A wearable device can be worn by a user, and can include a mode selector, one or more sensors, a signal detector to detect wireless signals, a feedback mechanism comprising at least one of a haptic system, an audio system, or a visual system and providing feedback outputs. The wearable device can also include a controller that can initiate a selected mode from a plurality of modes on the wearable device, where the selected mode causing the controller to execute a real-world game. In accordance with the selected mode, the controller can interpret sensor data from the one or more sensors and inputs from the signal detector, where the sensor data corresponds to a series of actions performed by the user utilizing the wearable device. Based the interpretations, the controller can generate feedback responses via a feedback mechanism.
FIG. 1B
Receive Mode Selection Input 200

Interpret Final SM Reports From State Machines Based On Mode Selection 205

Identify Sub-Mode Trigger In Final Reports 210

Reconfigure Report Interpretation Based On Sub-Mode Trigger 215

Generate Commands Based On Detected Final States In Accordance With Mode/Sub-mode Control Commands 220

Audio 222  Haptic 224  Visual 226  Control Commands 228

Transmit/Execute Commands 225

FIG. 2
Detect Signal From A Network Device

Perform Lookup To Determine Whether UID Is Associated With A Virtual Resource

UID Associated?

Generate Feedback Output indicating Proximate VR

Receive User input To Search For W.R.

Determine Direction Of Signal

Monitor Sensor Data to Determine Whether User Points Wearable Device Towards V.R. Signal

Adjust Feedback Depending On Point Direction

Allocate V.R. When Wearable Device Points in Correct Direction For Predetermined Period

FIG. 6
Detect Beacons From Network Devices 700

Record UIDs Of Network Devices 705

Perform Lookup In Association Table 710

UID Associated? 715

Yes 717

Allocate Virtual Resources in VR Log 720

Transmit Unassociated UIDs To Host Server 730

Receive Associated VR For UIDs 745

Flush UID Log 747

No 719

Allocate Virtual Resources in VR Log 720

Generate Request For UID Association 725

Select Virtual Resource Locally 740

Randomly 741

Sequen. 743

Flush VR Log 777

Initiate Gameplay Application 750

Render Gaming Environment On Display 755

Receive User Interactions 760

Expend, Trade, Accumulate Stored VRs Based On User Interactions 765

Receive User Input To Deactivate Application 770

Compile Accumulated VRs And Transmit To Host Server To Save Progress 775

Flush VR Log 777

FIG. 7A

FIG. 7B
Detect User Input Placing Wearable Device In Training Mode 1200

Via Mode Selector 1202 Via Connected MCD 1204

Detect Inductive Pairing With Self-Propelled Device 1205

Initiate Training Mode On Self-Propelled Device 1210

Synchronize Directionally With Self-Propelled Device 1215

Synchronize Gyros (Manual) 1217 Synchronize Gyros (Auto) 1219

Generate And Transmit Spin Command To Self-Propelled Device 1220

Detect Asymmetry in Rad Pattern 1225

Transmit Directional Calibration Command To Self-Propelled Device 1230

Track Self-Propelled Device Location 1240

Generate Output Instructing User To Perform Set Of Actions 1245

Based On Data From SPD 1247 Using Local Routine Set 1249

Initiate Timer 1250

Monitor Sensor Data To Determine Whether User Performs Set Of Actions 1255

Performed W/in Time Period? 1260

Yes 1261

No 1263

Terminate Training Session And Generate Score 1265

FIG. 12
Detect User Input Placing Wearable Device In Sword Mode 1300

Monitor Sensor Data To Determine Whether The User Has Grabbed/Grasped An Object 1305

Determine A Series of Actions Performed By The User With The Object From The Sensor Data 1310

In Response To Each Action, Generate Feedback Response 1315
- Haptic Response 1316
- Audio Response 1317
- Visual Response 1318

Receive User Input To Exit Sword Mode 1320

Deactivate Sword Mode 1325

**FIG. 13**
FIG. 14
FIG. 15
FIG. 16
FIG. 17

First Modular Sensing Device 1810
IMU 1812
  ACCL 1815
  GYRO 1817

PROC 1814
MEM 1816

Input 1811

Second Modular Sensing Device 1820
IMU 1822
  ACCL 1825
  GYRO 1827

PROC 1824
MEM 1826

Input 1813

Controller 1830
Processor 1832
Instructions 1845
Memory 1834
  --Application 1839
  --Inference Engine 1835

FIG. 18
TASK-ORIENTED FEEDBACK USING A MODULAR SENSING DEVICE

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of priority to U.S. Provisional Patent Application Ser. No. 62/274,514, entitled “PORTABLE SENSING DEVICE FOR PROCESSING GESTURES AS INPUT,” and filed on Jan. 3, 2016; and to U.S. Provisional Application Ser. No. 62/346,216, entitled “MODULAR SENSING DEVICE FOR PROCESSING GESTURES AS INPUT,” and filed on Jun. 6, 2016; the aforementioned priority applications being hereby incorporated by reference in their respective entireties.

BACKGROUND

[0002] Wearable device technology in consumer electronics is rapidly being integrated into routine user activities, such as sporting activities, content viewing or browsing, and task-oriented activities (e.g., gaming). Furthermore, wireless networks typically utilize protocols that enable wireless devices to detect signal sources from other devices for initiating data and communication links. Such networks are typically implemented using networking hardware, which may be incorporated in various wireless network devices, such as access points (APs), peer-to-peer (P2P) devices, wireless local area network (LAN) equipped devices, and the like—each advertising a unique identity (e.g., a media access control (MAC) address) indiscriminately to devices within range. Connections may be established with such devices to transmit and receive data.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] The disclosure herein is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings in which like reference numerals refer to similar elements, and in which:

[0004] FIG. 1A illustrates a block diagram of an example modular sensing device for use in controlling a remotely operated device, according to examples described herein;
[0005] FIG. 1B is a block diagram illustrating an example modular sensing device, as shown and described herein;
[0006] FIG. 2 is a flow chart describing an example method of generating commands by a modular sensing device, in accordance with examples described herein;
[0007] FIG. 3 illustrates an example modular sensing device insertable into a plurality of compatible apparatuses;
[0008] FIG. 4 is a block diagram illustrating an example modular sensing device for acquiring virtual or digital resources;
[0009] FIG. 5 is a block diagram illustrating an example virtual resource management system utilized by a wearable device;
[0010] FIG. 6 is a flow chart describing an example method of acquiring virtual resources using a wearable device;
[0011] FIGS. 7A and 7B are flow charts describing example methods of acquiring and utilizing virtual resources in connection with gameplay;
[0012] FIGS. 8A, 8B, and 8C illustrate unique identifier logs and association tables utilized in connection with virtual resource association, acquisition, and allocation;

FIG. 9 illustrates a wearable device pairing triggering virtual resource data sharing in connection with gameplay;

FIGS. 10A and 10B illustrate a wearable device pairing triggering an interactive mode between a pair of proximate users;

FIG. 11 is a flow chart describing an example method of implementing an interactive mode between a pair of user utilizing a corresponding pair of wearable devices;

FIG. 12 is a flow chart describing an example method of initiating a training mode on a wearable device in connection with a self-propelled device;

FIG. 13 is a flow chart describing an example method of implementing a wearable device in a sword mode;

FIG. 14 is a block diagram of an example computer system upon which examples described herein may be implemented;

FIG. 15 is a block diagram of a mobile computing device upon which examples described herein may be implemented;

FIG. 16 is a block diagram of an example portable sensing device upon which examples described herein may be implemented;

FIG. 17 illustrates an embodiment of multiple sensing devices that concurrently provide input for a program or application which utilizes the inputs, along with inferences which can be made about a person or object that carries the devices, according to one or more examples;

FIG. 18 illustrates a system which concurrently utilizes input from multiple modular sensing devices in connection with execution of an application or program;

FIG. 19 illustrates an example of a modular sensing device insertable into a wrist worn apparatus; and

FIG. 20 illustrates an implementation of the modularized sensing device, in accordance with examples described herein.

