



(51) International Patent Classification:

A63F 13/20 (2014.01) G06T 19/00 (2011.01)
A63F 13/40 (2014.01) A63F 13/332 (2014.01)
A63F 13/216 (2014.01) A63F 13/211 (2014.01)
A63F 13/217 (2014.01) A63F 13/45 (2014.01)
A63F 13/213 (2014.01)

(21) International Application Number:

PCT/US2014/057095

(22) International Filing Date:

24 September 2014 (24.09.2014)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

14/042,213 30 September 2013 (30.09.2013) US

(71) Applicant: **QUALCOMM INCORPORATED** [US/US];
ATTN: International IP Administration, 5775 Morehouse
Drive, San Diego, California 92121-1714 (US).

(72) Inventors: **DAS, Saumitra Mohan**; 5775 Morehouse
Drive, San Diego, California 92121 (US). **GUPTA, Rajar-**
shi; 5775 Morehouse Drive, San Diego, California 92121
(US). **PODURI, Sameera**; 5775 Morehouse Drive, San
Diego, California 92121 (US). **KHORASHADI, Behrooz**;
5775 Morehouse Drive, San Diego, California 92121 (US).

(74) Agents: **HUNTER CLARK PLLC** et al.; Suite 213-T,
900 Cummings Center, Beverly, Massachusetts 01915
(US).

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

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

[Continued on next page]

(54) Title: AUGMENTED REALITY APPARATUS, METHOD AND PROGRAM

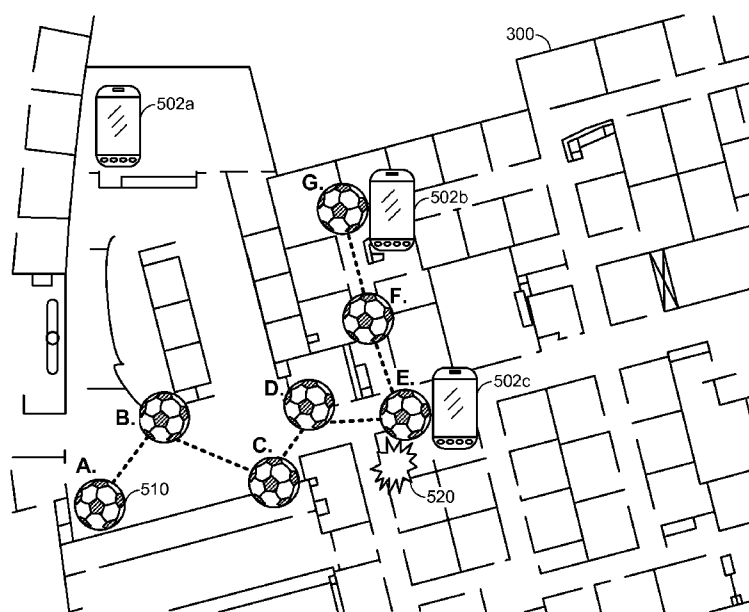


FIG. 5

(57) Abstract: Techniques for providing a user with an augmented virtuality (AV) experience are described herein. An example of a method of providing an AV experience includes determining a location of a mobile device, determining a context based on the location, obtaining AV object information, displaying the AV object information in relation to the context, detecting an interaction with the context, modifying the AV object information based on the interaction, and displaying the modified AV object information. The context may include weighting information. The weighting information may be based on Received Signal Strength Indication (RSSI) or Round-Trip Time (RTT) data. The weighting information may be associated with a composition of a physical object in the context. A user gesture may be received, and the AV object information may be modified based on the received gesture information.

**Declarations under Rule 4.17:**

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

AUGMENTED REALITY APPARATUS, METHOD AND PROGRAM

BACKGROUND

[0001] Mobile computing devices (mobile devices) provide users with access to a
5 variety of information via wireless communication systems. Some mobile computing
devices may be used for augmented reality (AR) applications, in which a real world
object is imaged and displayed on a screen along with computer generated information,
such as an image or textual information. AR can be used to provide information, either
graphical or textual, about a real world object, such as a building or product. Typically,
10 the location or other surrounding objects are not considered when rendering an AR
object.

SUMMARY

[0002] In addition for use in augmented reality applications, a mobile devices may
15 also enable users to explore Augmented Virtuality (AV) where real world objects or
events are merged into a virtual world. For example, a position determining module on
mobile device may be used to present a user with a context (e.g., a map, location,
images) of their surrounding environment (which comprises the virtual world) in
conjunction with a graphical object which depicts the relative location of an augmented
20 virtuality object. The mobile device can be configured to allow the user to interact with
the augmented virtuality object based on the location of the mobile device, as well as on
the features associated with object. An augmented virtuality object can be accessed via
a network, or previously stored on a mobile device. An augmented virtuality object can
be a virtual object with features (e.g., variables) such as size, shape, location, speed, and
25 direction. The virtual object may be associated with a graphic image or other animation
object based on a real world or fantastic objects (e.g., a ball, a flying coupon, a flood, a
swarm of bugs, a monster, etc...). The virtual objects may be used in business
application, such as marketing and advertising, as well as in educational and
recreational applications, such as training aids and games. The behavior of the virtual
30 object in the virtual world depicted on the mobile device may be based on an algorithm
or based on inputs from events happening in the real world context of the user or some
combination thereof.

[0003] An example of a method for providing augmented virtuality (AV) on a mobile device according to the disclosure includes determining a location of the mobile device, determining a context based on the location, obtaining AV object information, displaying the AV object information in relation to the context, detecting an interaction
5 with the context, modifying the AV object information based on the interaction, and displaying the modified AV object information.

[0004] Implementations of the method may include one or more of the following features. The context may include weighting information. The weighting information may be based on Received Signal Strength Indication (RSSI) or Round-Trip Time
10 (RTT) data. The weighting information may be associated with a composition of a physical object in the context. A user gesture may be received, and the AV object information may be modified based on the received gesture information. An alert may be provided to the user, and AV object information may be modified based on a reaction time, such that the reaction time is a difference between a time the alert signal is
15 provided and a time the gesture is received. AV object information may be received from an augmented virtuality (AV) server, the modified AV object information may be provided to the AV server.

[0005] An example of a method for providing augmented virtuality (AV) on a mobile device according to the disclosure includes determining a context based on a location of
20 the mobile device, generating a state for an AV object, determining a context interaction result, modifying the state of the AV object based on the context interaction result, determining a mobile device interaction result, modifying the state of the AV object based on the mobile device interaction result, and displaying the AV object.

[0006] Implementations of the method may include one or more of the following
25 features. The state of the AV object may include a direction variable. The state of the AV object may be modified based on the context interaction result including modifying the direction variable. Modifying the state of the AV object based on the mobile device interaction result may include modifying the direction variable. The state of the AV object may include a size variable. Modifying the state of the AV object based on the
30 context interaction result may include modifying the size variable. The mobile device interaction result may be determined by receiving a gesture from a user. A state of the mobile device may be modified based on the mobile device interaction result.

[0007] An example of a method of using virtual tools in Augmented Reality (AR) applications according to the disclosure includes determining a context based on a location of a mobile device, generating a state for a virtual threat, generating a state for a virtual tool, determining a context interaction result, determining a tool interaction
5 result, modifying the state of the virtual threat based on the context interaction result and the tool interaction result, and displaying the virtual threat and the virtual tool.

[0008] Implementations of the method may include one or more of the following features. The state of the virtual tool may include a location variable. The state of the virtual tool may be modified based a state of the mobile device. The tool interaction
10 result may be determined based on the modified state of the virtual tool. The tool interaction result may constrain the state of the virtual threat.

[0009] An example of a method of determining interaction variables in an Augmented Reality (AR) application according to the disclosure includes obtaining a context, determining location data corresponding to one or more areas within the context,
15 determining weighting information for each of the one or more areas, determining interaction variables based on the weighting information, and storing the interaction variables.

[0010] Implementations of the method may include one or more of the following features. The weighting information may be based on RSSI information. The weighting
20 information may be based on RTT information.

[0011] An example of a method for providing augmented reality (AR) on a mobile device according to the disclosure includes determining a context based on a location of the mobile device, establishing a control area within the context, generating a state for a roving threat, wherein the state of the roving threat is constrained by the context,
25 determining an engagement zone for the roving threat, wherein the engagement zone is constrained by the context, determining a state of the mobile device, wherein the state includes a current location, and initiating a first mini-game based on the state of the mobile device.

[0012] Implementations of the method may include one or more of the following
30 features. The first mini-game may be initiated when the location of the mobile device is within the control area. A second mini-game may be initiated when the location of the

mobile device is within the engagement zone. The engagement zone may be a line of sight variable. The line of sight variable may be constrained by the context.

[0013] An example of a method for providing Augmented Virtuality (AV) object information to one or more mobile devices according to the disclosure includes
5 registering a mobile device with an AV server, receiving AV object information and a peer position update from the AV server, and providing a position update to the AV server. In an implementation of the method, a gesture update can be provided to the AV server.

[0014] An example of a method for providing Augmented Virtuality (AV) object
10 information to one or more of mobile devices according to the disclosure includes registering a mobile device with an AV server, receiving an AV object physics engine module, and providing AV object state variable updates to the AV server.

[0015] An example of an apparatus for providing augmented virtuality (AV) on a mobile device according to the disclosure includes a memory module, a processor
15 coupled to the memory module and configured to determine a location of a mobile device, determine a context based on the location, obtain an AV object information, display the AV object information in relation to the context, detect an interaction with the context, modify the AV object information based on the interaction, and display the modified AV object information.

[0016] An example of an apparatus for providing augmented virtuality (AV) on a mobile device according to the disclosure includes a memory module, a processor
20 coupled to the memory module and configured to determine a context based on a location of the mobile device, generate a state for an AV object, determine a context interaction result, modify the state of the AV object based on the context interaction result, determine a mobile device interaction result, modify the state of the AV object
25 based on the mobile device interaction result, and display the AV object.

[0017] An example of an apparatus for providing augmented virtuality (AV) on a mobile device according to the disclosure includes a memory module, a processor
coupled to the memory module and configured to determine a context based on a
30 location of the mobile device, generate a state for an AV object, determine a context interaction result, modify the state of the AV object based on the context interaction

result, determine a mobile device interaction result, modify the state of the AV object based on the mobile device interaction result, and display the AV object.

5 **[0018]** An example of an apparatus for using virtual tools in Augmented Virtuality (AV) applications according to the disclosure includes a memory module, a processor coupled to the memory module and configured to determine a context based on a location of a mobile device, generate a state for a virtual threat, generate a state for a virtual tool, determine a context interaction result, determine a tool interaction result, modify the state of the virtual threat based on the context interaction result and the tool interaction result, and display the virtual threat and the virtual tool.

10 **[0019]** An example of an apparatus for determining interaction variables in an Augmented Virtuality (AV) application according to the disclosure includes a memory module, a processor coupled to the memory module and configured to obtain a context, determine location data corresponding to one or more areas within the context, determine weighting information for each of the areas, determine interaction variables
15 based on the weighting information, and store the interaction variables.

[0020] An example of an apparatus for providing a user with an Augmented Virtuality (AV) experience according to the disclosure includes a memory module, a processor coupled to the memory module and configured to determine a context based on a location of a mobile device, establish a control area within the context, generate a state
20 for a roving threat, wherein the state of the roving threat is constrained by the context, determine an engagement zone for the roving threat, wherein the engagement zone is constrained by the context, determine a state of the mobile device, wherein the state includes a current location, and initiate a first mini-game based on the state of the mobile device.

25 **[0021]** An example of a computer program product residing on a processor-executable storage medium according to the disclosure includes processor-executable instructions configured to cause a processor to determine a location of a mobile device, determine a context based on the location, obtain an AV object information, display the AV object information in relation to the context, detect an interaction with the context,
30 modify the AV object information based on the interaction, and display the modified AV object information.

[0022] An example of a computer program product residing on a processor-executable storage medium according to the disclosure includes processor-executable instructions configured to cause a processor to determine a context based on a location of a mobile device, generate a state for an AV object, determine a context interaction result, modify the state of the AV object based on the context interaction result, determine a mobile device interaction result, modify the state of the AV object based on the mobile device interaction result, and display the AV object.

[0023] An example of a computer program product residing on a processor-executable storage medium according to the disclosure includes processor-executable instructions configured to cause a processor to determine a context based on a location of a mobile device, generate a state for a virtual threat, generate a state for a virtual tool, determine a context interaction result, determine a tool interaction result, modify the state of the virtual threat based on the context interaction result and the tool interaction result, and display the virtual threat and the virtual tool.

[0024] An example of a computer program product residing on a processor-executable storage medium according to the disclosure includes processor-executable instructions configured to cause a processor to obtain a context, determine location data corresponding to one or more areas within the context, determine weighting information for each of the areas, determine interaction variables based on the weighting information, and store the interaction variables.

[0025] An example of a computer program product residing on a processor-executable storage medium according to the disclosure includes processor-executable instructions configured to cause a processor to determine a context based on a location of a mobile device, establish a control area within the context, generate a state for a roving threat, wherein the state of the roving threat is constrained by the context, determine an engagement zone for the roving threat, wherein the engagement zone is constrained by the context, determine a state of the mobile device, wherein the state includes a current location, and initiate a first mini-game based on the state of the mobile device.

30

[0026] Items and/or techniques described herein may provide one or more of the following capabilities, as well as other capabilities not mentioned. A context can be associated with a location of a mobile device. Weighting functions for the context can be determined. Augmented Virtuality (AV) objects are virtual objects which can be presented on one or more mobile devices. AV objects can model real world objects (e.g., a ball, flowing water, wild animals, bubbles, expanding gas, etc...), as well as imaginary objects (e.g., a magic fire ball, an alien robot, angry giants, flying coupons). AV objects can include state variables to model virtual and physical attributes. Users can interact with AV objects via sensors in a mobile device. The state variables of one or more AV objects can be modified or constrained based on the context. An AV server can host and operate AV applications. Designers can register AV applications on an AV server. AV applications, and associated AV objects, can be provided to mobile devices. AV applications can be executed in local and client configurations. Mobile devices can be configured to provide location and AV object state information to an AV server. An AV server can be configured to propagate AV object state information to one or more mobile devices. An AV object can represent a virtual tool (i.e., another AV object) and a user can interact with a virtual tool via sensors in a mobile device. Virtual tools can include state variables, which can be modified or constrained based on the context. Virtual tools can be used to interact with AV objects. A context can include one or more control areas. A mobile device can be configured to execute one or more mini-games based on its proximity to a control area. Other capabilities may be provided and not every implementation according to the disclosure must provide any, let alone all, of the capabilities discussed. The results of the interaction between an AV object and a context can be utilized in a wide range of business, education and entertainment applications. Further, it may be possible for an effect noted above to be achieved by means other than that noted, and a noted item/technique may not necessarily yield the noted effect.

BRIEF DESCRIPTION OF THE DRAWINGS

[0027] FIG. 1 is an exemplary network environment for an Augmented Virtuality (AV) system.

[0028] FIG. 2 is schematic block diagram of an exemplary mobile device.

- [0029] FIG. 3 is a top view of an exemplary configuration for determining weighting functions within a context.
- [0030] FIG. 4 is block flow diagram of an exemplary data transfer scheme between an AV server and one or more mobile devices.
- 5 [0031] FIG. 5 is a top view of an exemplary AV object in multiple states.
- [0032] FIG. 6 is a top view of an exemplary AV object travelling through a context.
- [0033] FIG. 7 is a top view of an exemplary expanding threat AV object.
- [0034] FIG. 8 is a top view of a context with an exemplary expanding threat and virtual tools AV objects.
- 10 [0035] FIG. 9 is a top view of a context with an exemplary virtual trainer AV object.
- [0036] FIG. 10 is a top view of a context with multiple control areas and exemplary roving threat AV objects
- [0037] FIG. 11 is a top view of a mobile device executing one or more mini-games.
- [0038] FIG. 12 is a block flow diagram of a process for displaying an AV object.
- 15 [0039] FIG. 13A is a block flow diagram of a process for determining interaction variables for areas within a context.
- [0040] FIG. 13B is an exemplary data structure for weighting information and interaction variables.
- [0041] FIG. 14A is block flow diagram of a process for providing position and
20 gesture updates to an AV server.
- [0042] FIG. 14B is a block flow diagram of a process for providing AV object position updates to an AV server.
- [0043] FIG. 15 is a block flow diagram of a process for modifying the state variables of an AV object.
- 25 [0044] FIG. 16 is a block flow diagram of a process for modifying the state variables of an AV object based on context or tool interaction.

[0045] FIG. 17 is a block flow diagram of process for initiating a mini-game on a mobile device.

5

DETAILED DESCRIPTION

[0046] Techniques are discussed herein for providing Augmented Virtuality (AV) objects to a user based on the location of the user and the context associated with the location. For example, a map of a real location is obtained from a positioning engine or server and provided to a mobile device. The map can include additional context
10 information such as a heatmap, which represents weighting functions associated with various objects and structural features in the location. The weighting functions can be based on RSSI (Received Signal Strength Indicator) and RTT (Round Trip Time) data obtained from access points. For example, the weighting functions can be used to determine the locations of walls and other obstacles or the way in which AV objects
15 will move or be mutated within that context. The context may include information about structural elements such as, for example, the locations of rooms, walls, doors, corridors, and other architectural features. The state variables (i.e., the state) of an AV object can be modified based on interaction with the context and the user. For example, a physics engine executing on an AV server or a mobile device may be used to calculate
20 the results of the interactions between the AV object and the context, and/or the interactions between the AV object and user. The state of an AV object is propagated or otherwise accessible to other mobile devices within a location. For example, the interactions between one or more AV objects with the context can be used in various business, educational and entertainment applications. These examples, however, are not
25 exhaustive.

[0047] Referring to FIG. 1, an exemplary network environment 100 for an Augmented Virtuality (AV) system is shown. Network environment 100 may include any number of mobile devices such as, for example, mobile devices 120 and 130. Mobile devices 120 and 130 may communicate wirelessly with an AV server 110 via
30 network 140. Mobile devices 120 and 130 may comprise any suitable mobile computing platform, including a cellular phone, a smart phone, a personal digital assistant, a low duty cycle communication device, a laptop computer, a portable media

player device, a personal navigation device, and a portable electronic device comprising a digital camera, to name a few examples. Network environment 100 may further include non-mobile computing platforms such as computing platform 122 that may also communicate with the AV server 110 via network 140. In at least some
5 implementations, a mobile device may be adapted to communicate by docking with computing platform 122 by wired or wireless communication, for example, without utilizing network 140.

[0048] Network 140 may comprise one or more of a wide area network (e.g., the Internet), a local area network (e.g., an intranet), and/or a personal area network. For
10 example, mobile devices 120 and 130 may communicate wirelessly with the AV server 110 via any suitable communication protocol, including cellular communication protocols such as CDMA, GSM, or WCDMA and/or wireless local area network (WLAN) protocols such as Wifi or WiMAX. Accordingly, network 140 may include one or more wireless transceivers in such implementations. Wireless transceivers of
15 network 140 may also be utilized by mobile devices 120 and 130 to obtain positioning information for estimating mobile device position.

[0049] In some implementations, a trilateration based approach may be utilized by a mobile device to estimate geographic position. For example, techniques including Advanced Forward Link Trilateration (AFLT) in CDMA or Enhanced Observed Time
20 Difference (EOTD) in GSM or Observed Time Difference of Arrival (OTDOA) in WCDMA may be used, which measures at a mobile device the relative times of arrival of wireless signals transmitted from each of several transmitter equipped base stations. As another example, a mobile device may estimate its position by obtaining a Media Access Control (MAC) address or other suitable identifier associated with a wireless
25 transceiver and correlating the MAC address or identifier with a known geographic location of that wireless transceiver.

[0050] Mobile devices 120 and 130 may further obtain wireless positioning signals from a positioning system 150 to estimate mobile device position. For example, positioning system 150 may comprise a Satellite Positioning System (SPS) and/or a
30 terrestrial based positioning system. Satellite positioning systems may include, for example, the Global Positioning System (GPS), Galileo, GLONASS, NAVSTAR, GNSS, a system that uses satellites from a combination of these systems, or any SPS

developed in the future. As used herein, an SPS will also be understood to include pseudolite systems. Terrestrial positioning techniques may also be used, such as WiFi/WLAN methods including using RSSI and/or RTT of WiFi and WLAN signals. The positioning system 150 may be a positioning server and the functionality of the positioning system 150 may be included in the AV server 110. It should be understood, however, that particular positioning techniques provided here are merely example positioning techniques, and that claimed subject matter is not limited in this respect.

