcripts of the voicemails. Consequently, the user can locate the timestamps within the voicemails, where the said timestamps are associated with the said query. The association algorithm may comprise a method selected from the group consisting of mapping to, being prior to, and being after. The time intervals between the said time-stamp and said query may be user-defined or automatic. For instance, the user can input the "happy lab" in the search field of "transcript", and input the "5 seconds after" in the field "association method"; the software will return all timestamps where the said timestamps occurs 5 seconds after the said query in the voicemail transcript. This way, the user can determine the correct time stamp in the voicemail. It should be appreciated segments of voicemail containing the query may also be generated and presented to the user. One aspect of the embodiment is illustrated in the flowchart below:

[00521] Figure 43: Flowchart for playing voicemails based on matching locations

[00522] Again, we allow text-only, audio-only, or hybrid search using methods mentioned in Section 2. All methods, e.g., inputting and submitting query, matching, ranking, and safeguarding, mentioned in Section 2, can be used here. Multiple results can be returned to the user with timestamps, previews (in all media, text, video

snapshot, etc.), ranking scores. The user has the freedom to select from one of the results.

[00523] It should be appreciated that the Videomark technology described in section 3 can also be used here for voicemail search and management.

[00524] It should be appreciated that because our definition to media includes all kinds of media that carry information, the proposed method can be used to classify and organize, including spam filtering and automated tagging, text-only messages too, including but not limited to, Short Message Service (SMS) messages and Internet-based chatting text. Hence those types of media will also benefit from this disclosed subject matter.

[00525] 5.1 User voice input for query

[00526] The user may input the query using voice recognition methods known in the field of computer industry, including but not limited to, Hidden Markov Models (HMM), Dynamic Time Warping (DTW), Neural Networks, Deep Learning Models. The voice input may be converted to text domain for further search. With user voice input, a flowchart for the software is illustrated below:

[00527] Figure 45: Matching speech-to-text queries with voicemails

[00528] Below is an alternative flowchart of voice input query search and voicemail playing where the searching and matching of query is done on the audio domain (e.g., sound waveforms, or transform of the sound waveforms such as spectrogram) instead of text domain (e.g., transcript). It should be appreciated that joint audio-text domain searching and matching can be performed. It should be further appreciated that a multi-tier searching process can be performed, where the searching and matching can be performed on the text domain first, and then on the audio-domain secondly.

[00529] Figure 46: Matching audio queries with voicemail in audio domain

[00530] 5.2 Call categorization and spam filtering

[00531] In one embodiment, the disclosed subject matter will automatically categorize or classify voicemails and incoming calls into different categories, including identifying spam voicemails. The classification can be done in text domain (based on Time-associated text information such as the transcript), audio domain (e.g., sound

waveforms, or any transformation of the waveforms), or combination of thereof. A rule-based approach or a statistical approach, or combination of thereof (such as probabilistic logic programming) may be used.

[00532] In rule-based approach, certain words/phrases/patterns will automatically trigger the classifier, a computer program that categorizes a voicemail or a call into different types, such as "important", "spam/junk", and "social". For example, a keyword such as "campus emergency" can trigger the classifier to consider this voicemail as "important" while words such as "on sale" might lead the classifier to believe that this voicemail is a spam. As another example, a call from certain callers, such as close family members, will be considered by the classifier as important while a call from a public phone number (e.g., 800-XXX-XXXX) may not be considered as important. The rules can be represented in many ways to computers. One approach is through logic programming. In one embodiment, we can use category names as predicate names. For example, important(1234567891) means that any calls from the number 1234567891 is important or spam(8001234567) means that the number 8001234567 is a spam number. As another example, the rule "any 800 number is a spam unless it is not tagged as spam by the user" and a user specified rule ("80033 10500 is not a spam" - AT&T's customer service number) can be express as:

[00533] spam(X) :- X=[800l*], not -spam(X).

[00534] -spam(80033 10500).

[00535] The keyword not is called negation as a failure, which is a powerful knowledge representation tool of non-monotonic logic. The notion - (dash) means a logical negation. Hence, the literal not -spam(X)

[00536] means "the number X is unknown to not be a spam". When a user manually tag a number X as not spam, we create a literal -spam(X) in the knowledge base.

[00537] In statistical approach, some training samples will be used to initialize the classifier, which can be provided by the service provider via manual annotation. For each caller (spam or not), the carrier will create a feature vector, which contains information about the call, such as length of call, time of call, calling frequency, how many people received calls from this, natural language features from the transcript of the call if it is a voicemail. Besides the feature vector, the carrier will label whether the

caller is a spam caller or not. For spam callers, they can get such info from many ways, such as users complaint. Then, a classifier can be trained to recognize spam callers. In one embodiment, a caller not labeled as spam is not necessarily not a spam caller - could be false negative. Then the classifier can treat callers not labeled as spam as unlabeled data and train itself accordingly. In another embodiment, the classifier can be updated when more data, especially those submitted by users become available. Users also tag voicemails or calls into different categories and such tagging can be used to update the classifier using machine learning approaches known in the field of computer science. In yet another embodiment, additional information can be provided to the users to help them decide and tag whether a caller is spam, such as how many people this caller have called over the past 24 hours, and the geographical variance of the destinations of the caller. In yet another embodiment, a user has the option to not to share his/her tags with the carrier nor allow the carrier to use data of his/her call records to update the system-wide classifier for reasons like privacy concern. Then the phone can use user tags to update the classifier locally, only for the user.

[00538] The statistical approach can have different modelings. In yet another embodiment, the statistical approach can be modeled as a regression problem where a score of spam likelihood will be outputted, instead of a binary decision, spam or not. Algorithms to train a statistical classifier or regressor include but are not limited to Support Vector Machine, Decision Tree, Artificial Neural Network, etc. In yet another embodiment, ensemble learning approaches such as boosting or bagging can be used to boost the performance of classifiers/regressors. In yet another embodiment, this problem can be solved using Deep Learning methods, such as Deep Neural Network, Convolutional Neural Network, Regression Neural Network, etc. Through a Deep Learning approach, the step of feature vector construction can be greatly reduced. For example, the natural language features from voicemails may not need to be extracted but can be feed into the deep learning modele as raw data.

[00539] The language features to train the classifier includes n-grams, structure features, semantic dimensions, etc. Examples : n-grams, structure features, semantic dimensions flowcharts. Other data features can be used to train the classifier too, such as calling time. We can also leverage social networking. For example, a spam call to your close friends might be a spam call to you and a spam call effecting many users might be a spam call to everyone. In one embodiment, if majority of friends to a user

on a social network platform consider a caller as spam, then it is a spam to the user. In another embodiment, the chance that a caller is spam to a user is determined by a

weighting equation $\hat{A} = \frac{\sum_{i=1}^N a_i b_i}{\Lambda}$ where $b_i=1$ or 0 means whether the i -th friend of the user tagged this caller as spam and d_i is the distance from the i -th friend to the user. The value of X determines the likelihood that this caller is also a spam to the user. The distance from the user to a friend on social network can be defined in various ways, such as how often they communicate, how frequently are they in the same picture, etc. The voicemail/call categorization system can use text information such as the transcripts or caller ID, and non-text information, such as the number, time and duration of the call, in separate or collectively fashion.

[00540] In one embodiment, a received voicemail/call will be classified in both voice and text domains. A speech-to-text conversion converts audio signal to text. Classifiers in voice and text domain will both make decision on the voicemail/call. The results from both classifiers will be fused via majority vote and presented to the user. The user can manually apply tags and those tags can be used to update classifiers. On top user-customized classifiers, system-wide classifier can also be trained and applied. For example, a frequent spam caller will be labeled as spam for all users. Information fusion algorithms will be used to make judgment when different classifiers disagree, using methods including but not limited to, maximum entropy method, JDL/DFIG model, Kalman filter, Dempster-Shafer algorithm, central limit theorem, Bayesian networks, etc. The classifiers, modeled as either multiple uni-class classifiers or multi-class classifier, can be trained using architectures, including but not limited to, Naive Bayes classifier, Hidden Markov Model (HMM), Artificial Neural Network (ANN), Support Vector Machine (SVM), and Deep Learning.

[00541] Figure 47: Flowchart of voicemail/call classification. Note that the classification can be done in voice or text domain only. Note that this approach also works for modeling the problem as regression, which does not binarily tell whether a call/caller is a spam but the likelihood.

[00542] In another aspect, the software will combine local information with information from other databases (e.g., database online; known spam caller/telemarketer) to form the database for spam filtering.

[00543] 6. Audiobooks/Podcasts/Radio search and management

[00544] The methods proposed here also work for media contents primarily focusing on audio, such as audible books, podcasts and radio (including Internet-based radio). Audiobooks will benefit greatly from the disclosed subject matters. In one embodiment, the user will be able to advance to the chapters and timestamps when certain queries occur. For instance, when the query "Steve Jobs" is inputted, the software can accurately locate the chapters where the said query occurs (eg. "Steve Jobs" is mentioned in the audio). In another embodiment, the user will be able to play the audio segments when the queries occurs. Similarly, both starting and ending points of the audio clips may be specified. Moreover, the search in the transcript will also enable user to manage the audiobook chapters or audio segments in a convenient way by topics, similarities and contents, within 1 audiobook or across a plurality of audiobooks. Similarly, podcasts can use the aforementioned features and technology described in the disclosed subject matter. Podcast in the form of video or audio-only can all benefit from the disclosed subject matter. In another aspect, recording of radio programs can also benefit from the disclosed subject matter, in a similar fashion.

[00545] A flowchart representing one aspect of the search technology applied in audiobook, podcast and radio recordings based on Time-associated text information is illustrated below. The software will allow user to play the audiobook, or podcast, or radio recording from the timestamp associated with the query words.

[00546] Figure 48: Flowchart of search in audiobook (including any audio-focused media entity, such as podcast and radio.)

[00547] Upon returning the results, the user may select the segment he/she prefers so the audiobook/podcast/radio will play from the timestamp of choice.

[00548] In another aspect, the results will be ranked before it is returned to the user:

[00549] Figure 49: Flowchart of search in audiobook (including any audio-focused media entity, such as podcast and radio.) with result ranking.

[00550] If the transcript of the audiobook, podcast or radio recordings are not available, the software will first transcribe the audio information to generate the

transcripts. In one aspect, the software will play the segment from the timestamp associated with the query matching locations, as illustrated in the flowchart below:

[00551] Figure 50: Flowchart of search in audiobook (including any audio-focused media entity, such as podcast and radio.) with transcribing and result ranking.

[00552] Another flowchart representing another aspect of audiobook/podcast/radio-recording search based on transcript is shown below. In this case, voice input with speech recognition may be used to guide the navigation through the chapters in the audiobook. For example, user may say "Jump to when Steve Jobs was fired", and the software will do so to play from the timestamp when Steve Jobs was fired in the audiobook; in another example, user may say "play the audiobook until Zuckerberg bought the Oculus". Furthermore, the artificial intelligence may also be integrated into this software. For instance, user can say "play the audiobook until my coffee ordered was delivered by Amazon", the software will play the audiobook until the coffee ordered on Amazon is delivered as indicated by the system notification. Such integrated intelligence can be realized on desktop, laptop, tablet computers, smartphones, smart systems (e.g., Amazon Alexa) or other embedded systems.

[00553] Figure 51: Flowchart of text-based audiobook (including any audio-focused media entity, such as podcast and radio.) search comprising of voice input and audiobook play.

[00554] Advertisement: In one embodiment, the queries that user inputs can be used for advertisement purposes, independent to the media entity or jointly with the content of the media entity. For instance, if the audiobook is supplied for free, at the end of each chapter the user may be required to listen to an audio advertisement. In another instance, the podcast advertisement may be based on the user queries for targeted advertising. In another example, the user queries and users feedback to advertisements can be used for crowd sourcing and machine learning for targeted advertising.

[00555] All methods, e.g., inputting and submitting query, matching, ranking, and safeguarding, mentioned in Section 2, can be used here. Multiple results can be returned to the user with timestamps, previews (in all media, text, video snapshot, etc.), ranking scores. The search can be on mixed fields. The user has the freedom to select from one of the results.

[00556] It should be appreciated that the audiobook, audio recordings and podcasts can also benefit from the Videomark technology described in section 3. Furthermore, the audiobook, audio recordings and podcasts can be managed and presented to the user using the Videomark technology.

[00557] It should also be appreciated that, for audiobook, the book text can be used instead of the transcript.

[00558] In one embodiment, the user can query the software using the "TymeTravel" syntax/methods described in the disclosed subject matter.

[00559] 7. Integrated Publishing: : text and media

[00560] Today, many lectures or tutorials are digitized into media. Textbooks are also digitized. However, a problem is prevailing: the text in textbooks are not automatically linked with the relevant digitized media lectures/tutorials/demonstrations/experiments. By exploring the transcript of the media lecturing materials, text in the textbook can be linked, highlighted, or displayed in other ways, in a synchronized fashion to the students. This shall provide a hybrid learning experience for the user.

[00561] In one embodiment, the textbook information can be linked to the relevant timestamps in the videos based on media-associated information such as the transcript of the video. For instance, when the student is reading the textbook where "Compton scattering" is mentioned, a hyperlink can be placed in the textbook linking to the corresponding timestamp where "Compton scattering" is discussed. In another embodiment, the media can a combination package of entities consist of videos and books. As such, the query inputted by the user will be searched in all the media entities in the package. For instance, when the query "Compton scattering" is inputted by the user, the corresponding matching results in both textbooks (location where the query keywords occurs) and in videos (the timestamps associated with the query) will be ranked and return to the user.

[00562] All methods, e.g., inputting and submitting query, matching, ranking, and safeguarding, mentioned in Section 2, can be used here. Multiple results can be returned to the user with timestamps, previews (in all media, text, video snapshot, etc.), ranking scores. The search can be on mixed fields. The user has the freedom to select from one of the results.

[00563] In the disclosed subject matter, synchronized textbook highlighting can be done in multiple ways. In one embodiment, each block (terms, phrases, clauses, sentences, etc.) in the textbook is extracted. Then the occurrences in the Time-associated text information, such as transcripts that match with blocks in the textbook, are located and ranked with matching scores. Each occurrence may mean a word, a phrase, a clause or a sentence. The mapping from textbook block to transcript occurrence (or occurrence in other Time-associated text information) may be stored in a database or other data structures for user to look up and query, so that the query speed is accelerated. One block in the textbook may match multiple occurrences in the transcript of the media, and one occurrence in the transcript and other Time-associated text information may also be matched to multiple blocks in the text, with different ranks and matching scores. The ranking algorithm is similar to the methods previously described in section 2. The matching system between the textbooks and the videos can be essentially considered as a multiple-to-multiple mapping system, mathematically. In one aspect, when playing the media entity such as videos, all textbook blocks matching the part of the transcript and other Time-associated text information under playing will be presented to the viewer, along with their locations in the textbook, ranks and matching scores. The user may preview these textbook information in a thumbnail window within the video player, in a picture-in-picture format. The users can visit any or all matches found, automatically, manually or semi-automatically. In one embodiment, the matching and ranking can be done using approximate/fuzzy matching to find the matches of phrases of least distance (string metric for measuring the difference between two sequences), which can be defined in many ways, such as editing distance, Levenshtein distance, etc. In another embodiment, matches can be found using topic modeling where the distance between a transcript line and a textbook sentence can be computed based on their topic vectors. Also, much additional information can be used to help here. For example, locations cited in the index of the textbook will have higher rank. All text matching algorithms (including but not limited to, Naive string searching algorithm, Rabin-Karp algorithm, Finite-state automaton search, Knuth-Morris-Pratt algorithm, Boyer-Moore algorithm, dynamic programming-based string alignment, Bitap algorithm, Aho-Corasick algorithm, Commentz-Walter algorithm) can be used here individually or collectively of any combination. The matching and ranking methods describe in previous sections of this disclosed subject matter can be applied for this purposes. There are also multiple ways to represent the matching to the

users, e.g., highlighting the corresponding text in the textbook or popping up in a window. It should be appreciated that the search can include multiple fields. For example, the student can search a keyword in the textbook and search a particular instrument in the video. In one aspect, the textbook itself can be considered to contain multiple fields, such as the main text body, section head, sidenotes, footnotes, examples, homework, and even the note that the student takes by him-/herself.

[00564] In yet another embodiment, the mapping between text blocks in the textbook and transcript blocks in the videos can also be done in a reversed way by first extracting blocks from the transcripts and then constructing the mapping from transcript blocks to their matching occurrences in the textbook. The presentation can also be done in a reversed way that when the user selects part of the textbook, the corresponding parts in the media, found through searching for matches in the transcript, are shown with ranks and scores. And the user can play any or all matches in the media.

[00565] It should be appreciated that it may not be necessary to find matching for every block in the textbook or every blocks in the transcript of the media. For example, search, indexing or crawling can be done only for important parts of the contents, such as the emphasized terms in the textbook (e.g., those in bold or italic font), or landmarks extracted or user annotated in the media or textbook

[00566] The textbook structure can provide navigation in video watching, or vice versa. For example, once the mapping between blocks in the text and those in transcript and other Time-associated text information are established, the hierarchy on the textbook can be transferred to different segments of the video. Video segmentation can be done using text-based segmentation methods mentioned above, pure audio/video segmentation methods, or simply by transferring delimits from the textbook hierarchy to the video. An illustration of textbook to transcript mapping is shown below:

[00567] Figure 52: Illustrating how a segment of speech is converted into text transcript and then corresponding part in the textbook is identified and highlighted.

[00568] In another embodiment, the matching and linking between the textbook and the audios/videos can also be enabled by the "Videomark" technology described in section 3 of the disclosed subject matter. For instance, the Videomark system can list all the organization of the information in the textbook and in the video. Furthermore, a

glossary can be generated and create links to the corresponding timestamps in the audios/videos and the corresponding text blocks in the textbook files.

[00569] It should be appreciated that matching/mapping between the first media and a second media, or matching/mapping between the first textbook and a second textbook, can also be done using methods substantially similar to the aforementioned aspects and embodiments. All methods, e.g., inputting and submitting query, matching, ranking, and safeguarding, mentioned in Section 2, can be used here. Multiple results can be returned to the user with timestamps, previews (in all media, text, video snapshot, etc.), ranking scores. The search can be on mixed fields. The user has the freedom to select from one of the results.

[00570] It should be appreciated that the disclosed subject matter can be applied to various application of publishing, such as traditional education, entertainment, online education, etc.

[00571] In one embodiment, the user can query the software using the "TymeTravel" syntax/methods described in the disclosed subject matter.

[00572] 8. Other time-associated data

[00573] The search methods in the disclosed subject matter can go beyond time-associated media and to be applied to any time-associated data. In one embodiment, the media could be a multi-channel physiological time series that has timed annotations, while the query is a medical condition that a plurality of symptoms occur with a time-relevant relationship. For example, generalized tonic-clonic seizure (GTCS, formerly grand mal), has two phases, tonic and clonic, that are about 10 to 20 seconds apart. A computer algorithm or a human being can recognize and annotate tonic and clonic phases from physiological time series, including but not limited to, electroencephalogram (EEG), gyroscope, and accelerometer, then the search algorithm will search both phases on multiple channels and set the window of these two phrase to at 20 seconds, in order to automatically alarm that a GTCS happened to the subject. It should be appreciated that the search, analysis, transmitting, management and any manipulation of other time-associated medical data, such as Electrocardiogram (ECG), Electromyography (EMG) can benefit from the disclosed subject matter. For instance, a medical record software can be made using the "Videomark" methods described in the disclosed subject matter. For instance, the different segments of the sensor data such

as EEG can be organized. Furthermore, the user can query the software using the "TymeTravel" syntax/methods described in the disclosed subject matter.

[00574] In another embodiment, the media can be stock indexes and the time-related annotation is the events that happen as time advances. The search query can be that "when S&P500 drops at most 100 points while NASDAQ drops at least 50 points". The two events that "S&P500 drops <100 points" and "NASDAQ drops 50+ points" are timed with stock index log. It should be appreciated that the search, analysis, transmitting, management and any manipulation of any time-associated data in finance can benefit from the disclosed subject matter. For example, some examples of data types that can benefit from the disclosed subject matter are: stock market, foreign exchange rate, commodity prices, bond prices, interests rates, cash flow, market cap, etc. For instance, a financial analysis software can be made using the "Videomark" methods described in the disclosed subject matter. For instance, the different segments of the financial data such as stock prices can be organized using the "Videomark" technology. Furthermore, the user can query the software using the "TymeTravel" syntax/methods described in the disclosed subject matter.

[00575] It should be further appreciated that the search, analysis, transmitting, management and manipulation of any time-associated data can benefit from the disclosed subject matter.

[00576] 9. Enhanced media enjoyment, composition and editing

[00577] In this section, we discuss several ways that our disclosed subject matter can provide better or new ways to enjoy media.

[00578] Landmark extraction and annotation-driven media watching: Various types of landmarks, such as entities (e.g., trademarks, names, brands), keyphrases (e.g., "a tragedy accident"), emphases (e.g., "I would like to explain again") can be extracted from the transcript to provide guidance for users. Landmarks can be user-defined/annotated or learned by the machine.

[00579] Users' interaction with the media player can teach the system to extract landmarks, e.g., from frequent queries, users' final choice of query locations, etc., if users give the system the permission to use their input to improve the system. Natural Language Processing (NLP) methods can also be used to extract landmarks. For example, named entity recognition can extract all important names from the transcript.

As another example, rule-based keyword extraction can identify the timestamps where new concepts are introduced. An effective rule is called "a kind of" rule, e.g., "A computer is a kind of electronic devices." Any sentence that fits the "a kind of" rule is likely about introducing a new concept and hence the sentence can be considered as a landmark.

[00580] With landmarks, users can play the media entity without following time lineage. For example, they can jump to the timestamps of different landmarks and watch only a few seconds to catch the most interesting or important parts. The "Videomark" technology described in the disclosed subject matter can be used to management the landmarks extracted and assist users to navigate through the video. A preview/trailer of the media can be automatically generated from landmarks.

[00581] In one embodiment, the segment of video before the landmark may be condensed into a fast forwarded trailer or preview. For instance, if the time constraint of "2 minutes before and 5 minutes after" is by default (or the time constraint can be specified by the user) and the landmark keyword of "Macintosh" is mentioned. The software will automatically generate a shortened video comprising of video segments associated with the landmark keyword ("Macintosh") and satisfying the constraint "minutes before and 5 minutes after" the said landmark keyword. It should be appreciated that the time constraint can be specified using the "TymeTravel" syntax specified in the disclosed subject matter. In another embodiment, the segment of video associated with the landmark timestamps may be condensed into a fast forwarded montage trailer when the screen is spatially spitted into a plurality of smaller videos. The user can select the relevant segment for viewing by clicking on the smaller video in the screen. It should be appreciated that the segments may be arranged per temporal order or per the ranking algorithms previously described in section 2. In yet another embodiment, the segment of video between 2 landmarks are slow motioned for the user to view.

[00582] It should be further appreciated that the landmark generated by the disclosed subject matter can be used in combination of existing user annotation landmarks or other existing database (such as X-Ray in Amazon video), to provide useful information to user and help the user to navigate through the view.

[00583] Media temporal segmentation from transcript or other audio-related text information: With the help of transcripts or other audio-related text information, the media segmentation can be performed in various way. In one embodiment, the segmentation is performed based time-associated text information such as transcripts. In another embodiment, the segmentation is performed based on the analysis of images of the media. In another embodiment, the segmentation is based on the metadata associated with the video. In yet another embodiment, the segmentation is based on a combination of images, metadata and time-associated text information.

[00584] In one embodiment, the media segmentation is performed temporally using transcripts or other time-associated text information: With the help of transcripts or other time-associated text information, a media stream can be segmented temporally. NLP-based segmentation has multiple approaches, including but not limited to, hidden Markov chain (HMM), lexical chains, word clustering, topic modeling, etc.. The landmarks discussed above can be used as features to teach and train computers to segment in a particular way. In another aspect, we may run topic modeling on all transcripts and cluster sentences based on their topics. Each cluster of sentences becomes a media entity to be presented to the user. Topic modeling algorithms include Latent Dirichlet Allocation (LDA), multi-grain LDA (MG-LDA), etc, can be used for segmentation based on topics. Clustering algorithms include connectivity-based clustering, centroid-based clustering, distribution-based clustering, density-based clustering, etc, can be used for segmentation of media. When using topic modeling on transcripts for segmentation, the text body to train the topic model can be at multiple scales, e.g., transcripts for all media on a website, transcript for a particular media entity, etc. Similarly, the unit of topic modeling can be of various sizes at different levels, such as at sentence level, at 100-sentence level, or at all-words-within-10-minute-interval (temporally defined). The timestamps associated with the segmentation based on transcript will be used as the timestamps for the media entity. In other words, the segmentation is first done on transcript (the text domain), resulting in beginning and end timestamps associated with each text segments, and then these timestamps are transferred to the segments of the videos. In essence, the pairs of the beginning and end timestamps become the delimiters for segmenting the videos.

[00585] A possible flowchart of the algorithm is as follows.

[00586] 1. Segment transcript using a text segmentation algorithm.

[00587] ↓

[00588] 2. Map the beginning and end of each text segment to beginning and end timestamps, respectively, as delimiters

[00589] ↓

[00590] 3. Use the delimiters to slice the video.

[00591] ↓

[00592] 4. return video segments

[00593] In yet another embodiment, the software may detect the discourse features of sentences, and use these features for media segmentation. For example, a sentence beginning with the word "next" is likely to be the beginning of a new segment. The media segmentation can be done at different levels of the transcript text body, such as topic level, sentence level and even word/phrase level. Different temporal constraints can be applied in combination with the text segmentation method to better segment the video. The algorithm with temporal constraint is represented below:

[00594] 1. Segment transcript using a text segmentation algorithm using a set of parameters.

[00595] 4

[00596] 2. Map the beginning and end of each text segment to beginning and end timestamps, respectively, as delimiters

[00597] 4

[00598] 3. If the beginning and end timestamps (delimiters) meets temporal constraints, use the delimiters to slice the video and proceed to Step 4. If the beginning and end timestamps (delimiters) do not meet temporal constraints,, return to Step 1 but use a different set of parameters to segment transcript.

[00599] 4

[00600] 4. return video segments

[00601] It should be appreciated that, in one aspect, in step 3 of previous flowchart, only the delimiters that do not meet time constraints are returned to step 1.

[00602] In one embodiment, we may allow users to segment the video, record the user segmentation data and use user segmentation to train the software to do so. Various machine learning technique discussed previously, such as supervised machine learning, unsupervised machine learning, deep learning and reinforcement learning can be applied generate algorithms for automatic segmentation.

[00603] Further, a summary can be generated for each segment from corresponding transcript and used as the caption for each scene. The composition of the summaries will form a synopsis of the video. Text summarization methods include but not limited to, TextRank, LexRank, and maximum entropy-based summarization. More detailed discussion of summarization will be discussed later.

[00604] The aforementioned segmentation method is novel as it may convert the videos to text-gated video clips for storage. By filtering, processing and combing various clips with a user-defined sequence, a new video can be synthesized. For instance, this method can be used to remove certain word such as "hell" from the video clips. In another aspect, clips with sentences starts with "I am" may be combined together sequentially to form a new video. The "Videomark" technology in the disclosed subject matter can be further combined with the segmentation method for media management and navigation.

[00605] Automated tagging from transcripts and other Time-associated text information Tagging is an effective way to represent the various aspects of a media entity. Lots of applications can be done on tags, e.g., finding similar media entities, advanced search, etc. Currently, media tags come from content creators (e.g., people who upload videos to service provider such as YouTube) or automated extraction from any text associated with it. If the content creator does not provide any tags nor accurate text for tag extraction, then the media piece will be tagless or mistagged.

[00606] So far, Time-associated text information such as transcripts have been ignored in tag extraction. With our disclosed subject matter, the user can use the more detailed contents of the media entity for accurately creating tags.

[00607] There are multiple methods to automatically extract tags in the disclosed subject matter. In one embodiment, the frequencies of all words may be counted and then the most frequent words (by absolute number, percentage, or any other metrics) becomes tags. For instance, if the word "Steve Jobs" have highest counts (e.g. 101

times) by absolute number, then "Steve Jobs" may be used as one of the tag; In another embodiment, a known frequency-based method, tf-idf, may be used. Such a method computes two parameters. TF, is the frequency of each word in each document, denoted as $tf(w, d)$, where w is a word and d is a document. Then we count the document frequency (denoted as DF), the number of documents that contain each word w , denoted as $df(w)$. The tf-idf score is calculated by a equation $tf(w, d)/\log(df(w) + 1)$. Finally, all words can be ranked based on the said tf-idf score and top words will be picked as tags of the document d . In another aspect, topic modeling may be used. All documents may be considered to have a probabilistic distribution over a finite amount of topics. Each topic is a probabilistic distribution over all words (e.g, top 50,000 frequent words in English). For each document, the top words of its top topics become its tags. The latent Dirichlet Allocation (LDA) or its variants, such as multi-grain LDA (MG-LDA) may be used for topical modelling. A combination method comprises of tf-idf and topical modeling may also be used for tagging. Other methods for tag extraction may also be used.

[00608] In another aspect, we may also leverage the information hidden in audio to extract tags or landmarks. For example, accent or emphasis in speech (such features can be detected in the audio domain by frequency analysis) usually mean important words. The vocal signature of the speaker may also extracted. The audio-domain tag and video-domain tag may be used separately or together, sequentially or simultaneously.

[00609] Summarization: With the power of NLP, a transcript can be turned into a summary, synopsis, or plot of the video. This feature will allow users to get a general idea of the media clip if the producer does not provide a synopsis. Text summarization algorithms include TextRank, PageRank, LexRank, maximum entropy-based summarization, etc. Some heuristics can be used, such as the beginning-of-paragraph heuristics (this method can be very effective for lecture videos). The summarization can be done at single-document level, or multi-document level, where a document can have various definitions, such as the transcript of an entire media entity, or the transcripts of all episode s of one season of a TV show. In another aspect, this method can also be used to teach the software to do automatic summarization. For example, most TV shows use the beginning of every episode to quickly review what happened in previous episode. The short review of what happened in previous episode is the summary of

previous episode. The transcript of the short review and the transcript of the entire previous episode can form a pair of data to train the computer to generate summaries. In another aspect, we can use this to generate trailers. Given the transcript of a lengthy media entity, we first summarize its transcripts, and then video/audio segments correspond to the summary sentences become the trailer. Note that here the "transcript" means all text associated with the media.

[00610] Cross-referencing generation: A very common case in speech is that the speaker will refer to a topic mentioned earlier or hint a topic to be discussed later. A link between the referred location (e.g., "now let's discuss Newton's Second Law") and the referring location (e.g., "we will discuss his second law later") can be established to help users jump back and forth between the corresponding timestamps. Crowdsourcing approach can solicit the linkage from users. The linkages labeled by users can be directly used to establish cross-references, or can be used to train the software to do so automatically or semi-automatically via machine learning. Features for machine learning may include word-to-word distance in the transcript, such as those established via vectorized representation of words or distance measure based on distance in a common-sense knowledge base, or the temporal distances described in the disclosed subject matter. The cross-reference generation can also be applied between different types of media, for instance, between video and textbook, or between audio and textbook, etc.

[00611] Knowledge extraction from transcript the transcript itself contains digitized human knowledge, especially those from tutorials or lectures. Because of the total length of video clips on different content-distribution websites, making use of them will bring tremendous impact to artificial intelligence. For example, by analyzing the transcript of 2015 movie "Jobs", the computer can learn that the CEO of Apple was John Sculley who fired Steve Jobs. Those knowledge can be used for further applications, including question answering (QA) where computer systems can automatically answer questions posed by humans in a natural language. Treating all transcripts and other audio-associated information (including subtitles, closed captions, etc.) as a huge text corpus, a knowledge base can be built from them. Ontology learning is one approach that can be applied, involving 8 steps: 1. Domain terminology extraction , 2. Concept discovery , 3. Concept hierarchy derivation , 4. Learning of non-taxonomic relations , 5. Rule discovery , 6. Ontology population , 7. Concept hierarchy

extension , 8. Frame and event detection. Other than ontology learning, information retrieval also studies knowledge base construction from text. Representative systems include OpenIE, NELL, etc. The knowledge extracting can even go beyond finding logical relationships between objects mentioned in the media. In one embodiment, named entity recognition (NER) can be employed to find all tools needed to fix a car if knowing the video is about fixing a car problem and being able to recognize what words in the transcripts are tools. In yet another embodiment, the knowledge learned from multiple videos can be merged into more comprehensive ones. For example, in one video it learns that wrench is a tool that we use to tight things (e.g., a sentence saying "let's tight it using a size 5 wrench") and in another video it learns screws are to be tightened (e.g, a sentence saying "the screw must be tightened firmly"), then it can learn that wrenches are tools to be applied onto screws.

[00612] 10. Recommendation systems based on Time-associated text information and media search

[00613] "See-also" recommendation: Currently, content providers (e.g., YouTube, Amazon Prime Video, etc.) do not use transcripts or other text information embedded in the media to recommend new media for users to consume after enjoying the current one. With the availability of transcripts and other Time-associated text information, the relevance between medias can be calculated in addition to existing sources. Using machine learning, this can be done in either supervised way or unsupervised way, or a combination thereof. First, text features of each document is extracted, such as the topic models or n-gram models. In one embodiment using supervised way, we can use the users' watching behavior to train the software. For example, 2 video clips that are frequently watches consecutively are likely to be very related. In another embodiment using unsupervised way, similarities between two transcripts can be calculated using their topic vectors or the frequency vector of n-grams. The media entities that are mostly close to the entity that the user just finished watching will be presented to the user. All approaches for estimating the similarity between any two media entities can be used individually or collectively in any form of combination. All natural language processing features mentioned above can be used here for machine learning. The features can be at different levels, e.g., word level, phrase level, sentence-level or even document level (e.g., vector representation of documents). The recommendation based on text information can be jointly used with

any existing recommendation approaches. For example, different recommendation approaches can each give a media entity a score and the final score is a function of those scores. The recommendation will be presented to the user based on final scores.

[00614] In another aspect of the embodiment, the query histories of user in the Time-associated text information described in the disclosed subject matter can be used for recommendation system. For example, the search term means what the user wants and the frequencies of search means how much the user want.

[00615] In another aspect of the embodiment, the time constraints can also be used for recommendation. The time constraints carries important information. For example, if user have been looking for "duration less than 2 minutes" around the timestamps associated with the query, this imply that the user wants more user information. .

[00616] The crowd sourcing feature will be enabled and statistics will be collected for analysis. For instance, if many users watched the "see-also" video recommended and quickly put in another query (eg. less than 40 seconds), that means the recommended video may not be well received and the recommendation needs to be updated.

[00617] Classical methods for recommendation generally have 3 groups, collaborative filtering, content-based filtering, and hybrid filtering. There are also some modern approaches, such as context-aware approach, semantic based approach, cross-domain approaches, and peer-to-peer approaches. In this disclosed subject matter, we can use any individual or a combination of all these methods.

[00618] 11. Dynamic Contextual Advertisement based on time-associated information, automatic selection of regions of ad-overlay, and automatic adjustment of overlaying advertisement

[00619] 11.1 Dynamic Contextual Advertisement based on time-associated information,

[00620] media entities, audio or video, have been used to deliver advertisements to users, especially when the user enjoys media content without paying, e.g., Spotify free version, YouTube free version, or Vudu "Movies on us". Currently, content providers do not use information included in audio or transcripts to match the

audiences/viewers and advertisements. The metadata for contextual advertisements are conventionally based on title, topic, genre, actor, director, band, user annotation of the media entity, as well as the user history and/or cookies (eg. search, browsing, purchase histories) to select and deliver contextual advertisement. This is especially problematic when the media entity is user uploaded without property text information (e.g., title, descriptions, etc.), resulting in improper ad targeting.

[00621] With transcripts or other Time-associated text information, the content of the media can be better understood and more matching advertisement can be provided. Advertisement matching approaches include, keyword matching (e.g., maximizing the overlap between keywords of an advertisement and the top word of a transcript), contextual advertisement, and collaborative filtering (eg. making automatic predictions about the interests of a user by collecting preferences or taste information from many users). All approaches for advertisement based on transcript of the current media entity that a user is watching, can be used separately or collectively in any free combination.

[00622] In another aspect, the targeted advertisement can be based on the user query history for searching in the audio-associated information. The advertisement can be selected based on user query history to enable targeted advertisement. The user query used in the Time-associated text information can be used for contextual advertisement purposes.

[00623] In another embodiment, the Time-associated text information, such as transcripts can be used to provide contexts for the media, in a more precisely, well-defined manner. Compared to conventional contextual advertisement approach using the typical metadata, transcripts provide more enriched contexts, with precise temporal definition with timestamps. For instance, a movie can cover a wide array of topics, in different sets. As such, grouping a movie into a genre such as "Romance", "Action", "Sci-Fi" are very insufficient. Even with more specialized categorization such as "a Star War movie", it is still not yet sufficient to differentiate one segment of the movie from other segments. For instance, in one movie, some video segments are about sports (characters are running), some segments can be about cars and flights (characters are driving to airports), and some segments are about romance (characters are staying in a hotel near by the beach). Based on Time-associated text information such as transcripts, the contexts of each video segments can be analyzed and extracted, using NLP, topic

modeling and automatic summarization approaches discussed previously. Other artificial intelligence approaches discussed previously can also be applied to analyze contexts. Consequently, the video can be automatically segmented into different segments and the corresponding contexts can be linked to each segments, using the transcripts and other Time-associated text information. As such, the contextual advertisements can be dynamically delivered based on the video segments, rather than the whole videos. For instance, when the video segments about sport is shown, a sport store ad is shown besides the video by the program; when the driving to airport segment is shown, ads about car dealerships and airlines can be delivered and shown besides the video by the program; when the segments about romance/hotel is shown ads about vocation resorts can be shown besides the video by the program.

[00624] Contextual ads as an application of our search algorithms. In one embodiment, the search technologies we discussed previously can be used for contextual advertisement. In one aspect, the keywords of advertisements in the advertisement network/database can be used as or used to synthesize the search queries to generate results of suitable timestamps in suitable media for contextual ad. For instance, if an advertisement is about running shoes, the keywords are running and shoes, those keywords can be used to search for suitable timestamps containing those words. Also the matches will be compared to the part of transcript close to the matches for the keyword, to make sure times near the matches are suitable for the advertisement as well. Methods we invented above on detecting the suitability of advertising can be used here. Note that keywords can be from multiple sources. Besides manually provided keywords by the advertiser, keywords can be generated from the content of the advertisement itself (such as objects recognized in the advertisement image/video or phrases extracted from the advertisement) or from the user activities (such as cookie or his/her search history).

[00625] A possible flowchart for management and placement of contextual advertisements in media such as video is shown below;

[00626] 1. Transcribe videos in the database

[00627] 4

[00628] 2. Input advertisement keywords as queries

[00629] 4

[00630] 3.Match queries with time-associated text information

[00631] ↓

[00632] 4. (optional) Determining whether each match is appropriate for delivering an ad and remove unsuitable matches

[00633] ↓

[00634] 5. Display contextual advertisement at matches of the said video

[00635] Please note that in the flowchart above, it is possible to specify what kind of matches can be used for advertising, e.g., matching score above a threshold. It is also possible to limit the total number or duration of advertisements to be delivered with one media entity (e.g., no more than 10 minutes of ads in a 2-hour long movie.)

[00636] Contextual ads with segmentation

[00637] In another embodiment, the contextual advertisement can implemented with media/video segmentation based on transcripts. A flowchart about how this process work is shown below:

[00638] 1. Segment the media entity into a plurality of segments based on transcripts

[00639] ↓

[00640] 2. analyze and extract context information of each segment

[00641] 4

[00642] 3. Link the context information with each segments with temporal correlation

[00643] 4

[00644] 4. Query the advertisement network/database for relevant advertisement based on contextual information

[00645] 4

[00646] 5. Retrieve relevant advertisements

[00647] 4

[00648] 6. display the advertisements according to timestamps of corresponding segments

[00649] ↓

[00650] 7. if user click or select the advertisement, redirect user to targeted website or other resources or add items to the shopping cart /wishlists

[00651] ↓

[00652] 8. (Optional) Return to step 1

[00653] The media segmentation can be performed in various way. In one embodiment, the segmentation is performed based time-associated text information such as transcripts. In another embodiment, the segmentation is performed based on the analysis of images of the media. In another embodiment, the segmentation is based on the metadata associated with the video. In yet another embodiment, the segmentation is based on a combination of images, metadata and time-associated text information.

