



(19) **United States**

(12) **Patent Application Publication**  
**Purdy**

(10) **Pub. No.: US 2012/0047152 A1**

(43) **Pub. Date: Feb. 23, 2012**

(54) **SYSTEM AND METHOD FOR PROFILE TAILORING IN AN AGGREGATE PROFILING SYSTEM**

**Publication Classification**

(51) **Int. Cl.** *G06F 17/30* (2006.01)  
(52) **U.S. Cl.** ..... 707/754; 707/E17.059

(75) **Inventor:** Sean T. Purdy, Durham, NC (US)

(57) **ABSTRACT**

(73) **Assignee:** Waldeck Technology, LLC, Wilmington, DE (US)

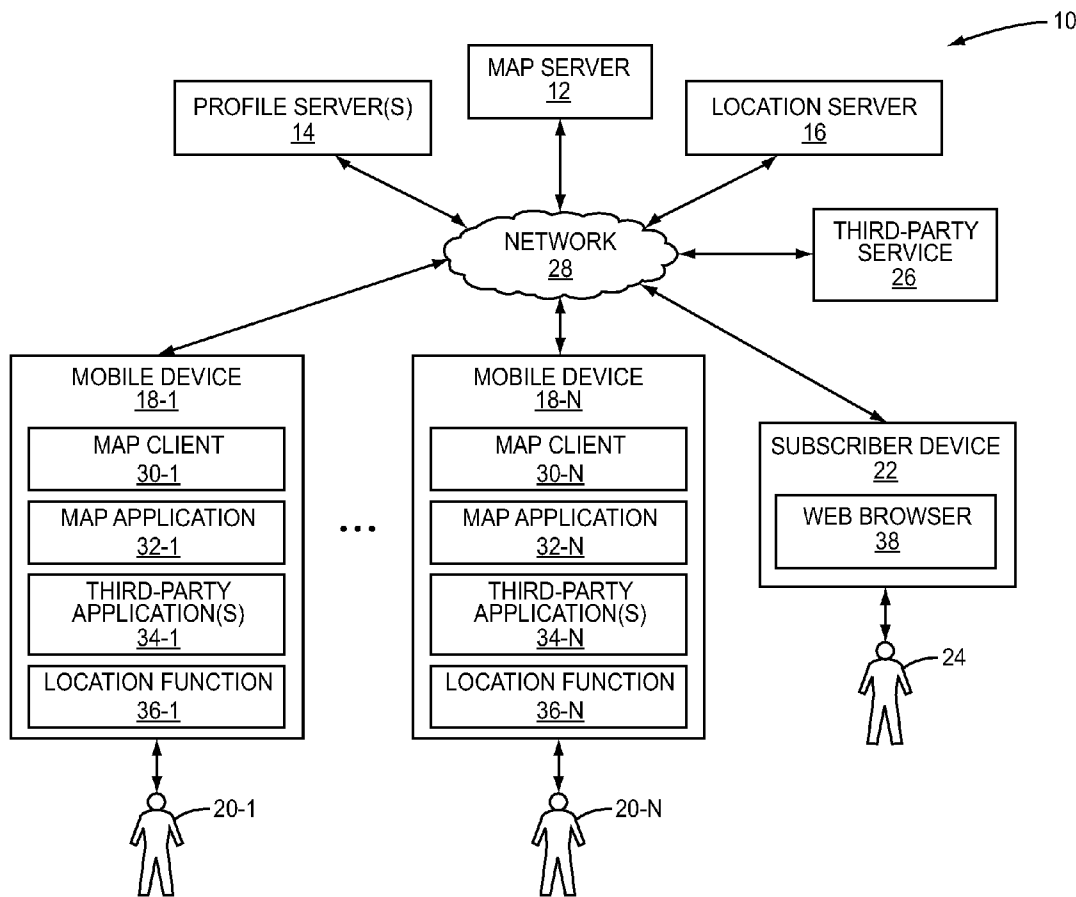
Systems and methods are disclosed for tailoring user profiles of users in a mobile aggregate profiling system. In one embodiment, the user profiles of the users in the mobile aggregate profiling system are tailored based on aggregate profile data relevant to current locations of the users. More specifically, in order to tailor the user profile of a user, aggregate profile data is obtained for the current location of the user. The user profile of the user is then filtered based on the aggregate profile data for the current location of the user. In one embodiment, the user profile is filtered at a server of the mobile aggregate profiling system. In another embodiment, the user profile is filtered at a mobile device of the user prior to sending the user profile to a server of the mobile aggregate profiling system.

(21) **Appl. No.:** 12/769,031

(22) **Filed:** Apr. 28, 2010

**Related U.S. Application Data**

(60) Provisional application No. 61/173,625, filed on Apr. 29, 2009.



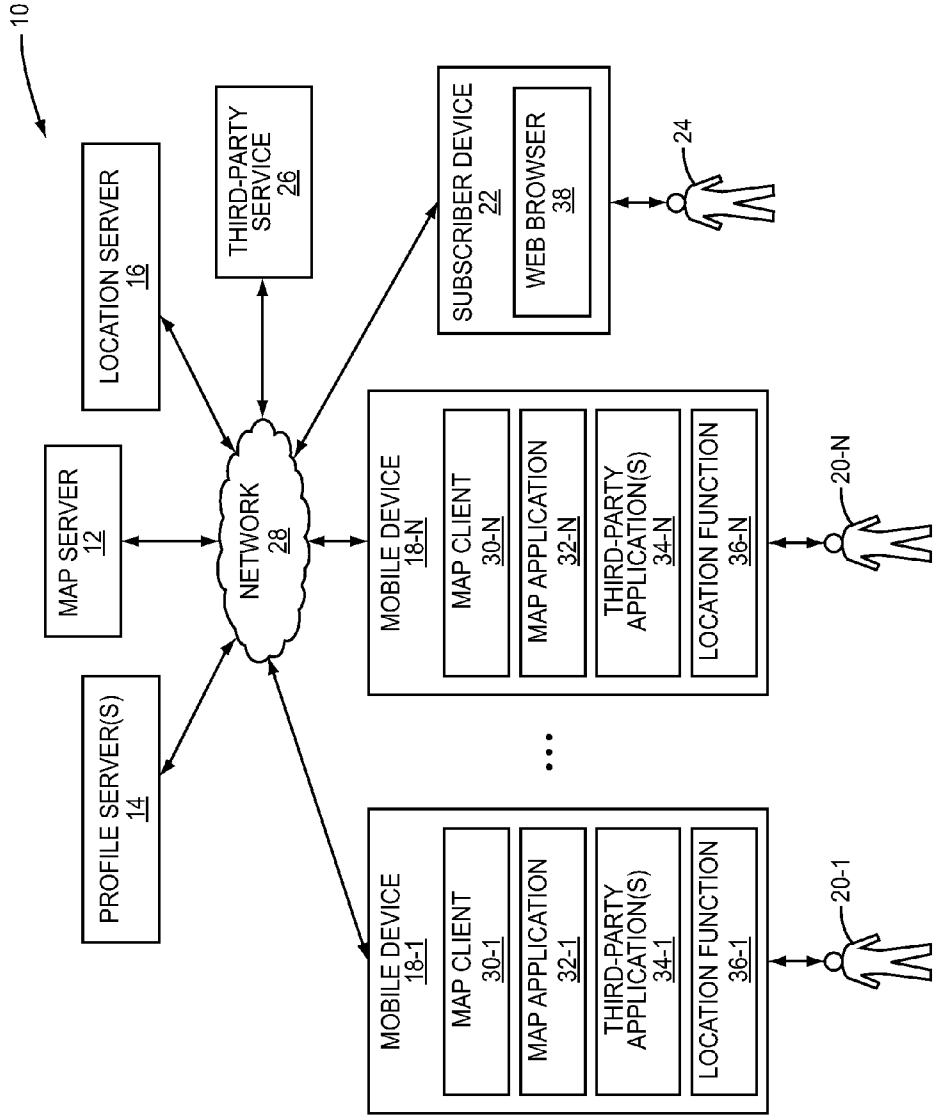


FIG. 1

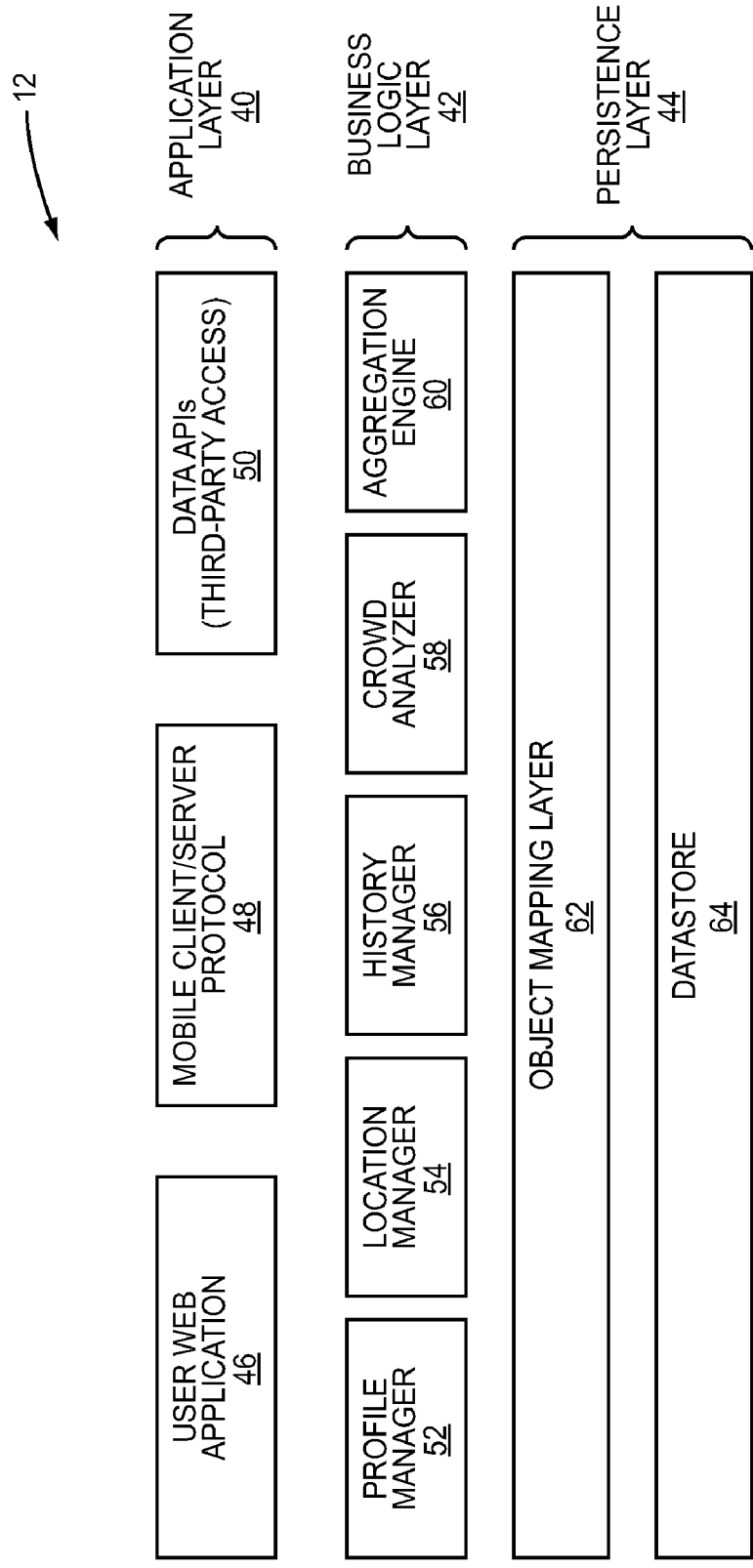
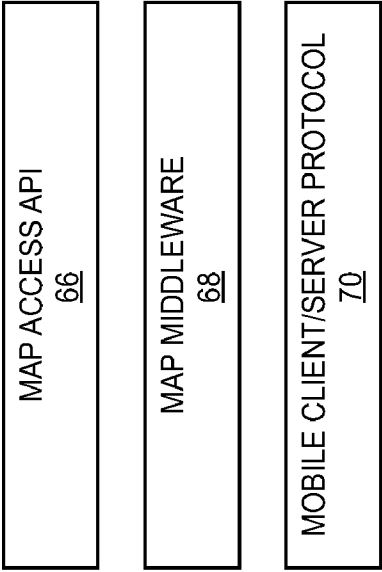
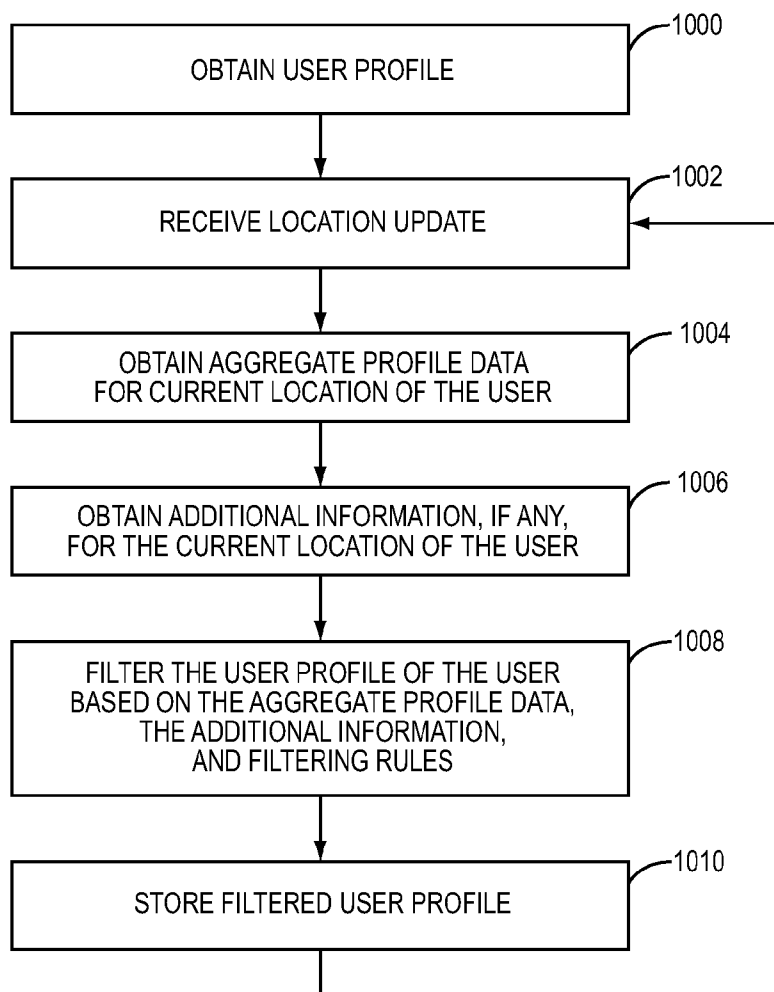


FIG. 2

30-1  
↘



**FIG. 3**



**FIG. 4**

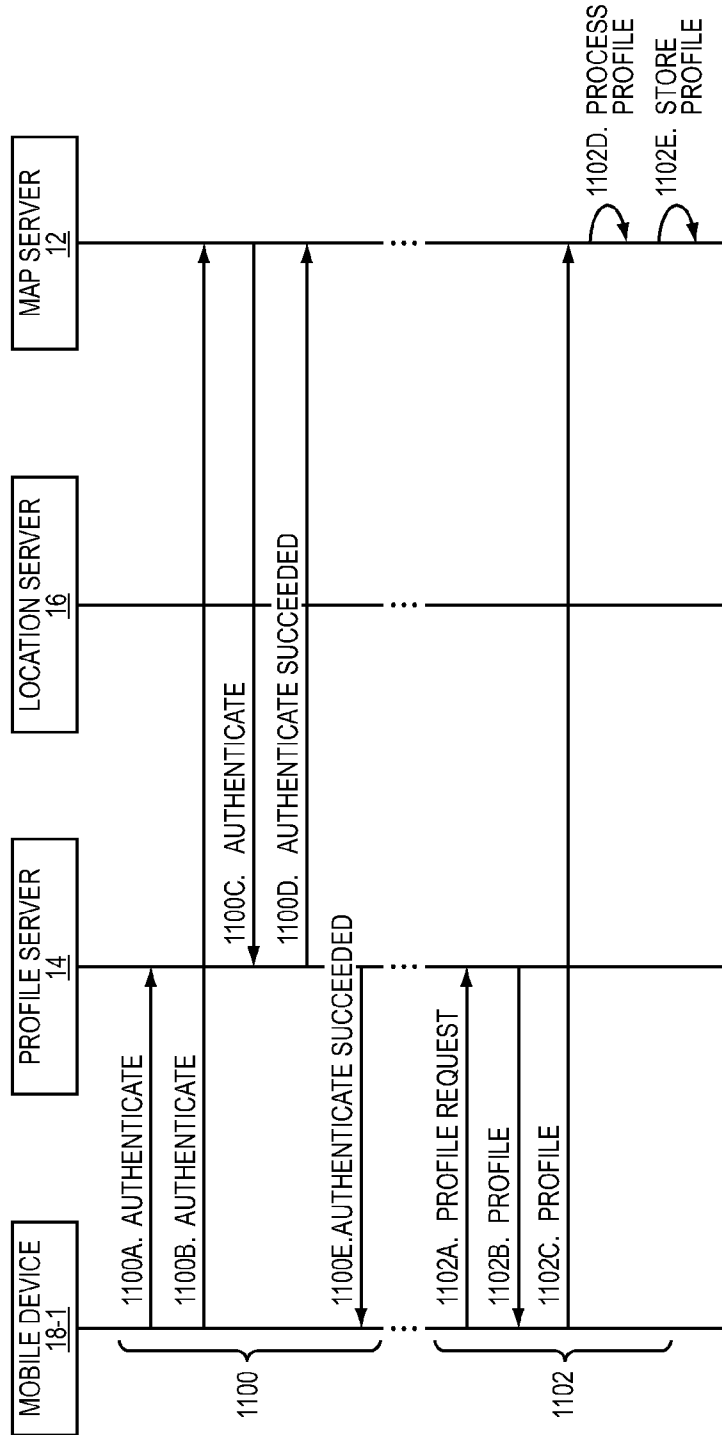


FIG. 5

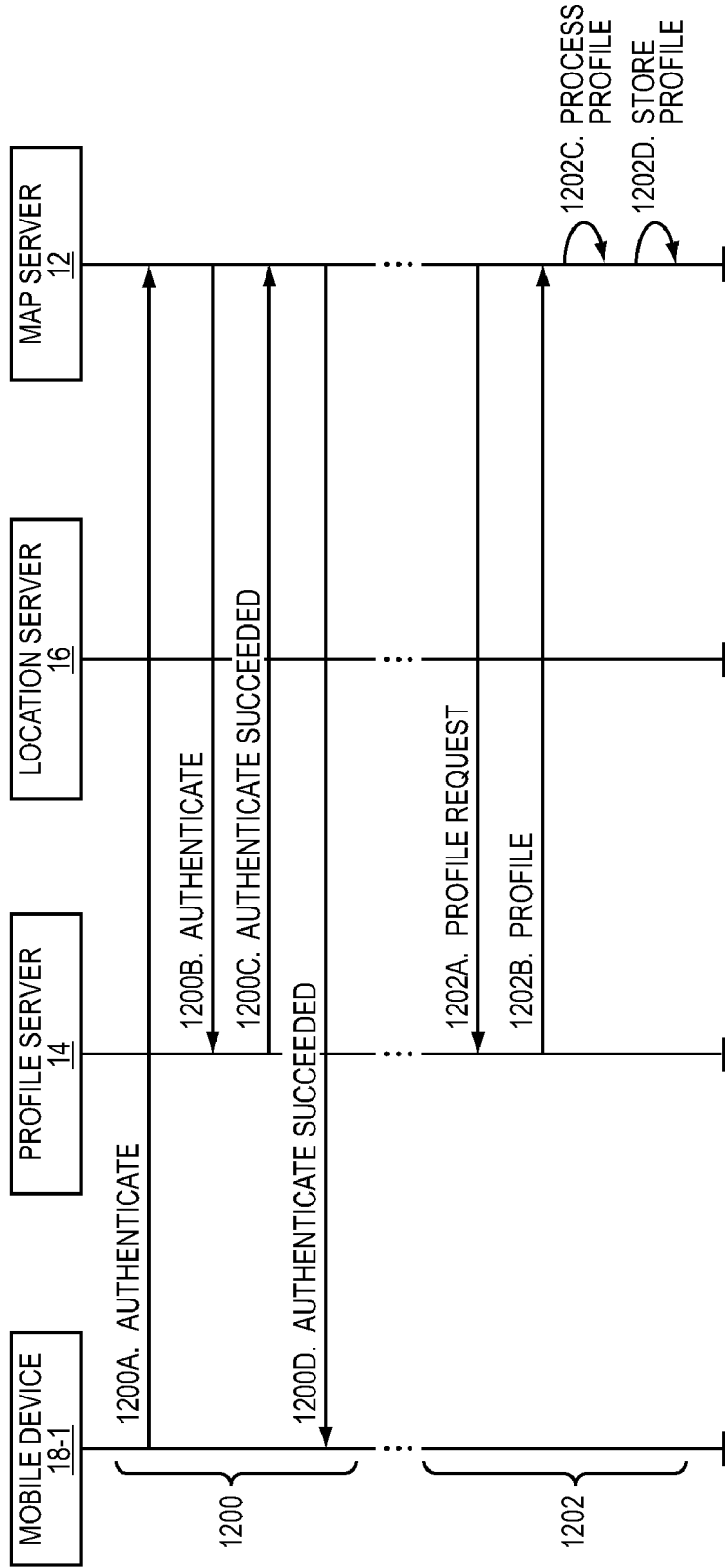


FIG. 6

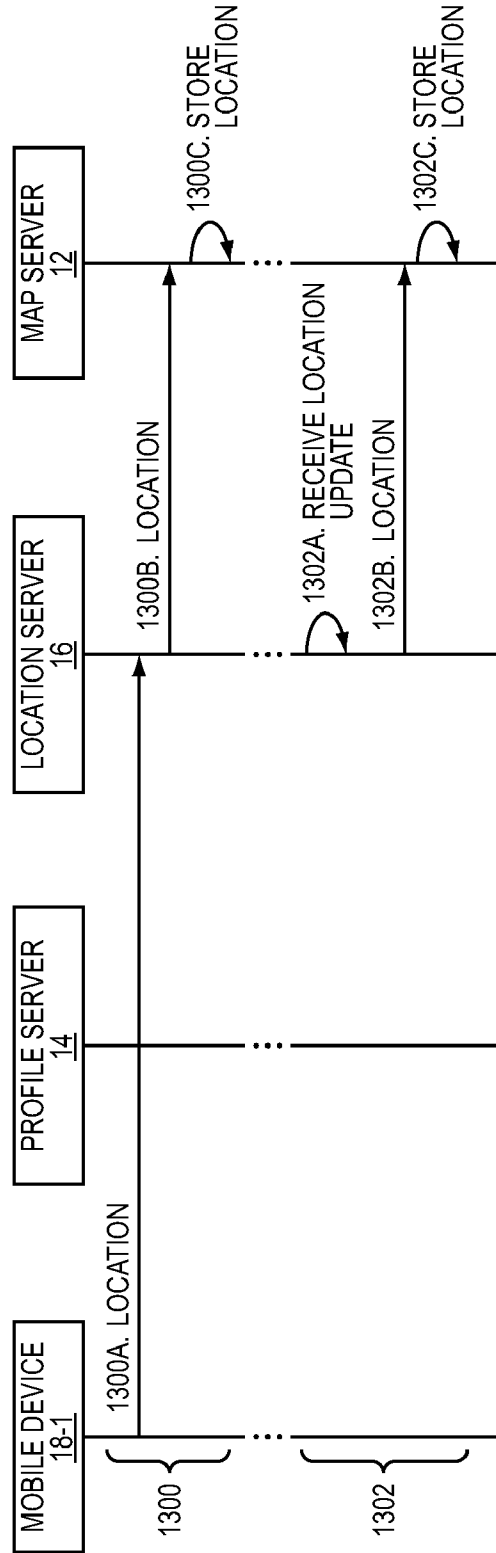
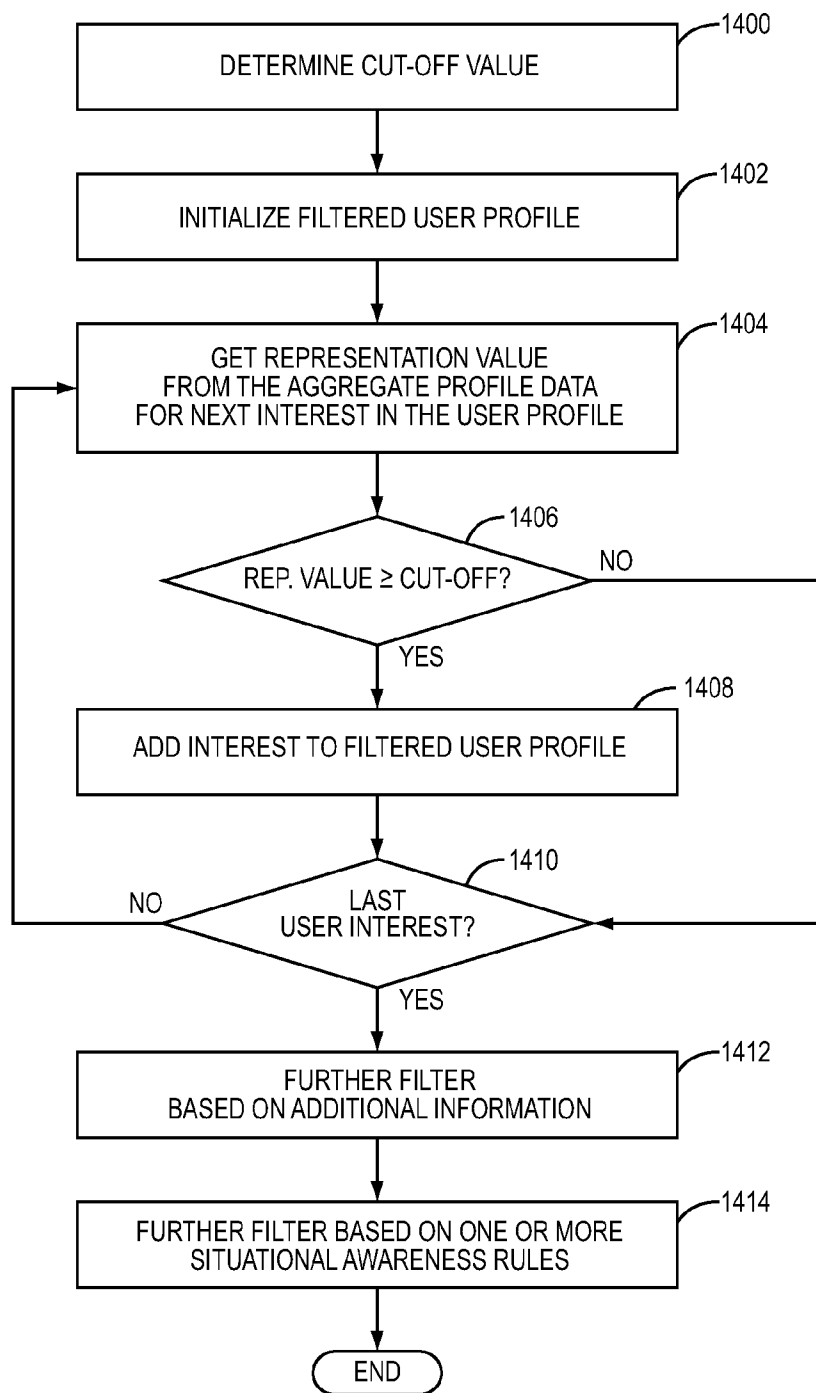
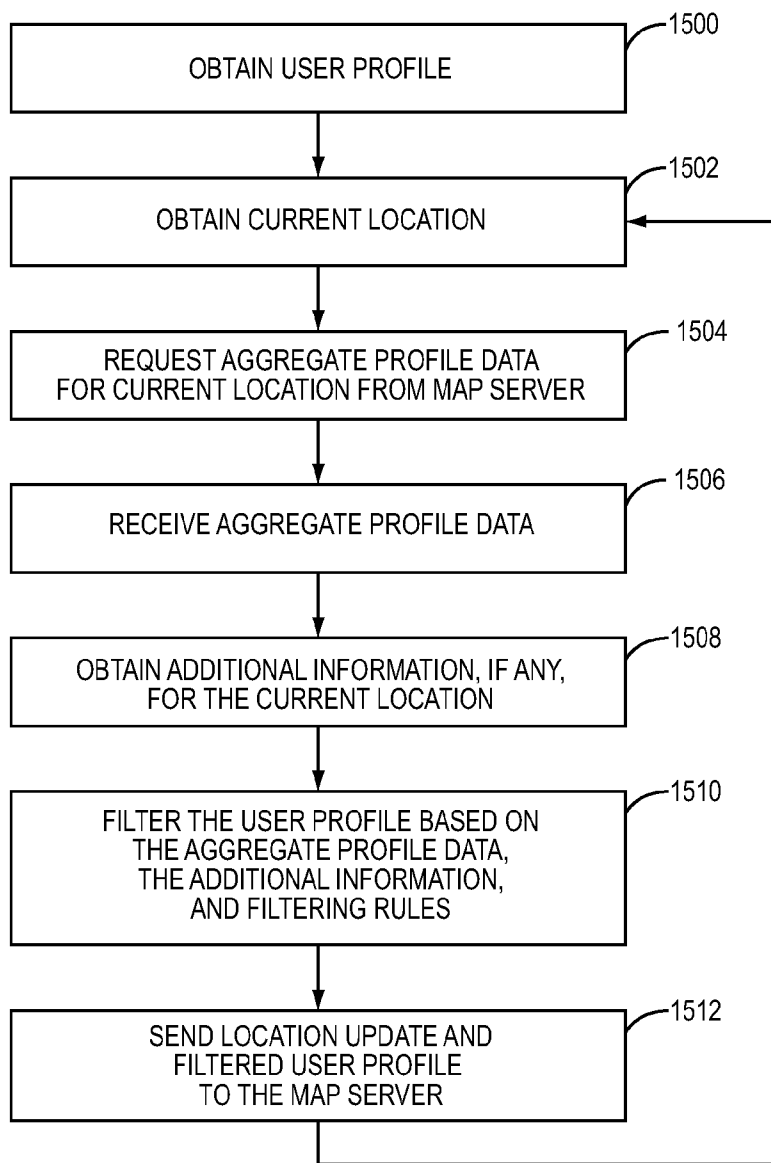