[0051] The AV server 110 may comprise one or more servers and/or other suitable computing platforms. Accordingly, the AV server 110 may include storage media 112 having database 114 and instructions 116 stored thereon. The storage media 112 may be considered a memory module. The AV server 110 may include one or more processors such as example processor 118 for executing instructions 116 and/or for facilitating storage and retrieval of data at database 114. The processor 118 is preferably an intelligent device, e.g., a personal computer central processing unit (CPU) such as those made by Intel® Corporation or AMD®, a microcontroller, an application specific integrated circuit (ASIC), etc. The AV server 110 may further include a communication interface 160 for facilitating communication with clients of network 140, including mobile devices 120 and 130, and computing platform 122.

[0052] To facilitate understanding of the disclosed implementations, instructions 116 may be at times described in terms of one or more modules for performing particular operations. As one example, instructions 116 may include an object interaction engine module 162 to determine the results of an interaction between an AV object and the context and/or a user. As another example, instructions 116 may include a content management module 164 to manage the deployment of AV object information within a network. Content management module 164 may include a number of tools to enable a user (e.g., a human user) to interact with the content management module, for example, via a mobile device (e.g., mobile device 120) or other suitable computing platform (e.g., computing platform 122). In at least some implementations, content management module 164 may include an object deployment tool 166 to enable a developer (i.e., user) to deploy AV objects within a network system, and an object registration tool 170 to enable a developer to register AV objects and manage access rights of other users to AV object content that has been deployed by that developer. Such tools may be accessible to a developer via input or output devices of a mobile device or other suitable

computing platform. For example, a graphical user interface (GUI) presented at a graphical display of a mobile device may enable a developer to define an AV object comprising one or more attribute files and other media content items by using one or more of tools 166, and 170.

- 5 **[0053]** Database 114 may include a variety of data, including one or more associated items. Associated item 176 may include, for example, at least one media content item 178 and other associated data elements, including for example: a context data 180, weighting information 182, metadata 186, and an object physics engine module 174. Database 114 may further include an object library 192 having one or more pre-existing
10 virtual objects 194.

- [0054]** The context data 180 may define a location relative to the location of a user, where media content such as a floor plan (e.g., media content item 178) may be presented. Context data 180 may be defined by a user via object deployment tool 166. For example, the context data 180 may be an electronic medium such as a digital floor
15 plan (e.g., AutoCAD .dwg file, a Building Information Model (BIM), Sketch-Up files).

- [0055]** Weighting information 182 may define the structural aspects of physical objects within a context. For example, map processing techniques can be used to process a building layout and assign properties and/or attributes to the building features. In an embodiment, RSSI and/or RTT models can be used to estimate the properties
20 and/or attributes of the building features. The weighting information may be used to model other physical objects such as obstructions and/or other environmental conditions within a context. Weighting information 182 may be defined by a user via object deployment tool 166. For example, a user may utilize object deployment tool 166 to enter context related input for the object interaction engine module 162.

- 25 **[0056]** Metadata 186 may include a rights definition 188 defining rights of users to access deployed media content items such as associated media content item 178. As a non-limiting example, a user may exclude select users from accessing a particular media content item (e.g., an AV object and associated files), or a user may grant select users the right to access a particular media content item within a context.

- 30 **[0057]** Metadata 186 may further include device state data 190 that is associated with media content item 178. Device state data 190 may indicate a position and/or

orientation of a mobile device, for example, while the mobile device is interacting with an AV object. In some implementations, a position of a mobile device may be represented as one or more latitude, longitude, and/or altitude values measured relative to a reference datum. Similarly, orientation of a mobile device may be represented by one or more roll, pitch, and/or yaw values measured relative to a reference datum. The state of a mobile device may include received gesture updates such as taps, swipes, and shakes. Such device state data may be based upon one or more positioning signals and/or inertial sensor measurements that are obtained at the mobile device as will be described in greater detail with reference to FIG. 2.

10 [0058] Referring to FIG. 2, a schematic block diagram of an example mobile device 200 is shown. Mobile device 200 may serve as a non-limiting example of mobile devices 120 and 130 of FIG. 1 for at least some implementations. Accordingly, mobile device 200 may include a communication interface 228 to facilitate communication with other computing platforms, such as an AV server 110, mobile devices 120, 130 and/or computing platform 122 of FIG. 1, for example. In an embodiment, the communication interface 228 may be configured to communicate with other networks (e.g., WAN, WLAN), and may be configured to be used in location determination (e.g., SPS signals). The communication interface 228 may enable wireless communication with communication networks including example network 140. Mobile device 200 may further include a receiver 230 to receive positioning signals from a positioning system such as positioning system 150, for example.

[0059] Mobile device 200 may include input devices 218 to receive user inputs. Input devices 218 may include, for example, one or more physical buttons, keyboards, controllers, microphones, pointing devices, and/or touch-sensitive surfaces. Mobile device 200 may include output devices 221 to present information to a user. As one example, output devices 221 may include a graphical display 222 to present a graphical user interface 224. In general, the graphical display 222 may comprise a touch-sensitive graphical display (e.g., a touch screen) capable of receiving user input. A touch-sensitive graphical display may utilize any suitable sensing technique for detecting user input at or near a display surface, including capacitive sensing, resistive sensing, and/or optical sensing techniques, for example. In such touch-sensitive implementations, graphical display 222 may serve as both an input device for receiving user input and an

output device for presenting graphical content. Output devices 221 may further include an audio speaker 226 for presenting audio content in some implementations.

[0060] In at least some implementations, a graphical user interface (GUI) 224 may be presented at graphical display 222. A GUI 224 may include an AV view 234 to present
5 AV objects with context information. Information layer 236 may comprise one or more rendered media content items such as rendered media content item 238 and one or more graphical control elements such as graphical control element 240. As a non-limiting example, rendered media content item 238 may correspond to a rendered state of media content item 178 such as an image, a video, an animation, among other suitable
10 graphical content or a portion thereof. Graphical control element 240 may include one or more graphical elements for enabling a user to provide user input for controlling functionality of the mobile device.

[0061] Mobile device 200 may include one or more inertial sensors 232 to obtain inertial sensor measurements on-board the mobile device. In at least some
15 implementations, inertial sensors 232 may comprise one or more accelerometers, gyroscopes, or other suitable devices for measuring an inertial state of a mobile device. In some implementations, an inertial state of a mobile device may be measured by inertial sensors 232 along multiple axes in Cartesian and/or polar coordinate systems to provide an indication of position and/or orientation of the mobile device.

20 [0062] It will be appreciated that inertial sensor measurements obtained from inertial sensors 232 may be processed (e.g., integrated over time) to obtain such estimates of mobile device position and/or orientation. As previously described, a position of a mobile device may be represented as one or more latitude, longitude, and/or altitude values and an orientation of a mobile device may be represented by one or more roll,
25 pitch, and/or yaw values measured relative to a reference datum. Mobile device 200 may include a camera 220 to obtain a camera view which may be presented at graphical display 222. Camera 220 may comprise a digital camera in some implementations having a number of optical elements including one or more lenses for focusing light and/or one or more light sensing elements for converting light into digital signals
30 representative of image and/or video data. As a non-limiting example, a light sensing element may comprise an optical pickup, charge-coupled device and/or photoelectric device for converting light into digital signals. Digital signals obtained from one or

more light sensing elements of the camera may be processed and/or rendered at graphical display 222 as camera view. Furthermore, in some implementations, a camera view obtained via camera 220 may be stored at storage media as image and/or video content. For example, digital signals obtained from camera 220 may be encoded and
5 stored at storage media 210 as .jpeg, .mpeg, or other suitable format. Such digital signals representative of a camera view may be transmitted to other computing platforms for storage and/or processing. For example, a camera view obtained at a mobile device may be transmitted to a server system or other suitable computing platform via a communication network or by docking with another computing platform.

10 **[0063]** Mobile device 200 may further include storage media 210 having a database 212 and instructions 214 stored thereon. The storage media 210 can be a memory module. Mobile device 200 may include one or more processors such as example processor 216 for executing instructions 214 and/or facilitating storage and retrieval of data at database 212. Database 212 may include a variety of data, including some or all
15 of the data elements previously described with reference to database 114 of FIG. 1. For example, database 212 may retain locally at mobile device 200, one or more of associated content item 176 and object library 192, among other data. It will be appreciated that such data may also be retained remotely, for example, at a server system such as the AV server 110 of FIG. 1 where it may be accessed by the mobile
20 device via a wireless network.

[0064] Instructions 214 may similarly include one or more of the previously described modules and/or tools of instructions 116 of FIG. 1. For example, instructions 214 may include previously described object interaction engine module 162 and content management module 164 retained locally at mobile device 200. Instructions 214 may
25 further include physics engine module 254 to model the motion and reaction of AV objects within a context. Instructions may further include a device state module 256 for processing positioning signals obtained via receiver 230 and/or inertial sensor measurements obtained via inertial sensors 232, as well as store state variables associated with an AV object. For example, device state module 256 may be adapted to
30 estimate a position and/or orientation of mobile device 200 based, at least in part, on such positioning signals and/or inertial sensor measurements. An estimated position and/or orientation of mobile device 200 may be stored at database 212, for example, as previously described with reference to device state data 190. In at least some

implementations, device state module 256 may be adapted to obtain device state data for a mobile device responsive to the interactions with an AV object. For example, an AV object can be a virtual dodge ball that can virtually hit the mobile device (e.g., based on the actual location of the mobile device), the device state module 256 can indicate that the mobile device has been hit, or is still free. Device state module 256 may be further adapted to associate the device state data with state specific media content items (e.g., as metadata), for example, at database 212 or database 114.

[0065] It will be appreciated that at least some elements of the example mobile device 200 of FIG. 2 may be present in other mobile or non-mobile computing platforms such as computing platform 122 of FIG. 1 to enable such computing platforms to perform one or more of the operations, processes, and methods described herein. For example, computing platform 122 of FIG. 1 may comprise storage media such as storage media 210 including instructions 214 and/or database 212, one or more processors such as processor 216, one or more input and output devices, and a communication interface such as communication interface 228.

[0066] Referring to FIG. 3, a top view of an exemplary configuration for determining weighting functions within a context is shown. The configuration includes a context 300, an access point 302, and three exemplary locations of one or more mobile devices 304a, 304b, 304c. The context 300 can represent a location including wall, corridors, and other obstacles. The term access point as used herein includes different variations of terrestrial transceivers (e.g., base stations, WiFi access points, femtocells). The access point 302 may allow the mobile devices 304a, 304b, 304c to connect to a wireless network using Wi-Fi, Bluetooth, a cellular communication technology such as Code Division Multiple Access (CDMA), Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA), Orthogonal Frequency Division Multiple Access (OFDMA), Single-Carrier Frequency Division Multiple Access (SC-FDMA), or any other suitable wireless technology and/or standard. In an embodiment, the access point 302 connected to the AV server 110 can communicate with one or more of the mobile devices 304a, 304b, 304c to determine or update weighting information. The AV server 110, or another positioning server may include the algorithms used to calculate weighting information. This information can come from current and/or historical data from one device or multiple devices (i.e., crowdsourcing), and may be kept on a central location (e.g., network element) and/or shared between mobile devices.

In an example, a map server (or other network device) can maintain and update a context with corresponding weighting information (e.g., a heatmap) based on input from one or more mobile devices. Additionally or alternatively, each mobile device can maintain and update weighting information and/or share the updated weighting
5 information with a server and/or other mobile device(s). Where a set of location data is associated with a particular context (e.g., building), a mobile device can store and maintain multiple sets of location data (e.g., maps) for multiple contexts. The location data and/or corresponding weighting information may also be time and/or date sensitive. For example, a heatmap in and/or around a crowded cafeteria during lunchtime on a
10 weekday may be different than when it is virtually empty at midnight on a weekend.

[0067] In an embodiment, the AV server 110 can determine a context (e.g. an area) in which a mobile device is located. This initial determination can be based on any of a variety of data, such as previous known location, dead reckoning data, sensor data, GNSS and other signal sources, and the like. Some embodiments may provide for
15 determining a rough location via a location tied to a particular MAC ID which is then used to determine a more accurate location. The context in which the mobile device is determined to be located can impact the of RSST and RTT measurements. The context can include information indicative of environmental conditions of one or more areas. The context can, for example, include a map, such as a layout of a building, which can
20 indicate the location of physical objects such as walls, windows, doors, and/or other structures that could impact the condition of the area's environment. The location data can further include weighting information, which can be generated by using map processing algorithms and may also be indicative of a condition of the area's environment (e.g., number of walls between the area and an access point, wall
25 composition type, a distance, hardware information of an access point, and/or a confidence level for RSSI and/or RTT data (e.g., distance measurements between the mobile device and an access point), etc.). The weighting information may pertain to a single access point, or more than one access points, with which RSSI and/or RTT data can be obtained.

30 [0068] RSSI and RTT data can weighted based on the condition of the environment. For example, the weighting, which can include weights for both RSSI and RTT data, can be calculated using one or more confidence metrics provided in the weighting information of the location data. It may also take into account specific information that

may not be included in the weighting information, such as device-specific information for a given mobile device and/or access point. For example, a mobile device may have an antenna and/or other signal-reception hardware that would cause RSSI data to be more accurate than other mobile devices, in which case more weight can be given to the RSSI data. A mobile device may be configured to measure the RSSI and RTT information and provide the measurement data to a position server, or the AV server 110. In another example, an access point may have certain hardware that causes it to have a processing delay that is less consistent than many other access points, in which case less weight can be given to RTT data measured using that access point. When the location of the mobile is known, the weighting information can be used to model the environment (e.g., the physical objects). The object interaction engine module 162 can utilize the weighting information to model a wall which an AV object cannot pass, model deformation to an object depending on the number or characteristics of walls passed, or model a corridor which will not interact with an AV object. The weighting information can be utilized, for example, to determine the density of the walls between the access point 302 and the mobile devices 304a, 304b, 304c which can further govern how virtual objects will traverse to those mobile devices. In an embodiment, the network device(s) can maintain a heatmap which can be used by the object interaction engine module 162.

[0069] Referring to FIG. 4, a block flow diagram of an exemplary data transfer scheme between an AV server and one or more mobile devices is shown. The transfer scheme includes an AV server 110, and user mobile devices 402a, 402b, 402c, (collectively 402) and a developer device 420. The mobile devices 402 and developer device 420 can be configured as previously discussed (e.g., mobile device 200). In an embodiment, mobile devices 402 and developer device 420 are configured to communicate with the AV server 110 through a network 140 and one or more access points (not shown). For purposes of explanation, the network 140 and access points are incorporated into a collection of inbound and outbound communication paths 410b-d, 411b-d, 420a. The outbound communication paths 410b-d (collectively 410) represent AV object information transmitted from the AV server 110 to the mobile devices, and the inbound communication paths 411b-d (collectively 411) represent AV object information transmitted from the mobile devices to the AV server 110. While shown as different communication paths, the inbound and outbound communication paths can be

a single path. The developer device 420 communicates with the AV server 110 over the communication path 421.

[0070] In operation, a developer can create a virtual object and utilize the developer device 420 (or other computing platform 122) to register an AV object into the database 114. For example, the developer device 420 can access the content management module 164 to create, or upload, AV objects and the associated items 176. The architecture of the data transfer scheme between the AV server 110 and the mobile devices 402 based on the AV object and the capabilities of the mobile devices 402. In an example, the AV server 110 can provide initial AV object information including the associated items 176 to the mobile device over communication path 410. The object interaction engine module 162 can utilize the AV object information to enable user interaction with an AV object. The physics engine module 254 executing locally on the mobile device can simulate the motion of the AV object and the state variables describing the AV object, and the mobile device can be updated. The state updates can be sent to the AV server over the inbound communication path 411 and then broadcast to the other mobile devices over outbound communication path 410. In an embodiment, the mobile devices 402 can send device state information to the AV server 110 via the inbound communication path 411, and the AV server 110 can execute an object interaction engine module 162 to determine the interactions and corresponding state of an AV object. The AV object state information can then be transmitted to the mobile devices 402 via the outbound communication path 410. In the case of a single player (e.g., only one mobile device 402a), the AV server can be used to initialize an application on a mobile device without the need for further communication between the mobile device and the AV server. Other data transfer schemes may be used transfer AV object and mobile device state information to and from the AV server 110 and the mobile devices 402.

[0071] Referring to FIG. 5, with further references to FIGS. 1 and 2, a top view of an exemplary AV object in multiple states is shown. The view includes a context 300, a one or more mobile devices 502a, 502b, 502c (collectively 502), a virtual ball 510 (i.e., an AV object), and an interaction event 520. The locations of the virtual ball 510 at various times are indicated with the letters A. through G. The AV server 110 can be configured to launch the virtual ball 510 at a random initial location variable, direction and velocity variables for the context 300. For example, the AV server 110 provides the

initial state information for the virtual ball 510 to each of the mobile devices 502. The initial state information indicates that the virtual ball 510 begins at location 'A' and proceeds to location 'B' at an initial speed. The location and movement of the virtual ball 510 appears on the graphical display 222 of each of the mobile devices 502. In an embodiment, the AV server 110 utilizes the object interaction engine module 162 to update the state information on a periodic basis (e.g., 50ms, 100ms, 200ms, 500ms). For example, the object interaction engine module 162 processes a current state of the virtual ball 510 in view of the context data 180 and the weighting information 182 to produce a new state for the virtual ball 510. As a result, the weighting information 182 and/or context data 180 are used to create virtual boundaries that interact with the virtual ball 510. For example, referring to location 'B', the virtual ball 510 virtually bounces off a wall and proceeds to location 'C'. The results calculated by the object interaction engine module 162 can change state of the virtual ball 510. The state variables associated with position, speed, and direction of the virtual ball 510 can be changed and then provided to the mobile devices 502. State variables may also be updated on one or more mobile devices 502 and provided to an AV server for dissemination to other mobile devices participating in the application (if any). Other applications can have additional state variables. For example, virtual environmental aspects such as wind speed and direction may be used to alter the state of an AV object.

[0072] Continuing the example of the virtual ball 510, after the deflecting at locations 'C' and 'D', the virtual ball 510 continues down the corridor toward location 'E' and mobile device 502c. A user of the mobile device 502c can monitor the progress of the virtual ball 510 as it travels from position 'A' to position 'E'. As the virtual ball 510 approaches, the user can enter a gesture into the mobile device 502c to create the interaction event 520. The gesture can include a screen tap or swipe, a shake, or a swing motion. Other inputs to the mobile device 502c can be used as gestures to initiate the interaction event 520. The timing and the nature of the gesture can be captured by the device state module 256 and sent to the AV server 110 as an input to the object interaction engine module 162. For example, when the virtual ball 510 arrives at location 'E', the user of the mobile device 502c flicks the graphical display 222 to send the virtual ball 510 off toward location 'F'. The results of the interaction event can modify the state of the virtual ball 510. In general, modifying the state of an AV object means changing the values of state variables associated AV object attributes (e.g.

direction and velocity). The virtual ball 510 can continue toward location 'G' and possibly virtually collide with the mobile device 502b. In a gaming application, a user may gain points for virtually hitting another user with an AV object.

[0073] The virtual ball application of FIG. 5 is exemplary only, and not a limitation.

5 Other applications can include objectives such as catching the AV object, or to find one or more AV objects (e.g., Easter egg hunt). In some gaming applications, the users can score points by using an AV object to hit other players (e.g., dodge ball) or by knocking the AV object (e.g., virtual ball) past a player (e.g., ping-pong). In an example, the user can be provided with an alert signal and must provide a gesture update to the mobile
10 device within a certain timeframe (e.g., a reaction time) in order to successfully interact with the virtual object (e.g., hot potato). Other games based on position and context can be created. In an embodiment, an AV object can be presented on the graphical display 222 as an augmented reality object in the AV view 234. For example, a user may utilize the camera 220 to obtain a near real time view of objects in their surroundings (e.g., a
15 hallway, a room), and an image of the virtual ball 510 travelling towards the user can be presented in relation to the objects in the camera view. The size of the AV object in the camera view can change based on the relative location of the AV object from the user (i.e., the virtual ball 510 can appear larger as it gets closer to the user).