DETAILED DESCRIPTION

[0025] Examples described herein relate to a multi-modal modular sensing device, worn or carried by a user (e.g., as a wrist-worn device), to enable a variety of interactions with other devices through sense movement of the modular sensing device. Among other activities, examples provide for a modular sensing device that can individually, or in combination with another device (e.g., controller device, such as a mobile computing device) control other devices, interact with compatible devices of other users, and/or operate in connection with task-oriented activities (e.g., gameplay). In some examples, the modular sensing device corresponds to a wearable device (e.g., a watch, a pendant for a necklace, a hat, glasses) can be placed in multiple modes to, for example, control the characteristics of movement of another device, or interact with the real-world. For example, the modular sensing device can control acceleration and maneuvering of a self-propelled device.

[0026] In certain aspects, a portable modular sensing device can include an inertial measurement unit (IMU), and can be operated in multiple modes in which gestures (e.g., arm gestures) may be translated based on the particular mode of the wearable device. In one example, the modular device can be insertable into multiple devices, such as a wrist worn device, clothing, an accessory device (e.g., a toy sabre or sword), a wearable pendant, ring, hat, and the like.
The modular sensing device can further include output devices, such as lighting elements (e.g., one or more LEDs), an audio device (e.g., a speaker), and/or a haptic feedback system. In one example, the modular sensing device further includes a charging port (e.g., a mini-universal serial bus port) to enable charging of a power source included in the device.

[0027] According to some examples, a modular sensing device is operable to detect its own movement in three-dimensional space, using the IMU. In some implementations, the IMU can be an integrated device. Alternatively, the IMU can be implemented through a combination of sensors, such as a three-dimensional accelerometer or gyroscope. In certain examples, the modular sensing device can include a processor and memory to interpret the sensor data, and to communicate interpreted sensor data to another device (e.g., mobile computing device) using a wireless connection (e.g., BLUETOOTH). In variations, the sensor data can generate data that is either raw data or processed, depending on the processing resources that are selected for the portable device, and the processing load which is implemented for the portable device.

[0028] In many implementations, the modular sensing device can generate output (e.g., haptic, audio, and/or visual output) or control commands for operating a remote controlled device by utilizing state machines in memory that correspond to individual gestures. Depending on a selected mode of the modular sensing device, processing resources of the modular sensing device can monitor logical state machines or automations that correspond to specified sensor data combinations (e.g., based on gestures performed by a user). For example, a single state machine can correspond to an initial arm raising gesture performed by the user wearing a wrist-worn device having the modular sensing device. A sensor profile from a single sensor or multiple sensors included in the IMU (e.g., gyroscope sensor and/or accelerometer) can indicate the initial arm raising gesture, which can trigger a state transition for the state machine. This state transition can cause processing resources of the modular sensing device to automatically generate a control command to cause the remote controlled device to accelerate.

[0029] Additionally, a second state machine can correspond to an increased angle in the arm raising gesture. When the sensor profile of the IMU indicates the gesture above a certain predetermined angle, a state transition can be triggered for the second state machine, which can trigger an increased acceleration command to be generated automatically. Several state machines can be comprised in memory of the modular sensing device, and can each correlate to a specified sensor data profile, from simple single-action gestures, to complex multiple-motion gestures. Furthermore, an interpretation engine (e.g., a processor) can interpret each state transition for each state machine based on a particular mode and/or sub-mode of the modular sensing device.

[0030] As used herein, a “modular sensing device” can comprise an electronic device that includes sensor resources for detecting its own movement, and of dimension and form factor suitable for being insertable into any number of devices configured for receiving the device. Numerous examples of modular sensing devices are provided in the context of a “wearable device,” such as a wrist worn device (e.g., watch, watch band, bracelet). But as noted by other examples, variations to the type and form factor of a wearable device can vary significantly, encompassing, for example, eyeglasses, hats, pendants, armbands, glasses and various other form factors. While many examples describe functionality in the context of a wearable device, embodiments extend such examples to other form factors in which a modular sensing device may be inserted, such as wands, fobs, wielded toys, or mobile communication devices.

[0031] In many examples, the wearable device can include one or more sensors to detect the device’s own movements. In particular, a wearable device includes an accelerometer and/or a gyroscopic sensor. In some examples, sensor data, corresponding to user gestures performed by the user wearing the wearable device, can be translated into control commands or data packets to be transmitted and implemented based on the selected mode of the wearable device. According to many examples, the wearable device can include an inductive interface to inductively pair with other devices, which can trigger a specified mode on the wearable device. For example, an inductive pairing between the wearable device and a self-propelled device can trigger a drive mode in which the wearable device can be utilized by the user to operate the self-propelled device. Additionally or alternatively, the wearable device can include an input mechanism, such as an analog or digital button, that enables the user to select a particular mode and/or scroll through a series of modes for the wearable device.

[0032] Among other functionality, some examples described herein provide for alternative modes of operation, including, for example (i) a “drive mode” in which the wearable device is utilized to control a self-propelled device; (ii) a “control mode” in which the wearable device is utilized in connection with smart home devices; (iii) a “finding mode” or “finder mode” in which the wearable device is utilized to detect virtual or digital resources; (iv) a “mining mode” which can be initiated by the user to collect virtual resources when they are detected in the finder mode; (v) a “training mode” in which the wearable device is utilized in connection with a self-propelled device to assist the user in training for certain achievements or for increasing the user’s abilities to perform task-oriented activities (e.g., increasing skills for a subsequent game or sporting activity); (vi) a “sword mode” in which the wearable device provides feedback (e.g., haptic, audio, and/or visual feedback) when the user performs actions while holding an object; (vii) a “default mode” in which the device monitors for and detects other proximate wearable devices (e.g., wearable devices of the same type) which enables the users to pair with each other’s wearable devices; (viii) a “interactive mode” or “battle mode” selectable in response to two or more device pairings in which users are able to interact with each other with predetermined sets of actions (e.g., offensive and defensive actions learned and perfected by users practicing in the training mode); (ix) a “sharing mode” selectable in response to two or more device pairings in which users can share information stored in each other’s wearable devices, or user accounts associated with the wearable devices (e.g., sharing collected virtual resources discovered and mined in the finder and mining modes to be expended or consumed in a gameplay environment); and (x) a “gaming mode” in which the wearable device can be utilized in connection with a game.

[0033] Still further, numerous examples make reference to a “self-propelled” device. A self-propelled device can include, for example, a device that can be wirelessly and
remotely controlled in its movement, whether the movement is on ground, water, or air. For example, a self-propelled device can include a wirelessly and remotely controlled drone, car, airplane, helicopter, boat, etc. While conventional examples enable control of a self-propelled device, conventional approaches generally utilize a perspective of the device being controlled. While some conventional devices, for example, enable a computing device held by the user to project a perspective of the device under control, examples described herein enable control of such devices to utilize an orientation of the user. Specifically, some examples include a modular sensing device that can determine an orientation of the user, and further enable control of the self-propelled device through an environment that accommodates or is in the perspective of the user, based on the orientation of the user (as determined by the modular sensing device). By way of example, the control of a self-propelled device can be projected through an orientation or perspective of the user for purpose of a virtual environment.

Examples include a wireless communication module (e.g., a BLUETOOTH low energy module) that enables communication of sensor data (e.g., raw sensor data from the accelerometer or gyroscope sensor), or translated data (i.e., translations of the sensor data based on the selected mode of the wearable device). In some examples, the sensor data may be relayed for translation by a mobile computing device before being transmitted to another device (e.g., a paired wearable device or a paired self-propelled device). In other examples, processing resources of the wearable device can execute mode instructions, based on the selected mode, to translate the sensor data (e.g., via use of state machines that trigger control commands or feedback based on a selected mode of the modular sensing device) for direct transmission to one or more other devices, as described herein.

