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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 January 2017 (19.01.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/011798 A1

(51) International Patent Classification:
H04N 21/234 (2011.01) *H04N 21/81* (2011.01)
H04N 21/262 (2011.01) *H04N 21/845* (2011.01)
H04N 21/454 (2011.01)

(21) International Application Number:
PCT/US2016/042621

(22) International Filing Date:
15 July 2016 (15.07.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/193,322 16 July 2015 (16.07.2015) US

(71) Applicant: **VIZIO INSCAPE TECHNOLOGIES, LLC** [US/US]; 39 Tesla, Irvine, California 92618 (US).

(72) Inventors: **NEUMEIER, Zeev**; 39 Tesla, Irvine, California 92618 (US). **COLLETTE, Michael**; 19 Tesla, Irvine, California 92618 (US).

(74) Agents: **WONG, Claudine** et al.; KILPATRICK TOWNSEND & STOCKTON LLP, 1100 Peachtree Street, Suite 2800, Atlanta, Georgia 30309 (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, 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, KN, KP, KR, 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).

Published:
— *with international search report (Art. 21(3))*

(54) Title: DETECTION OF COMMON MEDIA SEGMENTS

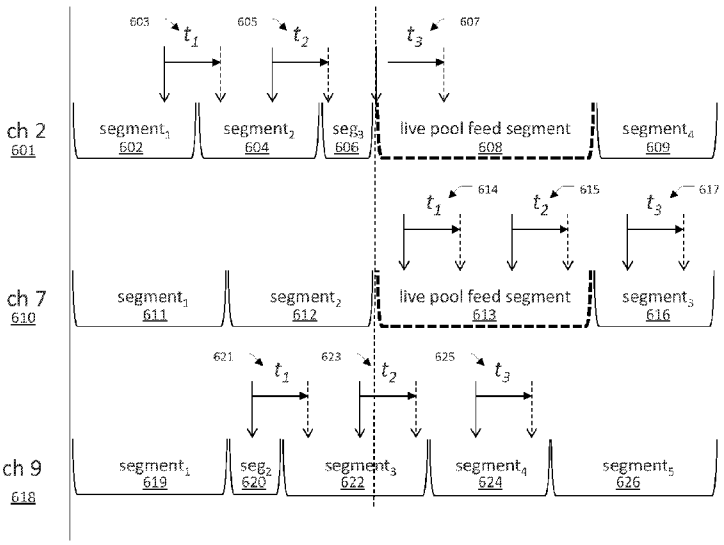


FIG. 6

(57) Abstract: Provided are systems, methods, and computer-program products for identifying a media content stream when the media content stream is playing an unscheduled media segment. A computing device may receive a plurality of media content streams, where at least two of the plurality of media content streams concurrently includes a same unscheduled media segment. The computing device may determine that the media display device is playing the unscheduled media segment by examining the media content available at the current time in each of the plurality of media content streams. The computing device may determine identification information from the media content included in the media content stream. The computing device may determine contextually-related content, which may be disabled while the unscheduled media segment is being played by the media display device. The computing device may display the media content stream and the contextually-related content after the unscheduled media segment has been played.

WO 2017/011798 A1

DETECTION OF COMMON MEDIA SEGMENTS

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] This application claims the benefit under 35 U.S.C. § 119(e) of U.S. Provisional Application No. 62/193,322, filed on July 16, 2015, the entirety of which is incorporated by reference.

[0002] This application is related to U.S. Patent Application No. 14/089,003 filed on November 25, 2013, now U.S. Patent No. 8,898,714, issued on November 25, 2014; U.S. Patent Application No. 14/217,075, now U.S. Patent no. 9,055,309, issued on June 9, 2015; U.S. Provisional Application No. 61/182,334 filed on May 29, 2009; U.S. Provisional Application No. 61/290,714 filed on December 29, 2009; U.S. Patent Application No. 12/788,748, now U.S. Patent No. 8,769,584, issued on July 1, 2014; and U.S. Patent Application No. 12/788,721, now U.S. Patent No. 595,781, issued on November 26, 2013, all of which are hereby incorporated by reference in their entirety.

BRIEF SUMMARY

[0003] Automatic Content Recognition (ACR) systems provide information about content being displayed at a particular point in time by a specific media display device. An ACR system may be implemented as a video matching system. Generally, a video matching system may operate by obtaining data samples from known video data sources, generating identification information from these samples, and storing identification information, along with known information about the video, in a database. A particular media devic can take data samples from an unknown video being displayed, generate identification information from the samples, and attempt to match the identification information with identification information stored in the database. When a match is found, the particular media device can receive, from the database, the information that is known about the video. Generally, the matching operations conducted by the media device can be conducted in real time, that is, as the video is being displayed on the device.

[0004] Provided are systems, methods, and computer-program products for identifying a media content stream when the media content stream is playing an unscheduled, common media

segment. In various implementations, a computing device may be configured to identify media content being played by a media display device at a particular time. The computing device may be configured to implement a video matching system. The computing device may receive a plurality of media content streams, where at least two of the plurality of media content streams concurrently includes a same unscheduled media segment. The computing device may be configured to determining that the media display device is playing the unscheduled media segment at a current time. To make this determination, the computing device may examine the media content available at the current time in each of the plurality of media content streams. The computing device may further be configured to determine identification information from the media content included in the media content stream that is being played by the media display device at the current time. The identification information may identify the media content stream. The computing device may further determine contextually-related content. The contextually-related content may be disabled while the unscheduled media segment is being played by the media display device. The computing device may further be configured to display the media content stream and the contextually-related content after the unscheduled media segment has been played.

[0005] In various implementations, the contextually-related content is selected using the identification information that identifies the media content stream. The contextually-related content may be provided to the media display device.

[0006] In various implementations, identifying the media content stream may include detecting a graphic superimposed onto the unscheduled media segment while the unscheduled media segment is being played by the media display device. The graphic may provide additional identification information for the media content stream.

[0007] In various implementations, media content used to determine the identification information may include media content included in the media content stream before or after the unscheduled media segment.

[0008] In various implementations, the computing device may be configured to determine that the media display device has been playing the unscheduled media segment since a beginning of the unscheduled media segment. In these implementations, the computing device may identify

the media content stream using identification information determined for media content included in the media content stream prior to the unscheduled media segment.

[0009] In various implementations, the computing device may be configured to determine that the media display device has been playing the unscheduled media segment since a point after a beginning of the unscheduled media segment. In these implementations, the computing device may identify the media content stream using identification information for media content included in the media content stream after the unscheduled media segment.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] Illustrative embodiments are described in detail below with reference to the following figures:

[0011] FIG. 1 illustrates a matching system that can identify unknown content;

[0012] FIG. 2 illustrates components of a matching system for identifying unknown data;

[0013] FIG. 3 illustrates an example of a video ingest capture system including a memory buffer 302 of a decoder;

[0014] FIG. 4 illustrates a video ingest capture system including a memory buffer of a decoder;

[0015] FIG. 5 illustrates an example of a video matching system for an interactive television system;

[0016] FIG. 6 illustrates an example where multiple content streams are carrying the same media segment at the same time;

[0017] FIG. 7 illustrates another example where multiple media streams, here illustrated as television channels, display the same media content at the same time;

[0018] FIG. 8 illustrates an example where multiple media content streams are carrying the same, common media segment at approximately the same time;

[0019] FIG. 9 illustrates one example of a graphic that has been superimposed onto a common media segment being played by a media display device;

[0020] FIG. 10 illustrates an example of a process that may be implemented when multiple media content streams include a same unscheduled media segment;

[0021] FIG. 11 is a chart illustrating point locations and the path points around them;

[0022] FIG. 12 is a chart illustrating a set of points that lie within radius 'r' distance from a query point 'x';

[0023] FIG. 13 is a chart illustrating possible point values where angles in 'n' dimension space are calculated as part of the candidate point qualification;

[0024] FIG. 14 is a chart illustrating a illustrating a combination of FIGS. 12 and 13, where a distance from query point 'x' and angle are applied to determine candidate 'suspects' for submission to the further matching system steps of path pursuit;

[0025] FIG. 15 is a chart illustrating self-intersecting paths and a query point; and

[0026] FIG. 16 is a chart illustrating three consecutive point locations and the path points around them.

DETAILED DESCRIPTION

[0027] Automatic Content Recognition (ACR) systems provide information about content being displayed at a particular point in time by a specific media display device. For example, an ACR system can provide information such as the channel being viewed, the title of the video, some text identifying the video or the content of the video, the portion of the video being viewed, one or more categories for the video, the author and/or producer of the video, and so on. This information can subsequently be used, for example, to provide viewership statistics for the video (e.g., how frequently the video has been viewed by how many people, at what times, etc.) and/or to suggest targeted content for the viewer, such as advertising or interactive content.

[0028] An ACR system may be implemented as a video matching system. Generally, a video matching system may operate by obtaining data samples from known video data sources, generating identification information from these samples, and storing identification information, along with known information about the video, in a database. A particular media device can use a similar process to identify unknown video content. Specifically, the media device can take data samples from an unknown video being displayed, generate identification information from the

samples, and attempt to match the identification information with identification information stored in the database. When a match is found, the particular media device can receive, from the database, the information that is known about the video. Generally, the matching operations conducted by the media device can be conducted in real time, that is, as the video is being displayed on the device.

[0029] A video matching system as described above may, however, have difficulty identifying the media content being displayed by a media device when the same content is being displayed on multiple channels at the same time. Furthermore, the system may not be able to determine, for example, which channel the viewer is watching, or what contextually-related information to provide.

[0030] One example of multiple channels displaying the same content at the same time occurs when multiple local televisions stations provide syndicated content to support "breaking news" stories. For example, a national broadcast agency (e.g., American Broadcasting Company (ABC), Columbia Broadcasting System (CBS), National Broadcasting Company (NBC), Fox Network, Cable News Network (CNN), etc.) may provide a video feed when political speeches, natural disasters, or human-made incidents occur. In this example, multiple national and/or local broadcast channels may pick up the video feed as it is being broadcast by the national broadcaster, and may re-broadcast the feed to local viewers. As result, multiple channels may be displaying identically video content at the same time. The same situation may occur in other contexts, such as when multiple channels display the same commercial or sporting event at the same time.

[0031] As noted above, a video matching system may rely on data samples collected from known video sources, such as local and national channels, where programming information may provide information such as the title or other identifying string for video content, and other information about the video content. When multiple channels display the same content, however, the video matching system may not be able to uniquely identify the video content. For example, the video matching system may associate one video with two or more channels. Subsequently, should a media device tune into one of the channels that is carrying the same content, the video matching system may not be able to determine which channel he media device has tuned into.

[0032] In various implementations, a video matching system may be configured to improve the accuracy of automated content recognition in the presence of ambiguity caused by a common video segment appearing simultaneously on multiple channels. Without accuracy improvements, the presence of common video segments displaying simultaneously on multiple channels being monitored may cause ambiguity in identifying content being displayed by a media device. In various implementations, the video matching system may use information from media content displayed on the media display device before and/or after the media device began to display the common video segment. Using this information, the video matching system is able to attach identification information to samples taken from the common video segment.

[0033] On national and local channels that may at some point display a common video segment, the channel may provide some uniquely identifiable content prior to displaying the common video segment. In various implementations, this unique content can be used to assist in identifying the common video segment. For example, in certain broadcast news, there are parts of the news program that are known to be from the local channel. For example, the news program may introduce and/or comment on the common video segment before the common video segment is displayed. The introductory segment may be referred to as "talking heads" segment, that is, a segment where two or more people are seated behind a desk and are framed in the video from the center chest up. When a "talking heads" segment is detected, a video matching system can, for example, add a new timeline signal or events that can be used to assist in identifying the common video segment.

[0034] In various implementations, a media display device that is configured to use the video matching system may obtain and track timeline signals or events, such as those that may be generated for "talking heads" segments. In these implementations, the media device may check for a timeline event when the media device encounters an unknown and possibly common video segment. In one case, the media device may not find a timeline event, which may indicate to the media device that the device has just been turned on or tuned in to the channel displaying the common video content. In this case, the data collection process of the media device may be configured to avoid using the common video segment for identification purposes. In other cases, the media device may use timeline events received before and/or after the common video segment to generate identification information for the common video segment.

[0035] In various implementations, the video matching may also use timeline events to more rapidly identify commercials. Generally, the amount of time a video matching system takes to obtain sufficient samples to match a known commercial with a high-degree of probability may be referred to as the "commercial confidence interval." Using the techniques described herein, te commercial confidence interval may be reduced when the media device uses timeline events within a specified window of past time.

I. AUDIO-VIDEO CONTENT

[0036] In various implementations, a video content matching system may be configured to identify media content being displayed by a media display device. In various implementations, the video content system can be used to provide contextually targeted content to a media display devices, where the targeted content is selected based the identified media content.

[0037] Media content includes video, audio, text, graphics, tactile representations of visual and/or audible data, and various combinations of visual, audible, or tactile information. For example, media content can include synchronized audio and video, such as movies or television shows. As another example, media content can include text and graphics, such as web pages. As another example, media content can include photographs and music, such as a photo slide show with a soundtrack.

[0038] A media display device can be a device capable of displaying various media content. Media display devices may include, for example, televisions systems. Television (TV) systems include, for example, televisions such as web TVs and connected TVs (also known as "Smart TVs"), and optionally equipment incorporated in or co-located with the TV, such as set-top boxes (STB), a digital video disc (DVD) players, and/or digital video recorders (DVR).

[0039] Connected TVs are TVs that are connected to a network, such as the Internet. In various implementations, a network connected TV may be connected to a local wired or wireless network, such as for example in a private home or in a business office. A Connected TV can run an application platform such as Google's Android, or some other platform configured to provide interactive, smartphone or tablet-like software applications, which also may be referred to as "apps."

[0040] In various implementations, a media display device may receive signals, such as television signals. Televisions signals include, for example, signals representing video and audio data, broadcast together and synchronized for simultaneous display. For example, television signals may include television programs and/or commercials. In some cases, television signals may include additional information relating to the audio-video content in a television signals. This additional data may be referred to as "metadata." The term "metadata" may also be used to describe information that is associated with video or audio-video content transmitted other than as television signals (e.g. transmitted as digitized and/or packetized data over a network). Metadata may include information about the content, such as information identifying the content, a description of the content, one or more categories for the content, an author and/or publisher of the content, and so on. Because metadata is transmitted with the content that the metadata is associated with, the metadata can be used to provide information about the content as the content is being viewed or played.

[0041] Not all media display devices have access to metadata. Hence, not all media display devices are able to determine what they are displaying or playing at any given moment. Without this information, a media display device may not be able to provide customized or individualized content or advertisements for a specific viewer. While some information about content being provide to a media display device may be available in the distribution pipeline, this information may be lost or removed by the time the content arrives at the media display device.

[0042] In some implementations, metadata may be provided with audio-video content using various methods. For example, in some implementations, identification information may be encoded into the content using watermarks. In these implementations, the identification information may be encoded so that the information is not lost when the content is compressed for transmission and decompressed for display. Such methods, however, may require that a receiving media display device be able to extract the identification information from the content. Additionally, these methods may not enable up-to-the-moment identification of a particular video being played, with fraction-of-a-second identification capability.

[0043] In various implementations, advancements in fiber optic and digital transmission technology have enabled the media industry to provide a large channel capacity, where "channels" include traditional broadcast channels, satellite signals, digital channels, streaming

content, and/or user-generated content. In some cases, media providers such as satellite systems may be referred to as Multichannel Video Programming Distributors (MVPD). In some implementations, media providers are also able to use the increase data capacity of modern transmission systems to provide some of interactive content, such as interactive television (ITV). The increased processing power of Smart TVS, set-top boxes, and similar devices may further enable interactive content.

[0044] Interactive television may enable television systems to serve as a two-way information distribution mechanism in a manner similar to the World Wide Web. Interactive televisions may provide a variety of marketing, entertainment, and educational capabilities, such as for example enabling a viewer to order an advertised product or service, compete against contestants in a game show, participate in a live classroom session, and so on. In some implementations, the interactive functionality may be controlled by a set-top box. In these implementations, the set-top box may execute an interactive program associated with video content, such as a TV broadcast. Interactive functionality may be displayed on the TV's screen and may include icons or menus to allow a viewer to make selections via the TV's remote control or a keyboard.

[0045] In various implementations, interactive content may be incorporated into audio-video content. In some implementations, the audio-video content may consist of a broadcast stream. A broadcast stream may also be referred to as a "channel" or a "network feed". The term "broadcast stream" may refer to a broadcast signal received by a television over, for example, an antenna, a satellite, a coaxial cable, a digital subscriber line (DSL) cable, a fiber optic cable, or some other transmission medium. In various implementations, interactive content may be incorporated into a audio-video content using "triggers." Triggers may be inserted into the content for a particular program. Content that includes triggers may be referred to as "enhanced program content" or an "enhanced television program" or and "enhanced video signal." Triggers may be used to alert the media display device (e.g., at a set-top box or a the processor in a Smart TV) that interactive content is available. The trigger may contain information about available content as well as where the interactive content can be found (e.g., a memory address, a network address, and/or a website address). A trigger may also contain information that can be displayed on the media display device to the viewer. For example, information provided by the trigger, may be displayed at the bottom of a screen provided by the media display device. The displayed

information may prompt the viewer to perform some action or choose amongst multiple of options.

II. VIDEO MATCHING

[0046] In various implementations, a video content system may be configured to identify media content that is being displayed or played by a media display device. In various implementations, information identifying the content being viewed at a particular moment in time can be used to capture and appropriately respond to a viewer's specific reaction, such as requesting that the content be rewound or requesting that a video be restarted from its beginning. Alternatively or additionally, the identification information can be used to trigger targeted content, such as advertisements, which may be provided by the content provider or an advertiser. Information identifying audio-video content can thus be used to provide viewer-customized, video-on-demand (VoD) capabilities to devices that otherwise do not have Smart TV capabilities.