[0074] Referring to FIG. 6, a top view of an exemplary AV object travelling through a
20 context 300 is shown. The view includes a context 300, a first mobile device 602, a second mobile device 604, an interaction area 606, and a fireball 610 (i.e., an AV object) represented in different states 610a-f. The fireball 610 is depicted as traveling through obstacles in the context 300. In an exemplary gaming application, the AV server 110 can produce a fantasy type virtual object such as a fireball 610. For example,
25 the object library 192 and associated item 176 can include features associated with the attributes of the fireball 610, as well as other information such as a scoring mechanism. For example, object library 192 can include:

- Fireball object = fireball.gif, physics = propagate through max 5 walls, radius reduces 20% with every wall pass.

30 [0075] The fireball 610 can be virtually launched with initial state variables to indicate an initial location, a speed and direction vector, as well as a size (i.e., radius). The initial state 610a indicates the initial location. The second state 610b indicates the

fireball 610 has moved in a direction toward the first mobile device 602. While the different states 610a-f are shown in discrete intervals in FIG. 6, the rate of data transfer over the inbound and outbound communication paths 410, 411, as well as other processing capabilities, may allow for a more continuous transition from state to state.

5 The object interaction engine module 162 can be configured to determine when the fireball 610 interacts with the context 300. For example, the object interaction engine module 162 can indicate that the fireball 610 is interacting with the structural features in the map (e.g., walls, doors, windows). The results of the interaction can change the state of the fireball 610 such as reducing the size of the fireball 610 based on weighting
10 information associated with the context features. For example, the fireball at a third state 610c interacts with a wall and thus is reduced in size as indicated in a fourth state 610d. The type and/or nature of a structural feature (e.g., context data 180, weighting information 182) can impact the results of the interaction calculations. Solid walls may reduce the size more significantly than windows or cubical dividers. For example, the
15 state change from the fourth state 610d to a fifth state 610e can be small based on the weighting information and/or context information for the area. The fireball 610 can have a maximum range (e.g., a sixth state 610f) and an area of effect based on a target area. The interaction area 606 is determined based on the state of the fireball 610 as compared to the state of the first mobile device 602. State variables associated with the
20 location of an AV object and a mobile device can be compared. In an example, the first mobile device 602 is in the path of the fireball 610 and thus is virtually hit. The second mobile device 604 moves to a new a new position and is not hit (e.g., does not interact with) the fireball 610.

[0076] In an example, the state of a mobile device can include application parameters,
25 that will impact in interaction calculations. Continuing the fantasy fireball example of FIG. 6, the first mobile device 602 can include state variables to indicate the mobile device is shielded, or otherwise protected, from an interaction with the fireball 610. Other state variables based on the specific AV object application can be used. In an example, the fireball 610 can have a Lock-On target aspect which will attempt to target
30 a specific user and/or mobile device. The context or weighting information can be used to determine if a mobile device is behind adequate cover (i.e., a thick wall which will block the fireball 610).

[0077] Referring to FIG. 7, a top view of a context with an exemplary expanding threat is shown. The view includes a context 300, mobile devices 702a-c, and an expanding threat AV object 710. In an example, the expanding threat 710 can represent flooding or a swarm of insects which expands from a seed location through the context 300. The rate and direction of the expansion can be constrained by the structural elements in the context 300. The context data 180 and/or weighting information 182 can be used to constrain the spread of the threat. The object library 192 and associated items 176 may include:

- Flood object = water.gif, physics = propagate through corridors and walls, volume reduces with distance

For example, a flood game application may require the users (e.g., the mobile devices 702a-c) to avoid contact with the expanding threat 710. In an embodiment, the physics engine module 254 executing on the mobile device 702a can determine a current state of the expanding threat 710 and provide the threat state information to the AV server 110.

The AV server can distribute the threat state information to additional mobile devices (e.g., the mobile devices 702b, 702c). In an embodiment, the AV server 110 can be configured to determine the state of the expanding threat 710 and distribute the state information to the mobile devices 702a-c.

[0078] Referring to FIG. 8, with further reference to FIG. 7, a top view of a context with an exemplary expanding threat and virtual tools AV objects is shown. The view includes a context 300, an expanding threat 710, a first mobile device 802a, a second mobile device 802b, a third mobile device 802c, a first set of virtual sandbags 810a, and a second set of virtual sandbags 810b. In an example, the expanding threat 710 is virtual flood AV object which is constrained by the context 300. A user may interact with virtual tools to help further constrain the flooding. For example, the virtual tools can represent virtual sandbags such as the first set virtual sandbags 810a and the second set of virtual sandbags 810b. In a training application, a user may be required to virtually pick up a virtual sandbag in a first location and then transport it to a second location. Such a virtual object can include a geofence around the object to determine when a mobile device is within a required proximity of the virtual object. The size (e.g., strictness, tolerance) of the geofence can depend on the positioning quality of a given venue. Referring to the example in FIG. 8, the first mobile device 802a is located near

(i.e., within the geofence) the first set of virtual sandbags 810a. The object interaction engine module 162 can adjust the state of the mobile device to indicate that one or more virtual sandbags from the first set of virtual sandbags 810a has been virtually picked up. The second mobile device 802b is located in an area that is near, or in the path of, the
 5 expanding threat 710. The state of the second mobile device can be modified to indicate that one or more virtual sandbags have been virtually dropped. The second set of virtual sandbags 810b represent virtual tools which will interact with the virtual threat. In the flooding example, the second set of virtual sandbags block, or otherwise constrain, the path of the expanding threat 710. In a game application, an object can be for one team
 10 to constrain the expanding threat to the detriment of an opposing team (e.g., the third mobile device 802 is about to be trapped by the flood). The expanding flood and virtual sandbags are exemplary only, and not limitation, as other AV objects can be used to represent virtual threats and virtual tools (e.g., a swarm of insects/bug spray, roving aliens/laser pistols, virtual tiger/cage door levers).

15 **[0079]** Referring to FIG. 9, a top view of a context with an exemplary virtual trainer AV object is shown. The view includes a context 300, a mobile device 902 and a virtual trainer 910. The virtual objects are not limited to projectiles and expanding threats as previously described. Other virtual objects can be developed to interact with the user based on location and context. For example, a virtual trainer 910 can be configured to
 20 ambush (e.g., interact) with a user at a particular location with little or no warning (i.e., in an effort to keep the user's senses sharp). The ambush may lead to a first mini-game on the mobile device 902 followed by a chase through the context 300. The results of the interaction event, and/or a subsequent second mini-game, may change the state of the mobile device 902 and/or the virtual trainer 910. The virtual trainer 910 may hide in
 25 another location waiting to be found (e.g., hide and seek). Other virtual beings can be utilized.

[0080] Referring to FIG. 10, a top view of a context with multiple control areas and exemplary roving threat AV objects is shown. The view includes a context 300, a first mobile device 1002a, a second mobile device 1002b, a third mobile device 1002c, a first
 30 control area 1004a, a second control area 1004b, a third control area 1004c, a fourth control area 1004d, a first roving threat 1010a, a second roving threat 1010b and a third roving threat 1010c. Each roving threat 1010a-c includes a line-of-sight (LOS) indicator 1011a-c respectively. The number, nature, and locations of the control areas

and the roving threats are exemplary only and not a limitation. The number of mobile devices is exemplary as a game application can be a single player or multiplayer configuration. In operation, the object library 192 can include virtual objects which define control areas within a context. The control areas can correspond to rooms, or be
5 bounded by other structural features as determined by the context data 180 and/or the weighting information 182. In an example, a context can be converted into a virtual starship such that each of the control areas represents an iconic control station. The first control areal 1004a can represent a bridge, the second control area 1004b can represent a weapons station, the third control area 1004c can represent an engineering area, and a
10 fourth control area 1004d can represent a store room. The object interaction engine module 162 can be configured to compute an interaction result based on the location of a mobile device in relation to the control areas. For example, the first mobile device 1002a is within the first control area 1004a and the resulting interaction result may be to initiate a bridge related mini-game (e.g., a first mini-game). The second mobile device
15 1002b is located within the third control area 1004c and the resulting interaction result may be to initiate an engineering related mini-game (e.g., a second mini-game). Other interaction results may be used and the corresponding states of the mobile devices and the control areas may be updated.

[0081] In an embodiment, the corridors between the control areas 1004a-d can include
20 one or more roving threats 1010a-b. Continuing the science fiction theme, the roving threats can represent virtual robot aliens that move about the context 300. The movement of the roving threats 1010a-c is constrained by the context data 180 and/or the weighting information 182. Each of the roving threats 1010a-c may include a LOS indicator 1011a-c (i.e., a line of sight variable) to indicate a general direction of interest
25 for the roving threat. In operation, the object interaction engine module 162 can be configured to compute an interaction result based on the location of the roving threat, as well as the current LOS indicator. The interaction result may be to initiate a roving threat related mini-game on the mobile device. In a game application, the states of the mobile devices, control areas, and the roving threats can be used to create a scoring
30 mechanism. For example, the game may create a mission for the users to complete, and then subsequently create emergencies for the users to react to. The users may have to move about the context from control area to control area to complete the mission. Virtual tools may also be included in the game application. These rules and objectives

are exemplary only as other game applications can be created to utilize context specific control areas and roving threats.

[0082] Referring to FIG. 11, with further reference to FIG. 10, a top view of a mobile device executing one or more mini-games is shown. The view includes a mobile device
5 1100 configured to execute and display a map application 1102a, a bridge mini-game application 1102b, a weapons mini-game application 1102c, and an engineering mini-game application 1102d. The map applications 1102a-d are exemplary only in view of the science fiction theme described with regards to FIG. 10. In operation, AV server 110 can store the map applications 1102a-b in the database 114 and provide the map
10 applications 1102a-d to the mobile device 1100. The map applications 1102a-d can be initiated based on results computed by the object interaction engine module 162. For example, an application can be initiated when the mobile device 1100 enters a corresponding control area. The mini-game applications can be themed to the respective control area. The bridge mini-game may require the user to complete a flight
15 simulator, the weapons mini-game may require the user to complete a target shooting game, and the engineering game may be a technical trivia game. A mini-game may include interacting with an AV object in an augmented reality application. For example, a user may use the camera view of the mobile device 1002c to obtain a view of the corridor leading to the roving threat 1010a. The display of the mobile device
20 1002c may concurrently show a live image of the corridor with an alien robot travelling down the corridor wherein the location of the alien robot is based on the state variables. The object of such a mini-game may be to virtually power-down the alien robot (e.g., via a tap on the robot image on the mobile device screen). Other mini-games may be used. The user's performance on a particular mini-game may change the state of the
25 mobile device, the control area, or other game application related state model (e.g., location of the starship, damage conditions, threat level). The results of the mini-games may also be used to temporarily modify the context data based on the theme (e.g., close a door, open an air lock, activate a transporter).

[0083] Referring to FIG. 12, with further reference to FIGS. 1-11, a process 1200 for
30 displaying an AV object includes the stages shown. The process 1200 is, however, an example only and not limiting. The process 1200 can be altered, e.g., by having stages added, removed, rearranged, combined, performed concurrently, and/or having single stages split into multiple stages. For example, stage 1206 described below of receiving

AV object information can be performed before stages 1202 and 1204. Still other alterations to the process 1200 as shown and described are possible.

[0084] At stage 1202, the mobile device 200, or the AV server 110, determines the location of the mobile device 200. The mobile device 200 may obtain wireless
5 positioning signals from satellite and/or terrestrial based positioning systems. A trilateration based approach may be utilized by the mobile device 200 to determine the location of the mobile device. The device state module 256 can be updated with the location information and a collection of mobile device state variables can be provided to the AV server 110, or stored locally on the mobile device 200. In an embodiment,
10 additional actions may be performed at the beginning of the process 1200. For example, a user can initiate a game, set a difficulty level, exchange game parameters, perform user authentication, or determine billing information. Other applications or operational steps may also be performed.

[0085] At stage 1202, the mobile device 200 or the AV server 110 determines a
15 context. The context can be a floor map with structural features (including interaction values) stored as context data 180 in the database 114. The context can include weighting information 182 such as a RSSI / RTT heatmap. Additional media content items 178 can be included in the context. The context can be provided to the mobile device 200 based on the location determined at stage 1202, or can be previously stored
20 on the mobile device 200 and accessed based on the location of the mobile device 200.

[0086] At stage 1206, the mobile device 200 can obtain AV object information from the AV server or from local memory. The AV object information can be stored in the object library 192 and can include one or more associated items 176 such as a media content item, an object physic engine, and metadata 186. The AV object information
25 can also include a set of state variables which represent the current state of the AV object. The state variables can represent the location, size, movement speed and direction, interaction variables, and other information to model the actions of the AV object. The AV object information can be provided to the mobile device 200 based on the location determined at stage 1202. For example, the AV object information can be
30 transmitted from the AV server 110 to the mobile device over the outbound communication path 410. The AV object can be previously stored on the mobile device

200 and persist in the storage media 210 (i.e., before the location is determined at stage 1202).

[0087] At stage 1208, the mobile device 200 is configured to display the AV object information in relation to the context. The mobile device 200 includes a graphical display 222 and an AV view 234 configured to display the context (e.g., a floor map, camera view) with an AV object as well as icons representing other mobile devices located in the context. For example, referring to FIG. 5, the context 300 is displayed with the virtual ball 510 and the mobile devices 502. The location of the AV object can be included as a state variable. The AV object state variables can be updated based on the execution of algorithms within the physics engine module 254 and or the object interaction engine module 162. The graphical display 222 can be configured to display the AV object based at least in part on the values of the state variables.

[0088] At stage 1210, the mobile device 200, or the AV server 110, are configured to detect an interaction. In an example, the object interaction engine module 162 can process the AV object information, the context information (e.g., context data 180 and/or weighting information 182), and the device state module 256 to detect an interaction. An interaction may occur, for example, when the AV object information indicates the AV object is near a structural element in the context (e.g., when a virtual ball bounces off the wall as depicted in FIG. 5). An interaction may be detected with the AV object information indicates the AV object is near a mobile device. An interaction may be detected based on gestures, or other device state information, input into the mobile device 200.

[0089] At stage 1212, the mobile device 200, or the AV server 110, are configured to modify the AV object information based on the interaction. The object interaction engine module 162 and/or physics engine module 254 are configured to compute an interaction result based on the detected interaction. The interaction result may indicate the speed and direction the AV object moves after the interaction. For example, a modification may be to reduce the value of AV object's speed state variable (i.e., decrease the virtual speed) based on the interaction results. The interaction result may indicate the size of the AV object is decreased after the interaction (i.e., modify the size state variable). Other state variables associated with AV object can be modified based on the interaction result. In the expanding threat example of FIG. 7, the interaction

result may constrain the expansion of the AV object. The interaction result can be based on an interaction with the context 300, the mobile device 802c, a virtual tool (e.g., virtual sandbags 810b), or on combinations therein. The modified AV object information can be distributed to other mobile devices via the network 140.

5 [0090] At stage 1214, the mobile device 200 is configured to display the modified AV object information. The modified AV object information can include changes based on the modified state variables (e.g., changes in speed, direction, size), as well as changes in media content items 178 (e.g., graphic files, LOS indicators 1011). The changes in the media content items 178 can correspond to changes in the state variables (i.e., a state
10 variable can be a pointer to a media content item). Displaying the modified AV object information can occur on a single mobile device, or on multiple mobile devices on the network 140.

[0091] Referring to FIG. 13A, with further reference to FIGS. 1-4, a process 1300 for determining interaction variables includes the stages shown. The process 1300 is,
15 however, an example only and not limiting. The process 1300 can be altered, e.g., by having stages added, removed, rearranged, combined, performed concurrently, and/or having single stages split into multiple stages.

[0092] At stage 1302, the AV server 110 can obtain context data 180. A developer can utilize a developer device 420 to access the database 114 and store the context 300. The
20 context can include, for example, a layout of a building having several areas. The context 300 can be a data file (e.g., .dwg, BIMXML) stored on the AV server 110 or a mobile device. The areas can be physically separated areas such as rooms, courtyards, hallways, and the like, or areas separated by non-physical boundaries, such as areas on a grid imposed on a map.

25 [0093] At stage 1304, the AV server 110 can be configured to determine location data corresponding to one or more areas within the context. The instructions 116 can include computer-readable instructions to cause the processor 118 to segment the context into areas. The segmentation can be based on structural features (e.g., walls, doors, windows) as well as an analysis of open space to determine the location of rooms,
30 courtyards, hallways, and the like.

[0094] At stage 1306, the AV server 110 can be configured to determine weighting information for each of the areas. In an example, the weighting information can be included in an architectural file as a property and/or attribute of a specific architectural feature (e.g., material, size, radius, thickness, etc...). The weighting information may be based on the number of architectural features (e.g., walls, composition of walls, distance) and other factors that can impact RSSI and RTT signal information. The weighting information can be determined based on results provided from one or more mobile devices in the context. Referring to FIG. 3, for example, the access point 302 can capture RSSI and RTT signal information for the areas occupied by each of the mobile devices 304a, 304b, 304c. Weighting information corresponding to each of the areas can be stored in the database 114. In an example, the RSSI and/or RTT information may be correlated to an architectural feature property such as a wall material. That is, an area within a context with low signal (i.e., high loss) may be correlated to a 'concrete' material, an area with a medium signal (i.e., medium loss) may be correlated to a 'drywall' material, and an area with a high signal (e.g., low loss) may be correlated to an open space. The weighting information for the context can be updated and refined over time as more mobile devices enter the context (e.g., crowdsourcing). The weighting information can be a heatmap which can overlay the physical layout of the context.

[0095] At stage 1308, the AV server 110 can be configured to determine interaction variables based on the weighting information. The interaction variables can be state variables and/or functions that are associated with the weighting information. The interaction variables may be based on the composition of a structural feature (e.g., a hard wall, a door, an open corridor), or the signal strength based on RSSI and/or RTT data. The interaction variables can be used to model the interaction between an AV object and an architectural feature within the context. For example, a virtual ball 510 may bounce off of a wall with a 'brick' material property, and break through a wall with a 'glass' material property. The interaction variables can be functions to determine, for example, the speed and direction of the virtual ball 510 as a result of the bounce or the break.

[0096] In an embodiment, the composition and placement of walls can include a virtual component and the interaction variables may be based on the virtual component. For example, the weighting information for certain walls may indicate brick or concrete.

This actual weighting information may be virtually designated to be 'stone' in a fantasy setting, or 'permacrete' in a science fiction setting. The interaction variables for the virtual designations may be used in the application. Other architectural or environmental features may be designated such virtual bookcases, tapestries, blast doors, or other virtual structures. The virtual characteristics may be based on the properties of the real feature. For example, the thickness and corresponding durability of a virtual stone wall may be relative to the characteristics (i.e., material, size, location, configuration) of real walls in the context. In operation, the object interaction engine module 162 can utilize the interaction variables or functions, as well as the feature properties, to determine an interaction result with an AV object. For example, referring to FIG. 6, a high value for an interaction variable may cause the state of the fireball 610 to reduce in size by a large amount, while a low value for an interaction variable may cause the state of the fireball 610 to reduce its size by a small amount. The impact of the interaction variables or functions may be different for different AV objects.

[0097] Referring to FIG. 13B, an exemplary data structure for weighting information and interaction variables is shown. The database 114 can include a collection of context data 180 and weighting information 182. In an example, the context data 180 can be electronic representations of the context 300. Examples of the electronic representations may include, but are not limited to, Computer Automated Design (CAD) files, BIM files (e.g., BIMXML), and other proprietary files. The example context data record 180a of FIG. 13B utilizes a BIMXML format including an attribute tag (i.e., 'Attrib Value' tag). The tag is exemplary only and not a limitation. The attribute tag may be used to reference weighting information 182. For example, the context data record 180a indicates that the material of the context feature is 'brick.' The weighting information 182 may be a collection of records in a database, or other file structure, to correlate feature properties (e.g., attribute tags) with interaction variables or functions. As an example, a weighting information data table 182a can include a list of AV Objects 1320 and a collection of interaction variables or functions. The weighting information data table 182a can be indexed by the architectural features (e.g., attribute tags) to be used, such as the material value example in FIG. 13B. For a given attribute tag (e.g., Null, Brick, Glass, Drywall, Concrete, etc...), and an AV Object (e.g., Ball, Water, Fire Ball, Robot), an interaction variable 1322 can be determined. For example, a Ball will interact with a Brick by utilizing the interaction function $B(v,d)$. The

function $B(v,d)$ can be a call to a function within the object interaction engine module 162. The function call may include parameters such the velocity (v) and direction (d) vector of the example. Other parameters and other functions may be used for the various AV Object and weighting information values.