[00654] In one embodiment, the media segmentation is performed temporally using transcripts or other time-associated text information: With the help of transcripts or other time-associated text information, a media stream can be segmented temporally. NLP-based segmentation has multiple approaches, including but not limited to, hidden Markov chain (HMM), lexical chains, word clustering, topic modeling, etc.. The landmarks discussed above can be used as features to teach and train computers to segment in a particular way. In another aspect, we may run topic modeling on all transcripts and cluster sentences based on their topics. Each cluster of sentences becomes a media entity to be presented to the user. Topic modeling algorithms include Latent Dirichlet Allocation (LDA), multi-grain LDA (MG-LDA), etc, can be used for segmentation based on topics. Clustering algorithms include connectivity-based clustering, centroid-based clustering, distribution-based clustering, density-based clustering, etc, can be used for segmentation of media. When using topic modeling on transcripts for segmentation, the text body to train the topic model can be at multiple scales, e.g., transcripts for all media on a website, transcript for a particular media entity, etc. Similarly, the unit of topic modeling can be of various sizes at different levels, such as at sentence level, at 100-sentence level, or at all-words-within-10-minute-interval (temporally defined). The timestamps associated with the segmentation based on transcript will be used as the timestamps for the media entity. In other words,

the segmentation is first done on transcript (the text domain), resulting in beginning and end timestamps associated with each text segments, and then these timestamps are transferred to the segments of the videos. In essence, the pairs of the beginning and end timestamps become the delimiters for segmenting the videos.

[00655] A possible flowchart of the algorithm is as follows.

[00656] 1. Segment transcript using a text segmentation algorithm.

[00657] ↓

[00658] 2. Map the beginning and end of each text segment to beginning and end timestamps, respectively, as delimiters

[00659] ↓

[00660] 3. Use the delimiters to slice the video.

[00661] ↓

[00662] 4. return video segments

[00663] In yet another embodiment, the software may detect the discourse features of sentences, and use these features for media segmentation. For example, a sentence beginning with the word "next" is likely to be the beginning of a new segment. The media segmentation can be done at different levels of the transcript text body, such as topic level, sentence level and even word/phrase level. Different temporal constraints can be applied in combination with the text segmentation method to better segment the video. The algorithm with temporal constraint is represented below:

[00664] 1. Segment transcript using a text segmentation algorithm using a set of parameters.

[00665] 4

[00666] 2. Map the beginning and end of each text segment to beginning and end timestamps, respectively, as delimiters

[00667] 4

[00668] 3. If the beginning and end timestamps (delimiters) meets temporal constraints, use the delimiters to slice the video and proceed to Step 4. If the beginning

and end timestamps (delimiters) do not meet temporal constraints,, return to Step 1 but use a different set of parameters to segment transcript.

[00669] ↓

[00670] 4. return video segments

[00671] It should be appreciated that, in one aspect, in step 3 of previous flowchart, only the delimiters that do not meet time constraints are returned to step 1.

[00672] In one embodiment, we may allow users to segment the video, record the user segmentation data and use user segmentation to train the software to do so. Various machine learning technique discussed previously, such as supervised machine learning, unsupervised machine learning, deep learning and reinforcement learning can be applied generate algorithms for automatic segmentation.

[00673] It should be appreciated that other advertisement technologies will be used in step 4 of last flowchart to determine the appropriate advertisement to match the context of the segment. For instance, user histories (browsing, search, bookmarks, etc), cookies, geographical locations, and other user information may be used. In one aspect, the advertisement network/database are the commercial network/database such as the Google AdSense. For instance, based on the contexts of each segments, the software can query AdSense to retrieve a relevant advertisement for display.

[00674] In another embodiment, the media (such as video, audiobook, podcast, etc) are first transcribed. The process can be represented by the following flowchart:

[00675] 1. Transcribe the media file to generate transcripts

[00676] ↓

[00677] 2. Segment the media file into a plurality of segments based on transcripts

[00678] ↓

[00679] 3. analyze and extract context information of each segment

[00680] 4

[00681] 4. Link the context information with each segments with temporal correlation

[00682] ↓

[00683] 5. Query the advertisement network/database for relevant advertisement based on contextual information

[00684] ↓

[00685] 6. Retrieve relevant advertisements

[00686] ↓

[00687] 7. display the advertisements according to timestamps of corresponding segments

[00688] 4

[00689] 8. if user click or select the advertisement, redirect user to targeted website or other resources or add items to the shopping cart /wishlists

[00690] 4

[00691] 9. (Optional) Return to step 1

[00692] In another embodiment, the segmentation of the video are based on multimodal joint analysis described previously in the patent. For instance, transcript-audio joint analysis can be performed for segmentation; in another aspect, image-transcript joint analysis can be performed for segmentation.

[00693] In another embodiment, the context of the video segments are determined by multimodal joint analysis described previously in the patent. For instance, transcript-audio joint analysis can be performed; in another aspect, image-transcript joint analysis can be performed to determine the context.

[00694] It should be appreciated that a plurality of advertisements maybe delivered for the same segments. For instance, for a segment with the context of hotel, more than 1 ads can be delivered, either sequentially or in parallel. In one aspect, if more than 1 ads are delivered within one segment, the ads may also be displayed repeatedly (ad 1, ad 2, ad3, ad 1, ad2, ad3,adl, ad2, ad3 etc).

[00695] It should also be appreciated that the ad itself can be a media entity and the context of the ad may be used in our disclosed subject matter.

[00696] Ideal timestamps for playing ads: In another embodiment, the software identify the ideal timestamps to start and end displaying contextual advertisements. For instance, based on the context of the segment, the timestamps related to phrases of interests in the transcripts can be extracted. For instance, in an action movie there is a video segment about a criminal breaking into the house. The timestamp when the owner said the word "we need a security system" (defined as T1) can be identified by the software as a good landmark for starting delivering ads about home security. The timestamp the criminal has left the house (defined as T2) may be a good landmark for stopping delivering ads about home security. For example, the software may start displaying contextual advertisements starting 5 seconds after the T1, and end displaying the advertisement 10 seconds after T2. The timestamps for ideal ad-delivery can be determined by NLP methods previously described. In one aspect, the ideal timestamps can be determined by video-text joint analysis or audio-text joint analysis. For instance, if the software detect there is a region with low texture and high homogeneity and the transcript at this time is highly relevant to the context, this can be an ideal timestamp to display and overlaying ad on top of the said region.

[00697] In one embodiment, the ideal timestamps for displaying ads are the timestamps corresponding to part in the transcript that best match with the keywords in ads. For example, if the keyword of ads is "shoes", then times around the occurrences of words like "wear" or "walk" are good times to display ads about shoes. The match can be done between any part (including all) of the ads and any part (including all) of the transcript. The match between words in ads and words in transcript can be computed based on word similarity (e.g., via word2vec) or topic similarity. To assist the matching, a sliding window may be imposed on the transcript (or the transcripts of both the ad and the media entity is the ad itself is also media). The slide window can be defined temporally (e.g., 10 seconds) or lexically (e.g., 10 consecutive words). Then the match between text in that window and the entire or part of the ads will be computed. A window can also be defined as a segment of the ads or the transcript.

[00698] Ad-compatible segments: In another embodiment, the video is categorized by the software into "ad-compatible segments" and "ad-incompatible segments". The ad-compatible segments are the ones which are more appropriate to anchor ads. For the ad-compatible segments, addition of ads will cause less repulsion from viewers; conversely, the ad-incompatible segments are the ones which is less

appropriate to anchor ads. For the ad-incompatible segments, addition of ads will cause more repulsion from viewers. For instance, in a world war 2 movie such as "The pearl harbor", the segment in which the male character is dancing with a nurse can be an ad-compatible segment, which has a context of "dancing" to anchor advertisements such as dancing classes or dancing studios. However, the segments when the Japanese Air Force attacks the Pearl Harbor is an ad-incompatible segments, and displaying ads during this time may upset the user and compromise their viewing experience significantly. The software will automatically determine whether a segment is ad-compatible or ad-incompatible by analyzing the Time-associated text information such as the transcripts. For instance, previously we discussed how to segment the video based on transcripts. Adding an advertisement in the middle of a segment will maximize the chance that the audience finishes watching this advertisement because he/she wants to continue watching the video. Further, the segments that the audience are most unlikely to skip can be detected and ads can be added there. The unlikelihood of skipping can be estimated in many ways, e.g., the topic similarity between a segment and all previous segments that the audience didn't skip, the topic similarity between a segment and all previous segments that the friends of the audience didn't skip. In addition to transcript, other information such as user annotation, comments, sentiment analysis (e.g., do not add ads at a segment which is sad like funeral scenes) can be applied to determine which video segments are ad-compatible. Furthermore, the software will analyze the transcript in conjunction with audio, and images. For instance, in an action movie, the software can treat the video segments with abundance of gun shots as ad-incompatible, as the gun shot scene are frequently the climax of the movies. The other instance, is the detection of actors wearing less clothes (bikinis, nude scenes) based on computer vision algorithms, and render those video segments ad-compatible. The process can be represented by the following flowchart

[00699] 1. Segment the media file into a plurality of segments based on transcripts

[00700] 4

[00701] 2. determine whether the segments are ad-compatible or not, based on analysis of transcripts, and/or audio, and/or image data

[00702] 4

[00703] 3. analyze and extract context information of each ad-compatible segment

[00704] ↓

[00705] 4. Link the context information with each ad-compatible segments with temporal correlation

[00706] ↓

[00707] 5. Query the advertisement network for relevant advertisement based on contextual information

[00708] ↓

[00709] 6. Retrieve relevant advertisements

[00710] 4

[00711] 7. display the advertisements according to timestamps of corresponding ad-compatible segments

[00712] 4

[00713] 8. if user click or select the advertisement, redirect user to targeted website or other resources or add items to the shopping cart /wishlists

[00714] 4

[00715] 8. (Optional) Return to step 1

[00716] Display overlaying advertisements: In one embodiment, the software dynamically display the overlaying advertisements (banner ads overlaying on a small part of the video) based on the dynamic contextual advertisement methods described above. As such, the software will display the advertisements as overlaying advertisements based on the contextual information of video segments. In one aspect, the software perform an alpha composition between overlaying ads and the video. As such, the overlaying ads are partially transparent.

[00717] Companion ads: In another embodiment, the dynamic contextual advertisements are delivered by software via companion ads, where the ads are display alongside the video outside the boundary of the video, such as the black strips on both

ends of a movie. As such, there is no spatial overlay between the video and the advertisements.

[00718] Traditional ads: In another embodiment, the dynamic contextual advertisements are delivered by software via traditional ads (in between video segments), where video is temporarily paused to play the advertisements. It should be appreciated that the aforementioned methods can determine the optimal time for delivering the ads. Also, segmentation based on transcript enable the ads to cause less distraction to the train of the thoughts of the user.

[00719] It should be appreciated that the overlaying ads and companion ads are images. In another aspects, the overlaying ads and companion ads are movies or GIFs.

[00720] In one embodiment, the user can insert a javascript code or do XXX to enable the dynamic contextual advertisement of the media.

[00721] In yet another aspect, the transcript information of the videos watched and the query history can be used in combination to determine the relevance of an advertisement. Furthermore, other advertisement selection methods known in the field of information technology may be used in conjunction of the methods described above.

[00722] When user clicks on ads: In one embodiment, for better user experiences, the software may not want to interrupt the user when he/she click on the dynamic ads. In one aspect, when the user click on the ads, the software will automatically add the ad service/products to his/her wishlist/shopping carts, so that the viewer can look into these products later. In another aspect, the software will compile a list of ads that user clicked on and send this list to the user via email. As such, the user can look at these ad items and make purchase decisions later on, without interrupting the viewing experiences. In yet another aspect, the software will integrate the advertisements user clicked into a personalized webpage for the user to view and shop, after viewing the video.

[00723] Coupon: In another embodiment, the software enable a feature for "watch and save". Discount such as coupons and special offers are activated when the viewer watch the video or media. The software displays coupons or special offers as advertisements. The flowchart is shown below:

[00724] 1. Segment the media file into a plurality of segments based on transcripts

[00725] 4

[00726] 2. analyze and extract context information of each segment

[00727] ↓

[00728] 3. Link the context information with each segments with temporal correlation

[00729] ↓

[00730] 4. Query the coupons database for relevant coupons or special offers based on contextual information

[00731] ↓

[00732] 5. Retrieve relevant coupons

[00733] 4

[00734] 6. display the coupons according to timestamps of corresponding segments

[00735] ↓

[00736] 7. if user click or select the coupons, coupons are activated

[00737] 4

[00738] 8. (Optional) Return to step 1

[00739] In one aspect, the coupon database is services such as Groupon.

[00740] It should be appreciated that the dynamic contextual advertisement can be used at home, on the go, or in movie theaters. It can be used on computers, smartphones, tablets, TV receivers, etc.

[00741] Smartphone/tablet application: The dynamic advertisement technology can be used in mobile streaming applications such as Netflix. The software may offer users the choice of playing less subscription fees when users activate the dynamic advertisement option.

[00742] Movie theater: In one embodiment, the dynamic advertisement technology can be used in the movie theaters. The dynamic targeted ad based on the transcript or other audio-associated information can be shown concurrent with the movie; in one aspect, the dynamic ad is shown on top of or below the movie, or alongside the movie. In another embodiment, the dynamic ad can be shown in between different segments of the movie. In another aspect, overlaying ads can be shown. The movie ticket price maybe subsidized by the advertisement at a reduced rate; in some cases, the movie ticket can be free.

[00743] TV: In one embodiment, the dynamic advertisement technology can be used in delivering traditional TV content. The TV channels may broadcast their program using the dynamic advertisement technology to deliver contextual ads. For instance, instead of interrupting the viewers periodically, more overlaying ads or companion ads can be used based on contexts, for a better user experience.

[00744] Streaming stick/TV box/TV receivers: In one embodiment, the dynamic advertisement technology can be running on streaming stick, TV boxes, TV receivers, or DVR machines.

[00745] Online education/ Massive open online course: in one embodiment, the dynamic advertisement technology can be used in delivering online education or Massive open online course (MOOC). The course lectures are very easy to transcribe. In addition to the contextual analysis based on transcripts, user preferences/history based on students' past exams/quizzes/activities will also facilitate the selection of the appropriate advertisements or coupons by the software. Overlaying ads and companion ads may be preferred in delivering educational contents as they cause less distractions to the user. The advertisements that students clicked on may be presented to the viewer after finishing the lecture, using the methods previously described, instead of redirecting students to the merchant website immediately. The advertisement may lower the overall tuition for the students. The student may still have to come to school for formal tests. The software may also prioritize the ads to be displayed. For instance, study-related ads, book-related ads or career-related ads may be given higher priority in being displayed.

[00746] Videochat/Audiochat advertisement: The dynamic contextual advertisement technology can also be used for videochat, audiochat and phone calls.

Based on the context of the information, relevant ads can be delivered to users by the software.

[00747] All methods used in Sections 9 and 10, such as recommendation, segmentation, and summarization, can be used here.

[00748] 11.2 Automatically finding regions of Video for displaying overlaying advertisements:

[00749] In another embodiment, the software will analyze the video frames of the segments using image analysis, and determine pixel locations and/or size of the advertisements to be delivered. In one aspect, computer vision algorithms can be used to understand the images. In one aspect, image features can be extracted from the video by the software to determine the regions to place the ad. For instance, out-of-focus regions of the images, regions without human faces, regions with less optical flow or other image features can be used to determine the regions for overlaying the ad.

[00750] The following image processing techniques may be used for analysis of video for ad placement: pixel-based operations, point-based operations, padaptortie thresholding, contrast stretching, histogram equalization, histogram matching, histogram operations, image enhancement, image filtering, noise removal, edge detection, edge enhancement, fourier transform and analysis, frequency-domain processing, image restoration, Restoration by the inverse Fourier filter, The Wiener-Helstrom Filter, Constrained deconvolution, Blind deconvolution, Iterative deconvolution and the Lucy-Richardson algorithm, Constrained least-squares restoration, Stochastic input distributions and Bayesian estimators, The generalized Gauss-Markov estimator, shape descriptors, Shape-preserving transformations, Shape transformation, affine transformation, The Procrustes transformation, projective transform, Nonlinear transformations, Warping, ,piecewise warp, piecewise affine warp, morphological processing, Dilation and erosion, Morphological opening and closing, Boundary extraction, Extracting connected components, Region filling, The hit-or-miss transformation, Morphological thinning, Skeletonization, Opening by reconstruction, The top-hat transformation, radial Fourier expansion, Statistical moments as region descriptors, Texture features based on statistical measures, Principal component analysis

[00751] Image segmentation, intensity thresholding, Region growing and region splitting, Split-and-merge algorithm, edge detection, Gaussians filters, The Canny edge detector, Interest operators, Watershed segmentation, Image segmentation with Markov random fields, or a combination thereof.

[00752] The following computer vision techniques may be used for analysis of video for ad placement: Point operators, Linear filtering, Pyramids and wavelets, Geometric transformations, Global optimization, Feature detection and matching (Points and patches, Edges, Lines, etc), Segmentation (Active contours, Split and merge, Mean shift and mode finding, Normalized cuts, Graph cuts and energy-based methods, etc), Feature-based alignment (2D and 3D feature-based alignment, Pose estimation, Geometric intrinsic calibration, etc), Structure from motion (Triangulation, Two-frame structure from motion, Factorization, Bundle adjustment, Constrained structure and motion, etc), Dense motion estimation (Translational alignment, Parametric motion, Spline-based motion, Optical flow, Layered motion, etc), Image stitching (Motion models, Global alignment, Compositing, etc), Computational photography techniques (Photometric calibration, High dynamic range imaging, Super-resolution and blur removal, Image matting and compositing, Texture analysis, synthesis and transfer, etc), Stereo correspondence (Epipolar geometry, Sparse correspondence, Dense correspondence, Local methods, Global optimization, Multi-view stereo, etc), 3D reconstruction (Active rangefinding, Surface representations, Point-based representations, Volumetric representations, Model-based reconstruction, Recovering texture maps and albedos, etc), Image-based rendering (View interpolation, Layered depth images, Light fields and Lumigraphs, Environment mattes, Video-based rendering), Recognition (Object detection, Face recognition, Instance recognition, Category recognition, Context and scene understanding, Recognition databases and test sets, etc), object identification, object detection, Content-based image retrieval, Pose estimation, Optical character recognition (OCR), 2D Code reading (QR codes, etc), Shape Recognition, pattern recognition, color recognition, or a combination thereof.

[00753] It should be appreciated that the regions selected by the software can be a static regions temporally (the region do not move across different nearby frames), or dynamic regions (eg. The regions is moving across nearby frames). In one aspect, when

the overlaying advertisement is displayed in a dynamic region, animation effects of "moving ad" or "fly in", "fly out" can be created.

[00754] Object recognition and object detection for placing overlaying advertisement: In one embodiment, object recognition or object detection is performed to determine where to place the overlaying ad in the video. The software finds and identifies objects in the video to facilitate placing of overlaying ad at a relevant and appropriate location, with relevant starting and end timestamps. For instance, cars in the video can be recognized, and the overlaying advertisement of car dealership can be placed on or nearby the said cars identified. In another example, dogs are identified and pet food advertisement can be overlaid in the region nearby the said dogs in the said video. In another example, Elvis Presley is identified in the videos and overlaying ad can be placed in the regions nearby Elvis. The regions and durations of overlaying ad placement in the video can be therefore determined, for improved viewing experience.

[00755] A possible flowchart for these processes is shown below:

[00756] 1. Analyze the video frames with object recognition or object detection algorithms

[00757] ↓

[00758] 2. Find regions containing or nearby objects of interests

[00759] ↓

[00760] 3. Retrieve the starting and end timestamps when the objects appear in the video

[00761] ↓

[00762] 4. (optional) if the regions identified meet the spatial constraints, proceed to Step 5

[00763] 4

[00764] 5. (optional) if the starting and end timestamps when the objects appear in the video meet the temporal constraints, proceed to Step 6

[00765] 4

[00766] 6. Retrieve relevant overlaying advertisements and display the advertisement in the regions selected

[00767] It should be appreciated that the spatial and time constraints can be set by the user manually, the software automatically, or by a combination thereof.

[00768] Possible object recognition algorithms are: Appearance-based methods (Edge matching, Divide-and-Conquer search, Greyscale matching, Gradient matching, Histograms of receptive field responses, Large model bases, etc), Feature-based methods (Interpretation trees, Hypothesize and test, Pose consistency, Pose clustering, Invariance, Geometric hashing, Scale-invariant feature transform (SIFT), Speeded Up Robust Features (SURF), BRIEF (Binary Robust Independent Elementary Features)), Bag-of-words model in computer vision, Recognition by parts, Viola-Jones object detection, SVM classification with histograms of oriented gradients (HOG) features, Image segmentation and blob analysis, or a combination thereof.

[00769] It should be appreciated that the region selected based on object recognition and detection may be either temporally static (geometrical center of the region selected do not move across different frames) or dynamic (geometrical center of the region selected move across different frames). In the dynamic region, in one aspect the displacement of the geometrical center of the region selected across different frames can be in accordance with the displacement/movement of the said object recognized, providing a pleasant ad viewing experience.

[00770] In one embodiment, plane detection is performed to identify the plane and orientation of the plane in the video. The overlaying ad can be therefore placed in the appropriate perspective and projection consistent with the plane orientation in the image. For instance, carpet flooring can be identified in the video and the overlaying ad can be placed on the region of the carpet with correct orientation and perspective, consistent with the carpet plane orientation and perspective in the video.

[00771] A possible flowchart for the processes containing plane detection is shown below:

[00772] 1. Analyze the video frames with plane detection algorithms

[00773] ↓

[00774] 2. Find regions containing the plane

[00775] ↓

[00776] 3. Retrieve the starting and end timestamps when the plane appears in the video

[00777] ↓

[00778] 4. (optional) if the regions identified meet the spatial constraints, proceed to Step 5

[00779] ↓

[00780] 5. (optional) if the starting and end timestamps when the plane appear in the video meet the temporal constraints, proceed to Step 6

[00781] ↓

[00782] 6. Retrieve relevant overlaying advertisements and display the advertisement in the regions selected, with a perspective and orientation consistent with the perspective and orientation of the said plane

[00783] Texture analysis for placing overlaying advertisement: In one embodiment, texture analysis is performed on the video to find the region and timestamps for placing the overlaying advertisement in the video. An image texture is a set of metrics calculated in image processing to quantify the perceived texture of an image, which gives information about the spatial arrangement of color or intensities in an image or selected region of an image or video. For instance, the regions with less complex texture and/or more homogenous spatial color distribution can be identified and overlaying advertisement can be delivered in these regions in video. The regions with less complex textures and/or repetitive textures typically implied that the is less image complexity in the region (eg. no human faces, less interesting details, etc), which can be used by the software for placing overlaying ad. The region with more homogenous spatial color distribution often is a region that is less interesting to the viewer (eg. flooring, wall, sky, etc), which can be used by the software for placing overlaying ad.

[00784] A possible flowchart involve texture analysis for displaying overlaying ad is shown below:

[00785] 1. Analyze the video frames with texture analysis algorithms

[00786] 4

[00787] 2. Find regions with desirable textures

[00788] ↓

[00789] 3. Retrieve the starting and end timestamps when the regions with desirable textures appear in the video

[00790] ↓

[00791] 4. (optional) if the regions identified meet the spatial constraints, proceed to Step 5

[00792] ↓

[00793] 5. (optional) if the starting and end timestamps when the regions with desirable textures appear in the video meet the temporal constraints, proceed to Step 6

[00794] 4

[00795] 6. Retrieve relevant overlaying advertisements and display the advertisement in the regions selected

[00796] It should be appreciated that the desirable texture can be set by the user manually, the software automatically, or by a combination thereof.

[00797] Possible texture analysis algorithms that can be applied for finding regions for placing overlaying advertisement are listed as follows: Structured Approach, Statistical Approach (Edge Detection, Co-occurrence Matrices, Laws Texture Energy Measures, Autocorrelation and Power Spectrum, etc), Fourier approach (Power spectrum, etc), or a combination thereof.

[00798] Face detection for placing overlaying advertisement: In another embodiment, regions containing human faces can be avoided for ad overlaying in video. It is generally undesirable to overlay ads on human faces.

[00799] A possible flowchart is shown below:

[00800] 1. Analyze the video frames with image analysis and/or computer vision algorithms

[00801] 4

[00802] 2. Find regions based on the results of image analysis and/or computer vision algorithms

[00803] ↓

[00804] 3. If the regions do not contain image features such as human faces, proceed to Step 4

[00805] ↓

[00806] 4. (optional) if the regions identified meet the spatial constraints, proceed to Step 5

[00807] ↓

[00808] 5. (optional) if the starting and end timestamps when the regions with desirable textures appear in the video meet the temporal constraints, proceed to Step 6

[00809] 4

[00810] 6. Retrieve relevant overlaying advertisements and display the advertisement in the regions selected

[00811] Some possible face detection algorithms that can be applied for placing overlaying advertisement are: Weak classifier cascades, Viola & Jones algorithm, PCA, ICA, LDA, EP, EBGM, Kernel Methods, Trace Transform, AAM, 3-D Morphable Model, 3-D Face Recognition, Bayesian Framework, SVM, HMM, Boosting & Ensemble, or a combination thereof.

[00812] Motion analysis for placing overlaying advertisement:

[00813] In yet another embodiment, motion analysis can be applied to select regions for placing overlaying ad on video. In one aspect, the regions with less motions are selected for overlaying ads. In another aspect, the regions with more motions are selected for overlaying ads. In one aspect, optical flow analysis, such as Lucas-Kanade method, is performed and regions with less optical flows, i.e., less motion, will be used for ad delivery. Other motion detection algorithms such as block-matching, template-matching, subtracting a sequence of frames, background and foreground segmentation, can also be applied to identify regions with less motion.

[00814] A possible flowchart involve texture analysis for displaying overlaying ad is shown below:

[00815] 1. Analyze the video frames with motion analysis algorithms

[00816] 4

[00817] 2. Find regions with desirable motion signature

[00818] ↓

[00819] 3. Retrieve the starting and end timestamps when the regions with desirable motion signature appear in the video

[00820] ↓

[00821] 4. (optional) if the regions identified meet the spatial constraints, proceed to Step 5

[00822] ↓

[00823] 5. (optional) if the starting and end timestamps when the regions with desirable motion signature appear in the video meet the temporal constraints, proceed to Step 6

[00824] 4

[00825] 6. Retrieve relevant overlaying advertisements and display the advertisement in the regions selected

[00826] In one aspect, the selected region can be static. In another aspect, the region selected can be dynamic, and the displacement of the region temporally across the frames are consistent with the average displacement/motion of pixels in the regions selected (eg. if a region containing a slow moving car is selected the geometrical center of the region selected displace temporally across different frames in accordance with the displacement/movement of the car).

[00827] Finding out-of-focus regions for overlying ad in video: In another embodiment, out-of-focus regions in the video are identified. In photography and production of the videos, the out-of-focus areas are typically background and less important. The software can identify the out-of-focus regions based on image analysis and overlay the advertisements on those regions.

[00828] A possible flowchart involve finding out-of-focus regions for displaying overlaying ad is shown below:

[00829] 1. Analyze the video frames with algorithms that identify out-of-focus regions

[00830] 4

- [00831] 2. Find out-of-focus regions
- [00832] ↓
- [00833] 3. Retrieve the starting and end timestamps when the regions with out-of-focus regions appear in the video
- [00834] ↓
- [00835] 4. (optional) if the regions identified meet the spatial constraints, proceed to Step 5
- [00836] ↓
- [00837] 5. (optional) if the starting and end timestamps when the out-of-focus regions appear in the video meet the temporal constraints, proceed to Step 6
- [00838] 4
- [00839] 6. Retrieve relevant overlaying advertisements and display the advertisement in the regions selected
- [00840] Some possible algorithms are: contrast detection, Fourier analysis (finding regions with more distribution of higher frequency components), analyze variance of pixel neighborhoods with sliding window, histogram analysis of pixel neighborhoods (eg. 4x4 pixel neighborhood), analyze gradients, or a combination thereof. It should be appreciated that any algorithms used in the field of autofocus control can be used for identifying the out-of-focus regions.
- [00841] Multiple perspectives: in one embodiment, the software automatically detects the video frames with similar perspectives. For instance, in a video captured by multiple cameras, there are image frames captured in different perspectives and angles. The image frames from the same camera angles may not be shown continuously, and the video producers often make switch between footages captured by different cameras back and forth throughout a video. In one aspect, the software can detect the different group of image frames captured from similar camera perspective, and place the same overlaying ad in more than one group of image frames, but skip displaying the said ad in the group of image frames that is temporally in between the said group of image frames captured from similar camera perspective.

[00842] Create animations in the region selected via displacement of advertisement: In another aspect, the size of the overlaying advertisement is smaller than the region selected by the software. The software will only use a subset of the region selected and automatically move the said advertisement within the region. As such, an animation effect can be created, such as sliding, flying in, flying out, etc. These animation pattern can be used to create animation effect of any overlaying advertisement in the region selected.

[00843] A possible flowchart of this process is shown below:

[00844] 1. Analyze the video frames with image analysis and/or computer vision algorithms

[00845] ↓

[00846] 2. Find regions based on the results of image analysis and/or computer vision algorithms

[00847] ↓

[00848] 3. (optional) if the regions identified meet the spatial constraints, proceed to Step 4

[00849] ↓

[00850] 4. (optional) if the starting and end timestamps when the regions with desirable textures appear in the video meet the temporal constraints, proceed to Step 5

[00851] 4

[00852] 5. Retrieve relevant overlaying advertisements

[00853] 4

[00854] 6. Retrieve relevant animation pattern and display the advertisement in the regions selected with animation pattern

[00855] Ranking in Displaying Overlaying Advertisement: in one embodiment, the regions can be ranked based on the spatial constraints and/or temporal constraints, as well as image analysis results (eg. texture properties, motion, object recognition results, contain human face or not, etc).

[00856] Penalty function for spatial locations: In one embodiment, the software imposes a penalty function on the regions selected by the software for ranking purpose. For instance, the software may impose a higher penalty function for regions towards the spatial central regions of the video, and less penalty function towards the spatial peripheral regions of the video. As such, the regions on the peripheral will be given priority for overlaying advertisement placement.

[00857] In one aspect, the ranking is based on score, denoted as the ranking score. The final ranking score can be a function combining the values of factors in various ways, or the combination of those ways, including but not limited to summation, subtraction, multiplication, division, exponent, logarithm, sigmoid, sine, cosine, softmax, etc. FMethods used in existing ranking algorithms may also be used, solely or as part of (including joint use) the ranking function. It should be appreciated that the ranking function does not necessarily have to be expressed analytically, and it could be a numerical transformation obtained or stored in many ways, including but not limited to, weighted sum of those factors, artificial neural networks (including neural networks for deep learning), support vector machines with or without kernel functions, or ensembled (e.g., via boosting or bagging, specialized methods, such as random forest for decision trees) versions of them or combinations of them. The function can be one transform, or a plurality of combination thereof.

[00858] A possible flowchart involve ranking of regions for displaying overlaying ad is shown below:

[00859] 1. Analyze the video frames with image analysis and/or computer vision algorithms

[00860] ↓

[00861] 2. Find regions based on the results of image analysis and/or computer vision algorithms

[00862] ↓

[00863] 3. (optional) if the regions identified meet the spatial constraints, proceed to Step 4

[00864] ↓

[00865] 4. (optional) if the starting and end timestamps when the regions with desirable textures appear in the video meet the temporal constraints, proceed to Step 5

[00866] ↓

[00867] 5. Rank regions selected

[00868] ↓

[00869] 6. Retrieve relevant overlaying advertisements and display the advertisement in the regions selected with higher ranks

[00870] Machine learning for placing overlaying ad: Various machine learning technique discussed previously, such as supervised machine learning, unsupervised machine learning, deep learning and reinforcement learning can be applied to identify the regions for overlaying advertisements. For instance, based on data of human selection of regions, the software can learn how to select regions using machine learning. The human selection data can be crowd-sourced.

[00871] 11.3 Automatic Adjusting Overlaying Advertisement settings:

[00872] In another embodiment, the software will analyze the color information of the region in the video frames for displaying overlaying ad and adjust the display settings of the overlaying advertisement dynamically. The display settings that may be adjusted are: color, transparency, contrast, saturation, size, animation pattern, duration, etc.

[00873] The flowchart for adjusting display settings of the overlaying ad based on image analysis is shown below:

[00874] 1. Analyze the regions of overlay based on image analysis

[00875] ↓

[00876] 2. Retrieve relevant overlaying advertisements

[00877] 4

[00878] 3. Adjust the display setting of the overlaying advertisements retrieved based on the results of image analysis of the said region of overlay

[00879] 4

[00880] 4. Display the advertisement in the said regions with the said color setting

[00881] Adjust color of the overlaying ad based on image analysis: For instance, if the region of overlaying contains primarily blue color, the software may adjust the advertisement to red color so that the ad will be more obvious. In one aspect, average color values of pixels in the regions or color histograms of pixels in the region can be calculated to represent the color properties of the said regions. Also, machine learning such as unsupervised learning or supervised learning can be performed to analyze the color information. For instance, k-means can be applied on the color histogram to identify the primary color composition of the region. PCA, ICA, SVM and other machine learning algorithms may be applied to analyze color information. In one aspect, the color information of the video frames in the region of overlay can be analyzed using HSV transform, where RGB information is converted to HSV image space. The HSV space data of the region can be further analyzed using histogram, average values, PCA, k-means or various machine learning algorithms.

[00882] The flowchart for adjusting color of the overlaying ad based on image analysis is shown below:

[00883] 1. Analyze the color information of regions of overlay

[00884] ↓

[00885] 2. Retrieve relevant overlaying advertisements

[00886] ↓

[00887] 3. Adjust the color setting of the overlaying advertisements retrieved based on the color analysis of the region of overlay

[00888] ↓

[00889] 4. Display the advertisement in the said regions with the said color setting

[00890] Adjusting ad Transparency of overlaying ads: In one embodiment, the transparency level of the overlaying ads are determined by the computer vision and image analysis of the relevant video frames. The software dynamically adjusts the alpha composition (the transparency level of the ad) based on the pixel values of the ads and pixel values of the video frames in the region where they overlay. For instance, if at the

region of overlay, the relevant video frames shows primarily a homogenous color with simple texture, the software will decrease the transparency level of the overlaying ads, as objects with homogenous color with simple texture are typically not important to viewers. However, if at the region of overlay, the relevant video frames shows complex texture, or shows special objects such as human faces, the transparency level of the ads will be increased, so that viewer will see these objects better after ad overlaying. As such, the user will have a much better viewing experience as the overlaying ads are less distracting.

[00891] The flowchart for adjusting transparency level of the overlaying ad based on image analysis is shown below:

[00892] 1. Analyze the regions of overlay based on image analysis

[00893] ↓

[00894] 2. Retrieve relevant overlaying advertisements

[00895] ↓

[00896] 3. Adjust the transparency level of the overlaying advertisements retrieved based on the results of image analysis of the said region of overlay

[00897] ↓

[00898] 4. Display the advertisement in the said regions with the said transparency level

[00899] Adjust size, duration and animation pattern of the overlaying ads: similar to color and transparency settings, other settings such as size, duration and animation pattern can also be automatically adjusted by the software.

[00900] The flowchart for adjusting size of the overlaying ad based on image analysis is shown below:

[00901] 1. Analyze the regions of overlay based on image analysis

[00902] 4

[00903] 2. Retrieve relevant overlaying advertisements

[00904] 4

[00905] 3. Adjust the size of the overlaying advertisements retrieved based on the results of image analysis of the said region of overlay

[00906] ↓

[00907] 4. Display the advertisement in the said regions with the said size

[00908] The flowchart for adjusting duration of the overlaying ad based on image analysis is shown below:

[00909] 1. Analyze the regions of overlay based on image analysis

[00910] ↓

[00911] 2. Retrieve relevant overlaying advertisements

[00912] 4

[00913] 3. Adjust the duration of the overlaying advertisements retrieved based on the results of image analysis of the said region of overlay

[00914] ↓

[00915] 4. Display the advertisement in the said regions with the said duration

[00916] The flowchart for adjusting animation pattern of the overlaying ad based on image analysis is shown below:

[00917] 1. Analyze the regions of overlay based on image analysis

[00918] ↓

[00919] 2. Retrieve relevant overlaying advertisements

[00920] ↓

[00921] 3. Adjust the animation pattern of the overlaying advertisements retrieved based on the results of image analysis of the said region of overlay

[00922] ↓

[00923] 4. Display the advertisement in the said regions with the said animation pattern

[00924] 11.4 Timed multimodal context ad-property file:

[00925] In one embodiment, the software can perform a comprehensive analysis of the media file to generate a timed multimodal context file. In the multimodal context

file, a plurality of timed information can be included, such as timed contexts, ideal ad-playing timestamps, timed ad-compatibility score, timed ad-layout, timed ad-adjustment, etc. The functionality of these properties are illustrated as following in the table in Figure 53.

[00926] In one embodiment, the file is organized as a data structure such as the a matrix, where the rows are corresponding to timestamps (eg. every row is a one second increment: row 1: 0h0m0s; row 2: 0h0m1s; row 3: 0h0m2s....), and columns are properties such as timed contexts, ideal ad-playing timestamps, timed ad-compatibility score, timed ad-layout, timed ad-adjustment, etc. An example is shown in Figure 54.

[00927] With the multimodal context file, the information about the video in how to display contextual ads are documented in detail. As such, the file can be used as a companion file with the video conveniently. It should be appreciated that in some cases these properties can be stored and transmitted to servers in real time, without writing into a file.

[00928] The flowchart for creating the multimodal context file is shown below:

[00929] 1. Transcribe the media file to generate transcripts

[00930] 4

[00931] 2. Segment the media file into a plurality of segments based on transcripts

[00932] ↓

[00933] 3. analyze and extract context information of each segment

[00934] ↓

[00935] 4. analyze and extract other properties such as ideal ad-playing timestamps, timed ad-compatibility score, timed ad-layout, timed ad-adjustment, etc

[00936] ↓

[00937] 5. writing the multimodal context file

[00938] ↓

[00939] 6. (optional) send the file or part of file data to server or remote site

[00940] ↓

[00941] 7. (Optional) Return to step 1

[00942] It should be appreciated that the file here does not have to be a file in a storage of a computer. It can be a data structure maintained in an online database.

[00943] 12. Text-/audio-gated media coding rate

[00944] A bottleneck for distant streaming is the network bandwidth because of the large data nature of media. However, in many cases, the video does not need to be updated at constant and/or high frequency. The media encoding rate can be varied based on the speed of text changing in the transcript. For example, when the professor is not talking, the encoding rate can be lower. If the professor is talking fast, the encoding rate needs to go high. One thing we can take advantage of is that most of the lectures do not have frequent changes and hence high encoding rate (e.g., 30 FPS for video) may not be needed.

[00945] There are multiple ways to implement this. Each line/block in the transcript has a timestamp for its beginning and end. A simple way is to set the coding rate proportional to the number of syllables during each line. Speech recognition can also be added here to improve. The real time interval that each word or even each syllable is spoken can be determined by speech-transcript alignment, such as universal phone models, finite state machine methods, time-encoded method, dynamic programming, etc. Once the interval of each word or syllable is detected, we can set the encoding rate following a function of the speed of syllable changing, for example, let the encoding rate be proportional to the speed that syllables come up. The speed of syllables is either directly detected or estimated by dividing the duration by the number of syllables in each word. This is an easy task for some languages, such as Chinese or Korean which are monosyllabic. In another embodiment, the syllables are directed detected by the audio analysis software and no transcription is needed.

[00946] Figure 55: The illustration above shows how one segment of speech is converted into text, then speech speed is estimated at word level in unit of syllable, and final video encoding rate is set to be related to the speech speed. All steps here can be done in multiple ways or measurements can be in any unit. In the ex-ample above, the total frame number reduces to 48 while without it the number is 450, assuming $\frac{1}{4}$ second per syllable in human speech.

[00947] 13. Virtual reality (VR) and augmented reality (AR) applications

[00948] Our software technology can be coupled with virtual reality hardware/software such as head mounted display, heads-up display and other display technologies. The other embodiments previously described in the disclosed subject matter may be applied to settings of VR or AR.