[0047] In various implementations, a video segment be identified by sampling, at periodic intervals, a subset of the pixel data being displayed on the screen of a media display device, and then finding similar pixel data in a content database. In some implementations, a video segment may be identified by extracting audio data associated with the video segment and finding similar audio data in a content database. In some implementations, a video segment may be identified by processing the audio data associated with the video segment using automated speech recognition techniques, and searching text transcriptions from known video content to locate matching text phrases. In some implementations, a video segment may be identified by processing metadata associated with the video segment.

[0048] In various implementations, a video matching system may be used to provide contextually targeted content to an interactive media display system. The contextually targeted content may be based on identification of a video segment being displayed, and also on the time at which the video segment is being played (day or evening, 3:00 in the afternoon, etc.) and/or the portion of the video segment that is currently being displayed (e.g., a current offset from the beginning of the video). Here, "playing time" and "offset time" may be used interchangeably to describe the part of a video that is currently being displayed.

[0049] In various implementations, a media display device equipped with a content matching system, having identified content that is presently being displayed or played by the media display device, may be able to deduce the subject matter of the content, and interact with the viewer accordingly. For example, the media display device may be able to provide instant access to video-on-demand versions of content, and/or to higher resolutions or 3D formats of the content. Additionally, the media display device may provide the ability to start over, fast forward, pause and re-wind the content. In various implementations, advertising can be included in the content. In these implementations, some or all advertising messages can be customized, such as for example to viewer's geographic location, demographic group, or shopping history. Alternatively or additionally, advertisements can be reduced in number or length, or be eliminated entirely.

[0050] In various implementations, once a video segment is identified, the offset time may be determined by sampling a subset of the pixel data (or associated audio data) being displayed or played by a media display device, and finding similar pixel (or audio) data in a content database. In various implementations, the offset time can be determined by extracting audio or image data associated with such video segment and finding similar audio or image data in a content database. In various implementations, the offset time can be determined by processing the audio data associated with such video segment using automated speech recognition techniques. In various implementations, the offset time can determined by processing metadata associated with such video segment.

[0051] In various implementations, a system for video matching can be included in a television system. In various implementations, the television system includes a connected TV. In various implementations, a video matching system may be included in part in a connected television and in part on a server connected to the connected television over the Internet.

[0052] FIG. 1 illustrates a matching system 100 that can identify unknown content. In some examples, the unknown content can include one or more unknown data points. In such examples, the matching system 100 can match unknown data points with reference data points to identify unknown video segments associated with the unknown data points. The reference data points can be included in a reference database 116.

[0053] The matching system 100 includes a client device 102 and a matching server 104. The client device 102 includes a media client 106, an input device 108, an output device 110, and one

or more contextual applications 126. The media client 106 (which can include a television system, a computer system, or other electronic device capable of connecting to the Internet) can decode data (e.g., broadcast signals, data packets, or other frame data) associated with video programs 128. The media client 106 can place the decoded contents of each frame of the video into a video frame buffer in preparation for display or for further processing of pixel information of the video frames. The client device 102 can be any electronic decoding system that can receive and decode a video signal. The client device 102 can receive video programs 128 and store video information in a video buffer (not shown). The client device 102 can process the video buffer information and produce unknown data points (which can referred to as "cues"), described in more detail below with respect to FIG. 3. The media client 106 can transmit the unknown data points to the matching server 104 for comparison with reference data points in the reference database 116.

[0054] The input device 108 can include any suitable device that allows a request or other information to be input to the media client 106. For example, the input device 108 can include a keyboard, a mouse, a voice-recognition input device, a wireless interface for receiving wireless input from a wireless device (e.g., from a remote controller, a mobile device, or other suitable wireless device), or any other suitable input device. The output device 110 can include any suitable device that can present or otherwise output information, such as a display, a wireless interface for transmitting a wireless output to a wireless device (e.g., to a mobile device or other suitable wireless device), a printer, or other suitable output device.

[0055] The matching system 100 can begin a process of identifying a video segment by first collecting data samples from known video data sources 118. For example, the matching server 104 collects data to build and maintain a reference database 116 from a variety of video data sources 118. The video data sources 118 can include media providers of television programs, movies, or any other suitable video source. Video data from the video data sources 118 can be provided as over-the-air broadcasts, as cable TV channels, as streaming sources from the Internet, and from any other video data source. In some examples, the matching server 104 can process the received video from the video data sources 118 to generate and collect reference video data points in the reference database 116, as described below. In some examples, video programs from video data sources 118 can be processed by a reference video program ingest

system (not shown), which can produce the reference video data points and send them to the reference database 116 for storage. The reference data points can be used as described above to determine information that is then used to analyze unknown data points.

[0056] The matching server 104 can store reference video data points for each video program received for a period of time (e.g., a number of days, a number of weeks, a number of months, or any other suitable period of time) in the reference database 116. The matching server 104 can build and continuously or periodically update the reference database 116 of television programming samples (e.g., including reference data points, which may also be referred to as cues or cue values). In some examples, the data collected is a compressed representation of the video information sampled from periodic video frames (e.g., every fifth video frame, every tenth video frame, every fifteenth video frame, or other suitable number of frames). In some examples, a number of bytes of data per frame (e.g., 25 bytes, 50 bytes, 75 bytes, 100 bytes, or any other amount of bytes per frame) are collected for each program source. Any number of program sources can be used to obtain video, such as 25 channels, 50 channels, 75 channels, 100 channels, 200 channels, or any other number of program sources. Using the example amount of data, the total data collected during a 24-hour period over three days becomes very large. Therefore, reducing the number of actual reference data point sets is advantageous in reducing the storage load of the matching server 104.

[0057] The media client 106 can send a communication 122 to a matching engine 112 of the matching server 104. The communication 122 can include a request for the matching engine 112 to identify unknown content. For example, the unknown content can include one or more unknown data points and the reference database 116 can include a plurality of reference data points. The matching engine 112 can identify the unknown content by matching the unknown data points to reference data in the reference database 116. In some examples, the unknown content can include unknown video data being presented by a display (for video-based ACR), a search query (for a MapReduce system, a Bigtable system, or other data storage system), an unknown image of a face (for facial recognition), an unknown image of a pattern (for pattern recognition), or any other unknown data that can be matched against a database of reference data. The reference data points can be derived from data received from the video data sources

118. For example, data points can be extracted from the information provided from the video data sources 118 and can be indexed and stored in the reference database 116.

[0058] The matching engine 112 can send a request to the candidate determination engine 114 to determine candidate data points from the reference database 116. A candidate data point can be a reference data point that is a certain determined distance from the unknown data point. In some examples, a distance between a reference data point and an unknown data point can be determined by comparing one or more pixels (e.g., a single pixel, a value representing group of pixels (e.g., a mean, an average, a median, or other value), or other suitable number of pixels) of the reference data point with one or more pixels of the unknown data point. In some examples, a reference data point can be the certain determined distance from an unknown data point when the pixels at each sample location are within a particular pixel value range.

[0059] In one illustrative example, a pixel value of a pixel can include a red value, a green value, and a blue value (in a red-green-blue (RGB) color space). In such an example, a first pixel (or value representing a first group of pixels) can be compared to a second pixel (or value representing a second group of pixels, where the second group of pixels is located in the same display buffer position as the first group of pixels) by comparing the corresponding red values, green values, and blue values respectively, and ensuring that the values are within a certain value range (e.g., within 0-5 values). For example, the first pixel can be matched with the second pixel when (1) a red value of the first pixel is within 5 values in a 0-255 value range (plus or minus) of a red value of the second pixel, (2) a green value of the first pixel is within 5 values in a 0-255 value range (plus or minus) of a green value of the second pixel, and (3) a blue value of the first pixel is within 5 values in a 0-255 value range (plus or minus) of a blue value of the second pixel. In such an example, a candidate data point is a reference data point that is an approximate match to the unknown data point, leading to multiple candidate data points (related to different media segments) being identified for the unknown data point. The candidate determination engine 114 can return the candidate data points to the matching engine 112.

[0060] For a candidate data point, the matching engine 112 can add a token into a bin that is associated with the candidate data point and that is assigned to an identified video segment from which the candidate data point is derived. A corresponding token can be added to all bins that correspond to identified candidate data points. As more unknown data points (corresponding to

the unknown content being viewed) are received by the matching server 104 from the client device 102, a similar candidate data point determination process can be performed, and tokens can be added to the bins corresponding to identified candidate data points. Only one of the bins corresponds to the segment of the unknown video content being viewed, with the other bins corresponding to candidate data points that are matched due to similar data point values (e.g., having similar pixel color values), but that do not correspond to the actual segment being viewed. The bin for the candidate video content segment that corresponds to the unknown video segment being viewed will have more tokens assigned to it than other bins for segments that do not correspond to the unknown video segment. For example, as more unknown data points are received, a larger number of reference data points that correspond to the bin are identified as candidate data points, leading to more tokens being added to the bin. Once a bin includes a particular number of tokens – that is, the bin reaches a predetermined threshold – the matching engine 112 can determine that the video segment associated with the bin is currently being displayed on the client device 102. A video segment can include an entire video program or a portion of the video program. For example, a video segment can be a video program, a scene of a video program, one or more frames of a video program, or any other portion of a video program.

[0061] FIG. 2 illustrates components of a matching system 200 for identifying unknown data. For example, the matching engine 212 can perform a matching process for identifying unknown content (e.g., unknown media segments, a search query, an image of a face or a pattern, or the like) using a database of known content (e.g., known media segments, information stored in a database for searching against, known faces or patterns, or the like). For example, the matching engine 212 receives unknown data content 202 (which can be referred to as a "cue") to be matched with a reference data point of the reference data points 204 in a reference database. The unknown data content 202 can also be received by the candidate determination engine 214, or sent to the candidate determination engine 214 from the matching engine 212. The candidate determination engine 214 can conduct a search process to identify candidate data points 206 by searching the reference data points 204 in the reference database. In one example, the search process can include a nearest neighbor search process to produce a set of neighboring values (that are a certain distance from the unknown values of the unknown data content 202). The nearest neighbor search process is discussed in further detail below. The candidate data points 206 are input to the matching engine 212 for conducting the matching process to generate a

matching result 208. Depending on the application, the matching result 208 can include video data being presented by a display, a search result, a determined face using facial recognition, a determined pattern using pattern recognition, or any other result.

[0062] In determining candidate data points 206 for an unknown data point (e.g., unknown data content 202), the candidate determination engine 214 determines a distance between the unknown data point and the reference data points 204 in the reference database. The reference data points that are a certain distance from the unknown data point are identified as the candidate data points 206. In some examples, a distance between a reference data point and an unknown data point can be determined by comparing one or more pixels of the reference data point with one or more pixels of the unknown data point, as described above with respect to FIG. 1. In some examples, a reference data point can be the certain distance from an unknown data point when the pixels at each sample location are within a particular value range. As described above, a candidate data point is a reference data point that is an approximate match to the unknown data point, and because of the approximate matching, multiple candidate data points (related to different media segments) are identified for the unknown data point. The candidate determination engine 114 can return the candidate data points to the matching engine 112.

[0063] FIG. 3 illustrates an example of a video ingest capture system 400 including a memory buffer 302 of a decoder. The decoder can be part of the matching server 104 or the media client 106. The decoder may not operate with or require a physical television display panel or device. The decoder can decode and, when required, decrypt a digital video program into an uncompressed bitmap representation of a television program. For purposes of building a reference database of reference video data (e.g., reference database 316), the matching server 104 can acquire one or more arrays of video pixels, which are read from the video frame buffer. An array of video pixels is referred to as a video patch. A video patch can be any arbitrary shape or pattern but, for the purposes of this specific example, is described as a 10x10 pixel array, including ten pixels horizontally by ten pixels vertically. Also for the purpose of this example, it is assumed that there are 25 pixel-patch positions extracted from within the video frame buffer that are evenly distributed within the boundaries of the buffer.

[0064] An example allocation of pixel patches (e.g., pixel patch 304) is shown in FIG. 3. As noted above, a pixel patch can include an array of pixels, such as a 10x10 array. For example, the

pixel patch 304 includes a 10x10 array of pixels. A pixel can include color values, such as a red, a green, and a blue value. For example, a pixel 306 is shown having Red-Green-Blue (RGB) color values. The color values for a pixel can be represented by an eight-bit binary value for each color. Other suitable color values that can be used to represent colors of a pixel include luma and chroma (Y, Cb, Cr, also called YUV) values or any other suitable color values.

[0065] A mean value (or an average value in some cases) of each pixel patch is taken, and a resulting data record is created and tagged with a time code (or time stamp). For example, a mean value is found for each 10x10 pixel patch array, in which case twenty-four bits of data per twenty-five display buffer locations are produced for a total of 600 bits of pixel information per frame. In one example, a mean of the pixel patch 304 is calculated, and is shown by pixel patch mean 308. In one illustrative example, the time code can include an "epoch time," which representing the total elapsed time (in fractions of a second) since midnight, January 1, 1970. For example, the pixel patch mean 308 values are assembled with a time code 412. Epoch time is an accepted convention in computing systems, including, for example, Unix-based systems. Information about the video program, known as metadata, is appended to the data record. The metadata can include any information about a program, such as a program identifier, a program time, a program length, or any other information. The data record including the mean value of a pixel patch, the time code, and metadata, forms a "data point" (also referred to as a "cue"). The data point 310 is one example of a reference video data point.

[0066] A process of identifying unknown video segments begins with steps similar to creating the reference database. For example, FIG. 4 illustrates a video ingest capture system 400 including a memory buffer 402 of a decoder. The video ingest capture system 400 can be part of the client device 102 that processes data presented by a display (e.g., on an Internet-connected television monitor, such as a smart TV, a mobile device, or other television viewing device). The video ingest capture system 400 can utilize a similar process to generate unknown video data point 410 as that used by system 300 for creating reference video data point 310. In one example, the media client 106 can transmit the unknown video data point 410 to the matching engine 112 to identify a video segment associated with the unknown video data point 410 by the matching server 104.

[0067] As shown in FIG. 4, a video patch 404 can include a 10x10 array of pixels. The video patch 404 can be extracted from a video frame being presented by a display. A plurality of such pixel patches can be extracted from the video frame. In one illustrative example, if twenty-five such pixel patches are extracted from the video frame, the result will be a point representing a position in a 75-dimension space. A mean (or average) value can be computed for each color value of the array (e.g., RGB color value, Y, Cr, Cb color values, or the like). A data record (e.g., unknown video data point 410) is formed from the mean pixel values and the current time is appended to the data. One or more unknown video data points can be sent to the matching server 104 to be matched with data from the reference database 116 using the techniques described above.

III. COMMON VIDEO SEGMENTS

[0068] FIG. 5 illustrates an example of a video matching system 500 for an interactive television (TV) system 501. An interactive television system 501 is given as an example of a media display device. In this example, the interactive television system 501 includes a TV client 503 and a contextual targeting client 502. The TV client 503 may receive television programming 505. The television programming 505 may include audio-video content, which may include audio, video, and/or video data with synchronized audio data. In some cases, the television programming 505 may include metadata. The TV client 503 may be configured to display the audio-video content, including displaying video content on a screen and/or playing audio content through speakers. The television programming 505 may be received from various sources, including broadcast television providers, satellite television providers, Internet media providers, audio-video playing devices (e.g. DVD players, DVR players, VCR players, etc.), and so on.

[0069] In various implementations, the video matching system may include a matching server 509. The matching server 509 may recognize media being displayed or played by the interactive television system 501. To provide video matching services, the matching server 509 may receive known media cue data 517 from an ingest server 520. The ingest server 520 may obtain data from various known sources, such as Video-on-Demand (VoD) content feeds 515a, local channel feeds 515b, and national channel feeds 515c. Each of these known sources may provide media data (e.g., video and/or audio) as well as information identifying the media data, such as

programming guides or metadata. In various implementations, the ingest server 520 may generate known media cue data 517 from the media data received from the various sources. The ingest server 520 may provide the known media cue data 517 to various recipients, including the matching server 509.

[0070] In various implementations, the ingest server 520 may also provide programming identification and time data 514. In various implementations, the programming identification and time data 514 is synchronized with the known media cue data 517, meaning that the programming identification and time data 514 identifies the known media cues 517 and/or provides the times at which the media associated with the known media cues are expected to be displayed. The program identification and time data 514 may also be called metadata.

[0071] In various implementations, the known media cue data 517 provide a cue or key for identifying video and/or audio data. The known media cue data 517 may have been taken from known audio-video media, such that the known media cues 517 can be associated with the name and/or some other identification information for the known audio-video media. As described in further detail below, the known media cues 517 can be matched against similar cues taken from media being displayed or played by the interactive television system 501. The matching server 509 may store the known media cue data 517 and the programming identification data 514 in a database 512.

[0072] In various implementations, the matching server 509 may include a channel recognition system 510. The channel recognition system 510 may receive unknown media cues 507a from the interactive television system 501. For example, the TV client 503 may take samples from audio-video data being displayed or played at any given time, and may generate cues from the samples. The TV client 503 may provide these cues, as unknown media cues 507a, to the channel recognition system 510 in the matching server 509. The channel recognition system 510 may then match the unknown media cues 507a against known media cues 517 to identify the media being displayed or played by the interactive television system 501.

[0073] In various implementations, the channel recognition system 510 determine a program identification for the unknown media cues 507a. The program identification may include a name or description, or some other information that identifies the media content being displayed by the interactive television system 501. The channel recognition system 510 may also provide a time,

where the time indicates he time at which the media was played by the interactive television system 501

[0074] In various implementations, the channel recognition system 510 may provide the program identification and time data 513 to a contextual targeting manager 511. Using the program identification and time data 513, the contextual manager 511 may determine contextually-related content 507b, including for example applications and advertising. The contextual targeting manager 511 may provide the contextually-related content 507b to the interactive television system 501. For example, the interactive television system 501 may include a contextual targeting engine 502 for managing the contextually-related content 507b. In some implementations, the contextual targeting manager 511 may also provide event triggers 507c to the contextual targeting system 502. The event trigger 507c may instruct the contextual targeting engine 502 to play or display the contextually-related content 507b. For example, an event trigger 507c may instruct the contextual targeting engine 502 to display contextually-related information overlays, where the information overlays are coordinated with the display of video content to which the information overlays are related. Alternatively or additionally, the event triggers 507c may cause substitute media, such as targeted advertising, to be displayed. In some implementations, the contextual targeting engine 502 may provide event confirmation 507d to the contextual targeting manager 511, indicating that the instructions provided by the event trigger 507c has been executed.