5 [0098] In an embodiment, the weighting information 182 may be indexed by signal parameters provided by the mobile device. For example, the attribute tags (e.g., Null, Brick, Glass, Drywall, Concrete) may be replaced with signal strength values, or other information based on the RSSI and/or RTT data. In this example, the selection of an interaction variable 1322 is based on AV Object and the signal parameters. In an
10 embodiment, the attribute tags in the weighting information data table 182a can be augmented by the signal parameters such that the selection of the interaction variable is based on the AV Object and a combination of the attribute tag and the signal parameters.

[0099] Referring to FIG. 14A, with further reference to FIG. 4, a process 1400 for
15 providing position and gesture updates to an AV server includes the stages shown. The process 1400 is, however, an example only and not limiting. The process 1400 can be altered, e.g., by having stages added, removed, rearranged, combined, performed concurrently, and/or having single stages split into multiple stages. For example, stage 1406 described below of receiving AV object information and peer position updates can
20 be performed before stage 1406. Still other alterations to the process 1400 as shown and described are possible.

[0100] At stage 1402, a mobile device 402a registers with the AV server 110 over the inbound communication path 411a. The registration process can be used to indicate that the mobile device 402a is ready to receive AV object information. The registration
25 process may include security procedure (e.g., passwords), and may automatically occur when the mobile device 402 joins the network 140 or enters a context 300.

[0101] At stage 1404, the mobile device 402 receives AV object information and peer position updates from the AV server 110 via the outbound communication path 410a. The AV object information can persist in the database 114 and can be provided to the
30 mobile device 402a after the registration in stage 1402. The AV object information can be previously stored in the database 212 on the mobile device 402a. The AV server 110 can be configured to receive peer position updates from other mobile devices (e.g., the

mobile devices 402b, 402c) via the inbound communication paths 411b, 411c and provide the peer position updates to them the mobile device 402a via the outbound communication path 410a. The AV object information and peer position updates can be one or more state variables associated with the AV object and the mobile devices 402b,
5 402c respectively.

[0102] At stage 1406, the mobile device 402a provides position and gesture updates to the AV server 110 via the inbound communication path 411a. The position and gesture updates can be one or more state variables in the device state module 256. The object interaction engine module 162 can utilize the position and gesture updates to determine
10 interaction events and/or update the AV object information. The AV server 110 can distribute the AV object information and position updates to the other mobile devices 402b, 402c as described at stage 1404.

[0103] Referring to FIG. 14B, with further reference to FIG. 4, a process 1450 for providing AV object position updates to an AV server includes the stages shown. The
15 process 1450 is, however, an example only and not limiting. The process 1450 can be altered, e.g., by having stages added, removed, rearranged, combined, performed concurrently, and/or having single stages split into multiple stages.

[0104] At stage 1452, a mobile device 402a registers with the AV server 110 over the inbound communication path 411a. The registration process can be used to indicate that
20 the mobile device 402a is ready to receive AV object information. The registration process may include security procedure (e.g., passwords), and may automatically occur when the mobile device 402 joins the network 140 or enters a context 300.

[0105] At stage 1454, the mobile device 402a receives AV object information including an AV object physics engine module 174 from the AV server 110. The object physics
25 engine module 174 executes locally (i.e., on the mobile device 402a) in the physics engine module 254. In operation, the physics engine module 254 simulates the motion of the AV object and updates the location or other AV object state variables. The rate of the updates can be based on specifications the processor 216 and the available bandwidth of the network 140. The physics engine module 254 and the object
30 interaction engine module 162 may be used determine the results of interactions between an AV object and the context. For example, the context and corresponding

weighting information can be stored on the mobile device and available for processing by the physic engine and object interaction engine modules.

[0106] At stage 1456, the mobile device 402a provides AV object state variable updates to the AV server via the inbound communication path 411b. The AV server can
5 broadcast the updated AV object state variables to the other mobile devices 402b, 402c. The AV server 110 can be configured to selectively provide updates to the other mobile devices 402b, 402c based on their corresponding state variables (e.g., position, game specific conditions).

[0107] Referring to FIG. 15, with further reference to FIG. 5, a process 1500 for
10 modifying the state variables for an AV object includes the stages shown. The process 1500 is, however, an example only and not limiting. The process 1500 can be altered, e.g., by having stages added, removed, rearranged, combined, performed concurrently, and/or having single stages split into multiple stages. For example, stage 1510 described below for determining a mobile device interaction result can be performed
15 before stages 1506 and 1508. Still other alterations to the process 1500 as shown and described are possible.

[0108] At stage 1502, a mobile device 502c or an AV server 110 can be configured to determine a context based on the location of the mobile device 502c. For example, the context 300 can be a floor diagram of an office building. The AV server 110 can
20 provide the context 300 to the mobile device 502c over the network 140 when the mobile device enters a location within the context 300. The context 300 may be previously stored on the mobile device 502c.

[0109] At stage 1504, a mobile device 502c or an AV server 110 can be configured to generate an initial state for an AV object. For example, the initial state of the virtual
25 ball 510 includes a position 'A' with initial direction and speed information. The initial state of the AV object can be generated by the mobile device.

[0110] At stage 1506, the object interaction engine module 162, executing on mobile device 502c or on the AV server 110, can be configured to process the state of an AV object with context data 180 and/or weighting information 182 (e.g., interaction
30 variables). An output from the object interaction engine module 162 can be an interaction result. For example, with the virtual ball 510 is located at position 'C', the

result is to modify (i.e., change the value of) the state variables associated with the direction and speed of the virtual ball 510 at stage 1508. Other interaction results are possible. For example, the context interaction result may reduce the size of the AV object (e.g., fireball 610), or constrain the AV object (e.g., expanding threat 710).

- 5 [0111] At stage 1510, the object interaction engine module 162, executing on mobile device 502c or on the AV server 110, can be configured to process the state of the AV object with mobile device state information. For example, when the virtual ball 510 reaches position 'E', the user of the mobile device 502c can enter a gesture, or other sensor input, to the mobile device 502c and modify the state of the mobile device 502c.
- 10 The interaction result may modify the state of the AV object at stage 1512. The interaction result may also modify the state of the mobile device 502c and/or the state of the context 300. For example, if the virtual ball 510 is a boulder, the interaction result may be to temporarily remove a wall or other structure in the context 300. The modified AV object can be displayed on the mobile device 502c, as well as on the other
- 15 mobile devices 502a, 502b.

- [0112] Referring to FIG. 16, with further reference to FIG. 8, a process 1600 for modifying the state variables of an AV object based on context or tool interaction includes the stages shown. The process 1600 is, however, an example only and not limiting. The process 1600 can be altered, e.g., by having stages added, removed,
- 20 rearranged, combined, performed concurrently, and/or having single stages split into multiple stages.

- [0113] At stage 1602, a mobile device 802a or an AV server 110 can be configured to determine a context based on the location of the mobile device 802a. For example, the context 300 can be a floor diagram of an office building. The context 300 can be
- 25 provided to the mobile device 802a over the network 140 when the mobile device enters a location within the context 300. The context 300 can be previously stored on the mobile device 802a.

- [0114] At stage 1604, a mobile device 802a or an AV server 110 can be configured to generate a state for a virtual threat. The initial state of the virtual threat may include an
- 30 initial position and movement variables. For example, the expanding threat 710 can include a seed position and a flood rate variable to determine how fast the expanding

threat 710 will flood an area. The initial state may also include variables to indicate the boundaries (e.g., a polygon) of the expanding threat 710.

[0115] At stage 1606, an object interaction engine module 162, executing on the mobile device 802a or on the AV server 110, can be configured to determine a context
5 interaction result. For example, the context 300 includes wall which will constrain the expanding threat 710. The context interaction result can modify the boundary variables in the expanding threat 710 state information. Other context interaction results may also be determined.

[0116] At stage 1608, a mobile device 802a or an AV server 110 can be configured to
10 generate a state for a virtual tool. The initial state of the virtual tool may include an initial position and an interaction variable. For example, the virtual sandbags 810a can be located in the context 300 in an area that is separate from the area occupied by the expanding threat. An object of a gaming application may be to require the users to relocate a virtual tool from a first location to a second location. A user may move a
15 virtual tool into a position to cause an interaction with another AV object. The mobile device 802b has moved the virtual sandbags 810b to interact with the expanding threat 710.

[0117] At stage 1610, an object interaction engine module 162, executing on the mobile device 802a or on the AV server 110, can be configured to determine a tool interaction
20 result. For example, the state of the virtual sandbags 810b and the state of the expanding threat 710 can be processed by the object interaction engine module 162 to determine the tool interaction result. The tool interaction result may modify the boundary state variables of the expanding threat 710.

[0118] At stage 1612, the state of the virtual threat can be modified based on the context
25 interaction result and/or the tool interaction result. The sandbag virtual tool is exemplary only and not a limitation. Other virtual tools can be used. For example, continuing the flooding example, a virtual tool could be a virtual drain, or a virtual freeze ray, or a virtual sponge. The order of calculation for the context interaction result and the tool interaction result are exemplary only. The results be calculated serially or
30 in parallel. In some applications, a synergy between a context area and a virtual tool can be used (e.g., the combined effect is greater than the individual effects added

individually). An application can have multiple virtual tools each with different interaction results for a given virtual threat.

[0119] Referring to FIG. 17, with further reference to FIG. 10, a process 1700 for process for initiating a mini-game on a mobile device includes the stages shown. The
5 process 1700 is, however, an example only and not limiting. The process 1700 can be altered, e.g., by having stages added, removed, rearranged, combined, performed concurrently, and/or having single stages split into multiple stages.

[0120] At stage 1702, a mobile device 1002a or an AV server 110 can be configured to determine a context based on the location of the mobile device 1002a. For example, the
10 context 300 can be a floor diagram of an office building. The context 300 can be provided to the mobile device 1002a over the network 140 when the mobile device enters a location within the context 300. The context 300 can be previously stored on the mobile device 1002a.

[0121] At stage 1704, a mobile device 1002a or an AV server 110 can be configured to
15 establish one or more control areas within the context. A control area can correspond to rooms, courtyards, hallways, and the like, or areas separated by non-physical boundaries, such as areas on a grid imposed on the context 300. A control area can be an AV object with corresponding state variables. The AV server 110 can broadcast the control area to other mobile devices (e.g., 1002b, 1002c).

[0122] At stage 1706, a mobile device 1002a or an AV server 110 can be configured to
20 generate a state for one or more roving threats. As an example, and not a limitation, an initial state of the roving threat may include an initial position, movement and line of sight variables. The movement and line of sight for the roving threat can be constrained by the context. For example, referring to FIG. 10, the roving threat 1010a represents an
25 alien robot configured to roam the corridors of the context 300. The roving threat 1010a can randomly change its LOS indicator 1011a in different directions. The state variables for the roving threat can be updated to indicate the direction of the LOS indicator 1011a.

[0123] At stage 1708, a mobile device 1002a or an AV server 110 can be configured to
30 determine an engagement zone for each of the roving threats. For example, an engagement zone may be a defined area around a roving threat. The engagement zone

may include LOS information, thus allowing a user to sneak-up behind the roving threat without being detected. The engagement zone can be constrained by the context 300.

The roving threat LOS may be blocked by walls or other structural items as defined in the context data 180 and/or the weighting information 182. At stage 1710, the state of the mobile device 1002a can be determined. The object interaction engine module 162 can be configured to compare the location of the mobile device with the location of the engagement zones for one or more roving threats at stage 1714. If the comparison indicates that the mobile device is within the roving threat engagement zone, the mobile device can be configured to execute a mini-game application at stage 1716. For example, a user may have to complete a combat related mini-game to attempt to defeat, or flee from, a roving threat. The results of the mini-game can impact the state variables for the mobile device and the roving threat. For example, the roving threat may have a damage variable which can be reduced as a result of a mini-game.

[0124] At stage 1712, the object interaction engine module 162 can be configured to determine if the current location of a mobile device is within a control area. For example, referring to FIG. 10, the mobile device 1002a is within the control area 1004a. While the mobile device 1002a is within the control area 1004a, a mini-game may be presented to the user at stage 1716. The mini-game may correspond to the control area (e.g., the simulated bridge example described above), and the results of the mini-game may impact the state variables of the mobile device, the control area, or other application based data fields.

[0125] As used herein, including in the claims, “or” as used in a list of items prefaced by “at least one of” indicates a disjunctive list such that, for example, a list of “at least one of A, B, or C” means A or B or C or AB or AC or BC or ABC (i.e., A and B and C), or combinations with more than one feature (e.g., AA, AAB, ABBC, etc.).

[0126] As used herein, including in the claims, unless otherwise stated, a statement that a function or operation is “based on” an item or condition means that the function or operation is based on the stated item or condition and may be based on one or more items and/or conditions in addition to the stated item or condition.

A wireless communication network does not have all communications transmitted wirelessly, but is configured to have at least some communications transmitted wirelessly.

[0127] Other examples and implementations are within the scope and spirit of the disclosure and appended claims. For example, due to the nature of software, functions described above can be implemented using software (e.g., computer program product) executed by a processor, hardware, firmware, hardwiring, or combinations of any of these. Such software can be a computer program product residing on a processor-executable storage medium. Features implementing functions may also be physically located at various positions, including being distributed such that portions of functions are implemented at different physical locations.

[0128] Further, more than one invention may be disclosed.

10 [0129] Substantial variations to described configurations may be made in accordance with specific requirements. For example, customized hardware might also be used, and/or particular elements might be implemented in hardware, software (including portable software, such as applets, etc.), or both. Further, connection to other computing devices such as network input/output devices may be employed.

15 [0130] Common forms of physical and/or tangible computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punch cards, paper tape, any other physical medium with patterns of holes, a RAM, a PROM, EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or
20 any other medium from which a computer can read instructions and/or code. For example, the computer-readable media can be computer program product residing on a processor-executable storage medium, the a computer program product including processor-executable instructions configured to cause a processor to perform one or more features described herein.

25 [0131] The methods, systems, and devices discussed above are examples. Various configurations may omit, substitute, or add various procedures or components as appropriate. For instance, in alternative configurations, the methods may be performed in an order different from that described, and that various steps may be added, omitted, or combined. Also, features described with respect to certain configurations may be
30 combined in various other configurations. Different aspects and elements of the configurations may be combined in a similar manner. Also, technology evolves and,

thus, many of the elements are examples and do not limit the scope of the disclosure or claims.

[0132] Specific details are given in the description to provide a thorough understanding of example configurations (including implementations). However, configurations may be practiced without these specific details. For example, well-known circuits, processes, algorithms, structures, and techniques have been shown without unnecessary detail in order to avoid obscuring the configurations. This description provides example configurations only, and does not limit the scope, applicability, or configurations of the claims. Rather, the preceding description of the configurations provides a description for implementing described techniques. Various changes may be made in the function and arrangement of elements without departing from the spirit or scope of the disclosure.

[0133] Also, configurations may be described as a process which is depicted as a flow diagram or block diagram. Although each may describe the operations as a sequential process, many of the operations can be performed in parallel or concurrently. In addition, the order of the operations may be rearranged. A process may have additional stages or functions not included in the figure. Furthermore, examples of the methods may be implemented by hardware, software, firmware, middleware, microcode, hardware description languages, or any combination thereof. When implemented in software, firmware, middleware, or microcode, the program code or code segments to perform the tasks may be stored in a non-transitory computer-readable medium such as a storage medium. Processors may perform the described tasks.

[0134] Components, functional or otherwise, shown in the figures and/or discussed herein as being connected or communicating with each other are communicatively coupled. That is, they may be directly or indirectly connected to enable communication between them.

[0135] Having described several example configurations, various modifications, alternative constructions, and equivalents may be used without departing from the spirit of the disclosure. For example, the above elements may be components of a larger system, wherein other rules may take precedence over or otherwise modify the application of the invention. Also, a number of operations may be undertaken before,

during, or after the above elements are considered. Accordingly, the above description does not bound the scope of the claims.

[0136] A statement that a value exceeds (or is more than) a first threshold value is equivalent to a statement that the value meets or exceeds a second threshold value that is slightly greater than the first threshold value, e.g., the second threshold value being one value higher than the first threshold value in the resolution of a computing system. A statement that a value is less than (or is within) a first threshold value is equivalent to a statement that the value is less than or equal to a second threshold value that is slightly lower than the first threshold value, e.g., the second threshold value being one value lower than the first threshold value in the resolution of a computing system.

WHAT IS CLAIMED IS:

1. A method for providing augmented virtuality (AV) on a mobile device, comprising:
 - determining a location of the mobile device;
 - 5 determining a context based on the location;
 - obtaining an AV object information;
 - displaying the AV object information in relation to the context;
 - detecting an interaction with the context;
 - modifying the AV object information based on the interaction; and
 - 10 displaying a modified AV object information.
2. The method of claim 1 wherein the context includes weighting information.
- 15 3. The method of claim 2 wherein the weighting information is based on Received Signal Strength Indication (RSSI) or Round-Trip Time (RTT) data.
4. The method of claim 2 wherein the weighting information is associated with a composition of a physical object in the context.
- 20 5. The method of claim 1 comprising:
 - receiving a gesture from a user; and
 - modifying the AV object information based on a received gesture.
- 25 6. The method of claim 5 comprising:
 - providing an alert signal to the user; and
 - modifying the AV object information based on a reaction time, wherein
 - the reaction time is a difference between a time the alert signal is provided and a time
 - the gesture is received.
- 30 7. The method of claim 1 comprising:
 - receiving the AV object information from an augmented virtuality (AV)
 - server; and

providing the modified AV object information to the AV server.

8. A method for providing augmented virtuality (AV) on a mobile device, comprising:

- 5 determining a context based on a location of the mobile device;
 generating a state for an AV object;
 determining a context interaction result;
 modifying the state of the AV object based on the context interaction
 result;
10 determining a mobile device interaction result;
 modifying the state of the AV object based on the mobile device
 interaction result; and
 displaying the AV object.

- 15 9. The method of claim 8 wherein the state of the AV object
 includes a direction variable.

10. The method of claim 9 wherein modifying the state of the AV
 object based on the context interaction result comprises modifying the direction
20 variable.

11. The method of claim 9 wherein modifying the state of the AV
 object based on the mobile device interaction result comprises modifying the direction
 variable.

25

12. The method of claim 8 wherein the state of the AV object
 includes a size variable.

13. The method of claim 12 wherein modifying the state of the AV
30 object based on the context interaction result comprises modifying the size variable.

14. The method of claim 8 wherein determining the mobile device
 interaction result comprises receiving a gesture from a user.

15. The method of claim 8 comprising modifying a state of the mobile device based on the mobile device interaction result.

5 16. A method of using virtual tools in Augmented Virtuality (AV) applications, comprising:

determining a context based on a location of a mobile device;
generating a state for a virtual threat;
generating a state for a virtual tool;
10 determining a context interaction result;
determining a tool interaction result;
modifying the state of the virtual threat based on the context interaction result and the tool interaction result; and
displaying the virtual threat and the virtual tool.

15

17. The method of claim 16 wherein the state of the virtual tool includes a location variable.

18. The method of claim 17 comprising modifying the state of the
20 virtual tool based a state of the mobile device.

19. The method of claim 18 wherein determining the tool interaction result is based on a modified state of the virtual tool.

25 20. The method of claim 16 wherein the tool interaction result constrains the state of the virtual threat.

21. A method of determining interaction variables in an Augmented
Virtuality (AV) application, comprising:
30 obtaining a context;
determining location data corresponding to a plurality of areas within the context;
determining weighting information for each of the plurality of areas;

determining interaction variables based on the weighting information;
and
storing the interaction variables.

5 22. The method of claim 21 wherein the weighting information is
based on RSSI information.