[00949] In one embodiment, the software may help the user navigate through video instructions. The user may wear a head mounted display, and the instructions are being navigated through using search methods previously described in this disclosed subject matter. For instance, the user may be fixing a car while watching a video tutorial. The user can use audio search to jump back and forth in the video to rewatch any part for as many times as he/she wants, e.g., "jump to where I need to install a filter using a size- 10 wrench". The user can also instruct the media player to pause as well as give a preview of the entire procedure using our landmark-based trailer composition. Alternatively, the user can use the voice recognition to input a query such as "engine" to search for the timestamps when the word "engine" is mentioned. In another embodiment, the "Videomark" technology can be used for guiding the navigation of the video in VR and AR. In yet another embodiment, the query in VR and AR can be specified by in the "Tymetravel" technology; the user can also use the "Tymetravel" to share the query with other people who are using computers/smartphone/VR/AR at remote end.

[00950] Instead of just watching the video, we can add computer vision into the video playing. What the user is seeing is captured by the camera or any sensor on the VR/AR device, our algorithm can run to recognize objects in the user's field of view and even detect which step the user is at, and thus to adjust the media playing accordingly and even to promote some instructions by extracting them from the transcript using the methods described previously.

[00951] The virtual reality that may has an display components, such as an LCD (liquid crystal display) display, an OLED (organic light emitting diode) display, a Liquid crystal on silicon (LCoS) display, a projection display; a head-mounted display (HMD), a head-mounted projection display (HMPD), an op-tical-see through display, a switchable optical see-through display, a selective occlusion see-through head-mounted display, and a video see-through display. Furthermore, the display may

comprise an augmented reality window, augmented monitors, a projection on the patient/projective head-mounted display, selective occlusion see-through head-mounted display, and retinal scanning display. The display may be stereoscopic, binocular or non-stereoscopic. The virtual reality display may have microphone and camera(s) built-in, and have wired or wireless connectivity.

[00952] In one embodiment, the VR display comprises a smartphone with a lens-integrated case (eg. google cardboard) to work with the software. In another embodiment, the VR set has at least one micro-display with lens. In yet another embodiment, the VR set is pupil-forming or non-pupil-forming configuration. In yet another embodiment, the software run on the smartphone, local tablets and local computers. In yet another embodiment, the software is run on the VR set (on its embedded computing device); in yet another embodiment, the software is run on the cloud, either fully or partially. Note that the display method to create VR/AR experience include but are not limited to, parallax display, polarized display, active shutter, anaglyph, etc.

[00953] 14. Voice memo/voice recording software with search based on Time-associated text information

[00954] Currently, the voice memo software only provides recording functionality and very basic memo management functionality. For instance, the audio clips or voice memos are only organized based on time of recording; therefore, the management and search within each voice memo based on the information in the recording (such as the transcript) is not available.

[00955] In one embodiment, the disclosed subject matter present a software that can record the voice memos, manage the memos, search within the memos and segment the memos into clips based on the Time-associated text information. The search methods, ranking methods described previously in the disclosed subject matter can be readily applied in the field of audio recording and voice memos. A representative flowchart implemented in the voice memo software is shown in the figure below:

[00956] Figure 56: A representative flowchart of voice memo search

[00957] Also, the software can play the audio clips from the timestamp associated with the query that is specified by the user, after ranking or returning the

results. It should be appreciated that the methods for specifying the time constraints previously discussed can be used in the voice memo software implementation.

[00958] In another aspect, the voice memo can implement the "Videomark" technology described in the disclosed subject matter. Consequently, the videomark as well as glossary for the audio clips or audio segments can be generated. Thus, the user can manage the audio clips or voice memo based on the information embedded in the clips (such as transcript) and will be able to jump to desirable timestamp within the correct audio clip easily.

[00959] In another aspect, the "TymeTravel" methods described in the disclosed subject matter can be implemented in the voice memo software. As such, the user can specify, share and manage the audio clips in a convenient way.

[00960] 15. Multimodal Karaoke

[00961] Currently, the Karaoke software, Karaoke machine, singing machine is not very convenient to use. For instance, the search for a particular song in the Karaoke machine is typically enabled by looking up by singers, looking up song titles, looking up by languages, or spell the title. The existing system is not convenient for the user if the user does not know the song title and singer name for the song.

[00962] With search methods previously described, Karaoke software and Karaoke machine can be developed with build in search capability. In one embodiment, the software can search for the event based on Time-associated text information such as lyrics. For instance, when the phrase "reminds me of childhood memories" is inputted into the software/machine as the query, the Karaoke software/machine will search, rank and return the results matching "reminds me of childhood memories" to user, when the phrase "reminds me of childhood memories" is within the lyrics of music video or sound track. The search and rank process is similar to the methods previously described in the disclosed subject matter.

[00963] In another embodiment, by clicking on the result, the software will take the user to the timestamp associated with the query (eg. when the phrase "reminds me of childhood memories" is sung in the lyrics/transcript/close caption). As such, the user can sing beginning at the desirable timestamp, without singing the whole song. It should be appreciated that karaoke singing functionalists known in the field of music entertainment and computer engineering can be enabled in the Karaoke software and

Karaoke machine, such as sing with instrument soundtrack only (without original singer's sound), sing with original singer's sound, tuning up & down the keys, look up songs by traditional methods (by singer names, genre, song titles, languages, rankings, etc), cut the unfinished songs, reorder the songs selected, update the music database, etc. A representative flowchart of this embodiment is shown in Figure 57.

[00964] It should be appreciated that the Karaoke software embodied by the disclosed subject matter can run on any computing devices, such as desktops, laptops, smartphones, tablet computers, smart watches, smart wearable devices, video game consoles, TV streaming devices, TV DVR, smart soundbar, smart speakers, smart audio power amplifier. It should be appreciated that the Karaoke software can run locally, on the cloud or on a local/cloud hybrid environment. It should be appreciated that the Karaoke software can be integrated with video streaming websites/providers and social media applications. The karaoke software use the following methods of input: touchscreen, keyboard, remote control, mouse, voice recognition, gesture recognition, etc.

[00965] In one embodiment, the Karaoke software or Karaoke machine support remote collaboration. Different from current software/machine that is designed for singing locally, the Karaoke software and machine in the disclosed subject matter supports singers from different geographical locations. The software has built-in communication capability to allow for a plurality of users to sing together, even if they are situated in different cities. In one aspect of the embodiment, the sound track/music video can be streamed from remote service provider which is cloud-based; in another aspect of the embodiment, the sound track/music video can be stored locally or downloaded to local computer before playing; in yet another aspect of the embodiment, the sound track/music video can be obtained using a hybrid approach of real time online streaming and local storage.

[00966] It should be appreciated that the remote collaboration can be enabled by communication technologies, wired or wireless. Possible communication technologies that can be applied comprise WiFi, Bluetooth, LAN, near-field communication (NFC), infrared communication, radio-frequency communication, TV cable, satellite TV communication, telephone communication, cellular network, 3G network, 4G network, etc.

[00967] In another embodiment, the Karaoke software or karaoke machine allow user to use instrument and form a band for performance and entertainment. Different from traditional software which only allow people to sing, the disclosed subject matter allow user to participate in a plurality of roles. For instance, the user can participate to play an instrument, such as a guitar, drums or piano. A representative flowchart is showing below

[00968] 1. Detect the number of users participating in a song performance

[00969] ↓

[00970] 2. (optional) Users input settings for the song performance

[00971] ↓

[00972] 3. Detect type of participation (eg. as singer, guitarist, pianist, drummer, audience, etc)

[00973] ↓

[00974] 4. Determine the type of sound track the sound track based on number of users and type of participation for users

[00975] 4

[00976] 5. (optional) determine whether all users will use the same type of sound track, or will use individualized sound track

[00977] 4

[00978] 6. Retrieve the correct sound track(s), locally, from the cloud, or from a combination thereof

[00979] 4

[00980] 7. Start playing the sound tracks to users; Acquire acoustic signal or acoustic/visual signal from users, mix the acoustic signals from users with the sound track and play the mixed sound track to users

[00981] 4

[00982] 8. (optional) acquire visual signal from users, and/or present the video signal representing collaboration (such as picture-in-picture format, video montage, virtual reality, augmented reality, or virtual avatars, etc) to users

[00983] ↓

[00984] 9. (optional) record the song performance in audio-only format or video format

[00985] ↓

[00986] 10. (Optional) Return to step 1

[00987] It should be appreciated that different users may be presented with different sound track and different sound mixing settings, based on user preferences and system environment (eg. type of computing device, type of hardware for sound decoding, type and number of speakers...etc) .

[00988] In one aspect of the embodiment, the user will see the visual signals or videos representing the collaboration. For instance, if one user participates as the singer and another user participates as the guitarist, the karaoke software or karaoke machine can present to the user about the music collaboration. The software can capture the videos of 2 users, process the videos (cropping, transforming, add visual effects) and present the processed video to users (eg. in picture-in-picture format). In another example, multiple users are participating in the performance, the software presents processed video in a montage format where each thumbnail window representing a user. In yet another aspect of the embodiment, each users can be represented by a digital avatar (an animated figure/character representing the user) . For instance, the karaoke software/machine can capture the gestures of of each user (through depth camera such as a Microsoft Kinect), and present the avatar in a animated way reflecting the gesture/body movement that users undergo as the perform. In yet another embodiment of the disclosed subject matter, each user can wear a virtual-reality headset or augmented-reality headset (eg. smart glasses; head-mounted display; wearable mount with optics to convert a smartphone into head-mounted display) to visualize the collaboration video. In yet another aspect of the embodiment, the software will capture the facial expressions of each user and digitally overlay the user-facial video over the digital avatar.

[00989] In one embodiment, the visualization of the collaboration video can be performed in 3 dimensional. 3D TVs based on polarization, parallax-barrier based display, lenticular display or head-mounted display can be used to enable 3D visualization.

[00990] In one embodiment, the user can also wear earphones or ear-buds to use the karaoke software/machine. In one aspect of the embodiment, the earphones can be used in conjunction with VR display or wearable displays.

[00991] In one embodiment, the user can play the instrument using virtual instruments. For instance, if the user is playing guitar, he/she may be presented with a virtual guitar on a touchscreen (of the phone, tablet computer, karaoke machine, computer, etc). The software will generate the sound simulating the instrument (eg. guitar), based on user inputs and pre-stored acoustic signature of the instrument (eg. how guitar sounds like). In another aspect of the embodiment, the input device can be a depth camera such as a microsoft Kinect which will record user's gestures representing the musical maneuvers to the instruments (the user will see the instrument displayed while he/she moves). In another aspect of the embodiment, the instrument sound can be automatic and semiautomatic. For instance, if the user decides to participate as a drummer, he/she can specify or program beats that the computer will handle automatically. Consequently, the user will have the option to perform the drumming automatically by the computer or semi-automatically.

[00992] In another embodiment, the instrument can be peripheral devices that can be plugged into the karaoke machine or a computing device (computer, tablet, smartphone, video game consoles, etc). For instance, the peripheral can be a custom device that can input commands represent musical maneuvers, such as a remote control in the form of a musical keyboard or guitar. It may also be a input device in the form of a drum sets representing the drum. In another aspect of the embodiment, the peripheral can be a musical keyboard, an electronic guitar, a drum or other electronic musical instruments that can be interfaced with the karaoke machine or computing devices. In another embodiment, the input can be a microphone recording an acoustic instrument, such as an acoustic guitar, a bagpipe, etc. As such, the software can have an algorithm to un-mix the acoustic signal from the background sound track being played, using technique known in the field of audio engineering and signal processing.

[00993] A representative flowchart of using instrument (virtual, digital or acoustic) is shown as follows:

[00994] 1. Detect the number of users participating in a song performance

[00995] ↓

[00996] 2. If there is at least 1 user decided to participate to play instrument, detect the type of input method representing the instruments in a song performance (eg. virtual drum set using touch screen, or remote control in the shape of guitar, or digital keyboard, or acoustic musical instruments, etc)

[00997] ↓

[00998] 3. (optional) Users input preferences for the song performance; adjust settings based on users' preferences

[00999] ↓

[001 000] 4. If there is at least 1 user decided to participate to play instrument, adjust settings based on the type of input methods

[001001] ↓

[001002] 5. Determine the type of sound track the sound track based on number of users, type of participation for users, and input methods representing the instruments

[001003] 4

[001 004] 6. (optional) determine whether all users will use the same type of sound track, or will use individualized sound track

[001005] 4

[001 006] 7. Retrieve the correct sound track(s), locally, from the cloud, or from a combination thereof

[001007] 4

[001 008] 8. Start playing the sound tracks to users;

[001009] 4

[001010] 9. If a plurality of users are singing, acquire acoustic signal or acoustic/visual signal from singing users using microphones;

[00101 1] 4

[001 012] 10. If a plurality of users are playing instruments, acquire acoustic signal or acoustic/visual signal from users based on the input methods representing the instruments

[001013] 4

[001014] 11. Mix the acoustic signals from users with the sound track and play the mixed sound track to users

[001015] ↓

[001016] 12. (optional) acquire visual signal from users, and/or present the video signal representing collaboration (such as picture-in-picture format, video montage, virtual reality, augmented reality, or virtual avatars, etc) to users

[001017] ↓

[001018] 13. (optional) record the song performance in audio-only format or video format

[001019] ↓

[001020] 14. (Optional) Return to step 1

[001021] In another embodiment, the karaoke software/machine can be integrated with social network software or have social network functionality. The song performance can be shared in real time to other social network users. The song recording can also be shared to social network users.

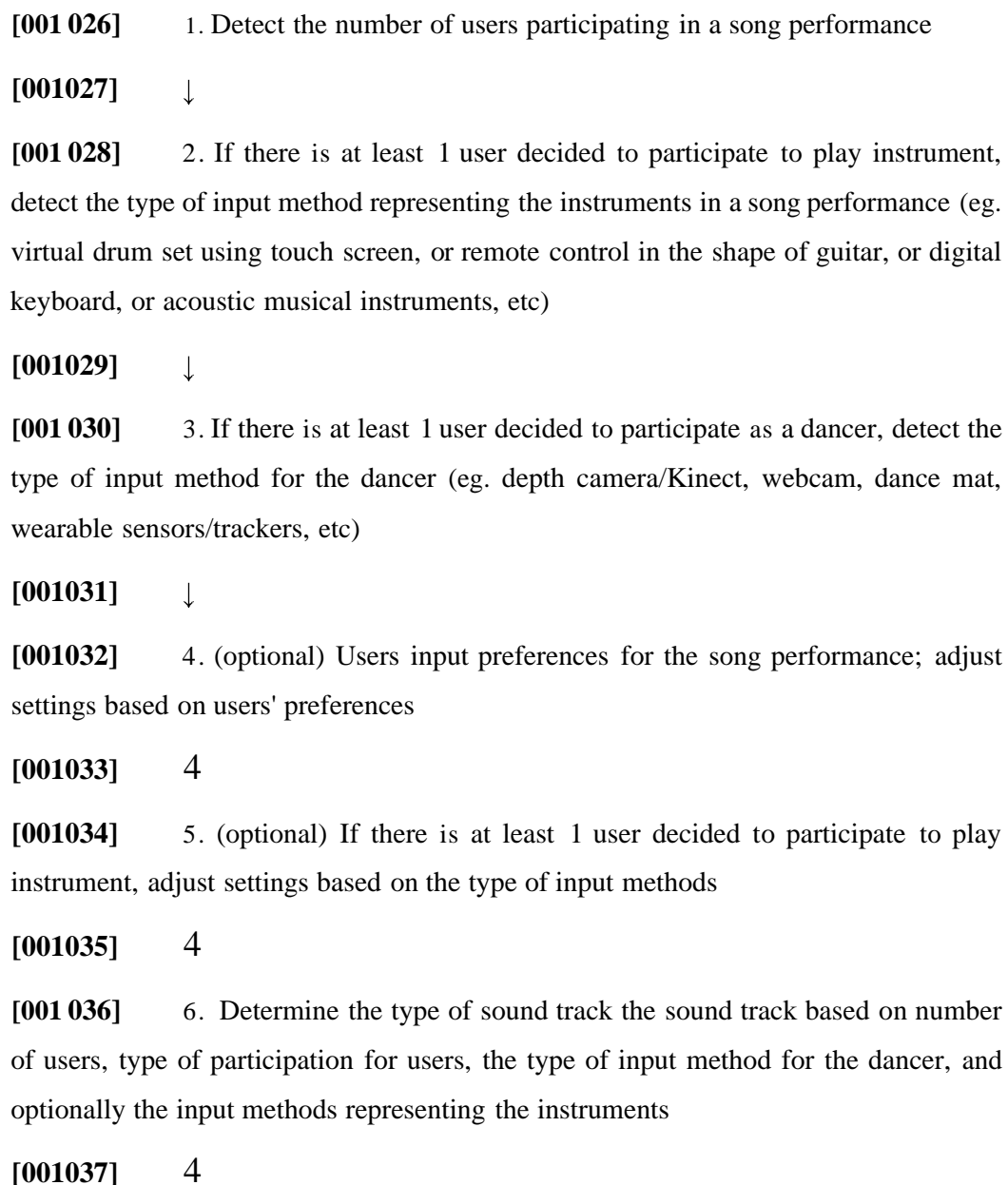
[001022] In another embodiment, the karaoke software/machine can also have media editing functionality, as previously described in the disclosed subject matter. As such, the user can edit the video recording using the karaoke software/machine.

[001023] In another embodiment, the karaoke software can run on existing video game consoles. In one aspect of the embodiment, additional custom remote controls can be plugged into the video game consoles as the input methods to represent the music instruments. For instance, a small remote control resembling a miniature drum set can be made for this purpose and interfaced with the game console.

[001024] In one embodiment, the karaoke software comprises a music video game. The said game can be played on a game console, a computer or mobile computing devices.

[001025] In yet another embodiment, the karaoke software/machine enables entertainment comprising dancing game and karaoke. In one aspect of the embodiment, the software can allow a plurality of users to participate in dancing and singing. For instance, user 1 can participate as the singer singing a song, while user 2 can participate

as a dancer who dances to the song. The body movements of user 2 can be tracked using depth camera such as Microsoft Kinect, a camera, Dance Mats or wearable sensors and trackers. It should be appreciated that the user 1 and 2 can be situated within the same room, or situated remotely and collaborate using the methods previously described. It should be appreciated that the dancer's performance can be presented to user using video the dancer, virtual avatar of the dancer, augmented reality, virtual reality, or a combination thereof. It should be further appreciated that the collaboration involving dancing, singing and instrument playing can also be enabled by the disclosed subject matter. A representative flowchart is shown below:



[001 038] 7. (optional) determine whether all users will use the same type of sound track, or will use individualized sound track of the same song

[001039] ↓

[001 040] 8. Retrieve the correct sound track(s), locally, from the cloud, or from a combination thereof

[001041] ↓

[001 042] 9. Start playing the sound tracks to users;

[001043] ↓

[001044] 10. If a plurality of users are singing, acquire acoustic signal or acoustic/visual signal from singing users using microphone;

[001045] 4

[001046] 11. (optional) If a plurality of users are playing instruments, acquire acoustic signal or acoustic/visual signal from singing users based on the input methods representing the instruments

[001047] 4

[001048] 12. If a plurality of users are dancing, acquire signal from the dancing user based on the input methods for the dancer

[001049] 4

[001050] 13. Mix the acoustic signals from users with the sound track and play the mixed sound track to users

[001051] 4

[001052] 14. If a plurality of users are dancing, present the video signal representing collaboration (such as picture-in-picture format, video montage, virtual reality, augmented reality, or virtual avatars, etc) to users. Optionally, acquire visual signal from non-dancing users may also be acquired and presented to users.

[001053] 4

[001 054] 15. (optional) record the song performance in audio-only format or video format

[001055] 4

[001 056] 16. (Optional) Return to step 1

[001 057] It should be appreciated that one user can participate with a plurality of roles concurrently or sequentially. For instance, a user can participate as a singer, dancer and a guitar player. As such, the aforementioned flowcharts can also solve the "one user, multiple roles" scenario.

[001058] In another embodiment, the karaoke software can rate the users' ensemble performance or individual performance and return a score. The score can be calculated using a plurality of algorithms. In one aspect of the embodiment, the singing performance is pre-calculated and the signer's performance is compared to the pre-calculated value, in the time domain or frequency domain. A score can be returned based on the disparity between the pre-calibrated value and user performance. In another aspect of the embodiment, the score is generated using machine learning algorithms. In supervised learning fashion, the computer will learn how to score the performance based on scores made by humans for previous performances.

[001059] In another embodiment, the karaoke software can be a plug-in for existing software of service provider, such as a youtube plugin, a web browser plugin, etc.

[001 060] The karaoke machine is a media-capable computing device running the aforementioned software. The karaoke machine can be implemented by software running on a plurality of computer architectures. In one embodiment, the software can run on a computer with an operating system and the media entity is a file. Such computer includes but is not limited to, a desktop or laptop running Windows/Linux/Mac OS, a smartphone or tablet running iOS/Android/Windows/Blackberry OS that mainly uses a touchscreen and microphone as user input. It should be appreciated that the computing device should have a build-in sound card or equivalent for decoding the music. The computing device may further comprise an audio power amplifier.

[001 061] The search is done using the computing power of the computer itself. In another embodiment, the software is an application (short as "app") that runs on a computer while the media entity is hosted on a remote server. ChromeBook, a smartphone or tablet running iOS/Android/Windows/Blackberry OS that mainly uses a touchscreen and microphone as user input are examples of this computer. The search,

ranking and other computations to generate the results are mainly done by the remote server, although some data to help the search such as buffer or index can be stored locally. Queries and results are sent between the computer and the remote server. In yet another embodiment, the software is a web browser that can play media and provide user a search interface, the media entity is on a remote server, and the computer can be of any of the forms above. The search, ranking and other computations to generate the results are mainly done by the remote server, although some data to help the search such as buffer or index can be stored locally.

[001 062] In yet another embodiment, the karaoke machine can be implemented in such a way that additional functionality comprising video gaming console and TV/video streaming can be enabled. In one aspect of the embodiment, there is a general-purpose computer implementing this hardware or a special purpose chip. The user can send queries for either local or remote media, using interface provided by this computer or an external device, e.g., a mobile app that connects to the computer or a remote control. The search results and media can be fetched locally or transmitted from remote end.

[001 063] In another embodiment, the karaoke machine can also be implemented substantially in hardware, either analog or digital, and comprising further functionalities such as DVD players, Blu-Ray player, media streaming stick, set top box, smart TV, music players, smartphones, tablet computers, wearable electronic devices, computer expansion cards or gadgets, USB peripherals devices, Bluetooth devices, WIFI routers, extenders, dongles and hotspots, audio systems, home theaters, on vehicle media systems (on car, on plane, on robot, on rocket and on ship), portable media players, game consoles, virtual reality displays, smart glasses, smart watches, projectors, monitors, desktops, smart audio player (e.g., Amazon Echo), etc.

[001 064] For the karaoke machine, the hardware components that may be used to implement the disclosed subject matter comprises of central processing unit (CPU, including microcontrollers and soft cores on field programmable gate array, FPGA), memory (such as random access memory), storage (such as flash-based storage, hard disk), communication components (such as antennas), sound cards or equivalent, etc. Instead of using a CPU to implement the disclosed subject matter, a programmable logic device (such as FPGA and Complex Programmable Logic Devices, CPLD), a specifically made integrated circuit (Application Specific Integrated Circuit, ASIC),

graphic processing unit (GPU) computing platform, and other hardware solutions, can also be used to implement our disclosed subject matter.

[001065] In one embodiment, a wireless karaoke machine consists of a microcontroller (i.e., system-on-chip, SoC), with a co-processor for video content decoding. The co-processor can extract the audio track, captions and video frames using a hardware circuit independent from the CPU. The co-processor fetches media stream from the microcontroller, which fetches media stream from the WiFi or cellular network wirelessly. The co-processor and the microcontroller may communicate via a PCI-E interface, for both media stream and control signals. The co-processor has direct HDMI interface or other video ports to output the content to any HDMI-compatible display. A mobile application software communicates with the microcontroller via Bluetooth to select the media to play and settings. In an alternative embodiment, the media decoding is done by a built-in component of the microcontroller (such as the ARM NEON graphics engine in ARM Cortex A-series) and the caption adding/rendering/generating is done by the CPU.

[001066] In one embodiment, the karaoke machine comprises a plurality of communication modules. Communication protocol such as WiFi, Bluetooth, LAN, near-field communication (NFC), infrared communication, radio-frequency communication, TV cable, satellite TV communication, telephone communication, cellular network, 3G network, 4G network can be enabled.

[001 067] In one embodiment, the karaoke machine comprise a smartphone.

[001068] In another embodiment, a wearable karaoke machine can be implemented. The karaoke machine comprises wearable display (eg. head-mounted display), earphones/speakers, microphone and a media-capable computing device. The karaoke machine may further comprise additional sensors and communication modules.

[001 069] 16. Nonlinear media navigation

[001 070] Here, we introduce a new way to represent time axis and the way that users interact with it, for media (including any time variant information such as stock trend or medical data) playing, editing, and any time-involving user interaction with the content. Through our disclosed subject matter, the movement of a pointing input (e.g., mouse or touch screen) will be mapped to time non-linearly instead of linear in existing prior art. By "media", we mean any carrier of time-variant information, including but

not limited to, books, audio, video (any sequence of pictures, e.g., brain MRI image of one patient over 10 years), time series (e.g., EKG, EEG, stock trend, gas price, etc.). media information is usually organized as "entities", such as a song (audio clip), a movie (video clip), a streaming feed. in By "playing", we mean any user interaction with the media (e.g., watch, listen, play, pause, stop, edit, annotate, distribute, broadcast, stream, fast-forward, rewind, double-speed), especially that involving representing different content along the time. By "computer", we mean any device that can provide the user a means to interact with the media, including but not limited to a laptop/desktop computer, a smartphone, a tablet, a virtual/augmented/mixed reality (VR/AR/MR) headset, a dedicated player, a kiosk. By "time", in one aspect we do not limit in the sense of time. For example, a page of a book, a sheet of a set of data, a file in a folder, can also be considered as a form similar to time. It can even go broader to any graphic user interface (GUI) with a sliding bar, e.g., for adjusting display brightness and audio volume on smartphones. Figures 58 and 59 show a page scroll for Google Books and a name scroll with first letter in names indicated on the scroll in Google Contact. It should be appreciated that the sliding bar/scroll bar represents different information in different media or software. For instance, in video player the sliding bar is called time bar or time axis and it represents advance of time. In the ebook, message application, email software or word processor the sliding bar represent the advance in the document (eg. page number, number of emails/messages, etc). It should be appreciate in different applications the advance rate, how fast the sliding bar/ scroll bar/time bar advances, are named differently. For instance, the advance rate in video player can be named as temporal resolution as it maps pixels in the GUI display to time.

[001 071] Figure 58: A page scroll of Google Books. Note that this book has 400+ pages and locating to the exact page is difficult, especially on computers of small displays.

[001 072] Figure 59: A list scroll based on first letter of names in Google Contacts. Currently, the scroll stops at words beginning with the letter N.

[001 073] 16.1. Time axis in media

[001 074] Playing media is a common function of computers now. A "time axis", part of the "playback control", is a representation of time as a line or an axis. The "time axis", also called a "time bar" or a "progress bar", displays where the media entity is

at, with respect to its full duration. A "playhead" is usually added onto the time axis to indicate the time. The time axis and/or playhead may automatically hide depending on different events in the software, e.g., no user input for a duration of time.

[001 075] Figure 62: The playback control of Apple Quicktime player. [Taken out of Quicktime Player help documentation <http://help.apple.com/quicktimeplayerx/mac/10.10/#/qtp6cee0761b>]

[001 076] Usually, the user interacts with the time axis and/or the playhead using a pointing device, such as a mouse or a touchscreen. We call where the pointing device is pointed at in the user interface as the "focus point" in this disclosed subject matter. A focus point is usually represented by an (X,Y) coordinate in unit of pixel on the graphical user interface (GUI). The GUI can be displayed on an computer monitor (touchscreens or not), smartphone display (LCD, OLED, etc), an digital projector, or in a virtual world through an VR/AR headset. The focus point is mapped to a time instant in the software. The spatial control/resolution of the pointing input (from the input device; frontend) is mapped to the temporal control/resolution of the media player/editor/etc (backend). Usually the time instant is displayed when the focus point is very close to the time axis and/or the playhead to hint the user where the focus point is at (Figure 63). It should be appreciated that a computer or smartphone can have a plurality of pointing input devices of various or same types.

[001077] Figure 63: A time instant and a preview frame shown above the time axis when the focus point (symbolized as a hand with index finger) is hovering over the time axis (no dragging).

[001 078] The user can interact with the time axis and/or playhead through many "actions", including but not limited to, mouse clicks, swipes on the screen, tapping on the screen, holding the mouse keys or constantly pressing the touchscreen. Those actions are mapped into different "controls" to the time axis and/or the playhead. A very common control, denoted as "dragging", is when the user drags and moves the playhead along the time axis to different parts of the media. Another very common control (and another focus of this patent) is "clicking", when the media interaction jumps to the time the user clicked into. The clicking operation is usually done by clicking and releasing a mouse button or touching and releasing the finger on a touchscreen. Note that the user can keep the finger on the touchscreen constantly before

releasing by swiping/sliding or rolling the finger. The user can also "hover" the focus point on top (not literally right above but when the software determines that) of the time axis.

[001 079] If the media is a visual content, such as a book or a video clip, a plurality of the contents or their preview/thumbnails will be shown, usually in a carousel view [Fig. 4A of Google's patent US20130307792A1 filed in 2012], when the focus point is over the time axis, in dragging or hovering.

[001080] In some examples, for each media entity, a focus point maps spatial dimension on the display to a time instant of the video player in a linear relationship, e.g., one pixel to 2 seconds. The user can only drag/click/hover to/at time instants that are a multiplier of the time resolution (eg. 2 second, represented by 1 pixel). In other words, the jump along time is discretely proportional to the distance (usually a projected distance to the time axis) that the focus point moves. This becomes a problem when the media is very long and/or the spatial resolution of focus point is too low. For example, clicking to a moment at the precision of second in an hour-long video on a 5-inch smartphone touchscreen in a cold winter is very difficult.

[001 081] 16. 2. Nonlinear mapping from focus point to time

[001 082] Hence, we propose a nonlinear mapping from focus point to time. The temporal solution is nonlinear along the time axis, and the switching/trigger from linear to nonlinear temporal solution can be trigger by different user interactions or information hidden in the media entity, such as annotations or advertisements. For simplicity sake, we introduce a concept "temporal-to-spatial ratio" in unit of second per pixel, to measure how much time will be moved when the pointing input moves along the time axis. By "high temporal resolution" or a "fine slide/jump over time" we mean that a minimum (minimum in the sense of being detected by the operating system or the driver program of the pointing device) movement of the pointing device maps to a very narrow rewind or forward along time axis of the media player. By "low temporal resolution" or a "coarse slide/jump over time" we mean that a minimum movement of the pointing device maps to a long rewind or forward along time axis. A temporal resolution is negatively correlated with the temporal-to-spatial ratio. For example, the temporal resolution can be the reciprocal of the temporal-to-spatial ratio.

[001083] In most GUI applications, the location of the pointing device ("focus point" in this disclosed subject matter) is a 2-D coordinate (X,Y), bounded by the resolution of the screen. Such a coordinate is also called a pixel. X and Y are, in most cases, non-negative integers, and both X and Y increase at a step of 1 pixel. For explanation sake, we call the direction of X the horizontal and the one of Y the vertical, and assume that the time axis of multimedia player is a straight horizontal line. In most cases, the X coordinate of the focus point is used to be mapped to a time while the Y coordinate has no influence. The computer software needs to establish a mapping/correspondence between X coordinate and time.

[001084] Consider any 3 focus points: (X1, Y1), (X2, Y2), and (X3, Y3), who map time to T1, T2 and T3, respectively. By linear mapping, we mean that $(T3 - T1)/(X3 - X1) = (T2 - T1)/(X2 - X1)$. By nonlinear mapping, we mean that $(T3 - T1)/(X3 - X1) \neq (T2 - T1)/(X2 - X1)$.

[001085] If the time instants corresponding to time points (X1, Y1) and (X1+1, Y2) are T1 and T2, respectively, then the temporal-to-spatial ratio is defined as $(T1 - T2)$ while the temporal resolution is defined as $1/(T1 - T2)$. Figure 64 below illustrates this. Note that the 3 points do not have to be positioned as in Figure 64.

[001086] Figure 64: Mapping in Computer GUI. The spatial location (X,Y) of pointing device is mapped to other properties such as time. .

[001087] Existing software may map the coordinate (X,Y) to time linearly. Consequently, the temporal-to-spatial ratio (or equivalently the temporal resolution) is a constant. But in our disclosed subject matter, the mapping is done nonlinearly. The temporal-to-spatial ratio (or equivalently the temporal resolution) is a function of many factors, such as the coordinate of the focus point, the features, the length of the multimedia entity, and the dimension of the multimedia player window. We have elaborated several ways to establish and update this function. Figure 65 is shows examples that the X-coordinates of pixels are linearly and nonlinearly mapped to time.

[001 088] Figure 65: Examples of mapping from X-coordinates of pixels to times linearly and non-linearly.

[001 089] 16.2.1 Focus point-weighted temporal-to-spatial ratio

[001090] In one embodiment, the temporal-to-spatial ratio is smaller (i.e., more precise control over time) for the part of the time axis displayed on the screen located near the focus point, while larger (i.e., less precise control over time) for the part of the time axis displayed on the screen located farther away from the focus point. The temporal resolution can be weighted by a function, such as the probability density of a radial basis function (RBF) which peaks (actually valleys) at the focus point and attenuates sideways. Examples of RBF include, Gaussian, multiquadrtic, inverse quadratic, inverse multiquadratic, polyharmonic spline, and thin plate spline. The rate that temporal resolution changes (the attenuation rate) is a function of various factors, e.g., proportional to the length of the media. The software designer specifies the maximum and minimum temporal-to-spatial ratio, e.g., smallest ratio as 1 second per pixel at the focus point and largest ratio as 10 seconds per pixel after 3σ (3 standard deviations) of the Gaussian distribution.

[001 091] Here is an example to establish a temporal-to-spatial ratio function based on focus point location in Figure 66.

[001 092] Figure 67 illustrates the idea where r_{max} and r_{min} are 5 seconds per pixel and 1 second per pixel respectively.

[001 093] Figure 67: Temporal-to-spatial ratio as a function of focus point and the length of the media entity. ("w/" means "with"; "w/o" means "without"). The temporal resolution ($1/\text{temporal-to-spatial ratio}$) is finer near the focus point.

[001 094] 16.2.2 Feature-weighted temporal resolution

[001 095] In another embodiment, the temporal resolution is weighted by features associated with the media entity. In other words, to the contrary to the focus-point-weighted approach previously discussed, the focus point is replaced by feature points in this embodiment. The features can include searches made by the user, audience, advertisements, annotations made by the audiences or producer, segmentations automatically generated or inserted by the producer, etc. For example, the temporal resolution can be relatively low (coarse sliding over time) at the beginning of the movie showing credits while relatively high (fine sliding over time) at the main body of the movie. As another example, the temporal resolution can be relatively high around/over famous scenes of the media entity (e.g., the four-note "short-short-short-long" motif at the beginning of Beethoven's Symphony No. 5 or "my mother says life is like a box of

chocolates" in the movie Forrest Gump) while relatively low in other parts, to allow the audience to precisely locate to the most enjoying segment. The "popularity" of a segment can be crowdsourced based on the data analytics and machine learning. For instance, the server can learn that many people watch scene X but skip scene Y. The frequency of temporally correlated user comments is another indicator to indicate the popularity of the scene.

[001 096] It should be appreciated that a plurality feature points can be allowed in one media entity. It should be further appreciated that different feature points can weight the temporal resolution function differently, depending on their types, user preferences, and other factors. Figure 16.6 shows one example that multiple feature points are used to weight the temporal-to-spatial ratio in one media entity. Note that one feature point causes less fine resolution than the other.

[001097] Figure 68: A media entity whose temporal-to-spatial ratio is weighted by two feature points

[001 098] Feature as user feedback as mentioned earlier, many features can be used to modulate the temporal resolution, such as text, image, audio, comment, tag, title, transcript, cross-references, data analytics from a plurality of users, recommendations, reviews, user history.

[001 099] One kind of features can be user feedback. The frequencies that all users click/drag/hover on/over the time axis can be used to generate the temporal resolution over the time axis. Basically, a region of denser clicks/drags/hovers will get finer temporal resolution while a region of sparser clicks/drags/hovers will get coarser temporal resolution. Figure 69 below is one embodiment where each short vertical bar resembles one click/drag/hover.

[001 100] Figure 69: Establish temporal-to-spatial resolution function from users' interaction with the time axis.

[001 101] The algorithm steps are:

[001 102] 1. Log the users interaction with the time axis.

[001 103] 2. Convert the log of users interaction with the time axis to a temporal-to-spatial ratio function.

[001 104] 16.2.3 Focus point and feature-weighted temporal solution

[001 105] It should also be appreciated that the temporal resolution can be controlled by the focus point and features jointly. In one embodiment, two temporal resolution functions, one based on focus point and the other based on features, are first computed independently, and then multiply to produce the final temporal resolution function. In another aspect, the final temporal resolution function can also be a weighted sum of the individual temporal resolution functions. In other words, the final temporal resolution function is a modulation result by user control and feature points. Besides modulation, many other operations can also be used to produce the final temporal resolution function, such as convolution.

[001 106] 16.3. User interaction with nonlinear time axis

[001 107] Existing software uses the flowchart on the left shown below to display the sliding/scrolling on time axis to users. The difference between our disclosed subject matter and existing software is on how to map the movement of focus point to time in the media playback. Figure 16.7 below shows existing approaches and our approach.

[001 108] Figure 70: Left. The flowchart of how existing software and other examples map the coordinate of pointing input device to time. Right. The flowchart of how our disclosed subject matter maps the coordinate of pointing input device to time. Pointing devices can be mouse, touchscreen, etc. Note that the other examples and our approach may run as a loop to constantly read the data from pointing devices and other parameters in the computing device.

[001 109] 16.3.1 Triggering of nonlinear time axis

[001 110] The nonlinear time axis can be triggered by the use of pointing device to activate this function in the software. In one embodiment, if the focus point has been in dragging and/or hovering mode over a location (or around a small region) on the time axis, with or without the playhead, for a certain amount of time (e.g., over 1 second), then the nonlinear time axis will turn on. In one aspect, a pronged dragging of hovering time (eg. >threshold) will trigger the nonlinear time axis.

[001 111] It should be appreciated that the function to compute temporal-to-spatial ratio will dynamically change when the focus point moves along the time axis. When the focus point moves, the places providing fine temporal resolutions (valleys in Figures 16.5 and 16.6) should also move. In one embodiment, if the focus point has been moved far enough from the location where the nonlinear time axis was previously triggered,

and the focus point has been in dragging and/or hover mode over a location on the time axis for a certain amount of time, nonlinear time axis will be re-triggered at the current location of the focus point.

[001 112] Example initial triggering and re-triggering algorithms are illustrated in Figure 71.

[001 113] Figure 71: Exam initial trigger (left) and re-triggering (right) of nonlinear time axis.

[001 114] In the examples above, the non-linear time axis is triggered automatically by the computer algorithms via analyzing the patterns that the user is interacting with the computer. Please note that the nonlinear time axis can also be triggered by the user on purpose through user-defined operation of the input devices, e.g., through configurable settings or dedicated user actions. For example, double click on the time axis, or moving the focus point way above the time axis while keep dragging, could mean turning on the nonlinear time axis, or multi-touch, or gesture recognition, or voice control

[001 115] It should be appreciated that the user interaction with the time axis can be done via a combination of a plurality of input devices, including pointing input devices. For example, the nonlinear time axis is only triggered when the user is also holding a button on the smartphone. Note that the user can opt to have the nonlinear time axis on constantly without triggering. Other input device such as keyboard, microphone, camera, depth camera, gaming controller, joysticks can also be used.

[001 116] 16.3.2 Displaying nonlinear time axis

[001 117] There are multiple ways to indicate that the nonlinear time axis has been triggers and shown to the user for interaction with it. The simplest way is to display a text notification such as "nonlinear time axis has been triggered". In one embodiment, the height of time axis will resemble the temporal resolution such that regions allowing fine sliding on time will see a higher/wider time axis. In another embodiment, the font displaying time(s) will change for different time instants or locations on the time axis. For example, the font size can be proportional to (inverse) temporal resolution resembling the different temporal resolution. In yet another embodiment, the text displaying time instants will be at different heights resembling the different temporal resolution. In yet another embodiment, a gradient of color or pattern will be used on the

time axis, of uniform or different height, to resemble the different temporal resolution (color-coded or colorbar to indicate temporal resolution). In yet another embodiment, different time instants will be displayed to directly indicate different time steps which are resulted from different temporal resolution, e.g., [1:23, 1:27, 1:29, 1:30, 1:31, 1:33, 1:37]. In yet another embodiment, a carousel preview of the media entity (e.g., frames, music notes, etc.) is shown and the size, shape, translation, rotation, distance of preview pieces resemble the different temporal resolution. For example, two close preview pieces mean coarse temporal resolution while two far apart preview pieces mean fine temporal resolution. In yet another embodiment, the text/preview transparency level of the time axis overlaying the video will resemble different temporal resolutions, e.g., fading away when going outward from the focus point. The transparency level of the time bar can be adjusted by alpha composition in different regions of the time bar. All the visualization methods mentioned here can be used individually or as a combination.