As used herein, “body part gestures” or “user gestures” include gestures performed by a user while utilizing the wearable device. For example, the wearable device may be a wrist-worn device, in which case the body part or user gestures may comprise arm gestures, and can include any number of physical movements or actions that affect the sensors of the wearable device when it is worn on the wrist. Such movements and actions can include shking, arm movements (e.g., raising, lowering, pointing, twisting, and any combination thereof), wrist movements, hand actions (such as grasping or grabbing), and like. However, the wearable device is not limited to wrist-worn devices, but may be utilized as a ring (e.g., a finger-worn device), an ankle-worn device, a neck-worn device, a head-worn device, a belt (e.g., a waist-worn device), etc. Thus, user gestures performed using the wearable device can be any actions or movements in which correlated sensor data from sensors of the device can be translated into commands, instructions, feedback, etc. depending on the mode of the wearable device.

Among other benefits, examples described herein achieve a technical effect of enhancing user interactivity with other devices and other users. Such interactivity may include utilizing the modular sensing device to control a self-propelled device, interact with other users of wearable devices, collect and share data, control smart home devices, interact with the real-world via a gaming interface through the modular sensing device, and the like.

One or more examples described herein provide that methods, techniques, and actions performed by a computing device, a modular sensing device, or a self-propelled device are performed programmatically, or as a computer-implemented method. Programmatically, as used herein, means through the use of code or computer-executable instructions. These instructions can be stored in one or more memory resources of the computing device. A programmatically performed step may or may not be automatic.

One or more examples described herein can be implemented using programmatic modules or components of a system. A programmatic module or component can include a program, a sub-routine, a portion of a program, or a software component or a hardware component capable of performing one or more stated tasks or functions. As used herein, a module or component can exist on a hardware component independently of other modules or components. Alternatively, a module or component can be a shared element or process of other modules, programs or machines.

Some examples described herein can generally require the use of computing devices, including processing and memory resources. For example, one or more examples described herein can be implemented, in whole or in part, on computing devices such as digital cameras, digital camcorders, desktop computers, cellular or smart phones, personal digital assistants (PDAs), virtual reality (VR), augmented reality (AR), or mixed reality (MR) headsets, laptop computers, printers, digital picture frames, and tablet devices. Memory, processing, and network resources may all be used in connection with the establishment, use, or performance of any example described herein (including with the performance of any method or with the implementation of any system).

Furthermore, one or more examples described herein may be implemented through the use of instructions that are executable by one or more processors. These instructions may be carried on a computer-readable medium. Machines shown or described with figures below provide examples of processing resources and computer-readable mediums on which instructions for implementing examples can be carried and/or executed. In particular, the numerous machines shown with examples include processor(s) and various forms of memory for holding data and instructions. Examples of computer-readable mediums include permanent memory storage devices, such as hard drives on personal computers or servers. Other examples of computer storage mediums include portable storage units, such as CD or DVD units, flash memory (such as carried on smart phones, multifunctional devices or tablets), and magnetic memory. Computers, terminals, network enabled devices (e.g., mobile devices, such as cell phones) are all examples of machines and devices that utilize processors, memory, and instructions stored on computer-readable mediums. Additionally, examples may be implemented in the form of computer programs, or a non-transitory computer usable carrier medium capable of carrying such a program.

System Description

FIG. 1A illustrates a block diagram of an example modular sensing device for use in controlling a remotely operated device, according to examples described herein. The modular sensing device 100 shown in FIG. 1A may be integrated with a wearable device 102 (e.g., a wrist-worn device, pendant, clothing, a hat, eyeglasses, an ankle-worn device, a ring, etc.), or can be inserted into a compartment
or otherwise included as a part of the wearable device 102. The wearable device 102, or modular sensing device 100 disposed therein, can include a mode selector 110, such as an analog button that enables the user to select a mode of the device. In one example, repeated user input 111 on the mode selector 110 enables the user to scroll through a list of available modes. Such modes include, but are not limited to, a drive mode, a control mode, a finder mode, a mining mode, a training mode, a sword mode, a default mode, an interactive or battle mode, a sharing mode, and a gaming mode. As shown in the block diagram, a user input 111 on the mode selector 110 can cause a processor 120 of the modular sensing device 100 to generate an output 132 that indicates the particular mode selected. For example, the modular sensing device 100 can include a number of output devices 130, such as an audio device, a haptic system, and/or visual elements (e.g., an LED array or display). Each user input 111 on the mode selector 110 can trigger an audio, haptic, and/or visual output 132, indicating the particular mode (e.g., a drive mode). As an example, the output 132 can comprise a voice output that states the selected mode, or a combination of voice, visual, and haptic output.

Additionally or alternatively, the user may connect the wearable device 102 with a mobile computing device, such as the user’s smart phone or tablet computer. Mode selection may be performed automatically by the user initializing a designated application of the mobile computing device, such as a smart home application, a controller application (e.g., to control a self-propelled device), a gaming application, and the like. In variations, the user can execute a designated application in connection with the wearable device 102 that enables the user to scroll through the various modes. The user may scroll through the modes on the mobile computing device, or via successive selection inputs on the mode selector 110, which can trigger the mobile computing device to display a selectable mode. In other variations, multiple types of inputs can be performed in the mode selector 110, such as tap gestures and tap and hold gestures, which can correlate to scrolling through the modes and selecting a particular mode respectively. As provided herein, the mode selector 110 can be an input mechanism such as an analog or digital button, a touch panel such as a touchpad, or a miniature touch-sensitive display device.

According to examples provided herein, the modular sensing device 100 can include a memory 115 that stores mode instruction sets executable by the processor 120 based on the mode selected by the user. Each mode instruction set can cause the processor 120 to interpret sensor data 137 in a particular manner. Along these lines, same or similar gestures 106 performed by the user can result in different generated outputs 132 or commands 108 depending on the mode selected by the user. In some implementations, selection of a particular mode can cause the processor 120 to initiate a communications module 125 of the modular sensing device 100 to establish a wireless connection with another device. In one example, the modular sensing device 100 can include a BLUETOOTH low energy module to enable communications with one or more other devices, such as a second modular sensing device or a remotely operated device 140.

The modular sensing device 100 can be relatively small in size compared to current computing devices, and in many implementations, the modular sensing device 100 does not include a power-intensive memory and computational resources. In such implementations, the memory and/or memory controller can be comprised of or implement a series of state machines that, when transitioned, can trigger a particular output automatically. Further description of the state machine implementations is provided below with respect to FIG. 1B. Furthermore, examples described herein with respect to the drive mode of the modular sensing device 100 can also be implemented in the state machine examples described herein.

As provided herein, the memory 115 can include a drive mode instruction set 117 executable by the processor 120 in response to a user input 111 on the mode selector 110. In some aspects, execution of the drive mode instructions 117 can cause the processor 120 to initiate the communications module 125 to establish a communications link 104 with a proximate remotely operated device 140. In variations, the modular sensing device 100 can include an induction interface 127 which can trigger the communication link 104 between the modular sensing device 100 and the remotely operated device 140. For example, upon selecting the drive mode, the user can place the wearable device 102 within inductive range (e.g., −2.5 cm) of the remotely operated device 140, which can include a corresponding induction interface. The inductive transfer of communication information can enable the modular sensing device 100 to establish the communication link accordingly.

According to examples described, the modular sensing device 100 can further include an inertial measurement unit (IMU) 135 that can comprise a gyroscope and an accelerometer for accurate measurement of the modular sensing device’s 100 movement and orientation. In variations, the IMU 135 can further include a magnetometer to, for example, assist in calibration based on the orientation. Once the communication link 104 is established in the drive mode, the processor 120 can monitor the sensor data 137 for the particular gestures 106 being performed by the user. In several examples, the gestures 106 can correspond to the user raising or lowering an arm, and/or performing additional arm motions. The sensor data 137 from the IMU 135 can comprise movement, position, and/or orientation information that the processor 120 can interpret in accordance with the drive mode. For example, gestures 106 performed by the user can be detected by the processor 120 via sensor data 137 from the IMU 135. Each of the gestures 106 can be interpreted by the processor 120 as one or more control commands 108 to be executed by the remotely operated device 140.