 23. The method of claim 21 wherein the weighting information is
based on RTT information.

10

 24. A method for providing augmented virtuality (AV) on a mobile
device, comprising:

 determining a context based on a location of the mobile device;
 establishing a control area within the context;
15 generating a state for a roving threat, wherein the state of the roving
threat is constrained by the context;
 determining an engagement zone for the roving threat, wherein the
engagement zone is constrained by the context;
 determining a state of the mobile device, wherein the state includes a
20 current location; and
 initiating a first mini-game based on the state of the mobile device.

 25. The method of claim 24 wherein the first mini-game is initiated
when the location of the mobile device is within the control area.

25

 26. The method of claim 24 comprising initiating a second mini-
game when the location of the mobile device is within the engagement zone.

 27. The method of claim 24 wherein the engagement zone comprises
30 a line of sight variable.

 28. The method of claim 27 wherein the line of sight variable is
constrained by the context.

29. An apparatus for providing augmented virtuality (AV) on a mobile device, comprising:
- 5 a memory module;
- a processor coupled to the memory module and configured to:
- determine a location of the mobile device;
- determine a context based on the location;
- obtain an AV object information;
- 10 display the AV object information in relation to the context;
- detect an interaction with the context;
- modify the AV object information based on the interaction; and
- display a modified AV object information.
- 15 30. The apparatus of claim 29 wherein the context includes weighting information.
31. The apparatus of claim 30 wherein the weighting information is based on Received Signal Strength Indication (RSSI) or Round-Trip Time (RTT) data.
- 20 32. The apparatus of claim 30 wherein the weighting information is associated with a composition of a physical object in the context.
33. The apparatus of claim 29 wherein the processor is further
- 25 configured to:
- receive a user gesture; and
- modify the AV object information based on a received gesture.
34. The apparatus of claim 33 wherein the processor is further
- 30 configured to:
- provide an alert signal to a user; and

modify the AV object information based on a reaction time,
wherein the reaction time is a difference between a time the alert signal is provided and
a time the user gesture is received.

5 35. The apparatus of claim 29 wherein the processor is further
configured to:
 receive the AV object information from an augmented virtuality
 (AV) server; and
 provide the modified AV object information to the AV server.

10

 36. An apparatus for providing augmented virtuality (AV) on a
mobile device, comprising:
 a memory module;
 a processor coupled to the memory module and configured to:
15 determine a context based on a location of the mobile device;
 generate a state for an AV object;
 determine a context interaction result;
 modify the state of the AV object based on the context interaction
result;
20 determine a mobile device interaction result;
 modify the state of the AV object based on the mobile device
interaction result; and
 display the AV object.

25 37. The apparatus of claim 36 wherein the state of the AV object
includes a direction variable.

 38. The apparatus of claim 37 wherein the processor is further
configured to modify the direction variable based on the context interaction result.
30

 39. The apparatus of claim 37 wherein the processor is further
configured to modify the direction variable based on the mobile device interaction
result.

40. The apparatus of claim 36 wherein the state of the AV object includes a size variable.

5 41. The apparatus of claim 40 wherein the processor is further configured to modify the size variable based on the context interaction result.

42. The apparatus of claim 36 wherein the processor is further configured to receive a gesture from a user.

10

43. The apparatus of claim 36 wherein the processor is further configured to modify a state of the mobile device based on the mobile device interaction result.

15 44. An apparatus for using virtual tools in Augmented Virtuality (AV) applications, comprising:

a memory module;

a processor coupled to the memory module and configured to:

determine a context based on a location of a mobile device;

20

generate a state for a virtual threat;

generate a state for a virtual tool;

determine a context interaction result;

determine a tool interaction result;

25 modify the state of the virtual threat based on the context interaction result and the tool interaction result; and

display the virtual threat and the virtual tool.

45. The apparatus of claim 44 wherein the state of the virtual tool includes a location variable.

30

46. The apparatus of claim 45 wherein the processor is further configured to modify the state of the virtual tool based a state of the mobile device.

47. The apparatus of claim 46 wherein the processor is further configured to determine the tool interaction result based on a modified state of the virtual tool.

5 48. The apparatus of claim 44 wherein the tool interaction result constrains the state of the virtual threat.

49. An apparatus for determining interaction variables in an Augmented Reality (AR) application, comprising:
10 a memory module;
a processor coupled to the memory module and configured to:
obtain a context;
determine location data corresponding to a plurality of areas
within the context;
15 determine weighting information for each of the plurality of areas;
determine interaction variables based on the weighting information; and
store the interaction variables.

20 50. The apparatus of claim 49 wherein the weighting information is based on RSSI information.

51. The apparatus of claim 49 wherein the weighting information is
25 based on RTT information.

52. An apparatus for providing augmented reality (AR) on a mobile device, comprising:
a memory module;
30 a processor coupled to the memory module and configured to:
determine a context based on a location of the mobile device;
establish a control area within the context;

generate a state for a roving threat, wherein the state of the roving threat is constrained by the context;

determine an engagement zone for the roving threat, wherein the engagement zone is constrained by the context;

5 determine a state of the mobile device, wherein the state includes a current location; and

initiate a first mini-game based on the state of the mobile device.

53. The apparatus of claim 52 wherein the processor is further
10 configured to initiate the first mini-game when the location of the mobile device is within the control area.

54. The apparatus of claim 52 wherein the processor is further
configured to initiate a second mini-game when the location of the mobile device is
15 within the engagement zone.

55. The apparatus of claim 52 wherein the engagement zone comprises a line of sight variable.

20 56. The apparatus of claim 55 wherein the line of sight variable is constrained by the context.

57. An apparatus for providing augmented virtuality (AV) on a mobile device, comprising:

25 means for determining a location of the mobile device;

 means for determining a context based on the location;

 means for obtaining an AV object information;

 means for displaying the AV object information in relation to the context;

30 means for detecting an interaction with the context;

 means for modifying the AV object information based on the interaction;

and

 means for displaying a modified AV object information.

58. The apparatus of claim 57 wherein the context includes weighting information.

5 59. The apparatus of claim 58 wherein the weighting information is based on Received Signal Strength Indication (RSSI) or Round-Trip Time (RTT) data.

60. The apparatus of claim 58 wherein the weighting information is associated with a composition of a physical object in the context.

10

61. The apparatus of claim 57 comprising:
means for receiving a gesture from a user; and
means for modifying the AV object information based on a received
gesture.

15

62. The apparatus of claim 61 comprising:
means for providing an alert signal to the user; and
means for modifying the AV object information based on a reaction time,
wherein the reaction time is a difference between a time the alert signal is provided and
20 a time the gesture is received.

20

63. The apparatus of claim 57 comprising:
means for receiving the AV object information from an augmented
virtuality (AV) server; and
25 means for providing the modified AV object information to the AV
server.

25

64. An apparatus for providing augmented virtuality (AV) on a
mobile device, comprising:
30 means for determining a context based on a location of the mobile
device;
means for generating a state for an AV object;
means for determining a context interaction result;

30

means for modifying the state of the AV object based on the context interaction result;

means for determining a mobile device interaction result;

means for modifying the state of the AV object based on the mobile device interaction result; and

means for displaying the AV object.

65. The apparatus of claim 64 wherein the state of the AV object includes a direction variable.

10

66. The apparatus of claim 65 wherein the means for modifying the state of the AV object based on the context interaction result comprises means for modifying the direction variable.

15 67. The apparatus of claim 65 wherein the means for modifying the state of the AV object based on the mobile device interaction result comprises means for modifying the direction variable.

20 68. The apparatus of claim 64 wherein the state of the AV object includes a size variable.

25 69. The apparatus of claim 68 wherein the means for modifying the state of the AV object based on the context interaction result comprises means for modifying the size variable.

70. The apparatus of claim 64 wherein the means for determining the mobile device interaction result comprises means for receiving a gesture from a user.

30 71. The apparatus of claim 64 comprising means for modifying a state of the mobile device based on the mobile device interaction result.

72. An apparatus for using virtual tools in Augmented Virtuality (AV) applications, comprising:

means for determining a context based on a location of a mobile device;
means for generating a state for a virtual threat;
means for generating a state for a virtual tool;
means for determining a context interaction result;
5 means for determining a tool interaction result;
means for modifying the state of the virtual threat based on the context
interaction result and the tool interaction result; and
means for displaying the virtual threat and the virtual tool.

10 73. The apparatus of claim 72 wherein the state of the virtual tool
includes a location variable.

74. The apparatus of claim 73 comprising means for modifying the
state of the virtual tool based a state of the mobile device.

15 75. The apparatus of claim 74 wherein determining the tool
interaction result is based on a modified state of the virtual tool.

76. The apparatus of claim 72 wherein the tool interaction result
20 constrains the state of the virtual threat.

77. An apparatus for determining interaction variables in an
Augmented Reality (AR) application, comprising:
means for obtaining a context;
25 means for determining location data corresponding to a plurality of areas
within the context;
means for determining weighting information for each of the plurality of
areas;
means for determining interaction variables based on the weighting
30 information; and
means for storing the interaction variables.

78. The apparatus of claim 77 wherein the weighting information is based on RSSI information.

79. The apparatus of claim 77 wherein the weighting information is
5 based on RTT information.

80. An apparatus for providing augmented virtuality (AV) on a mobile device, comprising:
means for determining a context based on a location of the mobile
10 device;
means for establishing a control area within the context;
means for generating a state for a roving threat, wherein the state of the roving threat is constrained by the context;
means for determining an engagement zone for the roving threat,
15 wherein the engagement zone is constrained by the context;
means for determining a state of the mobile device, wherein the state includes a current location; and
means for initiating a first mini-game based on the state of the mobile device.

20

81. The apparatus of claim 80 wherein the first mini-game is initiated when the location of the mobile device is within the control area.

82. The apparatus of claim 80 comprising means for initiating a
25 second mini-game when the location of the mobile device is within the engagement zone.

83. The apparatus of claim 80 wherein the engagement zone comprises a line of sight variable.
30

84. The apparatus of claim 83 wherein the line of sight variable is constrained by the context.

85. A computer program product residing on a processor-executable storage medium, the computer program product comprising processor-executable instructions configured to cause a processor to:

- 5 determine a location of a mobile device;
- determine a context based on the location;
- obtain an AV object information;
- display the AV object information in relation to the context;
- detect an interaction with the context;
- modify the AV object information based on the interaction; and
- 10 display a modified AV object information.

86. The computer program product of claim 85 wherein the context includes weighting information.

- 15 87. The computer program product of claim 86 wherein the weighting information is based on Received Signal Strength Indication (RSSI) or Round-Trip Time (RTT) data.

- 20 88. The computer program product of claim 86 wherein the weighting information is associated with a composition of a physical object in the context.

- 89. The computer program product of claim 85 further comprising instructions configured to cause the processor to:
 - 25 receive a user gesture; and
 - modify the AV object information based on a received gesture.

- 90. The computer program product of claim 89 further comprising instructions configured to cause the processor to:
 - 30 provide an alert signal to a user; and
 - modify the AV object information based on a reaction time, wherein the reaction time is a difference between a time the alert signal is provided and a time the user gesture is received.

91. The computer program product of claim 85 further comprising instructions configured to cause the processor to:

5 receive the AV object information from an augmented virtuality (AV) server; and
provide the modified AV object information to the AV server.

92. A computer program product residing on a processor-executable storage medium, the computer program product comprising processor-executable
10 instructions configured to cause a processor to:

determine a context based on a location of a mobile device;
generate a state for an AV object;
determine a context interaction result;
modify the state of the AV object based on the context interaction result;
15 determine a mobile device interaction result;
modify the state of the AV object based on the mobile device interaction result; and
display the AV object.

20 93. The computer program product of claim 92 wherein the state of the AV object includes a direction variable.

94. The computer program product of claim 93 further comprising instructions configured to cause the processor to modify the direction variable based on
25 the context interaction result.

95. The computer program product of claim 93 further comprising instructions configured to cause the processor to modify the direction variable based on the mobile device interaction result.

30

96. The computer program product of claim 92 wherein the state of the AV object includes a size variable.

97. The computer program product of claim 96 further comprising instructions configured to cause the processor to modify the size variable based on the context interaction result.

5 98. The computer program product of claim 92 further comprising instructions configured to cause the processor to receive a gesture from a user.

99. The computer program product of claim 92 further comprising instructions configured to cause the processor to modify a state of the mobile device
10 based on the mobile device interaction result.

100. A computer program product residing on a processor-executable storage medium, the computer program product comprising processor-executable instructions configured to cause a processor to:
15 determine a context based on a location of a mobile device;
generate a state for a virtual threat;
generate a state for a virtual tool;
determine a context interaction result;
determine a tool interaction result;
20 modify the state of the virtual threat based on the context interaction result and the tool interaction result; and
display the virtual threat and the virtual tool.

101. The computer program product of claim 100 wherein the state of
25 the virtual tool includes a location variable.

102. The computer program product of claim 101 further comprising instructions configured to cause the processor to modify the state of the virtual tool based a state of the mobile device.

30 103. The computer program product of claim 102 further comprising instructions configured to cause the processor to determine the tool interaction result based on a modified state of the virtual tool.

104. The computer program product of claim 100 wherein the tool interaction result constrains the state of the virtual threat.

5 105. A computer program product residing on a processor-executable storage medium, the computer program product comprising processor-executable instructions configured to cause a processor to:

 obtain a context;
 determine location data corresponding to a plurality of areas within the
10 context;
 determine weighting information for each of the plurality of areas;
 determine interaction variables based on the weighting information; and
 store the interaction variables.

15 106. The computer program product of claim 105 wherein the weighting information is based on RSSI information.

 107. The computer program product of claim 105 wherein the weighting information is based on RTT information.

20 108. A computer program product residing on a processor-executable storage medium, the computer program product comprising processor-executable instructions configured to cause a processor to:

 determine a context based on a location of a mobile device;
25 establish a control area within the context;
 generate a state for a roving threat, wherein the state of the roving threat is constrained by the context;
 determine an engagement zone for the roving threat, wherein the engagement zone is constrained by the context;
30 determine a state of the mobile device, wherein the state includes a current location; and
 initiate a first mini-game based on the state of the mobile device.

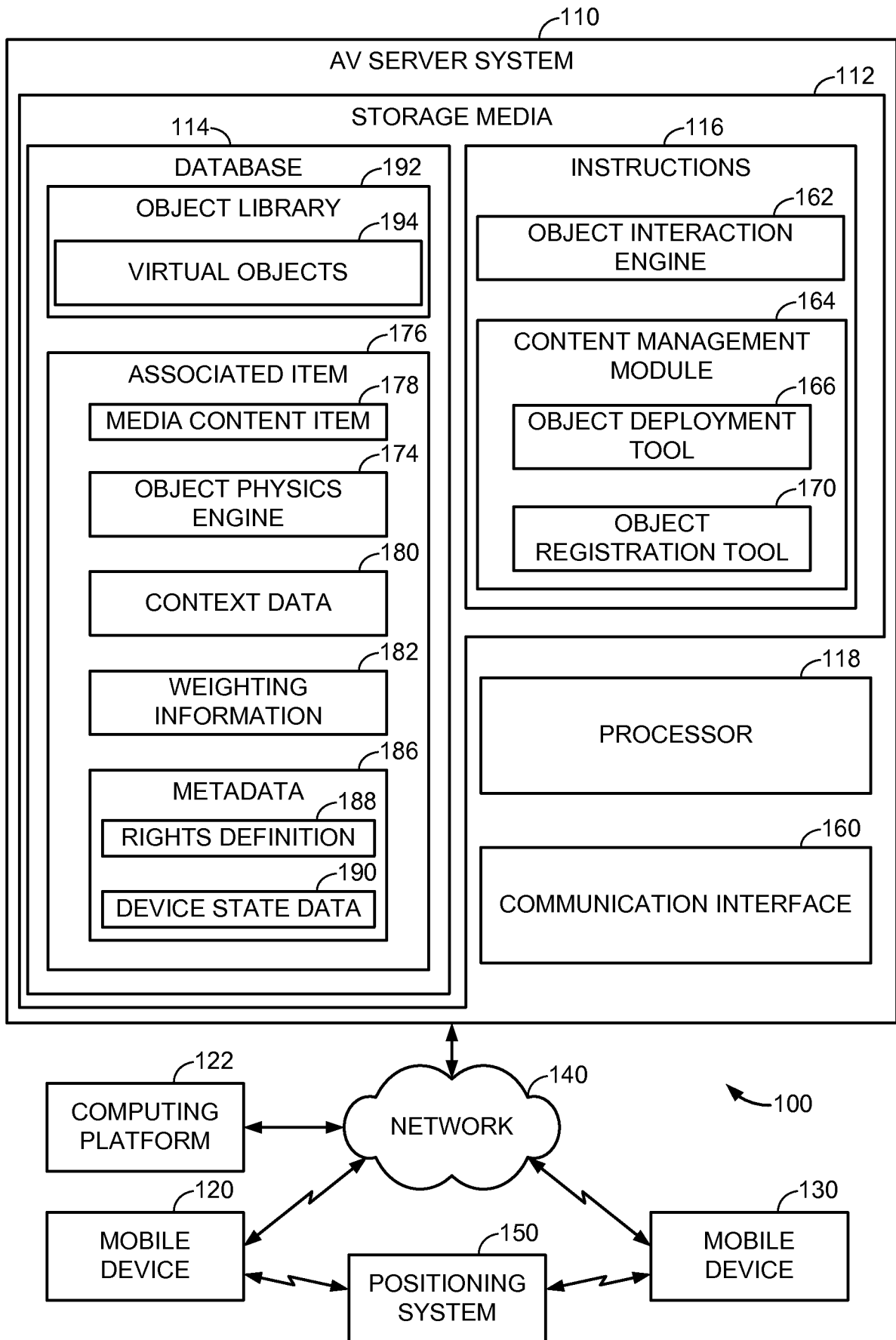
109. The computer program product of claim 108 further comprising instructions configured to cause the processor to initiate the first mini-game when the location of the mobile device is within the control area.

5 110. The computer program product of claim 108 further comprising instructions configured to cause the processor to initiate a second mini-game when the location of the mobile device is within the engagement zone.

10 111. The computer program product of claim 110 wherein the engagement zone comprises a line of sight variable.