[001 118] Figure 72 Example display of non-linear axis

[001 119] In another embodiment, more than 1 time axes can be presented to user simultaneously; these axes will have different starting and ending timestamps and temporal resolutions. For instance, axis 1 covers 0h0m0s-2h0m0s, with course temporal resolution; axis 2 covers 0h13m0s-0h25m0s, with moderate temporal resolution. The starting and ending point of time axes with finer resolution can be user defined, or user activated. In one embodiment, the starting and ending point of time axes with finer resolution is centered around focus point or features. In one aspect, some of the time axes can be displayed as a curved surface, or circle, instead of a straight line, as shown in figure 73.

[001 120] Figure 73 another Example display. Axes 1 and 2 have different temporal resolution; the starting and ending timestamps of time axis 2 is shown as the intersections with time axis 1. Axes 1 and 2 may be linear or nonlinear.

[001 121] It should be appreciated that the presentation ways does not limit to by visual ways. Sound, vibration, tactile, and other user interfaces can also be used. In one embodiment, play a sound, e.g., a chime, to notify that the nonlinear time axis is trigger, and then play beeps of different frequencies to resemble the different temporal resolutions, like the beep used by cars when back off against a wall (the beep frequency changes based on the distance to the wall).

[001 122] 16.4 Input without using pointing-based device, but swiping, rotating, hand gesture, etc.

[001 123] It should be appreciated that the rewind/forward along time axis does not have to be achieved by changing the focus point using pointing-based or 2-D coordinate-based inputs, such as a mouse or a touch screen. In some other types of inputs, the user can control the movement along the time axis without using or change the focus point. Such inputs could include the scroll wheel in a mouse.

[001 124] In one embodiment, the input device can be the keys on a keyboard, e.g. (pressing left arrow once means rewind and pressing right arrow once means forward). In another embodiment, the input device can be a multitouch touchpad that swiping to the left with two fingers means rewind while swiping to the right with two fingers means forward. In yet another embodiment, the input device is a rotary/touchable ring, that clockwise swiping or rotating means rewind while counter-clockwise swiping or rotating means forward. In yet another embodiment, a left-pointing hand gesture (captured by a camera, a radar - like Google Project Silo, a ultrasonic sensor, etc.) means rewind while a right-pointing hand gesture means forward. This embodiment can be used in game consoles, virtual reality or augmented reality handsets, etc.

[001 125] It should be further appreciated that when using these kinds of inputs, the nonlinear mapping may also jointly consider factors such as swiping or rotating speed. (Maybe we don't need this sentence)

[001 126] 16.5 Irregular display and UI

[001 127] In most computing systems, the display is a rectangular, the media player is also a rectangular window, and the time axis is a straight line parallel to one edge of the display. It should be appreciated that the user interface or computer display does not have to be in the shape of a rectangular, nor should the time axis be a straight and/or orthogonal line. For example, in a round-shape smartwatch, the time axis can be an arc centered at the center of the round-shape display. In another embodiment, the time axis can be a spiral. The user can control the media playback temporally by interacting with the arc or the spiral. For example, he/she can drag the playhead along the arc or the spiral.

[001 128] 16.7 Non-temporal advance in software, webpages, messages and documents

[001 129] It should be appreciated that our disclosed subject matter is not limited to rewind and forward on time axis for media playback. It can be used to enable scrolling and advancing in many different applications. This can be extremely useful when nowadays that swiping is a major interaction between users and mobile devices. By nonlinear advance rate, two same swipes can have different advance rates based on the content being swiped over.

[001 130] In the software, webpages, messages and documents, the counterpart of temporal-to-spatial ratio in video player is defined as "advance rate" in the disclosed subject matter, which is the ratio between the changing rate of elements (e.g., pages for books, pages for documents, lines in chat history, emails, etc.) being displayed to the amount of changes on a user input (e.g., pixels that the mouse dragged over, number of page-up or page-down key pressed, angles and/or speed that the mouse scrolling wheeling rotated, distances and/or speed that fingers swipes over a touchpad/touchscreen, etc.)

[001 131] In one embodiment, the nonlinear advance rate can be used for browsing a document (widely defined, in a word processing program, in a document reader program, in a web browser showing a webpage, or in a eBook reader like in our Figure 16.1). There is an internal structure of the document, such as headings at different levels. Higher advance rates should be allowed when browsing over lower level of content (such as the body text) while lower advance rates will be applied for headings. In this way, the users can quickly skim over body text (usually lengthy) while being able to precisely stop at the beginning of each heading. Heading level and advance rate is related. In one case, higher heading level (big and top, like chapter in books or <h1> tag in HTML) maps to slower advance rate while lower heading level (small and down, like subsection in books <h5> tag in HTML) maps to faster advance rate. In another case, some parts (such as references, homework and advanced topics in a textbook) will be given higher advancing rate while others (such as basic topics) lower. There are many features that can determine the nonlinear advance rate, such as heading, headers, titles, sections, footnotes, references, homeworks, glossary, user history, user data, data analytics from other users, whether the page has been read previously by user or not, annotations, tags, recommendations, comments.

[001 132] In another embodiment, the nonlinear advance rate can be used for browsing a document, webpage or software comprising financial statements or financial

data. There are many features that can determine the nonlinear advance rate, such as amount, debit/credit, timestamp, party involved in the transaction/entry, user history, user data, data analytics from other users, whether the page has been read previously by user or not, frequency of similar transaction/entry.

[001 133] In one embodiment, the nonlinear advance rate can be used for browsing over chatting history (Google Hangouts, Facebook Messenger, WhatsApp, Wechat, etc.), especially on mobile devices. The advance rate can be nonlinearly related to the properties of the chatting text. The properties can be of many kinds. In one case, when swiping the history of chatting, chats happened in a short period of time can be skimmed over quicker than chats spanning over a long period of time. In another case, when swiping the history of chatting, chats about similar topics/phrases can be skimmed over quicker than those about different topics. The software may allow slower swiping between two periods or two topics in the two cases respectively. All methods, especially temporal distance and text segmentation methods mentioned above or below, can be used to here.

[001 134] In one embodiment, the nonlinear advance rate can be used for browsing over news feeds in a social media platform (facebook, twitter, instagram, etc.), especially on mobile devices. The advance rate can be nonlinearly related to the properties of the item in news feed. The properties can be of many kinds. In one case, when swiping the news feeds, items from closely communicated friends will have slower advance rate while items from public accounts will have faster advance rate. In another case, when swiping the news feeds, items of lower relevance will have slower advance rate while items of higher relevance will have faster advance rate.

[001 135] In another embodiment, the nonlinear advance rate can be used for browsing over a contact list (or anything equivalent, such as a log of phone calls, a friend list on a social network, see our Figure 16.2), especially on a mobile device. In one case, frequent contacts, or numbers called, or friends commented, will be given slower advance rate while infrequent contacts will be given faster advance rate. In this way, the user can quickly skim over infrequent contacts while being able to precisely stop at important contacts.

[001 136] In another embodiment, the nonlinear advance rate can be used for browsing over a plurality of emails, especially on a mobile device. In one case, emails

that marked as important will be given slower advance rate while unimportant emails will be given faster advance rate. In this way, the user can quickly skim over unimportant emails while being able to precisely stop at important email. There are various factors and ways to determine whether an email is important, e.g., based on the sender, based on how many recipients, based on the text in the email, etc. There are many features that can determine the nonlinear advance rate, at least one of the said features is a feature selected from the group comprising of: the contact frequency between the email receiver and sender, whether the receiver is a recipient or a cc'ed recipient of the email, the importance of the email, the timestamp of the email, the length of the email, the domain name of the email sender, the subject of the email, the signature of the sender, whether the email is the initial email, whether the email is the reply email, whether the email is the forwarded email, the number of recipients of email, user data, user history, how frequent does the user reply to the previous emails of the sender, how soon do the user reply to the previous emails of the sender after the user saw the said emails, the length of the previous email communications between the receiver and the user. The nonlinear navigation can be implemented at both email list level and within individual emails.

[001 137] In another embodiment, the nonlinear advance rate can be used for browsing a list of search results, (widely defined, such as a list of products in shopping website, a list of movies in a movie distribution platform, a list of web search results, a list of geographical information related items like restaurants near by), especially on a mobile device. For explanation sake, we first introduce a term "aggregation index," meaning how close elements within a group are with each other. The definition of aggregation index varies for different types of information. For restaurants, in one case, those geographically close have higher aggregation index while those far away have lower aggregation index. For products, in one case, those of similar ratings and prices have higher aggregation index while those of quite different ratings and prices have lower aggregation index. In the end, the advance rate will be related to the aggregation rate. In one case, items of high aggregation rate will be given faster advance rate to allow quick skimming of similar items, while items of low aggregation rate will be given slower advance rate.

[001 138] It should be appreciated that the nonlinear navigation approach can be used to navigate and visualize numerous types of software/documents. Some of the

possible application for nonlinear navigation is listed as follows: list of media files, webpage, search results, web history, web browser bookmarks, text document, word processor, text editor, image, list of images, file browser, phone contact list, ebook, audio, audiobook, email, email list, text messenger, phone call history, digital music, karaoke song list, karaoke soundtrack, podcast, computer-aided design, application software, video games, computer games, smartphone applications, video game replay, calendar, podcast, radio, voice memo, phone voicemail, virtual reality content, augmented reality content, financial data, stock indexes, security prices, financial transaction data, financial statements, balance sheet, income statement, statement of changes in equity, cash flow statement, physiological data, medical data, medical sensor data, vital sign data, electrophysiological data, medical images, medical lab analysis data, industrial data, security data, military data, etc.

[001 139] 17. Hardware implementation

[001 140] The disclosed subject matter can be implemented by software running on a plurality of computer architectures. In one embodiment, the software can run on a computer with an operating system and the media entity is a file. Such computer includes but is not limited to, a desktop or laptop running Windows/Linux/Mac OS, a smartphone or tablet running iOS/Android/Windows/Blackberry OS that mainly uses a touchscreen and microphone as user input. The search is done using the computing power of the computer itself. In another embodiment, the software is an application (short as "app") that runs on a computer while the media entity is hosted on a remote server. ChromeBook, a smartphone or tablet running iOS/Android/Windows/Blackberry OS that mainly uses a touchscreen and microphone as user input are examples of this computer. The search, ranking and other computations to generate the results are mainly done by the remote server, although some data to help the search such as buffer or index can be stored locally. Queries and results are sent between the computer and the remote server. In yet another embodiment, the software is a web browser that can play media and provide user a search interface, the media entity is on a remote server, and the computer can be of any of the forms above. The search, rank-ing and other computations to generate the results are mainly done by the remote server, although some data to help the search such as buffer or index can be stored locally. In yet another embodiment, unlike a general-purpose computer that can run all kinds of programs, the hardware only serves a special function and users can

only access the limited functions/programs provided by it, such as a Karaoke machine, a TV stick or an Amazon Alexa. But there could be a general-purpose computer implementing this hard-ware or a special purpose chip. The user can send queries for either local or remote media, using interface provided by this computer or an external device, e.g, a mobile app that connects to the computer or a re-mote control. The search results and media can be fetched locally or transmitted from remote end.

[001 141] Herein software refers to any program running on any computing devices. But it can also be implemented substantially in hardware, either analog or digital, such as DVD players, Blu-Ray player, media streaming stick, set top box, smart TV, music players, smartphones, tablet computers, wearable electronic devices, computer expansion cards or gadgets, USB peripherals devices, Bluetooth devices, WIFI routers, extenders, dongles and hotspots, audio systems, home theaters, on vehicle media systems (on car, on plane, on robot, on rocket and on ship), portable media players, game consoles, virtual reality displays, smart glasses, smart watches, projectors, monitors, desktops, smart audio player (e.g., Amazon Echo), etc.

[001 142] The hardware components that may be used to implement the disclosed subject matter comprises of central processing unit (CPU, including microcontrollers and soft cores on field programmable gate array, FPGA), memory (such as random access memory), storage (such as flash-based storage, hard disk), communication components (such as antennas), etc. Instead of using a CPU to implement the disclosed subject matter, a programmable logic device (such as FPGA and Complex Programmable Logic Devices, CPLD), a specifically made integrated circuit (Application Specific Integrated Circuit, ASIC), graphic processing unit (GPU) computing platform, and other hardware solutions, can also be used to implement our disclosed subject matter.

[001 143] In one embodiment, a wireless media player consists of a microcontroller (i.e., system-on-chip, SoC), with a co-processor for video content decoding. The co-processor can extract the audio track, captions and video frames using a hardware circuit independent from the CPU. The co-processor fetches media stream from the microcontroller, which fetches media stream from the WiFi or cellular network wirelessly. There co-processor and the microcontroller may communicate via a PCI-E interface, for both media stream and control signals. The co-processor has direct HDMI interface or other video ports to output the content to any HDMI-compatible display. A

mobile application software communicates with the microcontroller via Bluetooth to select the media to play and settings. In an alternative embodiment, the media decoding is done by a built-in component of the microcontroller (such as the ARM NEON graphics engine in ARM Cortex A-series) and the caption adding/rendering/generating is done by the CPU.

[001 144] In one embodiment, the hardware implementation comprises communication modules. Communication protocol such as WiFi, Bluetooth, LAN, near-field communication (NFC), infrared communication, radio-frequency communication, TV cable, satellite TV communication, telephone communication, cellular network, 3G network, 4G network can be enabled.

[001 145] In one embodiment, the hardware implementation is media capable. The hardware implementation can comprises sound card, microphone, speakers and audio power amplifier.

[001 146] In another embodiment, the hardware implementation is wearable by the user. The hardware implementation comprises wearable display (eg. head-mounted display) and a media-capable computing device. The karaoke machine may further comprise earphones/speakers, microphone, additional sensors and communication modules.

[001 147] It should be appreciated any combination of software features enabled by aforementioned hard-ware implementation would satisfy the disclosed subject matter.

[001 148] Usage of Terms

[001 149] As used in this application, "component," "module," "system", "interface", and/or the like are generally intended to refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution. For example, a component may be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration, both an application running on a controller and the controller can be a component. One or more components may

reside within a process and/or thread of execution and a component may be localized on one computer and/or distributed between two or more computers.

[001 150] Unless specified otherwise, "first," "second," and/or the like are not intended to imply a temporal aspect, a spatial aspect, an ordering, etc. Rather, such terms are merely used as identifiers, names, etc. for features, elements, items, etc. For example, a first object and a second object generally correspond to object A and object B or two different or two identical objects or the same object.

[001 151] Moreover, "example" is used herein to mean serving as an instance, illustration, etc., and not necessarily as advantageous. As used herein, "or" is intended to mean an inclusive "or" rather than an exclusive "or". In addition, "a" and "an" as used in this application are generally be construed to mean "one or more" unless specified otherwise or clear from context to be directed to a singular form. Also, at least one of A and B and/or the like generally means A or B or both A and B. Furthermore, to the extent that "includes", "having", "has", "with", and/or variants thereof are used in either the detailed description or the claims, such terms are intended to be inclusive in a manner similar to the term "comprising".

[001 152] Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing at least some of the claims.

[001 153] Furthermore, the claimed subject matter may be implemented as a method, apparatus, or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof to control a computer to implement the disclosed subject matter. The term "article of manufacture" as used herein is intended to encompass a computer program accessible from any computer-readable device, carrier, or media. Of course, many modifications may be made to this configuration without departing from the scope or spirit of the claimed subject matter.

[001 154] Various operations of embodiments are provided herein. In an embodiment, one or more of the operations described may constitute computer readable instructions stored on one or more computer and/or machine readable media,

which if executed will cause the operations to be performed. The order in which some or all of the operations are described should not be construed as to imply that these operations are necessarily order dependent. Alternative ordering will be appreciated by one skilled in the art having the benefit of this description. Further, it will be understood that not all operations are necessarily present in each embodiment provided herein. Also, it will be understood that not all operations are necessary in some embodiments.

[001 155] Also, although the disclosure has been shown and described with respect to one or more implementations, equivalent alterations and modifications will occur to others skilled in the art based upon a reading and understanding of this specification and the annexed drawings. The disclosure includes all such modifications and alterations and is limited only by the scope of the following claims. In particular regard to the various functions performed by the above described components (e.g., elements, resources, etc.), the terms used to describe such components are intended to correspond, unless otherwise indicated, to any component which performs the specified function of the described component (e.g., that is functionally equivalent), even though not structurally equivalent to the disclosed structure. In addition, while a particular feature of the disclosure may have been disclosed with respect to only one of several implementations, such feature may be combined with one or more other features of the other implementations as may be desired and advantageous for any given or particular application.

CLAIMS

What is claimed is:

1. A computer-implemented method for searching media, comprising:
receiving a query, comprising a first term, for the media;
identifying a first result and a second result in time-associated information of the media based upon a determination that the first result comprises a first match of the first term and the second result comprises a second match of the first term; and
providing the first result and the second result based upon a first temporal property of the first match of the first term in the first result and a second temporal property of the second match of the first term in the second result.
2. The computer-implemented method of claim 1, the providing comprising providing the first result and the second result responsive to determining that the first temporal property and the second temporal property exceed a threshold temporal property.
3. The computer-implemented method of claim 1, the providing comprising providing the first result in association with a higher rank than the second result based upon a comparison of the first temporal property with the second temporal property.
4. The computer-implemented method of claim 1,
the query comprising a second term;
the identifying comprising identifying the first result and the second result based upon a determination that the first result comprises a first match of the second term and the second result comprises a second match of the second term; and
the providing comprising providing the first result and the second result based upon a third temporal property of the first match of the second term in the first result and a fourth temporal property of the second match of the second term in the second result.
5. The computer-implemented method of claim 4, comprising:
determining a first temporal distance based upon the first temporal property and the third temporal property; and

determining a second temporal distance based upon the second temporal property and the fourth temporal property.

6. The computer-implemented method of claim 5, the providing comprising providing the first result and the second result responsive to determining that the first temporal distance and the second temporal distance are less than a threshold temporal distance.

7. The computer-implemented method of claim 6, comprising:
determining the threshold temporal distance based upon the query.

8. The computer-implemented method of claim 5, the providing comprising providing the first result in association with a higher rank than the second result responsive to determining that the first temporal distance is less than the second temporal distance.

9. The computer-implemented method of claim 5,
the first temporal distance corresponding to a difference between a first timestamp of the first match of the first term in the first result and a second timestamp of the first match of the second term in the first result; and
the second temporal distance corresponding to a difference between a third timestamp of the second match of the first term in the second result and a fourth timestamp of the second match of the second term in the second result.

10. The computer-implemented method of claim 1, comprising:
prior to the identifying, transcribing the media to generate a transcript, the time-associated information comprising the transcript.

11. The computer-implemented method of claim 10, comprising:
prior to the identifying, translating the transcript from a first language to generate a second transcript in a second language, the time-associated information comprising the second transcript.

12. The computer-implemented method of claim 1, the time-associated information comprising text information of the media.
13. The computer-implemented method of claim 1, the identifying performed using at least one of a brute-force search, a satisfiability check, a temporal sliding window, clustering, unsupervised machine learning, supervised machine learning, reinforcement learning, deep learning or pre-indexing.
14. The computer-implemented method of claim 1, the providing comprising providing for presentation the first result and the second result.
15. The computer-implemented method of claim 1, the media comprising a soundtrack, and the time-associated information comprising lyrics of the soundtrack, the method comprising:
 - responsive to receiving a selection of the first result, providing for presentation a karaoke presentation based upon the first result.
16. The computer-implemented method of claim 1, comprising:
 - generating a list of index keys corresponding to the query, the list of index keys comprising:
 - a first index key corresponding to the first result and associated with a first portion of the media; and
 - a second index key corresponding to the second result and associated with a second portion of the media; and
 - responsive to receiving a selection of the first index key, providing access to the first portion of the media.
17. The computer-implemented method of claim 16, comprising:
 - after providing access to the first portion, editing the first portion of the media.
18. The computer-implemented method of claim 1, the time-associated information comprising at least one of text, a transcript, audio, a soundtrack, an

image, a comment, a user annotation, a summary, a landmark, a tag, a cross-reference or a review.

19. The computer-implemented method of claim 1, the media comprising at least one of a video, audio, an image or a document.

20. The computer-implemented method of claim 1, comprising:
extracting, from the time-associated information, at least one of a landmarks, tags, summaries or cross-references.

21. A computer-implemented method for supplementing media with content, comprising:
segmenting the media into a first portion and a second portion based upon time-associated text information of the media;
analyzing the time-associated text information of the media to determine a first context for the first portion and a second context for the second portion;
selecting a first content from a plurality of contents for the first portion based upon the first context and a second content from the plurality of contents for the second portion based upon the second context; and
supplementing the first portion of the media with the first content and the second portion of the media with the second content.

22. The computer-implemented method of claim 21, the selecting performed responsive to determining that the first portion is content-compatible based upon the first context and that the second portion is content-compatible based upon the second context.

23. The computer-implemented method of claim 22, comprising:
analyzing the time-associated text information of the media to determine a third context for a third portion of the media; and
not supplementing the third portion of the media with content responsive to determining that the third portion is content-incompatible based upon the third context.

24. The computer-implemented method of claim 21, comprising:
selecting a first timestamp from a plurality of timestamps in the first portion based upon a first match between the first content and a portion of the time-associated text information associated with the first timestamp;
selecting a second timestamp from a plurality of timestamps in the second portion based upon a second match between the second content and a portion of the time-associated text information associated with the second timestamp; and
supplementing the first portion of the media with the first content at the first timestamp and the second portion of the media with the second content at the second timestamp.
25. The computer-implemented method of claim 21, the media comprising a video.
26. The computer-implemented method of claim 21, comprising:
providing for presentation, in a movie theater, the first portion of the media supplemented with the first content and the second portion of the media supplemented with the second content.
27. The computer-implemented method of claim 21, the media comprising live television, the first portion of the media comprising a first portion of the live television and the second portion of the media comprising a second portion of the live television, the method comprising:
providing for presentation the first portion of the live television supplemented with the first content and the second portion of the live television supplemented with the second content.
28. The computer-implemented method of claim 21, the media comprising an educational lecture, the first portion of the media comprising a first portion of the educational lecture and the second portion of the media comprising a second portion of the educational lecture, the method comprising:
providing for presentation the first portion of the educational lecture supplemented with the first content and the second portion of the educational lecture supplemented with the second content.

29. The computer-implemented method of claim 21, comprising:
prior to the analyzing, transcribing the media to generate a transcript, the time-associated text information comprising the transcript.
30. The computer-implemented method of claim 21, the media comprising at least one of virtual reality content or augmented reality content.
31. A computer-implemented method for supplementing a video with content, comprising:
selecting a first content from a plurality of contents for the video;
selecting a first area from a plurality of areas in the video based upon image analysis of the video; and
supplementing the video with the first content at the first area.
32. The computer-implemented method of claim 31, the selecting the first area performed responsive to determining, based upon the image analysis, that the first area has a focus below a focus threshold.
33. The computer-implemented method of claim 31, the selecting the first area performed responsive to determining, based upon the image analysis, that the first area comprises a first image feature.
34. The computer-implemented method of claim 31, the selecting the first area performed responsive to determining, based upon the image analysis, that the first area does not comprise a representation of a face.
35. The computer-implemented method of claim 31, the selecting the first area performed responsive to determining, based upon the image analysis, that the first area has a texture below a texture threshold.
36. The computer-implemented method of claim 31, the selecting the first area performed responsive to determining, based upon the image analysis, that the first area has motion below a motion threshold.

37. A computer-implemented method for supplementing a video with content, comprising:
- selecting a first content from a plurality of contents for the video;
 - supplementing the video with the first content; and
 - adjusting one or more properties of the first content based upon image analysis of the video.
38. The computer-implemented method of claim 37, the adjusting comprising adjusting a color of the first content based upon the image analysis.
39. The computer-implemented method of claim 37, the adjusting comprising adjusting a transparency of the first content based upon the image analysis.
40. The computer-implemented method of claim 37, the adjusting comprising adjusting a size of the first content based upon the image analysis.
41. The computer-implemented method of claim 37, the adjusting comprising adjusting a duration of the first content based upon the image analysis.
42. A computer-implemented method, comprising:
- receiving a request to implement a performance with a first user and a second user;
 - determine that the first user is associated with a first type of participation in the performance;
 - determine that the second user is associated with a second type of participation in the performance;
 - selecting a first content from a plurality of contents for the first user based upon the first type of participation and a second content from the plurality of contents for the second user based upon the second type of participation;
 - providing the first content to the first user and the second content to the second user;
 - receiving a first signal from the first user in association with the performance and a second signal from the second user in association with the performance; and

generating a representation of the performance based upon a combination of at least three of the first signal, the second signal, the first content or the second content.

43. The computer-implemented method of claim 42, at least one of the first content or the second content comprising a soundtrack, the first signal comprising an acoustic signal of the first user, the second signal comprising a visual signal of the second user.

44. The computer-implemented method of claim 42, the first type of participation corresponding to singing, the second type of participation corresponding to dancing.

45. The computer-implemented method of claim 42, the first user at a different geographical location than the second user.

46. A computer-implemented method for navigating through media, comprising:
receiving a request to move a control along a first axis from a first portion of the first axis to a second portion of the first axis;

responsive to determining that the control is being moved along the first axis within the first portion, navigating through the media at a first rate of advancement based upon a first feature of the first portion; and

responsive to determining that the control is being moved along the first axis within the second portion, navigating through the media at a second rate of advancement based upon a second feature of the second portion, the first rate of advancement different than the second rate of advancement.

47. The computer-implemented method of claim 46, the media comprising at least one of a video, audio, an image, a document or an application interface.

48. The computer-implemented method of claim 46,
the media comprising a list of contacts; and
at least one of the first feature or the second feature comprising a frequency of contact between a user and a contact in the list of contacts.

49. The computer-implemented method of claim 46,
the media comprising a list of messages; and
at least one of the first feature or the second feature comprising a feature of a message in the list of messages.
50. The computer-implemented method of claim 46, at least one of the first feature or the second feature determined based upon information, of the media, comprising at least one of text, an image, audio, comments, tags, titles, a transcript, cross-references, data analytics from a plurality of users, recommendations, reviews or user history.
51. The computer-implemented method of claim 46, the first feature corresponding to a first distance of a focus point from the first portion and the second feature corresponding to a second distance of the focus point from the second portion.
52. The computer-implemented method of claim 46,
the first axis corresponding to time; and
the rate of advancement comprising temporal resolution.
53. The computer-implemented method of claim 46, comprising:
providing for presentation a representation of the moving of the control along a representation of the first axis.
54. The computer-implemented method of claim 1,
the query based upon a first context of a first content selected from a plurality of contents; and
the method comprising supplementing, with the first content, a first portion of the media associated with the first result.

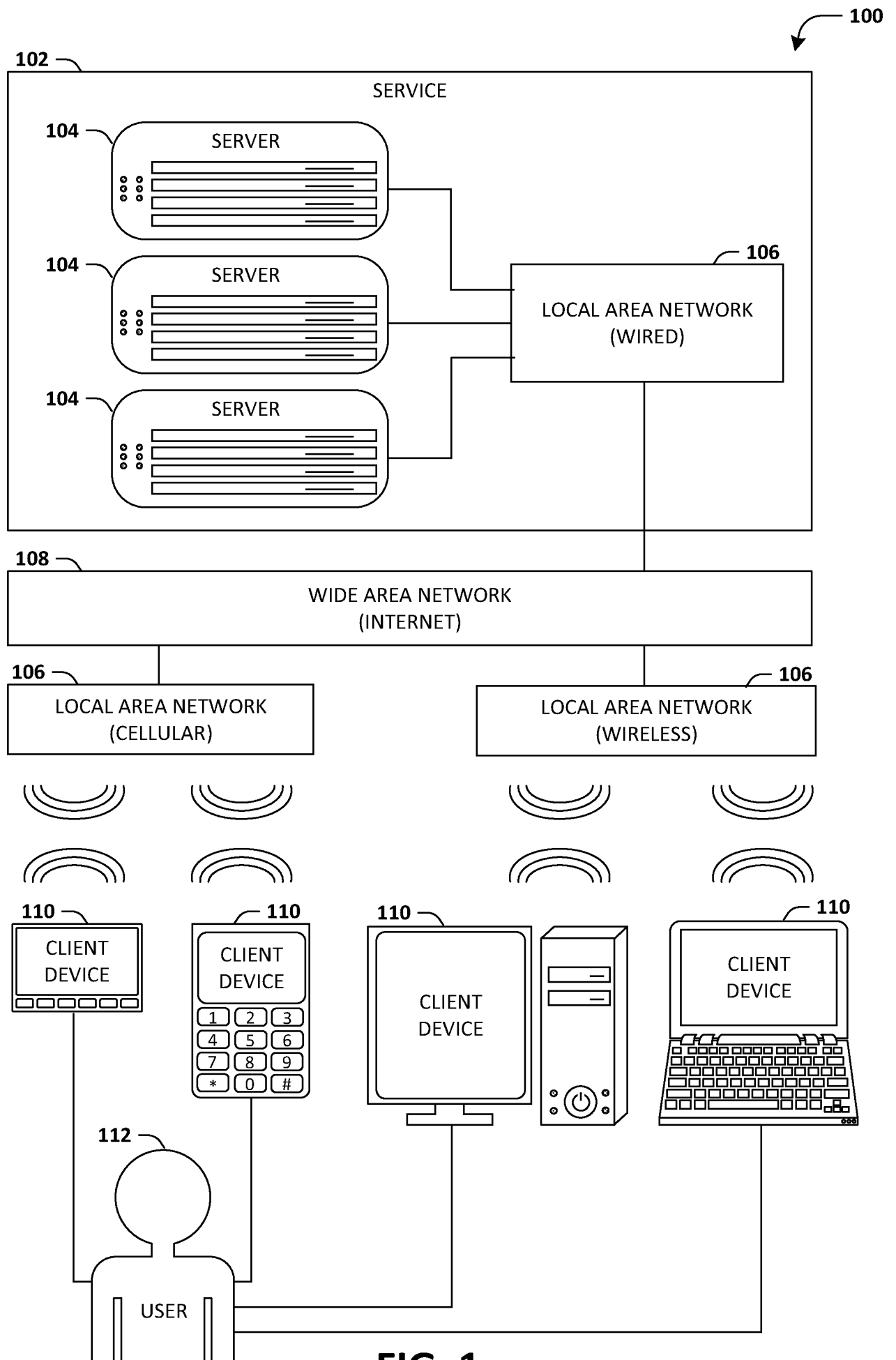
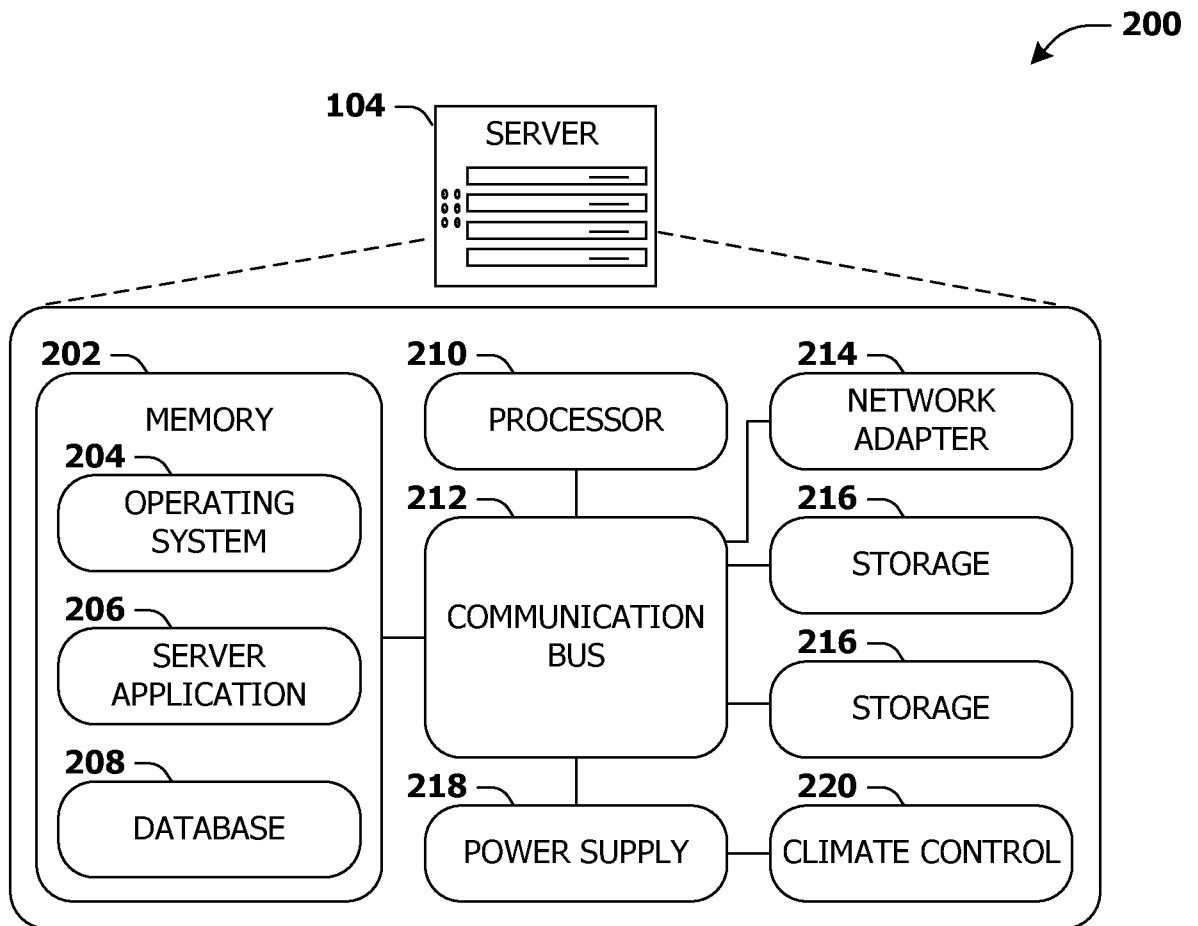
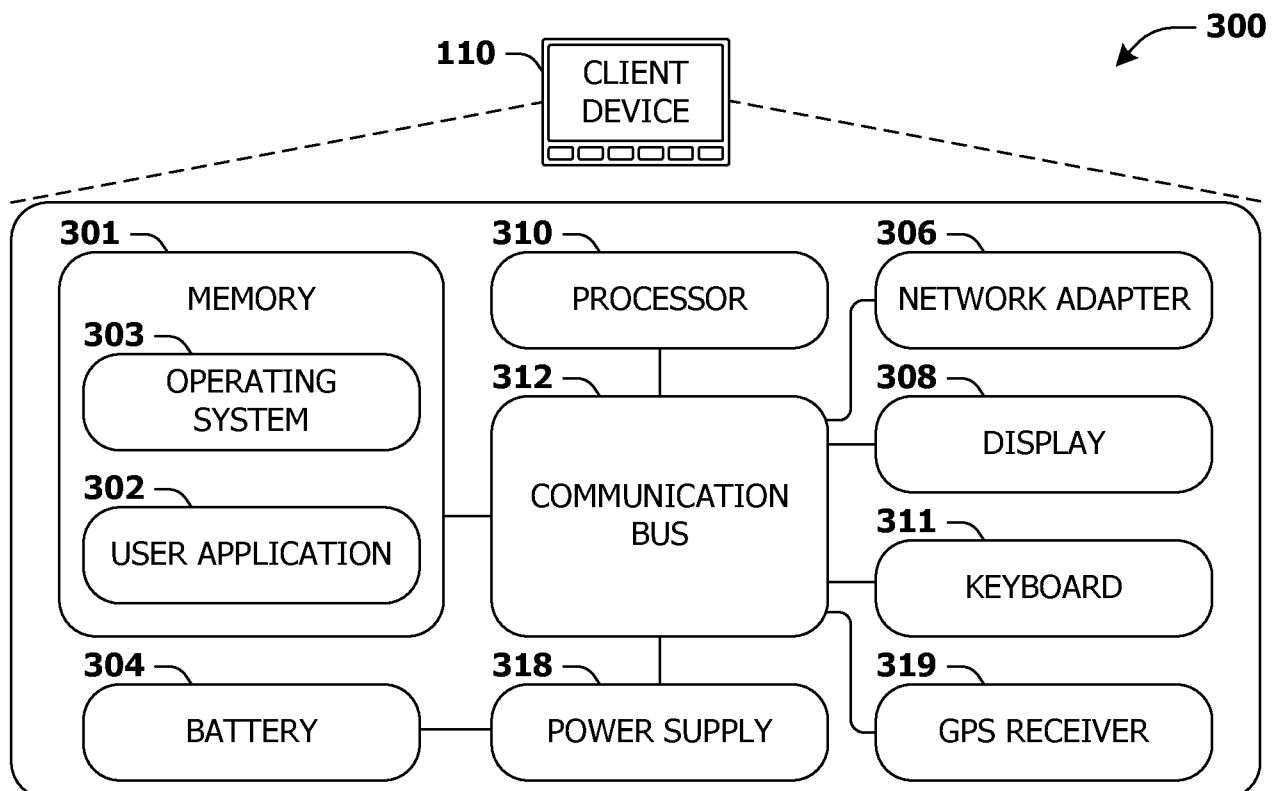
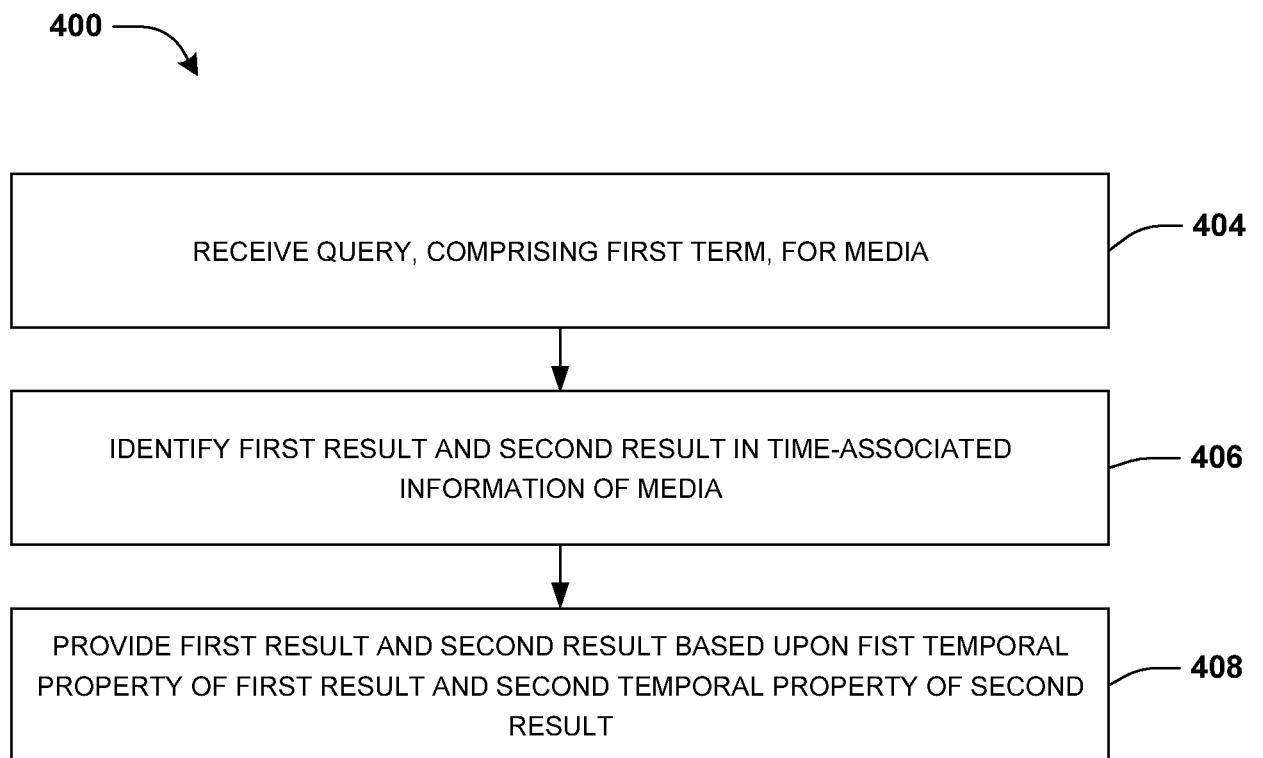
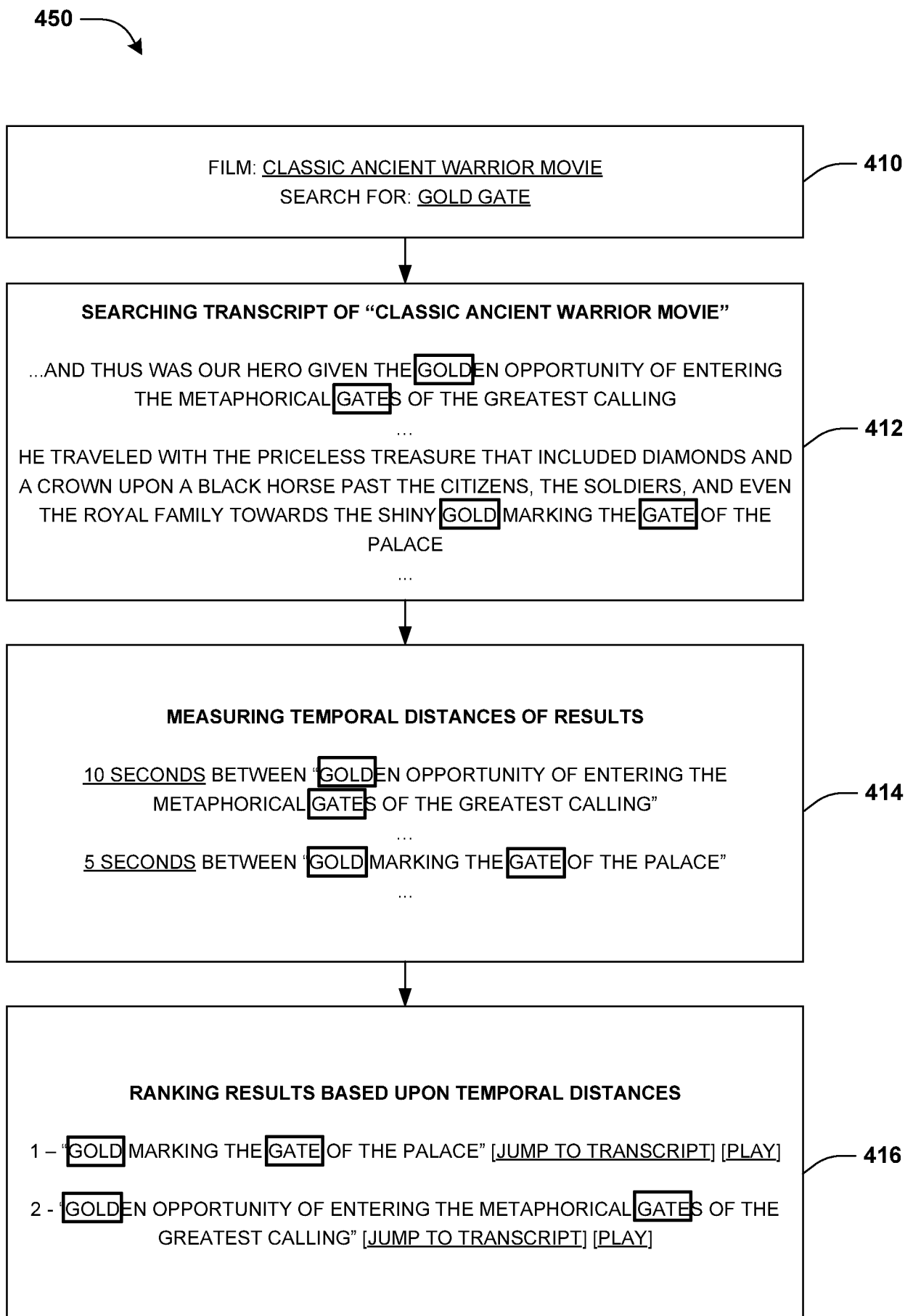
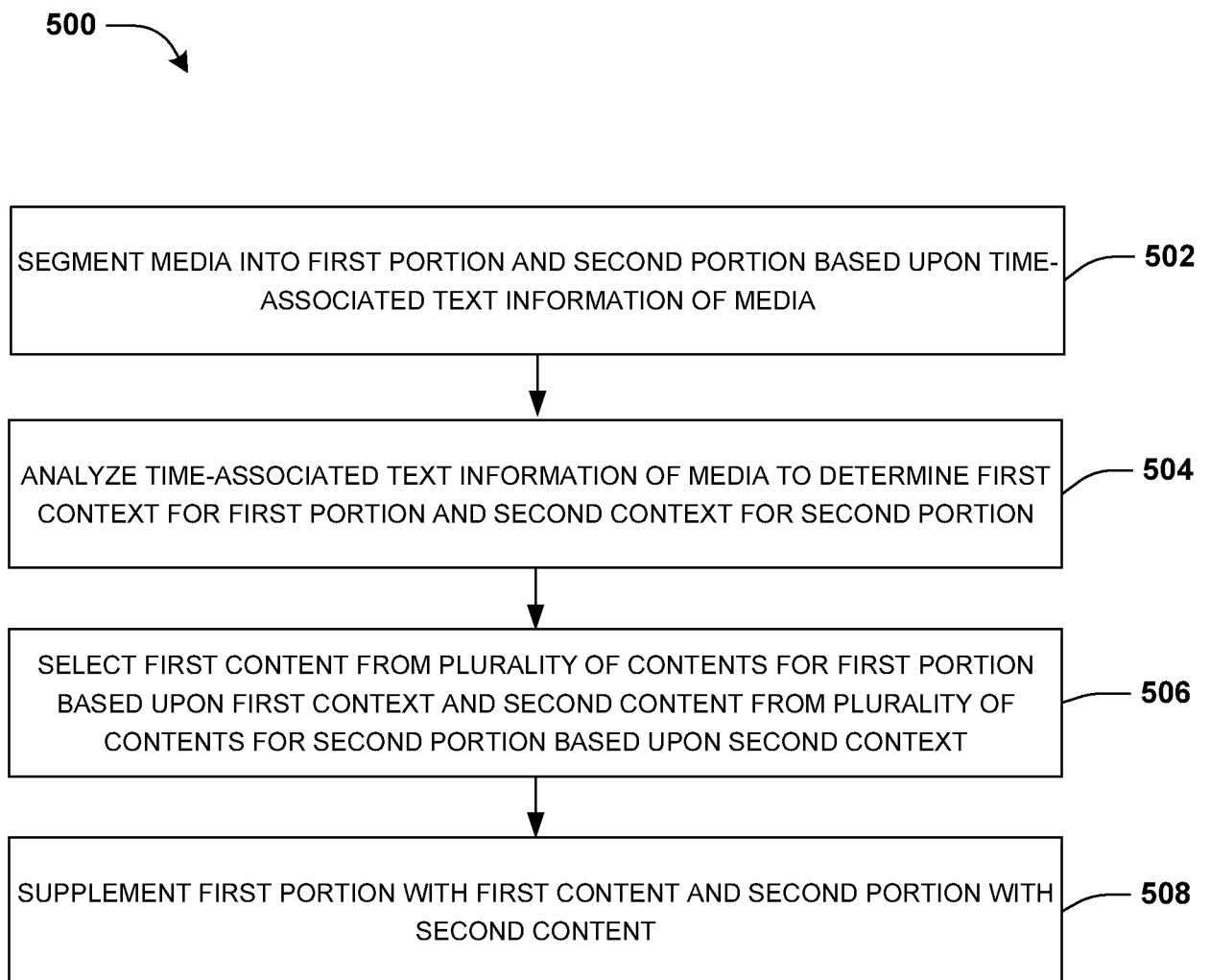