In one example, the drive mode can be automatically initiated in response to a particular detected gesture, regardless of the current mode of the modular sensing device 100. This gesture can correspond to a distinct sensor data signature that, when detected by the processor 120 in executing any mode, overrides that mode and initiates the drive mode automatically. Thus, upon detection of the distinct sensor data signature, the processor 120 can automatically initiate the communications module 125, establish the communications link 104 with the remotely operated device 140, and generate control commands 108 based on the detected gestures 106 performed by the user in the sensor data 137. The modular sensing device 100 may then dynamically transmit such control commands 108 to the remotely operated device 108 for execution as they are generated by the processor 120. In one example, the specific gesture
corresponds to a pushing motion with an arm wearing the wearable device 102 performed by the user. As provided herein, this pushing motion can correspond to a specified sensor data signature not used for any other mode, and is therefore distinct to enable the processor 120 to identify it irrespective of the current mode of the modular sensing device 100.

[0049] In certain examples, gestures 106 such as raising an arm can cause the processor 120 to generate acceleration commands to drive away from the user. Lowering the arm can cause the processor to generate deceleration and/or reverse commands. Further, moving the arm from side to side can cause the processor 120 to generate steering or directional commands. For example, moving the arm left can cause the remotely operated device 140 to turn left, and moving the arm right can cause the device 140 to turn right as the device 140 travels away from the user. Such control commands 108 may be processed by a controller of the remotely operated device 140, or may be directly executed on the drive motors of the device 140 in order to accelerate and maneuver the device 140 in accordance with the gestures 106 performed by the user.

[0050] Furthermore, in the drive mode, angular thresholds can be established in the drive mode instructions 117 that can determine the manner in which the processor 120 interprets the sensor data 137. When such thresholds are crossed, the processor 120 can alter interpretation of the sensor data 137 into alternative commands 108. For example, as the user raises the arm above an angular threshold (e.g., 45 degrees), and/or changes an orientation of the arm (e.g., palm down to palm up), the processor 120 can alter the interpretation of the sensor data 137 such that the remotely operated device 140 drives towards the user as the arm is raised. Furthermore, in such a state, the directional interpretation of the sensor data 137 can be reversed such that moving the arm left can cause the remotely operated device 140 to turn right, and moving the arm right can cause the remotely operated device 140 to turn left. This directional reversal triggered by the angular threshold, and in combination with the change in orientation of the user’s arm, can create a palm control illusion of the remotely operated device 140 by the user. Thus, in the drive mode, specified gestures detected in the sensor data 137 (e.g., the user’s arm rotating or crossing an angular threshold) can trigger the processor 120 to interpret the sensor data 137 differently, or inversely from an initial interpretation, in order to produce the illusion.

[0051] FIG. 1B is a block diagram illustrating an example modular sensing device, as shown and described herein. As provided herein, the modular sensing device 150 can be space limited, and can only include a limited amount of memory and computational resources. In such implementations, the modular sensing device 150 can represent each possible gesture that can be performed by a user as a state machine. Thus, for each gesture, a state machine corresponding to that gesture can either positively identify its gesture or negatively identify its gesture. When a positive gesture is identified, the state machine triggers a state transition which can cause an output generator 160 to generate a particular output accordingly. As provided herein, the output generator 160 and memory 180 can comprise a hardware controller implementing the various state machines to generate the output commands 164.

[0052] As an illustration, the modular sensing device 150 can include a memory 180 implementing a number of state machines (e.g., SM1 181, SM2 183, SM3 185, SM4 187, . . . , SMN 189), each being associated with a particular gesture. For example, SM1 181 can be associated with the user raising an arm, SM2 183 can be associated with the user lowering an arm, SM3 185 can be associated with the user pointing an arm to the right, and SM4 187 can be associated with the user pointing an arm to the left. Furthermore, any number of state machines may be implemented in the memory 210 representing any number of gestures. At any given time step, the state machines can be instantiated for each gesture type, and each state machine can continuously inspect the instantaneous sensor data 167 from the IMU 165 in order to initialize or instantiate, transition individual states along a state string, terminate a current state string, or trigger an accept state or final state. When the final state is triggered, this means that the particular gesture corresponding to that state machine has been performed by the user.

[0053] According to examples, each state machine can consist of a finite set of states (a fixed string of states), one or more inputs, one or more outputs, a transition function, and an initial state. The state machine can be linked to a particular gesture by way of a sensor data signature, which can comprise an accelerometer data signature, gyroscope data signature, or a combination of both. Furthermore, the state machine can be linear and directional, with each state having a particular error tolerance in its sensor data signature. A final state of a state machine can thus be triggered when the full sensor data signature of a particular gesture is matched to the sensor data 167 generated by the IMU 165.

[0054] In some aspects, if at any time after instantiating an associated gesture corresponding to a respective state machine is not being performed, the input string for that state machine, and for that particular instantiation, automatically terminates. At any given instance, the state machine either terminates from the outset (e.g., an initial aspect of the sensor data signature for the gesture is not matched), or instantiates the input string towards the final state. At any given state along the input string, the state machine can terminate if the gesture being performed diverges from the error tolerances built into the state machine. If, however, each state along the input string is transitioned accordingly (i.e., the sensor data 167 from the IMU 165 matches the sensor data signature for that state machine within its error tolerances), the final state is triggered for that state machine.

[0055] The memory 180 can include a state machine register 182 that can report such final state transitions 188 to an output generator 160 of the modular sensing device 150. The output generator 160 can be configured based on the particular mode of the modular sensing device 150. Accordingly, final state reports 188 from individual state machines can be interpreted differently, or can cause a particular output, depending on the mode. In other words, the output from the modular sensing device 150 for a particular gesture (e.g., a backhanded swinging motion) can be different depending on the mode initiated via the mode selector 155. Furthermore, certain final state reports 188 from the state machine register 182 can correspond to submode triggers 186 for the particular mode. Such submode triggers 186 may not trigger an output, but rather trigger the output generator 160 to alter interpretation of certain final state reports 188 in order to generate an alternative output.
Such outputs generated by the output generator 160 can comprise control commands 162 to operate a remotely operated device, such as acceleration, steering, and deceleration commands. In some aspects, the output generator 160 can generate output commands 164 for the modular sensing device’s 150 haptic system 192, visual system 194, and/or audio system 196. Thus, as final state reports 188 are received, the output generator 160 can cause the haptic 192, visual 194, and audio 196 systems of the modular sensing device 150 to produce individual or combined outputs. As described herein, such outputs can include vibration feedback or guidance, colored lights or display output, tactile outputs such as audible chimes that indicate positive and/or negative feedback, and voice output.

In the example of the drive mode described with respect to FIG. 1A, each completed arm gesture can correspond to a final state machine report 188, and can cause the output generator 160 to generate a control command 162 accordingly. In the example shown in FIG. 1B, the user raising an arm above a certain angle (e.g., five degrees) can cause a state machine corresponding to that gesture to transition to a final state. The state machine reporter 182 can report the final state 188 of that state machine to the output generator 160, which, based on the drive mode interpretation, can generate a control command 162 to initiate acceleration of the remotely operated device. This control command 162 may then be transmitted to the remotely operated device via the communications module 175 of the modular sensing device 150.

Referring to both FIGS. 1A and 1B, in certain modes, the communications module 125, 175 can act as a signal or beacon sensor which can provide signal data to the processor 120 or output generator 160. Based on the mode, the processor 120 or output generator 160 can generate a specified response based on the signal data. For example, the signal data can indicate a unique identifier of the signal source. The memory 115, 180 can include a UID log of specified signal sources that allow the user to interact with such sources based on the mode of the modular sensing device 100, 150. For example, if the signal source corresponds to another modular sensing device, the user can be enabled to select a mode in which the user can interact with the other user (e.g., play a physical game involving gestures).