[0075] In various implementations, the contextual targeting client 502 may alternatively or additionally provide viewership information to the matching server 509. For example, the contextual targeting client 502 may provide viewership information in addition to or instead of event confirmations 507d. In this implementations, the viewership information may include, for example, information about how often a particular media segment was played, what time of day or day of the week the media segment was played, what was played before and/or after the media segment, and/or on what channel the media segment was played. In some cases, the viewership information may also include information about a viewer, such as demographic information.

[0076] In various implementations, a media display device may configured with or connected to a video matching system. The video matching system may be able to identify media being displayed or played by the media display device at any given moment in time. As discussed

above, the video matching system may take video and/or audio samples for the media being played by the device, generate identifiers or "cues" from the samples, and then match the cues against a database. by identifying the media being displayed or played on a media display device, the video matching system may be able to provide contextually-related content, including applications, advertisements, and/or alternate media content.

[0077] When multiple content streams or channels that are available to a media display device play the same content, such as for example "breaking news," the content may not be uniquely identifiable. For example, without additional information, it may not be clear which channel is being displayed by the media display device.

[0078] FIG. 6 illustrates an example where multiple content streams are carrying the same media segment at the same time. FIG. 6 further illustrates examples of methods by which video matching system can accommodate multiple channels carrying the same media segment. In the illustrated example, three channels are given as examples of the multiple content streams. The three channels could be, for example, three broadcast television channels.

[0079] In this example, channel 2 601 plays two segments 602, 604 of regularly scheduled programming. Assuming that a media display device is playing channel 2 601, during two time intervals t1 603 and t2 605, the media display device sends samples from segment 1 602 and segment 2 604 to the video matching system. By the end of time interval t1 603, the video matching system is able to identify segment 1 602, and by the end of time interval t2 605, the video matching system is able to identify segment 2 604.

[0080] During a third segment 606, channel 2 601 is interrupted by common media segment, here a live pool feed segment 608. The live pool feed segment 608 of this example is common media segment being provided by, for example, a national broadcaster. The live pool feed segment 608 may be made available to each of its syndicated stations. An example of a live pool feed segment is "breaking news," that is, a national news story. Other example of a live pool feed segments include sporting events, syndicated programs, and commercials. During a time interval t3 607, the media display device may send samples from the live pool feed segment 608 to the video matching system.

[0081] The video matching system may determine that the samples provided during time interval t3 607 were for a live pool feed segment 608. In various implementations, the video matching system may make this determination based on finding matching cues for the live pool feed segment that are associated with multiple channels. Upon determining that channel 2 601 is displaying the live pool feed segment 608, in some implementations, the video matching system may treat the live pool segment 608 as a continuation of programming most recently detected on channel 2 601. In these implementations, the video matching system may make this determination based on a low probability that the viewer changed the channel at the exact moment that the live pool feed segment 608 started. In some implementations, the video matching system may further determine that the unexpected live pool feed segment 608 is not likely to be related to any scheduled interactive or targeted content. Hence, in these implementations, the scheduled interactive or targeted content may be suppressed or disabled. The targeted content may not be related to the unscheduled live pool feed segment 608, hence displaying, for example, interactive overlays may not be useful to the viewer.

[0082] At the conclusion of the live pool feed segment 608, channel 2 601 may display segment 4 609. In some cases, segment 4 609 may be scheduled programming, meaning that the live pool feed segment 608 has interrupted, or come at the end of, segment 3 606, and is being played instead of whatever was scheduled to follow segment 3 606. For example, segment 4 609 may be a continuation of the program of segment 3 606, picking up at the point in the program where the program would have been if the live pool segment 608 had not played. As another example, segment 4 609 may be a new program that was scheduled to start after segment 3 606. Segment 4 609 may start at the beginning of the new program, or some part of the beginning may have been overridden by the live pool feed segment.

[0083] In some cases, rather than overriding programming that would have been displayed between segment 3 606 and segment 4 609, the program of segment 3 606 may instead by suspended. Once the live pool feed segment 608 ends, the program of segment 3 606 may resume in segment 4 609, picking up where the program stopped in segment 3. Alternatively, the program of segment 3 606 may restart in segment 4 609. In some implementations, at the end of the live pool feed segment 608, the viewer may be given the option of resuming the program of segment 3 606 or starting the program over from the beginning.

[0084] Another example in FIG. 6 is provided by channel 7 610. On channel 7 610, a segment 1 611 and segment 2 612 are regularly scheduled. Thereafter, channel 7 610 is interrupted by a live pool segment 613. In this example, a media display device may tune into channel 7 610 shortly after the live pool segment 613 starts, or while the live pool segment 613 is airing. During two timer intervals t1 614 and t2 615, the media display device may send samples from the live pool feed segment 613 to the video matching system. The video matching system may subsequently determine that channel 7 610 is playing the live pool feed segment 613. In various implementations, the video matching system may make this determination, for example, based on finding a matching cue for the live pool feed segment 613 that is associated with multiple channels. In this example, the video matching system may not be able to determine which channel the media display device is currently playing. In some implementations, the video matching system may refrain from providing contextually-related content while the live pool feed segment is airing.

[0085] At the conclusion of the live pool feed segment 613, channel 7 610 reverts to scheduled programmed with segment 3 616. During a time interval t3 617, the media display device may send samples from segment 3 616 to the video matching system, and at this point, the video matching system may be able to determine that the media display device is tuned to channel 7 610. In some implementations, the video matching system may associate samples taken during time intervals t1 614 and t2 615 with channel 7 610. The video matching system may further provide contextually-related content related to channel 7 610 to the viewer.

[0086] A further example is illustrated in FIG. 6 by channel 9 618. In this example, channel 9 618 displays regularly scheduled segment 1 619, segment 2 620, segment 3 622, segment 4 624 and segment 5 626. Assuming that a media display device is tuned into channel 9 618, during time intervals t1 621, t2 623, and t3 625, the media display device sends samples from the segments 620, 622, 624 to a video matching system. In some implementations, the video matching system may determine that channel 9 618 did not include a live pool feed segment. The video matching system may further provide contextually-related content, such as for example replacing commercials with commercials of specific interest to the viewer, where the replacement commercials may be based on information previously collected from the media display device.

[0087] In some situations, a channel may start displaying a common pool feed after the common pool feed segment has started. FIG. 7 illustrates another example where multiple media streams, here illustrated as television channels, display the same media content at the same time. In this example, channel 2 701 displays regularly scheduled segment 1 702, segment 2 704, and segment 3 706 before switching to a live pool feed segment 708. Should a media display device be tune into channel 2 701 at about the time the live pool feed segment 708 starts, the media display device may send samples during time intervals t1 703, t2 705, t3 707, and t4 717 during a subsequent segment 4 709 to a video matching system. As discussed above, the video matching system may determine that the media display device is tuned into channel 2 701 upon receiving the samples from segment 4 709.

[0088] Channel 7 710 of this example displays regularly scheduled segment 1 711 and segment 2 712. Though the live pool feed segment 713 starts at point in time 721, channel 7 710 delays switching to the live pool feed segment 713. This delay may be because, for example, segment 2 712 ran overtime, because the programmers of channel 7 710 determined to allow segment 2 712 to finish, and/or because segment 2 712 included an introduction to the live pool feed segment 713. Assuming a media display device tuned into channel 7 710 at around time 721, the media display device may send samples during time intervals t1 714 and t2 715 to a video matching system. The video matching system may determine that channel 7 710 is playing the live pool feed segment 713, but may not be able to determine which channel the media display device is playing.

[0089] In another example scenario, a media display device may initially be tuned into channel 7 1310, and then at time 721 switch to channel 2 701. At time 721, channel 2 701 may already be playing the live pool feed segment 708. Because the live pool feed segment 708 is associated with multiple channels, during the time intervals t2 705 and t3 707, the video matching system may not be able to determine which channel the media display device has changed to. Once segment 4 709 displays, the video matching device can determine that the media display device is tuned into channel 2 701. Upon making this determination, the video matching system may associate samples from t2 705 and t3 707 with channel 2 701.

[0090] FIG. 8 illustrates an example where multiple media content streams are carrying the same, common media segment 810 at approximately the same time. In this example, two media

content streams 804a, 804b are being delivered to two different media display devices 806a, 806b. In other examples, many different media content streams, each carrying the same media content, may be being delivered to many different media display devices at the same time. In this example, the two media display devices 806a, 806b may be, for example, in the same household, being used by two different people 808a, 808b in the same family. For example, one family member 808a may be watching television programs on a TV 806a in the living room, while another family member 808b is watching television programs on a laptop in the study. Alternatively, the two media display devices 806a, 806b and the people using them 808a, 808b may be unrelated and in different locations.

[0091] At around the same time, the two people 808a, 808b maybe tune their display devices 806a, 806b into the same media segment 810. For example, the two people 808a, 808b may individually decide to watch the same movie. As a result, the two media display devices 806a, 806b may be displaying exactly the same media segment 810 at a given moment. Alternatively, at a given moment there may be a time of difference of a few seconds or a few minutes between the content being displayed by each device 806a, 806b. For example, the TV 806a may be a few seconds ahead of the laptop 806b in the movie.

[0092] In this example, the media content streams 804a, 804b are digital audio-video streams and/or audio streams being delivered over the Internet 850. For example, the media content streams may include movies, television shows, music, text, and/or images being provided by a website. The media content streams 804a, 804b may each be provided by different content providers, Provider A 802a and Provider B 802b. Providers A 802a and B 802b may be, for example, Internet movie, music, and/or television providers. Alternatively, in some cases, Provider A 802a and Provider B 802b may be the same content provider.

[0093] In the example of FIG. 8, both of the media display devices 806a, 806b may be connected to a computing device (not illustrated) that is configured with a video matching system. The video matching system may attempt to identify media content being played by each of the devices 806a, 806b, using cues taken from the media content being played, while it is being played. The video matching system may match these cues against a database, and from the database determine information such as the title of the media content being played, identities of the creator, producer, and/or distributor of the media content, an identity of the person who is

viewing the media content, and/or an identity of the viewer's device. For example, in this example, the video matching system may be able to obtain the identity of Provider A 802a by examining the media content stream 804a. As another example, the video matching system may be able to distinguish the TV 806a from a smartphone owned by the same person 808a. Additionally, the video matching system may be able to determine that the person's 808a TV 806a and smartphone are located in different places. The video matching system may further use this information to provide contextually-related content, such as interactive information and/or advertising, to the media display device, which may display the contextually-related content to the viewer.

[0094] As discussed above, when the two example media content streams 804a, 804b are displaying the same media segment 810 at about the same time, the video matching system may not be able to determine some information. For example, while the video matching system may be able to identify the TV 806a and the laptop 806b that are playing the common media segment 810, this information alone may not be enough for the video matching system to determine contextually-related content that is specific to each device. For example, should the video matching system be provided with information such as characteristics or an identity of the individual people 808a, 808b watching the two devices, the video matching system may be able to tailor contextually-related content for the first person 808a while providing different contextually-related content for the second person 808b.

[0095] To determine contextually-related content, the video matching system may use the methods discussed above with respect to FIGS. 6 and 7. In various implementations, the video matching system may determine identification information for other media content included in the media content streams 804a, 804b of FIG. 8. For example, in one case, the person 808a viewing the TV 806a may have been watching sports news prior to tuning into the common media segment 810. The video matching system may use the identification information for this prior media content to identify the media content stream 804a. For example, the video matching system may be able to identify the media content stream 804a as being associated with the TV 806a and/or the person 808a watching the TV 806a. The video matching system may further be able to provide contextually-related content to the TV 806a. For example, the video matching

system may provide news for the person's 808a favorite team and/or advertising for sporting events or sporting equipment.

[0096] As another example, the person 808b watching the laptop 806b may not have been using the laptop 806b to view media content prior to tuning into the common media segment 810. Instead, this person 808b may, either in the middle of the common media segment 810 or after, view other media content. For example, during a commercial break in the common media segment 810, the person 808b may use the laptop 806b to shop for school supplies. The video matching system may use this other media content, displayed in the middle of or after the common media segment 810, to identify the media content stream 804b. For example, the video matching system may identify the media content stream as associated with the laptop 806b and/or the person 808b using the laptop 806b. The video matching system may further provide contextually-related content to the laptop 806b. For example, the video matching system may provide advertising related to school supplies, suggestions for where to shop, and/or suggestions for where to find sales.

[0097] In various implementations, a video matching system may use other methods to identify a media content stream when the media content stream is playing a common media segment. In some cases, the provider of the media content stream may provide a graphic element, superimposed onto the common media segment. FIG. 9 illustrates one example of a graphic 902 that has been superimposed onto a common media segment being played by a media display device 900. In various implementations, the graphic 902 may be used by a content provider to identify the content provider, particularly when multiple content providers are playing a common media segment. For example, multiple local television channels may be showing the same national news segment at the same time. The local broadcaster may add a graphic 902 to the news segment, to identify itself. The graphic 902 may be, for example, a logo, program information, or other information.

[0098] FIG. 9 illustrates one example of the size and location of a graphic that may be used be some content providers. In other examples, the graphic 902 may be in the shape of a banner or ticker across the bottom or top of the screen, a column on the left or right of the screen, or may appear superimposed over the center of the screen.

[0099] In various implementations, a method for detecting a graphic overlay may examine the video display, and find video image edges. Video image edges can be detecting by looking for high contrast differences between sections of a screen of a media display device. The method may further include monitoring whether the detected edges remain stationary. When the detected edges remain in particular locations for longer than a short duration, the video matching systems may determine that it has found an on-screen graphic. For example, the video matching system may look for high-contrast difference across the bottom area of the screen, which may indicate the presence of an on-screen banner.

[0100] In various implementations, the video matching system described above can include methods for detecting a graphic overlay. For example, as discussed above, pixel patches can defined for the screen of a media display device. A "pixel patch" may be defined as a block of pixels that are sampled from the screen of a media display device. A pixel patch may contain some number of pixels, each of which can have, for example, RGB color values (or YUV or color values expressed in some other format). For the purposes of graphics overlay detection, the pixel patches may be, for example, 32 pixels wide by 32 pixels high, or a multiple of 32 pixels, such as 64 pixels wide by 64 pixels high. These example sizes may take advantage of the discrete cosine transform (DCT). A discrete cosine transform function can be performed by the video matching system. Edges of a graphic overlay can be detected by examining the coefficients in the lower right quadrant of the discrete cosine transform for each pixel patch

[0101] In various implementations, the detection process could also include detecting, over a predetermined length of time, whether high-frequency information from the discrete cosine transform is unchanged. When the high-frequency information does not change, a graphic overlay may be present. In these implementations, some onscreen graphics, such as scrolling banners, can be identified.

[0102] In various implementations, other onscreen graphic detection methods can use algorithms such as Sobel and Sharr, or can use an algorithm from the perceptual hashing family of image analysis. As with discrete cosine transform, these algorithms can also be used to detect edges, as well as corners, of graphical elements within video signals. In some cases, a pixel patch with an odd number of pixels, such as 3 pixels by 3 pixels, may be used in a convolution-coded stepwise sweep over a video area of interest, to search for edges.

[0103] In various implementations, detecting an onscreen graphic may begin with reducing the pixel information in a pixel patch from an 8-bit Red-Green-Blue (RGB) value an 8-bit monochrome value. Next, a Gaussian blur may be applied to reduce noise in the video information. Next the pixel matrix (that is, the resulting pixel patch) may be passed over the video area of interest. This matrix can then be used to calculate a first-order differential of the pixel values relative to either the vertical or horizontal axis of the video screen. The computed differential is left behind in the respective pixel locations. This differential can be examined for maximum values, which may indicate edges.

[0104] In various implementations, another method for detecting graphics overlays is to train the video matching system with various graphics that can be used for graphic overlays. An image matching algorithm can then be used to match the trained or learned graphics to the pixels on the screen. For example, the video matching system can use a perceptual hash (pHash) approach to perform the matching. Examples of other frame comparison methods include the Scale-invariant feature transform (SIFT) and Speeded Up Robust Features (SURF). In implementations where pHash is used, entire video frames can be quickly processed. The resulting hash values may be compared to reference video images. These reference video images may also processed using pHash, and may be supplied from a central server. One of the advantages of using pHash is that it may be able to reliably match coarse features (e.g. large rectangles or other shapes that may be used by graphic overlays) with relatively high insensitivity to contrast, brightness, or color changes. Another advantage of pHash is its ability to also match detailed individual video frames.

[0105] In various implementations, the video matching system may further maintain a library of different possible graphics overlay comparison candidates. Furthermore, the video matching system may use the library without increasing the total number of total image searches that are conducted within a unit of time. Specifically, in some implementations, the video matching system may track successful detections. Graphics overlay comparison candidates that match successfully and frequently may be more likely to match in the future, while candidates that match infrequently or that have not successfully matched are less likely to match in the future.

[0106] In various implementations, graphics overlay detection can be interleaved with processes for automated content recognition.

[0107] FIG. 10 illustrates an example of a process 1000 that may be implemented when multiple media content streams include a same unscheduled media segment. The process 1000 may be implemented by a computing device, where the computing device has been configured with a video matching system such as is described above.