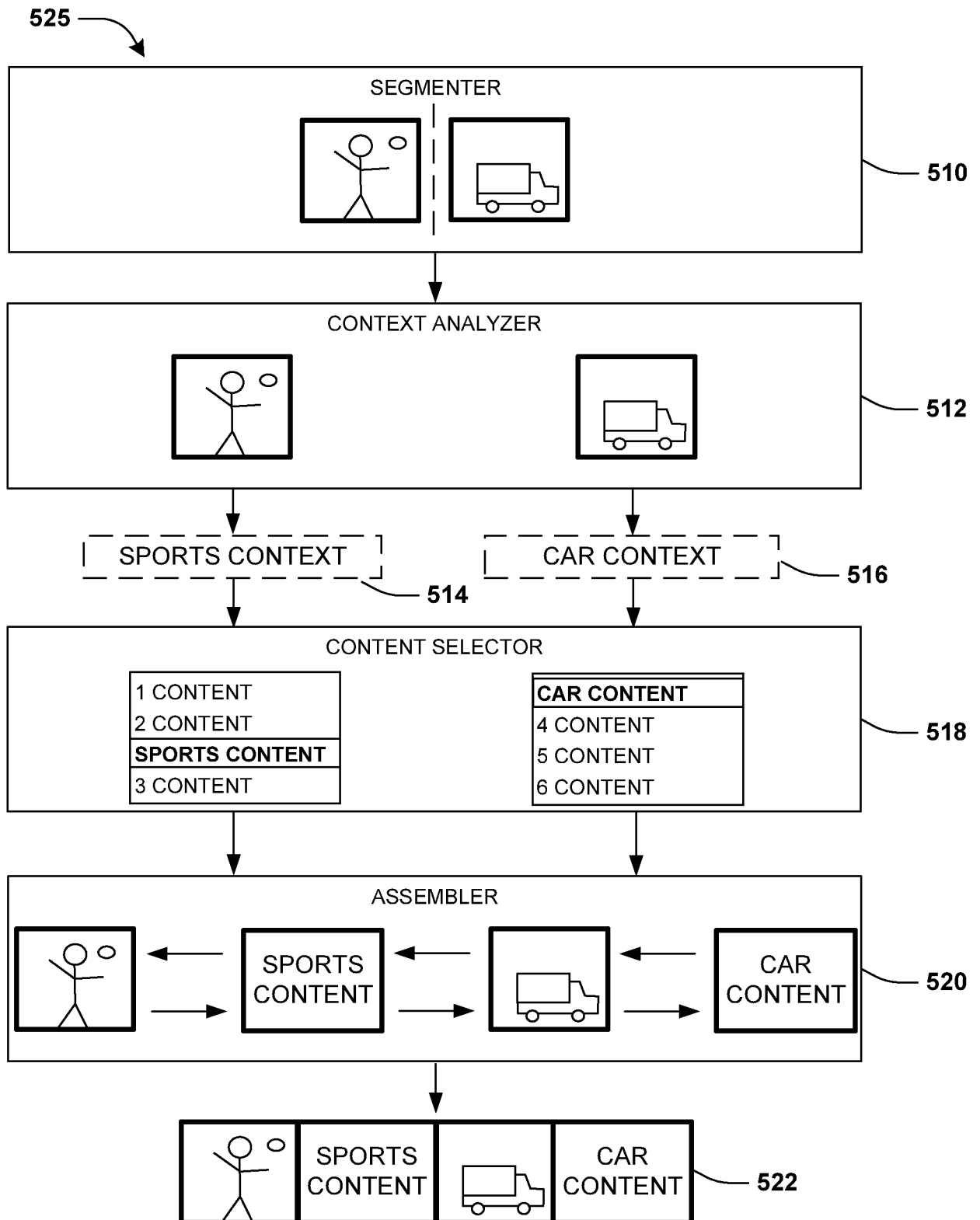
FIG. 1

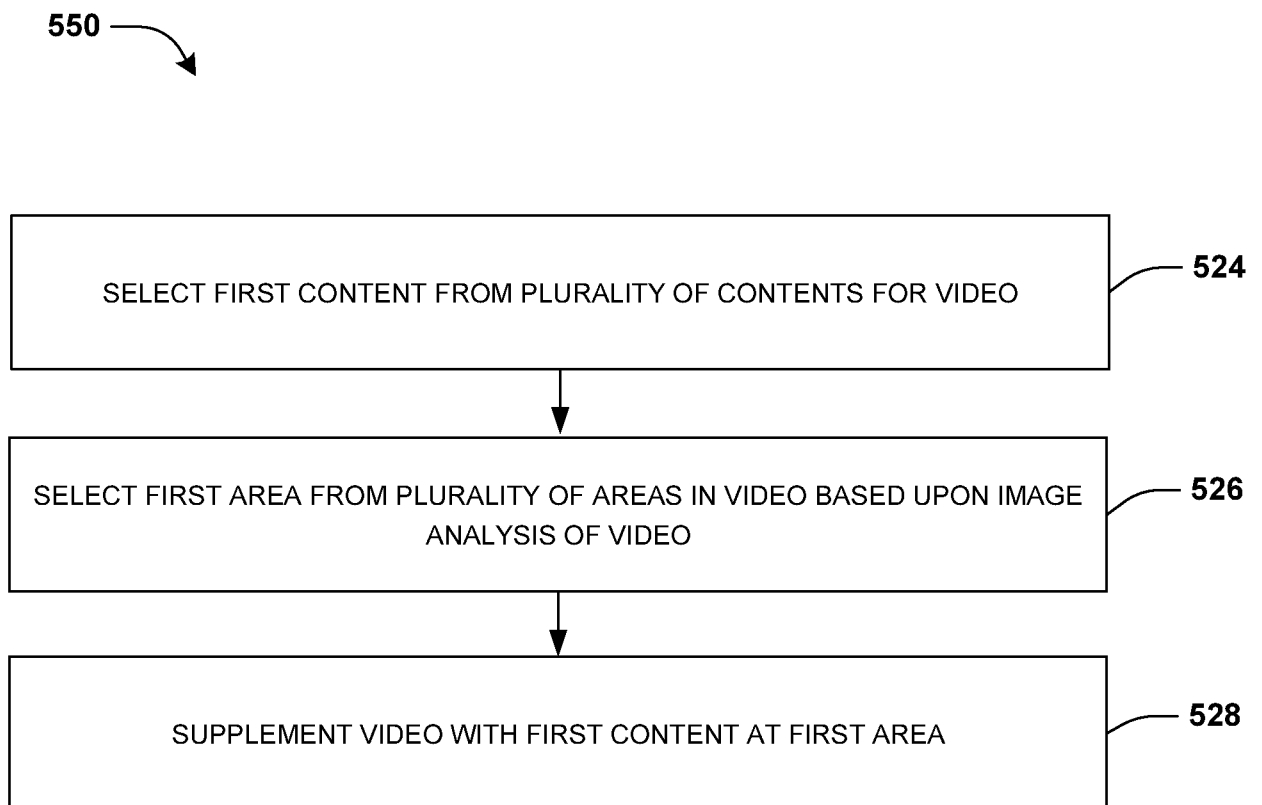
**FIG. 2****FIG. 3**

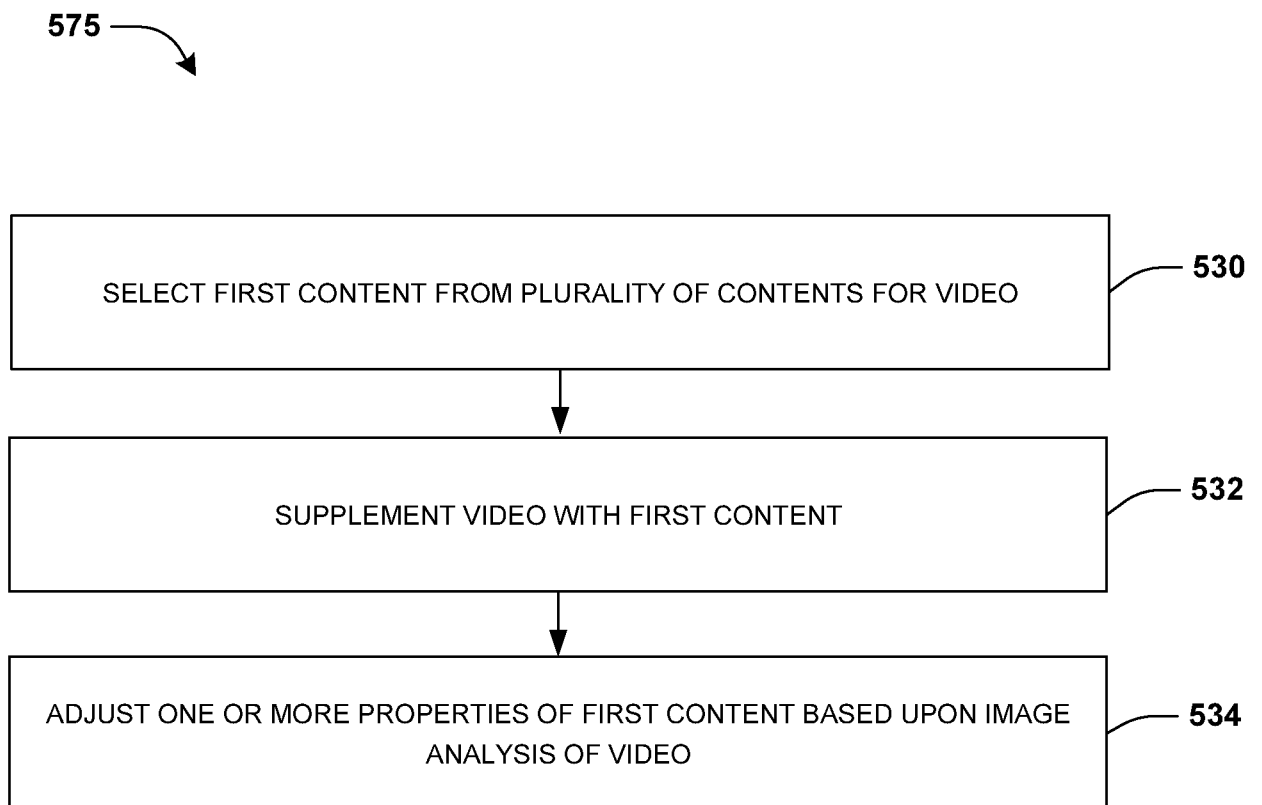
**FIG. 4A**

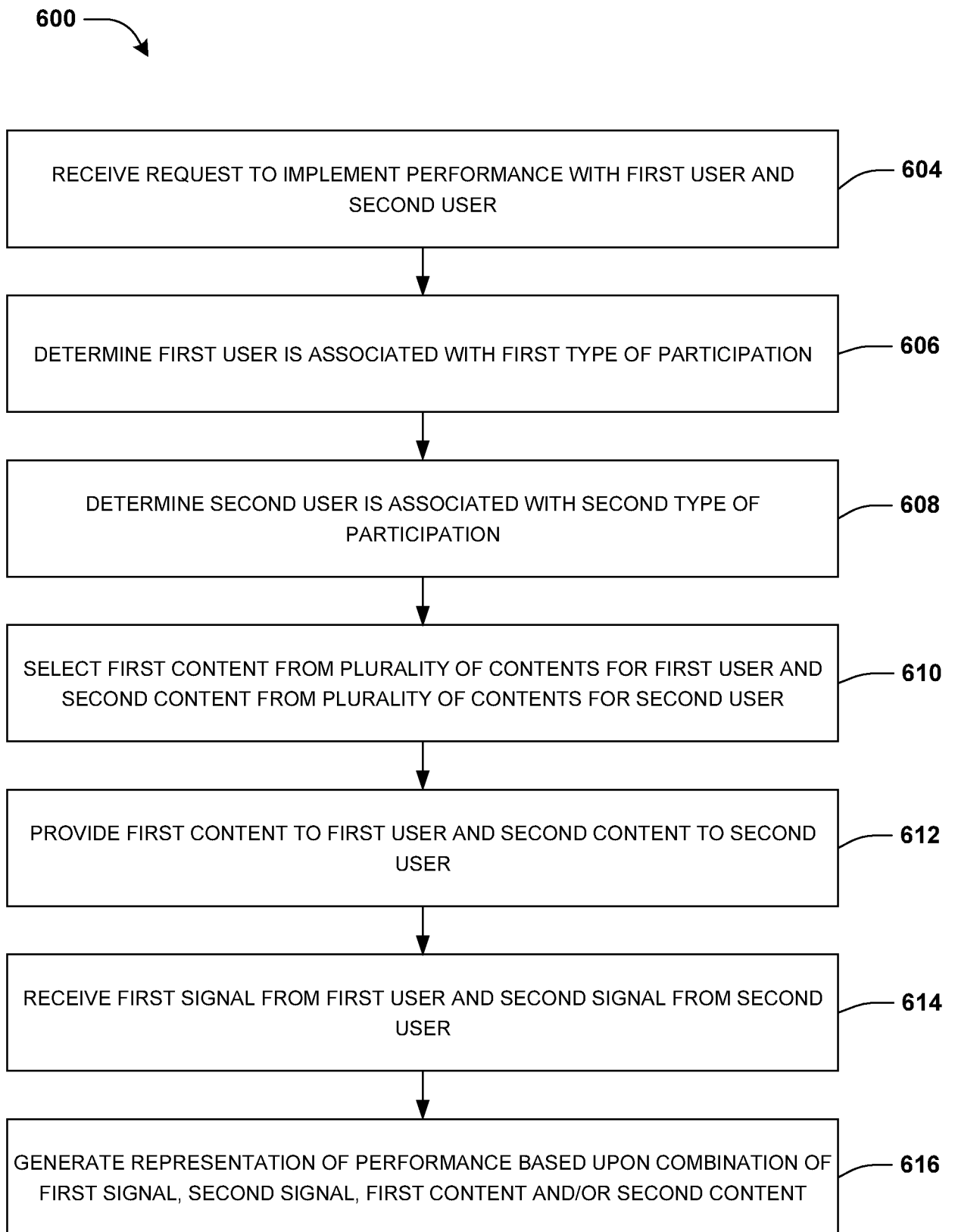
**FIG. 4B**

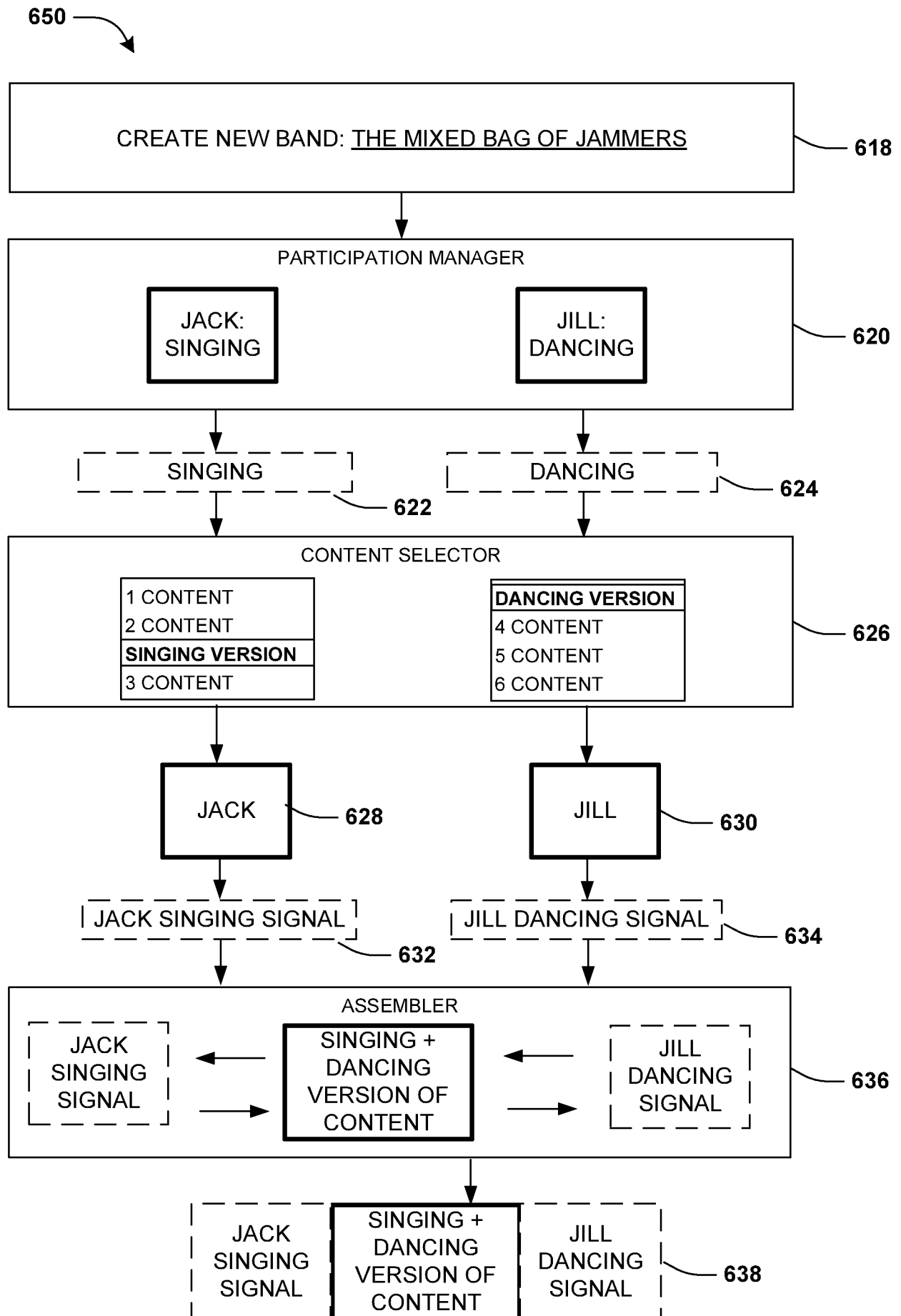
**FIG. 5A**

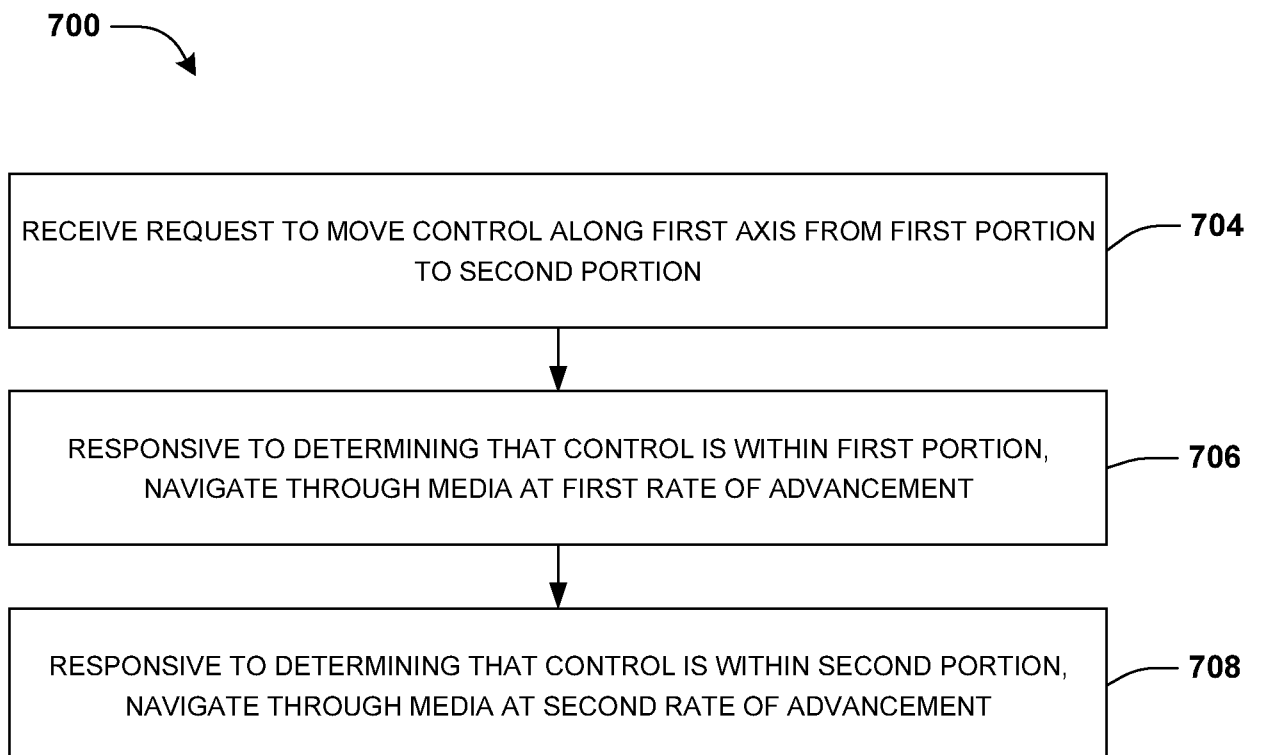
**FIG. 5B**

**FIG. 5C**

**FIG. 5D**

**FIG. 6A**

**FIG. 6B**

**FIG. 7A**

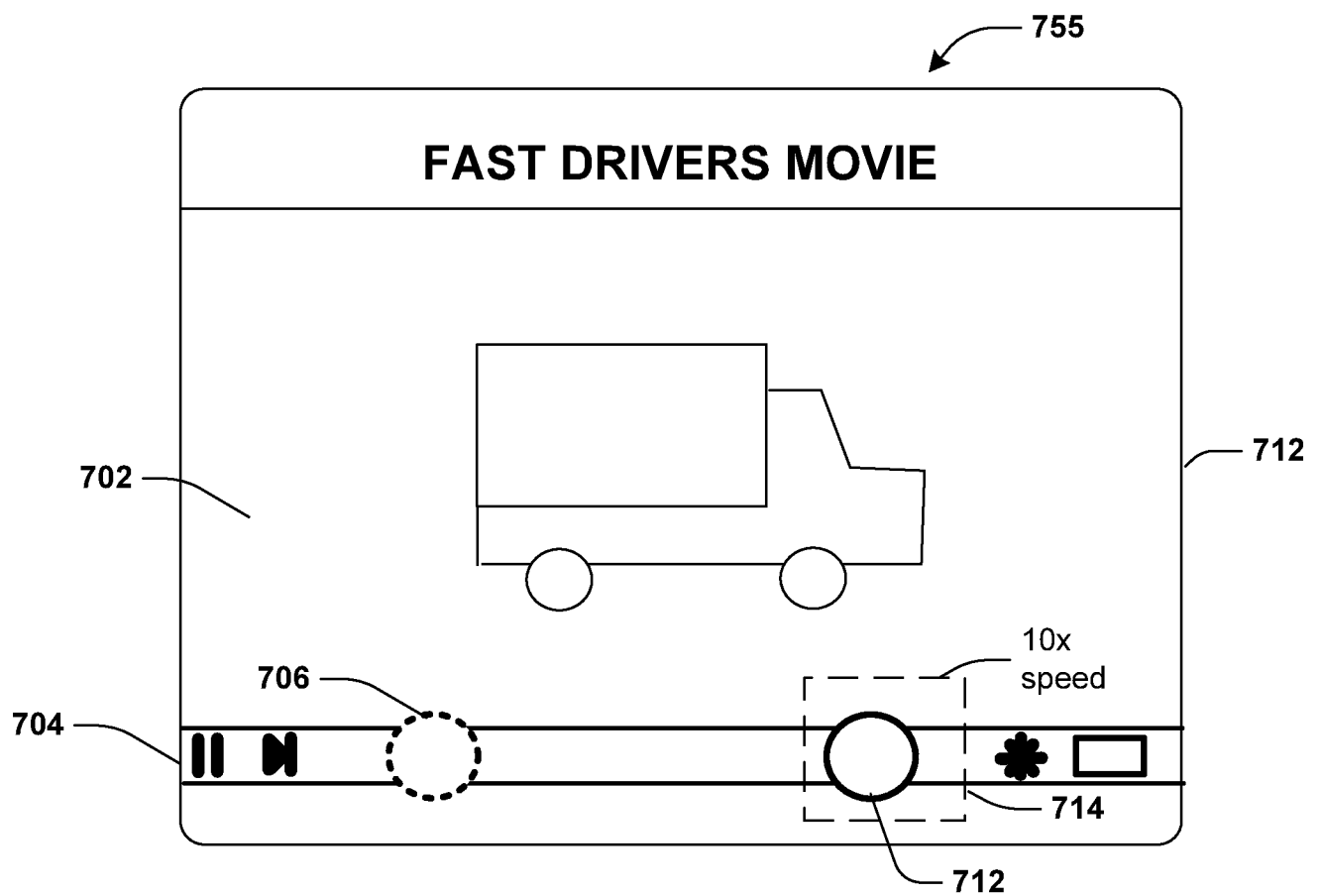
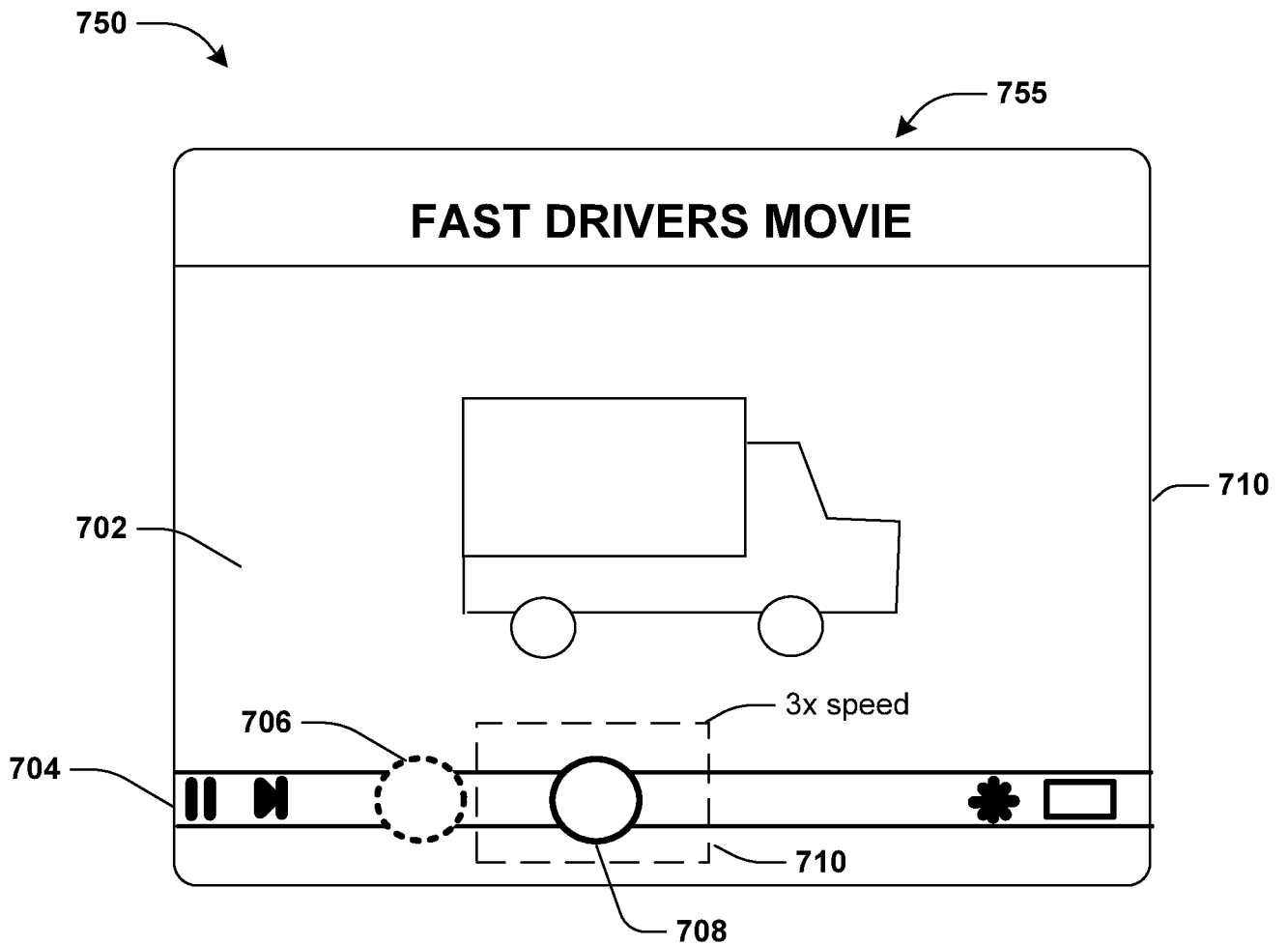
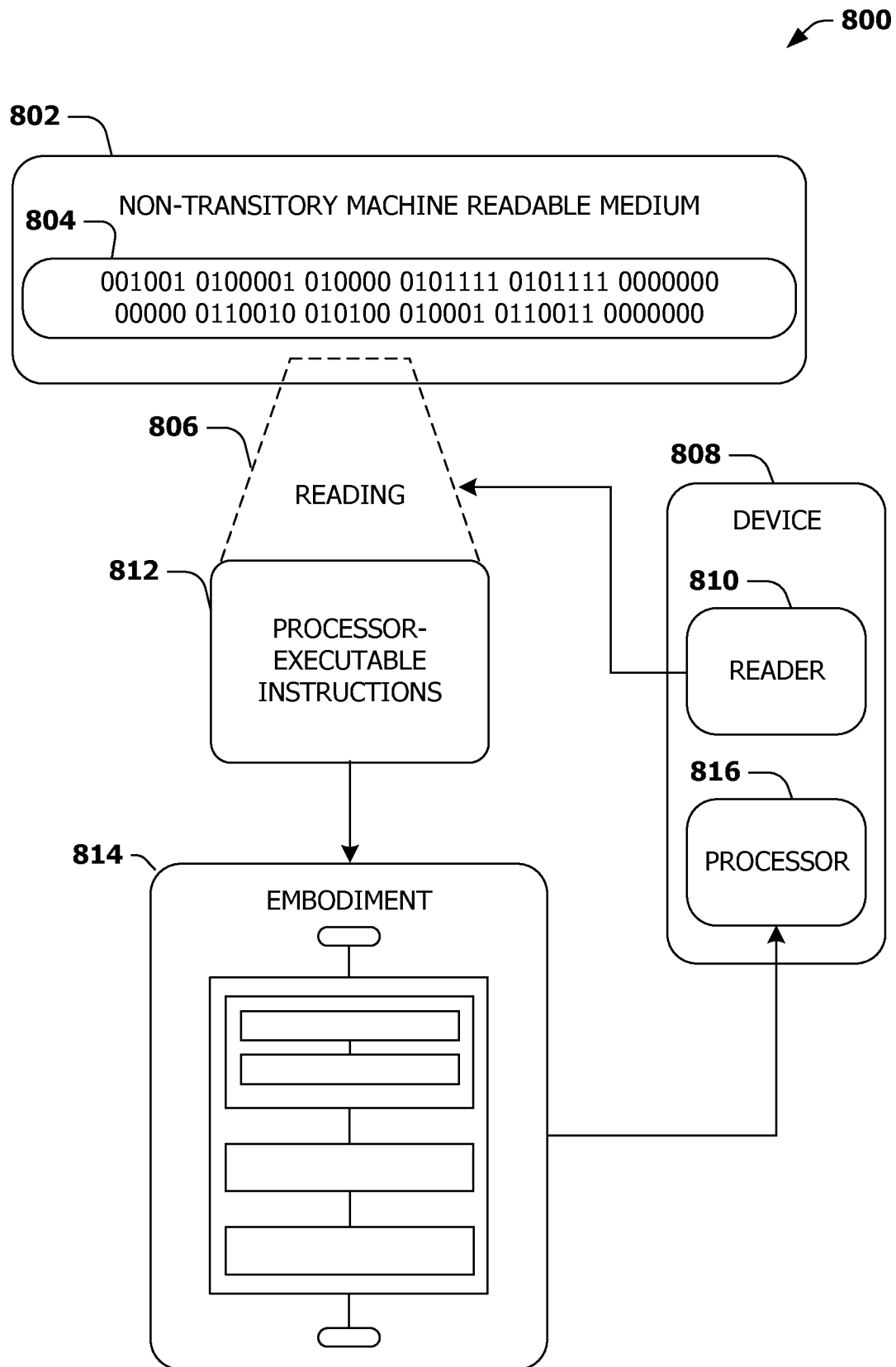
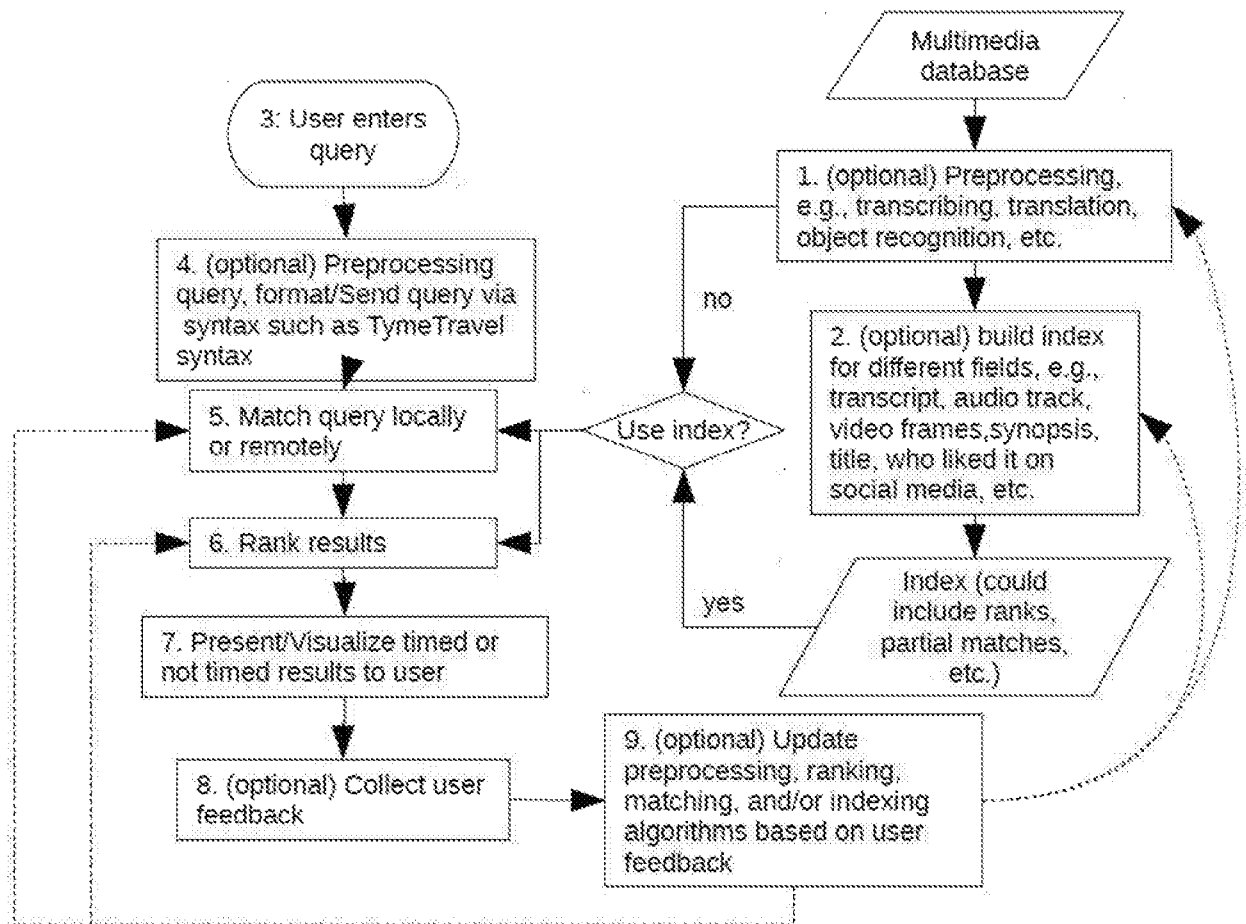
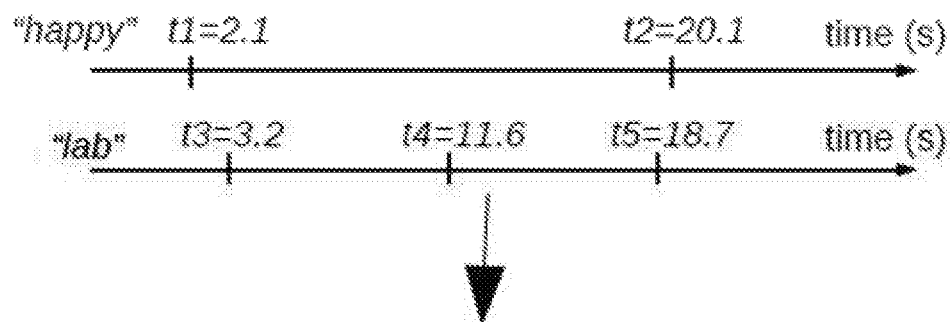


FIG. 7B

**FIG. 8**

**FIG. 9**

1. Find all timestamps that match individual terms

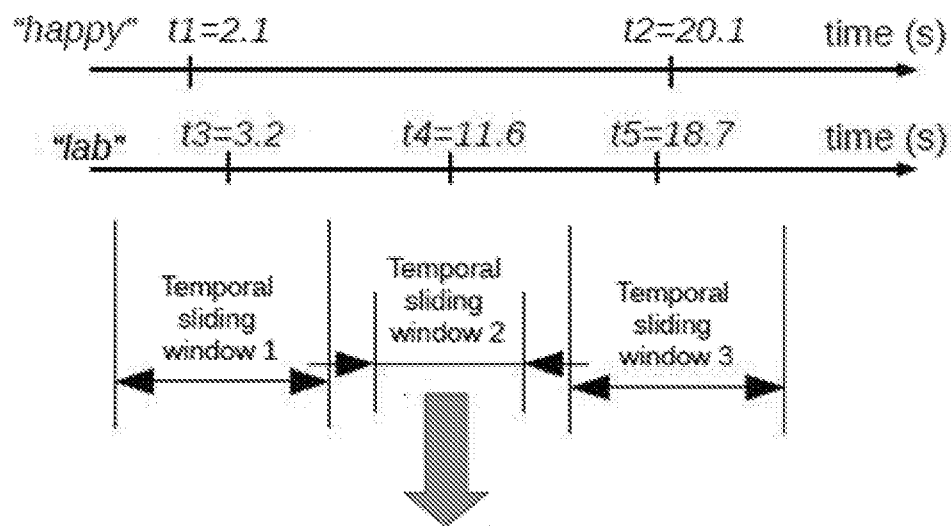


2. Enumerate all combinations of timestamps and check query satisfiability

"happy"	"lab"	satisfiability
$t1=2.1$	$t3=3.2$	true
$t1=2.1$	$t4=11.6$	false
$t1=2.1$	$t5=18.7$	false
$t1=20.1$	$t3=3.2$	false
...