FIG. 7





**FIG. 8**



**FIG. 9**

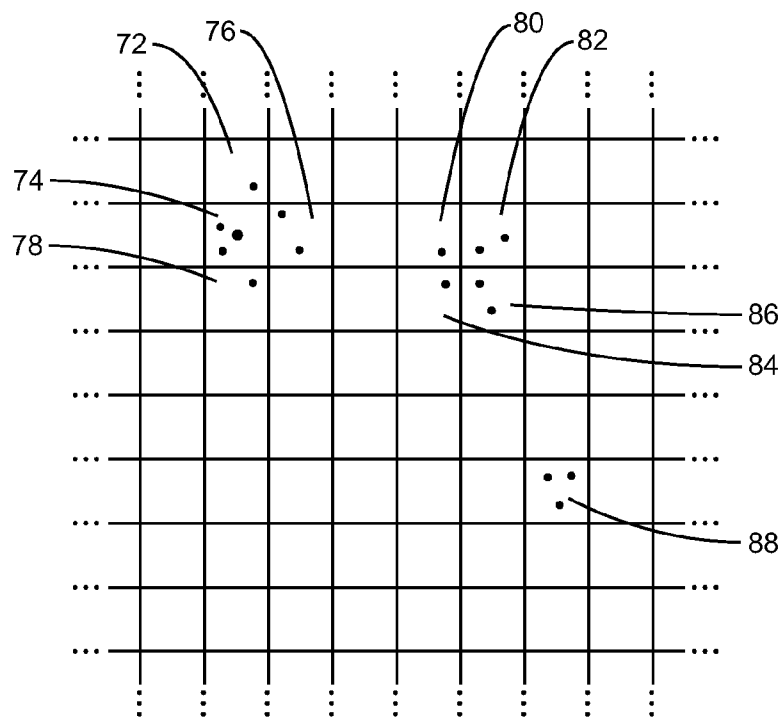


FIG. 10

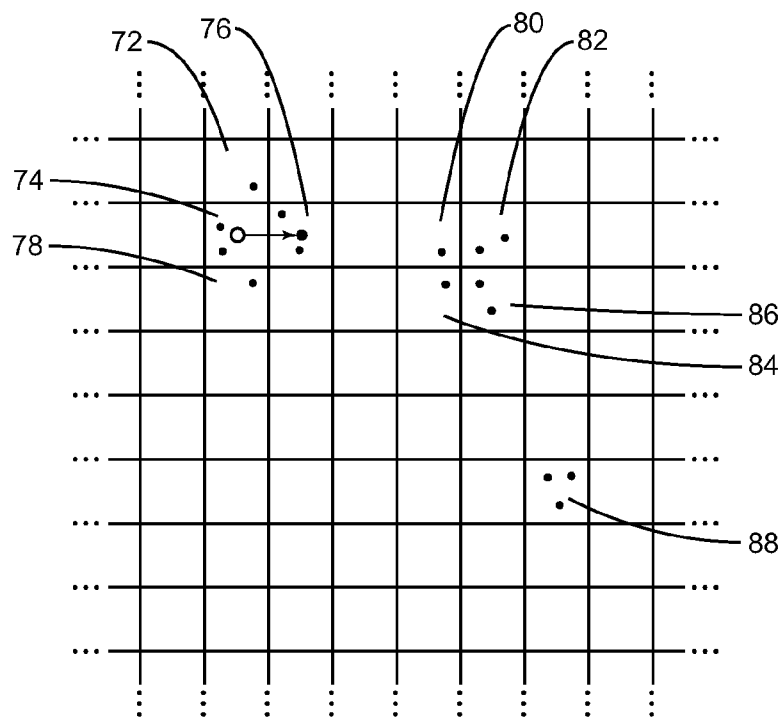


FIG. 11

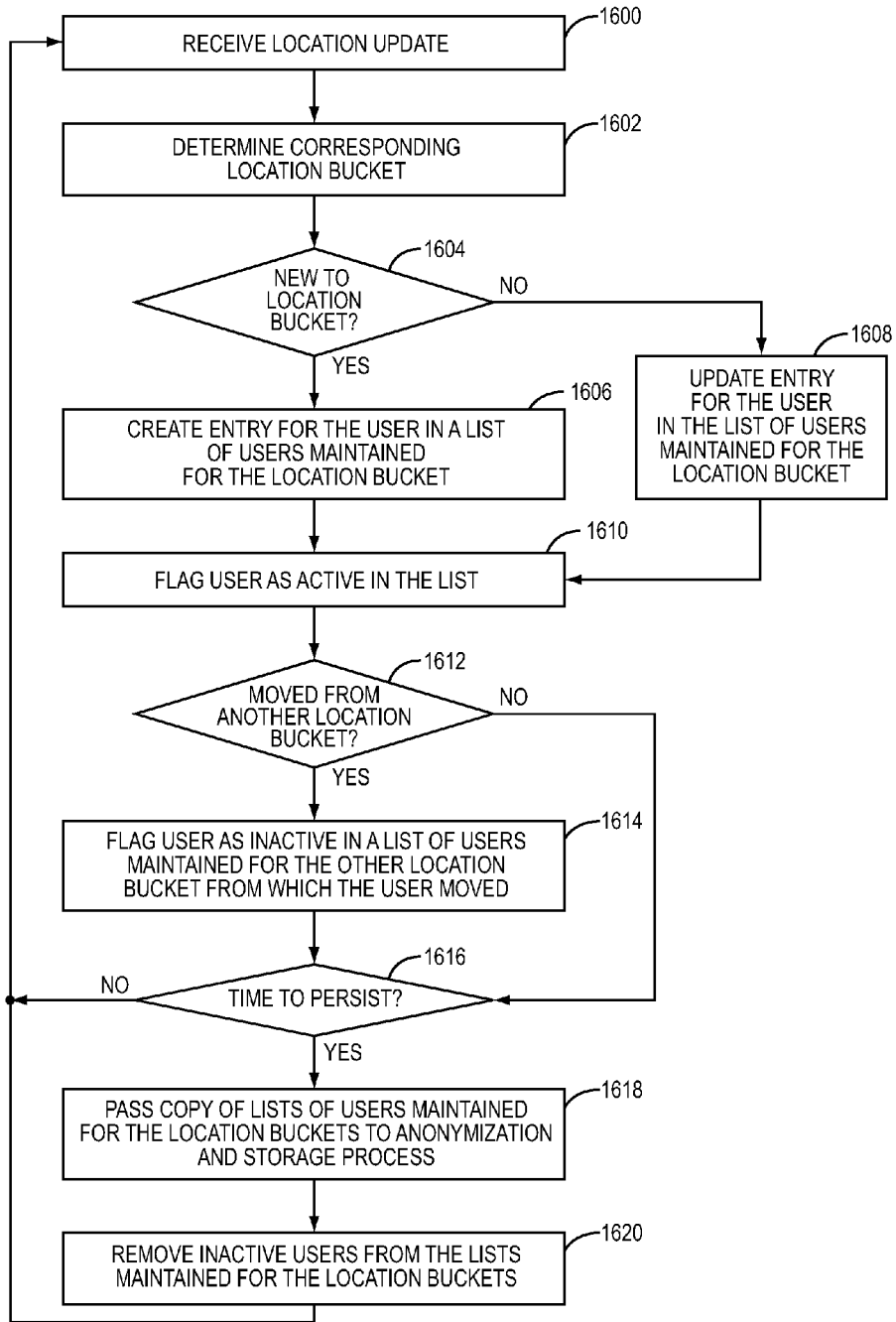
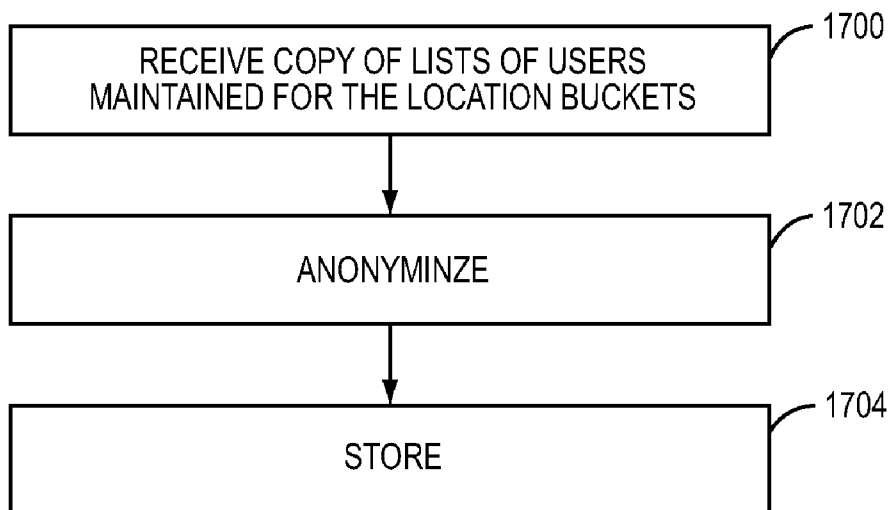


FIG. 12



**FIG. 13**

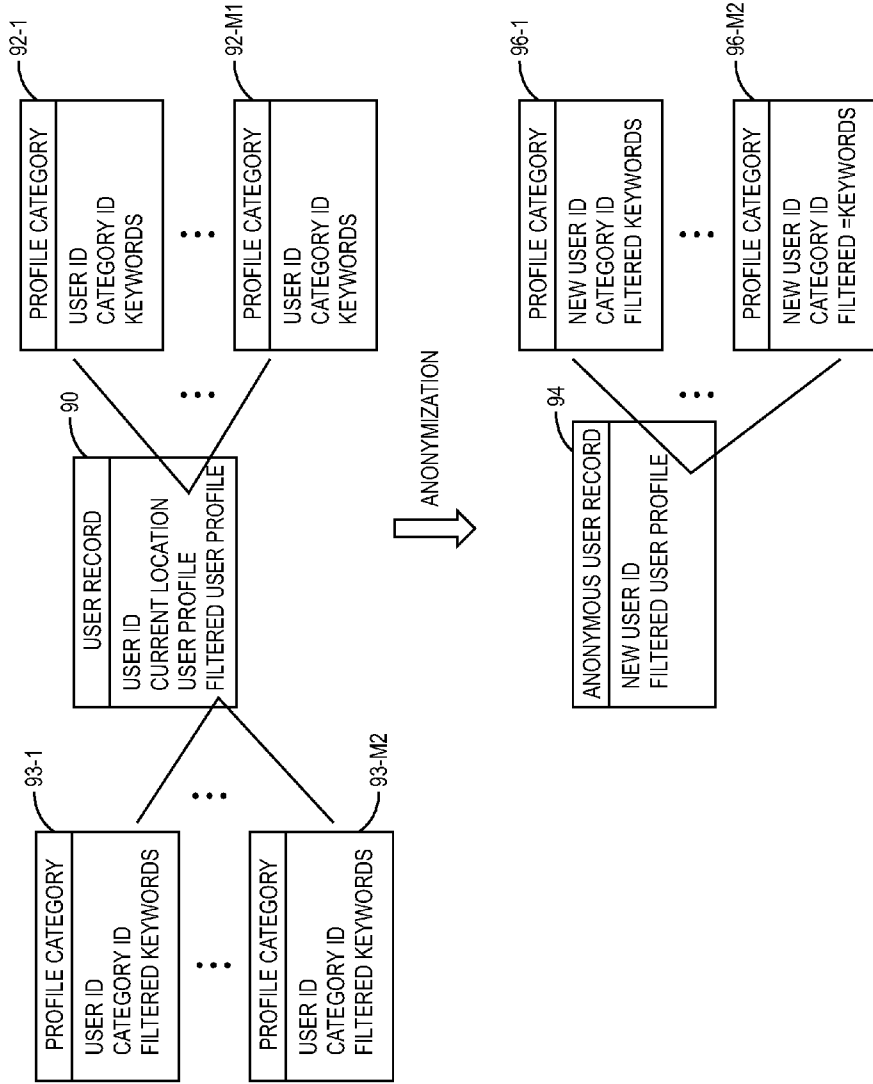
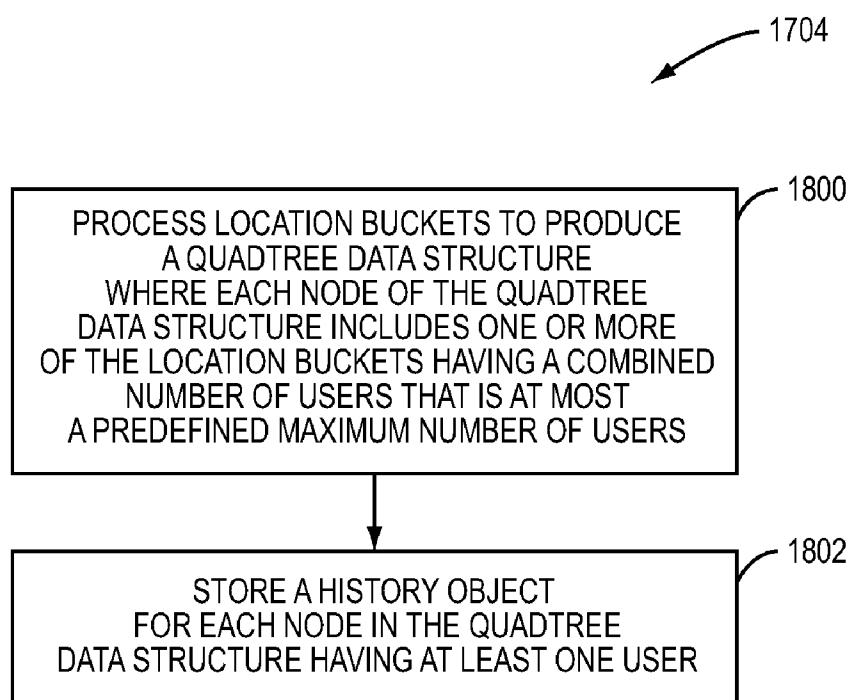