As described with respect to FIG. 1A, the interpretation of the sensor data 137 can be altered based on a particular gesture trigger, such as the user’s arm exceeding an angular threshold. The analog with respect to FIG. 1B comprises a state machine in the memory 180 that correlates to the particular threshold gesture. Thus, when the sensor data 167 indicates the specified threshold gesture, the correlated state machine can transition to its final state, which can be reported to the output generator 160 accordingly. Such a final state report 188 can comprise a sub-mode trigger 186, in that the sub-mode trigger 186 causes the output generator 160 to alter interpretation of certain final state reports 188 while remaining in the same mode. In the drive mode example, a sub-mode trigger 186 can correspond to the user raising an arm above a certain threshold (e.g., 45 degrees), which can cause the output generator 160 to, for example, reverse the interpretation of the final state reports 188 corresponding to the directional commands for the remotely operated device.

Additionally, each mode of the modular sensing device 150 can include any number of sub-mode triggers 186 that cause the output generator 160 to alter an interpretation of or disregard a particular final state report 188 corresponding to a specified sensor data signature. For example, the drive mode can include a set of angular gesture thresholds (e.g., raising an arm beyond 45 degrees, lower the arm below 45 degrees, turning the arm from palm down to palm up). A state machine can be dedicated—within the specified mode—to a sensor data signature indicating the user gesture across a gesture threshold. Thus, when the user gesture crosses the gesture threshold, the dedicated state machine can transition to its final state, which, when reported to the output generator 160 (i.e., as a sub-mode trigger 186), can cause the output generator 160 to alter interpretation of certain final state reports 188 within that mode. In some aspects, disregarded final state reports prior to the sub-mode trigger can now trigger a specified output (e.g., an audio, haptic, and/or visual output, or a particular control command 162).

In further examples, when the modular sensing device 150 is in a particular mode, a specific complex gesture—represented by a particular state machine in the memory 180—can cause the output generator 160 to reconfigure its interpretation of certain final state reports 188, execute a sub-mode, or automatically initialize a different mode for the modular sensing device 150. Accordingly, for any given instance, sensor data 167 from the IMU 165 can continuously cause the various state machines to instantiate, terminate, instigate state transitions, and/or transition to a final state. In aspects described, only when a state machine transitions to its final state does the output generator 160 generate output commands 164 and/or control commands 162 to provide feedback, operative control over a remotely operated device, guidance via the output devices, and/or task-based instructions (e.g., in accordance with a particular game).

Furthermore, the various aspects performed by the modular sensing device 100 described with respect to FIG. 1A may also be performed by example modular sensing devices 150 as shown and described with respect to FIG. 1B. Thus, the execution of designated modal instruction sets by the modular sensing device 100 of FIG. 1A—in which the processor 120 directly interprets the sensor data 137 based on the executing instruction set—may be substituted by the state machine examples as described with respect to FIG. 1B. In other words, the limited memory and computational resources of the modular sensing device 150 of FIG. 1B may be compensated by attributing sensor data signatures to state machines, requiring less memory and processing without sacrificing functionality.

Methodology

FIG. 2 is a flow chart describing an example method of generating output commands by a modular sensing device, in accordance with examples described herein. In the examples described with respect to FIG. 2, reference may be made to reference characters representing like features as shown and described with respect to FIGS. 1A and 1B. Furthermore, the methods and processes described with respect to FIG. 2 may be performed by an example modular sensing device 100, 150 as shown and described with respect to FIGS. 1A and 1B. Referring to FIG. 2, the modular sensing device 150 can receive a mode selection input (200). In some examples, the mode selection input can
comprise the user physically pushing an analog mode selector 155 on the modular sensing device 150. Additionally or alternatively, the mode selection input can comprise a specified gesture performed while using the modular sensing device 150.

[0065] In response to the mode selection input, the modular sensing device 150 can interpret the final state machine reports 188 in accordance with the mode selected (205). As provided herein, the modular sensing device 150 can operate in any number of modes, with each mode corresponding to controlling a remote device (e.g., drive mode), generating user tasks and feedback in connection with a remote device (e.g., training mode), generating user tasks and feedback in conjunction with another modular sensing device (e.g., playing a real-world game with another user), and/or generating standalone user tasks and feedback. As provided herein, user tasks can comprise instructions or suggestions to the user via the output devices of the modular sensing device in accordance with the selected mode. Such instructions or suggestions can be generated based on a programmed game, in which the user is to utilize the modular sensing device 150 to perform gestures and actions, search for a certain signal source, cause the remotely operated device 140 to perform a set of actions, and the like. Furthermore, feedback can comprise reactive output to the tasks actions to be performed by the user. For example, the feedback can comprise audio, visual, and/or haptic responses to actions indicating affirmative or negative completion of such tasks. In one example, the feedback indicates to the user whether an instructed task or gesture (e.g., indicated by the sensor data 167 and correlated state machine) has successfully been performed.

[0066] According to several examples, the final state machine reports 188 can be correlated to a specified output. Thus, based on each final state machine report 188 the modular sensing device 150 can generate commands in accordance with the mode and, when relevant, the sub-mode of the device 150 (220). As provided herein, such generated commands can include output commands 164 to output audio (222), visual (226), and/or haptic (224) output on the modular sensing device 150. Additionally or alternatively, the modular sensing device 150 can generate control commands 162 (228) for operating a remotely operated device 140. In either case, the modular sensing device 150 can transmit or execute the commands accordingly (225).

[0067] At any given time in any given mode, the modular sensing device 150 can identify a sub-mode trigger 186 in the final state machine reports 188 (210). In response to the sub-mode trigger 186, the modular sensing device 150 can reconfigure interpretation of one or more final reports from one or more corresponding state machines (215). Based on each final state machine report 188, the modular sensing device 150 can generate commands in accordance with the mode and the sub-mode of the device 150 (220), including the altered interpretations based on the sub-mode trigger(s) 186. As discussed above, such generated commands can include output commands 164 to output audio (222), visual (226), and/or haptic (224) output on the modular sensing device 150. Additionally or alternatively, the modular sensing device 150 can generate control commands 162 (228) for operating a remotely operated device 140. In either case, the modular sensing device 150 can transmit or execute the commands accordingly (225).

[0068] Modular Sensing Device

[0069] FIG. 3 illustrates an example modular sensing device insertable into a plurality of compatible apparatuses. The modular sensing device 300 shown in FIG. 3 can comprise various components and modules of the modular sensing devices 100, 150 shown in the block diagrams of FIGS. 1A and 1B. Referring to FIG. 3, the modular sensing device 300 can include a number of output devices, such as an LED array 310, an output audio device 320 (e.g., a speaker), and a haptic driver 360 (included within the device). Furthermore, the modular sensing device 300 can include a mode selector 330, which can comprise an analog or digital button to enable the user to select a particular mode of the device 300 by, for example, scrolling through a stored series of modes. The modular sensing device 300 can further include memory and processing resources 365 that can execute the selected mode (either in the state machine implementation (FIG. 1B) or the executed instruction set implementation (FIG. 1A) described herein).

[0070] In various aspects, the modular sensing device 300 also includes a communications interface 370 (e.g., a BLUETOOTH low energy, WiFi, WiGig, WiMAX, or cellular radio interface), and an IMU 340 to provide the memory and processing resources 365 with sensor data for detecting gestures performed by the user. As described herein, depending on the mode and sub-mode of the device 300 the memory and processing resources 365 interpret the sensor data to generate outputs via the output devices 310, 320, 360 and/or commands or responses to be output to a connected device via the communications interface 370 (e.g., a remotely operated device or another modular sensing device). Furthermore, in some implementations, the modular sensing device 300 can include an input interface 350 (e.g., a mini-USB port) to enable charging of one or more batteries and/or uploading of additional mode instructions. In variations, the modular sensing device 300 can include an induction interface to charge one or more batteries and/or to enable inductive pairing with a second device to establish a wireless communications link.

[0071] In the various examples described herein, the modular sensing device 300 can be insertable into or otherwise attachable to any number of compatible apparatuses, such as wearable devices 395 (wrist devices, rings, pendants, hats, glasses, etc.) wielded devices 385, companion toys or dolls, and the like. Furthermore, the modular sensing device 300 can be implemented in various other form factors, can be sewn into clothing, or can be mounted, glued, or otherwise attached to other apparatuses. Such apparatuses can each include a module holder 397, 397 into which the modular sensing device 300 may be inserted or otherwise mounted or attached. Thus, according to examples provided herein, the user can utilize the apparatuses into which the modular sensing device 300 has been inserted or attached, to perform various gestures in accordance with a selected mode of the modular sensing device 300.