[0108] At step 1002, the computing device may receive a plurality of media content streams. The computing device may be configured to identify media content being played by a particular media display device (e.g., a television, a tablet computer, a laptop, etc.) at a particular time. At least two of the plurality of media content streams may concurrently include a same unscheduled segment. For example, two media content streams may both include a "breaking news" segment, that is, a national broadcast of a significant event. As another example, to media content streams may both include the same streaming movie, where the movie was requested by different people using different media display devices. In this example, the media segment is "unscheduled" because there may not be a schedule of programming associated with the media display devices.

[0109] At step 1004, the computing device may determine that the particular media display device is playing the unscheduled media segment in a media content stream at a current time. The computing device may make this determining by examining the media content available at the current time in each of the plurality of media content streams. For example, the plurality of media content streams may include two or more local televisions channels, and two or more of these local television channels may both be receiving a breaking news feed.

[0110] At step 1006, the computing device may determine identification information from the media content included in the media content stream being played by the particular media display device at the current time. For example, the computing device may use identification information provided by media content that was played by the particular media display device before the unscheduled media segment. Alternatively or additionally, the computing device may use identification information provided by media content that is played after the unscheduled media segment. The identification information may identify the media content stream. For example, the identification information may identify a channel, a service provider, the particular media display device, and/or the person using the particular media display device.

[0111] At step 1008, the computing device may determine contextually-related content. Contextually-related content may include, for example, interactive information, advertisements,

and/or suggestions for additional content, among other things. The contextually-related content may be disabled while the unscheduled media segment is being played by the particular media display device.

[0112] At step 1010, the computing device may display the media content stream and the contextually-related content after the unscheduled media segment has been played. For example, the computing device may overlay the contextually-related information over media content that follows the unscheduled media segment. Alternatively or additionally, the computing device may insert the contextually-related information after the unscheduled media segment and before additional media content is played.

[0113] Various methods related to matching cues from unknown media content to candidates in a reference database will now be discussed in greater detail. These methods include the nearest neighbor search process discussed above with respect to FIG. 2.

[0114] As discussed above, a video matching system can be configured to identify a media content stream when the media content stream includes an unscheduled media segment. As further discussed above, identifying the media content stream may include identifying media content played by a media display device before or after the unscheduled media segment. Processes for identifying media content are discussed above with respect to FIG. 1. Specifically, the video content system may use samples taken from the display mechanism of the media content device (e.g., graphic and/or audio samples) and generate cues from these samples. The video matching system may then match the cues against a reference database, where the database contains cues for known media content.

[0115] The video matching system may further include various methods to improve the efficiency of finding matches in the database. The database may contain an enormous number of cues, and thus the video matching system may include algorithms for finding potential matches, or "candidates" to match against. The video matching system may further include algorithms to determine which candidate cues actually match cues generated from the media content device's display mechanism. Locating candidate cues may be more efficient than other methods for matching cue values against the values in the database, such as matching a cue against every entry in the database.

[0116] Nearest neighbor and path pursuit are examples of techniques that can be used to match unknown cues to candidate cues in the reference database. Path pursuit is a mathematical method for identifying a related sequence of points from among many possible points. Nearest neighbor is a method that can be used to identify candidate points for conducting a path pursuit. Below, an example applying path nearest neighbor and path pursuit to tracking video transmission using ambiguous cues is given, but the general concept can be applied to any field where match candidates are to be selected from a reference database.

[0117] A method for efficient video pursuit is presented. Video pursuit is the application path pursuit techniques to the problem of locating matching candidates in a video reference database given unknown video cues. Given a large number of video segments, the system must be able to identify in real time what segment a given query video input is taken from and in what time offset. The segment and offset together are referred to as the location. The method is called video pursuit since it must be able to efficiently detect and adapt to pausing, fast forwarding, rewinding, abrupt switching to other segments, and switching to unknown segments. Before being able to pursue live video the database is processed. Visual cues (a handful of pixel values) are taken from frames every constant fraction of a second and put in specialized data structure (note that this can also be done in real time). The video pursuit is performed by continuously receiving cues from the input video and updating a set of beliefs or estimates about its current location. Each cue either agrees or disagrees with the estimates, and they are adjusted to reflect the new evidence. A video location is assumed to be the correct one if the confidence in this being true is high enough. By tracking only a small set of possible "suspect" locations, this can be done efficiently.

[0118] A method is described for video pursuit but uses mathematical constructs to explain and investigate it. It is the aim of this introduction to give the reader the necessary tools to translate between the two domains. A video signal is comprised of sequential frames. Each can be thought of as a still image. Every frame is a raster of pixels. Each pixel is made out of three intensity values corresponding to the red, green, and blue (RGB) make of that pixel's color. In the terminology used herein, a cue is a list of RGB values of a subset of the pixels in a frame and a corresponding time stamp. The number of pixels in a cue is significantly smaller than in a

frame, usually between 5 and 15. Being an ordered list of scalar values, the cue values are in fact a vector. This vector is also referred to as a point.

[0119] Although these points are in high dimension, usually between 15 and 150, they can be imagined as points in two dimensions. In fact, the illustrations will be given as two dimensional plots. Now, consider the progression of a video and its corresponding cue points. Usually a small change in time produces a small change in pixel values. The pixel point can be viewed as "moving" a little between frames. Following these tiny movements from frame to frame, the cue follows a path in space like a bead would on a bent wire.

[0120] In the language of this analogy, in video pursuit the locations of the bead in space (the cue points) are received and the part of wire (path) the bead is following is looked for. This is made significantly harder by two facts. First, the bead does not follow the wire exactly but rather keeps some varying unknown distance from it. Second, the wires are all tangled together. These statements are made exact in section 2. The algorithm described below does this in two conceptual steps. When a cue is received, the algorithm looks for all points on all the known paths that are sufficiently close to the cue point; these are called suspects. This is done efficiently using the Probabilistic Point Location in Equal Balls algorithm. These suspects are added to a history data structure and the probability of each of them indicating the true location is calculated. This step also includes removing suspect locations that are sufficiently unlikely. This history update process ensures that on the one hand only a small history is kept but on the other hand no probable locations are ever deleted. The generic algorithm is given in Algorithm 1 and illustrated in FIG. 11.

Algorithm 1 Generic path pursuit algorithm.

```
1:  Set of suspects is empty
2:  loop
3:      Receive latest cue.
4:      Find path points who are close to it.
5:      Add them to the set of suspects.
6:      Based on the suspects update the location likelihood function.
7:      Remove from suspect set those who do not contribute to the
        likelihood function.
8:      if A location is significantly likely then
9:          Output the likely location.
10:     end if
11: end loop
```

[0121] The following sections begin with describing the Probabilistic Point Location in Equal Balls (PPLEB) algorithm in Section 1. the PPLEB algorithm is used in order to perform line 5 in Algorithm 1 above efficiently. The ability to perform this search for suspects quickly is crucial for the applicability of this method. In section 2 one possible statistical model is described for performing lines 6 and 7. The described model is a natural choice for the setup. It is also shown how it can be used very efficiently.

[0122] Section 1 - Probabilistic Point Location in Equal Balls

[0123] The following section describes a simple algorithm for performing probabilistic point location in equal balls (PPLEB). In the traditional PLEB (point location in equal balls), one starts with a set of n points x, in lR *d* and a specified ball of radius *r*. The algorithm is given O(poly(n)) preprocessing time to produce an efficient data structure. Then, given a query point x the algorithm is required to return all points x, such that $||x-x_i|| \leq r$. The set of points such that $||x-x_i|| \leq r$ geometrically lie within a ball of radius r surrounding the query x (see FIG. 12). This relation is referred to as x, being close to x or as x, and x being neighbors.

[0124] The problem of PPLEB and the problem of nearest neighbor search are two similar problems that received much attention in the academic community. In fact, these problems were among the first studied in the field of computational geometry. Many different methods cater to the case where the ambient dimension is small or constant. These partition the space in different ways and recursively search through the parts. These methods include KD-trees, cover-trees, and others. Although very efficient in low dimension, when the ambient dimension is high, they tend

to perform very poorly. This is known as the "curse of dimensionality". Various approaches attempt to solve this problem while overcoming the curse of dimensionality. The algorithm used herein uses a simpler and faster version of the algorithm and can rely on Local Sensitive Hashing.

[0125] Section 1.1 Locality Sensitive Hashing

[0126] In the scheme of local sensitive hashing, one devises a family of hash functions H such that:

$$\Pr_{u\sim U}(u(x) \neq u(y) \mid \|x-y\| \leq r) \equiv p$$

$$\Pr_{u\sim U}(u(x) \neq u(y) \mid \|x-y\| \geq 2r) \geq 2p$$

[0127] In words, the probability of x and y being mapped to the same value by h is significantly higher if they are close to each other.

[0128] For the sake of clarity, let us first deal with a simplified scenario where all incoming vectors are of the same length r' and $r' > \sqrt{2r}$. The reason for the latter condition will become clear later. First a random function $u \in U$ is defined, which separates between x and y according to the angle between them. Let $\vec{u}$ be a random vector chosen uniformly from the unit sphere S^{d-1} and let $u(x)=\text{sign}(\vec{u} \cdot x)$ (See FIG. 13). It is easy to verify that $\Pr_{u-U}(u(x)) \neq u(y)) = 0_{x,y}/\pi$. Moreover, for any points x, y, x', y' on a circle such that

$$\Pr_{u\sim U}(u(x) \neq u(y) \mid \|x-y\| \leq r) \equiv p$$

$$\Pr_{u\sim U}(u(x) \neq u(y) \mid \|x-y\| \geq 2r) \geq 2p$$

[0129] The family of functions H is set to be a cross product of t independent copies of u, i.e. $h(x)=[u1(x), \ldots, u_t(x)]$. Intuitively, one would like to have that if h(x)=h(y) then x and y are likely to be close to each other. Let us quantify that. First, compute the expected number of false positive mistakes n_{fp}. These are the cases for which h(x)=h(y) but ||x-y|| > 2r. A value t is found for which n_{fp} is no more than 1, i.e. one is not expected to be wrong.

$$E[n_{fi}] \leq n(1-2p)^t \leq 1$$

$$\rightarrow t \geq \log(1/n)/\log(1-2p)$$

[0130] Now, the probability that h(x)=h(y) given that they are neighbors is computed:

$$Pr(h(x) = h(y) \mid \|x - y\| \leq r) \geq (1-p)^{\log(1/n)/\log(1-2p)}$$

$$= (1/n)^{\log(1-p)/\log(1-2p)}$$

$$\geq 1/\sqrt{n}.$$

[0131] Note here that one must have that 2p<1 which requires r' > $\sqrt{2r}$. This might not sound like a very high success probability. Indeed, $1/\sqrt{n}$ is significantly smaller than ½. The next section will describe how to boost this probability up to ½.

[0132] Section 1.2 The Point Search Algorithm

[0133] function h maps every point in space to a bucket. Define the bucket function $B_h: \mathbb{R}^d \rightarrow 2^{[n]}$ of a point x with respect to hash function h as $B_h(x) \equiv \{x_i | h(x_i) = h(x)\}$. The data structure maintained is m=O($\sqrt{n}$) instances of bucket functions [Bh_1, ... , Bh_m]. When one searches for a point x, the function returns $B(x) = \cup i B_{h_j}(x)$. According to the previous section, there are two desired results:

$$Pr(x_i \epsilon B(x) | \|x_i - x\| \leq r) \geq 1/2$$

$$E[|B(x) \cap \{x_i | \|x - x_i\| > 2r\}|] \leq \sqrt{n}.$$

[0134] In other words, while with probability at least ½ each neighbor of x is found, one is not likely to find many non-neighbors.

[0135] Section 1.3 Dealing with Different Radii Input Vectors

[0136] The previous sections only dealt with searching through vectors of the same length, namely r'. Now described is how one can use the construction as a building block to support a search in different radii. As seen in FIG. 14, the space is divided into rings of exponentially growing width. Ring i, denoted by R_i, includes all points x_i such that $\|x_i\| \in \left[2r(1+\epsilon)^i, 2r(1+\epsilon)^{i+1}\right]$. Doing this achieves two ends. First, if x_i and x_j belong to the same ring, then $\|x_j\|/(1+\epsilon) \leq \|x_i\| \leq \|x_j\|(1+\epsilon)$. Second, any search can be performed in at most $1/\epsilon$ such rings. Moreover, if the maximal length vector in the data set is r' then the total number of rings in the system is O(log(r'/r)).

[0137] Section 2 The Path Pursuit Problem

[0138] In the path pursuit problem, a fixed path in space is given along with the positions of a particle in a sequence of time points. The terms particle, cue, and point will be used interchangeably. The algorithm is required to output the position of the particle on the path. This is made harder by a few factors: The particle only follows the path approximately; the path can be discontinuous and intersect itself many times; both particle and path positions are given in a sequence of time points (different for each).

[0139] It is important to note that this problem can simulate tracking a particle on any number of paths. This is simply done by concatenating the paths into one long path and interpreting the resulting position as the position on the individual paths.

[0140] More precisely, let path P be parametric curve $P: \mathbb{R} \rightarrow \mathbb{R}^d$. The curve parameter will be referred to as the time. The points on the path that are known to us are given in arbitrary time points t_i, i.e. n pairs $(t_i, P(t_i))$ are given. The particle follows the path but its positions are given in different time points, as shown in FIG. 15. Further, m pairs $(t'_j, x(t'_j))$ are given, where $x(t'_j)$ is the position of the particle in time t'_j.

[0141] Section 2.1 Likelihood Estimation

[0142] Since the particle does not follow the path exactly and since the path can intersect itself many times it is usually impossible to positively identify the position on the path the particle is actually on. Therefore, a probability distribution is computed on all possible path locations. If a

location probability is significantly probable, the particle position is assumed to be known. The following section describes how this can be done efficiently.

[0143] If the particle is following the path, then the time difference between the particle time stamp and the offset of the corresponding points on P should be relatively fixed. In other words, if x(t') is currently in offset t on the path then it should be close to P(t). Also, τ seconds ago it should have been in offset t- τ. Thus x(t'- τ) should be close to P(t- τ) (note that if the particle is intersecting the path, and x(t') is close to P(t) temporarily, it is unlikely that x(t'- τ) and P(t- τ) will also be close). Define the relative offset as Δ=t-t'. Notice that as long as the particle is following the path the relative offset Δ remains unchanged. Namely, x(t') is close to P(t'+Δ).

[0144] The maximum likelihood relative offset is obtained by calculating:

$$\Delta = \underset{\delta}{\operatorname{argmax}} \; Pr(x(t'_m), x(t'_{m-1}), \ldots, x(t'_1) \mid P, \delta)$$

[0145] In words, the most likely relative offset is the one for which the history of the particle is most likely. This equation however cannot be solved without a statistical model. This model must quantify: how tightly x follows the path; how likely it is that x jumps between locations; and how smooth the path and particle curves are between the measured points.

[0146] Section 2.2 Time Discounted Binning

[0147] Now described is a statistical model for estimating the likelihood function. The model makes the assumption that the particle's deviation away from the path distributes normally with standard deviation ar. It also assumes that at any given point in time, there is some non-zero probability the particle will abruptly switch to another path. This is manifested by an exponential discount with time for past points. Apart for being a reasonable choice for a modeling point of view this model also has the advantage of being efficiently updateable. For some constant time unit 1:, set the likelihood function to be proportional to f which is defined as follows:

$$f_m(\lfloor \delta / \tau \rfloor) = \sum_{j=1}^{m} \sum_{i=1}^{n} e^{-\left(\frac{\left|x(t'_j) - P(t_i + \delta)\right|}{\sigma r}\right)^2} (1-\zeta)^{t'_j - t_i}.$$

[0148] Here $\alpha<<1$ is a scale coefficient and is the probability that the particle will jump to a random location on the path in a given time unit.

[0149] Updating the function f efficiently can be achieved using the following simple observation.

$$f_m(\lfloor \delta/\tau \rfloor) = \sum_{i=1}^{n} e^{-\left(\frac{\|x(t'_m)-P(t_i+\delta)\|}{\sigma r}\right)^2} + f_{m-1}(\lfloor \delta/\tau \rfloor)(1-\zeta)^{t'_m - t'_{m-1}}$$

[0150] Moreover, since $\alpha <<1$, if $\|x(t'_m) - P(t_i)\| \geq r$, the follow occurs:

$$e^{-\left(\frac{\|x(t'_m)-P(t_i)\|}{\sigma r}\right)^2} \approx 0.$$

[0151] This is an important property of the likelihood function since the sum update can now performed only over the neighbors of $x(t'_j)$ and not the entire path. Denote by S the set of (t_i, $P(t_i)$) such that $\|x(t'_m) - P(t_i)\| \leq r$. The follow equation occurs:

$$f_m(\lfloor \delta/\tau \rfloor) = \sum_{(t_i,P(t_i))\in S \wedge \lfloor (t'_m - t_i)/\tau \rfloor = \lfloor \delta/\tau \rfloor} e^{-\left(\frac{\|x(t'_m)-P(t_i)\|}{\sigma r}\right)^2} + f_{m-1}(\delta)(1-\zeta)^{t'_m - t'_{m-1}}$$

[0152] This is described in Algorithm 2.2 below. The term f is used as a sparse vector that receives also negative integer indices. The set S is the set of all neighbors of $x(t_i)$ on the path and can be computed quickly using the PPLEB algorithm. It is easy to verify that if the number of neighbors of $x(t_i)$ is bounded by some constant n_{near} then the number of non-zeros in the vector f is bounded by n_{near}/ζ which is only a constant factor larger. The final stage of the algorithm is to output a specific value of δ if $f(\lfloor \delta/\tau \rfloor)$ is above some threshold value.

Algorithm 2 Efficient likelihood update.