FIG. 14



**FIG. 15**

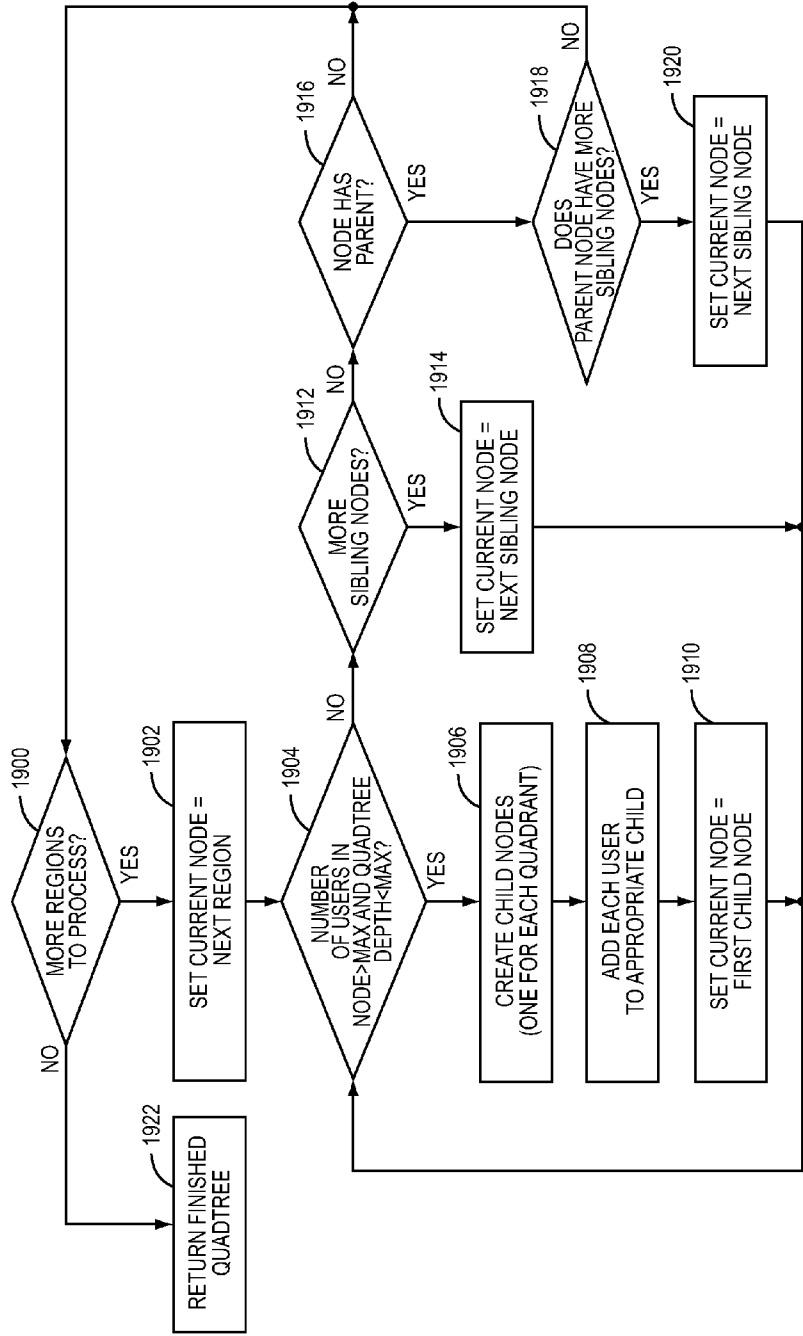


FIG. 16



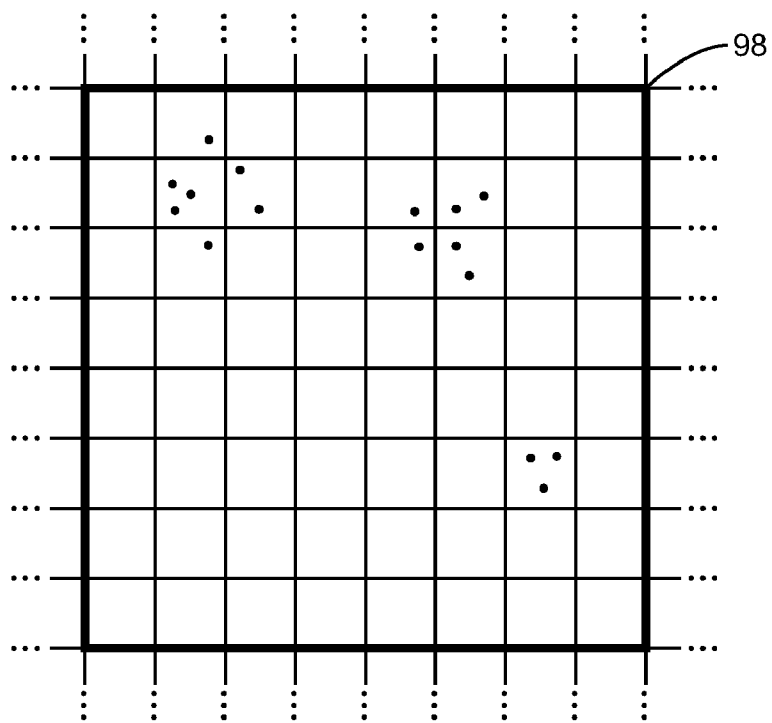


FIG. 17A

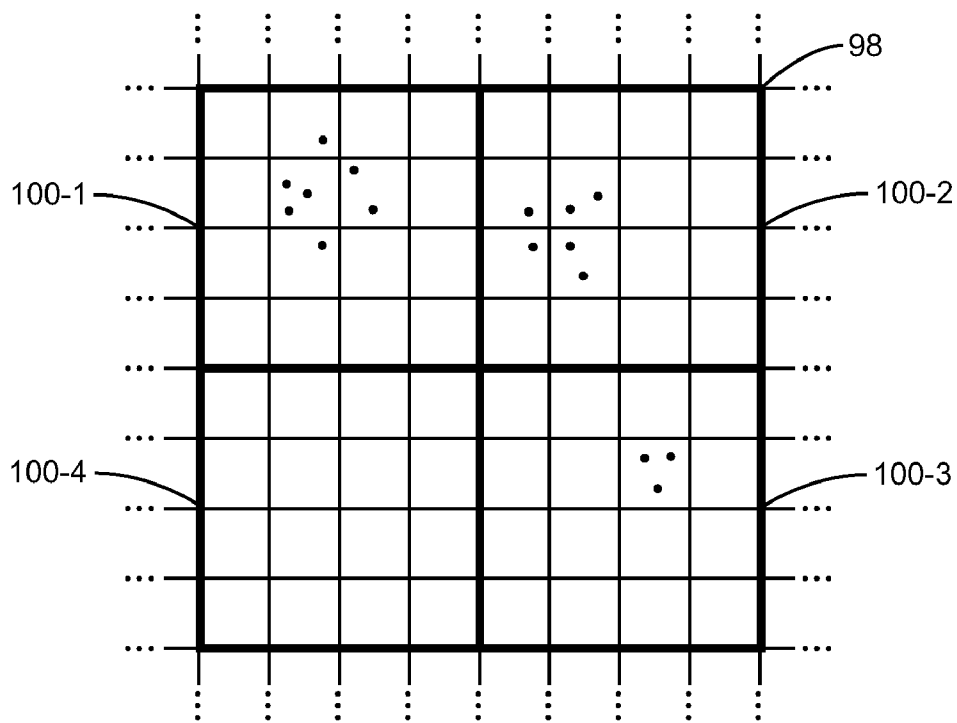


FIG. 17B

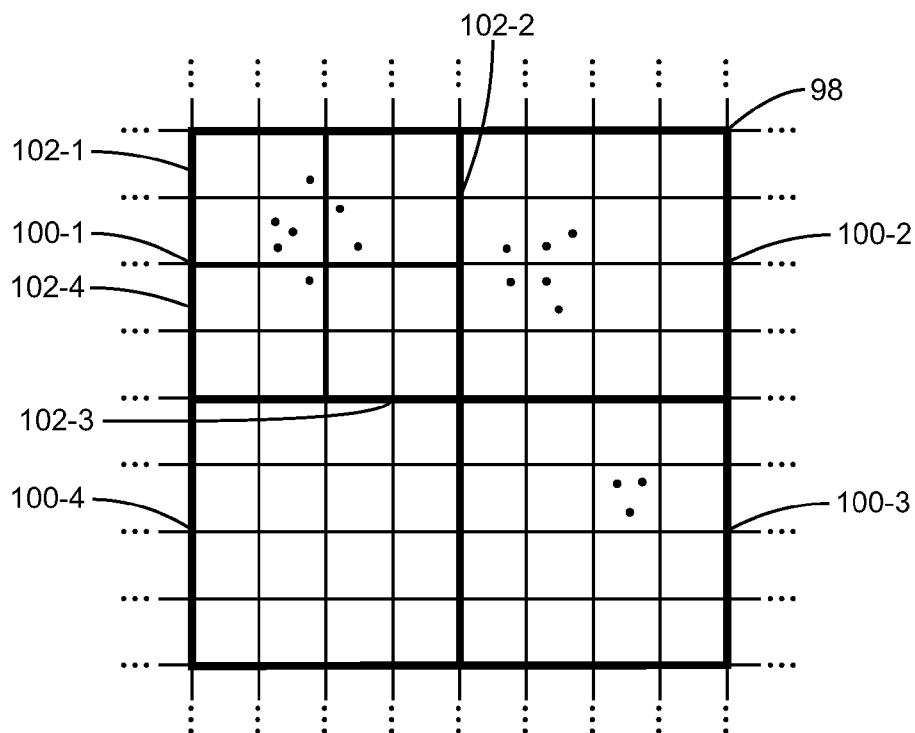


FIG. 17C

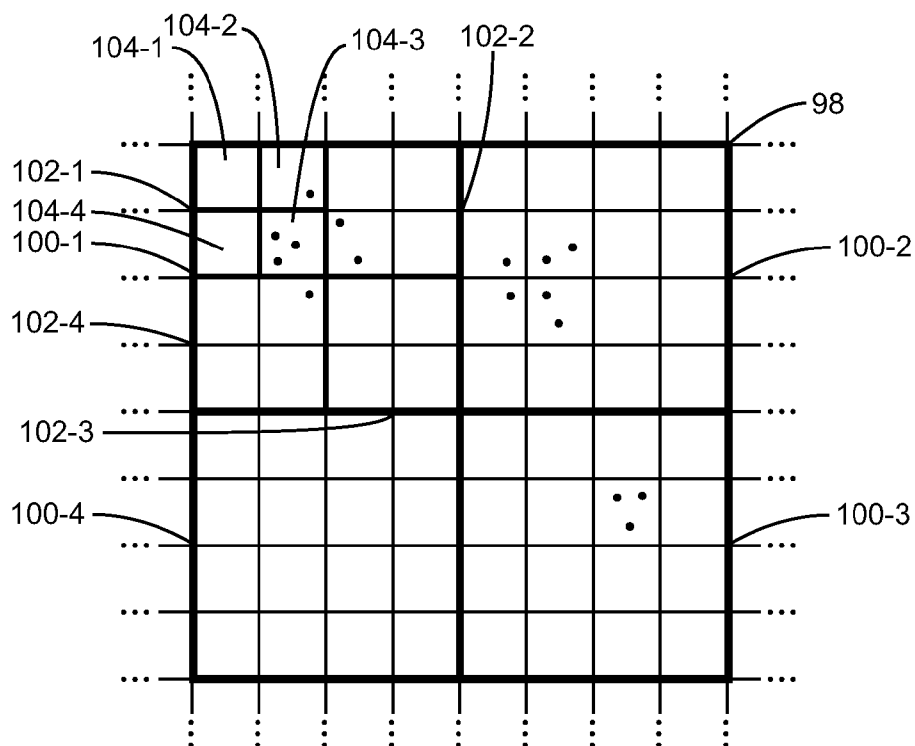


FIG. 17D

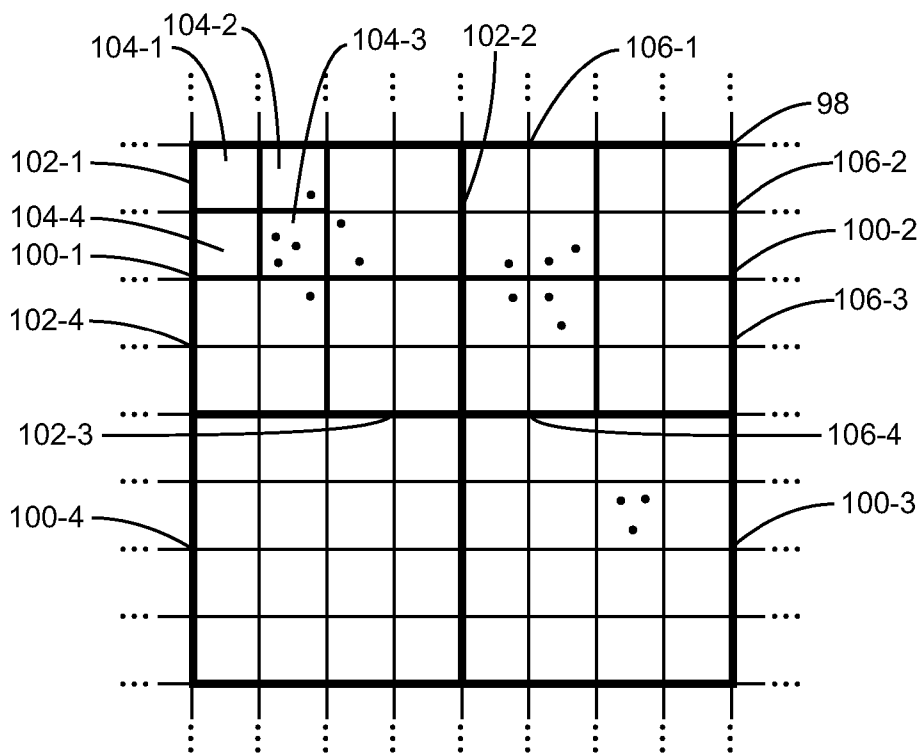


FIG. 17E

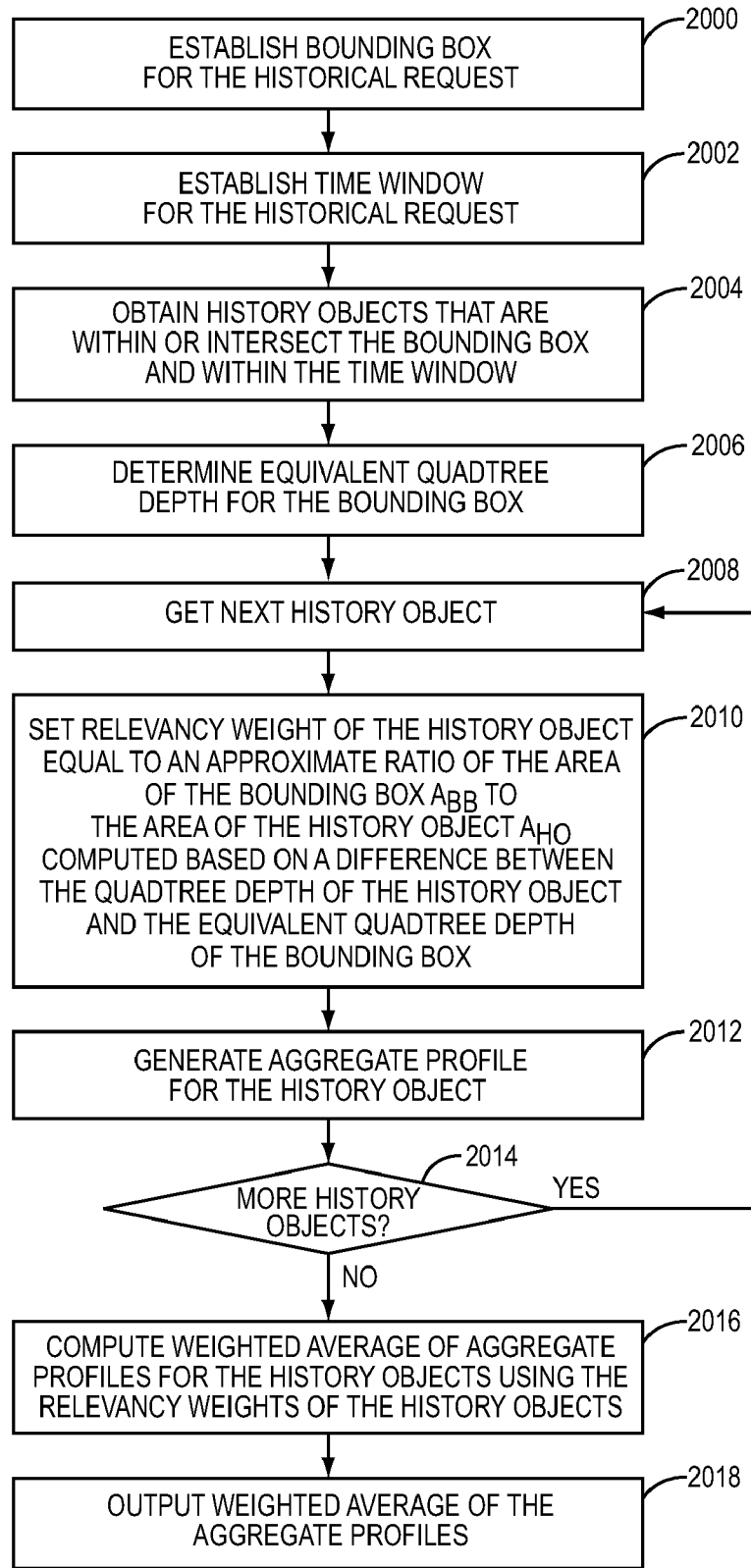


FIG. 18

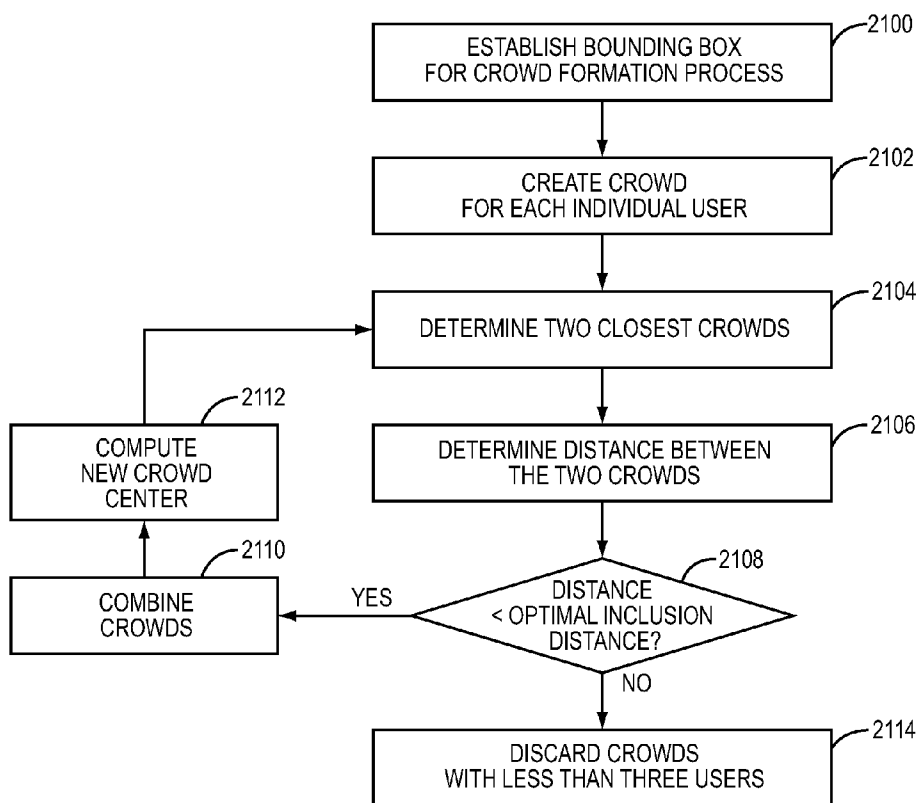
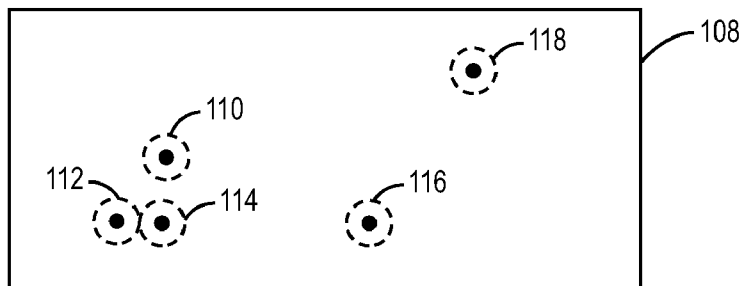
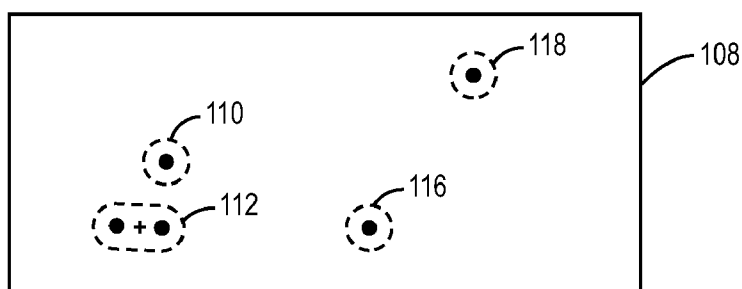


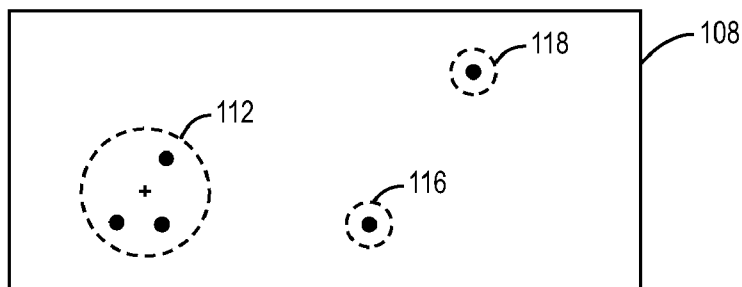
FIG. 19



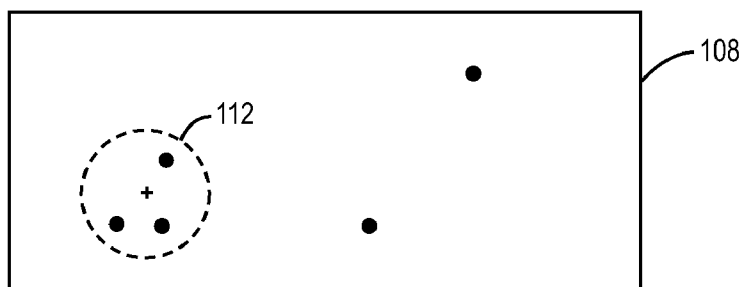
**FIG. 20A**



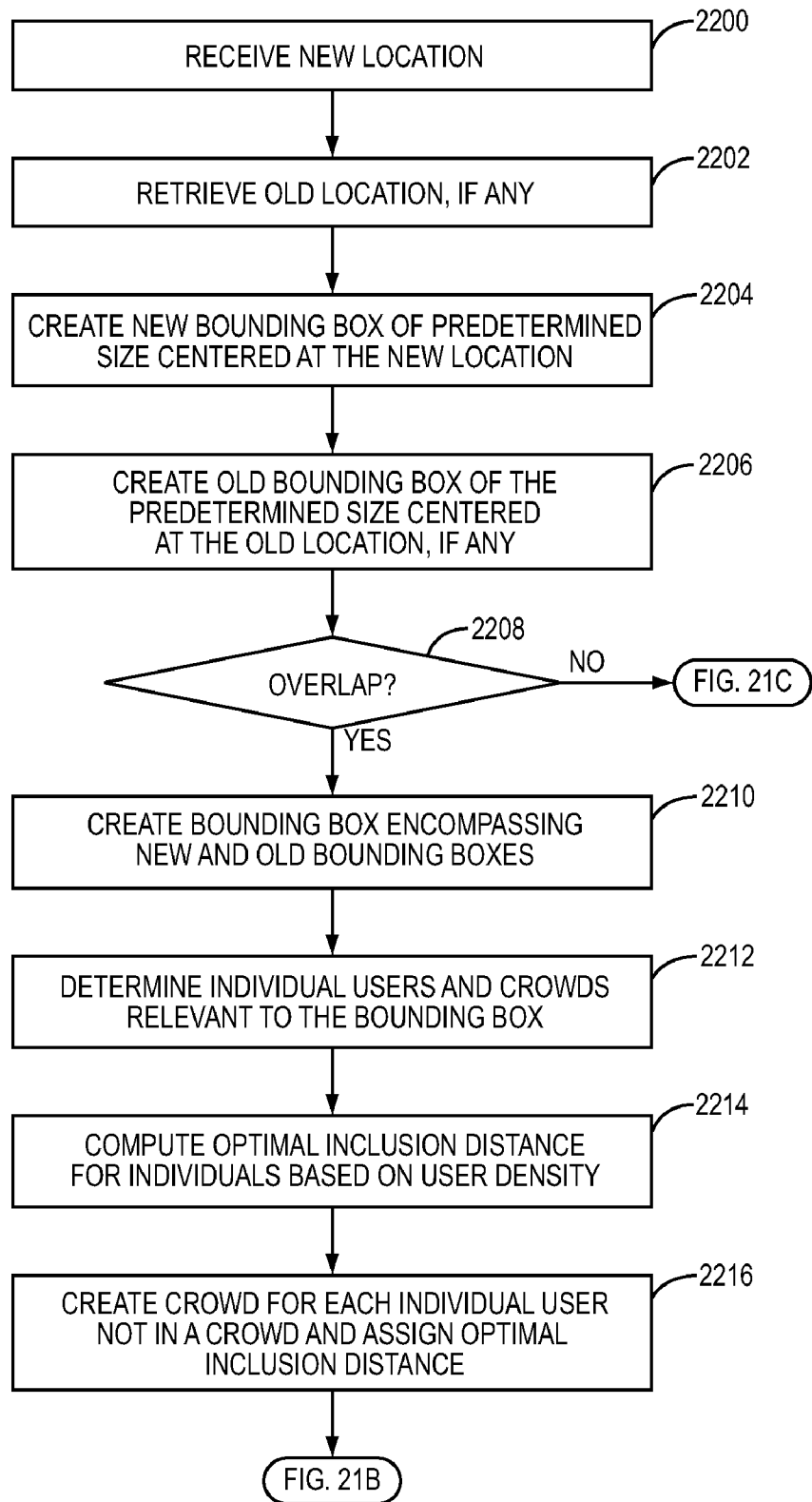
**FIG. 20B**



**FIG. 20C**



**FIG. 20D**



**FIG. 21A**

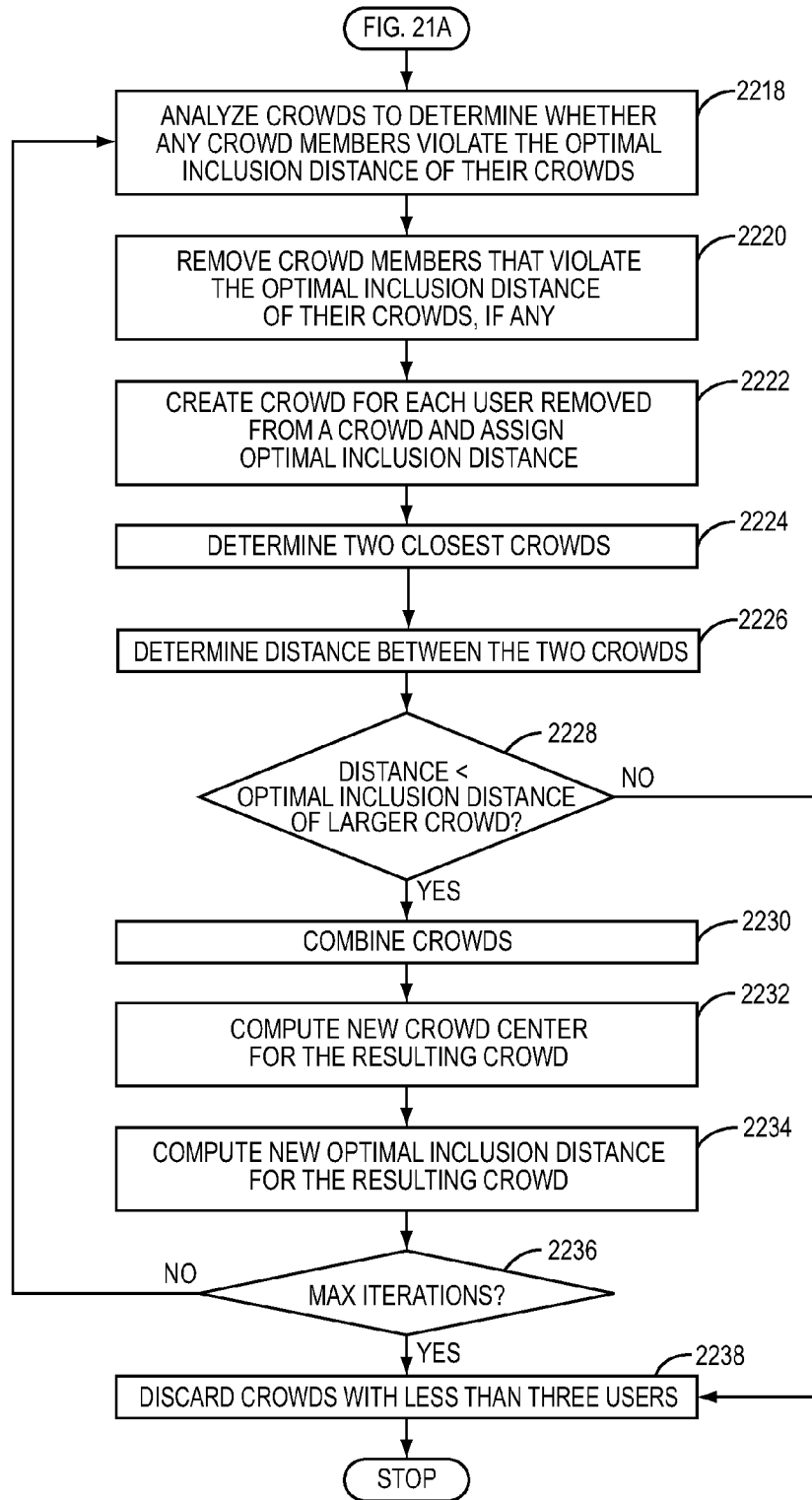


FIG. 21B



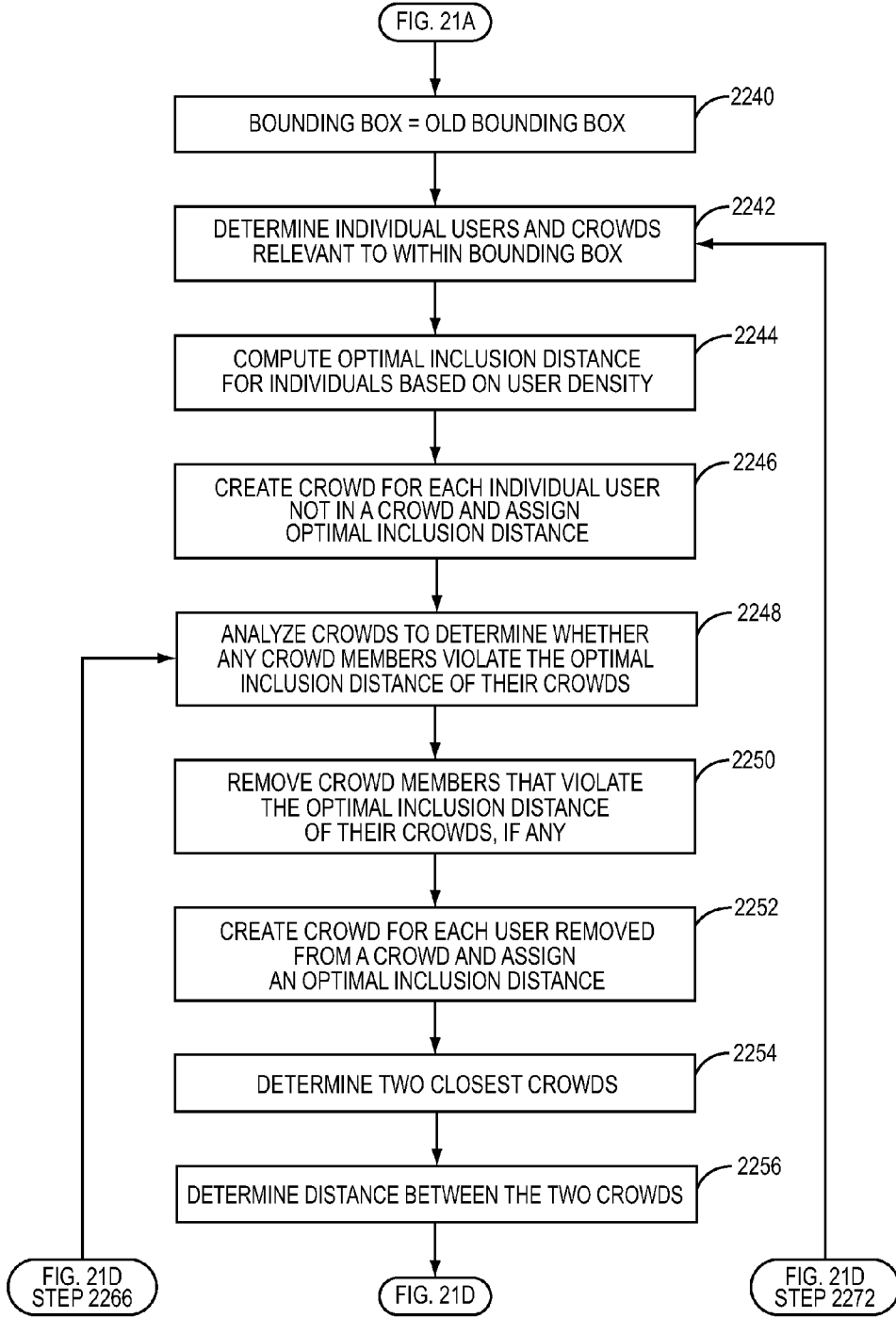


FIG. 21C

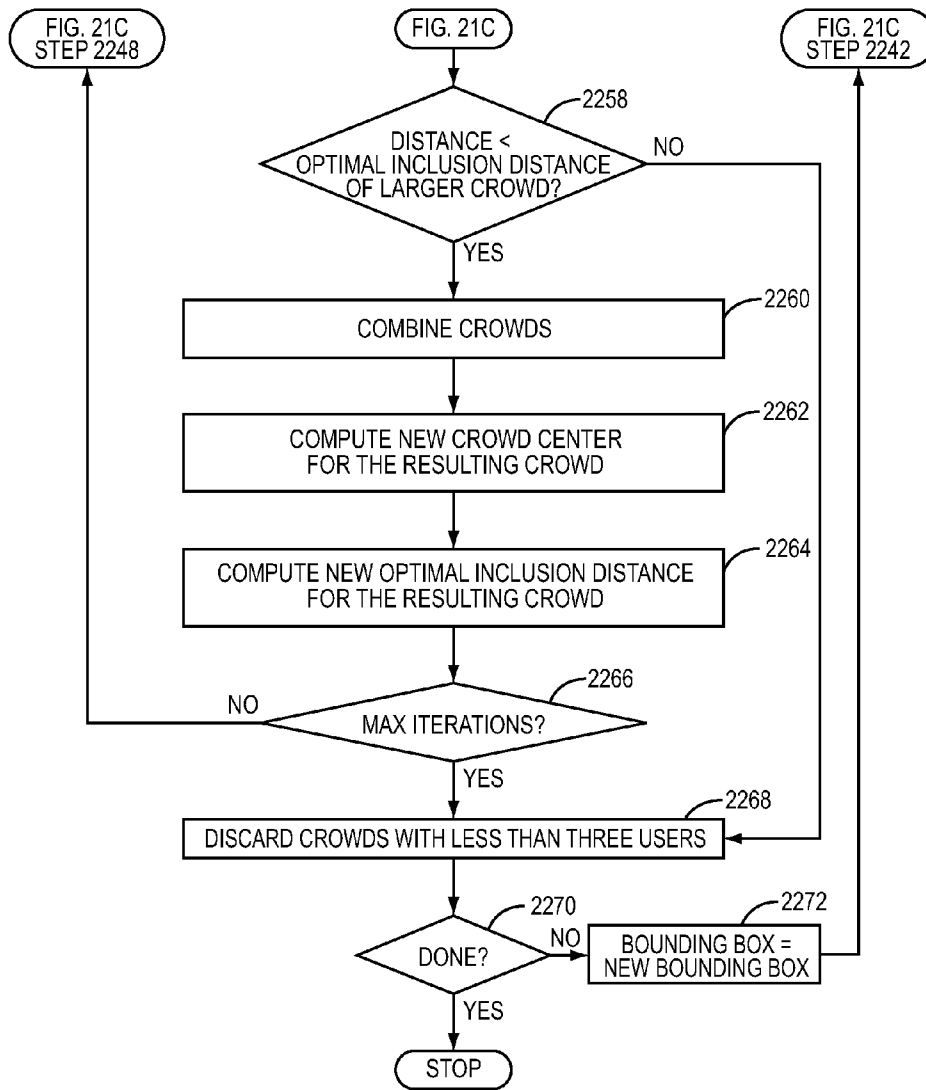
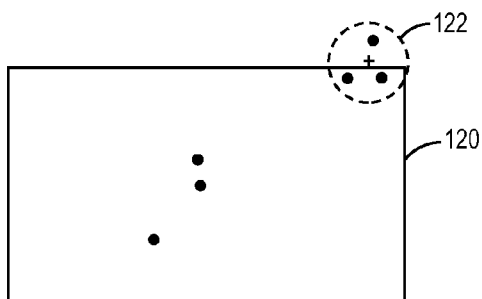
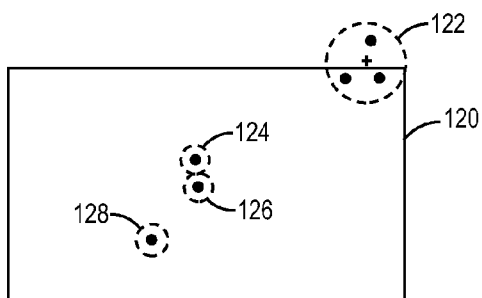