[0072] Finder and Mining Mode

[0073] FIG. 4 is a block diagram illustrating an example portable sensing device operable as a signal detection system for acquiring virtual or digital resources. In more detail, a wearable device 400 may be a standalone device or may be incorporated with a mobile computing device running an application specific to detecting and recording signals from network devices, and searching for those network devices associated virtual resources. Standalone devices may include a pocket device, such as a key fob, a wrist-worn...
device or other wearable device (as described herein), and the like. For mobile computing environments, the wearable device 400 may utilize mobile computing device resources, such as device hardware/firmware, or operate utilizing a combination of hardware and software. For example, the wearable device 400 may be enabled upon initiation of an application on the mobile computing device. Additionally or alternatively, the wearable device 400 may continue to run during hibernation or sleep mode of the application using background or standby resources. Alternatively, the wearable device 400 may be enabled in the finder mode 406 and mining mode 407 upon user input on a mode selector 405 of the wearable device 400, as shown and described with respect to FIG. 1. Selection and execution of the finder and/or mining mode can generate an execution command 408, which can enable the processing resources of the wearable device 400 to execute the respective mode instructions for translating sensor data and providing feedback responses.

[0074] Referring to FIG. 4, the wearable device 400 can include a signal interface 410 to detect emitted signals from any number of network-enabled devices 460. Such emitted signals may be background signals advertising the presence and media access control (MAC) addresses of such devices 460. For example, at time t1, the wearable device 400 may be within wireless range of a laptop computing device 462 having a unique identifier UID 1. The UID of the laptop computing device 462 (i.e., UID 1) may correspond to the MAC address of the laptop computing device 462. The wearable device 400 can include a UID recorder 420, which can receive UIDs 412 from the signal interface 410 and record the respective UIDs 412 in a UID log 432 of a local memory resource 430. Accordingly, the UID recorder 420 can log the laptop computing device’s 462 UID (i.e., UID 1) in the UID log 432.

[0075] As a further example, at time t2, the wearable device 400 may be within wireless range of an access point 463 having a MAC address UID 2. According to examples described herein, the signal interface 410 can receive the MAC address of the access point 463. For example, the signal interface 410 can receive a beacon from the access point 463 which includes the access point’s 463 MAC address (UID 2). The signal interface 410 can communicate UID 2 to the UID recorder 420, which can log UID 2 in the UID log 432.

[0076] According to some examples, the wearable device 400 can establish a network connection, e.g., via network 480, with a host server 470. Upon connecting with the host server 470, the wearable device 400 can transmit UID log data 482 from the local UID log 432 to the host server 470. Accordingly, the host server 470 can receive UID log data 482 and associate the respective UIDs 412 from the local UID log data 482 with a user account 476 associated with a user of the wearable device 400. In such examples, the remote host server 470 can perform any number of functions described herein. For example, the host server 470 can utilize the UID log data 482 from the wearable device 400 to identify the network devices (i.e., the laptop computing device 462 and the access point 463) with which the wearable device 400 was associated. The host server 470 can store a universal association list 472 that may list associations between, for example, UID 1 of the laptop computing device 462 and a specified virtual resource. Accordingly, the host server 470 can attribute an amount of the specified virtual resource associated with UID 1 to the user account 476 associated with the wearable device 400. Thus, if the specified virtual resource associated with UID 1 was, for example, a virtual ore, then the host server 470 could attribute a predetermined amount of virtual ore to the user account 476.

[0077] Additionally or alternatively, the host server 470 may perform a lookup in the universal associations list 472 and identify that UID 2 (corresponding to the access point 463) does not yet have an associated virtual resource. Thus, the host server 470 may select, either sequentially or randomly, a virtual resource from a virtual resource catalog 474 and log a new association between the selected virtual resource and UID 2 in the universal associations list 472. Accordingly, an amount of the new selected resource can be attributed to the user account 476, in which respective virtual resources may be accumulated based on wireless detection events between the wearable device 400 and the network devices 460.

[0078] Additionally or alternatively, one or more operations described with respect to the host server 470 may be performed by the wearable device 400 itself. For example, the wearable device 400 can include an association engine 440 which, upon detection of a wireless signal from a network device 460, can determine whether that device is locally associated with a virtual resource. Thus, the memory resource 430 of the wearable device 400 can include an association table 434 which may include predetermined associations 435 between the network devices 460 and virtual resources. As an example, at time t3, the wearable device 400 can come within wireless range of a mobile device 464, which can announce its presence using MAC address UID 3. The signal interface 410 can receive UID 3, and the association engine 440 can perform a lookup in the association table 434 to determine whether UID 3 is associated with a particular virtual resource (e.g., virtual oil). Thus, the wearable device 400 may flag the association between UID 3 and virtual oil until a connection with the host server 470 is established to attribute the virtual oil to the user account 476.

[0079] Alternatively, the wearable device 400 can include a resource allocator 450 and the memory resource 430 can include a virtual resource log 436, comprising an accumulation of the virtual resources collected by the user. The association engine 440 can transmit the associations 435 to the resource allocator 450, which in turn, can select and compile respective amounts of accumulated virtual resources (e.g., five units of virtual oil corresponding to the UID 3 detection event) in the virtual resource log 436.

[0080] According to one or more examples, at time t4, the wearable device 400 may come within range of a tablet computing device 465. UID 4, associated with the MAC address of the tablet computing device 465 may be received by the signal interface 410 and communicated to the association engine 440, which can determine from the association table 434 that UID 4 is associated with, for example, virtual coal. The association engine 440 can communicate the UID 4 association to the resource allocator 450, which can log an amount of virtual coal in the virtual resource log 436. Additionally or alternatively, if the virtual resources (e.g., virtual coal) of all recently detected devices 460 are logged in the virtual resource log 436, the signal detection
system can flush the UID log 432, since the UIDs of the network devices 460 are already known and associated in the association table 434.

[0081] Upon network connectivity with the host server 470 over the network 480, the signal detection system can transmit virtual resource log data 484 to the host server 470 to enable the host server 470 to attribute collected virtual resources to the user account 476 associated with the user of the wearable device 400. Various alternatives are contemplated. For standalone devices, the wearable device 400 can communicatively couple with a user’s computing device, such as a smartphone or tablet computer. The virtual resource log data 484 may then be communicated to the user’s computing device which can, in turn, attribute the collected virtual resources to the user’s account locally or via network connection to the host server 470. The collected virtual resources may be expended as a result of gameplay, or via various incentive-based programs increasingly prevalent in the mobile application marketplace to induce users to interact with stipulated content.

[0082] Additionally or alternatively, the consumption, accumulation, and/or expenditure of collected virtual resources may be performed dynamically in connection with task-oriented operations (e.g., gameplay in the game mode) on a mobile computing device 495 running a respective application utilizing such virtual resources. Further, a standalone wearable device 400 may incorporate near field communication (NFC) technology such that the supplementary task of collecting virtual resources may be performed during a user’s routine daily activities, and the user can compile the virtual resources via NFC link with the user’s computing device 495.

[0083] In accordance with the above examples, at time t5, the wearable device 400 may come within range of a mobile device 466 having a unique identifier (e.g., MAC address) UID 5. The signal interface 410 can communicate the mobile device’s 466 identifier to the UID recorder 420 which can record UID 5 in the UID log 432 of the local memory resource 430. The mobile device’s 466 identifier may be further communicated to the association engine 440 which can perform a lookup in the association table 434 to determine whether UID 5 is associated with a virtual resource.

[0084] If UID 5 is not associated with a virtual resource, the wearable device 400 can subsequently utilize the UID log 432 to communicate UID log data 482 to the host server 470 to receive an association between UID 5 and a virtual resource. Alternatively, the wearable device 400 can include a local virtual resource catalog, similar to the virtual resource catalog 474 of the host computer 470, and can perform a sequential or randomized selection of a virtual resource (e.g., virtual wood) to associate with UID 5.