FIG. 10

1. Find occurrences of individual terms and establish sliding window

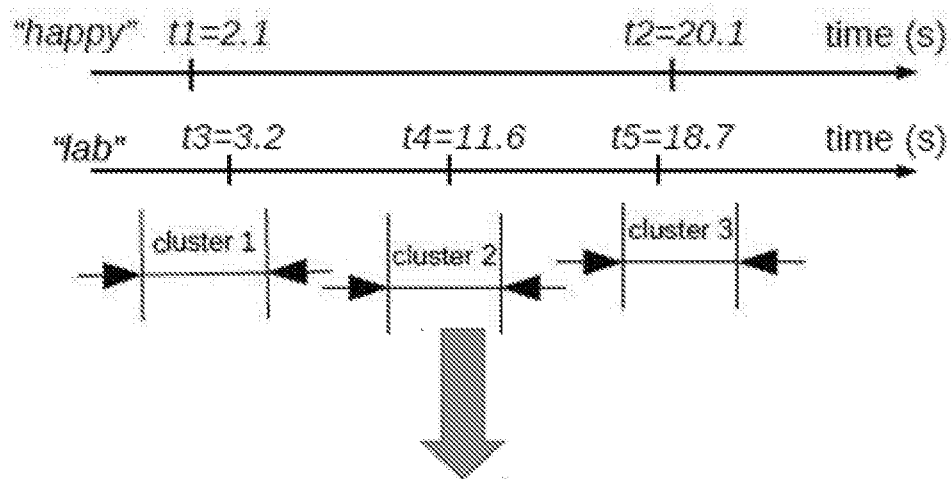


2. Check query satisfiability in each sliding window

window	satisfiability
1	true
2	false
3	true
...	...

FIG. 11

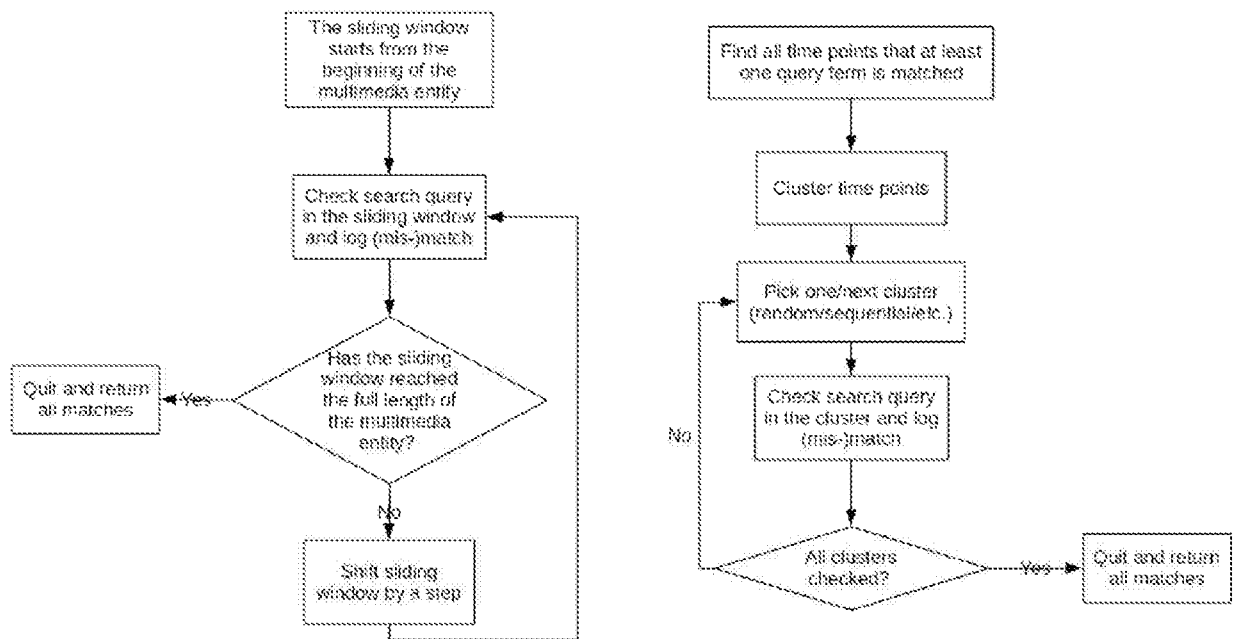
1. Find timestamps of individual terms and cluster the timestamps

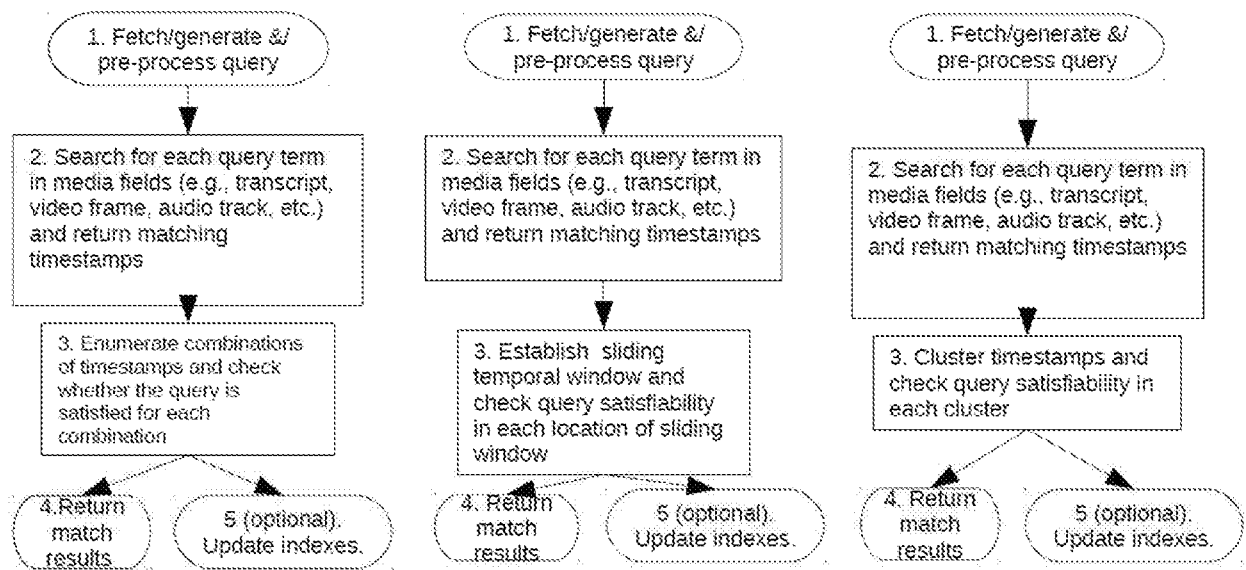


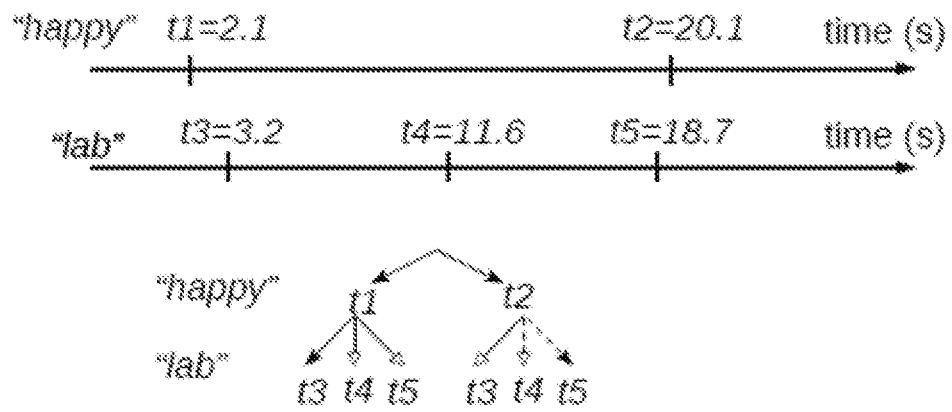
2. Check query satisfiability in each cluster

cluster	satisfiability
1	true
2	false
3	true
...	...

FIG. 12

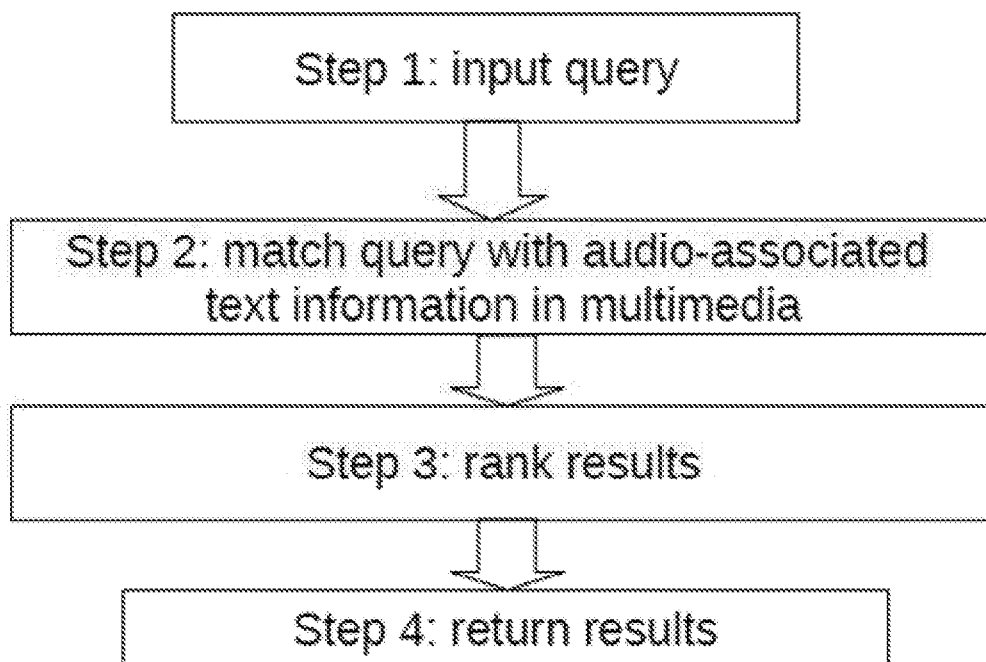
**FIG. 13**

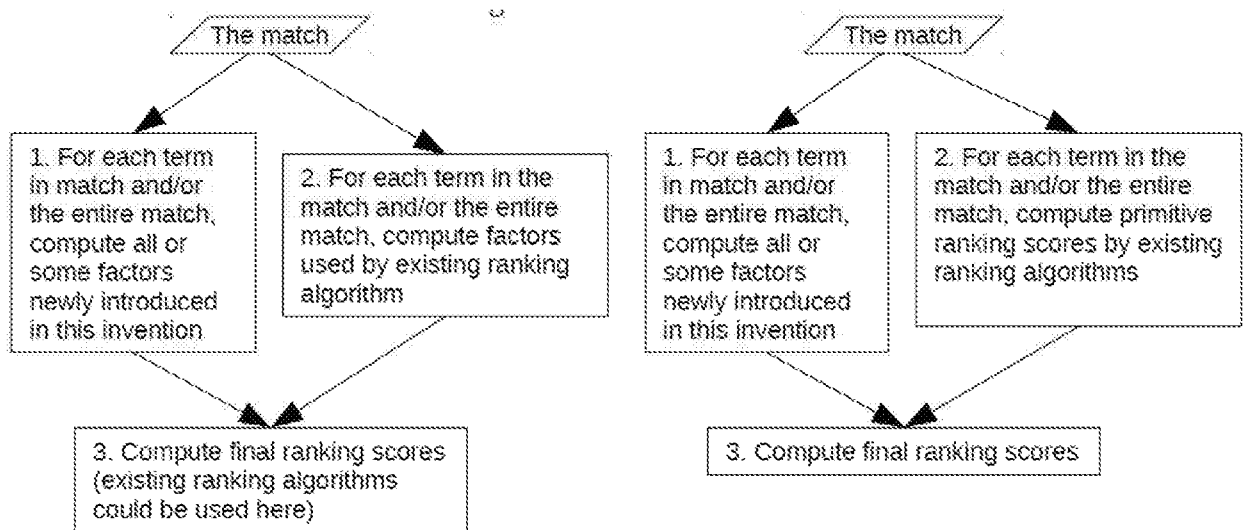
**FIG. 14**

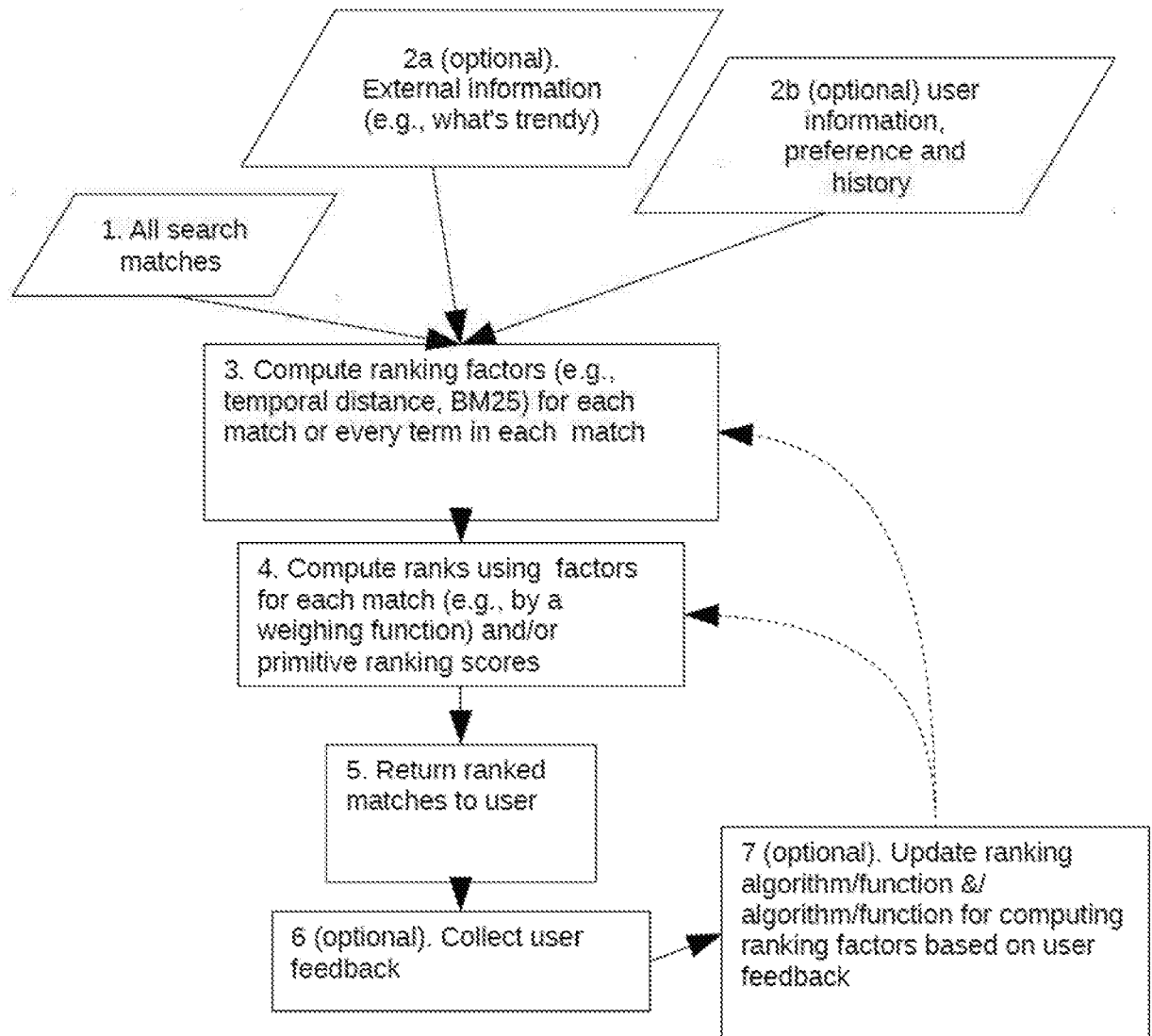
**FIG. 15**

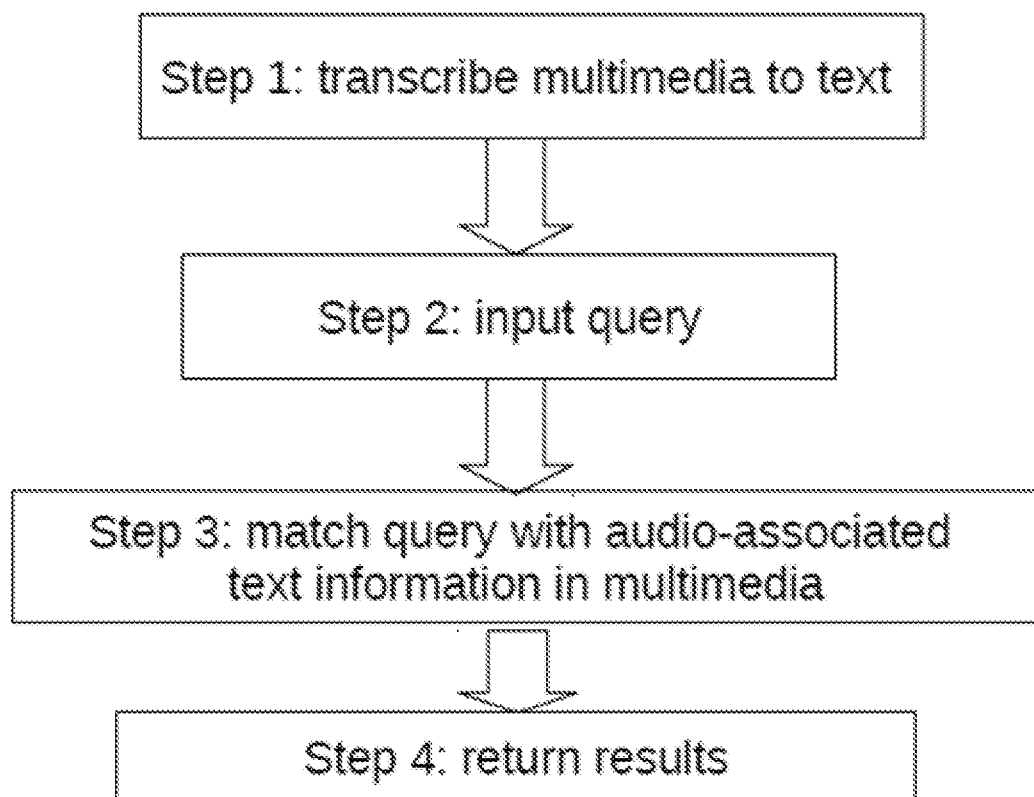
	0.0 (let's)	1.1 (take)	...	5.9	...	7.3	...	8.5	...	9.9	...	11.3	...
Italy	0	0	...	0	...	0	...	1	...	0	...	0	...
volca no	0	0	...	1	...	0	...	0	...	1	...	0	...
erupt	0	0	...	0	...	1	...	1	...	0	...	1	...

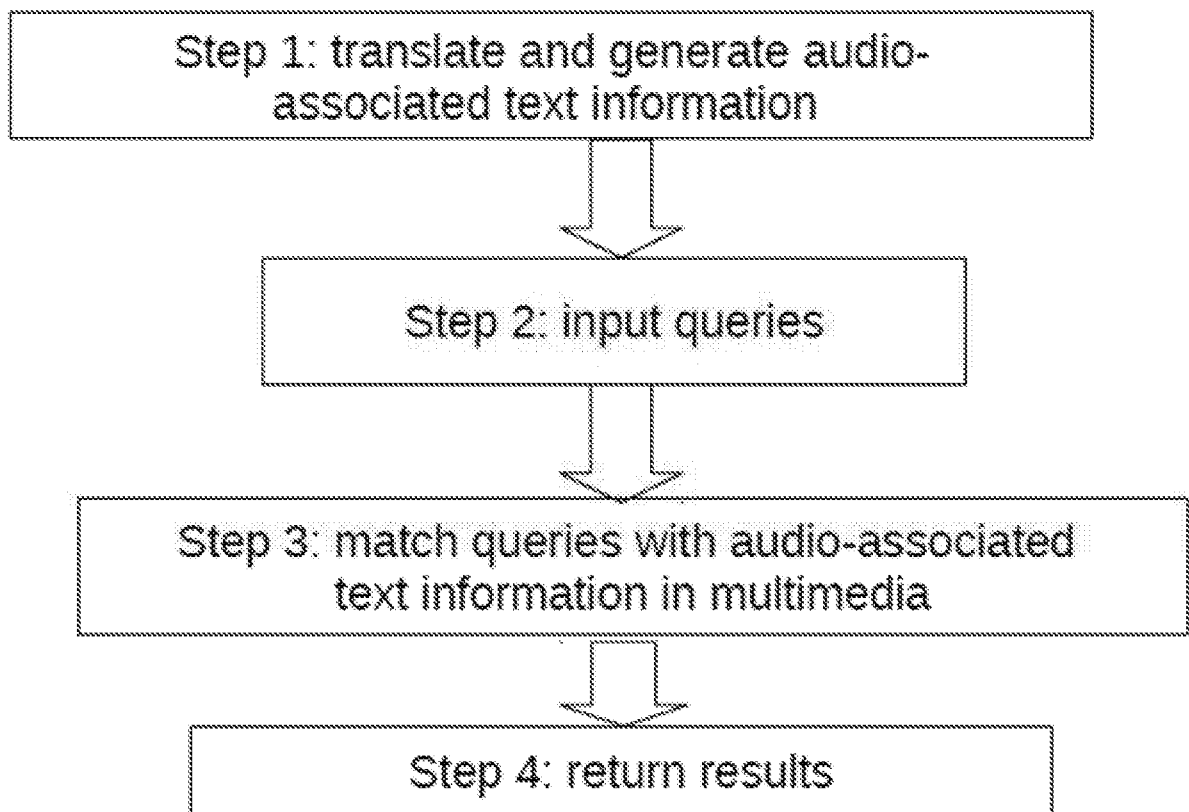
FIG. 16

**FIG. 17**

**FIG. 18**

**FIG. 19**

**FIG. 20**

**FIG. 21**

1. Rachel run away from wedding (episode 1, 1m 25s)
2. Friends meet each other in the Central Perk coffee shop (episode 1, 3m 05s)
3. Ross discussed that his ex-wife is a lesbian (episode 1, 6m 25s)
4. Rachel decided to stay with Monica (episode 1, 10m 05s)
5. Joey hit on Rachel (episode 1, 15m 25s)
6. Ross showed interest in Rachel (episode 1, 17m 21s)
7. Rachel become a waitress of Central Perk coffee shop (episode 1, 20m 25s)

FIG. 22

1. Rachel run away from wedding (episode 1, 1m 25s)
2. Friends meet each other in the Central Perk coffee shop (episode 1, 3m 05s)
3. Ross discussed that his ex-wife is a lesbian (episode 1, 6m 25s)
4. Rachel decided to stay with Monica (episode 1, 10m 05s)
5. Joey hit on Rachel (episode 1, 15m 25s)
6. Ross showed interest in Rachel (episode 1, 17m 21s)
7. Rachel become a waitress of Central Perk coffee shop (episode 1, 20m 25s)
8. Rachel is going out with Paulo (episode 5, 10m 25s)

FIG. 23

1. Rachel run away from wedding (episode 1, 1m 25s)
2. Friends meet each other in the Central Perk coffee shop (episode 1, 3m 05s)
3. Ross discussed that his ex-wife is a lesbian (episode 1, 6m 25s)
4. Rachel decided to stay with Monica (episode 1, 10m 05s) 4.1 Rachel checked out Monica's apartment (episode 1, 10m 05s) 4.2 Monica mentioned that she got the apartment from her grandma (episode 1, 10m 25s)
5. Joey hit on Rachel (episode 1, 15m 25s)
6. Ross showed interest in Rachel (episode 1, 17m 21s)
7. Rachel become a waitress of Central Perk coffee shop (episode 1, 20m 25s)
8. Rachel is going out with Paulo (episode 5, 10m 25s)

FIG. 24


	1. (episode 1, 1m 25s)
2. Friends meet each other in the Central Perk coffee shop (episode 1, 3m 05s)	
3. Ross discussed that his ex-wife is a lesbian (episode 1, 6m 25s)	
4. Rachel decided to stay with Monica (episode 1, 10m 05s)	
4.1 Rachel checked out Monica's apartment (episode 1, 10m 05s)	
4.2 Monica mentioned that she got the apartment from her grandma (episode 1, 10m 25s)	
5. Joey hit on Rachel (episode 1, 15m 25s)	
6. Ross showed interest in Rachel (episode 1, 17m 21s)	
7. Rachel become a waitress of Central Perk coffee shop (episode 1, 20m 25s)	
8. Rachel is going out with Paulo (episode 5, 10m 25s)	

FIG. 25



(episode 1, 1m 25s)

1. Rachel run away from wedding
2. Friends meet each other in the Central Perk coffee shop (episode 1, 3m 05s)
3. Ross discussed that his ex-wife is a lesbian (episode 1, 6m 25s)
4. Rachel decided to stay with Monica (episode 1, 10m 05s)
 - 4.1 Rachel checked out Monica's apartment (episode 1, 10m 05s)
 - 4.2 Monica mentioned that she got the apartment from her grandma (episode 1, 10m 25s)
5. Joey hit on Rachel (episode 1, 15m 25s)
6. Ross showed interest in Rachel (episode 1, 17m 21s)
7. Rachel become a waitress of Central Perk coffee shop (episode 1, 20m 25s)
8. Rachel is going out with Paulo (episode 5, 10m 25s)

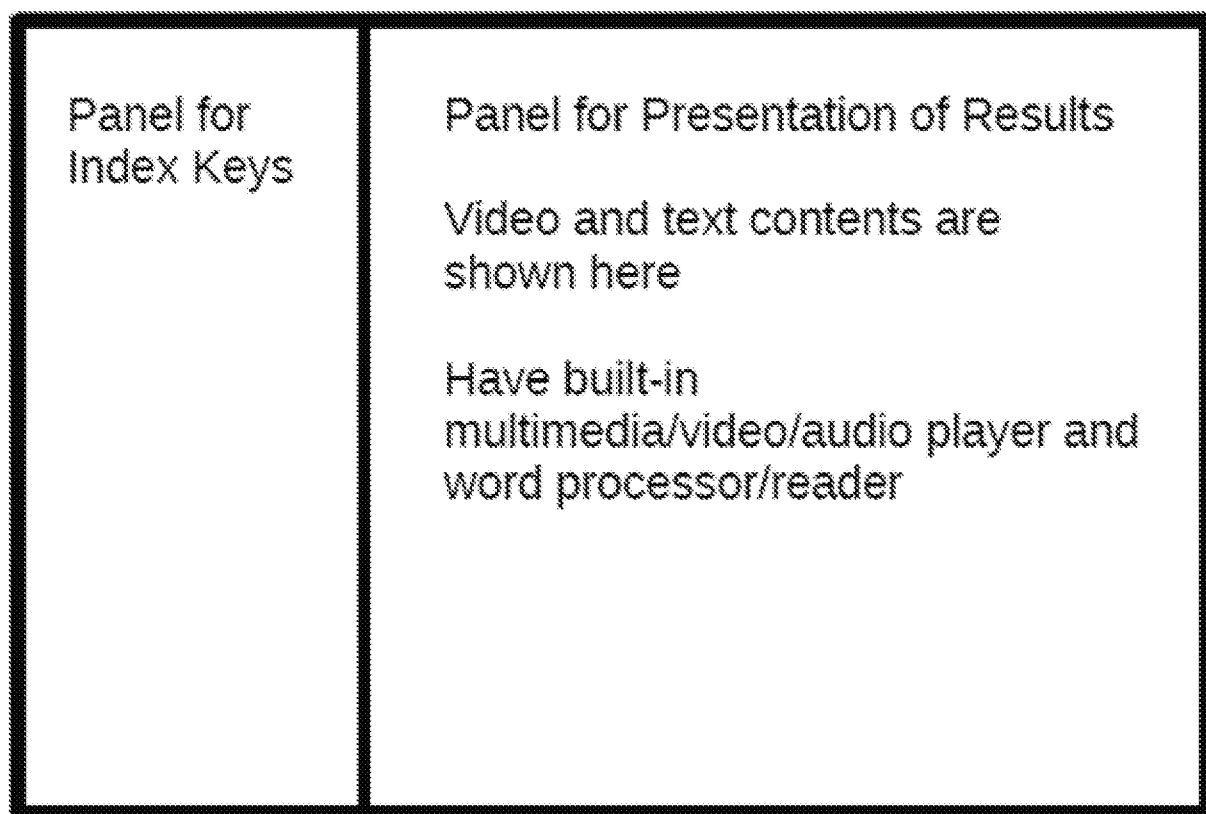
FIG. 26

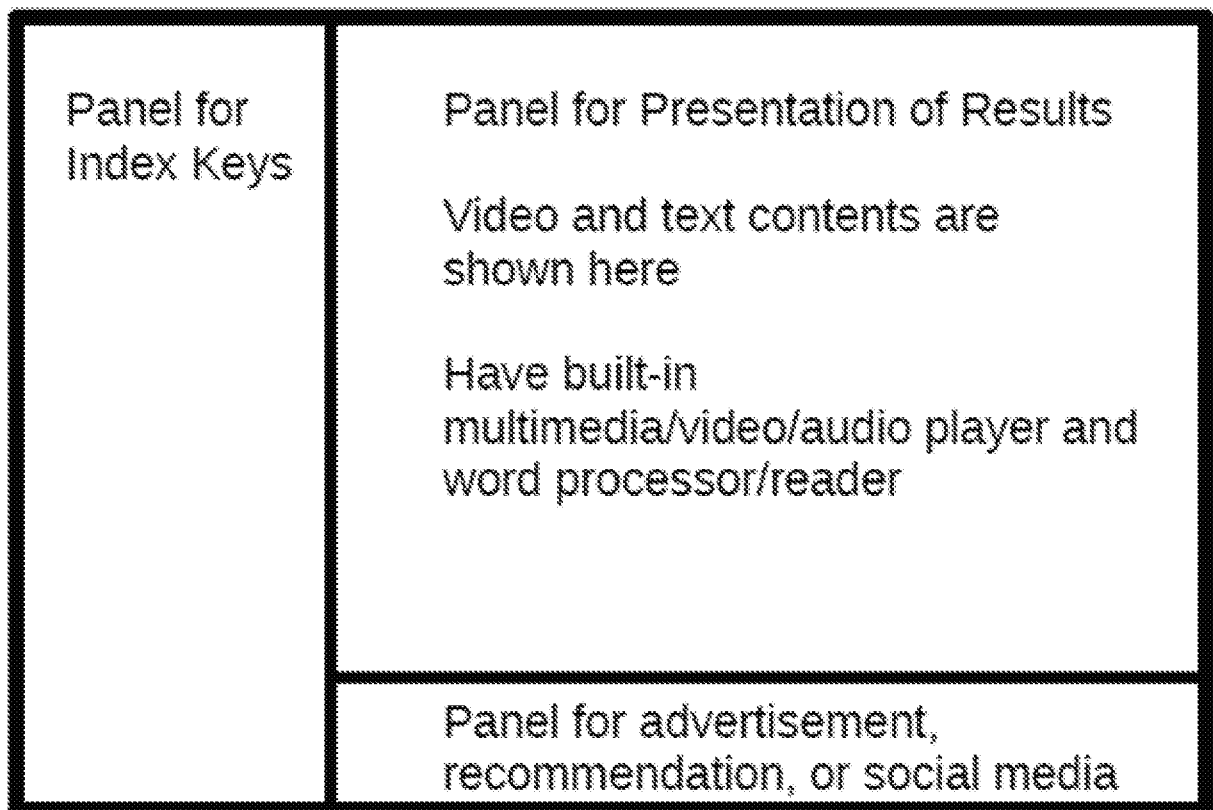
1. Friends meet each other in the Central Perk	1. a. Friends meet each other in the Central Perk coffee shop: timestamp 1 in episode 1
	1. b. Friends meet each other in the Central Perk coffee shop: timestamp 2 in episode 4 (show up upon user mouse-click)
	1. c. Friends meet each other in the Central Perk coffee shop: timestamp 3 in episode 5 (show up upon user mouse-click)
2. Chandler tells an awful joke timestamp 4 in episode 10	