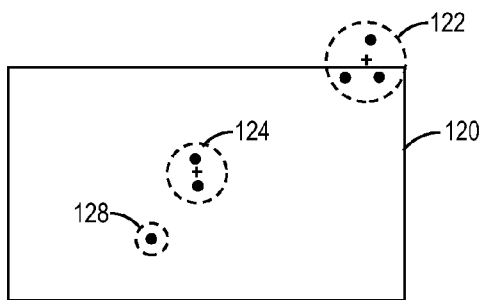
FIG. 21D



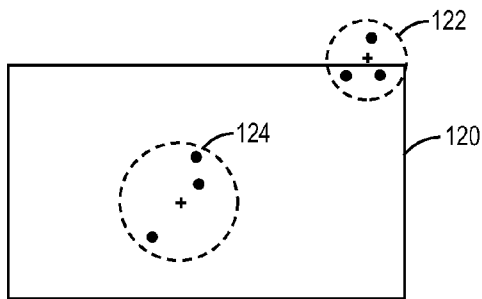
**FIG. 22A**



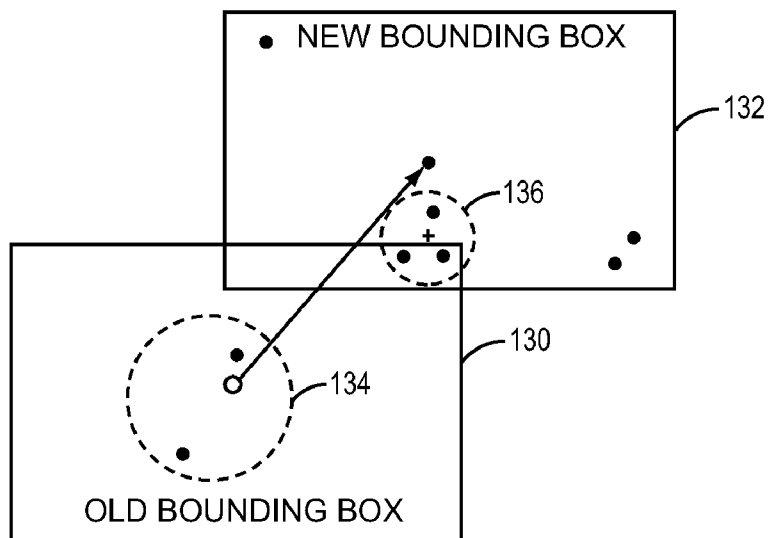
**FIG. 22B**



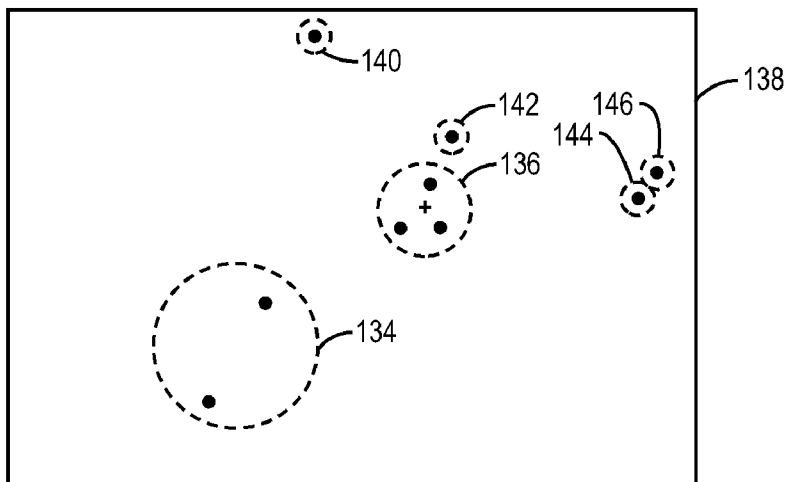
**FIG. 22C**



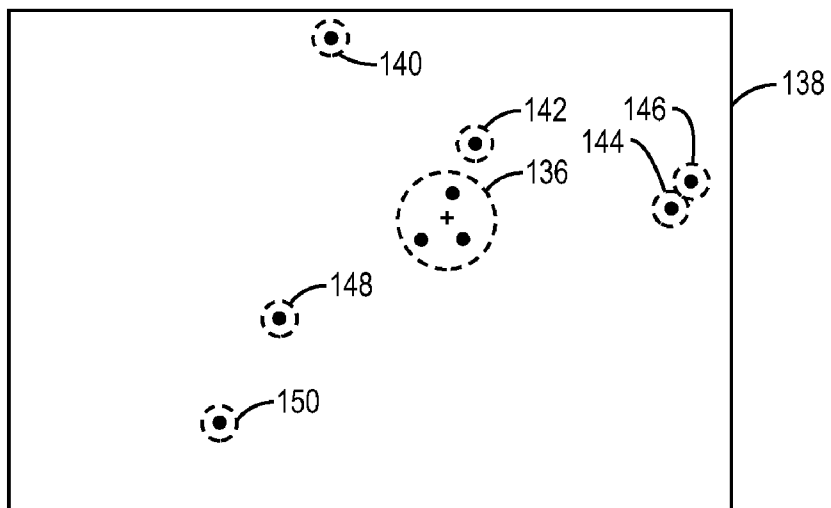
**FIG. 22D**



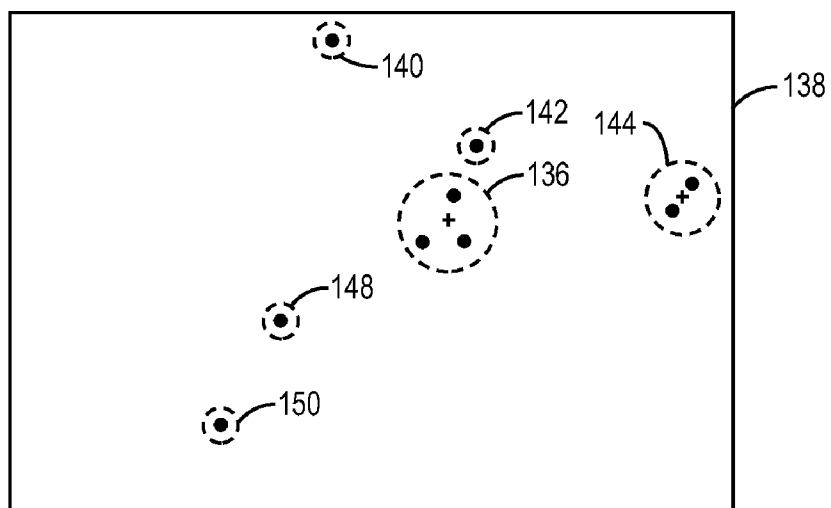
**FIG. 23A**



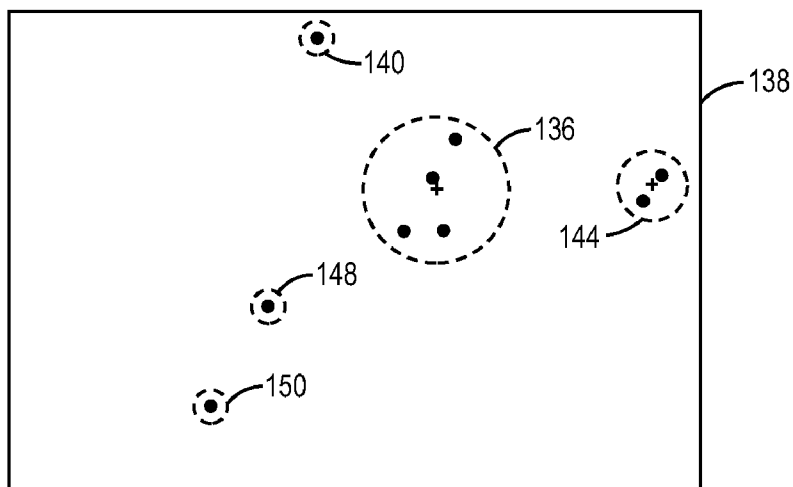
**FIG. 23B**



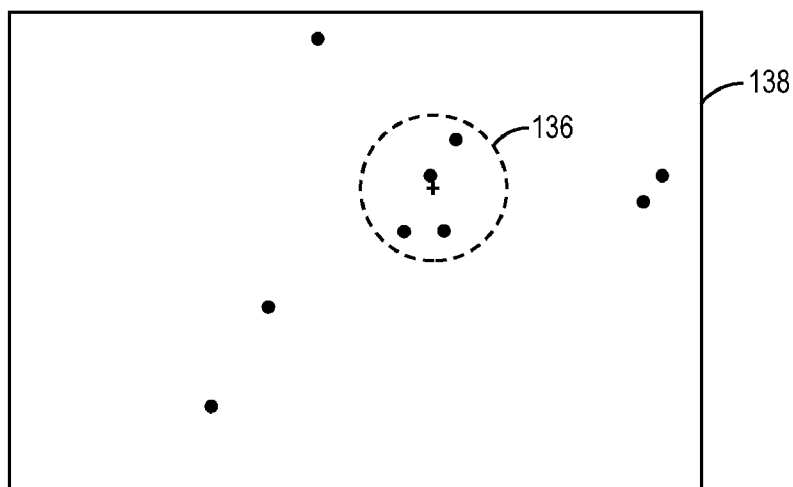
**FIG. 23C**



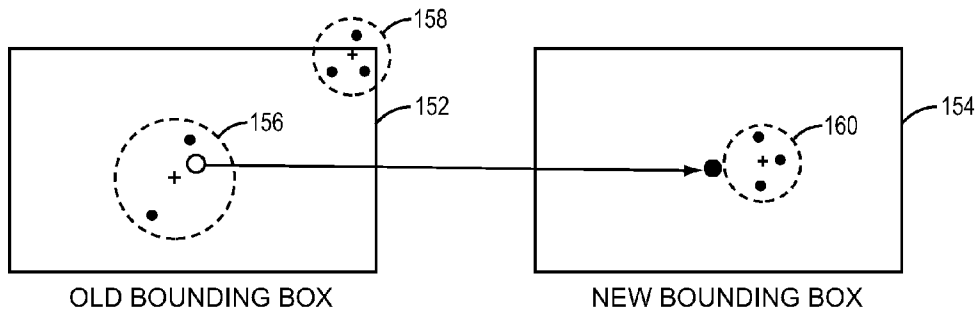
**FIG. 23D**



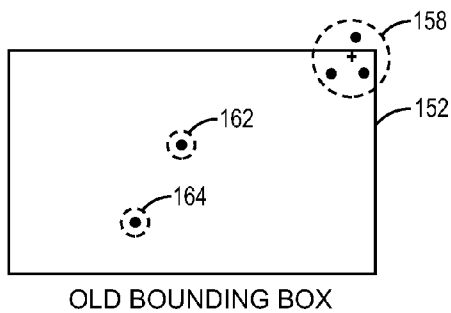
**FIG. 23E**



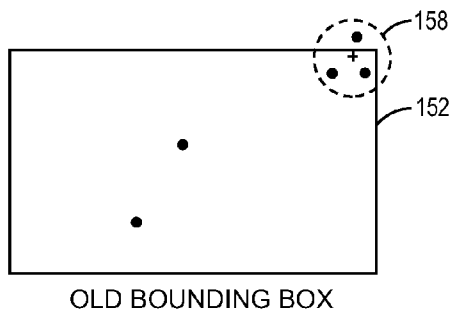
**FIG. 23F**



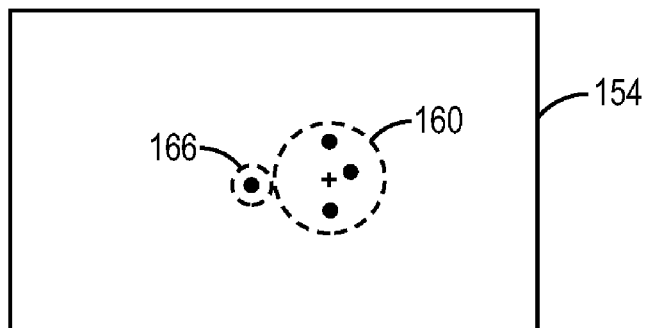
**FIG. 24A**



**FIG. 24B**

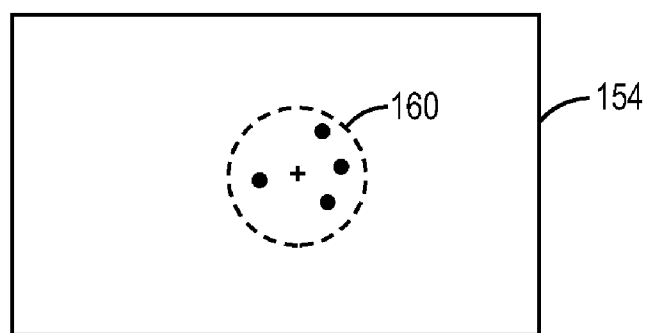


**FIG. 24C**



NEW BOUNDING BOX

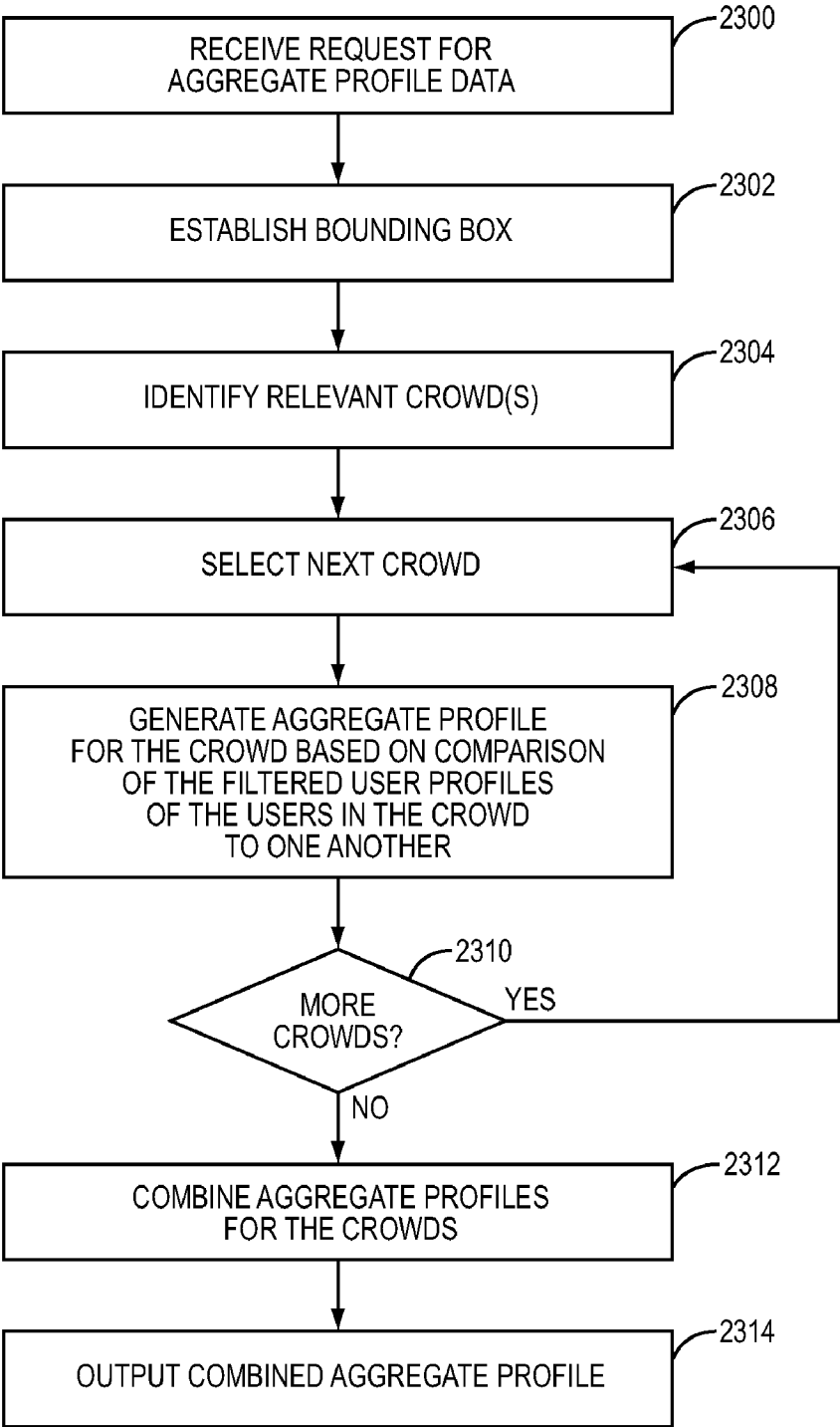
**FIG. 24D**



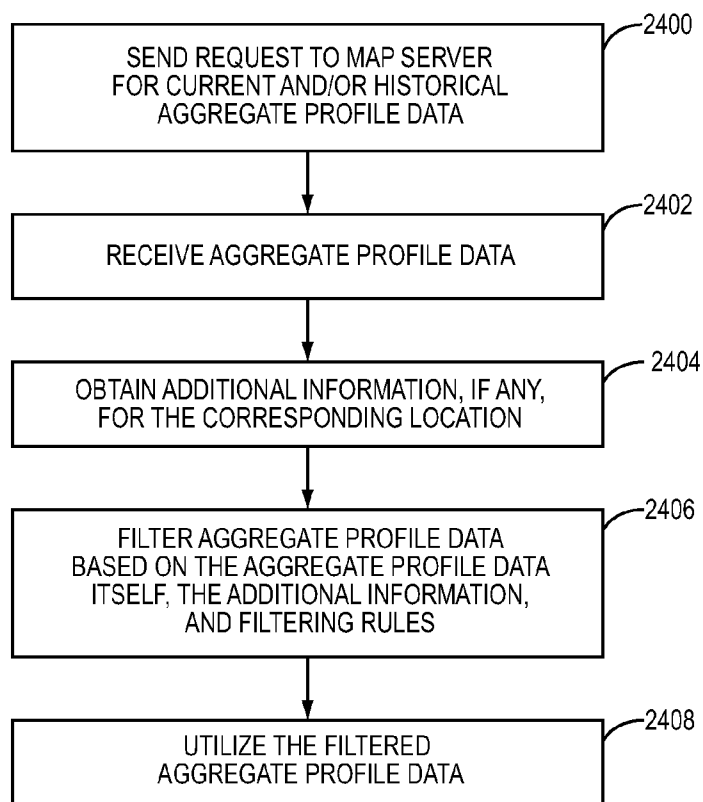
NEW BOUNDING BOX

**FIG. 24E**





**FIG. 25**



**FIG. 26**

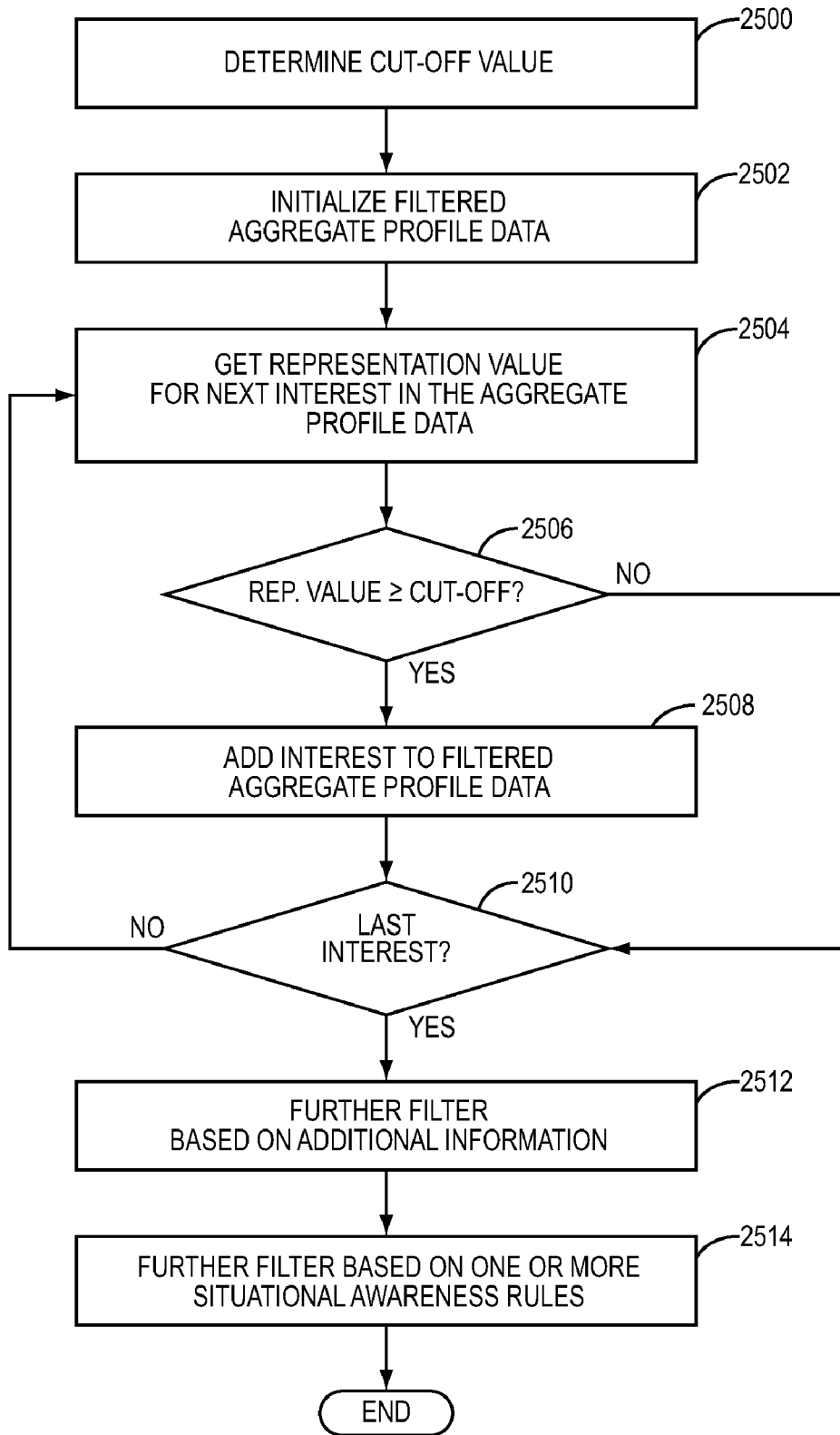


FIG. 27

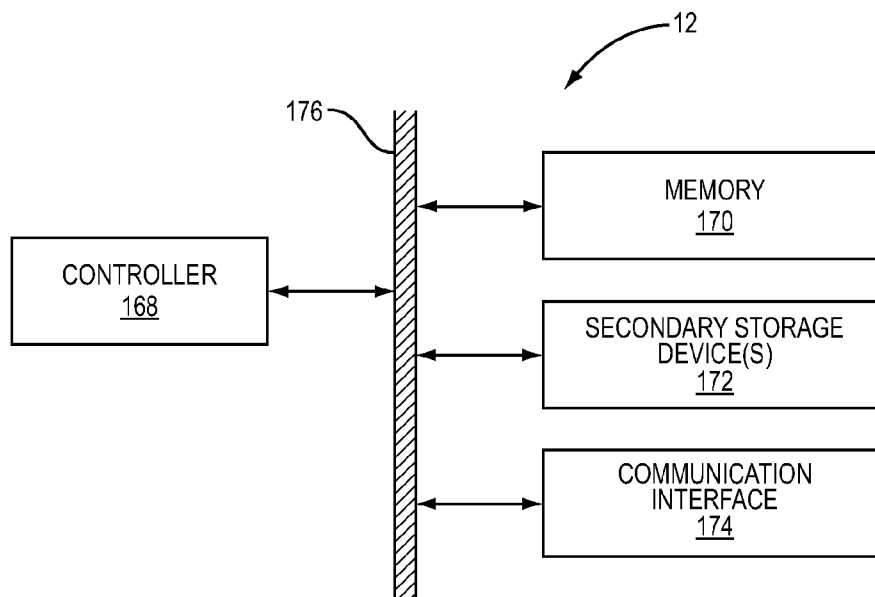


FIG. 28

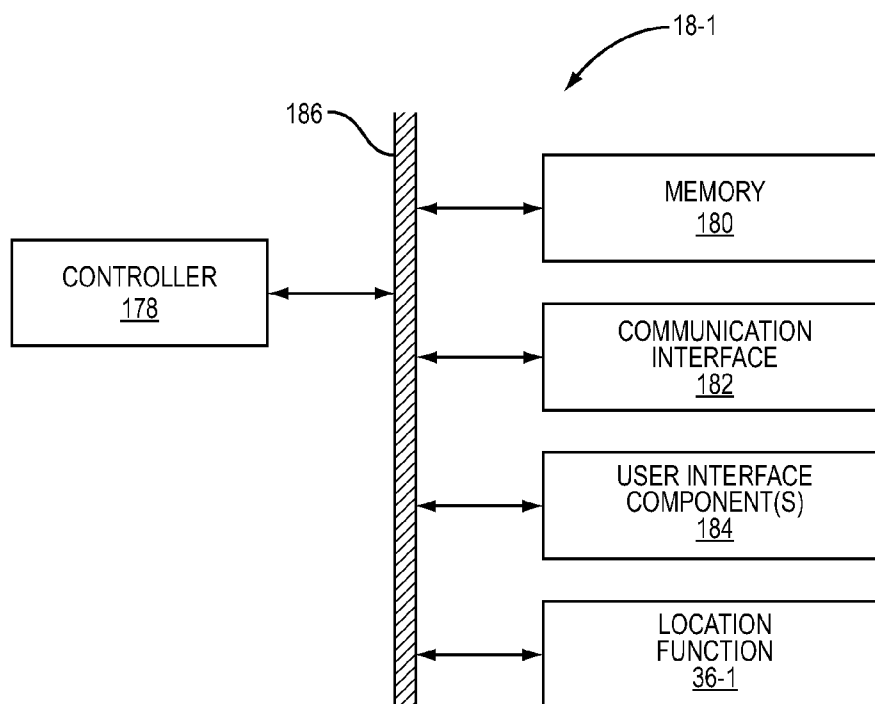


FIG. 29

**SYSTEM AND METHOD FOR PROFILE TAILORING IN AN AGGREGATE PROFILING SYSTEM**

**RELATED APPLICATIONS**

**[0001]** This application claims the benefit of provisional patent application Ser. No. 61/173,625, filed Apr. 29, 2009, the disclosure of which is hereby incorporated herein by reference in its entirety.

**FIELD OF THE DISCLOSURE**

**[0002]** The present disclosure relates to an aggregate profiling system and more particularly relates to tailoring a user profile of a user in an aggregate profiling system.

**BACKGROUND**

**[0003]** Mobile aggregate profiling systems are becoming increasingly popular. One such system is described in commonly owned and assigned U.S. patent application Ser. No. 12/645,535 entitled MAINTAINING A HISTORICAL RECORD OF ANONYMIZED USER PROFILE DATA BY LOCATION FOR USERS IN A MOBILE ENVIRONMENT, U.S. patent application Ser. No. 12/645,532 entitled FORMING CROWDS AND PROVIDING ACCESS TO CROWD DATA IN A MOBILE ENVIRONMENT, U.S. patent application Ser. No. 12/645,539 entitled ANONYMOUS CROWD TRACKING, U.S. patent application Ser. No. 12/645,544 entitled MODIFYING A USER'S CONTRIBUTION TO AN AGGREGATE PROFILE BASED ON TIME BETWEEN LOCATION UPDATES AND EXTERNAL EVENTS, U.S. patent application Ser. No. 12/645,546 entitled CROWD FORMATION FOR MOBILE DEVICE USERS, U.S. patent application Ser. No. 12/645,556 entitled SERVING A REQUEST FOR DATA FROM A HISTORICAL RECORD OF ANONYMIZED USER PROFILE DATA IN A MOBILE ENVIRONMENT, and U.S. patent application Ser. No. 12/645,560 entitled HANDLING CROWD REQUESTS FOR LARGE GEOGRAPHIC AREAS, all of which were filed on Dec. 23, 2009 and are hereby incorporated herein by reference in their entireties.

**[0004]** One issue with such systems is that a user may desire to share different portions of his user profile depending on the environment in which he is located. For instance, the user may desire to share one portion of his user profile when in an environment with business associates and share another portion of his user profile when in an environment with other users sharing similar personal interests. As such, there is a need for systems and methods that tailor a user's profile in an aggregate profiling system based on the environment in which the user is located.

**SUMMARY**

**[0005]** Systems and methods are disclosed for tailoring user profiles of users in a mobile aggregate profiling system. In one embodiment, the user profiles of the users in the mobile aggregate profiling system are tailored based on aggregate profile data relevant to current locations of the users. More specifically, in order to tailor the user profile of a user, aggregate profile data is obtained for the current location of the user. The user profile of the user is then filtered based on the aggregate profile data for the current location of the user. In one embodiment, the user profile is filtered at a server of the mobile aggregate profiling system. In another embodiment,

the user profile is filtered at a mobile device of the user prior to sending the user profile to a server of the mobile aggregate profiling system.

**[0006]** In one embodiment, the aggregate profile data includes an aggregate list of interests from user profiles of users historically located at or near the current location of the user and/or user profiles of users currently located at or near the current location of the user. In addition, for each interest in the aggregate list of interests, the aggregate profile data includes a representation value indicative of a degree to which that interest is included in the user profiles contributing to the aggregate profile data. The user profile of the user is filtered to remove interests from the user profile that correspond to interests in the aggregate profile data having representation values that are less than a cut-off value. In one embodiment, the cut-off value is a static cut-off value. In another embodiment, the cut-off value is a dynamic value. More specifically, in one embodiment, the cut-off value is a function of an amount of time that the user has been located at the user's current location, a number of times or a frequency at which the user has visited the user's current location, or a combination thereof.

**[0007]** Those skilled in the art will appreciate the scope of the present invention and realize additional aspects thereof after reading the following detailed description of the preferred embodiments in association with the accompanying drawing figures.

**BRIEF DESCRIPTION OF THE DRAWING FIGURES**

**[0008]** The accompanying drawing figures incorporated in and forming a part of this specification illustrate several aspects of the invention, and together with the description serve to explain the principles of the invention.

**[0009]** FIG. 1 illustrates a Mobile Aggregate Profile (MAP) system according to one embodiment of the present disclosure;

**[0010]** FIG. 2 is a block diagram of the MAP server of FIG. 1 according to one embodiment of the present disclosure;

**[0011]** FIG. 3 is a block diagram of the MAP client of one of the mobile devices of FIG. 1 according to one embodiment of the present disclosure;

**[0012]** FIG. 4 illustrates a process for filtering a user profile of a user at the MAP server of FIG. 1 according to one embodiment of the present disclosure;

**[0013]** FIG. 5 illustrates the operation of MAP system of FIG. 1 to obtain user profiles of the users at the MAP server according to one embodiment of the present disclosure;

**[0014]** FIG. 6 illustrates the operation of MAP system of FIG. 1 to obtain user profiles of the users at the MAP server according to another embodiment of the present disclosure;

**[0015]** FIG. 7 illustrates the operation of MAP system of FIG. 1 to obtain location updates for the users according to one embodiment of the present disclosure;

**[0016]** FIG. 8 illustrates a process for filtering a user profile of a user based on aggregate profile data for a current location of the user according to one embodiment of the present disclosure;

**[0017]** FIG. 9 illustrates a process for filtering a user profile of a user at the mobile device of the user according to another embodiment of the present disclosure;

**[0018]** FIGS. 10 and 11 graphically illustrate bucketization of users according to location for purposes of maintaining a

historical record of anonymized user profile data by location according to one embodiment of the present disclosure;

[0019] FIG. 12 is a flow chart illustrating the operation of a foreground bucketization process performed by the MAP server to maintain the lists of users for location buckets for purposes of maintaining a historical record of anonymized user profile data by location according to one embodiment of the present disclosure;

[0020] FIG. 13 is a flow chart illustrating the anonymization and storage process performed by the MAP server for the location buckets in order to maintain a historical record of anonymized user profile data by location according to one embodiment of the present disclosure;

[0021] FIG. 14 graphically illustrates anonymization of a user record according to one embodiment of the present disclosure;

[0022] FIG. 15 is a flow chart for a quadtree based storage process that may be used to store anonymized user profile data for location buckets according to one embodiment of the present disclosure;

[0023] FIG. 16 is a flow chart illustrating a quadtree algorithm that may be used to process the location buckets for storage of the anonymized user profile data according to one embodiment of the present disclosure;

[0024] FIGS. 17A through 17E graphically illustrate the process of FIG. 16 for the generation of a quadtree data structure for one exemplary base quadtree region;

[0025] FIG. 18 illustrates a process for generating historical aggregate profile data for utilization in filtering a user profile of a user according to the processes of FIGS. 4 and 9 according to one embodiment of the present disclosure;

[0026] FIG. 19 is a flow chart for a spatial crowd formation process according to one embodiment of the present disclosure;

[0027] FIGS. 20A through 20D graphically illustrate the crowd formation process of FIG. 19 for an exemplary bounding box;

[0028] FIGS. 21A through 21D illustrate a flow chart for a spatial crowd formation process according to another embodiment of the present disclosure;

[0029] FIGS. 22A through 22D graphically illustrate the crowd formation process of FIGS. 21A through 21D for a scenario where the crowd formation process is triggered by a location update for a user having no old location;

[0030] FIGS. 23A through 23F graphically illustrate the crowd formation process of FIGS. 21A through 21D for a scenario where the new and old bounding boxes overlap;

[0031] FIGS. 24A through 24E graphically illustrate the crowd formation process of FIGS. 21A through 21D in a scenario where the new and old bounding boxes do not overlap;

[0032] FIG. 25 illustrates a process for generating current aggregate profile data for utilization in filtering a user profile of a user according to the processes of FIGS. 4 and 9 according to one embodiment of the present disclosure;

[0033] FIG. 26 illustrates a process for filtering aggregate profile data received from the MAP server at a mobile device according to another embodiment of the present disclosure;

[0034] FIG. 27 illustrates a process for filtering aggregate profile data based on the aggregate profile data itself according to one embodiment of the present disclosure;

[0035] FIG. 28 is a block diagram of the MAP server of FIG. 1 according to one embodiment of the present disclosure; and

[0036] FIG. 29 is a block diagram of one of the mobile devices of FIG. 1 according to one embodiment of the present disclosure.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0037] The embodiments set forth below represent the necessary information to enable those skilled in the art to practice the embodiments and illustrate the best mode of practicing the embodiments. Upon reading the following description in light of the accompanying drawing figures, those skilled in the art will understand the concepts of the disclosure and will recognize applications of these concepts not particularly addressed herein. It should be understood that these concepts and applications fall within the scope of the disclosure and the accompanying claims.

[0038] FIG. 1 illustrates a Mobile Aggregate Profile (MAP) system 10 (hereinafter "system 10") according to one embodiment of the present disclosure. In this embodiment, the system 10 includes a MAP server 12, one or more profile servers 14, a location server 16, a number of mobile devices 18-1 through 18-N having associated users 20-1 through 20-N, a subscriber device 22 having an associated subscriber 24, and a third-party service 26 communicatively coupled via a network 28. The network 28 may be any type of network or any combination of networks. Specifically, the network 28 may include wired components, wireless components, or both wired and wireless components. In one exemplary embodiment, the network 28 is a distributed public network such as the Internet, where the mobile devices 18-1 through 18-N are enabled to connect to the network 28 via local wireless connections (e.g., WiFi or IEEE 802.11 connections) or wireless telecommunications connections (e.g., 3G or 4G telecommunications connections such as GSM, LTE, W-CDMA, or WiMAX connections).

[0039] As discussed below in detail, the MAP server 12 operates to obtain current locations, including location updates, and user profiles of the users 20-1 through 20-N of the mobile devices 18-1 through 18-N. The current locations of the users 20-1 through 20-N can be expressed as positional geographic coordinates such as latitude-longitude pairs, and a height vector (if applicable), or any other similar information capable of identifying a given physical point in space in a two-dimensional or three-dimensional coordinate system. Using the current locations and user profiles of the users 20-1 through 20-N, the MAP server 12 is enabled to provide a number of features such as, but not limited to, maintaining a historical record of anonymized user profile data by location, generating aggregate profile data over time for a Point of Interest (POI) or Area of Interest (AOI) using the historical record of anonymized user profile data, identifying crowds of users using current locations and/or user profiles of the users 20-1 through 20-N, generating aggregate profiles for crowds of users at a POI or in an AOI using the user profiles of users in the crowds, and crowd tracking. While not essential, for additional information regarding the MAP server 12, the interested reader is directed to U.S. patent application Ser. No. 12/645,535 entitled MAINTAINING A HISTORICAL RECORD OF ANONYMIZED USER PROFILE DATA BY LOCATION FOR USERS IN A MOBILE ENVIRONMENT, U.S. patent application Ser. No. 12/645,532 entitled FORMING CROWDS AND PROVIDING ACCESS TO CROWD DATA IN A MOBILE ENVIRONMENT, U.S. patent application Ser. No. 12/645,539 entitled ANONYMOUS CROWD

TRACKING, U.S. patent application Ser. No. 12/645,544 entitled MODIFYING A USER'S CONTRIBUTION TO AN AGGREGATE PROFILE BASED ON TIME BETWEEN LOCATION UPDATES AND EXTERNAL EVENTS, U.S. patent application Ser. No. 12/645,546 entitled CROWD FORMATION FOR MOBILE DEVICE USERS, U.S. patent application Ser. No. 12/645,556 entitled SERVING A REQUEST FOR DATA FROM A HISTORICAL RECORD OF ANONYMIZED USER PROFILE DATA IN A MOBILE ENVIRONMENT, and U.S. patent application Ser. No. 12/645,560 entitled HANDLING CROWD REQUESTS FOR LARGE GEOGRAPHIC AREAS, all of which were filed on Dec. 23, 2009 and have been incorporated herein by reference in their entireties. Note that while the MAP server 12 is illustrated as a single server for simplicity and ease of discussion, it should be appreciated that the MAP server 12 may be implemented as a single physical server or multiple physical servers operating in a collaborative manner for purposes of redundancy and/or load sharing.