[0085] If UID 5 has already been associated with a virtual resource (e.g., virtual wood), as resultant from the lookup performed by the association engine 440, then the association 435 can be communicated to the resource allocator 450, which can allocate a predetermined amount of virtual wood in the virtual resource log 436. Once a connection between the wearable device 400 and the host server 470 is established, the virtual resource log data 484, which can comprise the accumulated virtual resources since the last established connection, can be communicated to the host server 470 for attribution to the user’s account 476. Once the virtual resource log data 484 is communicated, the virtual resource log 436 in the wearable device 400 can be flushed.

[0086] Various limitations to resource allocation are realized. For example, the signal interface 410 of the wearable device 400 may be restricted to only receive virtual resources associated with network devices 460 upon establishing a connection with the respective device, as opposed to merely detecting the network device 460. Furthermore, the allocated amounts of a particular virtual resource associated with multiple network devices 460 may be diverse. For example, the resource allocator 450 may allocate more virtual resources when the wearable device 400 comes within contact of a less accessible access point (e.g., an access point at a remote location). Furthermore, at any given expiration time (e.g., after an expiration period or a time-limited gaming session), the UID log 432, the association table 434, and/or the virtual resource log 436 may be reset or otherwise reconfigured.

[0087] Examples described with respect to FIG. 4 illustrate resource association and allocation being performed by the wearable device 400. However, according to some aspects, the mobile computing device 495 can perform various processes as described with regard to the wearable device 400. For example, the mobile computing device 495 can receive the UID data 484 for each detected network device 460, and can locally perform lookups for the associations and allocations of the virtual resources. Accordingly, the memory 430 shown as a component of the wearable device 400, may be included in the mobile computing device 495 and/or accessible by the mobile computing device 495, running a designated application, in the host server 470.

[0088] As further described herein, upon detecting a particular network device 460 associated with a particular virtual resource in the finder mode, the wearable device 400 can initiate the feedback mechanism to notify the user that virtual resources are proximate to the user’s current location. The feedback output can include haptic feedback in combination with lighting and/or audio feedback. In many aspects, the user can be notified and enabled to select the mining mode, in which the user can “mine” the virtual resource from the network device 460. In the mining mode, the feedback mechanism can be initiated and adjusted dynamically to enable the user to search for a direction towards the virtual resource to acquire a certain allocation, and/or to locate the exact position of the network device to acquire a certain allocation, as described below with respect to FIG. 6.

[0089] In still further implementations, the wearable device 400 can utilize location-based resources (not shown) in order to identify virtual resources associated with a particular waypoint. For example, a gameplay environment may preconfigure virtual resources in GPS locations in the real world. The wearable device 400 can identify such locations (e.g., when the wearable device 400 is within a certain proximity of a particular waypoint) and provide feedback via a feedback mechanism that enables the user to search for and mine the virtual resource associated with the waypoint.

[0090] FIG. 5 is a block diagram illustrating an example virtual resource management system utilized by a portable sensing device (e.g., wearable device 400), or a mobile computing device 495 executing an application in connection with a wearable device 400. The virtual resource management system 500 can include features from the
wearable device 400 described with respect to FIG. 1A. Furthermore, one or more components of the virtual resource management system 500 may be comprised in any number of electronic devices, such as smartphones, tablet computers, personal computers, laptop devices, wearable computing systems, and the like. Accordingly, the virtual resource management system 500 can comprise an application-based program running on any of such devices, enabling a user to collect, accumulate, expend, and consume virtual resources in conjunction with task-oriented operations performed on the device. For example, the user's device may run an application specific to virtual resource gameplay, rendering a gaming environment 578 on a display 590. The user may perform interactions 594 using the display 590 in order to utilize compiled virtual resources in the virtual resource log 565, as shown in FIG. 5.

Additionally or alternatively, the virtual resource management system 500 shown and described with respect to FIG. 5 can be provided with UIDs 512 from, for example, a wearable device in a finder mode that operates to locate virtual resources as a user walks or otherwise travels across various networks and comes within network range of various network devices. In some aspects, the wearable device can transmit UIDs of various detected network devices to the mobile computing device, which can connect with the host server 570 to determine whether the detected network devices are associated with virtual resources. If so, the mobile computing device can transmit a confirmation signal to the wearable device which can provide feedback to the user (e.g., audio, haptic, and/or visual feedback) indicating that virtual resources are nearby. The mobile computing device or the wearable device itself can determine a direction towards the network device associated with the virtual resource. Furthermore, the generated feedback on the wearable device can enable the user to select the mining mode, allowing the user to first identify the direction towards the virtual resource(s) (e.g., via dynamically provided feedback indicative of the direction the user points the wearable device), and optionally locate the actual position of the network resource device in order to be allocated with a certain amount of the virtual resource. In some aspects, the wearable device can include an IMU to determine the user's orientation and a direction in which the user is facing in order to provide the feedback. The mining mode may terminate automatically upon acquiring the virtual resources, and the wearable device may revert, automatically, back to finder mode.

Referring to FIG. 5, the virtual resource management system 500 can include a signal detector 510 (e.g., the wearable device) to detect UIDs 512 of various network devices, as discussed above with respect to FIG. 4. The UIDs 512 may be logged in a UID Log 530 by a UID module 520. As shown in FIG. 5, various UIDs (i.e., UID XYZ 532, UID EFG 534, UID NOP 536, UID TUV 538, and UID DK 539), each associated with a respective network device, have been logged in the UID log 530. The UIDs 512 may be communicated to an association engine 540, which can perform lookups in an association table 560 to determine whether a respective UID is associated with a respective virtual resource. Thus, in the example shown in FIG. 5, UID XYZ 532 is associated with the virtual resource, virtual wood 562, UID EFG 534 is associated with virtual ore 564, UID NOP 536 is associated with virtual food items 566, and UID TUV 538 is associated with virtual workers 568.

Virtual resources can further include various forms of digital currency or money, or images, video clips, 3-dimensional (3D) interactive models, sound clips, a minigame, or other content that may be presented to the user in response to finding the virtual resource. Thus, the association table 560 can be provided with any type of virtual resource described herein. Furthermore, virtual resources may be stored for subsequent task-oriented activities (e.g., virtual gold or virtual money to be expended in a subsequent game), or may require immediate consumption (e.g., a "mined" video clip discovered by the user which can be viewed once and deleted).

In accordance with examples described herein, when the signal detector 510 detects say, an access point corresponding to UID NOP 536, UID NOP 536 may be communicated to the association engine 540, which, upon performing a lookup in the association table 560, can determine that UID NOP 536 is associated with virtual food items 566. The association engine 540 can communicate this association 542 to a resource engine 550, which can log allocations 552 of virtual resources in a virtual resource log 565, as shown in FIG. 5. Thus, once the association engine 540 communicates the association of UID NOP 536 with virtual food items 566 to the resource engine 550, the resource engine 550 can allocate a predetermined amount of virtual food 566 in the virtual resource log 565. As shown in the virtual resource log 565 of FIG. 5, the user has accumulated 1777 units of virtual food 566, in addition to 765 units of virtual wood 562, 925 units of virtual ore 564, and 25 units of virtual workers 568. However, the user has yet to accumulate any virtual crystal, virtual coal, or virtual oil, which may be essential items in connection with the rendered gameplay 592 on the display 590.

Continuing with examples provided herein, as the user comes within range of say, a remote access point associated with UID IJK 539, the signal detector 510 can communicate UID IJK 539 to the association engine 540, which upon performing a lookup in the association table 560, can determine that UID IJK 539 is not yet associated with a virtual resource (i.e., unknown 569). In accordance with examples described herein, the association engine 540 may attribute UID IJK 539 with a particular virtual resource (e.g., virtual oil) by referencing a local virtual resource catalog, similar to the virtual resource catalog 574 of the host server 570. Alternatively, the association engine 540 may compile an association call 544, which can be communicated to the host server 570 upon establishing a network connection via network 580. The host server 570 can sequentially or randomly select a virtual resource (e.g., virtual oil), from the virtual resource catalog 574, and log the new association between UID IJK 539 and virtual oil in a universal associations list 572. The host server 570 can respond to association calls 544 with new associations 596 logged in the universal associations list 572. The new associations 596 can be communicated to the virtual resource management system 500 over the network 580 via a communication interface 595 of the virtual resource management system 500. Accordingly, the new associations 596 can be received by the association engine 540, which can log the new associations 596 in the association table 560.