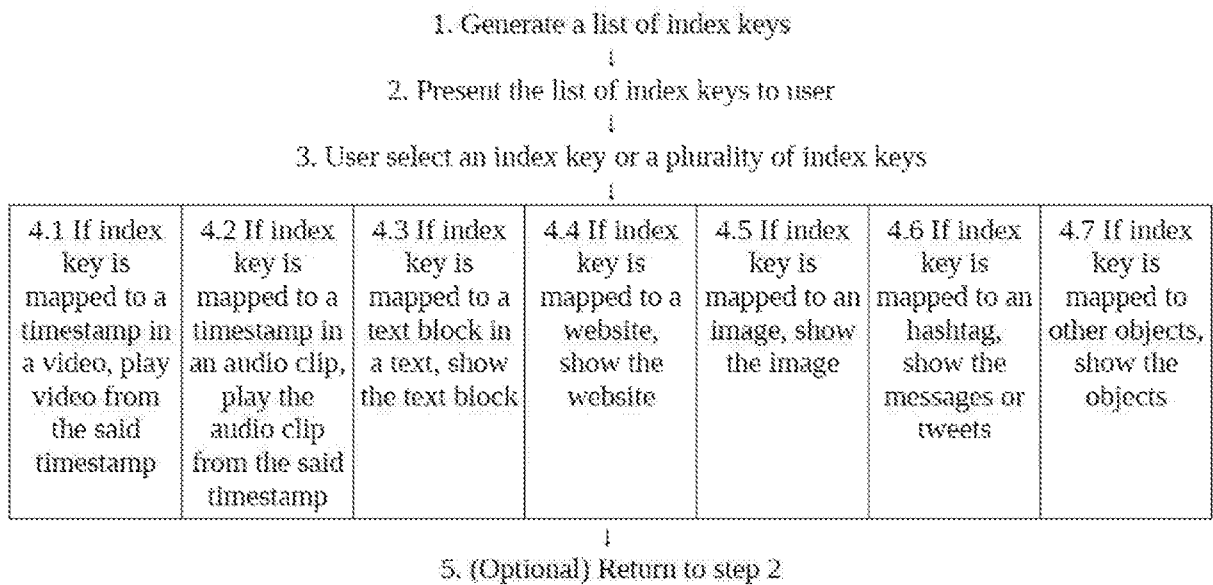
FIG. 27

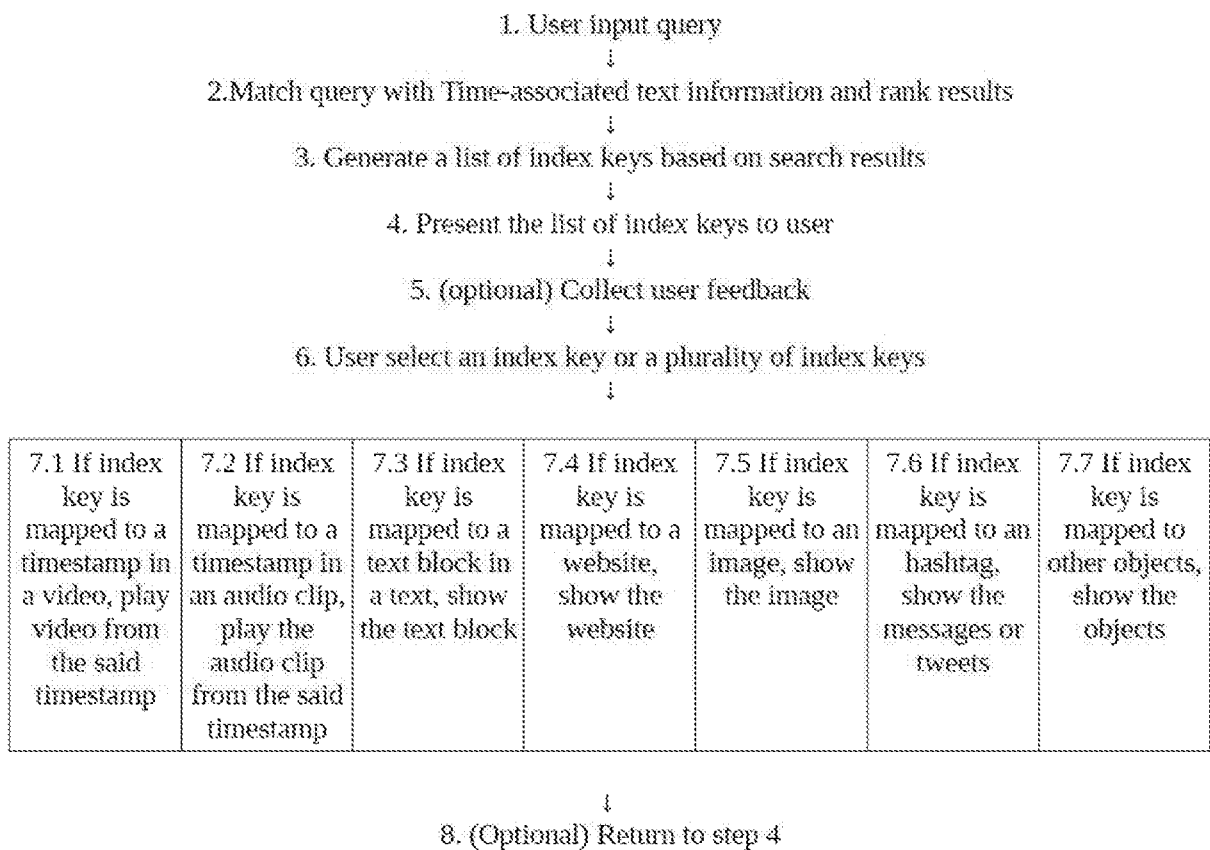
1. Law of reflection (in textbook)
2. Demonstration of Law of reflection (in video): timestamp 1 in file 1
3. Law of refraction (in textbook)
4. Demonstration of Law of refraction (in video): timestamp 2 in file 1
5. Discussion of geometric optics (on website)
6. Further discussion of geometric optics (in audio) : timestamp 3 in file 2
7. discussions from students (on Tweeter): hashtags
8. Summary graph of geometric optics (in picture)

FIG. 28

**FIG. 29**

**FIG. 30**

**FIG. 31**

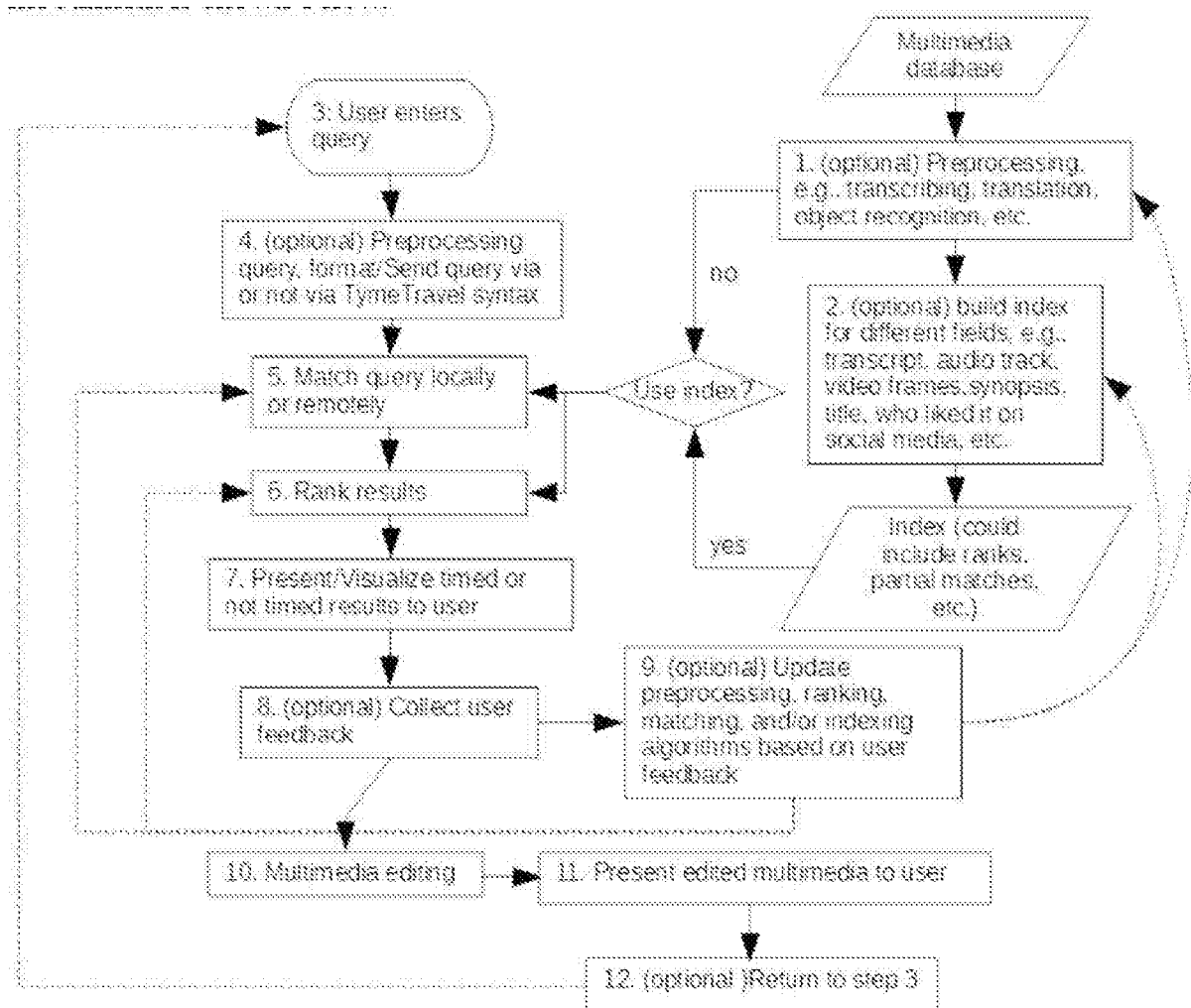
**FIG. 32**

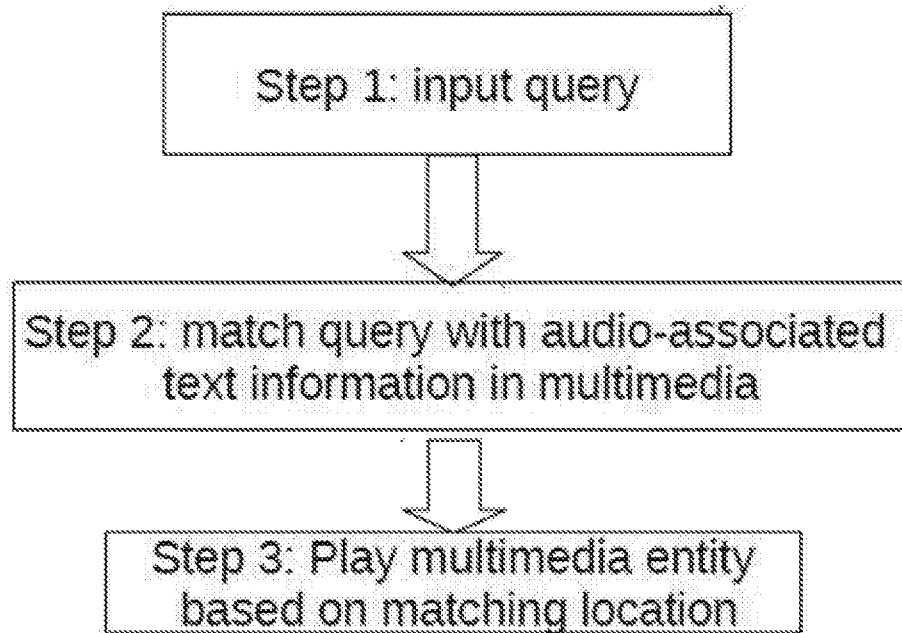
1. Rachel meet Ross: season 1, episode 1, timestamp 2m 20s
2. Ross has a crush on Rachel: season 1, episode 1, timestamp 5m 20s
3. Rachel has a crush on Ross: season 2, episode 1, timestamp 3m 30s
4. Rachel and Ross go out: 4.1 Ross made a list of pros and cons; relationship goes bad: season 2, episode 5, timestamp 3m 30s
4.2 Rachel and Ross starts to go out: season 2, episode 8, timestamp 13m 30s
4.3 Ross is jealous of Rachel working with a male colleague: season 3, episode 3, timestamp 12m 30s
5. Rachel and Ross break up: season 4, episode 1, timestamp 3m 10s
6. Rachel and Ross had sex and Rachel is pregnant: season 8, episode 12, timestamp 1m 10s
7. Rachel and Ross had a daughter: season 9, episode 2, timestamp 11m 10s
8. Rachel and Ross is going to get married: season 10, episode 24, timestamp 18m 10s

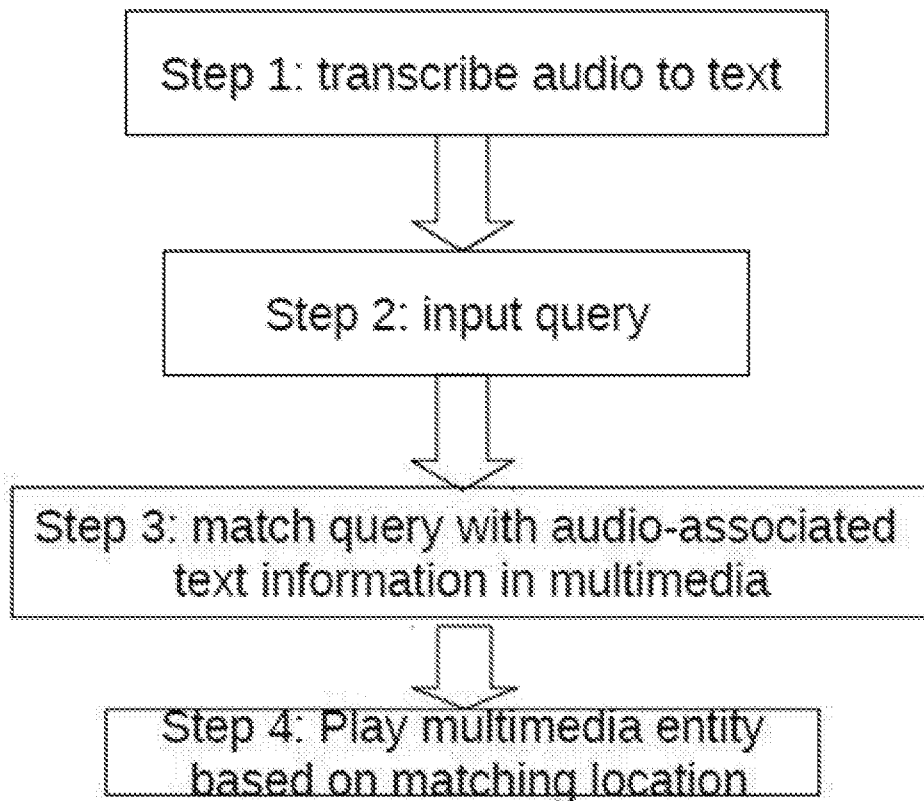
FIG. 33

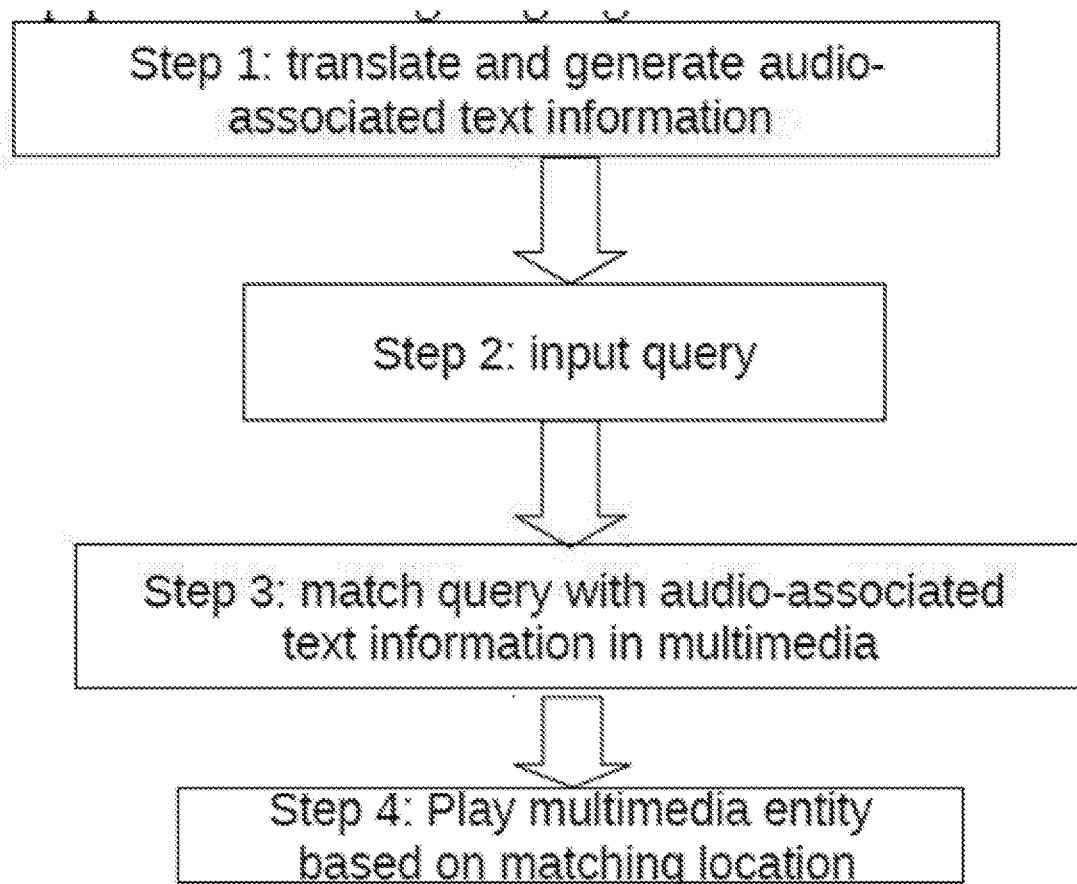
1. Rachel and Ross: plot 1 (timestamp 1)
2. Rachel and Ross: plot 2 (timestamp 2)
3. Rachel and Ross: plot 3 (timestamp 3)
4. Rachel and Ross: plot 4 4.1 Rachel and Ross: plot 4.1 (timestamp 4) 4.2 Rachel and Ross: plot 4.2 (timestamp 5) 4.3 Rachel and Ross: plot 4.3 (timestamp 6)
5. Rachel and Ross: plot 5 (timestamp 7)
6. Rachel and Ross: plot 6 (timestamp 8)
7. Rachel and Ross: plot 7 (timestamp 9)
8. Rachel and Ross: plot 8 (timestamp 10)

FIG. 34

**FIG. 35**

**FIG. 36**

**FIG. 37**

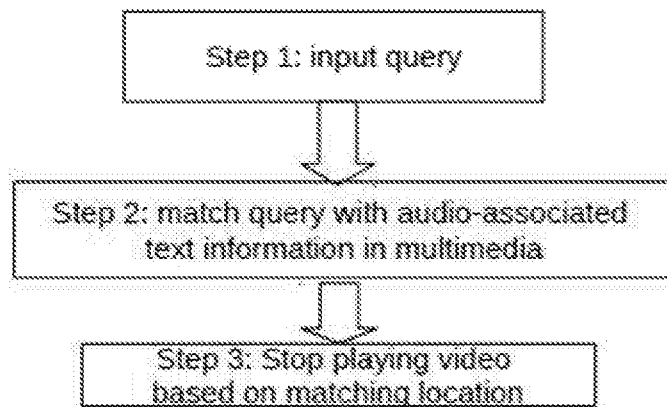
**FIG. 38**

Input your query here: "Happy" AND "Lab"

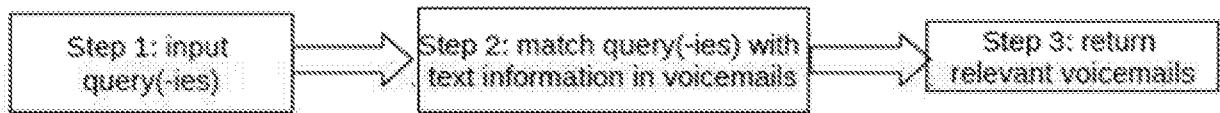
Search Results:
Instruction: move cursor over the video dip to preview; dip on the video dip with the desirable timestamp to play the video from that timestamp in full screen

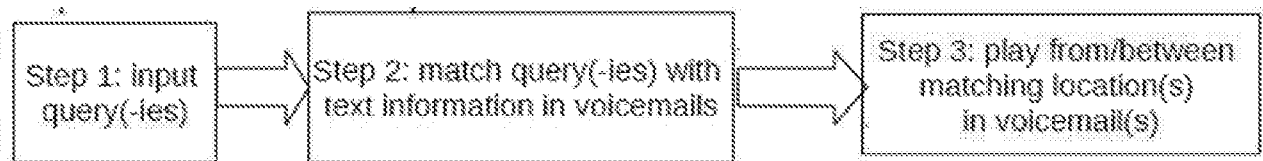
Timestamp 1: 3m25s (video dip 1 is shown here)	Timestamp 2: 8m35s (video dip 3 is shown here)
Timestamp 3: 33m35s (video dip 3 is shown here)	Timestamp 4: 38m15s (video dip 4 is shown here)

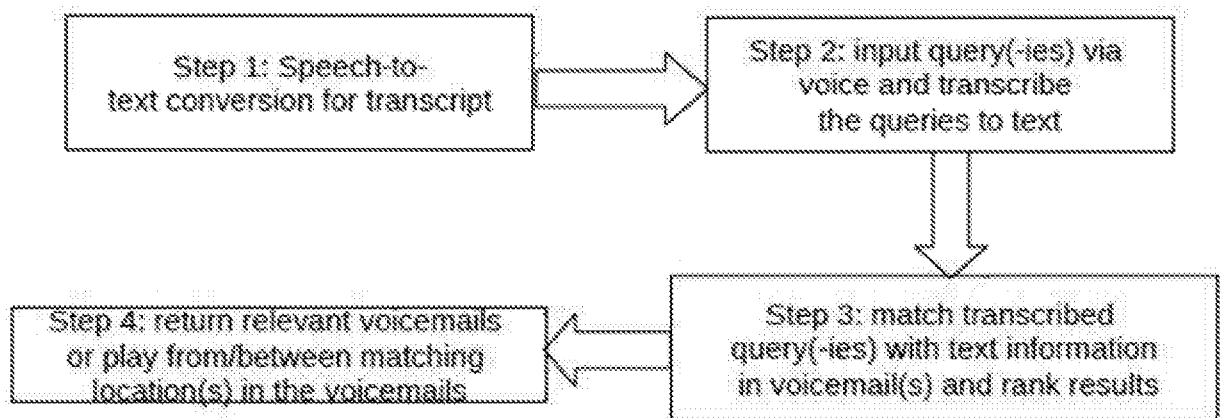
FIG. 39

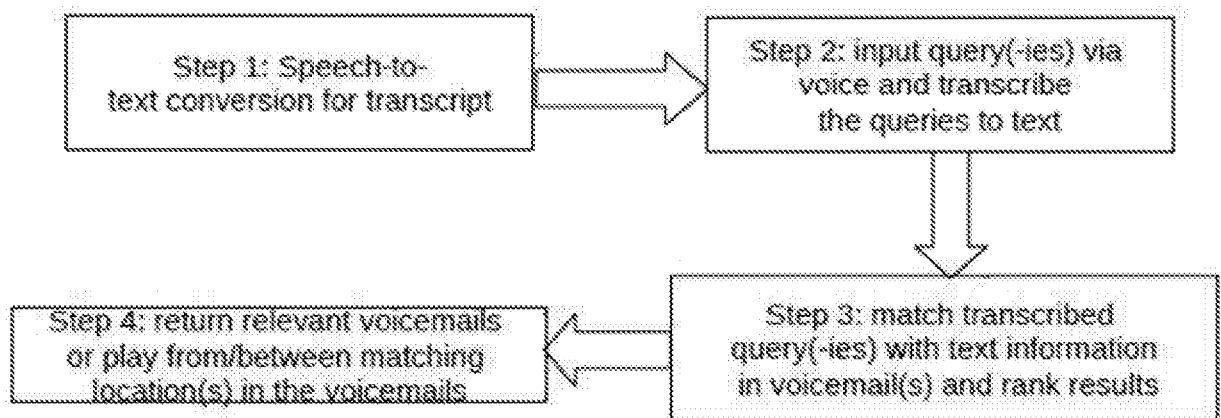
**FIG. 40**

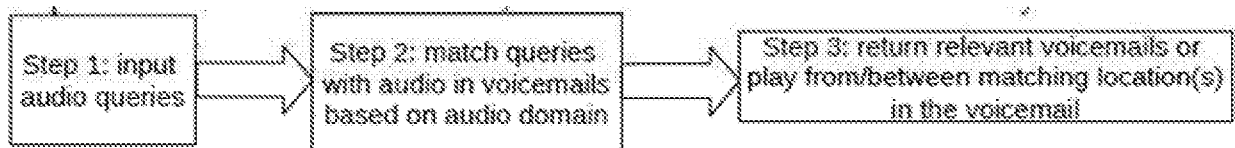
**FIG. 41**

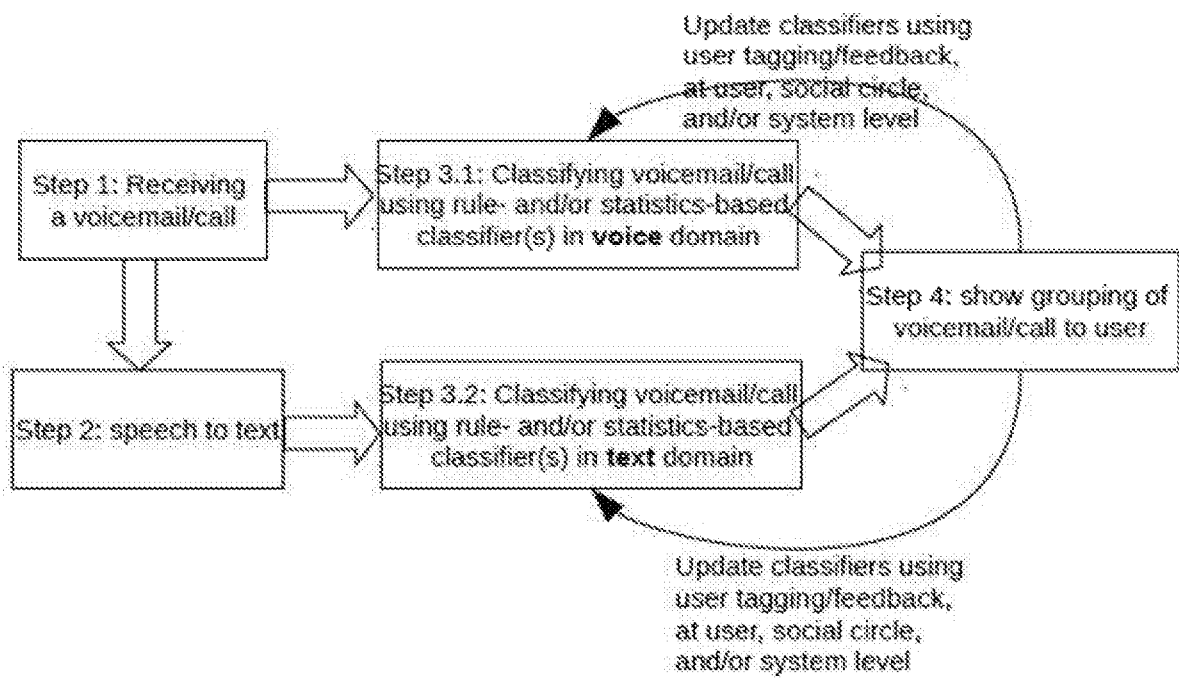
**FIG. 42**

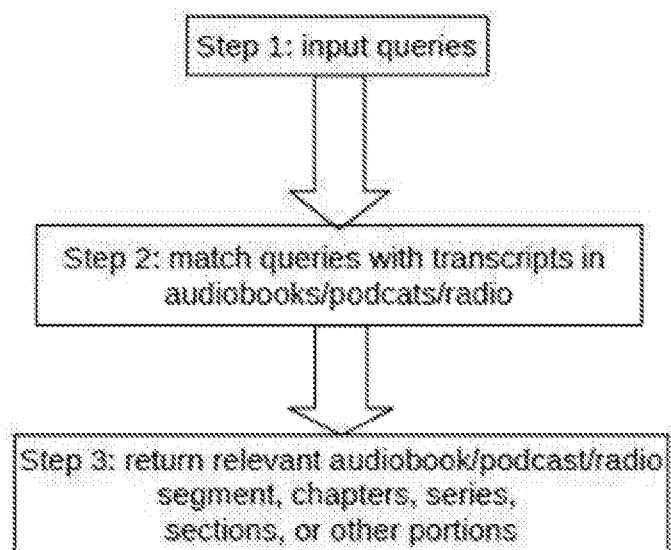
**FIG. 43**

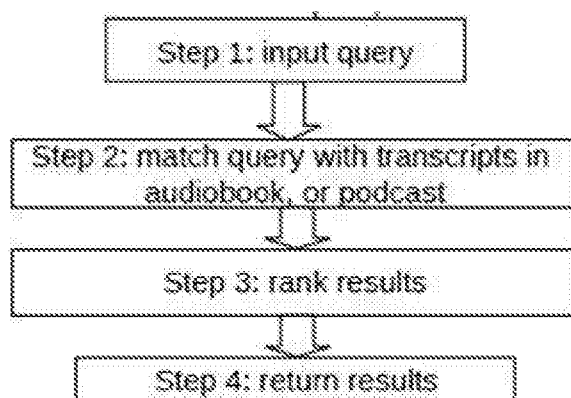
**FIG. 44**

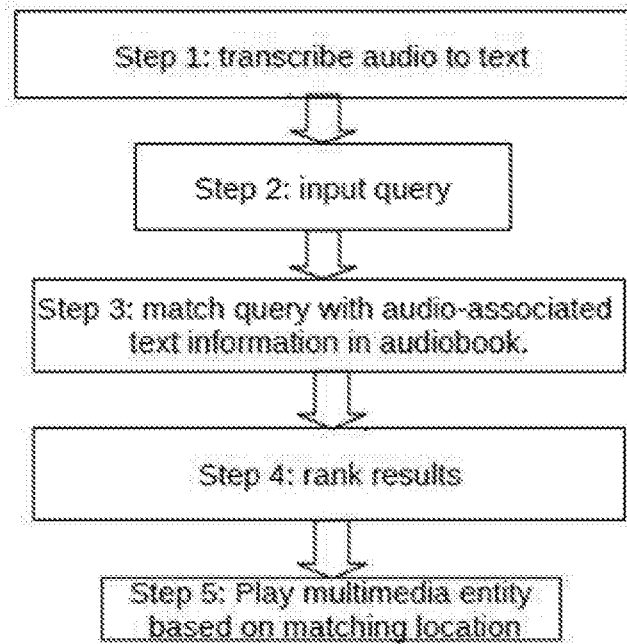
**FIG. 45**

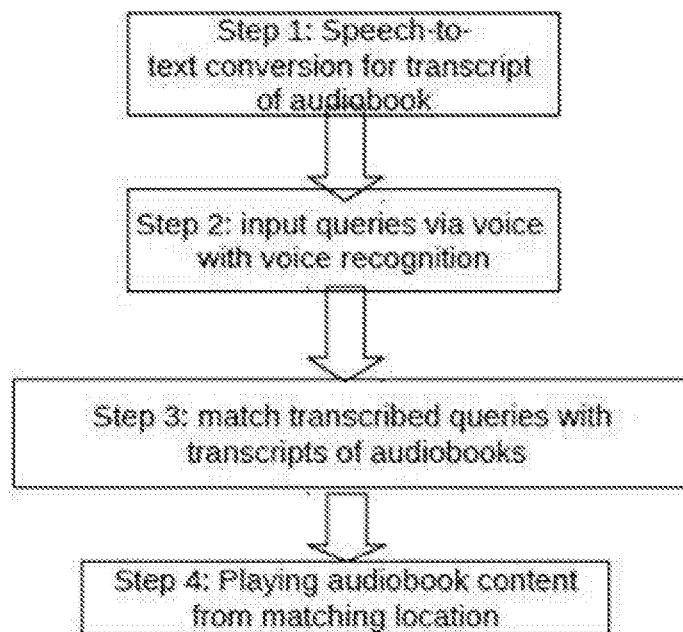
**FIG. 46**

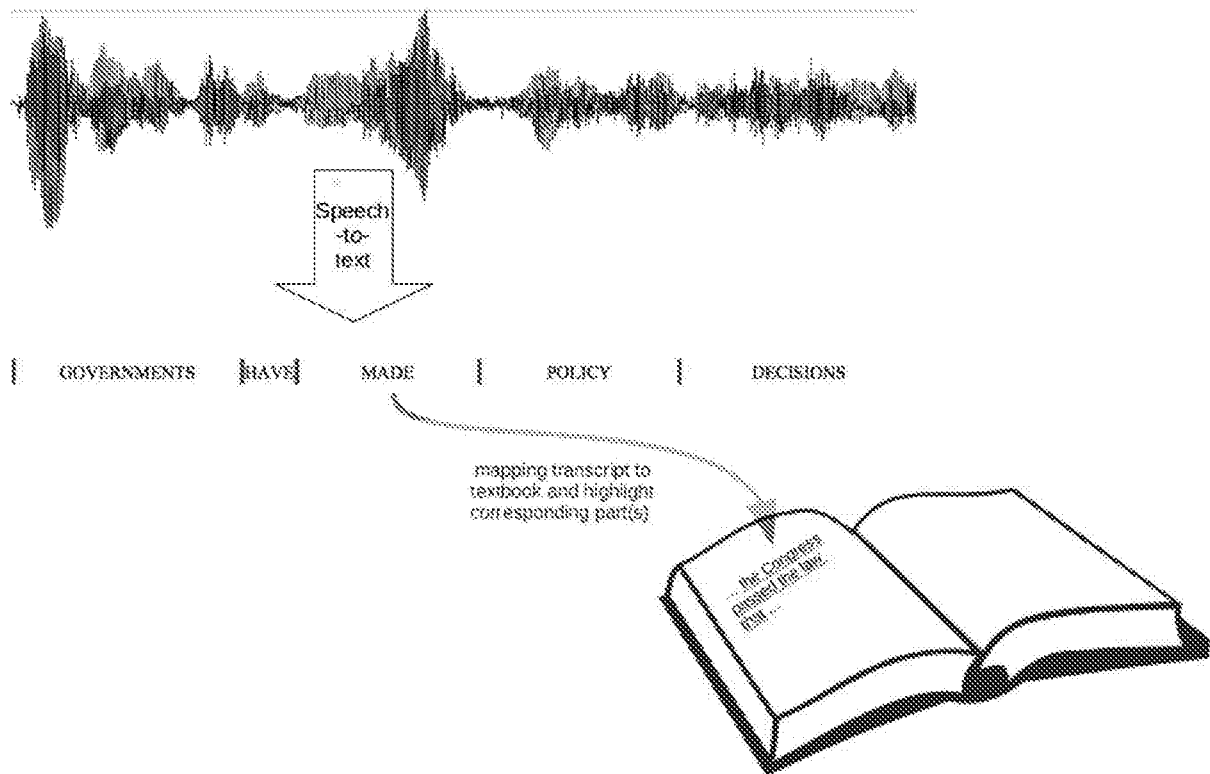
**FIG. 47**

**FIG. 48**

**FIG. 49**

**FIG. 50**

**FIG. 51**

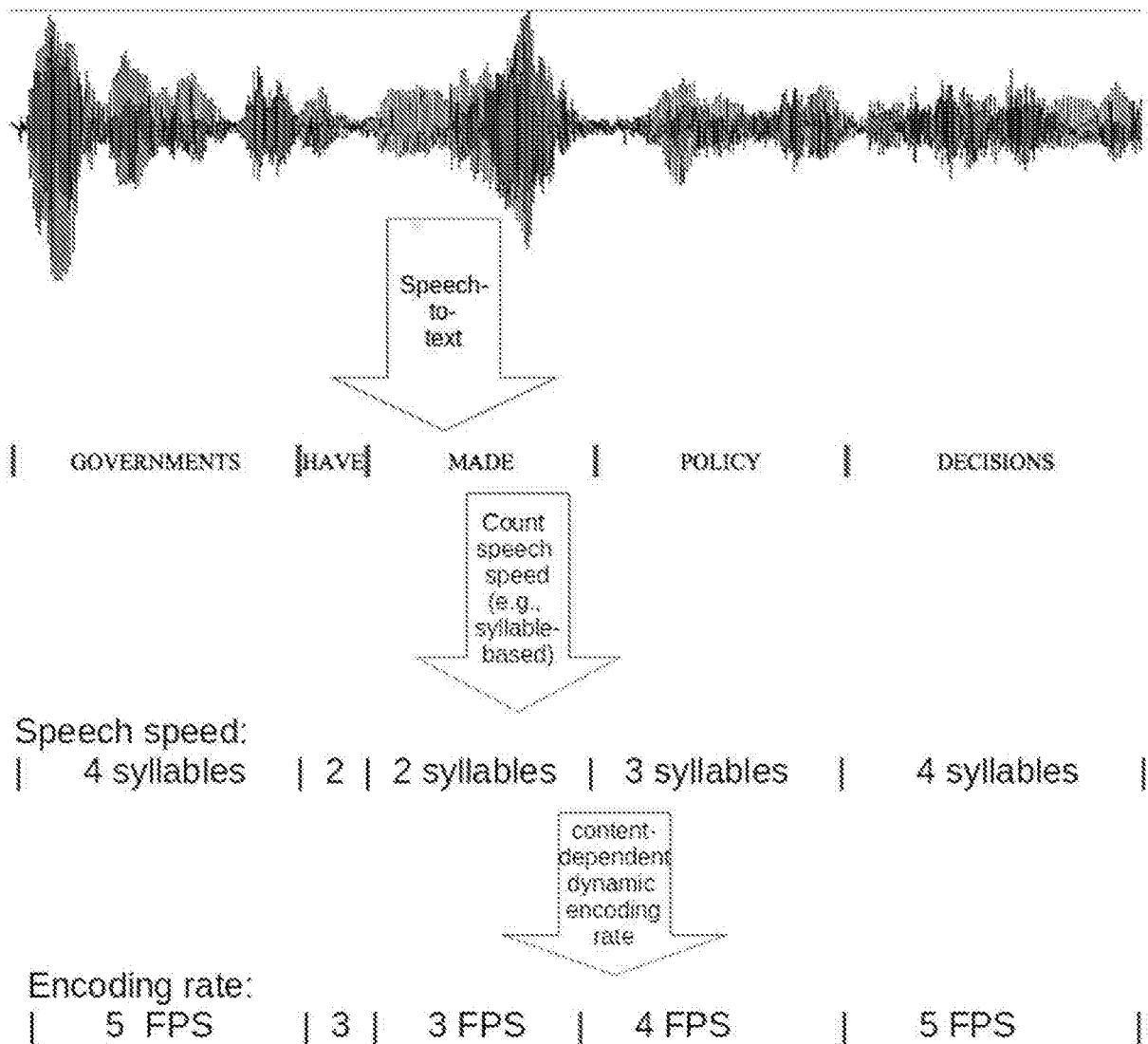
**FIG. 52**

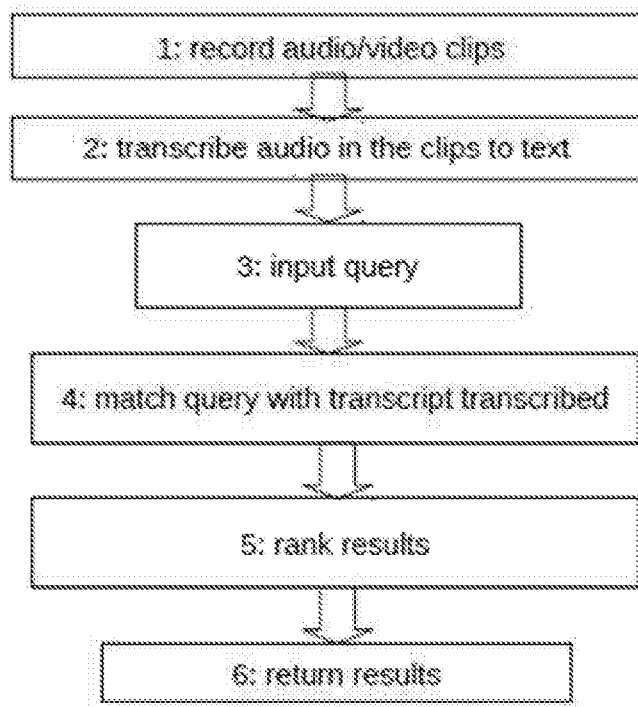
Properties in the file	Functionality
Timed contexts	Context information with timestamps Example: 0h0m0s- 0h4m08s: coffee 0h4m09s- 0h8m18s: shopping 0h8m19s- 0h20m18s: cooking
Ideal ad-playing timestamps	Suggested ideal ad-playing timestamps for starting and ending playing ads Example: Play advertisement at: 0h1m0s- 0h2m00s; 0h4m29s- 0h6m18s; 0h8m19s- 0h12m00s
timed ad-compatibility score	Whether the video segments are ad-compatible or not, with timestamps Example: 0h0m0s- 0h4m08s: ad-compatible 0h4m09s- 0h8m18s: ad-incompatible 0h8m19s- 0h20m18s: ad-compatible
timed ad-layout	Where to place overlaying ads on top of the video (region of overlay), with timestamps Example: 0h0m0s- 0h4m08s: matrix 1 describing where the ad is 0h4m09s- 0h8m18s: matrix 2 describing where the ad is 0h8m19s- 0h20m18s: matrix 3 describing where the ad is For instance, the matrix may indicate where the ad is using the index of the pixels: (x, y, ad_layout), where x and y are spatial index of the pixel, ad_layout describes where the ad is. For instance, the pixels for placing the ad can have the ad_layout value of 1, and other pixels can have the the ad_layout value of 0 With timestamps, the matrix can become (x,y,ad_layout, t)
timed ad-adjustment	How to adjust the overlaying ads (transparency, color, size, animation pattern, etc) on top of the video, with timestamps Example: 0h0m0s- 0h4m08s: matrix 1 describing adjusting overlaying ad 0h4m09s- 0h8m18s: matrix 2 describing adjusting overlaying ad 0h8m19s- 0h20m18s: matrix 3 describing adjusting overlaying ad For instance, the matrix may indicate where the ad is using the index of the pixels: (x, y, ad_adjustment), where x and y are spatial index of the pixel, and ad_adjustment is another matrix describes how to adjust the ad. The ad_adjustment may be in the format of (transparency, color).