[0040] In general, in one embodiment, the one or more profile servers 14 operate to store user profiles for a number of persons including the users 20-1 through 20-N of the mobile devices 18-1 through 18-N. For example, the one or more profile servers 14 may be servers providing social network services such as the Facebook® social networking service, the MySpace® social networking service, the LinkedIn® social networking service, or the like. As discussed below, in one embodiment, the MAP server 12 is enabled to directly or indirectly obtain the user profiles of the users 20-1 through 20-N of the mobile devices 18-1 through 18-N, or filtered versions of the user profiles, using the one or more profile servers 14. The location server 16 generally operates to receive location updates from the mobile devices 18-1 through 18-N and make the location updates available to entities such as, for instance, the MAP server 12. In one exemplary embodiment, the location server 16 is a server operating to provide Yahoo!'s FireEagle service.

[0041] The mobile devices 18-1 through 18-N may be mobile smart phones, portable media player devices, mobile gaming devices, or the like. Some exemplary mobile devices that may be programmed or otherwise configured to operate as the mobile devices 18-1 through 18-N are the Apple® iPhone®, the Palm® Pre™, the Samsung Rogue™, the BlackBerry® Storm™, and the Apple® iPod Touch® device. However, this list of exemplary mobile devices is not exhaustive and is not intended to limit the scope of the present disclosure.

[0042] The mobile devices 18-1 through 18-N include MAP clients 30-1 through 30-N, MAP applications 32-1 through 32-N, third-party applications 34-1 through 34-N, and location functions 36-1 through 36-N, respectively. Using the mobile device 18-1 as an example, the MAP client 30-1 is preferably implemented in software. In general, in the preferred embodiment, the MAP client 30-1 is a middleware layer operating to interface an application layer (i.e., the MAP application 32-1 and the third-party applications 34-1) to the MAP server 12. More specifically, the MAP client 30-1 enables the MAP application 32-1 and the third-party applications 34-1 to request and receive data from the MAP server 12. In addition, the MAP client 30-1 enables applications, such as the MAP application 32-1 and the third-party applications 34-1, to access data from the MAP server 12. Note that while the MAP client 30-1 is implemented separately from the MAP application 32-1 and the third-party applica-

tions 34-1 in this embodiment, one of ordinary skill in the art will appreciate that the MAP client 30-1, or functions of the MAP client 30-1, may alternatively be implemented in the MAP application 32-1 and the third-party applications 34-1.

[0043] The MAP application 32-1 is also preferably implemented in software. The MAP application 32-1 generally provides a user interface component between the user 20-1 and the MAP server 12. More specifically, among other things, the MAP application 32-1 may enable the user 20-1 to initiate historical requests for historical data or crowd requests for crowd data (e.g., aggregate profile data and/or crowd characteristics data) from the MAP server 12 for a POI or AOI. The MAP application 32-1 may also enable the user 20-1 to configure various settings. For example, the MAP application 32-1 may enable the user 20-1 to select a desired social networking service (e.g., Facebook®, MySpace®, LinkedIn®, etc.) from which to obtain the user profile of the user 20-1 and provide any necessary credentials (e.g., username and password) needed to access the user profile from the social networking service.

[0044] The third-party applications 34-1 are preferably implemented in software. The third-party applications 34-1 operate to access the MAP server 12 via the MAP client 30-1. The third-party applications 34-1 may utilize data obtained from the MAP server 12 in any desired manner. As an example, one of the third-party applications 34-1 may be a gaming application that utilizes historical aggregate profile data to notify the user 20-1 of POIs or AOIs where persons having an interest in the game have historically congregated.

[0045] The location function 36-1 may be implemented in hardware, software, or a combination thereof. In general, the location function 36-1 operates to determine or otherwise obtain the location of the mobile device 18-1. For example, the location function 36-1 may be or include a Global Positioning System (GPS) receiver.

[0046] The subscriber device 22 is a physical device such as a personal computer, a mobile computer (e.g., a notebook computer, a netbook computer, a tablet computer, etc.), a mobile smart phone, or the like. The subscriber 24 associated with the subscriber device 22 is a person or entity. In general, the subscriber device 22 enables the subscriber 24 to access the MAP server 12 via a web browser 38 to obtain various types of data, preferably for a fee. For example, the subscriber 24 may pay a fee to have access to historical aggregate profile data for one or more POIs and/or one or more AOIs, pay a fee to have access to crowd data such as aggregate profiles for crowds located at one or more POIs and/or located in one or more AOIs, pay a fee to track crowds, or the like. Note that the web browser 38 is exemplary. In another embodiment, the subscriber device 22 is enabled to access the MAP server 12 via a custom application.

[0047] Lastly, the third-party service 26 is a service that has access to data from the MAP server 12 such as a historical aggregate profile data for one or more POIs or one or more AOIs, crowd data such as aggregate profiles for one or more crowds at one or more POIs or within one or more AOIs, or crowd tracking data. Based on the data from the MAP server 12, the third-party service 26 operates to provide a service such as, for example, targeted advertising. For example, the third-party service 26 may obtain anonymous aggregate profile data for one or more crowds located at a POI and then provide targeted advertising to known users located at the POI based on the anonymous aggregate profile data. Note that while targeted advertising is mentioned as an exemplary

third-party service 26, other types of third-party services 26 may additionally or alternatively be provided. Other types of third-party services 26 that may be provided will be apparent to one of ordinary skill in the art upon reading this disclosure.

[0048] Before proceeding, it should be noted that while the system 10 of FIG. 1 illustrates an embodiment where the one or more profile servers 14 and the location server 16 are separate from the MAP server 12, the present disclosure is not limited thereto. In an alternative embodiment, the functionality of the one or more profile servers 14 and/or the location server 16 may be implemented within the MAP server 12.

[0049] FIG. 2 is a block diagram of the MAP server 12 of FIG. 1 according to one embodiment of the present disclosure. As illustrated, the MAP server 12 includes an application layer 40, a business logic layer 42, and a persistence layer 44. The application layer 40 includes a user web application 46, a mobile client/server protocol component 48, and one or more data Application Programming Interfaces (APIs) 50. The user web application 46 is preferably implemented in software and operates to provide a web interface for users, such as the subscriber 24, to access the MAP server 12 via a web browser. The mobile client/server protocol component 48 is preferably implemented in software and operates to provide an interface between the MAP server 12 and the MAP clients 30-1 through 30-N hosted by the mobile devices 18-1 through 18-N. The data APIs 50 enable third-party services, such as the third-party service 26, to access the MAP server 12.

[0050] The business logic layer 42 includes a profile manager 52, a location manager 54, a history manager 56, a crowd analyzer 58, and an aggregation engine 60, each of which is preferably implemented in software. In one embodiment, the profile manager 52 operates to obtain the user profiles of the users 20-1 through 20-N from the one or more profile servers 14, filter the user profiles of the users 20-1 through 20-N, and store the filtered user profiles and, optionally, the user profiles of the users 20-1 through 20-N in the persistence layer 44. In another embodiment, the profile manager 52 operates to obtain the user profiles of the users 20-1 through 20-N from the mobile devices 18-1 through 18-N of the users 20-1 through 20-N, filter the user profiles of the user 20-1 through 20-N, and store the filtered user profiles and, optionally, the user profiles of the users 20-1 through 20-N in the persistence layer 44. In yet another embodiment, the profile manager 52 operates to obtain filtered user profiles of the users 20-1 through 20-N from the mobile devices 18-1 through 18-N of the users 20-1 through 20-N and stored the filtered user profiles of the users 20-1 through 20-N in the persistence layer 44. In yet another embodiment, the profile manager 52 operates to obtain the user profiles of the users 20-1 through 20-N directly or indirectly from the one or more profile servers 14 or directly from the mobile devices 18-1 through 18-N of the users 20-1 through 20-N and store the user profiles of the users 20-1 through 20-N in the persistence layer 44.

[0051] The location manager 54 operates to obtain the current locations of the users 20-1 through 20-N including location updates. The current locations of the users 20-1 through 20-N may be obtained directly from the mobile devices 18-1 through 18-N and/or obtained from the location server 16. The history manager 56 generally operates to maintain a historical record of anonymized user profile data by location. The crowd analyzer 58 operates to form crowds of users. In one embodiment, the crowd analyzer 58 utilizes a spatial crowd formation algorithm. However, the present disclosure

is not limited thereto. In addition, the crowd analyzer 58 may also operate to track crowds. The aggregation engine 60 generally operates to provide aggregate profile data in response to requests from the mobile devices 18-1 through 18-N, the subscriber device 22, and the third-party service 26. The aggregate profile data may be historical aggregate profile data or current aggregate profile data.

[0052] The persistence layer 44 includes an object mapping layer 62 and a datastore 64. The object mapping layer 62 is preferably implemented in software. The datastore 64 is preferably a relational database, which is implemented in a combination of hardware (i.e., physical data storage hardware) and software (i.e., relational database software). In this embodiment, the business logic layer 42 is implemented in an object-oriented programming language such as, for example, Java. As such, the object mapping layer 62 operates to map objects used in the business logic layer 42 to relational database entities stored in the datastore 64. Note that, in one embodiment, data is stored in the datastore 64 in a Resource Description Framework (RDF) compatible format.

[0053] In an alternative embodiment, rather than being a relational database, the datastore 64 may be implemented as an RDF datastore. More specifically, the RDF datastore may be compatible with RDF technology adopted by Semantic Web activities. Namely, the RDF datastore may use the Friend-Of-A-Friend (FOAF) vocabulary for describing people, their social networks, and their interests. In this embodiment, the MAP server 12 may be designed to accept raw FOAF files describing persons, their friends, and their interests. These FOAF files are currently output by some social networking services such as LiveJournal and Facebook®. The MAP server 12 may then persist RDF descriptions of the users 20-1 through 20-N as a proprietary extension of the FOAF vocabulary that includes additional properties desired for the system 10.

[0054] FIG. 3 illustrates the MAP client 30-1 of FIG. 1 in more detail according to one embodiment of the present disclosure. This discussion is equally applicable to the other MAP clients 30-2 through 30-N. As illustrated, in this embodiment, the MAP client 30-1 includes a MAP access API 66, a MAP middleware component 68, and a mobile client/server protocol component 70. The MAP access API 66 is implemented in software and provides an interface by which the MAP client 30-1 and the third-party applications 34-1 are enabled to access the MAP client 30-1. The MAP middleware component 68 is implemented in software and performs the operations needed for the MAP client 30-1 to operate as an interface between the MAP application 32-1 and the third-party applications 34-1 at the mobile device 18-1 and the MAP server 12. The mobile client/server protocol component 70 enables communication between the MAP client 30-1 and the MAP server 12 via a defined protocol.

[0055] FIG. 4 illustrates a process performed by the MAP server 12 to tailor user profiles of one or more of the users 20-1 through 20-N according to one embodiment of the present disclosure. In general, using the user 20-1 as an example, the user 20-1 may desire to reveal different portions of his user profile depending on his environment. For example, the user 20-1 may desire to reveal interests in his user profile related to his work when he is in a work or business environment and reveal interests in his user profile related to his personal life when he is in an environment around other users that have similar personal interests. Using



the process of FIG. 4, the profile manager 52 operates to provide such tailoring of the user profile of the user 20-1.

[0056] First, the profile manager 52 of the MAP server 12 obtains a user profile of a user (step 1000). For this discussion, the user is the user 20-1. However, this discussion is equally applicable to the other users 20-2 through 20-N. As discussed below in detail, in one embodiment, the profile manager 52 obtains the user profile of the user 20-1 directly or indirectly from the one or more profile servers 14. In another embodiment, the user profile of the user 20-1 may be created and maintained at the mobile device 18-1, in which case the profile manager 52 obtains the user profile of the user 20-1 from the mobile device 18-1 of the user 20-1. As discussed below in detail, the user profile of the user 20-1 includes a number of interests of the user 20-1. The interests are preferably expressed as keywords.

[0057] Next, the location manager 54 receives a location update for the user 20-1 of the mobile device 18-1 (step 1002). As described below in detail, in one embodiment, the location manager 54 receives the location update for the user 20-1 directly from the mobile device 18-1 of the user 20-1. In another embodiment, the location manager 54 receives the location update for the user 20-1 from the location server 16. The location update provides the current location of the user 20-1. Note that, in one exemplary alternative embodiment, rather than obtaining the user profile separately in step 1000, both the user profile and the current location may be included in the update received in step 1002.

[0058] In this embodiment, after receiving the location update, the profile manager 52 obtains aggregate profile data for the current location of the user 20-1 (step 1004). The aggregate profile data includes current aggregate profile data for users currently located at or near the current location of the user 20-1, historical aggregate profile data for users previously located at or near the current location of the user 20-1, or both current aggregate profile data and historical aggregate profile data. The current aggregate profile data includes an aggregate list of interests from filtered user profiles of users currently located at or near the current location of the user 20-1. In one embodiment, the users currently located at or near the current location of the user 20-1 are users in one or more crowds of users located within a defined bounding region centered at or otherwise encompassing the current location of the user 20-1. In addition, for each interest in the aggregate list of interests, the aggregate profile data preferably includes a representation value defining a degree to which the interest is expressed in the filtered user profiles of the users currently located at or near the current location of the user 20-1. In one embodiment, the representation value for an interest is expressed as a number of occurrences or user matches of the interest in the filtered user profiles of the users located at or near the current location of the user 20-1. In another embodiment, the representation value for an interest is expressed as a ratio of a number of occurrences or user matches of the interest in the filtered user profiles of the users located at or near the current location of the user 20-1 over a total number of users. Note that if the number of users at or near the current location of the user 20-1 are users in multiple crowds of users, then the current aggregate profile data may include a separate aggregate profile (i.e., aggregate list of interests and, preferably, corresponding representation values) for each crowd of users or a combined aggregate profile for all of the crowds of users. The combined aggregate profile for all of the crowds of users may include, for example, an

aggregate list of interests for all of the crowds and, preferably, corresponding representation values for the interests across all of the crowds.

[0059] In one embodiment, the historical profile data includes one or more historical aggregate profiles. Each historical aggregate profile is an aggregate profile for users located at or near the current location of the user 20-1 during a corresponding time period that occurred in the past. More specifically, in one embodiment, the users located at or near the current location of the user 20-1 during the corresponding time period are users for which filtered user profiles are stored in history objects relevant to a bounding region encompassing the location of the user 20-1 during the corresponding time period, as discussed below. Further, each historical aggregate profile includes an aggregate list of interests from the filtered user profiles of the users located at or near the current location of the user 20-1 during the corresponding time period. In addition, each historical aggregate profile preferably includes corresponding representation values for the interests in the historical aggregate profile, where the representation values define degrees to which the interests are represented in the filtered user profiles of the users located at or near the current location of the user 20-1 during the corresponding time period. In one embodiment, a representation value for an interest is expressed as a number of occurrences or user matches of the interest in the filtered user profiles of the users located at or near the current location of the user 20-1 during the corresponding time period. In another embodiment, the representation value for an interest is expressed as a ratio of a number of occurrences or user matches of the interest in the filtered user profiles of the users located at or near the current location of the user 20-1 over a total number of users during the corresponding time period.

[0060] Note that for the discussion herein, it is assumed that that the user profiles of all of the users 20-1 through 20-N are filtered as described herein and, as such, the filtered user profiles are used to generate both the current and historical aggregate profile data. However, in some embodiments, the user profiles of some of the users 20-1 through 20-N may not be filtered, in which case the user profiles of those users are used to generate both current and historical aggregate profile data.

[0061] In this embodiment, the profile manager 52 also obtains additional information, if any, for the current location of the user 20-1 (step 1006). Note that step 1006 is optional. The additional information may be any type of information about the current location of the user 20-1 that may assist the profile manager 52 in determining what interests in the user profile of the user 20-1 should or should not contribute to aggregate profile data accessible from the MAP server 12. For example, if the current location of the user 20-1 indicates that the user 20-1 is located at a POI, the additional data may be data describing that POI. As a more specific example, if the current location of the user 20-1 indicates that the user 20-1 is at an arena at which different types of events are held (e.g., sporting events, music concerts, etc.), the profile manager 52 may obtain additional information such as, for example, information describing an event currently being held at the arena. The additional information may be obtained from an internal database or repository maintained by the MAP server 12, where this internal database or repository is manually populated by a community of users such as the users 20-1 through 20-N, employees of an owner of the MAP server 12, or the like. Alternatively, the internal database or repository

may be populated via an automated process where, for example, websites associated with a number of known POIs are analyzed to obtain the additional information for the POIs. In another embodiment, the MAP server 12 may obtain the additional information for the current location of the user 20-1 from one or more remote sources that operate to maintain such information.

[0062] Next, the profile manager 52 filters the user profile of the user 20-1 based on the aggregate profile data for the current location of the user 20-1, the additional information obtained for the current location of the user 20-1, and one or more defined filtering rules (step 1008). In general, the profile manager 52 filters the user profile of the user 20-1 based on the aggregate profile data and the additional information for the current location of the user 20-1 to remove any interests from the user profile of the user 20-1 that are not desired to be revealed in the current environment of the user 20-1. More specifically, as described below in detail, the one or more defined filtering rules are rules that enable the profile manager 52 to determine, for each interest in the user profile of the user 20-1, whether to filter that interest based on the aggregate profile data and/or the additional information for the current location of the user 20-1. The one or more defined filtering rules may include system-defined rules, rules defined by the user 20-1, or a combination thereof.

[0063] Lastly, the filtered user profile of the user 20-1 is stored in the datastore 64 of the MAP server 12 (step 1010). Thereafter, the filtered user profile of the user 20-1, rather than the user profile of the user 20-1, is utilized by the MAP server 12 to provide aggregate profile data. At this point, the process returns to step 1002 and is repeated for the next location update for the user 20-1. As such, as the current location of the user 20-1 changes and/or as the aggregate profile data for the current location of the user 20-1 changes, the filtered user profile of the user 20-1 is updated. Further, as described below, in some embodiments, more of the user profile of the user 20-1 may be revealed as the user 20-1 spends more time at the same location.

[0064] FIG. 5 illustrates step 1000 of FIG. 4 in more detail according to one exemplary embodiment of the present disclosure. More specifically, FIG. 5 illustrates a process by which user profiles of the users 20-1 through 20-N are obtained by the MAP server 12 according to one exemplary embodiment of the present disclosure. In this example, the user profile of the user 20-1 is obtained by the MAP server 12. However, this process may also be used to obtain the user profiles of the other users 20-2 through 20-N. First, an authentication process is performed (step 1100). For authentication, in this embodiment, the mobile device 18-1 authenticates with the profile server 14 (step 1100A) and the MAP server 12 (step 1100B). In addition, the MAP server 12 authenticates with the profile server 14 (step 1100C). Preferably, authentication is performed using OpenID or similar technology. However, authentication may alternatively be performed using separate credentials (e.g., username and password) of the user 20-1 for access to the MAP server 12 and the profile server 14. Assuming that authentication is successful, the profile server 14 returns an authentication succeeded message to the MAP server 12 (step 1100D), and the profile server 14 returns an authentication succeeded message to the MAP client 30-1 of the mobile device 18-1 (step 1100E).

[0065] At some point after authentication is complete, a user profile process is performed such that a user profile of the user 20-1 is obtained from the profile server 14 and delivered

to the MAP server 12 (step 1102). In this embodiment, the MAP client 30-1 of the mobile device 18-1 sends a profile request to the profile server 14 (step 1102A). In response, the profile server 14 returns the user profile of the user 20-1 to the mobile device 18-1 (step 1102B). The MAP client 30-1 of the mobile device 18-1 then sends the user profile of the user 20-1 to the MAP server 12 (step 1102C). Note that while in this embodiment the MAP client 30-1 sends the complete user profile of the user 20-1 to the MAP server 12, in an alternative embodiment, the MAP client 30-1 may filter the user profile of the user 20-1 according to criteria specified by the user 20-1. For example, the user profile of the user 20-1 may include demographic information, general interests, music interests, and movie interests, and the user 20-1 may specify that the demographic information or some subset thereof is to be filtered, or removed, before sending the user profile to the MAP server 12.

[0066] Upon receiving the user profile of the user 20-1 from the MAP client 30-1 of the mobile device 18-1, the profile manager 52 of the MAP server 12 processes the user profile (step 1102D). More specifically, in the preferred embodiment, the profile manager 52 includes social network handlers for the social network services supported by the MAP server 12. Thus, for example, if the MAP server 12 supports user profiles from the Facebook® social networking service, the MySpace® social networking service, and the LinkedIn® social networking service, the profile manager 52 may include a handler for the Facebook® social network, a handler for the MySpace® social network, and a handler for the LinkedIn® social network. The social network handlers process user profiles to generate user profiles for the MAP server 12 that include lists of keywords for each of a number of profile categories. The profile categories may be the same for each of the social network handlers or different for each of the social network handlers. Thus, for this example assume that the user profile of the user 20-1 is from the Facebook® social networking service. The profile manager 52 uses the handler for the Facebook® social network to process the user profile of the user 20-1 to map the user profile of the user 20-1 from the Facebook® social networking service to a user profile for the MAP server 12 including lists of keywords for a number of predefined profile categories. For example, for the handler for the Facebook® social network, the profile categories may be a demographic profile category, a social interaction profile category, a general interests profile category, a music interests profile category, and a movie interests profile category. As such, the user profile of the user 20-1 from the Facebook® social networking service may be processed by the handler for the Facebook® social network to create a list of keywords such as, for example, liberal, High School Graduate, 35-44, College Graduate, etc. for the demographic profile category; a list of keywords such as Seeking Friendship for the social interaction profile category; a list of keywords such as politics, technology, photography, books, etc. for the general interests profile category; a list of keywords including music genres, artist names, album names, or the like for the music interests profile category; and a list of keywords including movie titles, actor or actress names, director names, movie genres, or the like for the movie interests profile category. In one embodiment, the profile manager 52 may use natural language processing or semantic analysis. For example, if the Facebook® user profile of the user 20-1 states that the user 20-1 is 20 years old, semantic analysis may result in the

keyword of 18-24 years old being stored in the user profile of the user 20-1 for the MAP server 12.

**[0067]** After processing the user profile of the user 20-1, the profile manager 52 of the MAP server 12 stores the resulting user profile for the user 20-1 (step 1102E). More specifically, in one embodiment, the MAP server 12 stores user records for the users 20-1 through 20-N in the datastore 64 (FIG. 2). The user profile of the user 20-1 is stored in the user record of the user 20-1. The user record of the user 20-1 includes a unique identifier of the user 20-1, the current location of the user 20-1, the user profile of the user 20-1, and, once filtering is performed, the filtered user profile of the user 20-1. Note that the user profile of the user 20-1 may be updated as desired. For example, in one embodiment, the user profile of the user 20-1 is updated by repeating step 1102 each time the user 20-1 activates the MAP application 32-1.

**[0068]** FIG. 6 illustrates step 1000 of FIG. 4 in more detail according to another exemplary embodiment of the present disclosure. More specifically, FIG. 6 illustrates a process by which the MAP server 12 obtains the current location of the user 20-1 according to another exemplary embodiment of the present disclosure. This example is directed to obtaining the user profile of the user 20-1. However, this process may also be used to obtain the user profiles of the other users 20-2 through 20-N. First, an authentication process is performed (step 1200). For authentication, in this embodiment, the mobile device 18-1 authenticates with the MAP server 12 (step 1200A), and the MAP server 12 authenticates with the profile server 14 (step 1200B). Preferably, authentication is performed using OpenID or similar technology. However, authentication may alternatively be performed using separate credentials (e.g., username and password) of the user 20-1 for access to the MAP server 12 and the profile server 14. Assuming that authentication is successful, the profile server 14 returns an authentication succeeded message to the MAP server 12 (step 1200C), and the MAP server 12 returns an authentication succeeded message to the MAP client 30-1 of the mobile device 18-1 (step 1200D).

**[0069]** At some point after authentication is complete, a user profile process is performed such that a user profile of the user 20-1 is obtained from the profile server 14 and delivered to the MAP server 12 (step 1202). In this embodiment, the profile manager 52 of the MAP server 12 sends a profile request to the profile server 14 (step 1202A). In response, the profile server 14 returns the user profile of the user 20-1 to the profile manager 52 of the MAP server 12 (step 1202B). Note that while in this embodiment the profile server 14 returns the complete user profile of the user 20-1 to the MAP server 12, in an alternative embodiment, the profile server 14 may return a filtered version of the user profile of the user 20-1 to the MAP server 12. The profile server 14 may filter the user profile of the user 20-1 according to criteria specified by the user 20-1. For example, the user profile of the user 20-1 may include demographic information, general interests, music interests, and movie interests, and the user 20-1 may specify that the demographic information or some subset thereof is to be filtered, or removed, before sending the user profile to the MAP server 12.

**[0070]** Upon receiving the user profile of the user 20-1, the profile manager 52 of the MAP server 12 processes to the user profile (step 1202C). More specifically, as discussed above, in the preferred embodiment, the profile manager 52 includes social network handlers for the social network services supported by the MAP server 12. The social network handlers

process user profiles to generate user profiles for the MAP server 12 that include lists of keywords for each of a number of profile categories. The profile categories may be the same for each of the social network handlers or different for each of the social network handlers.

**[0071]** After processing the user profile of the user 20-1, the profile manager 52 of the MAP server 12 stores the resulting user profile for the user 20-1 (step 1202D). More specifically, in one embodiment, the MAP server 12 stores user records for the users 20-1 through 20-N in the datastore 64 (FIG. 2). The user profile of the user 20-1 is stored in the user record of the user 20-1. The user record of the user 20-1 includes a unique identifier of the user 20-1, the current location of the user 20-1, the user profile of the user 20-1, and, once filtering is performed, the filtered user profile of the user 20-1. Note that the user profile of the user 20-1 may be updated as desired. For example, in one embodiment, the user profile of the user 20-1 is updated by repeating step 1202 each time the user 20-1 activates the MAP application 32-1.

**[0072]** FIG. 7 illustrates a process by which location updates are sent to the MAP server 12 according to one embodiment of the present disclosure. In one embodiment, the current location of the user 20-1 received by the MAP server 12 in step 1002 of FIG. 4 is received via this process. This example illustrates location updates for the user 20-1 of the mobile device 18-1. However, this process may be used for location updates for the other users 20-2 through 20-N of the other mobile devices 18-2 through 18-N. First, in this example, a process is performed such that a current location of the mobile device 18-1 and thus a current location of the user 20-1 is obtained by the MAP server 12 (step 1300). In this embodiment, the MAP application 32-1 of the mobile device 18-1 obtains the current location of the mobile device 18-1 from the location function 36-1 of the mobile device 18-1. The MAP application 32-1 then provides the current location of the user 20-1 of the mobile device 18-1 to the location server 16 (step 1300A). Note that step 1300A may be repeated periodically or in response to changes in the location of the mobile device 18-1 in order to provide location updates for the user 20-1 to the MAP server 12. The location server 16 then provides the current location of the user 20-1 to the MAP server 12 (step 1300B). The location server 16 may provide the current location of the user 20-1 to the MAP server 12 automatically in response to receiving the current location of the user 20-1 from the mobile device 18-1 or in response to a request from the MAP server 12.

**[0073]** In response to receiving the current location of the mobile device 18-1, the location manager 54 of the MAP server 12 stores the current location of the mobile device 18-1 as the current location of the user 20-1 (step 1300C). More specifically, in one embodiment, the current location of the user 20-1 is stored in the user record of the user 20-1 maintained in the datastore 64 of the MAP server 12. Note that, preferably, only the current location of the user 20-1 is stored in the user record of the user 20-1. In this manner, the MAP server 12 maintains privacy for the user 20-1 since the MAP server 12 does not maintain a historical record of the location of the user 20-1. As discussed below in detail, historical data maintained by the MAP server 12 is anonymized in order to maintain the privacy of the users 20-1 through 20-N.

**[0074]** As discussed above, the use of the location server 16 is particularly beneficial when the mobile device 18-1 does not permit background processes. As such, if the mobile device 18-1 does not permit background processes, the MAP

application 32-1 will not provide location updates for the user 20-1 to the location server 16 unless the MAP application 32-1 is active. However, other applications running on the mobile device 18-1 (or some other device of the user 20-1) may provide location updates to the location server 16 for the user 20-1 when the MAP application 32-1 is not active. This is illustrated in step 1302 where the location server 16 receives a location update for the user 20-1 from another application running on the mobile device 18-1 or an application running on another device of the user 20-1 (step 1302A). The location server 16 then provides the location update for the user 20-1 to the MAP server 12 (step 1302B). In response, the location manager 54 updates and stores the current location of the user 20-1 in the user record of the user 20-1 (step 1302C). In this manner, the MAP server 12 is enabled to obtain location updates for the user 20-1 even when the MAP application 32-1 is not active at the mobile device 18-1.

[0075] FIG. 8 illustrates step 1008 of FIG. 4 in more detail according to one embodiment of the present disclosure. More specifically, FIG. 8 illustrates a process for filtering the user profile of the user 20-1 based on aggregate profile data for the current location of the user 20-1 and, optionally, additional information obtained for the current location of the user 20-1 according to one embodiment of the present disclosure. First, a cut-off value is determined (step 1400). As discussed below, the cut-off value is utilized to filter interests in the user profile of the user 20-1 that correspond to interests in the aggregate profile data having representation values that are less than the cut-off value. In one embodiment, the cut-off value is a predefined, static value.

[0076] In another embodiment, the cut-off value is a function of a highest representation value of the representation values for the interests in the aggregate profile data. More specifically, the cut-off value may be a defined percentage of the highest representation value in the aggregate profile data. The defined percentage may be a system-defined percentage such as, for example, 50%, or  $\frac{1}{2}$ . Alternatively, the defined percentage may be a user-configurable percentage that is defined by the user 20-1 via, for example, a Graphical User Interface (GUI) provided by the MAP application 32-1. As such, the cut-off value may be determined by first determining the highest representation value from the representation values for the interests in the aggregate profile data. Then, the cut-off value is computed as the defined percentage of the highest representation value.

[0077] In another embodiment, the cut-off value is a function of a highest representation value of the representation values for the interests in the aggregate profile data and an amount of time that the user 20-1 has been located at the same location. In one embodiment, the amount of time that the user 20-1 has been located at the same location is an amount of time that the user 20-1 has been located within a defined tolerance distance from a corresponding reference location. With regard to the reference location, in one embodiment, when a first location update is received for the user 20-1, the location of the user 20-1 at that time is defined as the reference location for the user 20-1, and the current time and optionally date are stored as a reference timestamp for the user 20-1. As long as the user 20-1 remains within a predefined tolerance distance from the reference location, the reference location remains the same and the user 20-1 is determined to be at the same location for purposes of revealing more of the user profile of the user 20-1 over time. Once the user 20-1 moves outside of the tolerance distance from the reference location,

the location of the user 20-1 at that time is stored as the reference location for the user 20-1, and the reference timestamp is set to the time and optionally date at that time.