112. The computer program product of claim 111 wherein the line of sight variable is constrained by the context.

1/18

**FIG. 1**

2/18

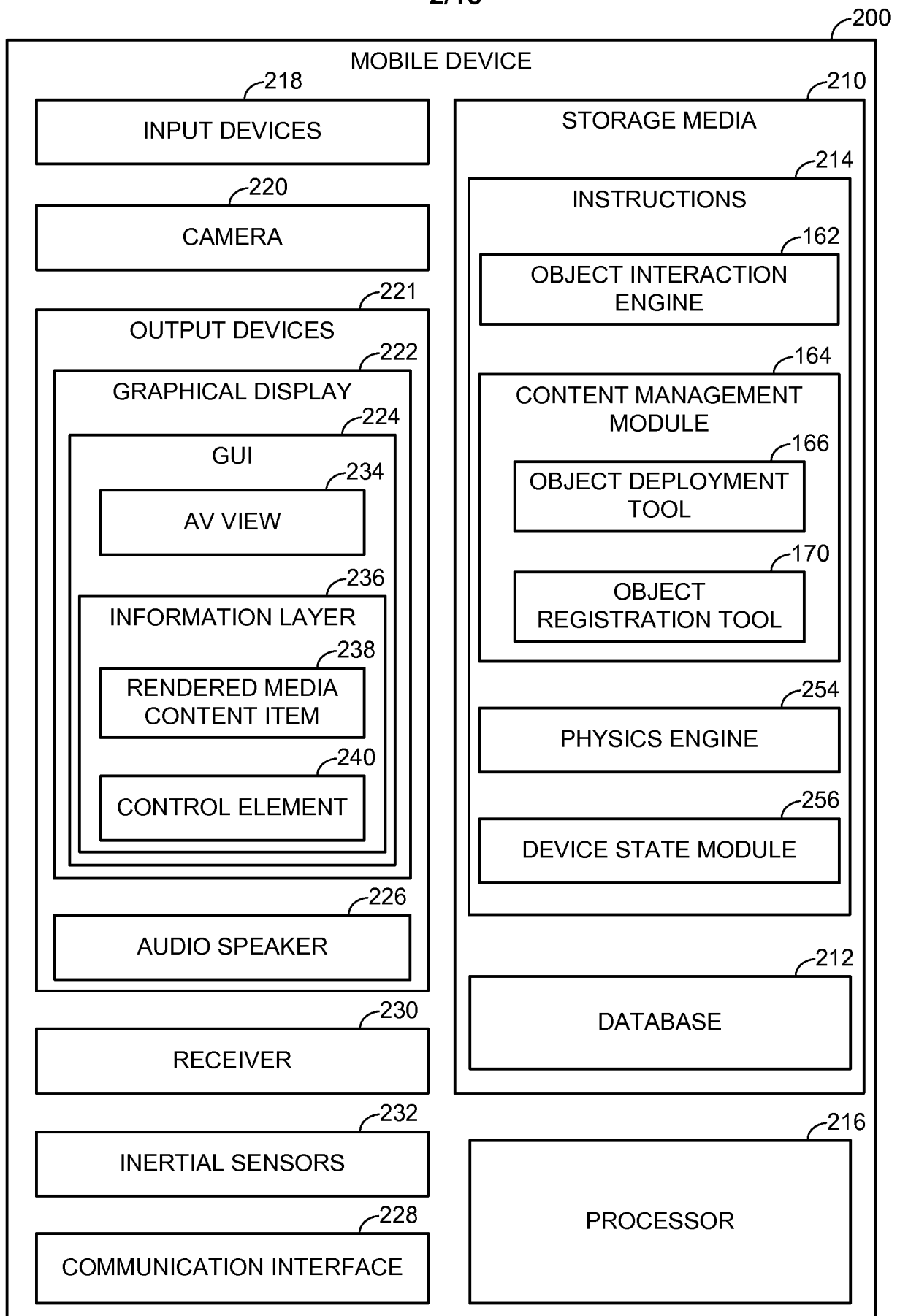


FIG. 2

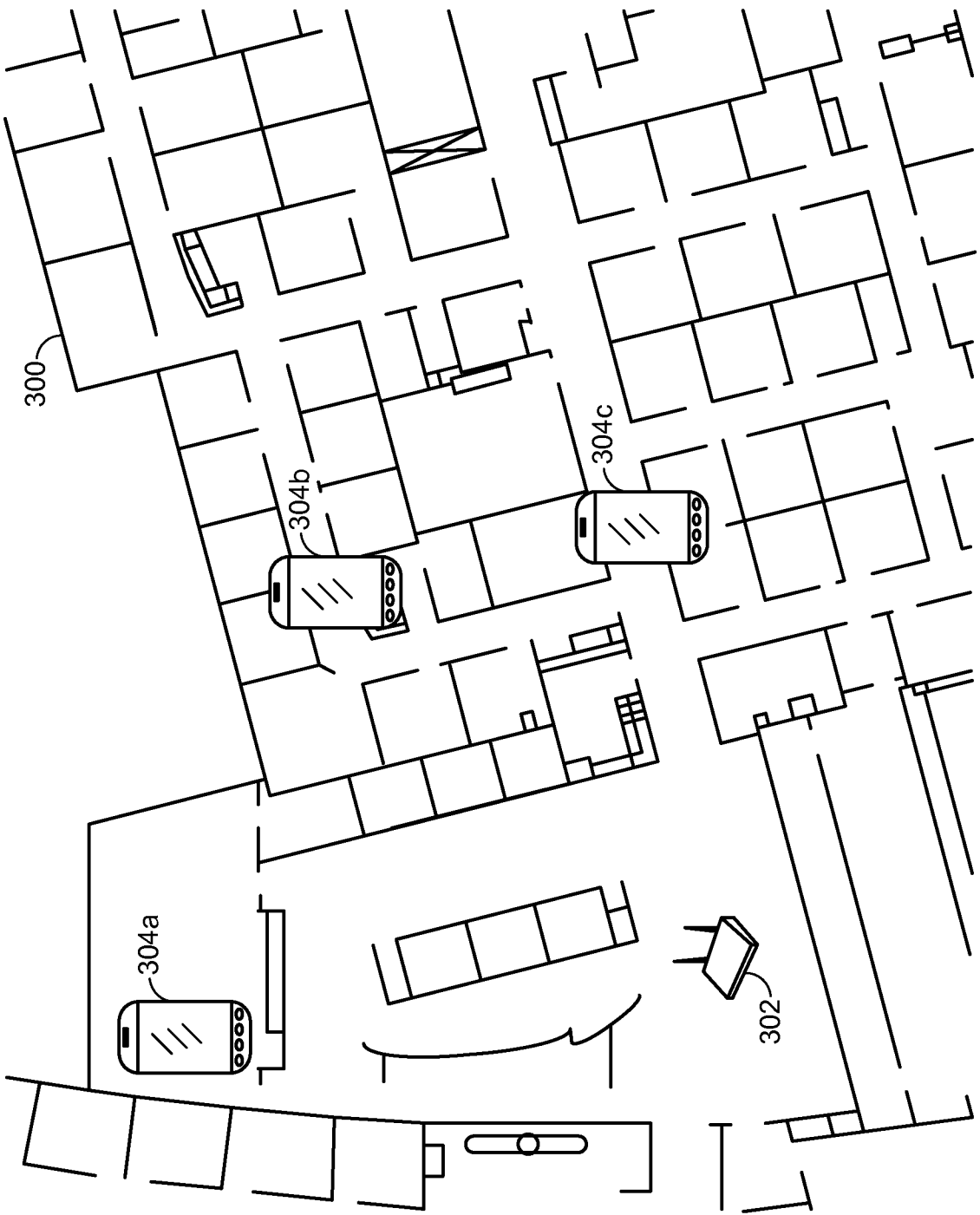


FIG. 3

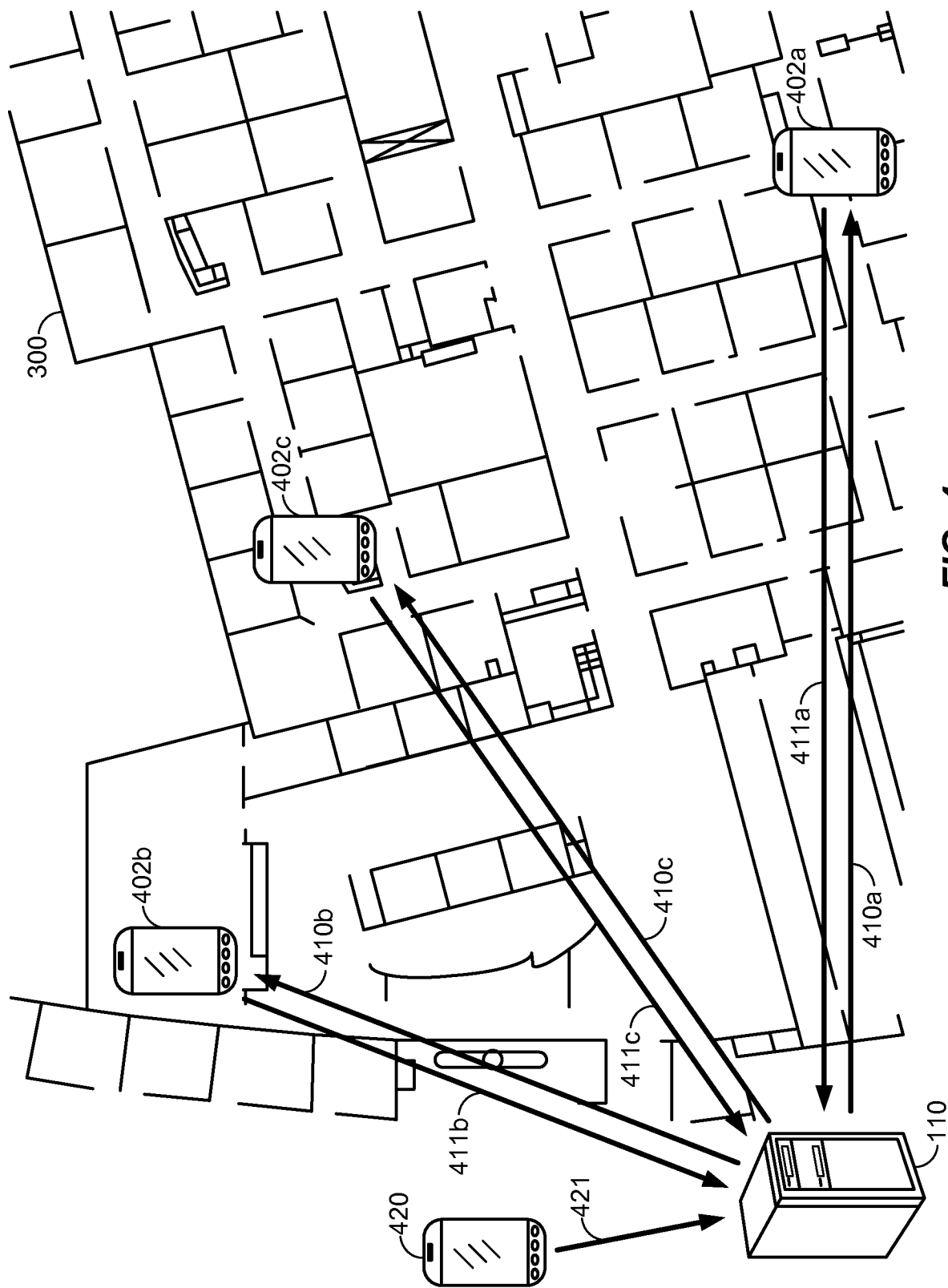


FIG. 4

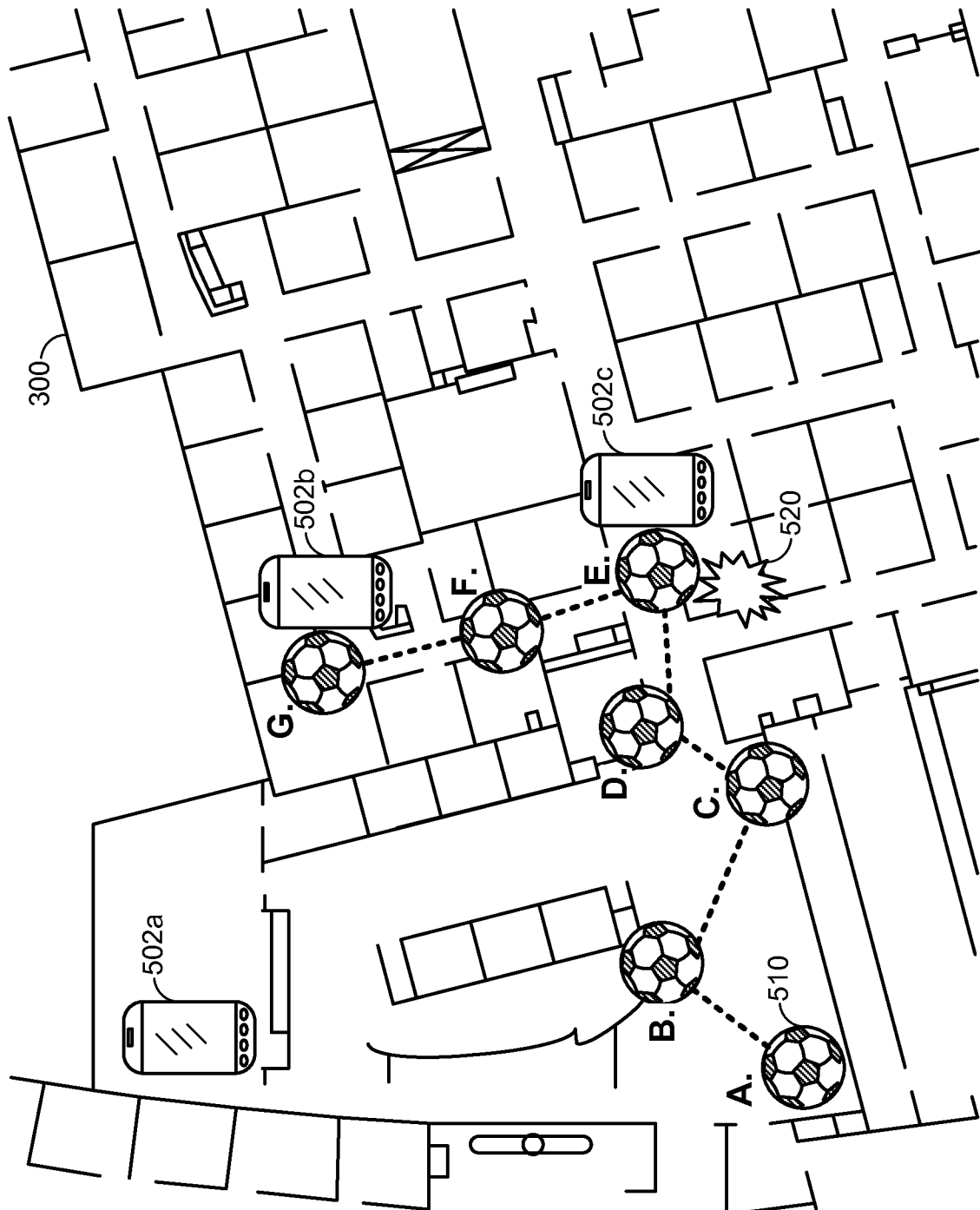


FIG. 5