1: $f \leftarrow 0$
2: while $(t_j', x(t_j')) \in$ INPUT do
3: $f \leftarrow (1-\xi)^{t_j'-t_i'} f$
4: $S \leftarrow \{(t_i, P(t_i)) \mid \|x(t_j') - P(t_i)\| \leq r\}$
5: for $(t_i, P(t_i)) \in S$ do
6: $\delta \leftarrow t_j' - t_i$
7: $f(\lfloor \delta/\tau \rfloor) \leftarrow f(\lfloor \delta/\tau \rfloor) + e^{-\left(\frac{\|x(t_j')-P(t_i')\|}{\sigma_r}\right)^2}$
8: end for
9: Set all f values below threshold ϵ to zero.
10: end while

[0153] FIG. 11 gives three consecutive point locations and the path points around them. Note that neither the bottom point nor middle one alone would have been sufficient to identify the correct part of the path. Together, however, they are. Adding the top point increases the certainty that the particle is indeed of the final (left) curve of the path.

[0154] In FIG. 12, given a set of n (grey) points, the algorithm is given a query point (black) and returns the set of points that lie within distance r from it (the points inside the circle). In the traditional setting, the algorithm must return all such points. In the probabilistic setting each such point should be returned only with some constant probability.

[0155] FIG. 13 illustrates the values of $u(x_1)$, $u(x_2)$, and $u(x)$. Intuitively, the function u gives different values to x_1 and x_2 if the dashed line passes between them and the same value otherwise. Passing the dashed line in a random direction ensures that the probability of this happening is directly proportional to angle between x_1 and x_2.

[0156] FIG. 15 shows that by dividing the space into rings such that ring R_i is between radius $2r(1+\epsilon)^i$ and $2r(1+\epsilon)^{i+1}$, it can be made sure that any two vectors within a ring are the same length up to $(1+\epsilon)$ factors and that any search is performed in at most $1/\epsilon$ rings.

[0157] FIG. 15 shows a self-intersecting paths and a query point (in black). It illustrates that without the history of the particle positions it is impossible to know where it is on the path.

[0158] FIG. 16 gives three consecutive point locations and the path points around them. Note that neither $x(t_1)$ nor $x(t_2)$ alone would have been sufficient to identify the correct part of the path. Together however they are. Adding $x(t_3)$ increases the certainty that the particle is indeed of the final (left) curve of the path.

[0159] In the preceding description, for the purposes of explanation, specific details are set forth in order to provide a thorough understanding of various examples. However, it will be apparent that various examples may be practiced without these specific details. For example, circuits, systems, networks, processes, and other components may be shown as components in block diagram form in order not to obscure the examples in unnecessary detail. In other instances, well-known circuits, processes, algorithms, structures, and techniques may be shown without unnecessary detail in order to avoid obscuring the examples. The figures and description are not intended to be restrictive.

[0160] The preceding description provides exemplary illustrations only, and is not intended to limit the scope, applicability, or configuration of the disclosure. Rather, the preceding description of the examples will provide those skilled in the art with an enabling description for implementing the various examples. It should be understood that various changes may be made in the function and arrangement of elements without departing from the spirit and scope of the invention as set forth in the appended claims.

[0161] Also, it is noted that individual examples may be described as a process which is depicted as a flowchart, a flow diagram, a data flow diagram, a structure diagram, or a block diagram. Although a flowchart may describe the operations as a sequential process, many of the operations can be performed in parallel or concurrently. In addition, the order of the operations may be re-arranged. A process is terminated when its operations are completed, but could have additional steps not included in a figure. A process may correspond to a method, a function, a procedure, a subroutine, a subprogram, etc. When a process corresponds to a function, its termination can correspond to a return of the function to the calling function or the main function.

[0162] The term “machine-readable storage medium” or “computer-readable storage medium” includes, but is not limited to, portable or non-portable storage devices, optical storage devices, and various other mediums capable of storing, containing, or carrying instruction(s) and/or data.

A machine-readable storage medium or computer-readable storage medium may include a non-transitory medium in which data can be stored and that does not include carrier waves and/or transitory electronic signals propagating wirelessly or over wired connections. Examples of a non-transitory medium may include, but are not limited to, a magnetic disk or tape, optical storage media such as compact disk (CD) or digital versatile disk (DVD), flash memory, memory or memory devices. A computer-program product may include code and/or machine-executable instructions that may represent a procedure, a function, a subprogram, a program, a routine, a subroutine, a module, a software package, a class, or any combination of instructions, data structures, or program statements. A code segment may be coupled to another code segment or a hardware circuit by passing and/or receiving information, data, arguments, parameters, or memory contents. Information, arguments, parameters, data, or other information may be passed, forwarded, or transmitted using any suitable means including memory sharing, message passing, token passing, network transmission, or other transmission technique.

[0163] Furthermore, examples may be implemented by hardware, software, firmware, middleware, microcode, hardware description languages, or any combination thereof. When implemented in software, firmware, middleware or microcode, the program code or code segments to perform the necessary tasks (e.g., a computer-program product) may be stored in a machine-readable medium. A processor(s) may perform the necessary tasks.

[0164] Systems depicted in some of the figures may be provided in various configurations. In some examples, the systems may be configured as a distributed system where one or more components of the system are distributed across one or more networks in a cloud computing system.

[0165] As described in further detail above, certain aspects and features of the present disclosure relate to identifying unknown data points by comparing the unknown data points to one or more reference data points. The systems and methods described herein improve the efficiency of storing and searching large datasets that are used to identify the unknown data points. For example, the systems and methods allow identification of the unknown data points while reducing the density of the large dataset required to perform the identification. The techniques can be applied to any system that harvests and manipulates large volumes of data. Illustrative examples of these systems include automated content-based searching systems (e.g.,

automated content recognition for video-related applications or other suitable application), MapReduce systems, Bigtable systems, pattern recognition systems, facial recognition systems, classification systems, computer vision systems, data compression systems, cluster analysis, or any other suitable system. One of ordinary skill in the art will appreciate that the techniques described herein can be applied to any other system that stores data that is compared to unknown data. In the context of automated content recognition (ACR), for example, the systems and methods reduce the amount of data that must be stored in order for a matching system to search and find relationships between unknown and known data groups.

[0166] By way of example only and without limitation, some examples described herein use an automated audio and/or video content recognition system for illustrative purposes. However, one of ordinary skill in the art will appreciate that the other systems can use the same techniques.

[0167] Substantial variations may be made in accordance with specific requirements. For example, customized hardware might also be used, and/or particular elements might be implemented in hardware, software (including portable software, such as applets, etc.), or both. Further, connection to other access or computing devices such as network input/output devices may be employed.

[0168] In the foregoing specification, aspects of the various implementations are described with reference to specific examples thereof, but those skilled in the art will recognize that the implementations is not limited thereto. Various features and aspects of the above-described implementations may be used individually or jointly. Further, examples can be utilized in any number of environments and applications beyond those described herein without departing from the broader spirit and scope of the specification. The specification and drawings are, accordingly, to be regarded as illustrative rather than restrictive.

[0169] In the foregoing description, for the purposes of illustration, methods were described in a particular order. It should be appreciated that in alternate examples, the methods may be performed in a different order than that described. It should also be appreciated that the methods described above may be performed by hardware components or may be embodied in sequences of machine-executable instructions, which may be used to cause a machine, such as a general-purpose or special-purpose processor or logic circuits programmed with the instructions to perform the methods. These machine-executable instructions may be stored on one or more

machine readable mediums, such as CD-ROMs or other type of optical disks, floppy diskettes, ROMs, RAMs, EPROMs, EEPROMs, magnetic or optical cards, flash memory, or other types of machine-readable mediums suitable for storing electronic instructions. Alternatively, the methods may be performed by a combination of hardware and software.

[0170] Where components are described as being configured to perform certain operations, such configuration can be accomplished, for example, by designing electronic circuits or other hardware to perform the operation, by programming programmable electronic circuits (e.g., microprocessors, or other suitable electronic circuits) to perform the operation, or any combination thereof.

[0171] While illustrative examples of the application have been described in detail herein, it is to be understood that the inventive concepts may be otherwise variously embodied and employed, and that the appended claims are intended to be construed to include such variations, except as limited by the prior art.

WHAT IS CLAIMED IS:

1. A method, comprising:

receiving, at a computing device, a plurality of media content streams, wherein the computing device is configured to identify media content being played by a media display device at a particular time, and wherein at least two of the plurality of media content streams concurrently include a same unscheduled media segment;

determining that the media display device is playing the unscheduled media segment in a media content stream at a current time, wherein determining includes examining the media content available at the current time in each of the plurality of media content streams;

determining identification information from the media content included in the media content stream being played by the media display device at the current time, wherein the identification information identifies the media content stream;

determining contextually-related content, wherein contextually-related content is disabled while the unscheduled media segment is being played by the media display device; and

displaying the media content stream and the contextually-related content after the unscheduled media segment has been played.

2. The method of claim 1, wherein the contextually-related content is selected using the identification information, and wherein the contextually-related content is provided to the media display device.

3. The method of claim 1, wherein identifying the media content stream includes detecting a graphic superimposed onto the unscheduled media segment while the unscheduled media segment is being played by the media display device, wherein the graphic provides additional identification information for the media content stream.

4. The method of claim 1, wherein the media content used to determine the identification information includes media content included in the media content stream before or after the unscheduled media segment.

5. The method of claim 1, further comprising:

determining that the media display device has been playing the unscheduled media segment since a beginning of the unscheduled media segment; and

identifying the media content stream using identification information determined for media content included in the media content stream prior to the unscheduled media segment.

6. The method of claim 1, further comprising:

determining that the media display device has been playing the unscheduled media segment since a point after a beginning of the unscheduled media segment;

identifying the media content stream using identification information for media content included in the media content stream after the unscheduled media segment.

7. A computing device, comprising:

one or more processors; and

a non-transitory computer-readable medium including instructions that, when executed by the one or more processors, cause the one or more processors to perform operations including:

receiving a plurality of media content streams, wherein the computing device is configured to identify media content being played by a media display device at a particular time, and wherein at least two of the plurality of media content streams concurrently include a same unscheduled media segment;

determining that the media display device is playing the unscheduled media segment in a media content stream at a current time, wherein determining includes examining, the media content available at the current time in each of the plurality of media content streams;

determining identification information from the media content included in the media content stream being played by the media display device at the current time, wherein the identification information identifies the media content stream;

determining contextually-related content, wherein contextually-related content is disabled while the unscheduled media segment is being played by the media display device; and

displaying the media content stream and the contextually-related content after the unscheduled media segment has been played.

8. The computing device of claim 7, wherein the contextually-related content is selected using the identification information, and wherein the contextually-related content is provided to the media display device.

9. The computing device of claim 7, wherein the instructions for identifying the media content stream include instructions that, when executed by the one or more processors, cause the one or more processors to perform operations including:

detecting a graphic superimposed onto the unscheduled media segment while the unscheduled media segment is being played by the media display device, wherein the graphic provides additional identification information for the media content stream.

10. The computing device of claim 7, wherein the media content used to determine the identification information includes media content included in the media content stream before or after the unscheduled media segment.

11. The computing device of claim 7, further comprising instructions that, when executed by the one or more processors, cause the one or more processors to perform operations including:

determining that the media display device has been playing the unscheduled media segment since a beginning of the unscheduled media segment; and

identifying the media content stream using identification information determined for media content included in the media content stream prior to the unscheduled media segment.

12. The computing device of claim 7, further comprising instructions that, when executed by the one or more processors, cause the one or more processors to perform operations including:

determining that the media display device has been playing the unscheduled media segment since a point after a beginning of the unscheduled media segment;

identifying the media content stream using identification information for media content included in the media content stream after the unscheduled media segment.

13. The computing device of claim 7, wherein the instructions for identifying media content being played by the media display device at a particular time include instructions that, when executed by the one or more processors, cause the one or more processors to perform operations including:

obtaining samples from the media content;

generating a cue from the samples; and

matching the cue against a database of cues, wherein the database includes identification information.

14. The computing device of claim 7, wherein the instructions for identifying media content being played by the media display device at a particular time include instructions that, when executed by the one or more processors, cause the one or more processors to perform operations including:

using data from the media content, wherein the computing device is capable of obtaining the data from video data or audio data included in the media content.

15. A computer-program product tangibly embodied in a non-transitory machine-readable storage medium, including instructions that, when executed by one or more processors, cause the one or more processors to:

receive a plurality of media content streams, wherein the computing device is configured to identify media content being played by a media display device at a particular time, and wherein at least two of the plurality of media content streams concurrently include a same unscheduled media segment;

determine that the media display device is playing the unscheduled media segment in a media content stream at a current time, wherein determining includes examining, the media content available at the current time in each of the plurality of media content streams;

determine identification information from the media content included in the media content stream being played by the media display device at the current time, wherein the identification information identifies the media content stream;

determine contextually-related content, wherein contextually-related content is disabled while the unscheduled media segment is being played by the media display device; and

display the media content stream and the contextually-related content after the unscheduled media segment has been played.

16. The computer-program product of claim 15, wherein the contextually-related content is selected using the identification information, and wherein the contextually-related content is provided to the media display device.

17. The computer-program product of claim 15, wherein the instructions for identifying the media content stream include instructions that, when executed by the one or more processors, cause the one or more processors to:

detect a graphic superimposed onto the unscheduled media segment while the unscheduled media segment is being played by the media display device, wherein the graphic provides additional identification information for the media content stream.

18. The computer-program product of claim 15, wherein the media content used to determine the identification information includes media content included in the media content stream before or after the unscheduled media segment.

19. The computer-program product of claim 15, further comprising instructions that, when executed by the one or more processors, cause the one or more processors to:

determine that the media display device has been playing the unscheduled media segment since a beginning of the unscheduled media segment; and

identify the media content stream using identification information determined for media content included in the media content stream prior to the unscheduled media segment.

20. The computer-program product of claim 15, further comprising instructions that, when executed by the one or more processors, cause the one or more processors to:

determine that the media display device has been playing the unscheduled media segment since a point after a beginning of the unscheduled media segment;

identify the media content stream using identification information for media content included in the media content stream after the unscheduled media segment.

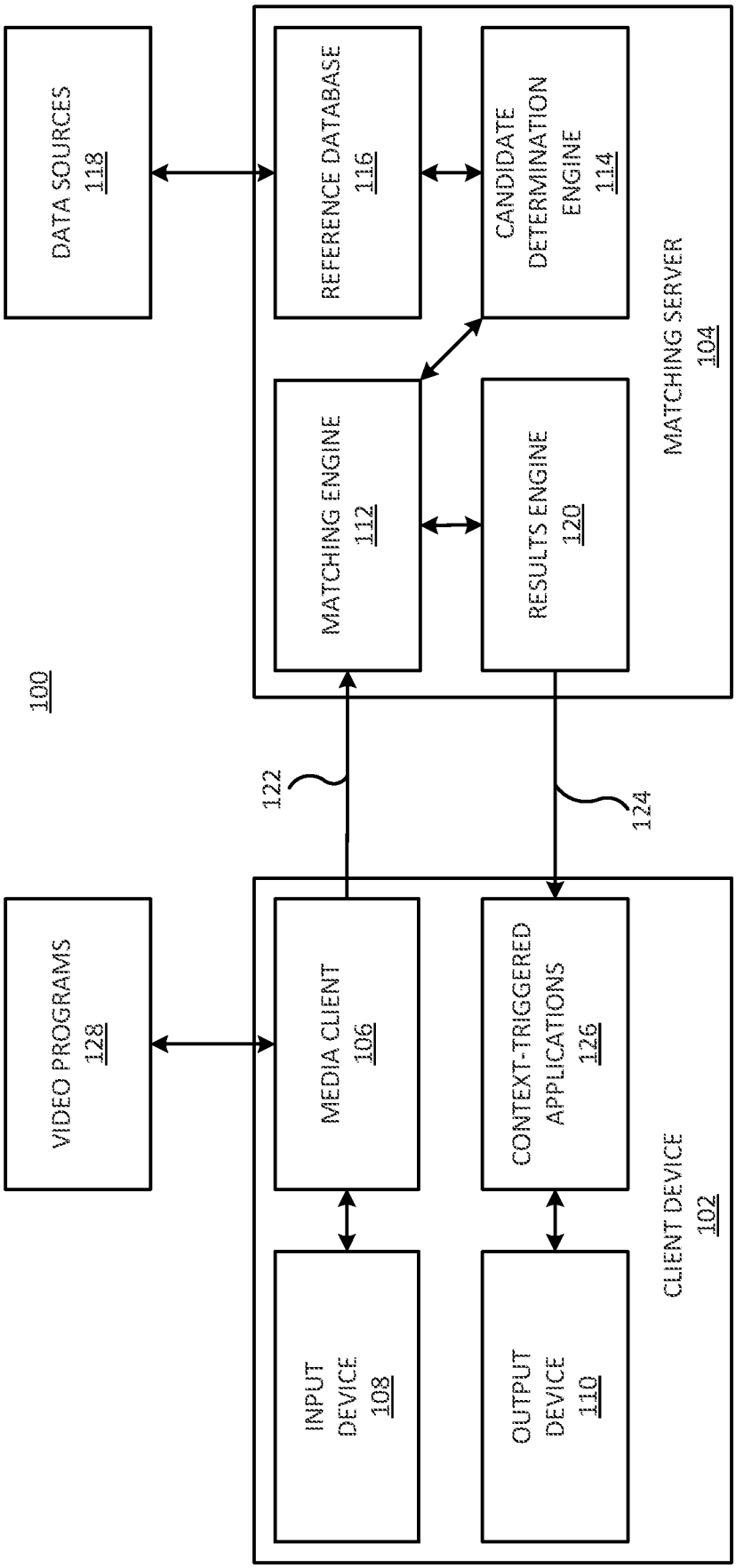

100
DATA SOURCES
118
REFERENCE DATABASE
116
CANDIDATE DETERMINATION ENGINE
114
MATCHING SERVER
104
MATCHING ENGINE
112
RESULTS ENGINE
120
122
124
VIDEO PROGRAMS
128
MEDIA CLIENT
106
CONTEXT-TRIGGERED APPLICATIONS
126
CLIENT DEVICE
102
INPUT DEVICE
108
OUTPUT DEVICE
110