[0078] The amount of time that the user 20-1 has been at or near the current location is determined by first determining if the current location of the user 20-1 is within the predefined tolerance distance from the reference location stored for the user 20-1. If so, the user 20-1 is determined to have remained at the same location, and the amount of time that the user 20-1 has been at that location is computed as a difference between the current time or timestamp associated with the corresponding location update received for the user 20-1 and the reference timestamp stored for the user 20-1. Otherwise, the user 20-1 is determined not to be at the same location and, as such, the reference location and reference timestamp are updated and the amount of time that the user 20-1 has been at the same location is set to zero. The cut-off value is computed as a function of the aggregate profile data and the amount of time that the user 20-1 has been at the same location.

[0079] More specifically, in one embodiment, the cut-off value decreases from an initial cut-off value to a minimum cut-off value as the amount of time that the user 20-1 remains at the same location increases from a minimum value (e.g., 0) to a maximum value in an adjustment period. In this manner, as the user 20-1 remains at the same location, more of the user profile of the user 20-1 is revealed (i.e., included in the filtered user profile of the user 20-1). In one embodiment, the initial cut-off value is a predefined, system value. In another embodiment, the initial cut-off value is computed as a predefined percentage of the highest representation value in the aggregate profile data. The predefined percentage may be system-defined or configurable by the user 20-1. For example, the predefined percentage may be 50%, or  $\frac{1}{2}$ , such that the initial cut-off value is half of the highest representation value in the aggregate profile data. Then, an adjustment value is subtracted from the initial cut-off value to provide the cut-off value, where the adjustment value is a function of the amount of time that the user 20-1 has been at the location.

[0080] The adjustment value subtracted from the initial cut-off value goes from a defined minimum value (e.g., 0) to a defined maximum value (e.g., half the highest representation value) over the adjustment period (e.g., 1 hour). By increasing the subtracted adjustment value over time, the cut-off value decreases from the initial cut-off value to the minimum cut-off value over the adjustment period. More specifically, a number of time thresholds within the adjustment period and corresponding adjustment values are defined. For example, the time thresholds and corresponding adjustment values may be defined such that the adjustment value is 10% of the highest representation value in the aggregate profile after the user 20-1 has remained at the same location for 20 minutes, 20% of the highest representation value in the aggregate profile after the user 20-1 has remained at the same location for 30 minutes, 30% of the highest representation value in the aggregate profile after the user 20-1 has remained at the same location for 40 minutes, 40% of the highest representation value in the aggregate profile after the user 20-1 has remained at the same location for 50 minutes, and 50% of the highest representation value in the aggregate profile after the user 20-1 has remained at the same location for 60 minutes. Therefore, continuing this example, if the amount of time that the user 20-1 has been at the same location is 35 minutes, then the adjustment value is set to 20%

of the highest representation value, and this adjustment value is subtracted from the initial cut-off value to provide the cut-off value.

**[0081]** In one embodiment, the adjustment period, the number of time thresholds, and the corresponding adjustment values are static and may be either be system-defined or configurable by the user **20-1**. In another embodiment, the adjustment period, the number of time thresholds, and/or the adjustment values are dynamic. More specifically, the adjustment period, the number of time thresholds, and/or the adjustment values may change each time the user **20-1** changes locations (e.g., a new reference location is set in response to the user **20-1** no longer being in the same location). For example, the adjustment period may be defined as an initial adjustment period plus or minus a variance value. Both the variance value and whether the variance value is added or subtracted from the initial adjustment period may be generated using corresponding random number generators, which may be implemented in software or hardware. Further, a possible maximum value for the variance value may be increased as a frequency at which the user **20-1** has visited the same location and/or a number of times that the user **20-1** has visited the same location increases. Alternatively, the adjustment period may be generated using a random number generator such that a different adjustment period is used each time the user **20-1** changes locations.

**[0082]** In addition to or as an alternative to varying the adjustment period, the adjustment values for the time thresholds within the adjustment period may be varied. For example, the maximum adjustment value may be defined as an initial maximum adjustment value (e.g., 50% of the highest representation value in the aggregate profile data) plus or minus a variance value. The variance value and whether the variance value is added or subtracted from the initial maximum adjustment value may be generated using random number generators each time the user **20-1** changes locations (e.g., each time a new reference location is set in response to the user **20-1** no longer being in the same location). The maximum value for the variance value may be a static value or may increase as the frequency at which the user **20-1** visits the same location or the number of times that the user **20-1** has visited the same location. Alternatively, the maximum adjustment value may be generated using a random number generator where minimum and maximum values for the maximum adjustment value for purposes of random number generation may be static values or may vary depending on the frequency at which the user **20-1** has visited the same location or the number of times that the user **20-1** has visited the same location.

**[0083]** In addition or alternatively, while the discussion above describes embodiments where the cut-off value is a function of the amount of time that the user **20-1** has been at the same location, in a similar manner, the cut-off value may be a function of an amount of time that the user **20-1** has been in the same group of users. The same group of users may be determined to be the same as long as a predefined threshold number or percentage of users in the group of users remains the same. Here, the group of users may be, for example, a crowd of users formed by the MAP server **12**. In a similar manner, the adjustment period, the adjustment values for the time thresholds within the adjustment period, and/or the number of time thresholds within the adjustment period may additionally or alternatively vary based whether the user **20-1** has previously been in the same group of users, the frequency

at which the user **20-1** has been in the same group of users, the number of times that the user **20-1** has been in the same group of users, or the like. Information regarding the groups of users that the user **20-1** has been in may be maintained by the MAP server **12** or the mobile device **18-1** of the user **20-1**.

**[0084]** Once the maximum adjustment value has been generated, the maximum adjustment value is assigned to a last time threshold of the time thresholds defined for the adjustment period. The adjustment values for the other time thresholds in the adjustment period may be defined as:

$$\text{adjustment\_value}_i = \frac{\text{adjustment\_value}_{MAX} - \text{adjustment\_value}_{MIN}}{\text{Num}} \cdot i,$$

for  $i = 1 \dots \text{Num}$ ,

**[0085]** where  $\text{adjustment\_value}_i$  is the  $i$ -th adjustment value in the adjustment period,  $\text{adjustment\_value}_{MAX}$  is the maximum adjustment value,  $\text{adjustment\_value}_{MIN}$  is a defined minimum adjustment value which may be zero, and Num is the number of time thresholds in the adjustment period. The number of time thresholds (Num) may be a static value (e.g., 5) or may vary depending on the adjustment period. For example, the time thresholds may occur every 15 minutes such that the number of time thresholds (Num) varies depending on the length of the adjustment period. For example, if the adjustment period is 1 hour, the number of time thresholds (Num) may be 4. In contrast, if the adjustment period is 1.5 hours, the number of time thresholds (Num) is 6.

**[0086]** In addition to or as an alternative to varying the minimum cut-off value, the adjustment period, and/or the adjustment values for the time thresholds in the adjustment period, and the number of time thresholds in the adjustment period may vary. The number of time thresholds may change each time the user **20-1** changes locations (e.g., a new reference location is set in response to the user **20-1** no longer being in the same location). For example, the number of time thresholds may be generated using a random number generator and a defined minimum and maximum value each time the user **20-1** changes locations.

**[0087]** Note that, in order to provide the aforementioned features for varying the adjustment period, the number of time thresholds, and/or the corresponding adjustment values based on the frequency at which or number of times that the user **20-1** has visited the same location or the number of times that the user **20-1** has visited the same location, this information is maintained or otherwise accessible. For example, a historical record of locations (e.g., POIs) visited by the user **20-1** and the frequency at which or number of times that the user **20-1** has visited each location may be maintained by the mobile device **18-1**, the MAP server **12**, or the location server **16**.

**[0088]** After determining the cut-off value, the filtered user profile of the user **20-1** is initialized (step **1402**). The filtered user profile of the user **20-1** is initialized by creating the filtered user profile if it does not already exist. If the filtered user profile already exists, all interests are removed from the filtered user profile. Then, the representation value from the aggregate profile data for the current location of the user **20-1** for the next interest in the user profile of the user **20-1** is obtained (step **1404**). A determination is then made as to whether the representation value is greater than or equal to the cut-off value (step **1406**). If not, the process proceeds to step

**1410.** Otherwise, the interest is added to the filtered user profile of the user **20-1** (step **1408**). At this point, whether proceeding from step **1406** or **1408**, a determination is made as to whether the last interest from the user profile of the user **20-1** has been processed (step **1410**). If not, the process returns to step **1404** and is repeated.

**[0089]** Once all of the interests from the user profile of the user **20-1** have been processed, the filtered user profile of the user **20-1** may be further filtered based on the additional information obtained for the current location of the user **20-1** (step **1412**). For example, the additional information may be expressed as one or more keywords, or the additional information may be processed to extract one or more keywords. The filtered user profile may then be further filtered to remove any interests in the filtered user profile that do not match the one or more keywords provided as or extracted from the additional information for the current location to at least a defined threshold degree. As an example, if the user **20-1** is currently located at an arena at which a music concert is being held, the additional information for the current location may include keywords such as music, concert, a keyword reflecting a music genre of the concert, a keyword corresponding to a name of the performer for the concert, or the like. Interests in the filtered user profile of the user **20-1** that do not match or, in some embodiments, are not directly or indirectly related to at least one of the keywords in or extracted from the additional information may be removed from the filtered user profile of the user **20-1**. Note that step **1412** is optional.

**[0090]** In addition, in this embodiment, the filtered user profile of the user **20-1** may be further filtered based on one or more situational awareness rules (step **1414**). More specifically, in one embodiment, a community of users such as, but not limited to, the users **20-1** through **20-N** define and submit situational awareness rules to the MAP server **12**. For example, a University of North Carolina (UNC) study may define and submit a rule stating that any interests indicating that that a user is a fan, alumni, or student of UNC is not to be revealed if that person is surrounded by more users having user profiles having interests indicating that they are fans, alumni, or students of Duke University than users having user profiles having interests indicating that they are fans, alumni, or students of UNC. The mobile device **18-1** may provide information defining a situation of the user **20-1**, where this information is then compared to the one or more situational awareness rules to further filter the user profile of the user **20-1**. Note that like step **1412**, step **1414** is also optional.

**[0091]** Before proceeding, an embodiment where the user profile filtering process has multiple modes of operation is to be described. In general, there may be two modes of operation, namely, a Sharing mode and a Blend mode. In the Sharing mode, user profile filtering is performed such that the cut-off value determined in step **1400** is a function of time, frequency at which the user **20-1** visits the same location, and/or the number of times that the user **20-1** has visited the same location. In the Blend mode, the user profile filtering is performed such that the cut-off value is static (e.g., remains fixed at a static percentage of the highest representation value in the aggregate profile data). In one embodiment, the user **20-1** is enabled to manually select the desired mode of operation via the MAP application **32-1** of the mobile device **18-1**. In addition or alternatively, either the MAP server **12** or the mobile device **18-1** may monitor the selected mode of operation with respect to location and then recommend a mode of operation to the user **20-1** when the user **20-1** enters a new

location based on the mode of operation that the user **20-1** previously selected when at the same or a similar location.

**[0092]** FIG. **9** illustrates the operation of a mobile device to tailor a user profile of an associated user according to another embodiment of the present disclosure. In general, in FIG. **9**, filtering of the user profile is performed at the mobile device rather than the MAP server **12**. Otherwise, this process is similar to that described above with respect to FIG. **4**. Again, using the user **20-1** of the mobile device **18-1** as an example, first, the MAP application **32-1** of the mobile device **18-1** obtains the user profile of the user **20-1** (step **1500**). In one embodiment, the MAP application **32-1** obtains the user profile of the user **20-1** from one of the profile servers **14**. In another embodiment, the MAP application **32-1** provides a GUI that enables the user **20-1** to manually define the user profile of the user **20-1** at the mobile device **18-1**. In addition to the user profile of the user **20-1**, the MAP application **32-1** obtains the current location of the user **20-1** from the location function **36-1** (step **1502**).

**[0093]** Next, the MAP application **32-1** sends a request to the MAP server **12** for aggregate profile data for the current location of the user **20-1** (step **1504**). In response to the request, the MAP application **32-1** receives aggregate profile data for the current location from the MAP server **12** (step **1506**). As discussed above, the aggregate profile data includes current aggregate profile data and/or historical aggregate profile data for the current location of the user **20-1**. In addition, the MAP application **32-1** may obtain additional information for the current location of the user **20-1** (step **1508**). In one embodiment, as discussed above, the MAP server **12** maintains a database or repository of additional information for a number of locations, such as POIs. The MAP application **32-1** may then query the MAP server **12** for any additional information for the current location of the user **20-1**. In another embodiment, the MAP application **32-1** queries one or more remote sources, other than the MAP server **12**, for any additional information for the current location of the user **20-1**.

**[0094]** Next, the MAP application **32-1** filters the user profile of the user **20-1** based on the aggregate profile data for the current location of the user **20-1**, the additional information for the current location of the user **20-1**, and one or more filtering rules (step **1510**). The filtering process is the same as that described above with respect to FIG. **8**. As such, the details will not be repeated. Once the user profile has been filtered to provide the filtered user profile of the user **20-1**, the MAP application **32-1** sends the filtered user profile to the MAP server **12** (step **1512**). The MAP server **12** stores the filtered user profile as the user profile of the user **20-1**. At this point, the process returns to step **1502** and is repeated such that the filtered user profile is updated over time as the location and/or aggregate profile data for the current location of the user **20-1** changes.

**[0095]** FIGS. **10** through **18** illustrate the operation of the MAP server **12** to maintain a historical record of anonymized user profile data and to provide historical aggregate profile data based thereon according to one embodiment of the present disclosure. Historical aggregate profile data provided by the MAP server **12** using the following processes may be used to filter user profile data as described above. As illustrated in FIG. **10**, in the preferred embodiment, the history manager **56** maintains lists of users located in a number of geographic regions, or "location buckets." Preferably, the location buckets are defined by floor(latitude, longitude) to a desired resolution. The higher the resolution, the smaller the

size of the location buckets. For example, in one embodiment, the location buckets are defined by floor(latitude, longitude) to a resolution of  $\frac{1}{10,000}$  of a degree such that the lower left-hand corners of the squares illustrated in FIG. 10 are defined by the floor(latitude, longitude) values at a resolution of  $\frac{1}{10,000}$  of a degree. In the example of FIG. 10, users are represented as dots, and location buckets 72 through 88 have lists of 1, 3, 2, 1, 1, 2, 1, 2, and 3 users, respectively.

[0096] As discussed below in detail, at a predetermined time interval such as, for example, 15 minutes, the history manager 56 makes a copy of the lists of users in the location buckets, anonymizes the filtered user profiles of the users in the lists to provide anonymized user profile data for the corresponding location buckets, and stores the anonymized user profile data in a number of history objects. Note that if filtered user profiles of any of the users in the lists are not available, the user profiles of those users may be anonymized and stored in the history objects. In one embodiment, a history object is stored for each location bucket having at least one user. In another embodiment, a quadtree algorithm is used to efficiently create history objects for geographic regions (i.e., groups of one or more adjoining location buckets).

[0097] FIG. 11 graphically illustrates a scenario where a user moves from one location bucket to another, namely, from the location bucket 74 to the location bucket 76. As discussed below in detail, assuming that the movement occurs during the time interval between persistence of the historical data by the history manager 56, the user is included on both the list for the location bucket 74 and the list for the location bucket 76. However, the user is flagged or otherwise marked as inactive for the location bucket 74 and active for the location bucket 76. As discussed below, after making a copy of the lists for the location buckets to be used to persist the historical data, users flagged as inactive are removed from the lists of users for the location buckets. Thus, in sum, once a user moves from the location bucket 74 to the location bucket 76, the user remains in the list for the location bucket 74 until the predetermined time interval has expired and the anonymized user profile data is persisted. The user is then removed from the list for the location bucket 74.

[0098] FIG. 12 is a flow chart illustrating the operation of a foreground “bucketization” process performed by the history manager 56 to maintain the lists of users for location buckets according to one embodiment of the present disclosure. First, the history manager 56 receives a location update for a user (step 1600). For this discussion, assume that the location update is received for the user 20-1. The history manager 56 then determines a location bucket corresponding to the updated location (i.e., the current location) of the user 20-1 (step 1602). In the preferred embodiment, the location of the user 20-1 is expressed as latitude and longitude coordinates, and the history manager 56 determines the location bucket by determining floor values of the latitude and longitude coordinates, which can be written as floor(latitude, longitude) at a desired resolution. As an example, if the latitude and longitude coordinates for the location of the user 20-1 are 32.24267381553987 and -111.9249213502935, respectively, and the floor values are to be computed to a resolution of  $\frac{1}{10,000}$  of a degree, then the floor values for the latitude and longitude coordinates are 32.2426 and -111.9249. The floor values for the latitude and longitude coordinates correspond to a particular location bucket.

[0099] After determining the location bucket for the location of the user 20-1, the history manager 56 determines

whether the user 20-1 is new to the location bucket (step 1604). In other words, the history manager 56 determines whether the user 20-1 is already on the list of users for the location bucket. If the user 20-1 is new to the location bucket, the history manager 56 creates an entry for the user 20-1 in the list of users for the location bucket (step 1606). Returning to step 1604, if the user 20-1 is not new to the location bucket, the history manager 56 updates the entry for the user 20-1 in the list of users for the location bucket (step 1608). At this point, whether proceeding from step 1606 or 1608, the user 20-1 is flagged as active in the list of users for the location bucket (step 1610).

[0100] The history manager 56 then determines whether the user 20-1 has moved from another location bucket (step 1612). More specifically, the history manager 56 determines whether the user 20-1 is included in the list of users for another location bucket and is currently flagged as active in that list. If the user 20-1 has not moved from another location bucket, the process proceeds to step 1616. If the user 20-1 has moved from another location bucket, the history manager 56 flags the user 20-1 as inactive in the list of users for the other location bucket from which the user 20-1 has moved (step 1614).

[0101] At this point, whether proceeding from step 1612 or 1614, the history manager 56 determines whether it is time to persist (step 1616). More specifically, as mentioned above, the history manager 56 operates to persist history objects at a predetermined time interval such as, for example, every 15 minutes. Thus, the history manager 56 determines that it is time to persist if the predetermined time interval has expired. If it is not time to persist, the process returns to step 1600 and is repeated for a next received location update, which will typically be for another user. If it is time to persist, the history manager 56 creates a copy of the lists of users for the location buckets and passes the copy of the lists to an anonymization and storage process (step 1618). In this embodiment, the anonymization and storage process is a separate process performed by the history manager 56. The history manager 56 then removes inactive users from the lists of users for the location buckets (step 1620). The process then returns to step 1600 and is repeated for a next received location update, which will typically be for another user.

[0102] FIG. 13 is a flow chart illustrating the anonymization and storage process performed by the history manager 56 at the predetermined time interval according to one embodiment of the present disclosure. First, the anonymization and storage process receives the copy of the lists of users for the location buckets passed to the anonymization and storage process by the bucketization process of FIG. 12 (step 1700). Next, anonymization is performed for each of the location buckets having at least one user in order to provide anonymized user profile data for the location buckets (step 1702). Anonymization prevents connecting information stored in the history objects stored by the history manager 56 back to the users 20-1 through 20-N or at least substantially increases a difficulty of connecting information stored in the history objects stored by the history manager 56 back to the users 20-1 through 20-N. Lastly, the anonymized user profile data for the location buckets is stored in a number of history objects (step 1704). In one embodiment, a separate history object is stored for each of the location buckets, where the history object of a location bucket includes the anonymized user profile data for the location bucket. In another embodiment, as discussed below, a quadtree algorithm is used to



efficiently store the anonymized user profile data in a number of history objects such that each history object stores the anonymized user profile data for one or more location buckets.

[0103] FIG. 14 graphically illustrates one embodiment of the anonymization process of step 1702 of FIG. 13. In this embodiment, anonymization is performed by creating anonymous user records for the users in the lists of users for the location buckets. The anonymous user records are not connected back to the users 20-1 through 20-N. More specifically, as illustrated in FIG. 14, each user in the lists of users for the location buckets has a corresponding user record 90. The user record 90 includes a unique user identifier (ID) for the user and the current location of the user. In addition, in the embodiment where filtering of user profiles is performed at the MAP server 12 (e.g., FIG. 4), the user record 90 also includes both the user profile of the user and the filtered user profile of the user. Alternatively, in the embodiment where filtering of user profiles is performed by corresponding mobile devices (e.g., FIG. 9), the user record 90 includes only the filtered user profile of the user. The user profile includes keywords for each of a number of profile categories, which are stored in corresponding profile category records 92-1 through 92-M1. Each of the profile category records 92-1 through 92-M1 includes a user ID for the corresponding user which may be the same user ID used in the user record 90, a category ID, and a list of keywords for the profile category. Likewise, the filtered user profile includes keywords for each of a number of profile categories, which are stored in corresponding profile category records 93-1 through 93-M2. Each of the profile category records 93-1 through 93-M2 includes a user ID for the corresponding user, which may be the same user ID used in the user record 90, a category ID, and a list of filtered keywords for the profile category.

[0104] For anonymization, an anonymous user record 94 is created from the user record 90. In the anonymous user record 94, the user ID is replaced with a new user ID that is not connected back to the user, which is also referred to herein as an anonymous user ID. This new user ID is different than any other user ID used for anonymous user records created from the user record of the user for any previous or subsequent time periods. In this manner, anonymous user records for a single user created over time cannot be linked to one another.

[0105] In addition, anonymous profile category records 96-1 through 96-M2 are created for the profile category records 93-1 through 93-M2 for the filtered user profile. In the anonymous profile category records 96-1 through 96-M2, the user ID is replaced with a new user ID, which may be the same new user ID included in the anonymous user record 94. The anonymous profile category records 96-1 through 96-M2 include the same category IDs and lists of filtered keywords as the corresponding profile category records 93-1 through 93-M2. Note that the location of the user is not stored in the anonymous user record 94. With respect to location, it is sufficient that the anonymous user record 94 is linked to a location bucket.

[0106] In another embodiment, the history manager 56 performs anonymization in a manner similar to that described above with respect to FIG. 14. However, in this embodiment, the profile category records from the filtered user profiles of the group of users in a location bucket, or the group of users in a number of location buckets representing a node in a quadtree data structure (see below), may be selectively randomized among the anonymous user records of those users.

In other words, each anonymous user record would have a user profile including a selectively randomized set of profile category records (including keywords) from a cumulative list of profile category records from the filtered user profiles for all of the users in the group.

[0107] In yet another embodiment, rather than creating anonymous user records 94 for the users in the lists maintained for the location buckets, the history manager 56 may perform anonymization by storing an aggregate user profile for each location bucket, or each group of location buckets representing a node in a quadtree data structure (see below). The aggregate user profile may include a list of all keywords and potentially the number of occurrences of each keyword in the filtered user profiles of the corresponding group of users. In this manner, the data stored by the history manager 56 is not connected back to the users 20-1 through 20-N.

[0108] FIG. 15 is a flow chart illustrating the storing step (step 1704) of FIG. 13 in more detail according to one embodiment of the present disclosure. First, the history manager 56 processes the location buckets using a quadtree algorithm to produce a quadtree data structure, where each node of the quadtree data structure includes one or more of the location buckets having a combined number of users that is at most a predefined maximum number of users (step 1800). The history manager 56 then stores a history object for each node in the quadtree data structure having at least one user (step 1802).

[0109] Each history object includes location information, timing information, data, and quadtree data structure information. The location information included in the history object defines a combined geographic area of the location bucket(s) forming the corresponding node of the quadtree data structure. For example, the location information may be latitude and longitude coordinates for a northeast corner of the combined geographic area of the node of the quadtree data structure and a southwest corner of the combined geographic area for the node of the quadtree data structure. The timing information includes information defining a time window for the history object, which may be, for example, a start time for the corresponding time interval and an end time for the corresponding time interval. The data includes the anonymized user profile data generated from the filtered user profiles of the users in the list(s) maintained for the location bucket(s) forming the node of the quadtree data structure for which the history object is stored. In addition, the data may include a total number of users in the location bucket(s) forming the node of the quadtree data structure. Lastly, the quadtree data structure information includes information defining a quadtree depth of the node in the quadtree data structure.

[0110] FIG. 16 is a flow chart illustrating a quadtree algorithm that may be used to process the location buckets to form the quadtree data structure in step 1800 of FIG. 15 according to one embodiment of the present disclosure. Initially, a geographic area served by the MAP server 12 is divided into a number of geographic regions, each including multiple location buckets. These geographic regions are also referred to herein as base quadtree regions. The geographic area served by the MAP server 12 may be, for example, a city, a state, a country, or the like. Further, the geographic area may be the only geographic area served by the MAP server 12 or one of a number of geographic areas served by the MAP server 12. Preferably, the base quadtree regions have a size of  $2^n \times 2^n$  location buckets, where n is an integer greater than or equal to 1.



[0111] In order to form the quadtree data structure, the history manager 56 determines whether there are any more base quadtree regions to process (step 1900). If there are more base quadtree regions to process, the history manager 56 sets a current node to the next base quadtree region to process, which for the first iteration is the first base quadtree region (step 1902). The history manager 56 then determines whether the number of users in the current node is greater than a predefined maximum number of users and whether a current quadtree depth is less than a maximum quadtree depth (step 1904). In one embodiment, the maximum quadtree depth may be reached when the current node corresponds to a single location bucket. However, the maximum quadtree depth may be set such that the maximum quadtree depth is reached before the current node reaches a single location bucket.

[0112] If the number of users in the current node is greater than the predefined maximum number of users and the current quadtree depth is less than a maximum quadtree depth, the history manager 56 creates a number of child nodes for the current node (step 1906). More specifically, the history manager 56 creates a child node for each quadrant of the current node. The users in the current node are then assigned to the appropriate child nodes based on the location buckets in which the users are located (step 1908), and the current node is then set to the first child node (step 1910). At this point, the process returns to step 1904 and is repeated.

[0113] Once the number of users in the current node is not greater than the predefined maximum number of users or the maximum quadtree depth has been reached, the history manager 56 determines whether the current node has any more sibling nodes (step 1912). Sibling nodes are child nodes of the same parent node. If so, the history manager 56 sets the current node to the next sibling node of the current node (step 1914), and the process returns to step 1904 and is repeated. Once there are no more sibling nodes to process, the history manager 56 determines whether the current node has a parent node (step 1916). If so, since the parent node has already been processed, the history manager 56 determines whether the parent node has any sibling nodes that need to be processed (step 1918). If the parent node has any sibling nodes that need to be processed, the history manager 56 sets the next sibling node of the parent node to be processed as the current node (step 1920). From this point, the process returns to step 1904 and is repeated. Returning to step 1916, if the current node does not have a parent node, the process returns to step 1900 and is repeated until there are no more base quadtree regions to process. Once there are no more base quadtree regions to process, the finished quadtree data structure is returned to the process of FIG. 15 such that the history manager 56 can then store the history objects for nodes in the quadtree data structure having at least one user (step 1922).

[0114] FIGS. 17A through 17E graphically illustrate the process of FIG. 16 for the generation of the quadtree data structure for one exemplary base quadtree region 98. FIG. 17A illustrates the base quadtree region 98. As illustrated, the base quadtree region 98 is an 8x8 square of location buckets, where each of the small squares represents a location bucket. First, the history manager 56 determines whether the number of users in the base quadtree region 98 is greater than the predetermined maximum number of users. In this example, the predetermined maximum number of users is 3. Since the number of users in the base quadtree region 98 is greater than

3, the history manager 56 divides the base quadtree region 98 into four child nodes 100-1 through 100-4, as illustrated in FIG. 17B.

[0115] Next, the history manager 56 determines whether the number of users in the child node 100-1 is greater than the predetermined maximum, which again for this example is 3. Since the number of users in the child node 100-1 is greater than 3, the history manager 56 divides the child node 100-1 into four child nodes 102-1 through 102-4, as illustrated in FIG. 17C. The child nodes 102-1 through 102-4 are children of the child node 100-1. The history manager 56 then determines whether the number of users in the child node 102-1 is greater than the predetermined maximum number of users, which again is 3. Since there are more than 3 users in the child node 102-1, the history manager 56 further divides the child node 102-1 into four child nodes 104-1 through 104-N, as illustrated in FIG. 17D.

[0116] The history manager 56 then determines whether the number of users in the child node 104-1 is greater than the predetermined maximum number of users, which again is 3. Since the number of users in the child node 104-1 is not greater than the predetermined maximum number of users, the child node 104-1 is identified as a node for the finished quadtree data structure, and the history manager 56 proceeds to process the sibling nodes of the child node 104-1, which are the child nodes 104-2 through 104-4. Since the number of users in each of the child nodes 104-2 through 104-4 is less than the predetermined maximum number of users, the child nodes 104-2 through 104-4 are also identified as nodes for the finished quadtree data structure.

[0117] Once the history manager 56 has finished processing the child nodes 104-1 through 104-4, the history manager 56 identifies the parent node of the child nodes 104-1 through 104-4, which in this case is the child node 102-1. The history manager 56 then processes the sibling nodes of the child node 102-1, which are the child nodes 102-2 through 102-4. In this example, the number of users in each of the child nodes 102-2 through 102-4 is less than the predetermined maximum number of users. As such, the child nodes 102-2 through 102-4 are identified as nodes for the finished quadtree data structure.

[0118] Once the history manager 56 has finished processing the child nodes 102-1 through 102-4, the history manager 56 identifies the parent node of the child nodes 102-1 through 102-4, which in this case is the child node 100-1. The history manager 56 then processes the sibling nodes of the child node 100-1, which are the child nodes 100-2 through 100-4. More specifically, the history manager 56 determines that the child node 100-2 includes more than the predetermined maximum number of users and, as such, divides the child node 100-2 into four child nodes 106-1 through 106-4, as illustrated in FIG. 17E. Because the number of users in each of the child nodes 106-1 through 106-4 is not greater than the predetermined maximum number of users, the child nodes 106-1 through 106-4 are identified as nodes for the finished quadtree data structure. Then, the history manager 56 proceeds to process the child nodes 100-3 and 100-4. Since the number of users in each of the child nodes 100-3 and 100-4 is not greater than the predetermined maximum number of users, the child nodes 100-3 and 100-4 are identified as nodes for the finished data structure. Thus, at completion, the quadtree data structure for the base quadtree region 98 includes the child nodes 104-1 through 104-4, the child nodes 102-2 through 102-4, the child nodes 106-1 through 106-4, and the child nodes 100-3 and 100-4, as illustrated in FIG. 17E.