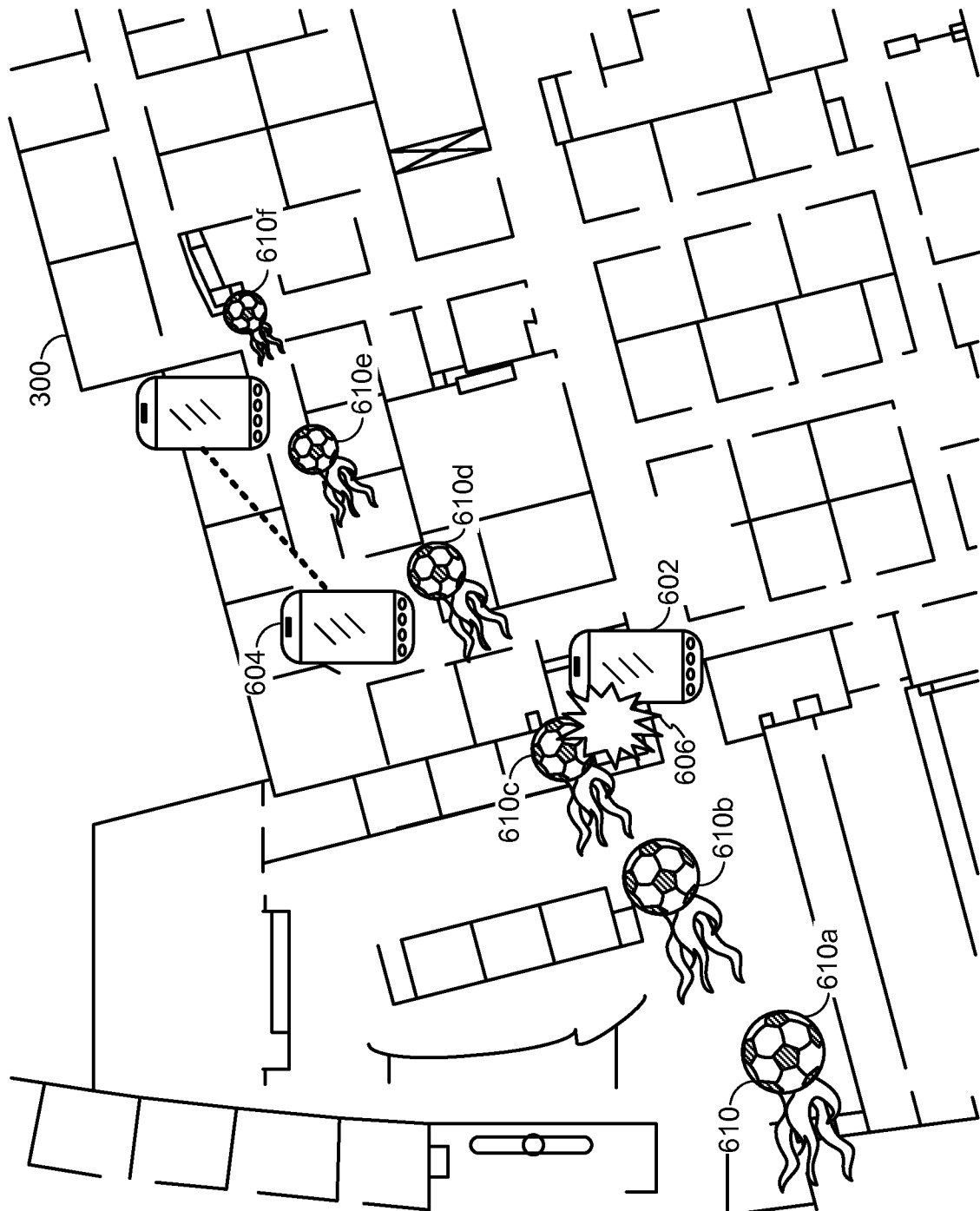


FIG. 6

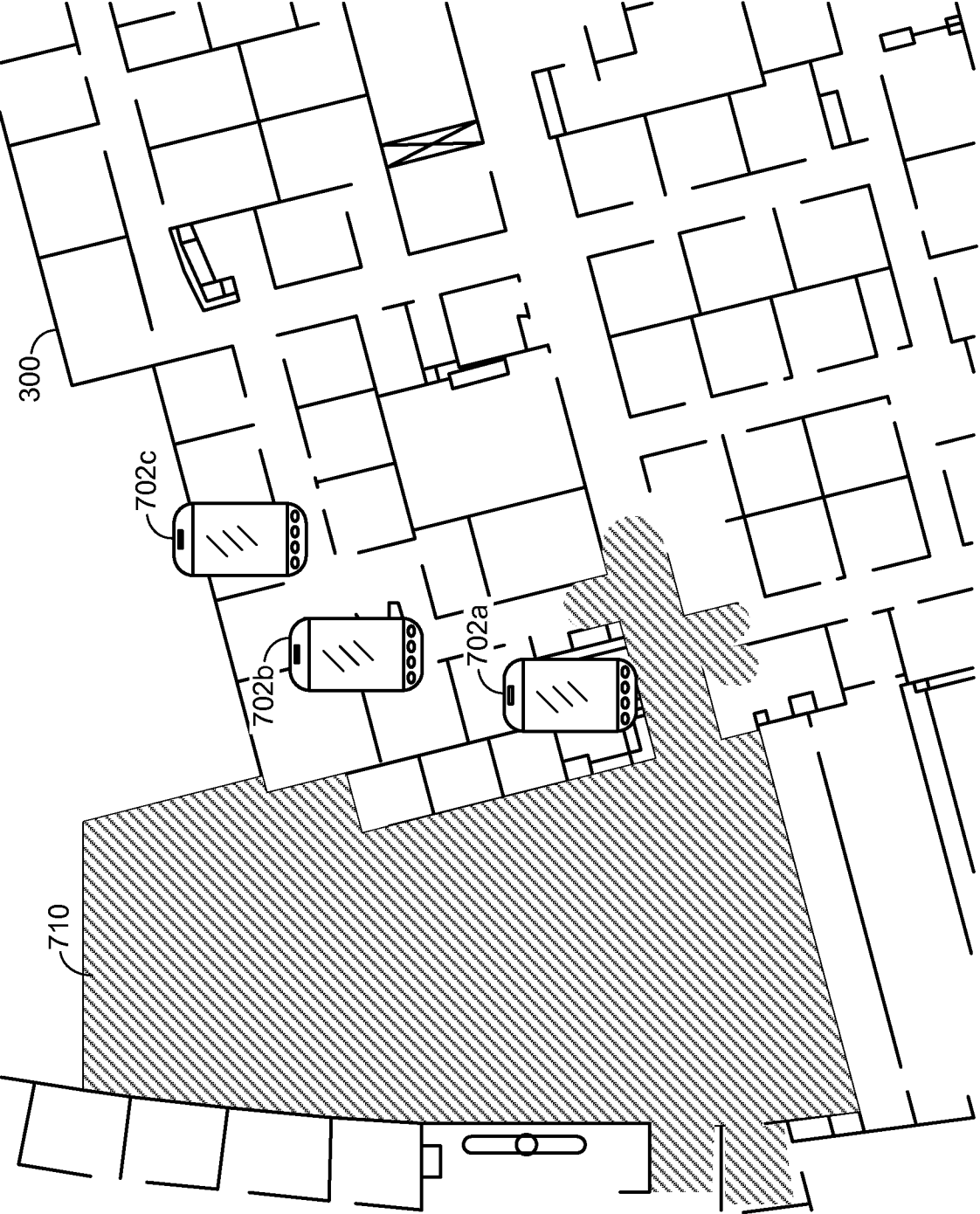


FIG. 7

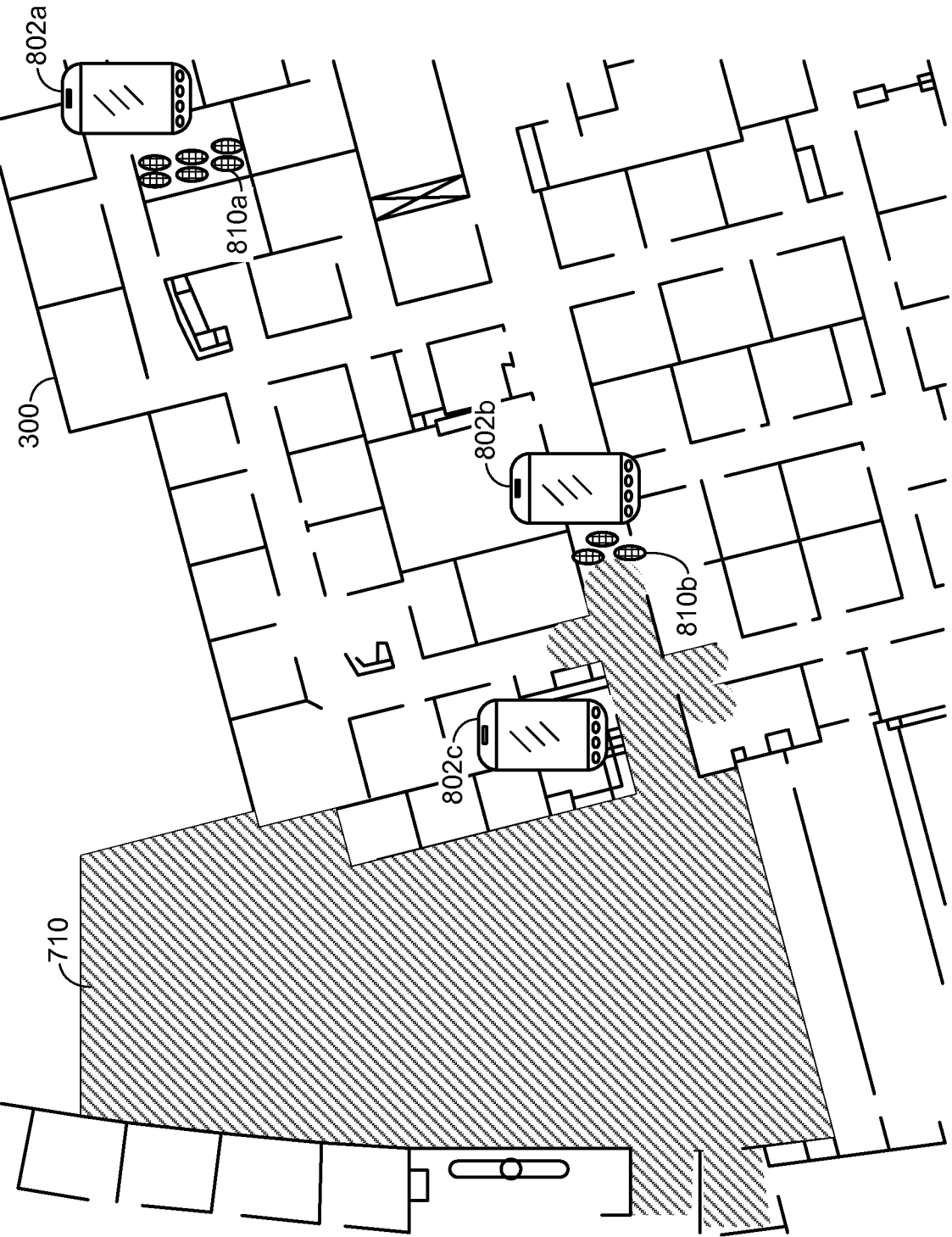


FIG. 8

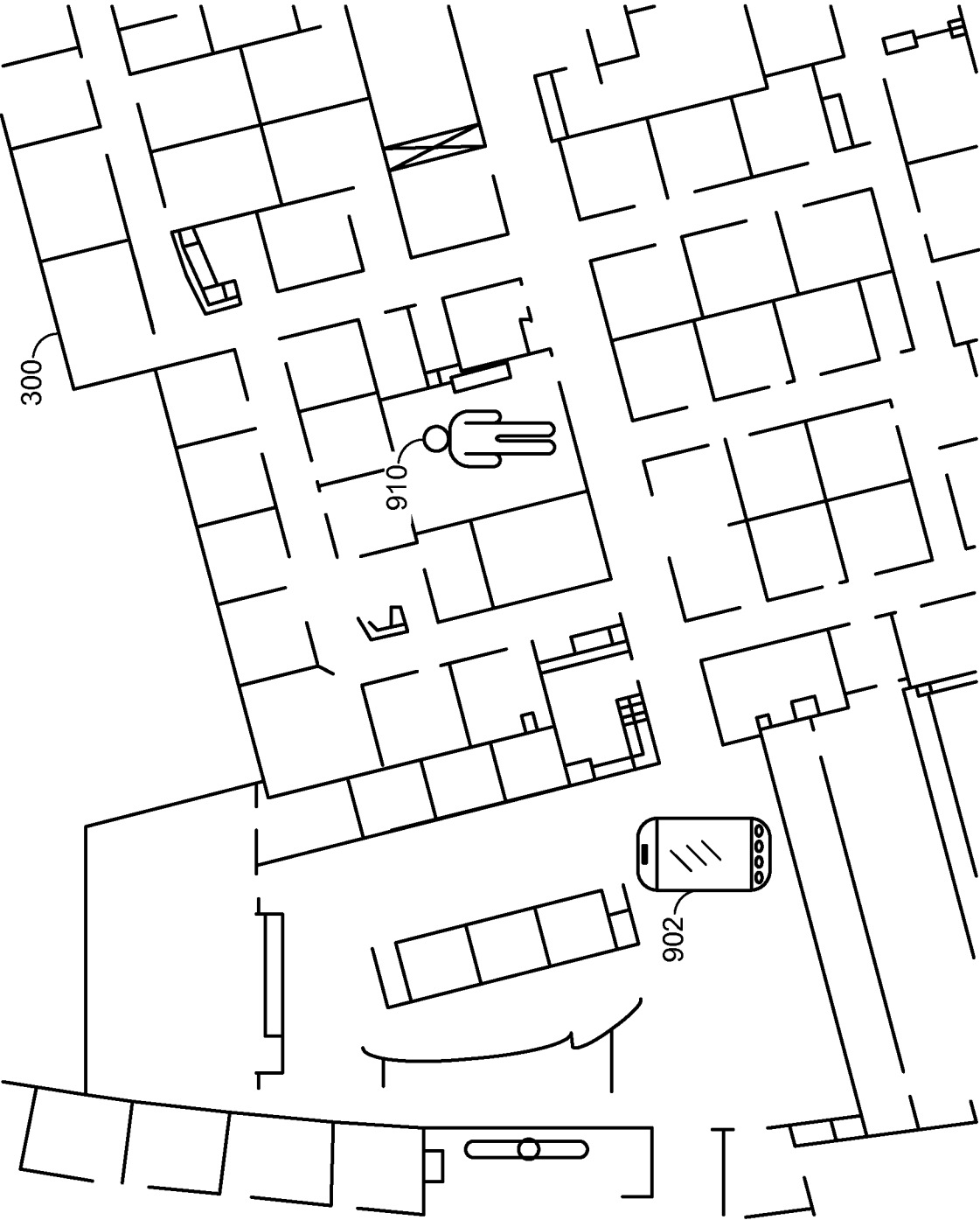


FIG. 9

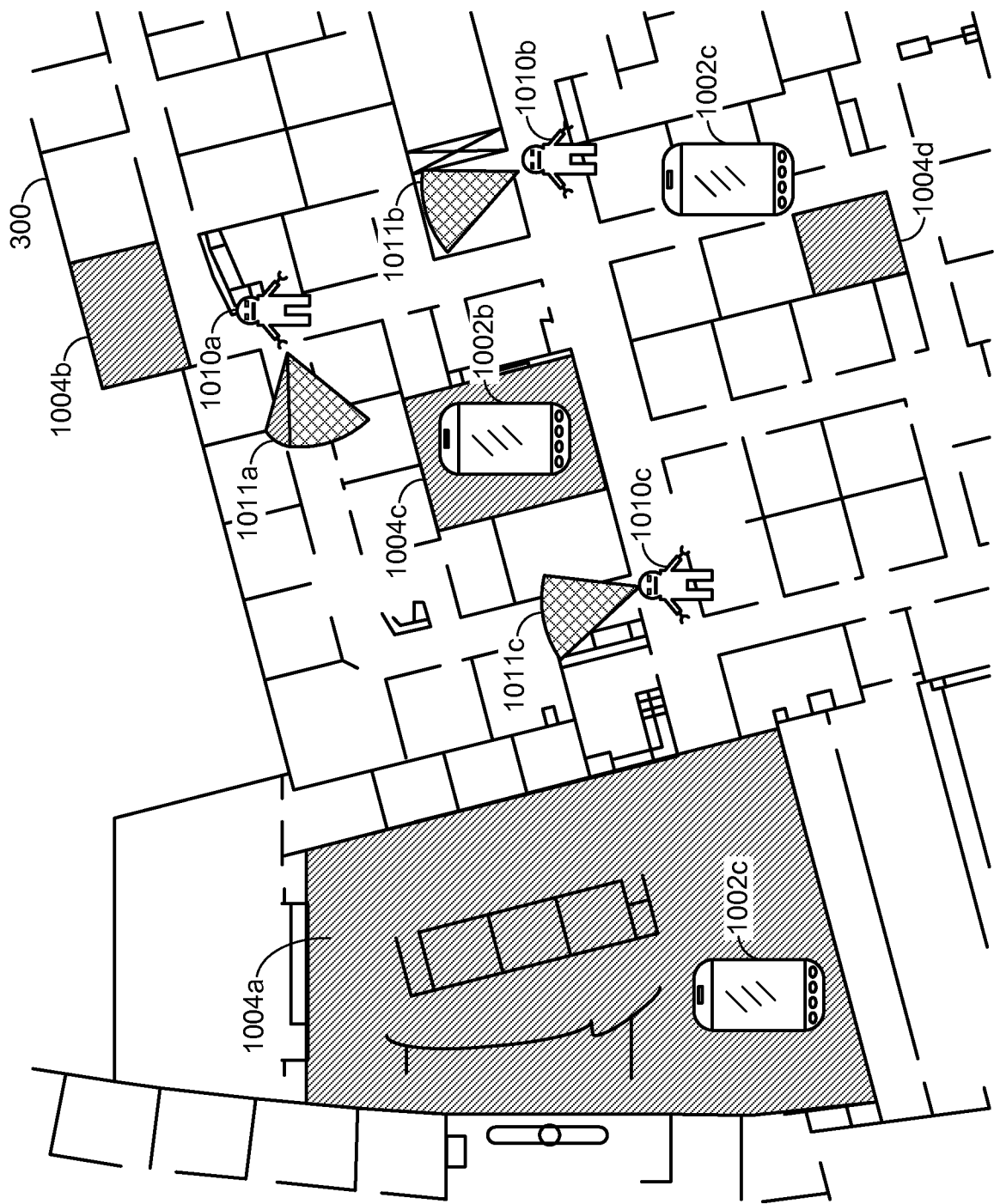


FIG. 10

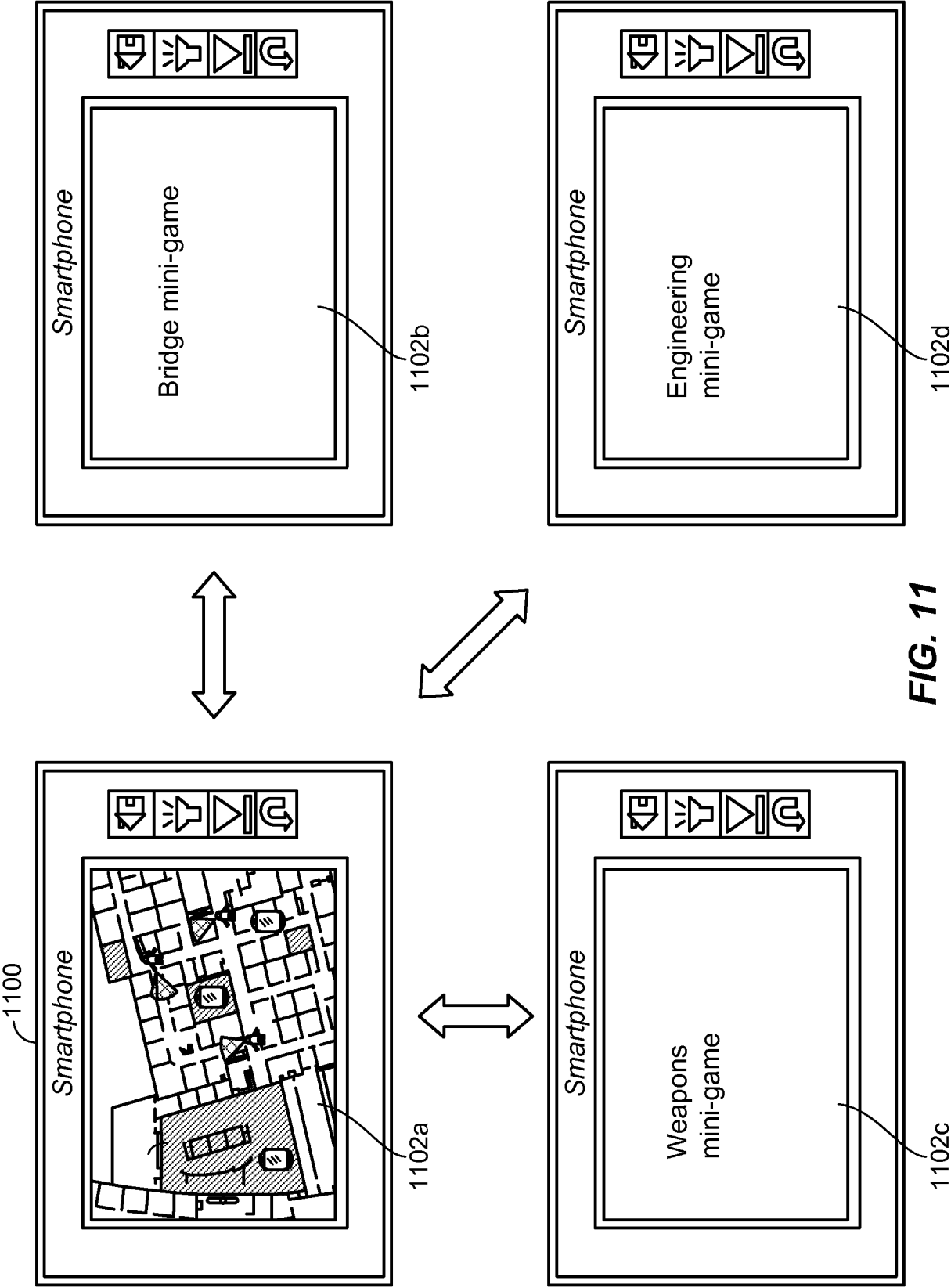
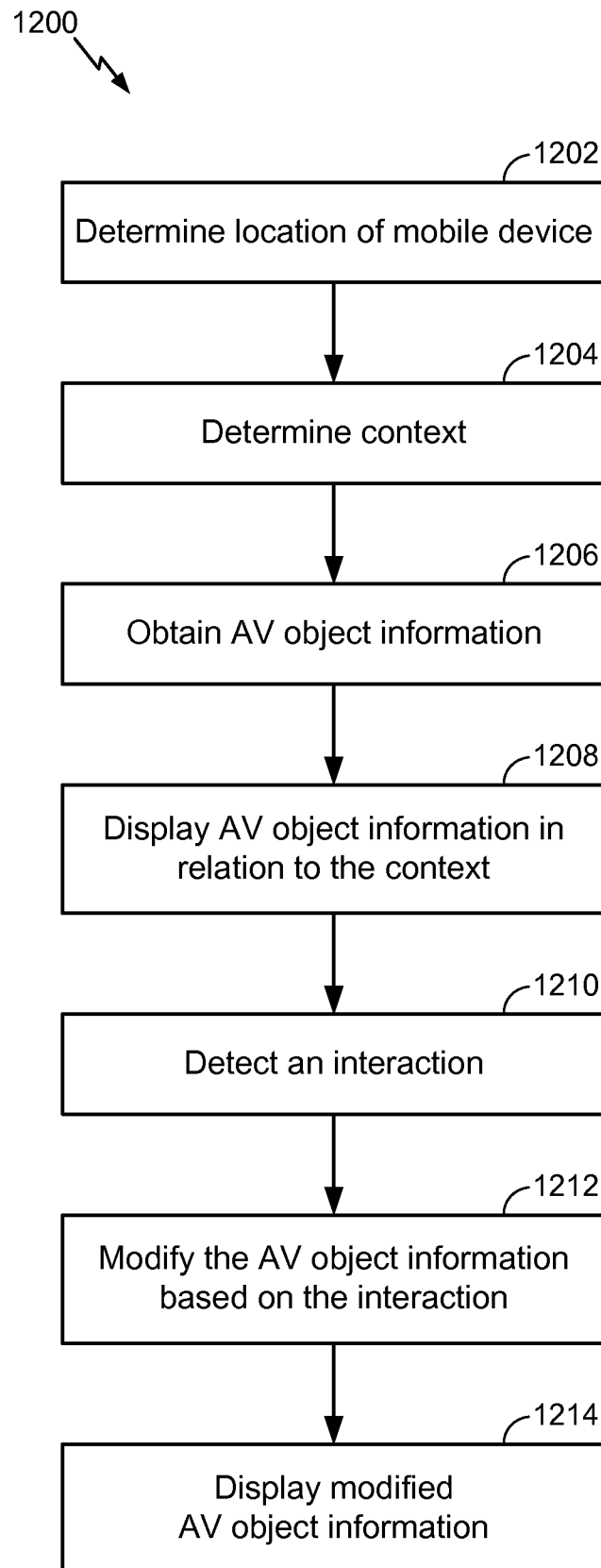
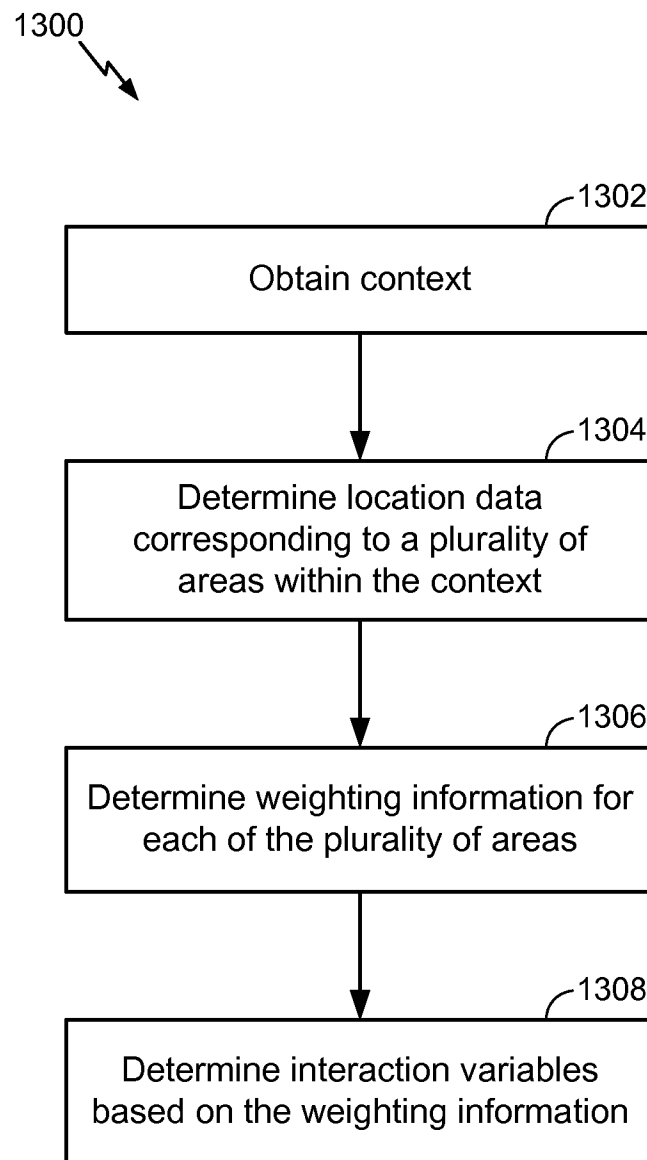