FIG. 53

Time (hours, minutes, seconds)	context	ideal ad-playing timestamps	ad-compatibility score	ad-layout	ad-adjustment
0,0,0	coffee	start	1	Matrix 1 describing where the ad is	Matrix A describing adjustment of overlaying ad
0,0,1	coffee	stop	1	Matrix 2 describing where the ad is	Matrix B describing adjustment of overlaying ad
0,0,2	sugar	null	0	null	null
0,0,3	sugar	null	0	null	null

FIG. 54

**FIG. 55**

**FIG. 56**

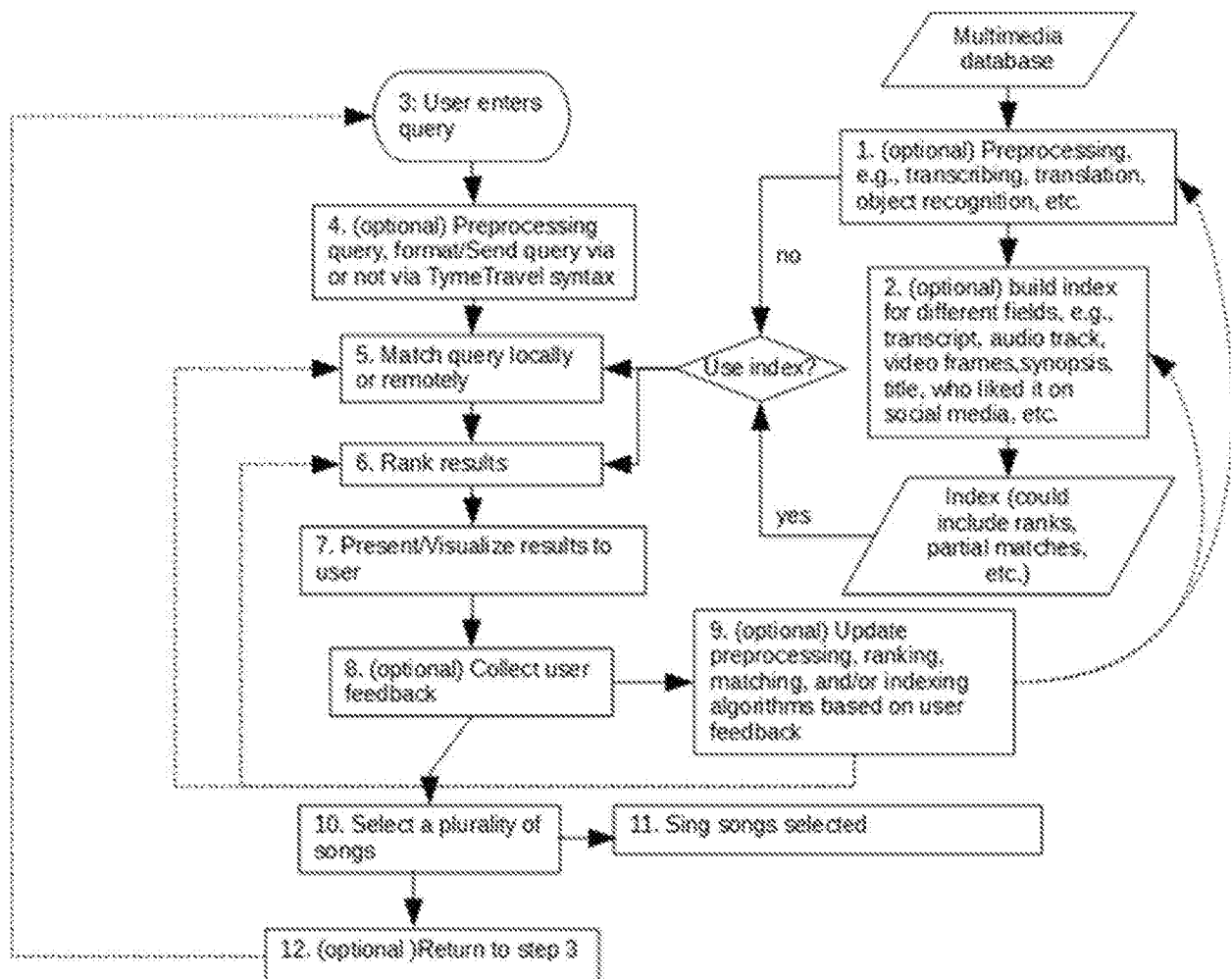


FIG. 57

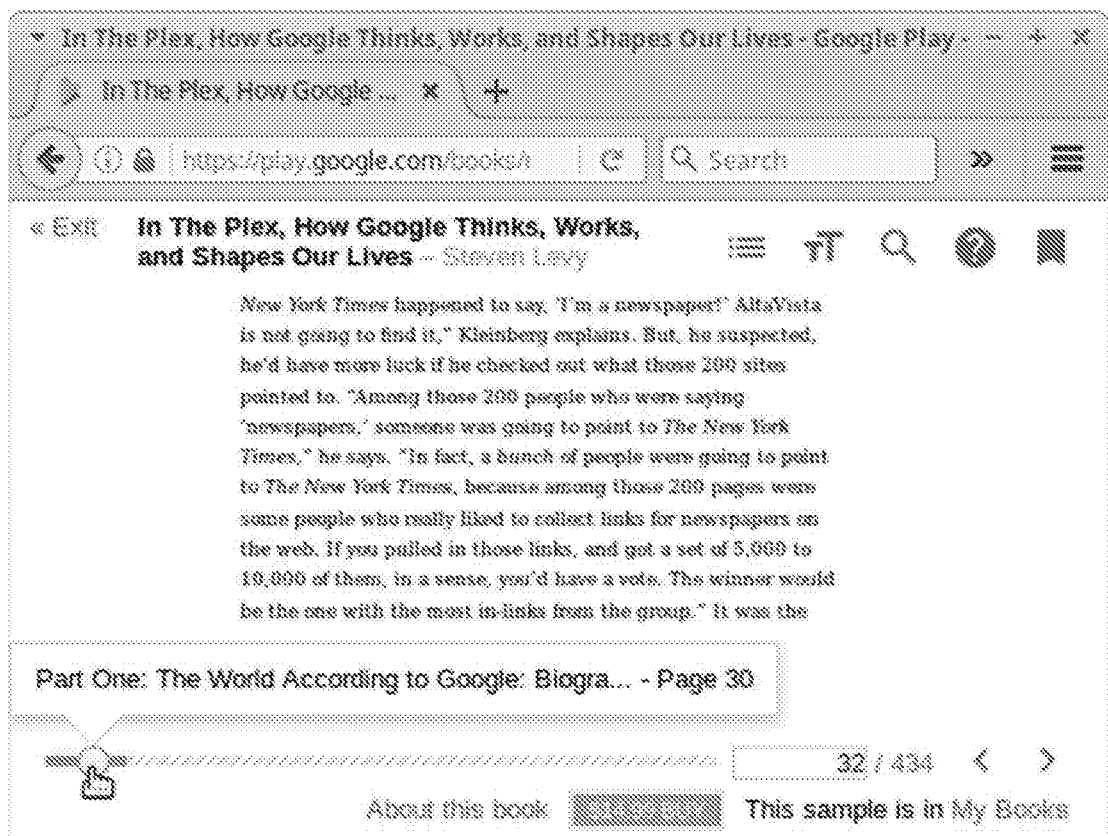


FIG. 58

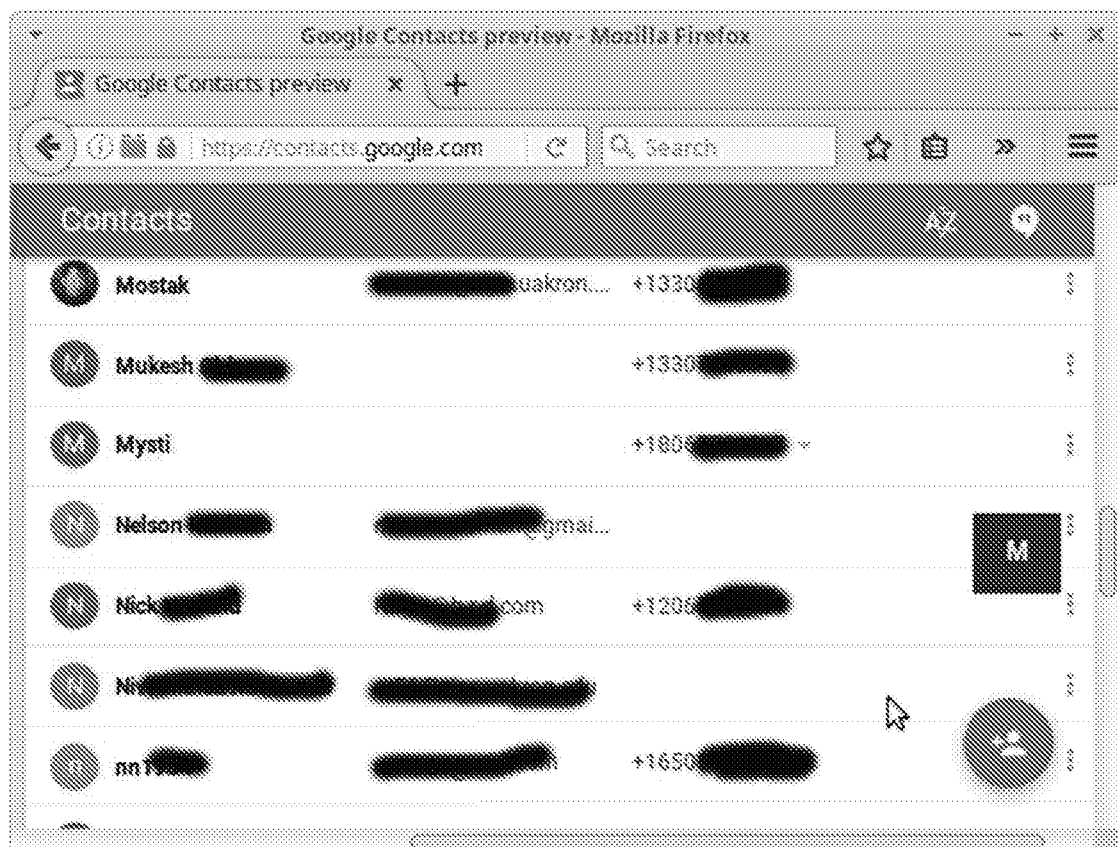
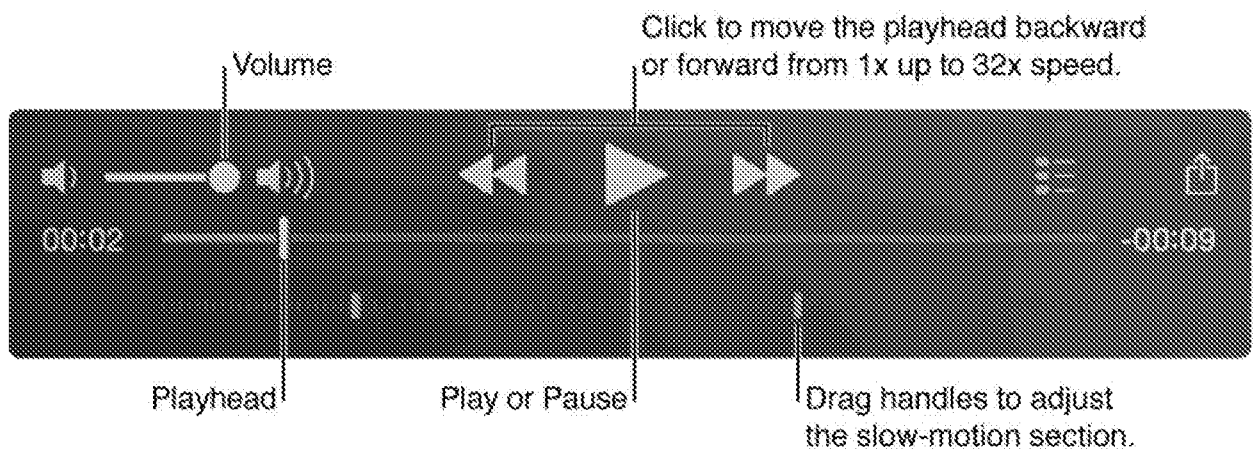
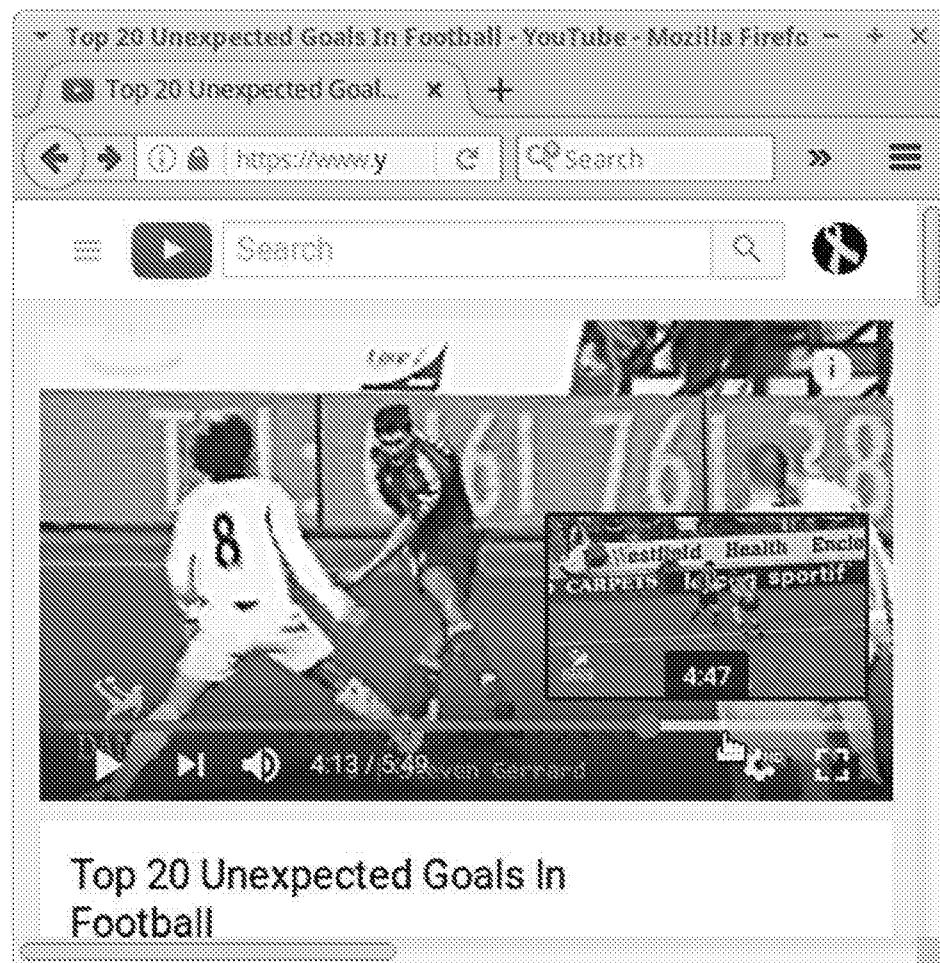
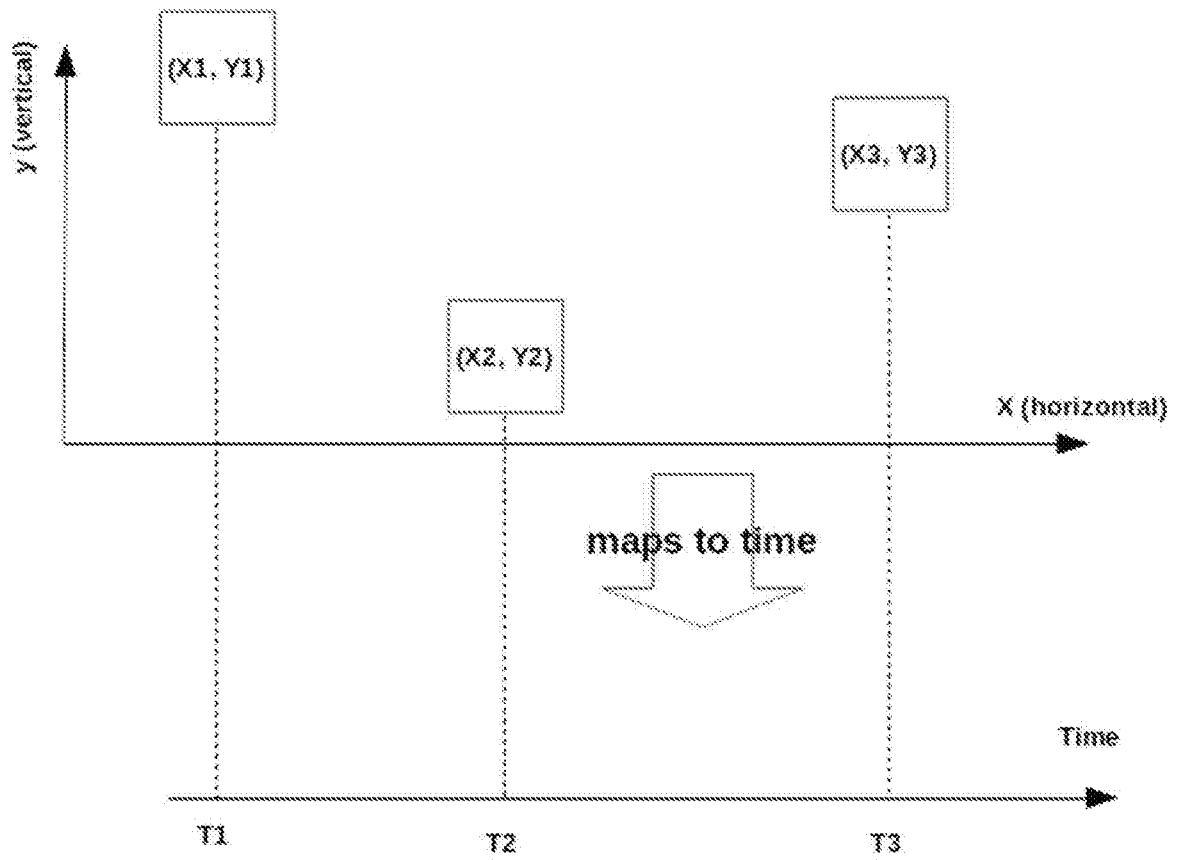
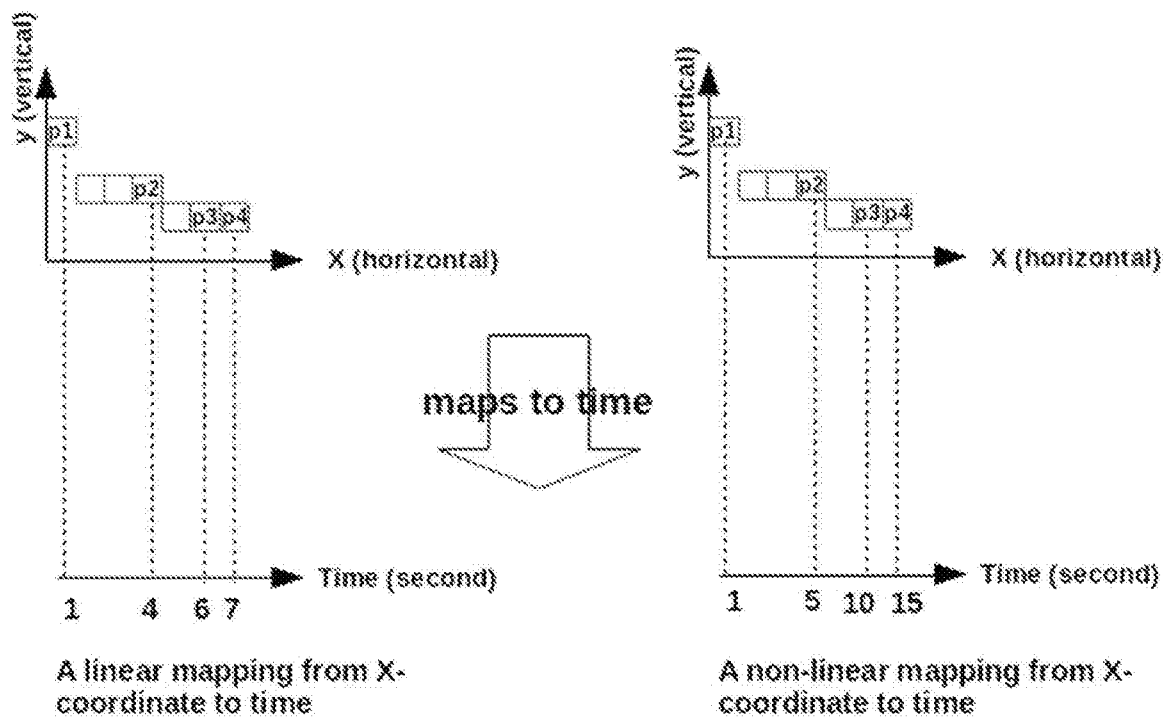


FIG. 59

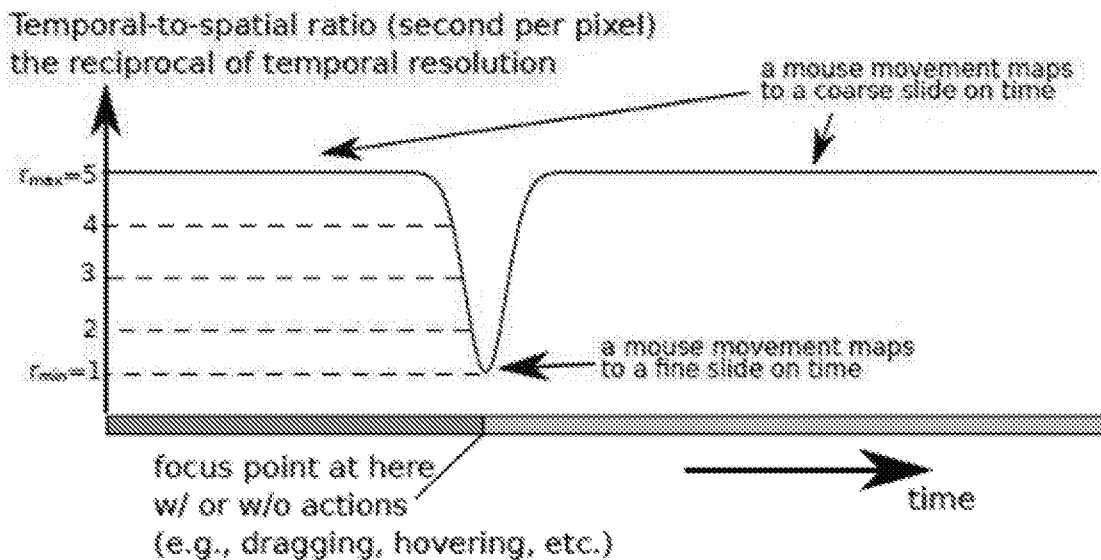
**FIG. 62**

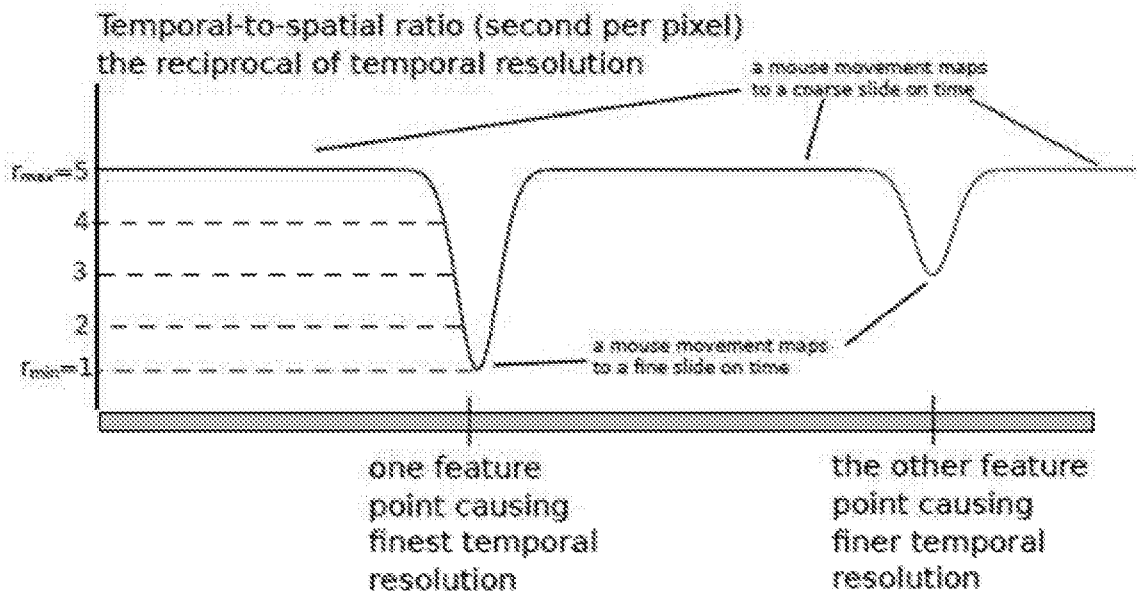
**FIG. 63**

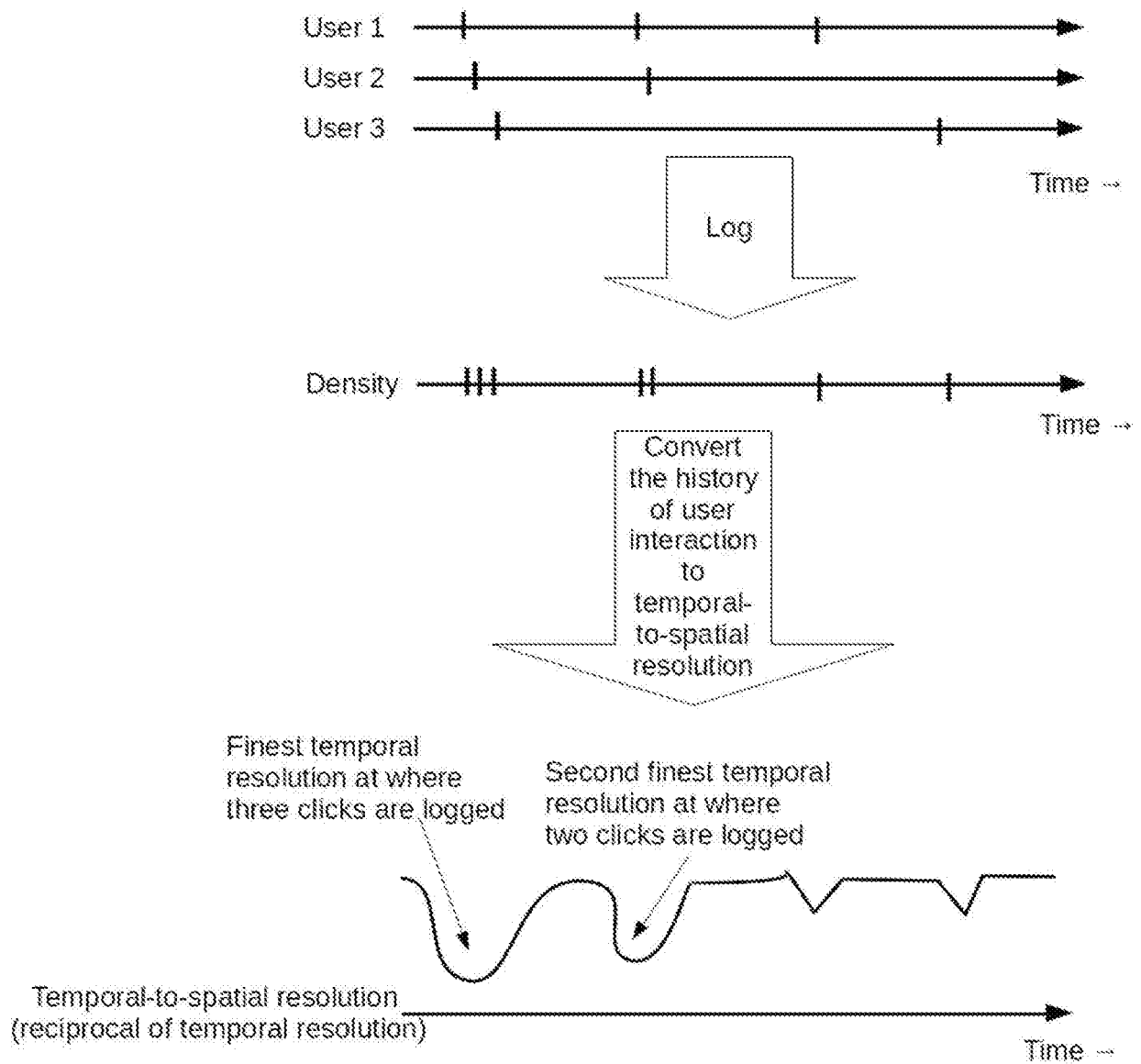
**FIG. 64**

**FIG. 65**

Let the minimum and maximum temporal-to-spatial ratios be r_{max} and r_{min} . Let the standard deviation of the Gaussian function be half of the video length $\sigma = \frac{\text{video length in second}}{2}$. Let the X coordinate, the coordinate along the time axis, of the focus point be μ . Then the temporal-to-spatial resolution can be computed as: $\left\lceil r_{min} + (r_{max} - r_{min}) \frac{1}{G(\sigma, \mu)} \right\rceil$, where $G(\sigma, \mu) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ is the probabilistic density function of Gaussian distribution, x is the distance from a pixel on time axis to the focus point in unit of pixel. The two ceiling brackets on both sides rounds the ratio to integers.

FIG. 66**FIG. 67**

**FIG. 68**

**FIG. 69**

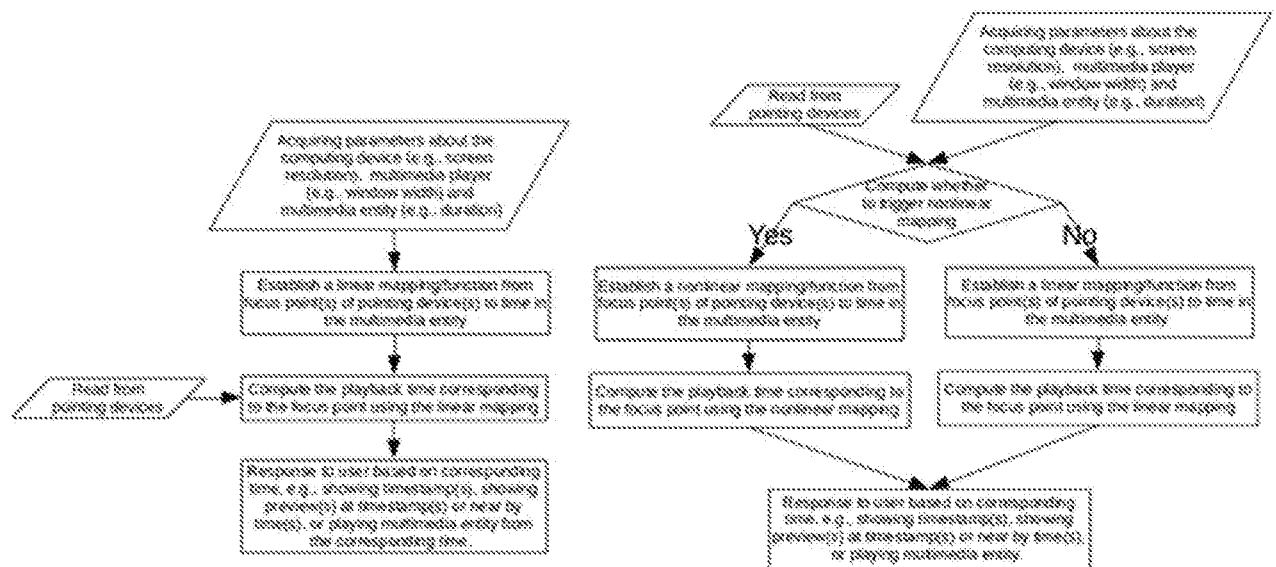
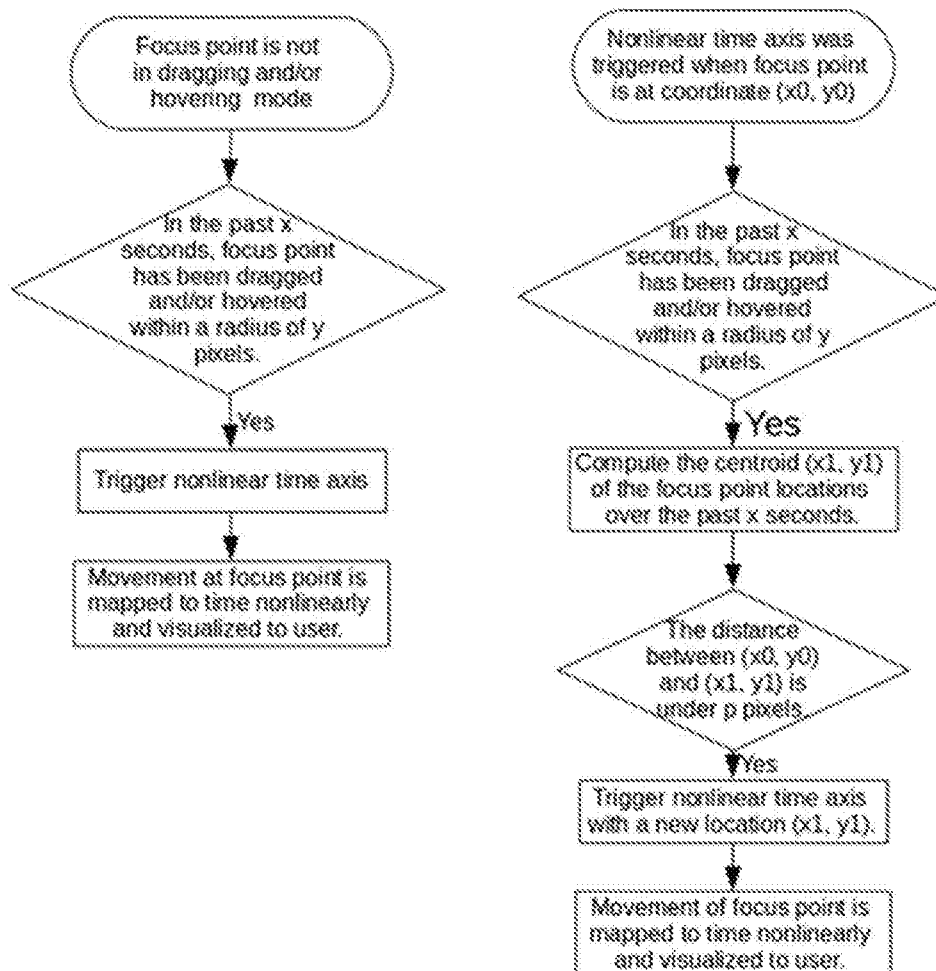
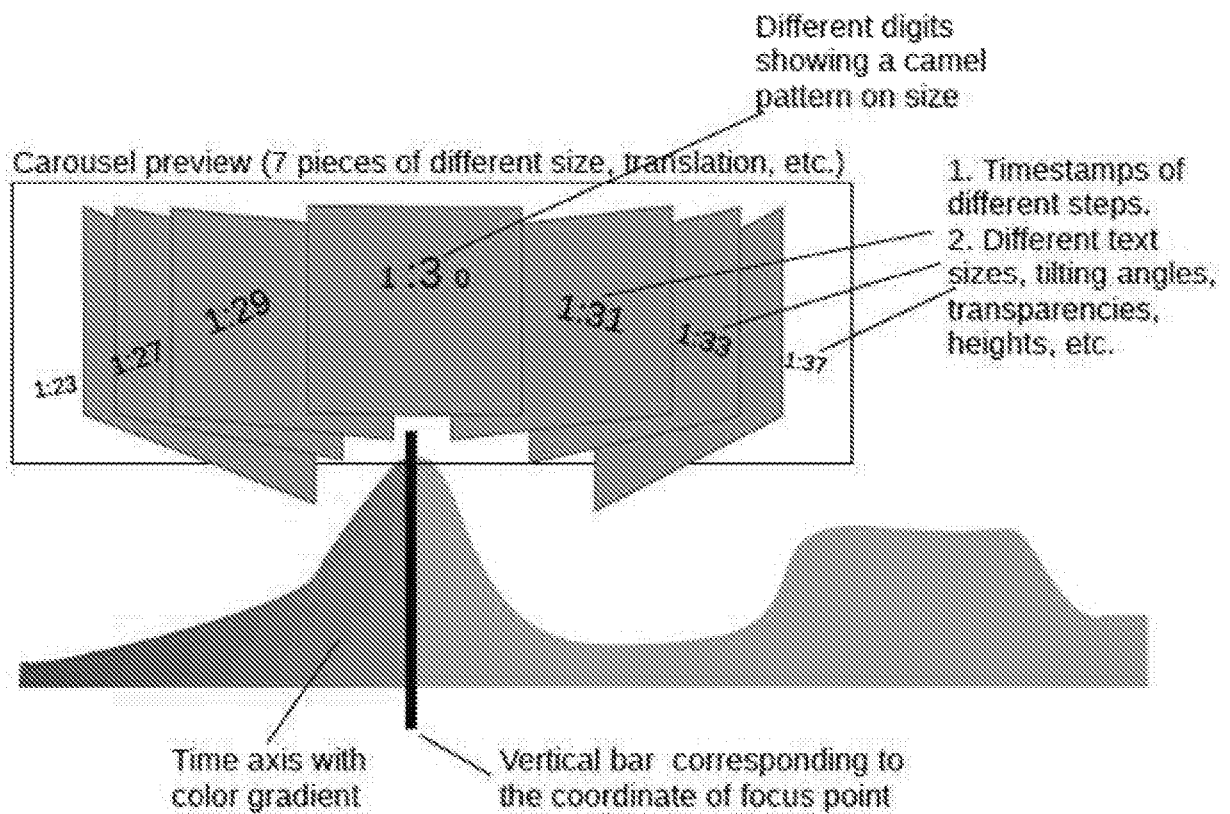


FIG. 70

**FIG. 71**

**FIG. 72**

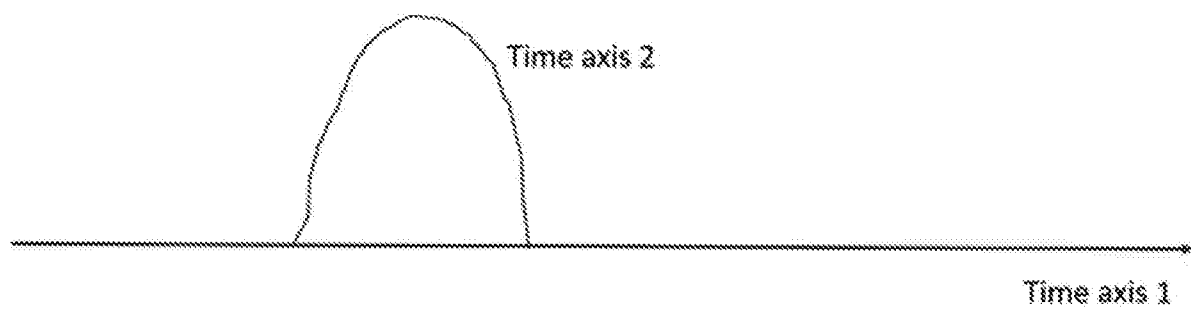


FIG. 73

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2017/013829

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

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

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

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

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

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

1-20, 54

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- ☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- ☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2017/013829

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F17/30
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal , WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	wo 2015/157711 AI (GOOGLE INC [US]) 15 October 2015 (2015-10-15) paragraph [0007] paragraph [0008] paragraph [0010] paragraph [0035] paragraph [0088] paragraph [0094] paragraph [0047] - paragraph [0050] paragraph [0069] paragraph [0087]	1-20,54
X	us 2015/293996 AI (LIU Y; LIU Y Y) 15 October 2015 (2015-10-15) paragraph [0006] - paragraph [0112] ; figure 13 ----- - / - -	1-20



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

23 February 2017

Date of mailing of the international search report

22/05/2017

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Papani kolaou, N

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2017/013829

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>AU 2009 212 772 A1 (CANON KK) 10 March 2011 (2011-03-10) the whole document</p> <p>-----</p>	1-20

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2017/013829

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2015157711	A1	15-10-2015	NONE

US 2015293996	A1	15--10--2015	EP 3129901 A1 15--02--2017
			US 2015293996 A1 15--10--2015

AU 2009212772	A1	10--03--2011	NONE

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

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

1. claims: 1-20, 54

a method of searching for media content

2. claims: 21-41

a method of automatically annotating media content

3. claims: 42-45

method of accessing media content using a user's voice or other user information

4. claims: 46-53

method of playing back media content