[0119] As discussed above, the history manager 56 stores a history object for each of the nodes in the quadtree data structure including at least one user. As such, in this example, the history manager 56 stores history objects for the child nodes 104-2 and 104-3, the child nodes 102-2 and 102-4, the child nodes 106-1 and 106-4, and the child node 100-3. However, no history objects are stored for the nodes that do not have any users (i.e., the child nodes 104-1 and 104-4, the child node 102-3, the child nodes 106-2 and 106-3, and the child node 100-4).

[0120] FIG. 18 illustrates a process for generating historical aggregate profile data according to one embodiment of the present disclosure. First, upon receiving a historical request, the history manager 56 establishes a bounding box for the historical request (step 2000). In this embodiment, the historical request is a historical request for historical aggregate profile data for a user for which user profile filtering is to be performed, as described above with respect to FIGS. 4 and 9. In one embodiment where user profile filtering is performed at the MAP server 12 (e.g., FIG. 4), the historical request is from the profile manager 52 of the MAP server 12. In another embodiment where user profile filtering is performed at the mobile devices 18-1 through 18-N (e.g., FIG. 9), the historical request is from one of the mobile devices 18-1 through 18-N as part of the filtering process. Using the user 20-1 as an example, the historical request is for historical aggregate profile data for the current location of the user 20-1. As such, the bounding box is a geographic area of a predetermined shape and size centered at or otherwise encompassing the current location of the user 20-1. Note that while a bounding box is used in this example, other geographic shapes may be used to define a bounding region for the historical request (e.g., a bounding circle).

[0121] In addition to establishing the bounding box, the history manager 56 establishes a time window for the historical request (step 2002). In one embodiment, the historical request defines a relative time period such as, for example, the last week, the last month, or the like. The time window for the historical request may then be determined based on the relative time period. For example, if the historical request is for the last week and the current date and time are Sep. 17, 2009 at 10:00 pm, the history manager 56 may generate the time window as Sep. 10, 2009 at 10:00 pm through Sep. 17, 2009 at 10:00 pm. In another embodiment, a system-defined time window relative to the current time may be used as the time window for the historical request.

[0122] Next, the history manager 56 obtains history objects relevant to the bounding box and the time window for the historical request from the datastore 64 of the MAP server 12 (step 2004). The relevant history objects are history objects recorded for time periods within or intersecting the time window and for locations, or geographic areas, within or intersecting the bounding box for the historical request. The history manager 56 also determines an equivalent depth of the bounding box ( $D_{BB}$ ) within the quadtree data structures used to store the history objects (step 2006). More specifically, the area of the base quadtree region (e.g., the base quadtree region 98) is referred to as  $A_{BASE}$ . Then, at each depth of the quadtree, the area of the corresponding quadtree nodes is  $(1/4)^{D_{BB}} * A_{BASE}$ . In other words, the area of a child node is  $1/4^{th}$  of the area of the parent node of that child node. The history manager 56 determines the equivalent depth of the bounding box ( $D_{BB}$ ) by determining a quadtree depth at which the area

of the corresponding quadtree nodes most closely matches an area of the bounding box ( $A_{BB}$ ).

[0123] Note that equivalent quadtree depth of the bounding box ( $D_{BB}$ ) determined in step 2006 is used below in order to efficiently determine the ratios of the area of the bounding box ( $A_{BB}$ ) to areas of the relevant history objects ( $A_{HO}$ ). However, in an alternative embodiment, the ratios of the area of the bounding box ( $A_{BB}$ ) to the areas of the relevant history objects ( $A_{HO}$ ) may be otherwise computed, in which case step 2006 would not be needed.

[0124] At this point, the history manager 56 gets the next history object from the relevant history objects obtained in step 2004 (step 2008). Next, the history manager 56 sets a relevancy weight for the history object, where the relevancy weight is indicative of a relevancy of the history object to the bounding box (step 2010). For instance, a history object includes anonymized user profile data for a corresponding geographic area. If that geographic area is within or significantly overlaps the bounding box, then the history object will have a high relevancy weight. However, if the geographic area only overlaps the bounding box slightly, then the history object will have a low relevancy weight. In this embodiment, the relevancy weight for the history object is set to an approximate ratio of the area of the bounding box ( $A_{BB}$ ) to an area of the history object ( $A_{HO}$ ) computed based on a difference between the quadtree depth of the history object ( $D_{HO}$ ) and the equivalent quadtree depth of the bounding box ( $D_{EQ}$ ). The quadtree depth of the history object ( $D_{HO}$ ) is stored in the history object. More specifically, in one embodiment, the relevancy weight of the history object is set according to the following:

$$\text{relevancy} = \frac{A_{BB}}{A_{HO}} \cong \left(\frac{1}{4}\right)^{D_{HO} - D_{BB}},$$

for  $D_{HO} > D_{BB}$ , and

[0125]  $\text{relevancy} = 1$ , for  $D_{HO} \leq D_{BB}$ .

[0126] Next, the history manager 56 generates an aggregate profile for the history object by comparing the filtered user profiles of the anonymous user records stored in the history object to one another (step 2012). The resulting aggregate profile for the history object includes an aggregate list of interests and corresponding representation values for the interests in the aggregate list of interests. The aggregate list of interests includes all of the interests, which are expressed as keywords, from the filtered user profiles of the anonymous user records stored in the history object. Further, for each interest, or keyword, in the aggregate list of interests, the aggregate profile for the history object includes a representation value defining a degree to which the interest is expressed in the filtered user profiles of the anonymous user records stored in the history object. In one embodiment, for each interest, the representation value for the interest is a number of user matches, or occurrences, for the interest across all of the filtered user profiles of the anonymous user records stored in the history object. In another embodiment, for each interest, the representation value for the interest is a ratio, or percentage, of the number of user matches for the interest across all of the filtered user profiles of the anonymous user records stored in the history object over a total number of anonymous user records stored in the history object.

[0127] The history manager 56 then determines whether there are more relevant history objects to process (step 2014). If so, the process returns to step 2008 and is repeated until all of the relevant history objects obtained in step 2004 have been processed. Once all of the relevant history objects have been processed, the history manager 56 combines the aggregate profiles of the history objects to provide a combined aggregate profile, which is referred to herein as historical aggregate profile data for the current location of the user 20-1. More specifically, in this embodiment, the history manager 56 computes a weighted average of the aggregate profiles for the history objects using the relevancy weights of the history objects (step 2016). In order to compute the weighted average of the aggregate profiles for the history objects, the history manager 56 combines the interests in the aggregate lists of interests from the aggregate profiles generated for the history objects to provide a combined list of interests that includes all of the interests from the aggregate profiles generated for the history objects. In addition, for each interest in the combined list of interests, the history manager 56 computes a weighted average of the representation value(s) for that interest in the aggregate profiles generated for the history objects. For example, the weighted average of the representation values for each of the interests in the combined list of interests may be computed as:

$$\text{representation\_value}_{AVG,INTEREST\_j} = \frac{\sum_{i=1}^n (\text{relevancy}_i \cdot \text{representation\_value}_{i,INTEREST\_j})}{\sum_{i=1}^n \text{relevancy}_i}$$

where  $\text{relevancy}_i$  is the relevancy weight computed in step 2010 for the  $i$ -th history object,  $\text{representation\_value}_{i,INTEREST\_j}$  is the relevancy value for  $j$ -th interest, or keyword, in the  $i$ -th history object, and  $n$  is the number of relevant history objects. In a similar manner, if the representation values and thus the weighted averages of the representation values for the interests are expressed as numbers of user matches, or occurrences, in one embodiment, the history manager 56 further computes a weighted average of the total number of users for the relevant history objects, which may be computed as:

$$\text{total\_users}_{AVG} = \frac{\sum_{i=1}^n (\text{relevancy}_i \cdot \text{total\_users}_i)}{\sum_{i=1}^n \text{relevancy}_i}$$

[0128] where  $\text{relevancy}_i$  is the relevancy weight computed in step 2010 for the  $i$ -th history object,  $\text{total\_users}_i$  is the total number of users from the aggregate profile of the  $i$ -th history object, and  $n$  is the number of history objects in the list for the output time band.

[0129] Lastly, the history manager 56 outputs the weighted average of the aggregate profiles of the history objects computed in step 2016 as the historical aggregate profile data for the current location of the user 20-1 (step 2018). More specifically, in this embodiment, the historical aggregate profile data includes the combined list of interests for the aggregate profiles generated for the history objects in step 2012. In

addition, the historical aggregate profile data includes, for each interest in the combined list of interests, the weighted average of the representation values for that interest across the history objects. For each interest in the historical aggregate profile data, the weighted average of the representation values for that interests across the history objects is referred to herein as a representation value for that interest in the historical aggregate profile data. Lastly, in one embodiment, the historical aggregate profile data may include the weighted average of the total number of users across the history objects.

[0130] It should be noted that the process for generating historical aggregate profile data above is exemplary and not intended to limit the scope of the present disclosure. Other processes for generating historical aggregate profile data may be used. For example, crowds of users may formed using, for instance, the crowd formation processes described below. Crowds of users may then be tracked by storing corresponding crowd snapshots of those crowds over time, where the crowd snapshots include location data defining the locations of the corresponding crowds and profile data (e.g., anonymized user profile data) for the users in the corresponding crowds at the times at which the crowd snapshots were created and stored. Historical aggregate profile data may then be generated for the current location of, for example, the user 20-1 by combining the profile data from crowd snapshots relevant to a bounding region of a predetermined shape and size that encompasses (e.g., is centered at) the current location of the user 20-1. While not essential, for more details regarding crowd tracking, the interested reader is directed to U.S. patent application Ser. No. 12/645,539 entitled ANONYMOUS CROWD TRACKING, which was filed on Dec. 23, 2009 and has been incorporated herein by reference in its entirety.

[0131] FIGS. 19 through 25 illustrate the operation of the MAP server 12 to form crowds of users and to provide current aggregate profile data based thereon according to one embodiment of the present disclosure. Current aggregate profile data provided by the MAP server 12 using the following processes may be used to filter user profile data as described herein. For example, current aggregate profile data may be generated using the processes described below and used either by the MAP server 12 to filter user profiles of one or more of the users 20-1 through 20-N (e.g., FIG. 4) or by one or more of the mobile devices 18-1 through 18-N to filter the user profiles of the corresponding users 20-1 through 20-N (e.g., FIG. 9).

[0132] FIG. 19 is a flow chart for a spatial crowd formation process according to one embodiment of the present disclosure. Note that, in one embodiment, this process is performed in response to a request for aggregate profile data for the current location of a user from the profile manager 52 of the MAP server 12 as part of the user profile filtering process described above with respect to FIG. 4. In another embodiment, this process is performed in response to a request for aggregate profile data for the current location of a user from a corresponding mobile device as part of the user profile filtering process described above with respect to FIG. 9. However, this process is not limited thereto.

[0133] First, the crowd analyzer 58 establishes a bounding box for the crowd formation process (step 2100). Note that while a bounding box is used in this example, other geographic shapes may be used to define a bounding region for the crowd formation process (e.g., a bounding circle). In one embodiment, if crowd formation is performed in response to

a specific request for the current location of, for example, the user 20-1, the bounding box is established based on the current location of the user 20-1 such that the bounding box is a geographic area of a predetermined size centered at or otherwise encompassing the current location of the user 20-1.

[0134] The crowd analyzer 58 then creates a crowd for each individual user in the bounding box (step 2102). More specifically, the crowd analyzer 58 queries the datastore 64 of the MAP server 12 to identify users currently located within the bounding box. Then, a crowd of one user is created for each user currently located within the bounding box. Next, the crowd analyzer 58 determines the two closest crowds in the bounding box (step 2104) and determines a distance between the two crowds (step 2106). The distance between the two crowds is a distance between crowd centers of the two crowds. Note that the crowd center of a crowd of one is the current location of the user in the crowd. The crowd analyzer 58 then determines whether the distance between the two crowds is less than an optimal inclusion distance (step 2108). In this embodiment, the optimal inclusion distance is a predefined static distance. If the distance between the two crowds is less than the optimal inclusion distance, the crowd analyzer 58 combines the two crowds (step 2110) and computes a new crowd center for the resulting crowd (step 2112). The crowd center may be computed based on the current locations of the users in the crowd using a center of mass algorithm. At this point the process returns to step 2104 and is repeated until the distance between the two closest crowds is not less than the optimal inclusion distance. At that point, the crowd analyzer 58 discards any crowds with less than three users (step 2114). Note that throughout this disclosure crowds are only maintained if the crowds include three or more users. However, while three users is the preferred minimum number of users in a crowd, the present disclosure is not limited thereto. The minimum number of users in a crowd may be defined as any number greater than or equal to two users.

[0135] FIGS. 20A through 20D graphically illustrate the crowd formation process of FIG. 19 for an exemplary bounding box 108. In FIGS. 20A through 20D, crowds are noted by dashed circles, and the crowd centers are noted by cross-hairs (+). As illustrated in FIG. 20A, initially, the crowd analyzer 58 creates crowds 110 through 118 for the users in the geographic area, where, at this point, each of the crowds 110 through 118 includes one user. The current locations of the users are the crowd centers of the crowds 110 through 118. Next, the crowd analyzer 58 determines the two closest crowds and a distance between the two closest crowds. In this example, at this point, the two closest crowds are crowds 112 and 114, and the distance between the two closest crowds 112 and 114 is less than the optimal inclusion distance. As such, the two closest crowds 112 and 114 are combined by merging crowd 114 into crowd 112, and a new crowd center (+) is computed for the crowd 112, as illustrated in FIG. 20B. Next, the crowd analyzer 58 again determines the two closest crowds, which are now crowds 110 and 112. The crowd analyzer 58 then determines a distance between the crowds 110 and 112. Since the distance is less than the optimal inclusion distance, the crowd analyzer 58 combines the two crowds 110 and 112 by merging the crowd 110 into the crowd 112, and a new crowd center (+) is computed for the crowd 112, as illustrated in FIG. 20C. At this point, there are no more crowds separated by less than the optimal inclusion distance. As such, the crowd analyzer 58 discards crowds having less than three users, which in this example are crowds 116 and

118. As a result, at the end of the crowd formation process, the crowd 112 has been formed with three users, as illustrated in FIG. 20D.

[0136] FIGS. 21A through 21D illustrate a flow chart for a spatial crowd formation process according to another embodiment of the present disclosure. This process may be used to proactively form and maintain crowds of users at the MAP server 12. Subsequently, the filtered user profiles of users in one or more crowds of users at or near the current location of, for example, the user 20-1 may be combined to provide current aggregate profile data for the current location of the user 20-1, which may then be used to filter the user profile of the user 20-1 as described above.

[0137] In this embodiment, the spatial crowd formation process is triggered in response to receiving a location update for one of the users 20-1 through 20-N and is preferably repeated for each location update received for the users 20-1 through 20-N. As such, first, the crowd analyzer 58 receives a location update, or a new location, for a user (step 2200). Assume that, for this example, the location update is received for the user 20-1. In response, the crowd analyzer 58 retrieves an old location of the user 20-1, if any (step 2202). The old location is the current location of the user 20-1 prior to receiving the new location. The crowd analyzer 58 then creates a new bounding box of a predetermined size centered at the new location of the user 20-1 (step 2204) and an old bounding box of a predetermined size centered at the old location of the user 20-1, if any (step 2206). The predetermined size of the new and old bounding boxes may be any desired size. As one example, the predetermined size of the new and old bounding boxes is 40 meters by 40 meters. Note that if the user 20-1 does not have an old location (i.e., the location received in step 2200 is the first location received for the user 20-1), then the old bounding box is essentially null. Also note that while bounding “boxes” are used in this example, the bounding areas may be of any desired shape.

[0138] Next, the crowd analyzer 58 determines whether the new and old bounding boxes overlap (step 2208). If so, the crowd analyzer 58 creates a bounding box encompassing the new and old bounding boxes (step 2210). For example, if the new and old bounding boxes are 40×40 meter regions and a 1×1 meter square at the northeast corner of the new bounding box overlaps a 1×1 meter square at the southwest corner of the old bounding box, the crowd analyzer 58 may create a 79×79 meter square bounding box encompassing both the new and old bounding boxes.

[0139] The crowd analyzer 58 then determines the individual users and crowds relevant to the bounding box created in step 2210 (step 2212). The crowds relevant to the bounding box are crowds that are within or overlap the bounding box (e.g., have at least one user located within the bounding box). The individual users relevant to the bounding box are users that are currently located within the bounding box and not already part of a crowd. Next, the crowd analyzer 58 computes an optimal inclusion distance for individual users based on user density within the bounding box (step 2214). More specifically, in one embodiment, the optimal inclusion distance for individuals, which is also referred to herein as an initial optimal inclusion distance, is set according to the following equation:

$$\text{initial\_optimal\_inclusion\_dist} = a \cdot \sqrt{\frac{A_{\text{BoundingBox}}}{\text{number\_of\_users}}},$$

where  $a$  is a number between 0 and 1,  $A_{\text{BoundingBox}}$  is an area of the bounding box, and  $\text{number\_of\_users}$  is the total number of users in the bounding box. The total number of users in the bounding box includes both individual users that are not already in a crowd and users that are already in a crowd. In one embodiment,  $a$  is  $\frac{2}{3}$ .

[0140] The crowd analyzer 58 then creates a crowd for each individual user within the bounding box that is not already included in a crowd and sets the optimal inclusion distance for the crowds to the initial optimal inclusion distance (step 2216). At this point, the process proceeds to FIG. 21B where the crowd analyzer 58 analyzes the crowds relevant to the bounding box to determine whether any of the crowd members (i.e., users in the crowds) violate the optimal inclusion distance of their crowds (step 2218). Any crowd member that violates the optimal inclusion distance of his or her crowd is then removed from that crowd (step 2220). The crowd analyzer 58 then creates a crowd of one user for each of the users removed from their crowds in step 2220 and sets the optimal inclusion distance for the newly created crowds to the initial optimal inclusion distance (step 2222).

[0141] Next, the crowd analyzer 58 determines the two closest crowds for the bounding box (step 2224) and a distance between the two closest crowds (step 2226). The distance between the two closest crowds is the distance between the crowd centers of the two closest crowds. The crowd analyzer 58 then determines whether the distance between the two closest crowds is less than the optimal inclusion distance of a larger of the two closest crowds (step 2228). If the two closest crowds are of the same size (i.e., have the same number of users), then the optimal inclusion distance of either of the two closest crowds may be used. Alternatively, if the two closest crowds are of the same size, the optimal inclusion distances of both of the two closest crowds may be used such that the crowd analyzer 58 determines whether the distance between the two closest crowds is less than the optimal inclusion distances of both of the two closest crowds. As another alternative, if the two closest crowds are of the same size, the crowd analyzer 58 may compare the distance between the two closest crowds to an average of the optimal inclusion distances of the two closest crowds.

[0142] If the distance between the two closest crowds is less than the optimal inclusion distance, the two closest crowds are combined or merged (step 2230), and a new crowd center for the resulting crowd is computed (step 2232). Again, a center of mass algorithm may be used to compute the crowd center of a crowd. In addition, a new optimal inclusion distance for the resulting crowd is computed (step 2234). In one embodiment, the new optimal inclusion distance for the resulting crowd is computed as:

$$\text{average} = \frac{1}{n+1} \cdot \left( \text{initial\_optimal\_inclusion\_dist} + \sum_{i=1}^n d_i \right),$$

-continued

$$\text{optimal\_inclusion\_dist} = \text{average} + \sqrt{\left( \frac{1}{n} \cdot \sum_{i=1}^n (d_i - \text{average})^2 \right)},$$

where  $n$  is the number of users in the crowd and  $d_i$  is a distance between the  $i$ th user and the crowd center. In other words, the new optimal inclusion distance is computed as the average of the initial optimal inclusion distance and the distances between the users in the crowd and the crowd center plus one standard deviation.

[0143] At this point, the crowd analyzer 58 determines whether a maximum number of iterations have been performed (step 2236). The maximum number of iterations is a predefined number that ensures that the crowd formation process does not indefinitely loop over steps 2218 through 2234 or loop over steps 2218 through 2234 more than a desired maximum number of times. If the maximum number of iterations has not been reached, the process returns to step 2218 and is repeated until either the distance between the two closest crowds is not less than the optimal inclusion distance of the larger crowd or the maximum number of iterations has been reached. At that point, the crowd analyzer 58 discards crowds with less than three users, or members (step 2238) and the process ends.

[0144] Returning to step 2208 in FIG. 21A, if the new and old bounding boxes do not overlap, the process proceeds to FIG. 21C and the bounding box to be processed is set to the old bounding box (step 2240). In general, the crowd analyzer 58 then processes the old bounding box in much the same manner as described above with respect to steps 2212 through 2238. More specifically, the crowd analyzer 58 determines the individual users and crowds relevant to the bounding box (step 2242). The crowds relevant to the bounding box are crowds that are within or overlap the bounding box (e.g., have at least one user located within the bounding box). The individual users relevant to the bounding box are users that are currently located within the bounding box and not already part of a crowd. Next, the crowd analyzer 58 computes an optimal inclusion distance for individual users based on user density within the bounding box (step 2244). More specifically, in one embodiment, the optimal inclusion distance for individuals, which is also referred to herein as an initial optimal inclusion distance, is set according to the following equation:

$$\text{initial\_optimal\_inclusion\_dist} = a \cdot \sqrt{\frac{A_{\text{BoundingBox}}}{\text{number\_of\_users}}},$$

where  $a$  is a number between 0 and 1,  $A_{\text{BoundingBox}}$  is an area of the bounding box, and  $\text{number\_of\_users}$  is the total number of users in the bounding box. The total number of users in the bounding box includes both individual users that are not already in a crowd and users that are already in a crowd. In one embodiment,  $a$  is  $\frac{2}{3}$ .

[0145] The crowd analyzer 58 then creates a crowd of one user for each individual user within the bounding box that is not already included in a crowd and sets the optimal inclusion distance for the crowds to the initial optimal inclusion distance (step 2246). At this point, the crowd analyzer 58 analyzes the crowds for the bounding box to determine whether

any crowd members (i.e., users in the crowds) violate the optimal inclusion distance of their crowds (step 2248). Any crowd member that violates the optimal inclusion distance of his or her crowd is then removed from that crowd (step 2250). The crowd analyzer 58 then creates a crowd of one user for each of the users removed from their crowds in step 2250 and sets the optimal inclusion distance for the newly created crowds to the initial optimal inclusion distance (step 2252).

[0146] Next, the crowd analyzer 58 determines the two closest crowds in the bounding box (step 2254) and a distance between the two closest crowds (step 2256). The distance between the two closest crowds is the distance between the crowd centers of the two closest crowds. The crowd analyzer 58 then determines whether the distance between the two closest crowds is less than the optimal inclusion distance of a larger of the two closest crowds (step 2258). If the two closest crowds are of the same size (i.e., have the same number of users), then the optimal inclusion distance of either of the two closest crowds may be used. Alternatively, if the two closest crowds are of the same size, the optimal inclusion distances of both of the two closest crowds may be used such that the crowd analyzer 58 determines whether the distance between the two closest crowds is less than the optimal inclusion distances of both of the two closest crowds. As another alternative, if the two closest crowds are of the same size, the crowd analyzer 58 may compare the distance between the two closest crowds to an average of the optimal inclusion distances of the two closest crowds.

[0147] If the distance between the two closest crowds is less than the optimal inclusion distance, the two closest crowds are combined or merged (step 2260), and a new crowd center for the resulting crowd is computed (step 2262). Again, a center of mass algorithm may be used to compute the crowd center of a crowd. In addition, a new optimal inclusion distance for the resulting crowd is computed (step 2264). As discussed above, in one embodiment, the new optimal inclusion distance for the resulting crowd is computed as:

$$\text{average} = \frac{1}{n+1} \cdot \left( \text{initial\_optimal\_inclusion\_dist} + \sum_{i=1}^n d_i \right),$$

$$\text{optimal\_inclusion\_dist} = \text{average} + \sqrt{\left( \frac{1}{n} \cdot \sum_{i=1}^n (d_i - \text{average})^2 \right)},$$

where n is the number of users in the crowd and  $d_i$  is a distance between the  $i$ th user and the crowd center. In other words, the new optimal inclusion distance is computed as the average of the initial optimal inclusion distance and the distances between the users in the crowd and the crowd center plus one standard deviation.

[0148] At this point, the crowd analyzer 58 determines whether a maximum number of iterations have been performed (step 2266). If the maximum number of iterations has not been reached, the process returns to step 2248 and is repeated until either the distance between the two closest crowds is not less than the optimal inclusion distance of the larger crowd or the maximum number of iterations has been reached. At that point, the crowd analyzer 58 discards crowds with less than three users, or members (step 2268). The crowd analyzer 58 then determines whether the crowd formation process for the new and old bounding boxes is done (step 2270). In other words, the crowd analyzer 58 determines

whether both the new and old bounding boxes have been processed. If not, the bounding box is set to the new bounding box (step 2272), and the process returns to step 2242 and is repeated for the new bounding box. Once both the new and old bounding box have been processed, the crowd formation process ends.

[0149] FIGS. 22A through 22D graphically illustrate the crowd formation process of FIGS. 21A through 21D for a scenario where the crowd formation process is triggered by a location update for a user having no old location. In this scenario, the crowd analyzer 58 creates a new bounding box 120 for the new location of the user, and the new bounding box 120 is set as the bounding box to be processed for crowd formation. Then, as illustrated in FIG. 22A, the crowd analyzer 58 identifies all individual users currently located within the bounding box 120 and all crowds located within or overlapping the bounding box. In this example, crowd 122 is an existing crowd relevant to the bounding box 120. Crowds are indicated by dashed circles, crowd centers are indicated by cross-hairs (+), and users are indicated as dots. Next, as illustrated in FIG. 22B, the crowd analyzer 58 creates crowds 124 through 128 of one user for the individual users, and the optional inclusion distances of the crowds 124 through 128 are set to the initial optimal inclusion distance. As discussed above, the initial optimal inclusion distance is computed by the crowd analyzer 58 based on a density of users within the bounding box 120.

[0150] The crowd analyzer 58 then identifies the two closest crowds 124 and 126 in the bounding box 120 and determines a distance between the two closest crowds 124 and 126. In this example, the distance between the two closest crowds 124 and 126 is less than the optimal inclusion distance. As such, the two closest crowds 124 and 126 are merged and a new crowd center and new optimal inclusion distance are computed, as illustrated in FIG. 22C. The crowd analyzer 58 then repeats the process such that the two closest crowds 124 and 128 in the bounding box 120 are again merged, as illustrated in FIG. 22D. At this point, the distance between the two closest crowds 122 and 124 is greater than the appropriate optimal inclusion distance. As such, the crowd formation process is complete.

[0151] FIGS. 23A through 23F graphically illustrate the crowd formation process of FIGS. 21A through 21D for a scenario where the new and old bounding boxes overlap. As illustrated in FIG. 23A, a user moves from an old location to a new location, as indicated by an arrow. The crowd analyzer 58 receives a location update for the user giving the new location of the user. In response, the crowd analyzer 58 creates an old bounding box 130 for the old location of the user and a new bounding box 132 for the new location of the user. Crowd 134 exists in the old bounding box 130, and crowd 136 exists in the new bounding box 132.

[0152] Since the old bounding box 130 and the new bounding box 132 overlap, the crowd analyzer 58 creates a bounding box 138 that encompasses both the old bounding box 130 and the new bounding box 132, as illustrated in FIG. 23B. In addition, the crowd analyzer 58 creates crowds 140 through 146 for individual users currently located within the bounding box 138. The optimal inclusion distances of the crowds 140 through 146 are set to the initial optimal inclusion distance computed by the crowd analyzer 58 based on the density of users in the bounding box 138.

[0153] Next, the crowd analyzer 58 analyzes the crowds 134, 136, and 140 through 146 to determine whether any

members of the crowds **134**, **136**, and **140** through **146** violate the optimal inclusion distances of the crowds **134**, **136**, and **140** through **146**. In this example, as a result of the user leaving the crowd **134** and moving to his new location, both of the remaining members of the crowd **134** violate the optimal inclusion distance of the crowd **134**. As such, the crowd analyzer **58** removes the remaining users from the crowd **134** and creates crowds **148** and **150** of one user each for those users, as illustrated in FIG. 23C.

[0154] The crowd analyzer **58** then identifies the two closest crowds in the bounding box **138**, which in this example are the crowds **144** and **146**. Next, the crowd analyzer **58** computes a distance between the two crowds **144** and **146**. In this example, the distance between the two crowds **144** and **146** is less than the initial optimal inclusion distance and, as such, the two crowds **144** and **146** are combined. In this example, crowds are combined by merging the smaller crowd into the larger crowd. Since the two crowds **144** and **146** are of the same size, the crowd analyzer **58** merges the crowd **146** into the crowd **144**, as illustrated in FIG. 23D. A new crowd center and new optimal inclusion distance are then computed for the crowd **144**.

[0155] At this point, the crowd analyzer **58** repeats the process and determines that the crowds **136** and **142** are now the two closest crowds. In this example, the distance between the two crowds **136** and **142** is less than the optimal inclusion distance of the larger of the two crowds **136** and **142**, which is the crowd **136**. As such, the crowd **142** is merged into the crowd **136** and a new crowd center and optimal inclusion distance are computed for the crowd **136**, as illustrated in FIG. 23E. At this point, there are no two crowds closer than the optimal inclusion distance of the larger of the two crowds. As such, the crowd analyzer **58** discards any crowds having less than three members, as illustrated in FIG. 23F. In this example, the crowds **140**, **144**, **148**, and **150** have less than three members and are therefore removed. The crowd **136** has three or more members and, as such, is not removed. At this point, the crowd formation process is complete.

[0156] FIGS. 24A through 24E graphically illustrate the crowd formation process of FIGS. 21A through 21D in a scenario where the new and old bounding boxes do not overlap. As illustrated in FIG. 24A, in this example, the user moves from an old location to a new location. The crowd analyzer **58** creates an old bounding box **152** for the old location of the user and a new bounding box **154** for the new location of the user. Crowds **156** and **158** exist in the old bounding box **152**, and crowd **160** exists in the new bounding box **154**. In this example, since the old and new bounding boxes **152** and **154** do not overlap, the crowd analyzer **58** processes the old and new bounding boxes **152** and **154** separately.