FIG. 11

12/18

**FIG. 12**

13/18

**FIG. 13A**

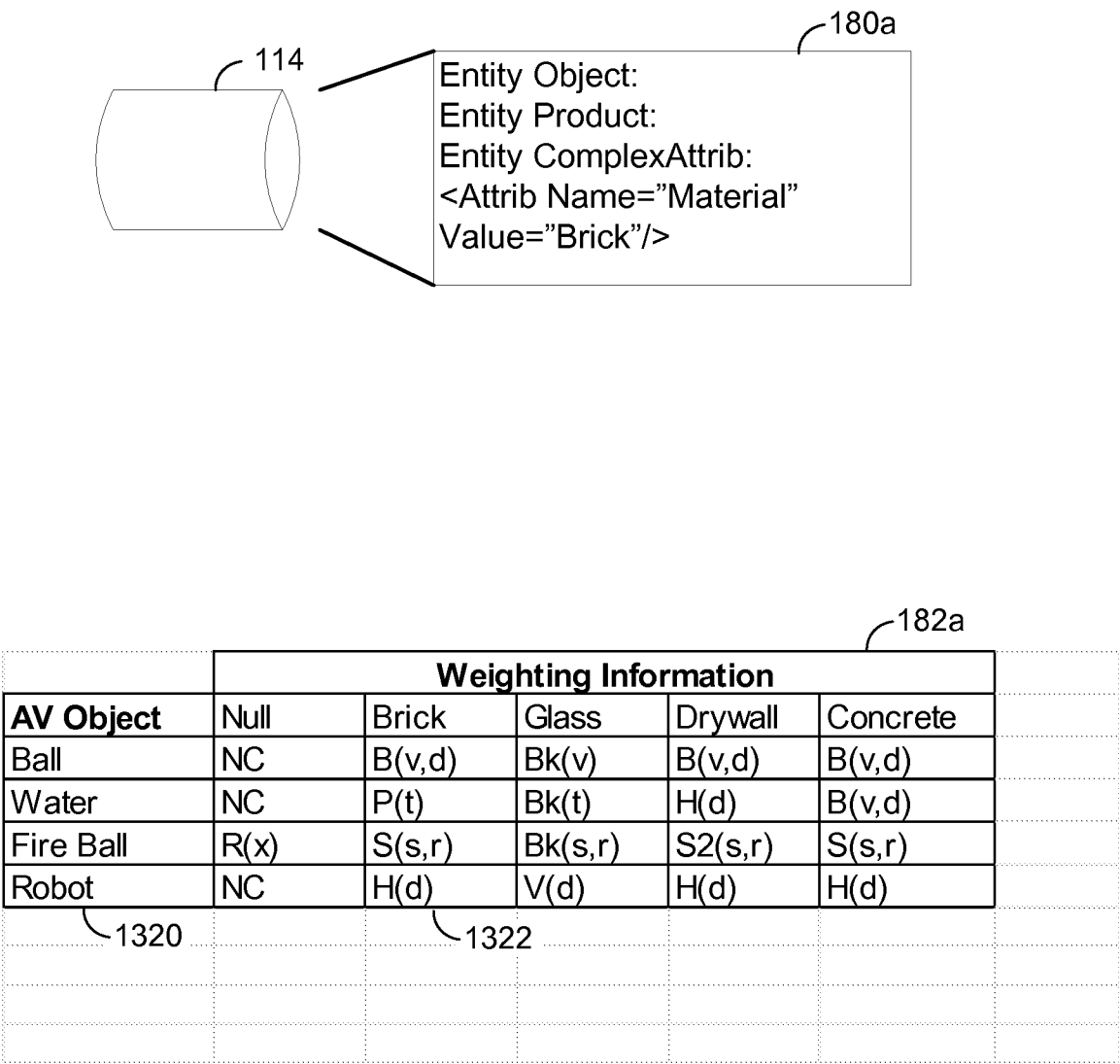
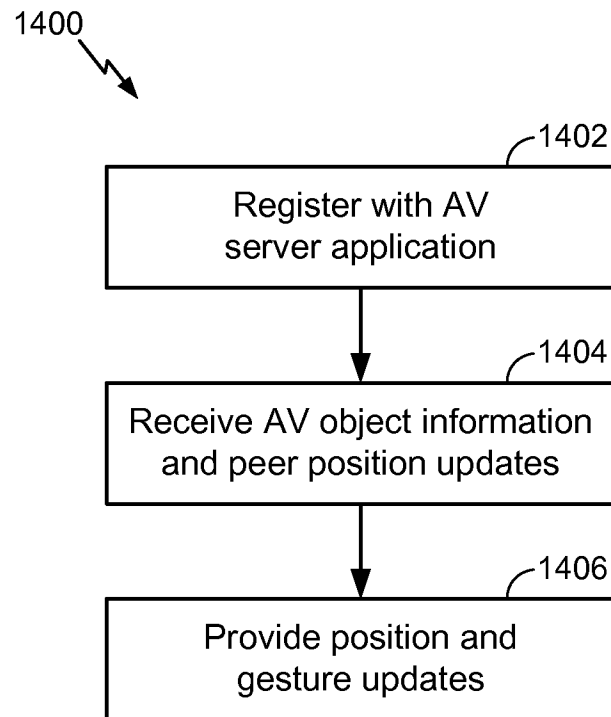
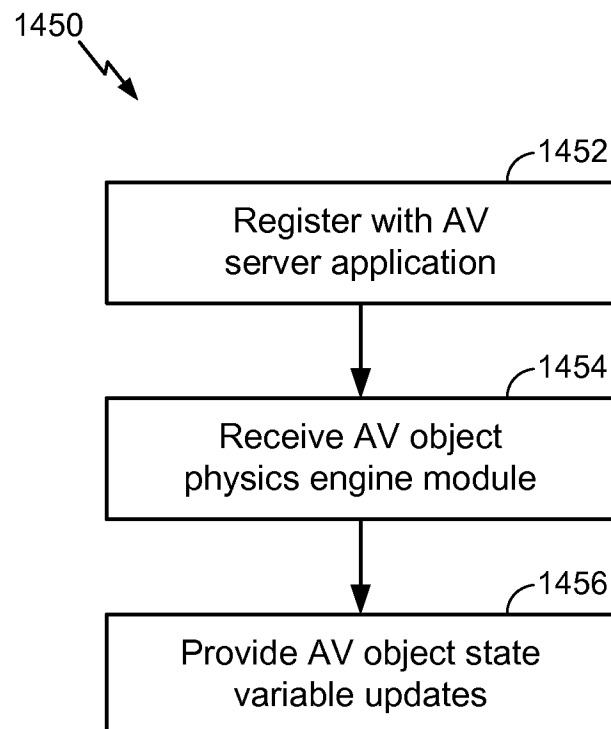
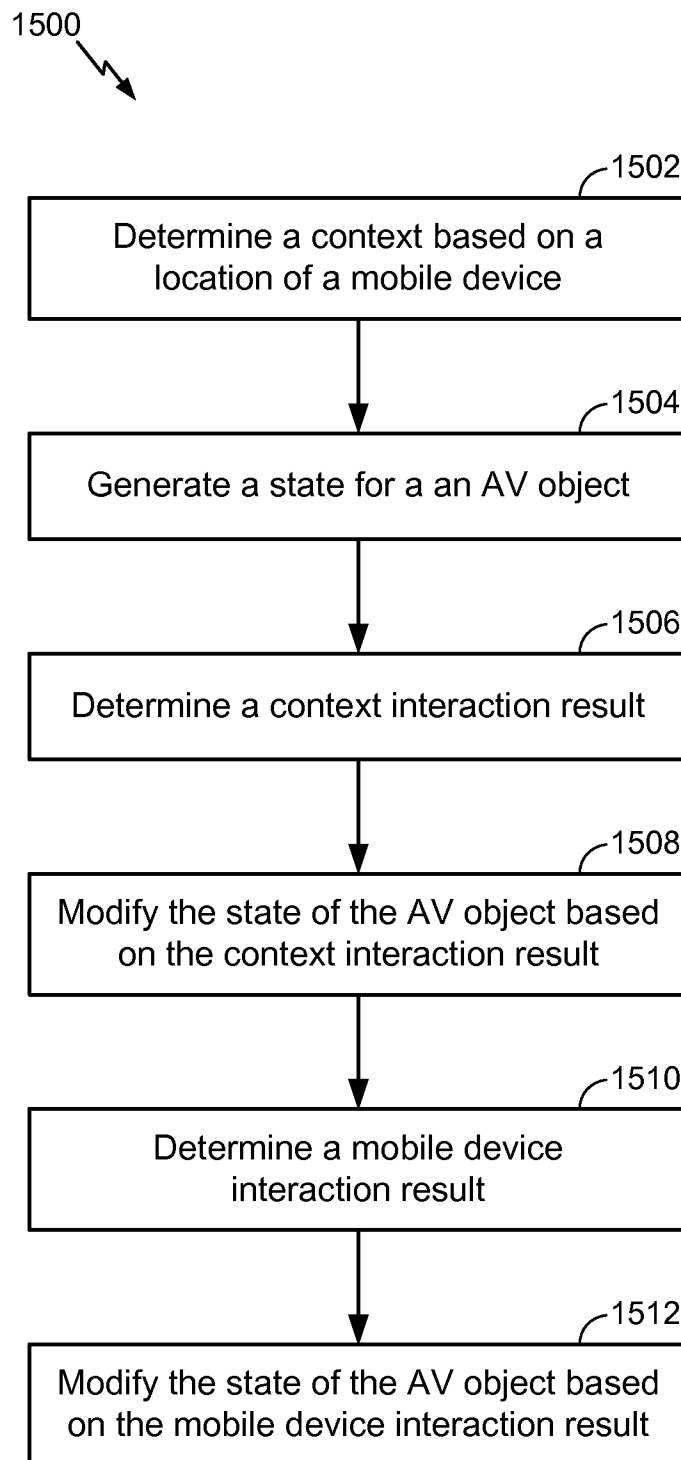


FIG. 13B

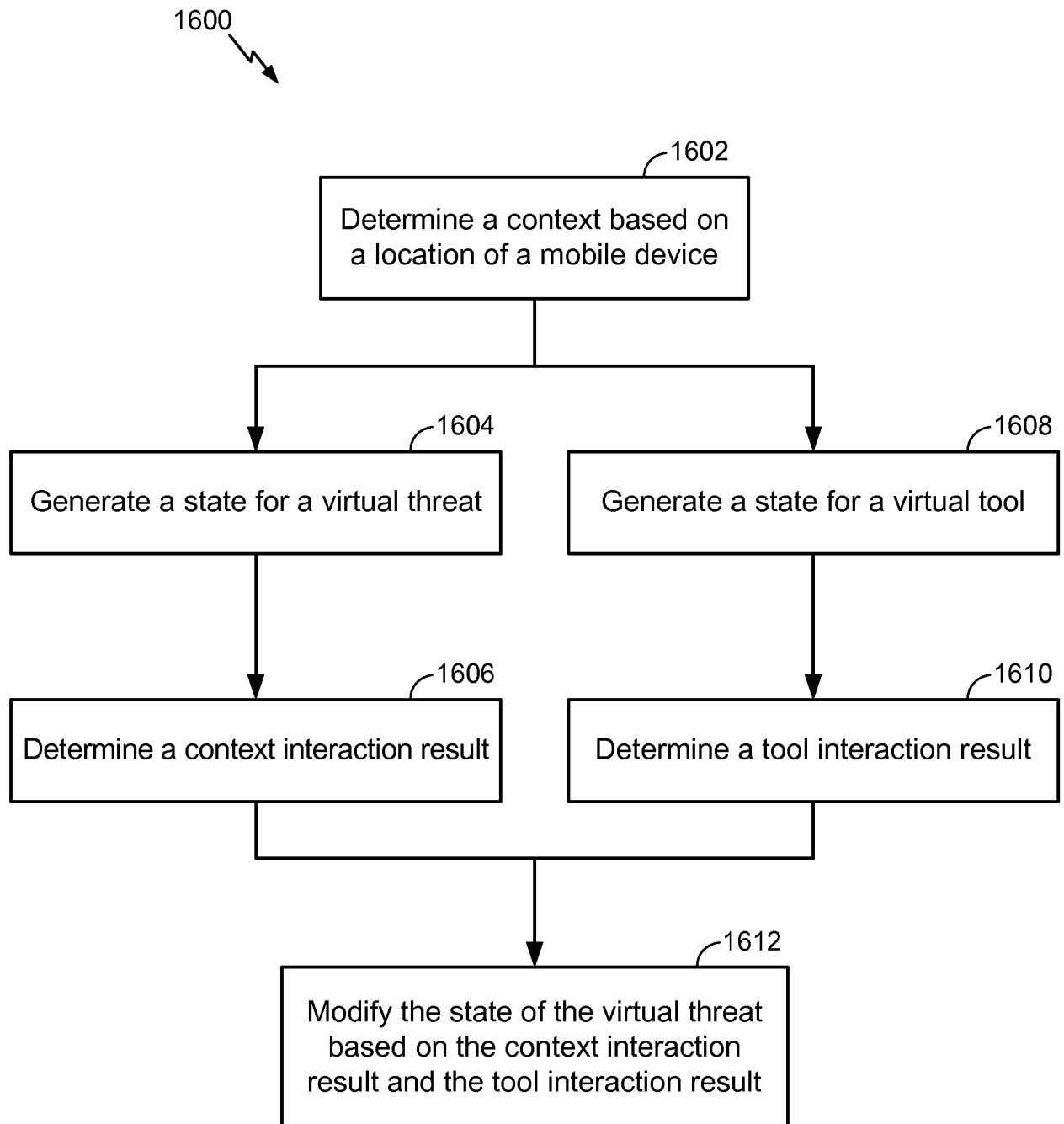
15/18

**FIG. 14A****FIG. 14B**

16/18

**FIG. 15**

17/18

**FIG. 16**

18/18

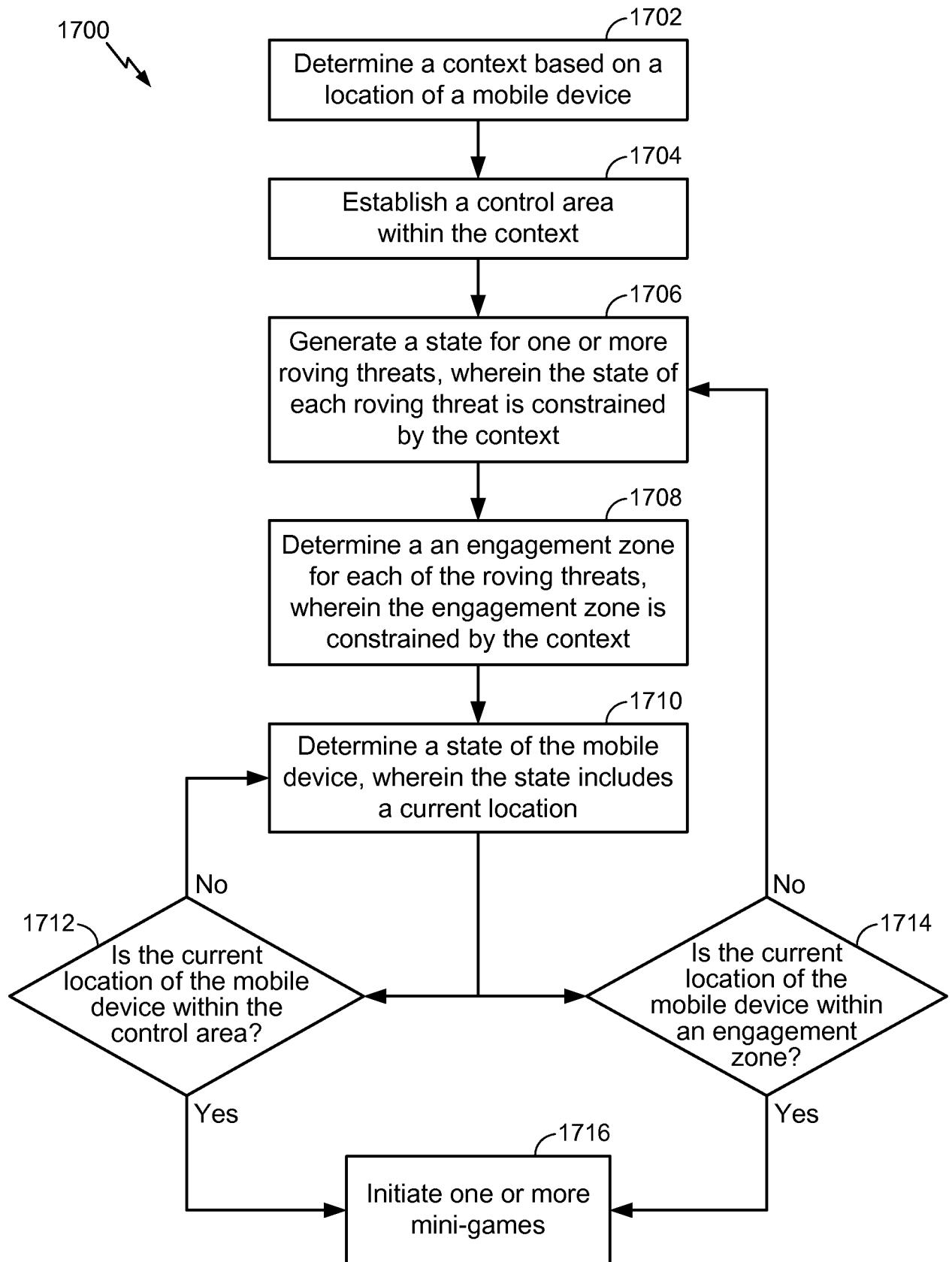


FIG. 17

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2014/057095

A. CLASSIFICATION OF SUBJECT MATTER

INV. A63F13/20 A63F13/40 A63F13/216 A63F13/217 A63F13/213

G06T19/00

ADD. A63F13/332 A63F13/211 A63F13/45

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

A63F G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, PAJ, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2012/195461 A1 (LAWRENCE ASHOK INIGO ROY [US]) 2 August 2012 (2012-08-02) abstract paragraph [0002] figures 4-6 figures 7,8 paragraph [0013] - paragraph [0015] paragraph [0019] - paragraph [0020] ----- -/-	8-15, 36-43, 64-71, 92-99



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

14 January 2015

Date of mailing of the international search report

21/01/2015

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Garton, Paul

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2014/057095

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2012/315992 A1 (GERSON ELAD [US] ET AL) 13 December 2012 (2012-12-13) abstract paragraph [0026] paragraph [0045] paragraph [0060] paragraph [0073] -----	8-15, 36-43, 64-71, 92-99
A	WO 2011/109167 A1 (EMPIRE TECHNOLOGY DEV LLC [US]; LEUNG HENRY [CA]) 9 September 2011 (2011-09-09) abstract paragraph [0002] paragraph [0019] paragraph [0047] -----	8-15, 36-43, 64-71, 92-99

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2014/057095

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012195461 A1	02-08-2012	US 2012195461 A1	02-08-2012
		WO 2012106370 A2	09-08-2012

US 2012315992 A1	13-12-2012	TW 201250613 A	16-12-2012
		US 2012315992 A1	13-12-2012
		WO 2012170315 A2	13-12-2012

WO 2011109167 A1	09-09-2011	CN 102668605 A	12-09-2012
		JP 5566472 B2	06-08-2014
		JP 2013515244 A	02-05-2013
		KR 20120094007 A	23-08-2012
		KR 20140041925 A	04-04-2014
		US 2011217962 A1	08-09-2011
		WO 2011109167 A1	09-09-2011