FIG. 1

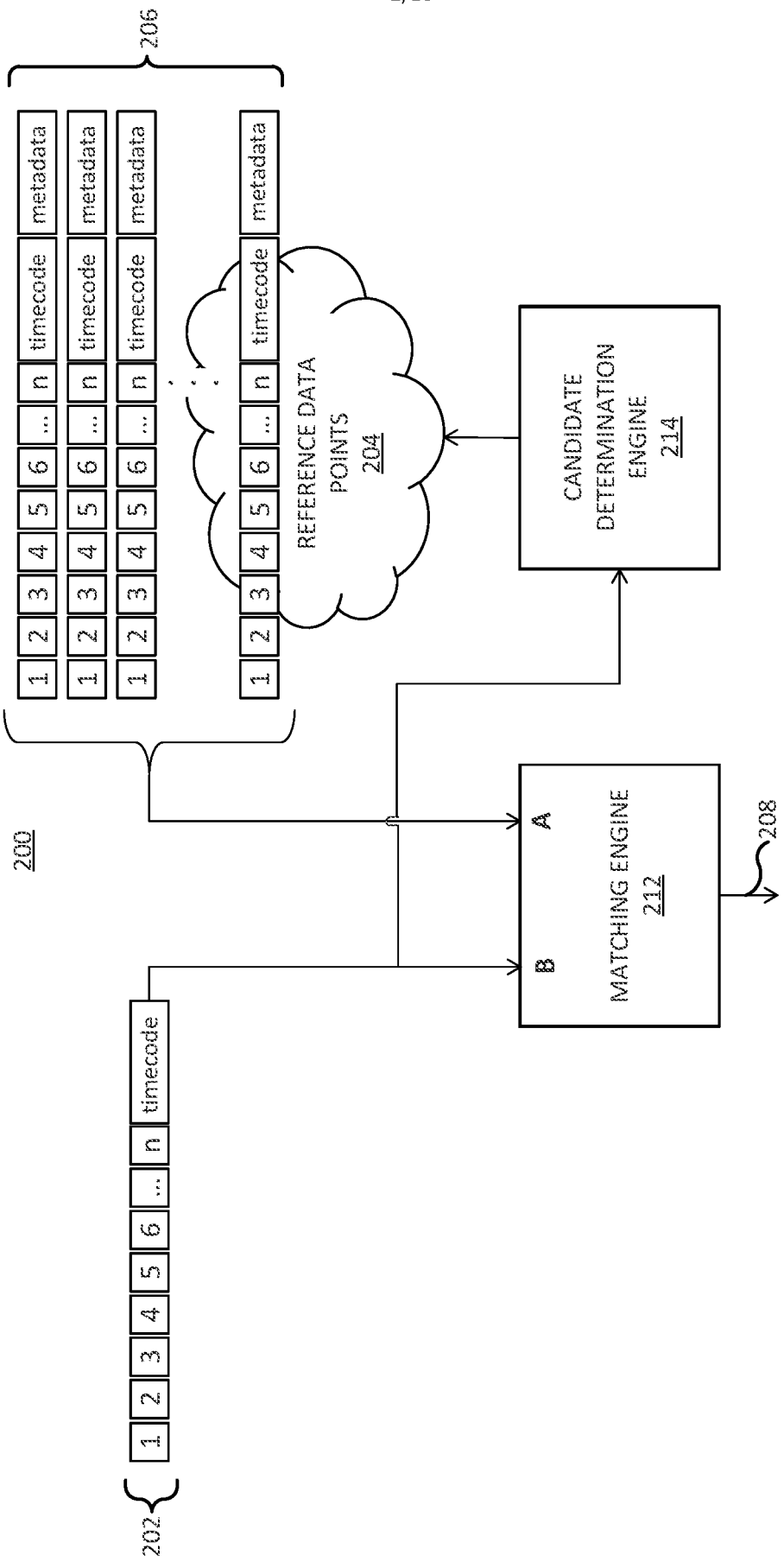

206
1 2 3 4 5 6 ... n timecode metadata
REFERENCE DATA POINTS 204
CANDIDATE DETERMINATION ENGINE 214
200
A
B
MATCHING ENGINE 212
208
1 2 3 4 5 6 ... n timecode
202

FIG. 2

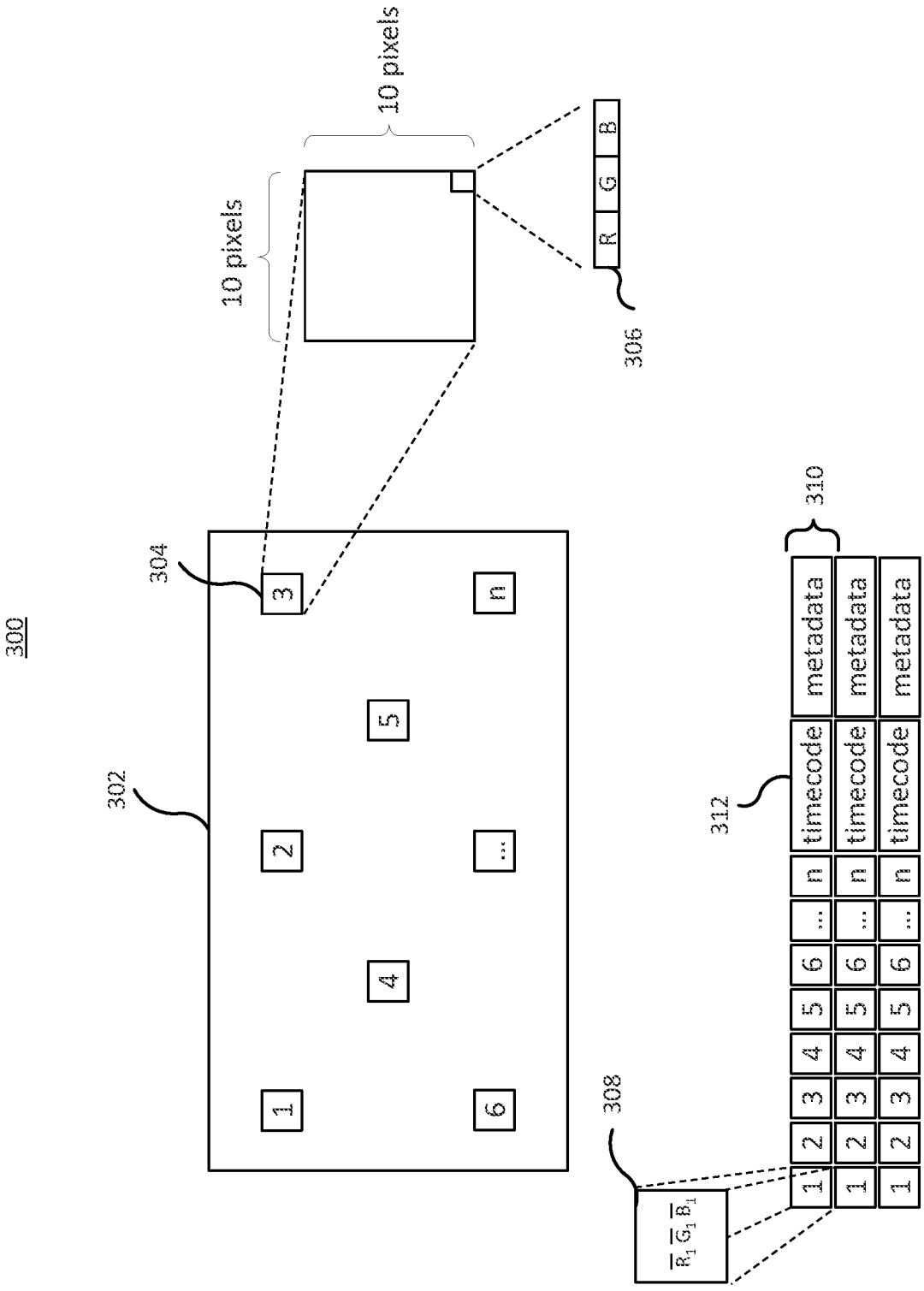

300
302
304
306
308
310
312
10 pixels
10 pixels
R
G
B
1
2
3
4
5
6
...
n
timecode
metadata

FIG. 3

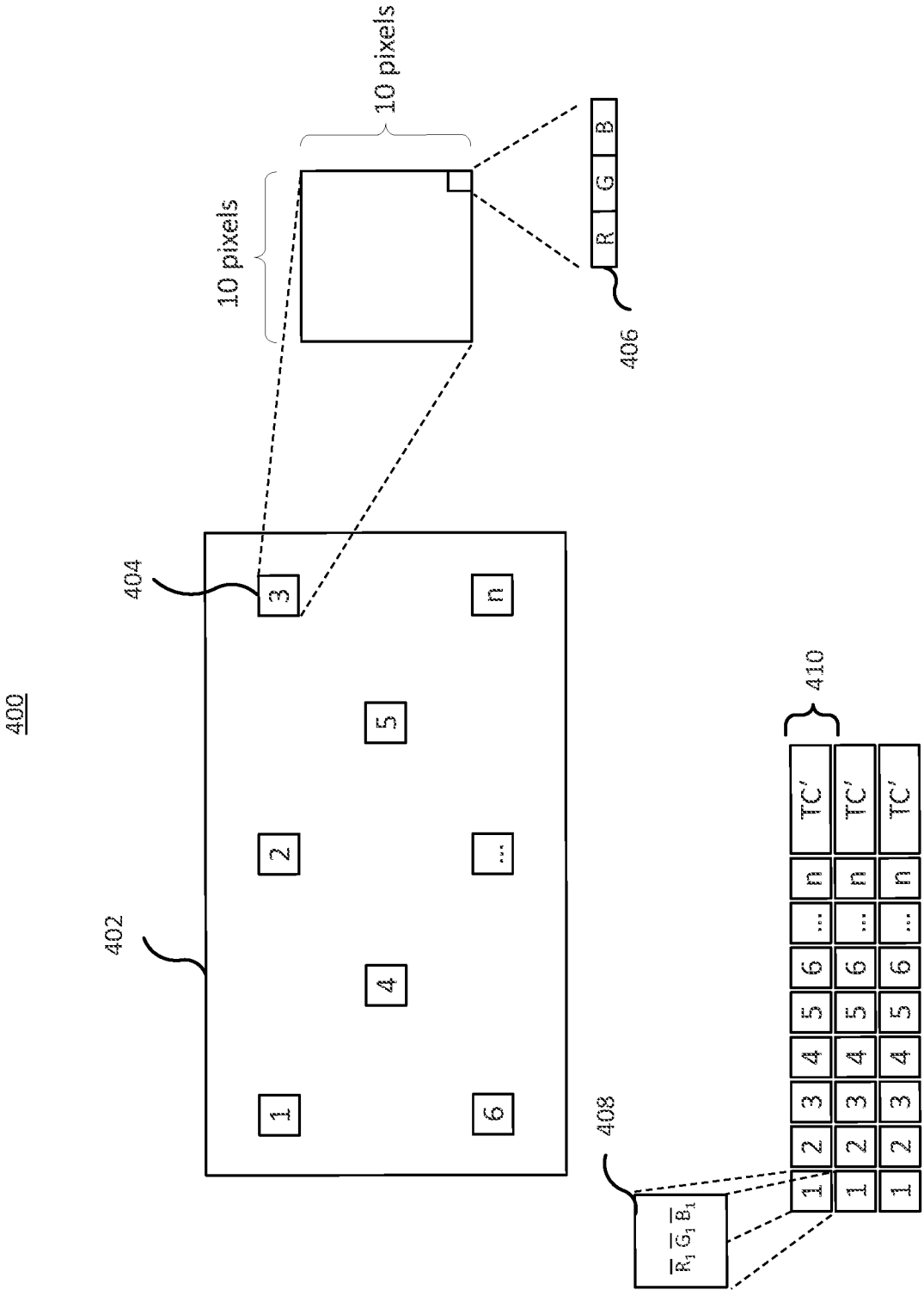


FIG. 4

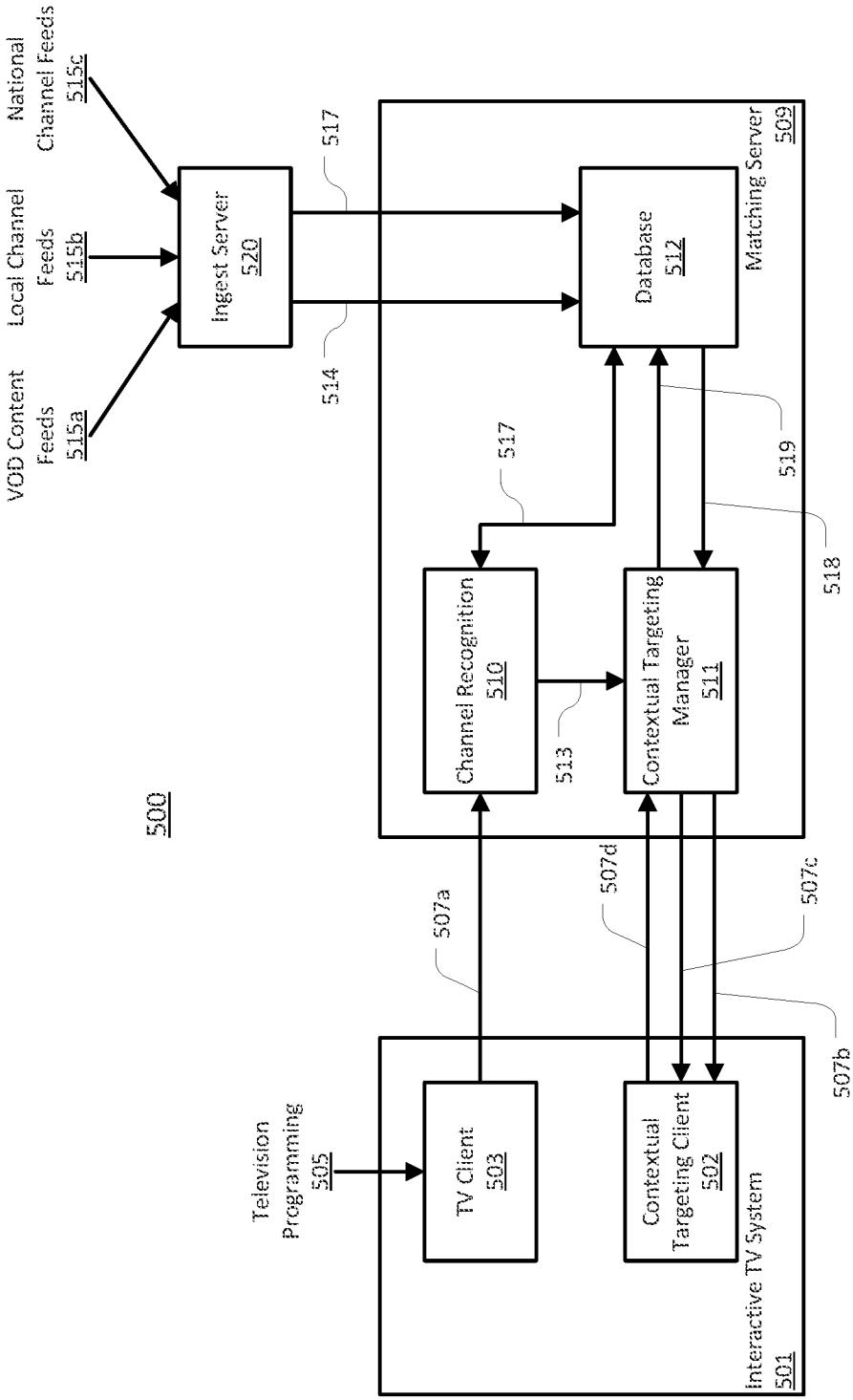

500
Interactive TV System
501
Matching Server
509
Television Programming
505
TV Client
503
Contextual Targeting Client
502
Channel Recognition
510
Contextual Targeting Manager
511
Database
512
Ingest Server
520
VOD Content Feeds
515a
Local Channel Feeds
515b
National Channel Feeds
515c
507a
507b
507c
507d
513
514
517
517
518
519