[0157] More specifically, as illustrated in FIG. 24B, as a result of the movement of the user from the old location to the new location, the remaining users in the crowd **156** no longer satisfy the optimal inclusion distance for the crowd **156**. As such, the remaining users in the crowd **156** are removed from the crowd **156**, and crowds **162** and **164** of one user each are created for the removed users as shown in FIG. 24C. In this example, no two crowds in the old bounding box **152** are close enough to be combined. As such, processing of the old bounding box **152** is complete, and the crowd analyzer **58** proceeds to process the new bounding box **154**.

[0158] As illustrated in FIG. 24D, processing of the new bounding box **154** begins by the crowd analyzer **58** creating a

crowd **166** of one user for the user. The crowd analyzer **58** then identifies the crowds **160** and **166** as the two closest crowds in the new bounding box **154** and determines a distance between the two crowds **160** and **166**. In this example, the distance between the two crowds **160** and **166** is less than the optimal inclusion distance of the larger crowd, which is the crowd **160**. As such, the crowd analyzer **58** combines the crowds **160** and **166** by merging the crowd **166** into the crowd **160**, as illustrated in FIG. 24E. A new crowd center and new optimal inclusion distance are then computed for the crowd **160**. At this point, the crowd formation process is complete.

[0159] FIG. 25 illustrates a process for generating aggregate profile data, or more specifically current aggregate profile data, for one or more crowds in response to a request for aggregate profile data for a current location of a user according to one embodiment of the present disclosure. First, a request for aggregate profile data for a current location of a user is received by the crowd analyzer **58** (step 2300). In this example, the request is for aggregate profile data for the current location of the user **20-1**. In one embodiment, the request is received from the profile manager **52** as part of the profile filtering process of FIG. 4. In another embodiment, the request is received from the mobile device **18-1** of the user **20-1** as part of the profile filtering process of FIG. 9. Next, the crowd analyzer **58** establishes a bounding box for the request (step 2302). The bounding box is a geographic area of a predetermined size centered at or otherwise encompassing the current location of the user **20-1**. While a bounding “box” is used in this example, a bounding region of any desired shape and size may be used.

[0160] Next, the crowd analyzer **58** identifies one or more crowds that are relevant to the bounding box established for the request (step 2304). The one or more crowds that are relevant to the bounding box are preferably crowds located within or overlapping the bounding box at the current time. After the crowd analyzer **58** has identified the one or more relevant crowds, the relevant crowds are passed to the aggregation engine **60**. The aggregation engine **60** then selects a next crowd to process from the one or more relevant crowds, which for the first iteration is the first relevant crowd (step 2306). The aggregation engine **60** then generates an aggregate profile for the crowd based on a comparison of the filtered user profiles of the users in the crowd to one another (step 2308). Note that in this embodiment it is assumed that filtered user profiles are created and stored for all of the users **20-1** through **20-N**. However, the present disclosure is not limited thereto. In an alternative embodiment, profile filtering may not be performed for some of the users **20-1** through **20-N**. In this case, the user profiles of any users in the crowd for which filtered user profiles have not been created and stored are used when generating the aggregate profile for the crowd.

[0161] The aggregate profile for the crowd includes an aggregate list of interests of the users in the crowd and representation values for the interests in the aggregate list of interests for the crowd. As discussed above, in one embodiment, the interests are expressed as keywords in a number of profile categories. In one embodiment, for each interest, or keyword, the representation value for the interest is a number of user matches, or occurrences, of the interest across all of the filtered user profiles of the users in the crowd. In another embodiment, for each interest, or keyword, the representation value for the interest is a ratio of a number of user matches, or

occurrences, of the interest across the filtered user profiles of all of the users in the crowd over a total number of users in the crowd.

[0162] Once the aggregate profile of the crowd is generated, the aggregation engine 60 determines whether there are more crowds to process (step 2310). If so, the process returns to step 2306 and is repeated for the next crowd. Once aggregate profiles have been generated for all of the relevant crowds, the aggregate profiles for the crowds are combined (step 2312). The aggregate profiles may be combined by generating a combined list of interests that includes all of the interests from the aggregate profiles of the crowds. Then, for each interest in the combined list of interests, the representation values for that interest in the aggregate profiles are combined to provide a combined representation value for the interest. For example, if the representation values are expressed as numbers of user matches, then the number of user matches for a particular interest in each of the aggregate profiles for the crowds that include that interest may be summed to provide the combined representation value for that interest. As another example, if the representation values are expressed as ratios of the user matches to total numbers of users, then the ratios for a particular interest in each of the aggregate profiles for the crowds that include that interest may be averaged to provide the combined representation value for that interest.

[0163] Lastly, the aggregation engine 60 outputs the combined aggregate profile for the one or more crowds as the current aggregate profile data for the current location of the user 20-1 (step 2314). The combined aggregate profile includes the combined list of interests generated from the aggregate profiles for the one or more relevant crowds and corresponding combined representation values for the interests in the combined list of interests. The combined representation values may also be referred to herein as representation values for the corresponding interests in the current aggregate profile data.

[0164] Before proceeding, an alternative embodiment should be discussed. In an alternative embodiment, rather than establishing a bounding box for identifying the one or more relevant crowds, the crowd analyzer 58 may identify a current crowd in which the user 20-1 is located or a current crowd in which the user 20-1 is to be located based on the current location of the user 20-1. This identified crowd may then be the only relevant crowd considered for purposes of current aggregate profile generation. The aggregation engine 60 may then generate an aggregate profile for the identified crowd and output the aggregate profile of the identified crowd as the aggregate profile data for the current location of the user 20-1. In this alternative embodiment, steps 2302, 2306, 2310, and 2312 of FIG. 25 would not be needed. Also, in step 2314, the aggregate profile of the crowd would be output as the aggregate profile data for the current location of the user 20-1.

[0165] In another alternative embodiment, rather than utilizing crowds to generate current aggregate profile data, the MAP server 12 may establish a bounding region of a predetermined shape and size that is centered at or that otherwise encompasses the current location of the user 20-1. The MAP server 12 may then identify all of the other users 20-2 through 20-N that are currently located within the bounding region and aggregate the filtered user profiles of the identified users to provide the current aggregate profile data for the current location of the user 20-1.

[0166] In the embodiments described above, user profile filtering is performed at either the MAP server 12 or the corresponding mobile devices 18-1 through 18-N prior to utilizing the user profiles at the MAP server 12 to provide aggregate profile data. FIG. 26 illustrates an embodiment where filtering is performed on aggregate profile data after receiving the aggregate profile data from the MAP server 12 according to another embodiment of the present disclosure. Using the MAP application 32-1 of the mobile device 18-1 as an example, first, the MAP application 32-1 sends a request to the MAP server 12 for current and/or historical aggregate profile data (step 2400). In response, the MAP server 12 generates the requested aggregate profile data. Note that in this embodiment, the MAP server 12 preferably does not create or store filtered user profiles for the users 20-1 through 20-N. As such, the user profiles, rather than the filtered user profiles, of the users 20-1 through 20-N are used to generate aggregate profile data. The MAP server 12 then returns the requested aggregate profile data to the MAP application 32-1 of the mobile device 18-1.

[0167] At this point, the MAP application 32-1 receives the requested aggregate profile data from the MAP server 12 (step 2402). Next, the MAP application 32-1 may obtain additional information, if any, for a corresponding location for which the aggregate profile data was requested in a manner similar to that described above (step 2404). In one embodiment, the aggregate profile data is for the current location of the user 20-1 and, as such, the additional information is also for the current location of the user 20-1. The MAP application 32-1 then filters aggregate profile data based on the aggregate profile data itself, the additional information, and one or more filtering rules (step 2406). Lastly, the MAP application 32-1 utilizes the filtered aggregate profile data (step 2408). For example, the MAP application 32-1 may present the filtered aggregate profile data to the user 20-1. As another example, the MAP application 32-1 may utilize the filtered aggregate profile data to perform a desired function.

[0168] FIG. 27 illustrates step 2408 of FIG. 26 in more detail according to one embodiment of the present disclosure. In order to filter the aggregate profile data, the MAP application 32-1 first determines a cut-off value (step 2500). The cut-off value may be determined in the same manner as that described above with respect to step 1400 of FIG. 8. For example, the cut-off value may be a function of the highest representation value of the representation values for the interests in the aggregate profile data. Further, if the location for which the aggregate profile data has been obtained is the current location of the user 20-1, the cut-off value may also be a function of an amount of time that the user 20-1 has been at the same location, a frequency at which the user 20-1 visits the location, a number of times that the user 20-1 has visited the location, or a combination thereof.

[0169] After determining the cut-off value, filtered aggregate profile data is initialized (step 2502). The filtered aggregate profile is initialized by creating a version of the aggregate profile data having no interests. Then, the representation value for the next interest in the aggregate profile data is obtained (step 2504). A determination is then made as to whether the representation value is greater than or equal to the cut-off value (step 2506). If not, the process proceeds to step 2510. Otherwise, the interest is added to the filtered aggregate profile data (step 2508). At this point, whether proceeding from step 2506 or 2508, a determination is made as to whether



the last interest in the aggregate profile data has been processed (step 2510). If not, the process returns to step 2504 and is repeated.

[0170] Once all of the interests in the aggregate profile data have been processed, the filtered aggregate profile data may be further filtered based on the additional information obtained for the corresponding location in a manner similar to that described above (step 2512). Note that step 2512 is optional. In addition, in this embodiment, the filtered aggregate profile data may be further filtered based on one or more situational awareness rules in the manner described above (step 2514). Note that like step 2512, step 2514 is also optional.

[0171] The processes described above enable either user profile filtering or aggregate profile filtering to control what profile information is revealed through aggregate profile data depending on the environments in which the users 20-1 through 20-N are located. However, a user may attempt to thwart this control mechanism by, for example, altering his user profile to include a particular interest in an attempt to bypass filtering rules configured to hide that there are one or more other users with that interest nearby. For example, user A may define a filtering rule such that interest A in user A's user profile is always filtered, or hidden, regardless of the amount of time that user A has remained at the same location unless the aggregate profile data for user A's current location indicates that there is at least one other user at or near user A's current location having a user profile that includes interest A. User B may attempt to bypass user A's rule by altering his user profile to include interest A even though user B does not in fact have an interest in interest A in an attempt to have user A's interests in interest A revealed via the aggregate profile data for the current location.

[0172] The following discussion describes processes by which such bypass attempts can be thwarted based on tracking ages of interests in the user profiles, and thus the filtered user profiles, of the users 20-1 through 20-N. In order to track the ages of the interests in the user profiles, and thus filtered user profiles, of the users 20-1 through 20-N, the profile manager 52 stores an age timestamp for each interest in the user profiles, and thus filtered user profiles, of the users 20-1 through 20-N. Using the user 20-1 as an example, the profile manager 52 initially sets age timestamps for the interests in the user profile of the user 20-1 to a current time at the time when the profile manager 52 first obtains the user profile of the user 20-1. Thereafter, each time an interest is added to the user profile of the user 20-1, an age timestamp corresponding to the time at which the interest was added is created and stored for that interest in the user profile of the user 20-1. Similarly, each time an interest is modified in the user profile of the user 20-1, the age timestamp for that interest is updated to reflect the time at which the interest was modified. Using the age timestamps, ages of the interests in the user profiles of the users 20-1 through 20-N can be determined. Note that the age timestamps for the interests in the user profiles of the users 20-1 through 20-N are also stored in the filtered user profiles of the users 20-1 through 20-N for the interests remaining after filtering.

[0173] In one embodiment, the ages of the interests in the filtered user profiles of the users 20-1 through 20-N are used when generating aggregate profile data at the MAP server 12 to be used in the profile filtering process of FIG. 4, FIG. 9, or FIG. 26. More specifically, when generating historical aggregate profile data according to the process of FIG. 18, in step

2012, interests in the filtered user profiles of the anonymous user records stored in the history object that have ages that are less than a predetermined threshold age (e.g., 1 day, 1 week, or the like) are ignored, or not considered, when generating the aggregate profile for the history object. In this manner, interests that were recently added or modified at the time of creating the history object do not contribute to the aggregate profile generated for the history object and thus the historical aggregate profile data.

[0174] In an alternative embodiment, when generating historical aggregate profile data according to the process of FIG. 18, in step 2012, interests in the filtered user profiles of the anonymous user records stored in the history object that are relevant to one or more filtering rules of the user for which the historical aggregate profile data is being generated (e.g., the user whose user profile is to be filtered based on the historical aggregate profile data) and that have ages that are less than a predetermined threshold age are ignored, or not considered, when generating the aggregate profile for the history object. An interest is relevant to a filtering rule when that interest, if included in the historical aggregate profile data, would be utilized for determining whether the filtering rule is satisfied. In this manner, interests recently added or modified at the time of creating and storing the history object do not contribute to the historical aggregate profile data.

[0175] In a similar manner, when generating current aggregate profile data according to the process of FIG. 25, in step 2308, interests in the filtered user profiles of the users in the crowd that have ages that are less than a predetermined threshold age (e.g., 1 day, 1 week, or the like) are ignored, or not considered, when generating the aggregate profile for the crowd. In this manner, interests recently added or modified in the filtered user profiles of the users in the crowd do not contribute to the aggregate profile data for the crowd. As such, any attempt to thwart the filtering rules defined for filtering a user profile of a user cannot be bypassed by another user in a relevant crowd by that other user adding interests to his user profile.

[0176] In an alternative embodiment, when generating current aggregate profile data according to the process of FIG. 25, in step 2308, interests in the filtered user profiles of the users in the crowd that are relevant to one or more filtering rules of the user for which the current aggregate profile data is being generated (e.g., the user whose user profile is to be filtered based on the current aggregate profile data) and that have ages that are less than a predetermined threshold age are ignored, or not considered, when generating the aggregate profile for the crowd. An interest is relevant to a filtering rule when that interest, if included in the current aggregate profile data, would be utilized for determining whether the filtering rule is satisfied. In this manner, interests recently added or modified in the filtered user profiles of the users in the crowd that would alter the filtering of the user profile of the user do not contribute to the aggregate profile data for the crowd. As such, any attempt to thwart the filtering rules defined for filtering the user profile of the user cannot be bypassed by another user in a relevant crowd by that other user adding interests to his user profile.

[0177] In order to illustrate some of the concepts described above, the following exemplary use cases are provided. However, these use cases are exemplary and are not intended to limit the scope of the present disclosure.

Ted at a Conference:

[0178] 1. Ted is at a business conference for restaurant equipment. Since Ted still likes to share information about the

fact that he is involved with a pizza franchise and see how many other conference attendees are as well, he leaves his MAP application on his mobile device running. Normally, this might cause some of Ted's more personal information from his user profile to be shown in the aggregate profile for the conference but he has set up rules around the more personal parts of his profile so that only people that share his love for UNC can see that in the aggregate profile for the conference.

2. Ted enters the building where the conference is being held and the MAP application on his mobile device (e.g., mobile phone) sends an update to the server for his current location, his current situation, and, if not already sent, his user profile.

**[0179]** Determining the Appropriate Rules and Situations:

**[0180]** 1. The MAP server gathers information about the event associated with that location and the current aggregate profile for the location and finds that it matches the criteria that Ted has identified as a "work" situation, which tells the MAP server to send an alert to Ted's MAP client to see if he wants to go into a Blend mode to hide his more personal and unrelated information.

3. Ted's MAP application causes his mobile device to beep at him, and he pulls his mobile device out and sees that it is asking if he wants to enter Blend mode. Ted tells the system to remember his preference for the conference event and that he does want to switch to Blend mode.

**[0181]** Blend Mode:

**[0182]** 1. In Blend mode, Ted's user profile will be filtered such that his filtered user profile includes only those interests that are related to the aggregate profile data for his current location and the attributes (i.e., additional information) obtained for the event. In this case, the additional information for Ted's current location indicates that he is currently at a business type event related to the food industry and equipment.

**[0183]** 2. Based on the aggregate profile data and the additional information obtained for his current location, Ted's user profile is filtered such that his filtered user profile includes only those portions dealing with the franchises he is involved with and the fact that he is involved mainly with pizza restaurant equipment and portions dealing with some related awards and articles referenced in his user profile. All of these portions of Ted's user profile are shown as part of the conference aggregate profile, but his love for college basketball, and particularly UNC, are not shared since they are not strongly represented by the location metadata or aggregate profile.

Ted on the Town:

**[0184]** 1. After a long day at the conference, Ted has seen some friendly faces and made some new friends and they go out on the town. Ted's MAP application on his mobile device updates the MAP server on his current location, and the MAP server determines that it is not associated with the conference anymore. As such, operation switches Ted's profile filtering setting to a Sharing mode.

2. Ted and his friends find a nice local pub and grab some food and drinks.

**[0185]** Sharing Mode:

**[0186]** 1. Ted's MAP application is constantly updating the MAP server on his location. Even though Ted

is in Sharing mode now, when he is first associated with a location, his profile is filtered as if he was in the Blend mode.

**[0187]** 2. As Ted spends more time at a location, the filtering of his user profile becomes less restrictive such that more of his user profile begins to show in the aggregate profile data for the location. By the time they have spent a half hour at the restaurant, some of Ted's interests in bowling and college sports are starting to appear in the aggregate profile data for the location. After another half an hour or so, Ted's interest in college basketball appears in the aggregate profile data.

3. After about an hour has passed, everyone pays their part of the bill, and they all use their MAP-enabled mobile devices to find some fun places to hop around to, but don't end up staying for very long at any of the places.

**[0188]** Restarting the Loosening Process at New Locations:

**[0189]** 1. The system continues to get updates about Ted's location. Once Ted's location is no longer associated with the restaurant, his profile restarts the loosening process, meaning that his profile is once again restricted to the Blend mode view.

4. Finally, Ted and the three most stalwart of the crowd end up at an all night diner where they talk late into the night and the next morning. Soon they are the majority of who is left in the diner such that the aggregate profile data for the diner begins to show some more personal information about the remaining users. It comes to light via conversation that there are a few rather vehement Duke fans among the remaining group. Ted is a staunch UNC fan, but having just met these individuals and considering that these individuals could be good potential business contacts, the general level of intoxication, and the fact that UNC recently lost to Duke, Ted would rather not let it be known that he is a UNC fan.

**[0190]** Automatic Screening through Rules:

**[0191]** 1. The system had identified that there at least two Duke fans present based off of the user profiles of two of the four people present and had followed the rule that Ted had set for his UNC affiliation that his interest as a UNC fan is to be revealed only if the ratio of UNC to Duke fans was at least even. In addition or alternatively, aggregate profile data may be generated such that Ted's interest as a UNC fan would only be revealed in the aggregate profile data generated for other UNC fans but not for Duke fans unless the ratio of UNC to Duke fans was at least even.

5. Ted heads to the bathroom, but his new business contacts are a little curious about the lukewarm response that their UNC bashing jokes got from Ted. They have Nosy Ned pull out his mobile device and use his MAP application to look at the aggregate profile data for the diner. Ned doesn't see any UNC fan representation in the diner's aggregate profile data. Ned does not have a Duke fan affiliation in his profile so taking it a step further he tries falsely adding that he is a UNC fan to his user profile in order to see if he can potentially sidestep rules Ted may have set upon his UNC fan information sharing if Ted actually is a UNC fan.

**[0192]** Profile Attribute Aging to Protect against Interest Falsification:

**[0193]** 1. The system acknowledges the update to Ned's profile. However, since the system has all changes to Ned's user profile time-stamped, it quickly

identifies that this change was very recently made and therefore will not acknowledge it as qualifying Ned for any rules requiring this newly added interest until this newly added interest has been in the Ned's user profile for a sufficient amount of time.

6. As Ted comes back out he checks his MAP application but he does not see Ned's update. Therefore, to Ted, the aggregate profile contains no UNC fans.

**[0194]** Anti-Fishing and Anti-Reverse Persecution Hiding:

**[0195]** 1. The system recognizes that Ned is not allowed to see Ted's UNC fan interest. So, in order to protect Ned from being identified as someone who recently changed his profile, potentially in order to try and see information he could not before, and to keep Ted from mistakenly trying to find someone at the location who shares his UNC fan interest, the system hides Ned's profile update from Ted until Ned's update is old enough to be counted toward Ted's rules.

**[0196]** 2. The system allows all other users who have no rules excluding Ned from seeing their UNC fan attribute, even if it is simply allowing Ned to see that they do not have the UNC attribute. As such, Ned's other two Duke fan friends are able to see a UNC fan appear in the aggregate profile.

Ted Over Time:

**[0197]** 1. The next morning at the second day of the conference, Ted arrives bleary-eyed and forgets that he should switch to Blend mode, but the system recognizes the conference event when it receives the update to Ted's location and therefore automatically switches to Blend mode again.

2. Ted doesn't stay out as late the next night but does meet some of his friends including the three Duke fans from the evening before for drinks and food at a new restaurant.

**[0198]** Crowd Awareness

**[0199]** 1. The system identifies that Ted is with a similar set of people as from the other night so it adjusts the interval at which the restrictions on his profile are loosened in order to confuse any attempts the other users might make to tie particular interests to him.

3. Ted continues to meet with friends including the three Duke fans over the rest of the week while he is attending the conference and they start to frequent one or two locations.

**[0200]** Location Awareness

**[0201]** 1. The system identifies that Ted is frequenting certain locations and therefore once again adjusts the loosening interval for his profile in order to keep people at those locations from being able to tie particular changes in the aggregate profile to him.

**[0202]** FIG. 28 is a block diagram of the MAP server 12 according to one embodiment of the present disclosure. As illustrated, the MAP server 12 includes a controller 168 connected to memory 170, one or more secondary storage devices 172, and a communication interface 174 by a bus 176 or similar mechanism. The controller 168 is a microprocessor, digital Application Specific Integrated Circuit (ASIC), Field Programmable Gate Array (FPGA), or the like. In this embodiment, the controller 168 is a microprocessor, and the application layer 40, the business logic layer 42, and the object mapping layer 62 (FIG. 2) are implemented in software and stored in the memory 170 for execution by the controller 168. Further, the datastore 64 (FIG. 2) may be implemented in the one or more secondary storage devices 172. The second-

ary storage devices 172 are digital data storage devices such as, for example, one or more hard disk drives. The communication interface 174 is a wired or wireless communication interface that communicatively couples the MAP server 12 to the network 28 (FIG. 1). For example, the communication interface 174 may be an Ethernet interface, local wireless interface such as a wireless interface operating according to one of the suite of IEEE 802.11 standards, or the like.

**[0203]** FIG. 29 is a block diagram of the mobile device 18-1 according to one embodiment of the present disclosure. This discussion is equally applicable to the other mobile devices 18-2 through 18-N. As illustrated, the mobile device 18-1 includes a controller 178 connected to memory 180, a communication interface 182, one or more user interface components 184, and the location function 36-1 by a bus 186 or similar mechanism. The controller 178 is a microprocessor, digital ASIC, FPGA, or the like. In this embodiment, the controller 178 is a microprocessor, and the MAP client 30-1, the MAP application 32-1, and the third-party applications 34-1 are implemented in software and stored in the memory 180 for execution by the controller 178. In this embodiment, the location function 36-1 is a hardware component such as, for example, a GPS receiver. The communication interface 182 is a wireless communication interface that communicatively couples the mobile device 18-1 to the network 28 (FIG. 1). For example, the communication interface 182 may be a local wireless interface such as a wireless interface operating according to one of the suite of IEEE 802.11 standards, a mobile communications interface such as a cellular telecommunications interface, or the like. The one or more user interface components 184 include, for example, a touchscreen, a display, one or more user input components (e.g., a keypad), a speaker, or the like, or any combination thereof.

**[0204]** Those skilled in the art will recognize improvements and modifications to the preferred embodiments of the present invention. All such improvements and modifications are considered within the scope of the concepts disclosed herein and the claims that follow.

What is claimed is:

1. A computer-implemented method comprising:
  - obtaining aggregate profile data for a current location of a user; and
  - filtering a user profile of the user based on the aggregate profile data for the current location of the user to thereby provide a filtered user profile for the user.
2. The method of claim 1 wherein the aggregate profile data comprises historical aggregate profile data.
3. The method of claim 1 wherein the aggregate profile data comprises current aggregate profile data.
4. The method of claim 1 wherein filtering the user profile of the user comprises, for each interest of a plurality of interests in the user profile of the user, filtering the interest if a representation value for a corresponding interest in the aggregate profile data is less than a cut-off value.
5. The method of claim 4 wherein for each interest of the plurality of interests in the aggregate profile data, the aggregate profile data comprises a representation value for the interest that defines a degree to which the interest is expressed in a plurality of user profiles of a plurality of users that contributed to the aggregate profile data.
6. The method of claim 4 wherein the cut-off value is a predetermined, static value.
7. The method of claim 4 wherein the cut-off value is a dynamic value.

8. The method of claim 4 wherein the cut-off value is a function of an amount of time that the user has been located at the same location.

9. The method of claim 8 wherein the amount of time that the user has been located at the same location is an amount of time that the user has been located within a predefined tolerance from a predefined reference location.

10. The method of claim 8 wherein the cut-off value decreases as the amount of time that the user has been located at the same location increases.

11. The method of claim 8 wherein the cut-off value decreases from an initial cut-off value to a minimum cut-off value as the amount of time that the user has been located at the same location increases from a minimum value to a maximum value in an adjustment period.

12. The method of claim 11 wherein the adjustment period is a static adjustment period.

13. The method of claim 11 wherein the adjustment period is a dynamic adjustment period.

14. The method of claim 11 wherein the adjustment period is a randomly selected value from a defined range of values.

15. The method of claim 11 wherein filtering the user profile of the user comprises randomly selecting a value from a defined range of values as the adjustment period such that the value is used as the adjustment period as long as the user remains at the same location.

16. The method of claim 11 wherein the adjustment period is a combination of a predefined initial adjustment period and a randomly selected variance value from a defined range of variance values.

17. The method of claim 16 wherein the randomly selected variance value remains the same as long as the user remains at the same location.

18. The method of claim 11 wherein the adjustment period is a function of a frequency at which the user has previously visited the same location.

19. The method of claim 11 wherein the adjustment period is a function of a number of times that the user has previously visited the same location.

20. The method of claim 11 wherein the minimum cut-off value is a randomly selected value from a defined range of values.

21. The method of claim 11 wherein filtering the user profile of the user comprises randomly selecting a value from a defined range of values as the minimum cut-off value such that the value is used as the minimum cut-off value as long as the user remains at the same location.

22. The method of claim 11 wherein the minimum cut-off value is a combination of a predefined initial minimum cut-off value and a randomly selected variance value from a defined range of variance values.

23. The method of claim 22 wherein the randomly selected variance value remains the same as long as the user remains at the same location.

24. The method of claim 11 wherein the minimum cut-off value is a function of a frequency at which the user has previously visited the same location.

25. The method of claim 11 wherein the minimum cut-off value is a function of a number of times that the user has previously visited the same location.

26. The method of claim 1 wherein filtering the user profile of the user comprises filtering the user profile of the user

based on the aggregate profile data for the current location of the user and additional information obtained for the current location of the user.

27. The method of claim 1 wherein filtering the user profile of the user comprises filtering the user profile of the user based on the aggregate profile data for the current location of the user and one or more situational awareness rules defined by a community of users.

28. The method of claim 1 wherein the aggregate profile data is an aggregation of a plurality of user profiles of a plurality of users, and interests in the plurality of user profiles having ages that are less than a predefined threshold age do not contribute to the aggregate profile data.

29. The method of claim 1 wherein the aggregate profile data is an aggregation of a plurality of user profiles of a plurality of users, and interests in the plurality of user profiles that are relevant to one or more filtering rules that define a manner in which the user profile of the user is to be filtered based on the aggregate profile data and that have ages that are less than a predefined threshold age do not contribute to the aggregate profile data.

30. The method of claim 1 wherein filtering the user profile of the user comprises:

filtering the user profile of the user based on the aggregate profile data for the current location of the user without regard to an amount of time that the user has been at the same location if configured in a first mode of operation; and

filtering the user profile of the user based on the aggregate profile data for the current location of the user and an amount of time that the user has been at the same location if configured in a second mode of operation.

31. The method of claim 1 further comprising utilizing the filtered user profile for the user to provide aggregate profile data in a mobile aggregate profiling system.

32. A computing device comprising:

a communication interface; and

a controller associated with the communication interface and adapted to:

obtain aggregate profile data for a current location of a user; and

filter a user profile of the user based on the aggregate profile data for the current location of the user to thereby provide a filtered user profile for the user.

33. The computing device of claim 32 wherein the computing device is a server in a mobile aggregate profiling system.

34. The computing device of claim 32 wherein the computing device is a mobile device of the user.

35. A computer-readable medium storing software for instructing a controller of a computing device to:

obtain aggregate profile data for a current location of a user; and

filter a user profile of the user based on the aggregate profile data for the current location of the user to thereby provide a filtered user profile for the user.

36. A computer-implemented method comprising:

obtaining aggregate profile data for a desired location; and filtering the aggregate profile data based on the aggregate profile data itself.

37. The method of claim 36 wherein filtering the aggregate profile data comprises, for each interest of a plurality of interests in the aggregate profile data, filtering the interest if a

representation value for the interest in the aggregate profile data is less than a cut-off value.

**38.** The method of claim **37** wherein the desired location is a location of a user for which the aggregate profile data is obtained and filtered, and the cut-off value is a function of an amount of time that the user has been located at the same location.

**39.** The method of claim **37** wherein the cut-off value decreases from an initial cut-off value to a minimum cut-off value as an amount of time that the user has been located at the same location increases from a minimum value to a maximum value in an adjustment period.

**40.** The method of claim **39** wherein the adjustment period is a dynamic adjustment period.

**41.** The method of claim **39** wherein the minimum cut-off value is a dynamic value.

**42.** A mobile device comprising:  
a communication interface; and  
a controller associated with the communication interface and adapted to:  
obtain aggregate profile data for a desired location; and  
filter the aggregate profile data based on the aggregate profile data itself.

**43.** A computer-readable medium storing software for instructing a controller of a computing device to:  
obtain aggregate profile data for a desired location; and  
filter the aggregate profile data based on the aggregate profile data itself.

\* \* \* \* \*