FIG. 5

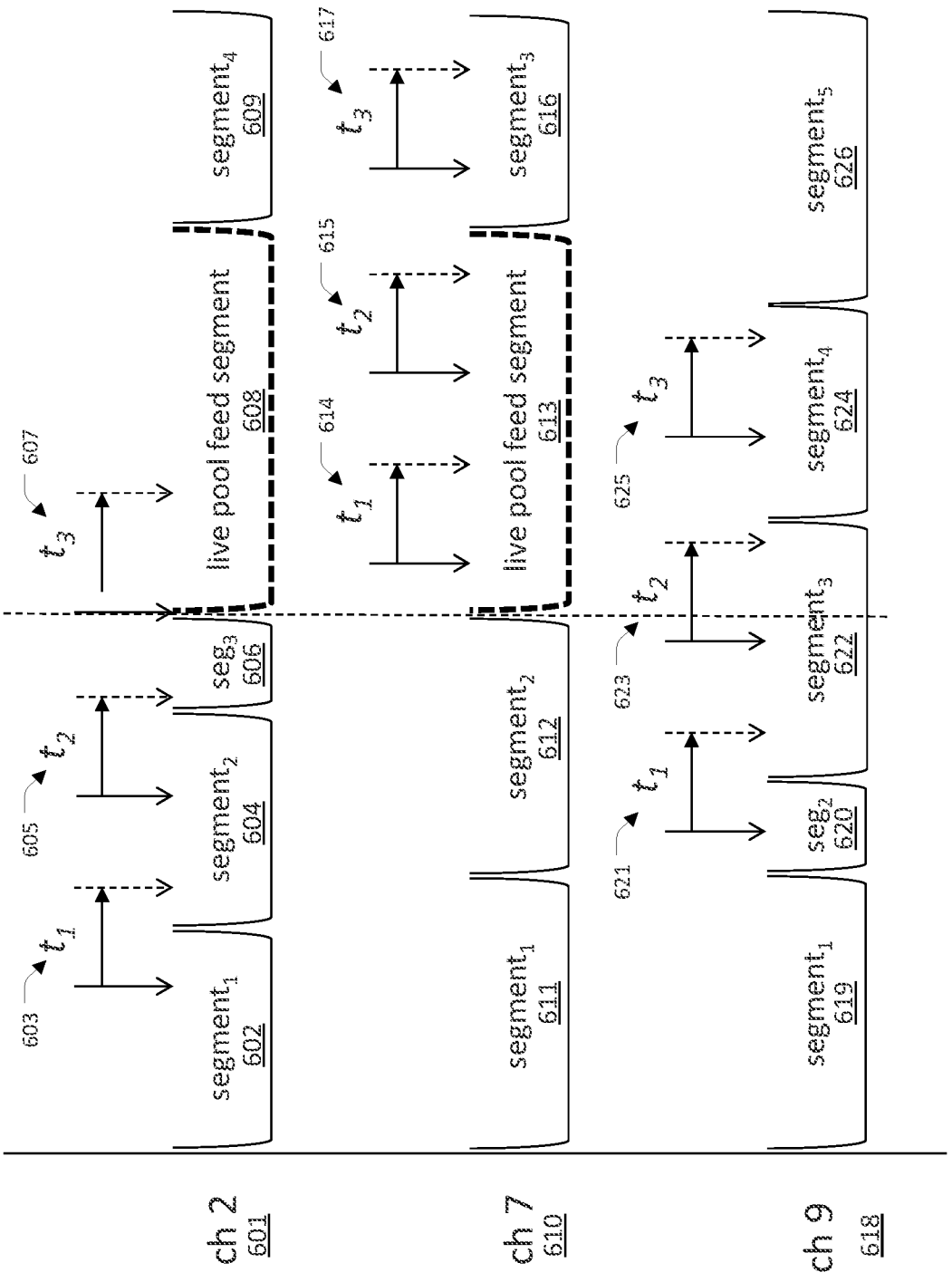

ch 2
601
ch 7
610
ch 9
618
segment1
602
segment2
604
seg3
606
segment4
609
live pool feed segment
608
segment1
611
segment2
612
live pool feed segment
613
segment3
616
segment1
619
seg2
620
segment3
622
segment4
624
segment5
626
603
605
607
614
615
617
621
623
625
t1
t2
t3

FIG. 6

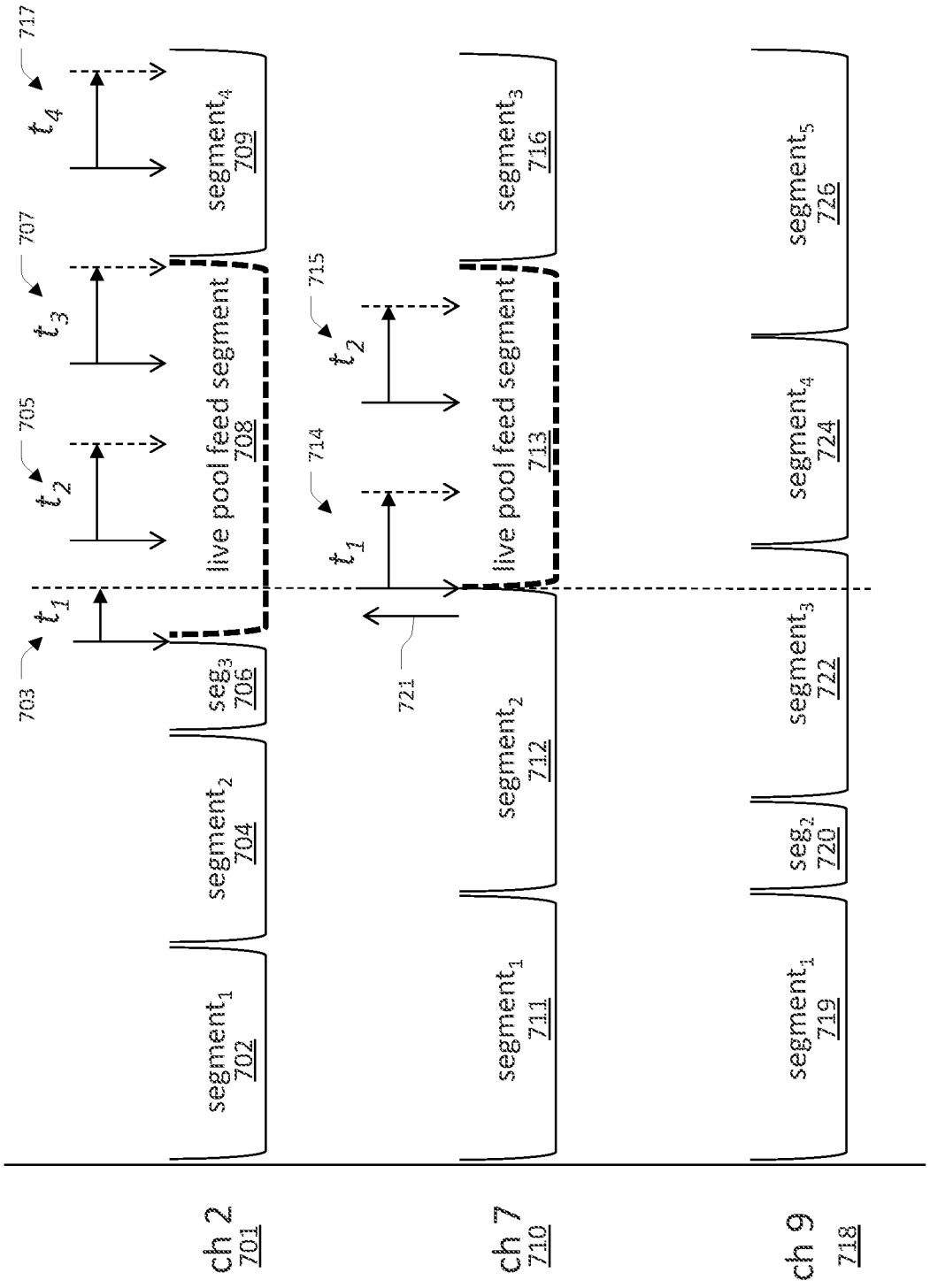

ch 2
701
ch 7
710
ch 9
718
segment1
702
segment2
704
seg3
706
segment4
709
segment1
711
segment2
712
segment3
716
segment1
719
seg2
720
segment3
722
segment4
724
segment5
726
live pool feed segment
708
live pool feed segment
713
t1
t2
t3
t4
703
705
707
717
714
715
721

FIG. 7

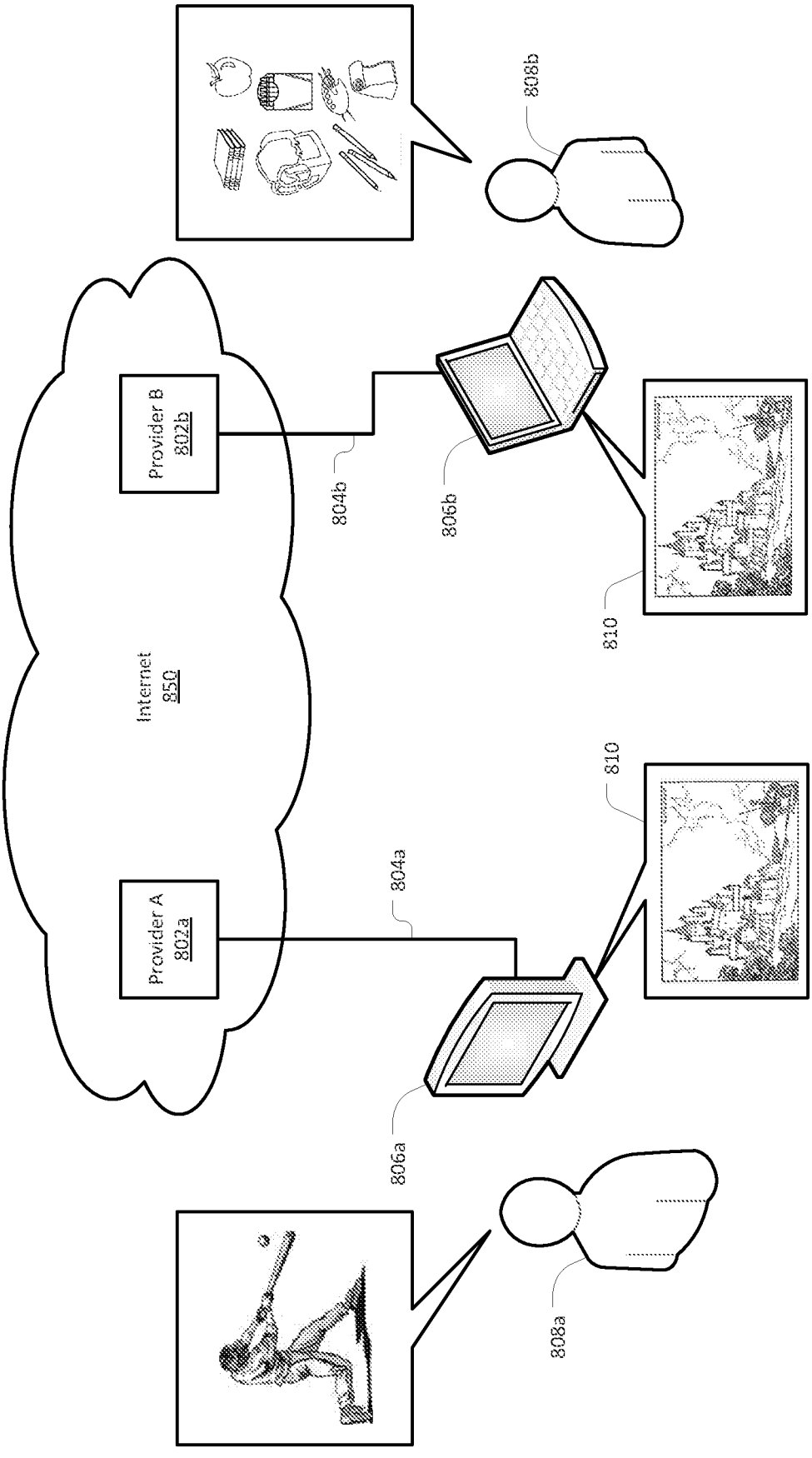

Internet
850
Provider A
802a
Provider B
802b
804a
804b
806a
805b
808a
808b
810
810

FIG. 8

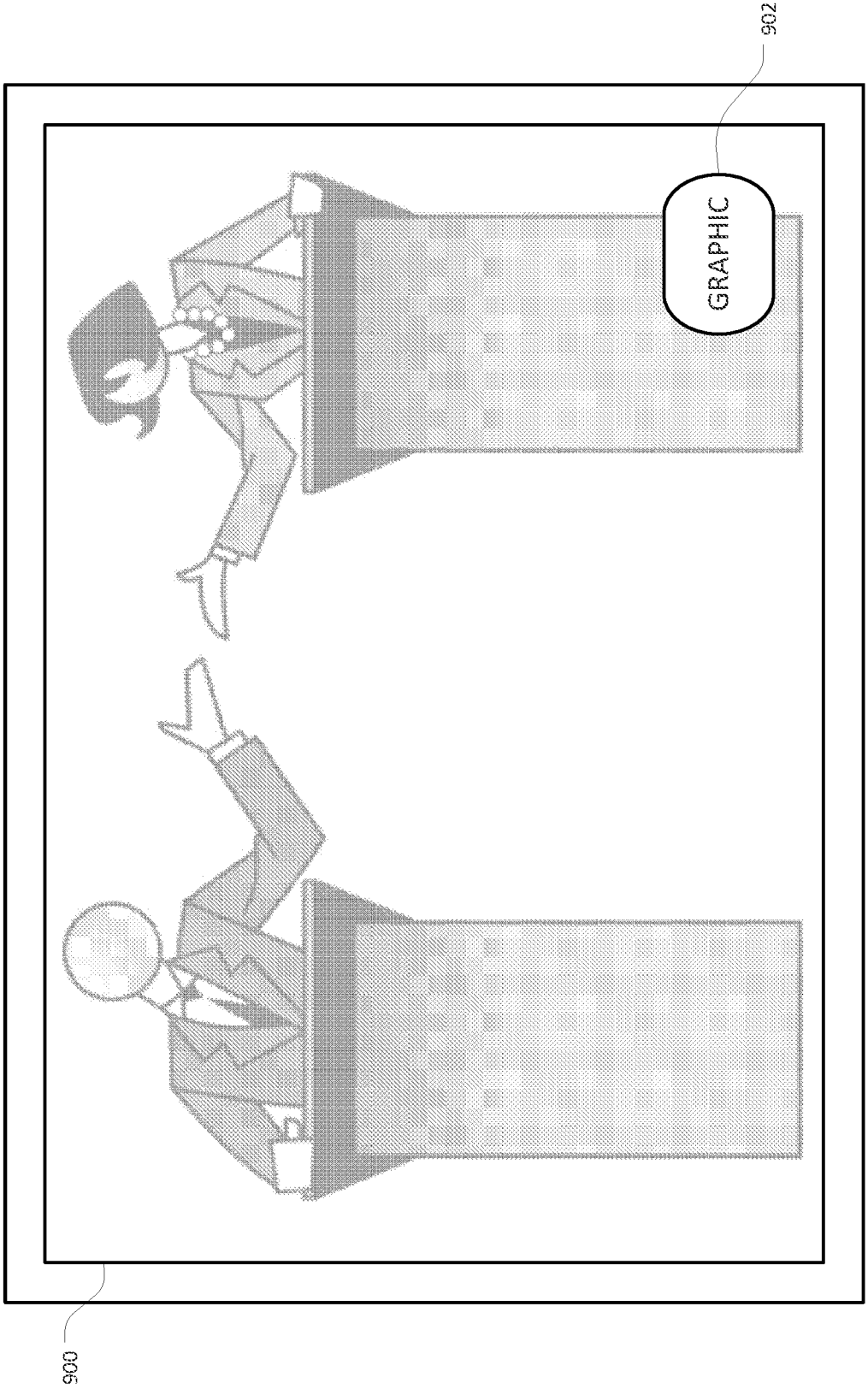


FIG. 9

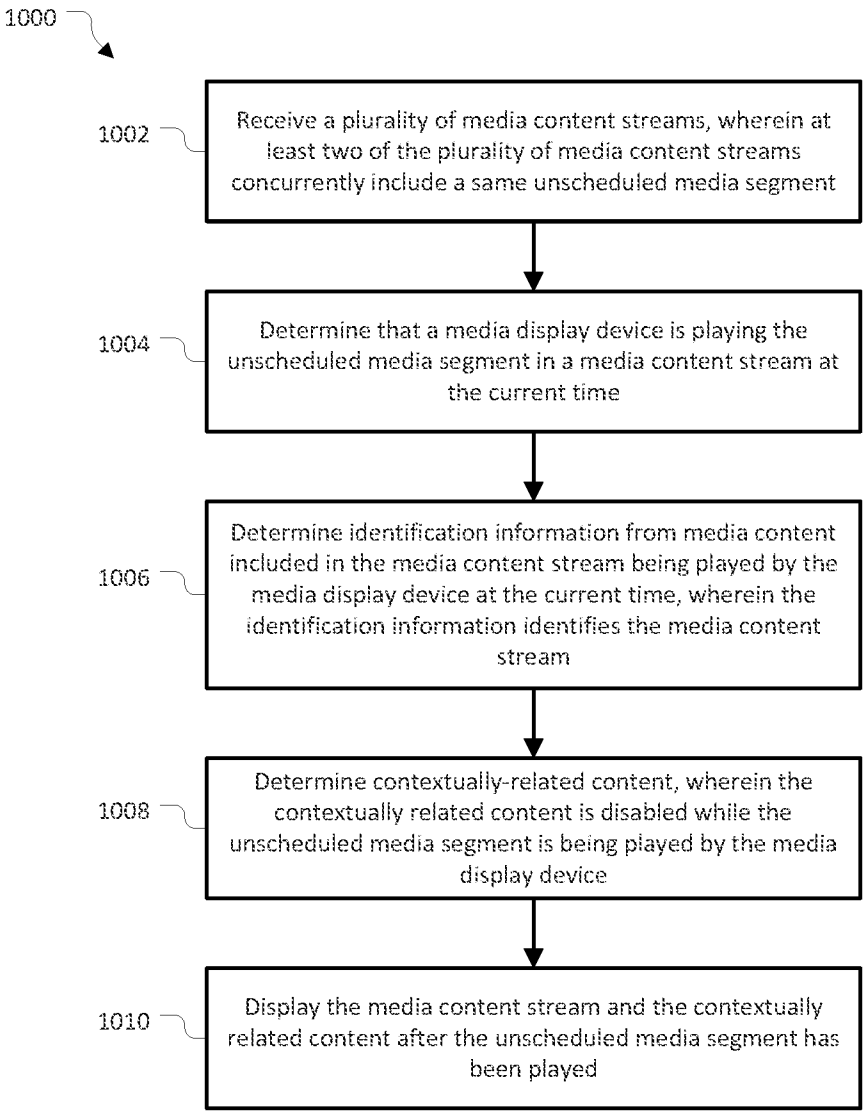

1000
1002
Receive a plurality of media content streams, wherein at least two of the plurality of media content streams concurrently include a same unscheduled media segment
1004
Determine that a media display device is playing the unscheduled media segment in a media content stream at the current time
1006
Determine identification information from media content included in the media content stream being played by the media display device at the current time, wherein the identification information identifies the media content stream
1008
Determine contextually-related content, wherein the contextually related content is disabled while the unscheduled media segment is being played by the media display device
1010
Display the media content stream and the contextually related content after the unscheduled media segment has been played

FIG. 10

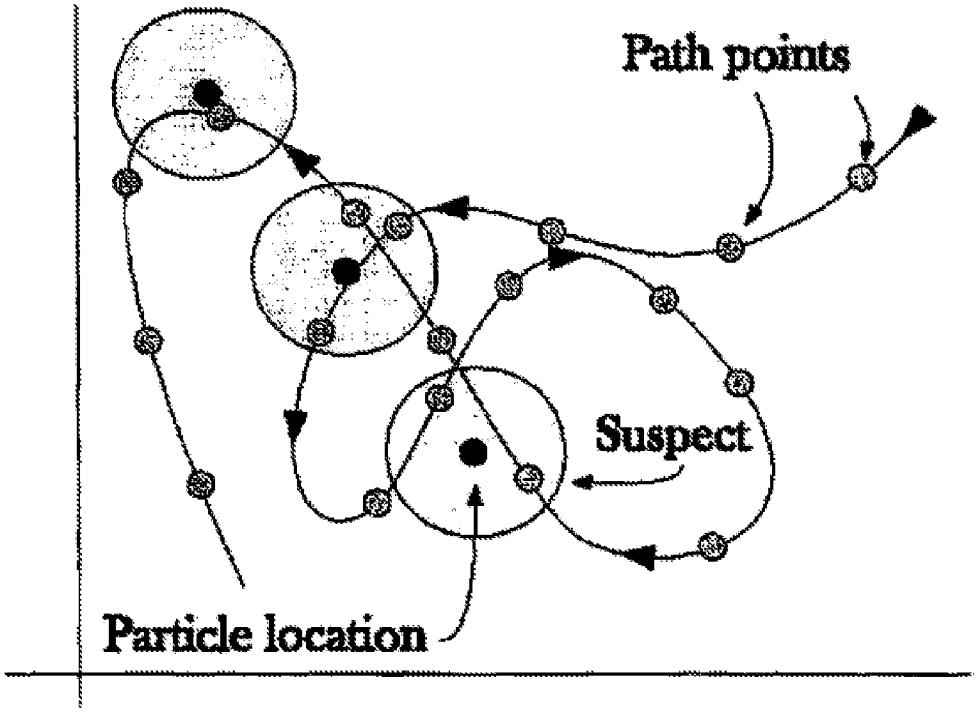


FIG. 11

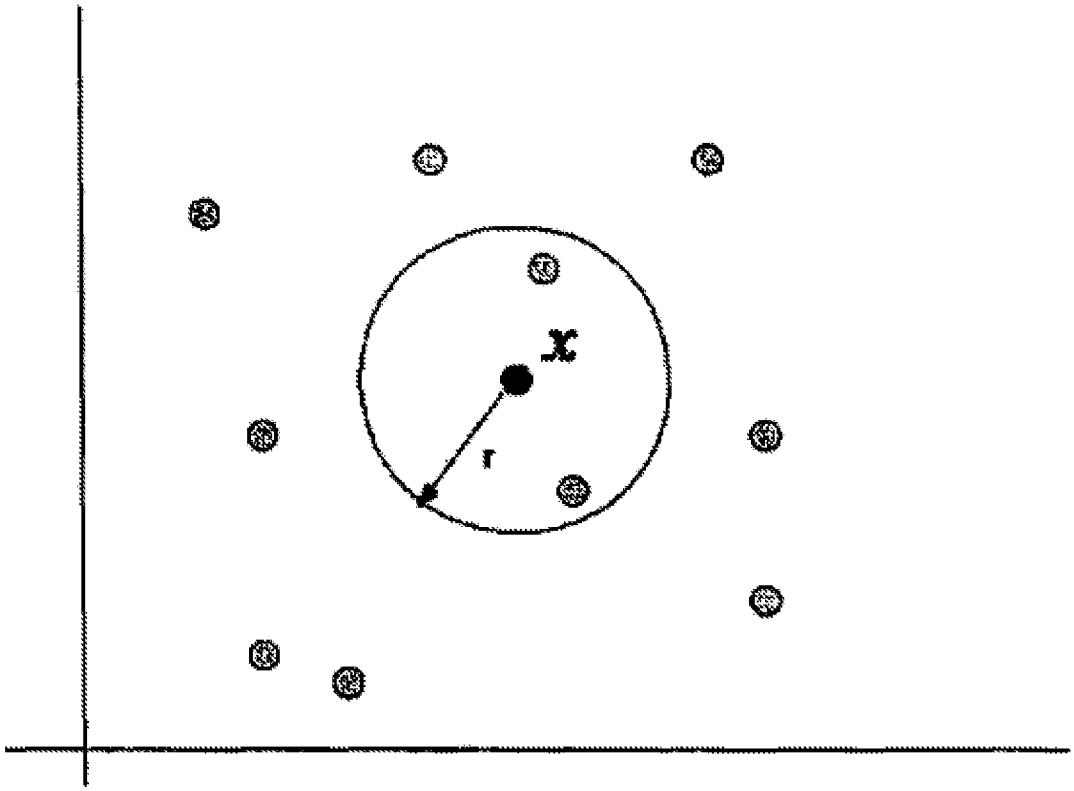


FIG. 12

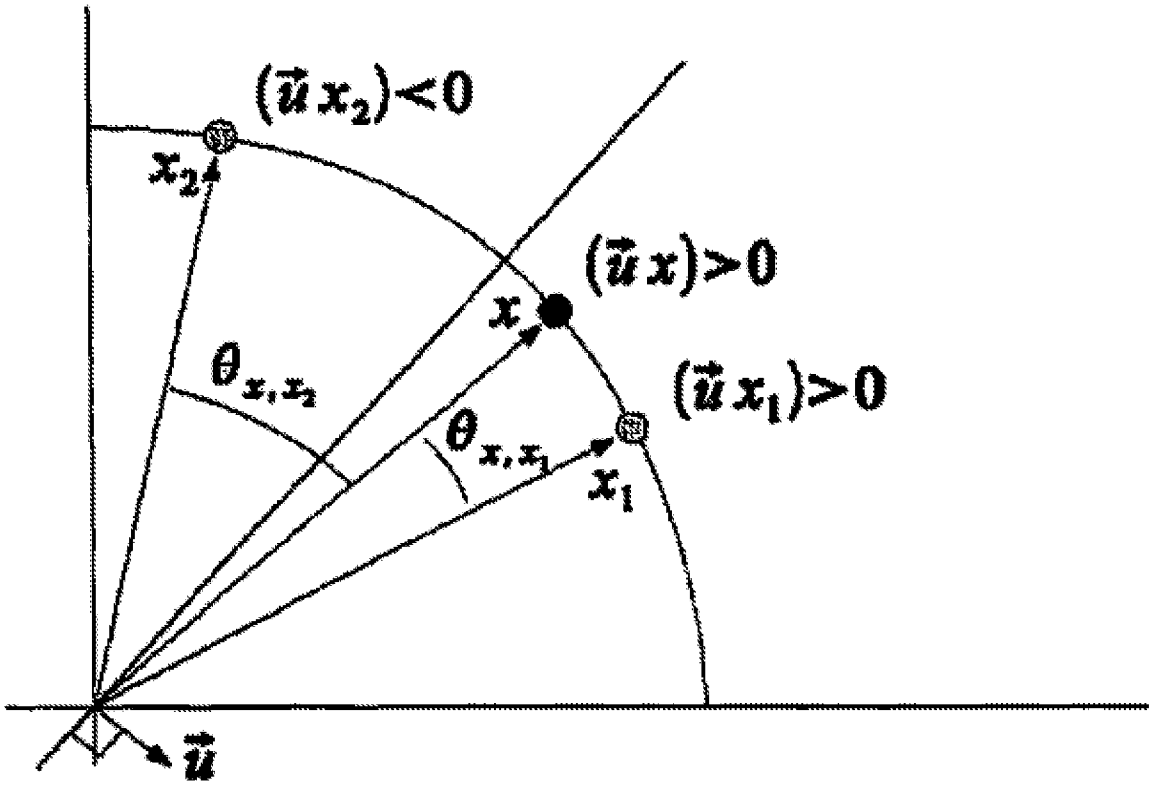


FIG. 13

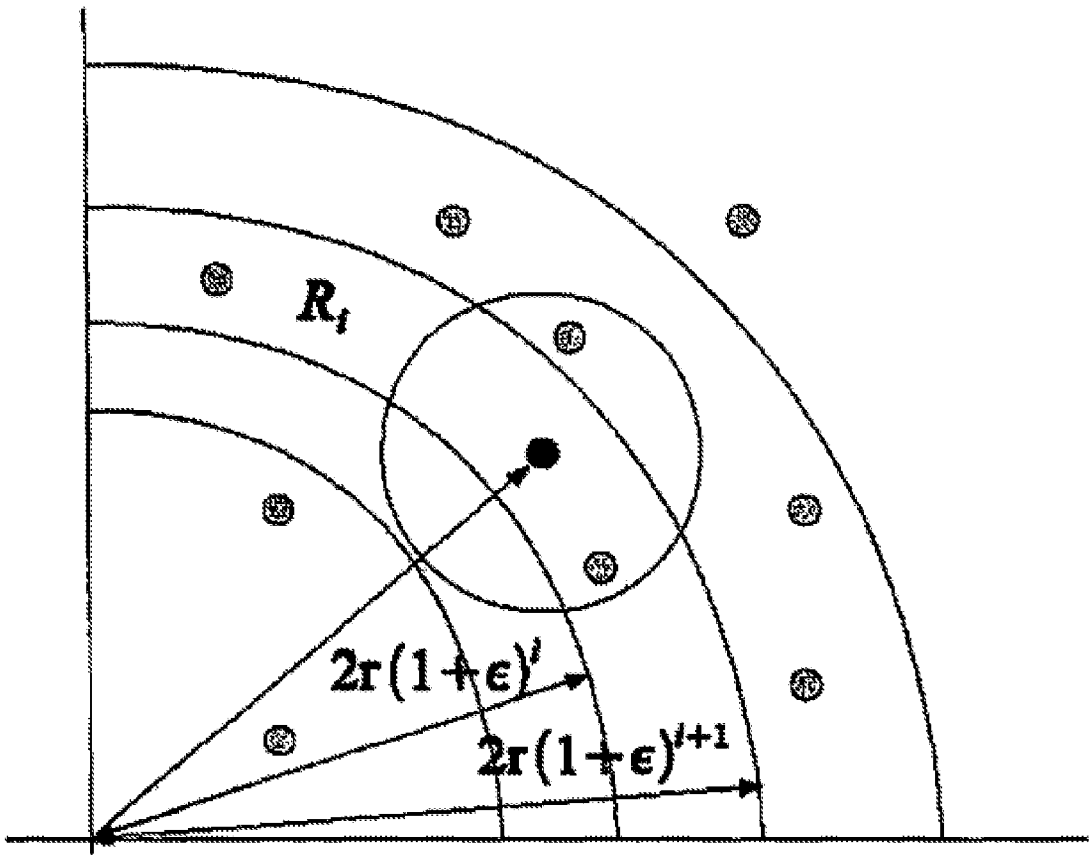


FIG. 14

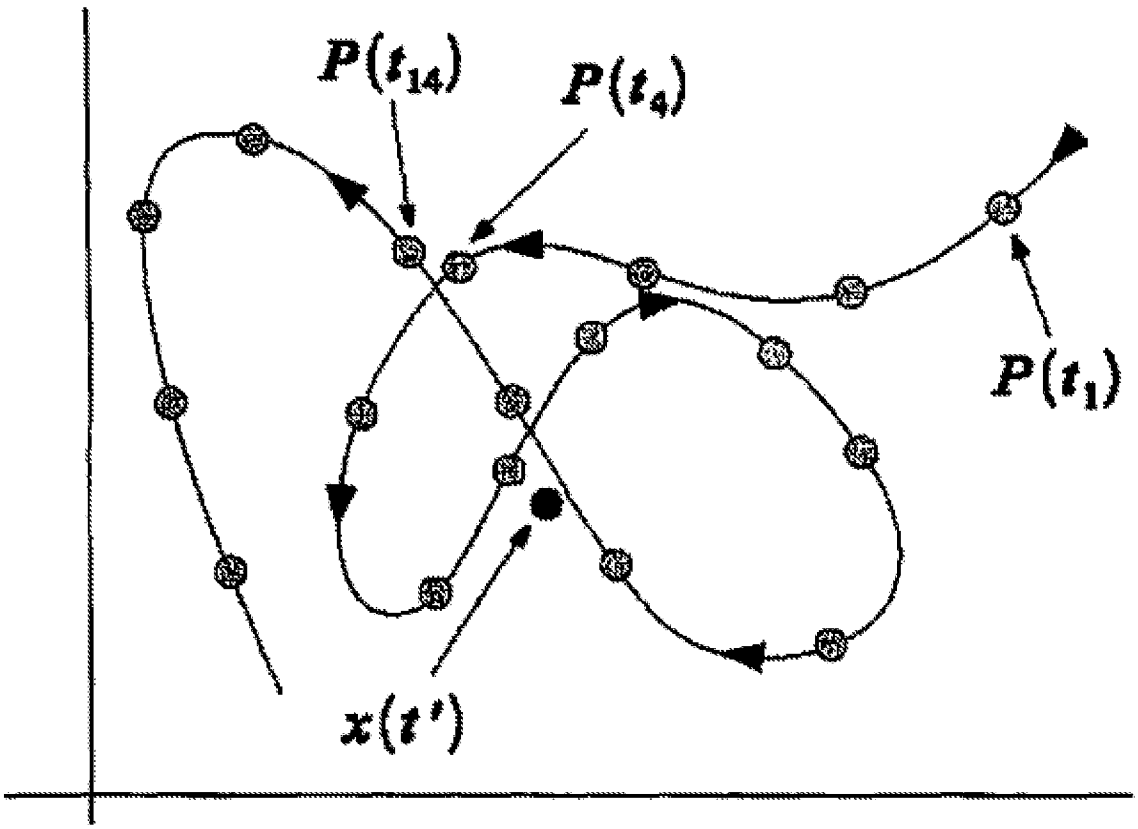


FIG. 15

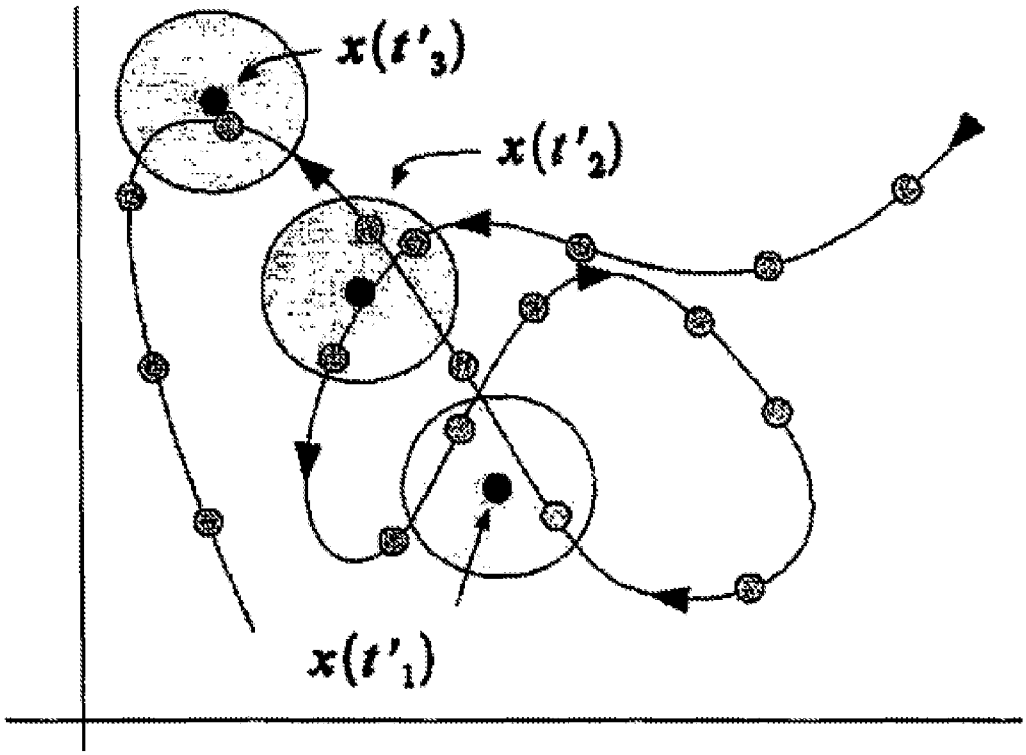


FIG. 16

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2016/042621

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N21/234 H04N21/262 H04N21/454 H04N21/81 H04N21/845
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)
H04N

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	US 2014/201769 A1 (NEUMEIER ZEEV [US] ET AL) 17 July 2014 (2014-07-17) paragraphs [0044] - [0057], [0078], [0089]; figures 1, 10	1-12, 15-20
X	US 2015/100979 A1 (MOSKOWITZ ALAN [US] ET AL) 9 April 2015 (2015-04-09) paragraphs [0006], [0021], [0029] - [0033]; figures 1,2	1,7,15
X	EP 2 541 963 A2 (TV INTERACTIVE SYSTEMS INC [US]) 2 January 2013 (2013-01-02) paragraphs [0021], [0061] - [0072], [0095]; figures 6,8,10	1,7, 13-15

☐ 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	Date of mailing of the international search report
30 September 2016	11/10/2016

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Hindelang, Thomas

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2016/042621

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2014201769 A1	17-07-2014	US 2014201769 A1	17-07-2014
		US 2014201772 A1	17-07-2014
		US 2014201773 A1	17-07-2014
		US 2014201774 A1	17-07-2014
		US 2014201787 A1	17-07-2014
US 2015100979 A1	09-04-2015	NONE	
EP 2541963 A2	02-01-2013	CN 102771115 A	07-11-2012
		EP 2520084 A2	07-11-2012
		EP 2541963 A2	02-01-2013
		WO 2011090540 A2	28-07-2011
		WO 2011090541 A2	28-07-2011