Thus, in the example described, the association between UID IJK 539 and virtual oil may be transmitted through the communication interface 595 of the virtual resource management system 500 to the association engine...
which can update the association table 560 to replace "unknown" with virtual oil.

Furthermore, the resource engine 550 can manage the resource log 565 based on both user interactions 594 performed via the rendered gameplay 592 on the display 590, and detected UIDs 512 and received associations 542 based on such UIDs 512. In various implementations, the user interactions 594 in connection with the rendered gameplay 592 can cause virtual resources stored in the virtual resource log 565 to be expended. Such expenditures 567 in the rendered gameplay 592 can correlate to a depletion of the relevant virtual resource in the virtual resource log 565. Various related examples are contemplated. For example, the user interactions 594 can cause respective virtual resources to be traded, consumed, accumulated, invested, or expended in accordance with the rendered gameplay 592.

In the provided example above, the detection of UID UK 539—now associated with virtual oil—can cause the resource engine 550 to attribute a predetermined amount of virtual oil (e.g., five units) in the virtual resource log 565. Furthermore, as shown in FIG. 5, user interactions 594 in connection with the rendered gameplay 592 can cause dynamic updates, by the resource engine 550, to the virtual resource log 565. The virtual resource log 565 can be dynamic in nature, in connection with the rendered gameplay 592 and the detection of UIDs 512. The resource engine 550 can continuously perform allocations 552 and expenditures 567 of virtual resources in the virtual resource log 565. Additionally or alternatively, the resource engine 550 can communicate accumulated virtual resources 597 to the host server 570, which may attribute the accumulated virtual resources 597 to the user’s account 576 to save progress data corresponding to the rendered gameplay 592.

The rendered gameplay 592 may be application or software-based utilizing any number of resources of the user’s computing device. The rendered gameplay 592 may be provided in connection with augmented reality, virtual reality, or a virtually generated gaming environment 578 provided by the host server 570. For augmented reality implementations, the gaming environment 578 provided can comprise virtual features rendered in a real world environment. For example, a camera included on the user’s computing device may be utilized to capture real-world images or video, and the environment 578 may be rendered thereon. Furthermore, the rendered gameplay 592 may be incorporated in conjunction with the use of a remotely operated self-propelled device. In such implementations, the rendered gameplay may include virtual controls to remotely control the self-propelled device. Accordingly, the virtual resource management system 500 can operate in connection with the self-propelled device, which can incorporate one or more features of the virtual resource management system 500, such as the signal detection features (i.e., signal detector 510, UID module 520, and UID log 530).

Finder and Mining Mode Methodology

FIG. 6 is a flow chart describing an example method of acquiring virtual resources using a wearable device. In the below discussion of FIG. 6, reference may be made to like reference characters representing various features of FIG. 4 for illustrative purposes. Furthermore, the method described in connection with FIG. 6 may be performed by, for example, the wearable device 400 as illustrated in FIG. 4, or a wearable device 400 in communication with a mobile computing device 495 running a designated application, as shown in FIG. 4. Referring to FIG. 6, the wearable device 400 can initiate a finder mode and initially detect a signal, such as an advertising beacon, from a network device, such as an access point (600). The beacon can include the network device’s MAC address, or other UID representing the network device.

The wearable device 400 can then perform a lookup in an association table 434 to determine whether the UID is associated with a given virtual resource (605). In some aspects, the wearable device 400 can transmit the UID to the mobile computing device 495, which can make the determination accordingly. At decision block (610), the wearable device 400 (or mobile computing device 495) identifies from the association table 434 whether the UID is associated. If the UID is not associated with a virtual resource (614), then the wearable device 400 can continue to monitor and detect signals from network devices (600), and/or generate a request for a UID association and retrieve the virtual resource association. However, if the UID is associated with a given virtual resource (612), then the wearable device 400, or user utilizing the wearable device 400, can be allocated a predetermined amount of the given virtual resource in the virtual resource log 436 in the following manner.

Upon determining that the network device is associated with a virtual resource, the wearable device 400 can generate feedback to the user indicating the proximate virtual resource (615). The wearable device 400 may then receive a user input (e.g., on an input mechanism or a mode selector on the wearable device) to search for the virtual resource, thereby initiating the mining mode (420). In the mining mode, the wearable device 400 can determine a direction towards the network signal (e.g., the advertising beacon having the UID associated with the virtual resource) (625). Then the wearable device 400 can monitor the sensor data, from one or more sensors of the wearable device 400, to determine whether the user points the wearable device towards or away from the virtual resource signal (630). While the user points the wearable device 400 in differing directions, the wearable device 400 can adjust the feedback dynamically (635).

For example, the feedback, which can comprise a combination of haptic, audio, and visual feedback, can increase in intensity as the user points the wearable device 400 towards the network device, and decrease in intensity as the user points the wearable device 400 away from the network device. Accordingly, once the user has pointed the wearable device 400 in the correct direction for a predetermined time period (e.g., three seconds), the wearable device 400 can acquire a certain allocation of the virtual resource, and store or otherwise enable the user to consume or expend the acquired resource (640).

FIGS. 7A and 7B are low level flow charts describing example processes for managing virtual resources in connection with signal detection and gameplay. In the below discussion of FIGS. 7A and 7B, reference may be made to like reference characters representing various features of FIG. 5 for illustrative purposes. Furthermore, the low level method described in connection with FIGS. 7A and 7B may be performed by, for example, the virtual resource management system 500 as illustrated in FIG. 5. Referring to FIG. 5, the virtual resource management system 500 can detect advertising signals (e.g., beacons) from network devices (700). Such advertising signals may include UIDs, such as
MAC addresses for the detected network devices. The virtual resource management system 500 can then record the UIDs of the network devices in the UID log 530 (705).

[0106] The virtual resource management system 500 may perform a lookup in the association table 560 (710) to determine whether a respective UID is associated with a respective virtual resource (715). If the respective UID is associated with a respective virtual resource (717), then the virtual resource management system 500 can allocate a predetermined amount of the respective virtual resource in the virtual resource log 565 (720). Thereafter, the process may begin again with the detection of advertising signals from network devices (700). However, if the respective UID is not associated with a respective virtual resource (719), the virtual resource management system 500 can generate a request, or an association call 544, to create a new association 596 for the respective UID (725).

[0107] The virtual resource management system 500 can transmit the request to the host server 570 when a network connection is established (730). Alternatively, the virtual resource management system 500 can locally select a respective virtual resource from a local virtual resource catalog (740). Such a selection may be made sequentially (743), for example, if the virtual resource catalog is a sequential list of virtual resources. Or, the selection may be made randomly (741) from the virtual resource catalog by way of a random selection technique.

[0108] In various implementations, the virtual resource management system 500 can receive, remotely or locally, associated virtual resources for unassociated UIDs (745). Thus, the respective unassociated UID may be associated with a respective virtual resource, and thereafter the virtual resource management system 500 can allocate a predetermined amount of the respective virtual resource in the virtual resource log 565 (720). After all detected UIDs are associated with their respective virtual resources, the virtual resource management system 500 can flush the UID log 747, since storing the UIDs may no longer be necessary.

[0109] The process as described in connection with FIG. 7B may be performed before, after, or in conjunction with the process as described with respect to FIG. 7A. Referring to FIG. 7B, the virtual resource management system 500 may receive a user input (e.g., via a touch input on a touch-sensitive display) to launch a gaming application associated with trading, consuming, accumulating, earning, investing, or otherwise expending virtual resources. Based on the user input, the virtual resource management system 500 can initiate the gameplay application to place the wearable device in game mode (750). In doing so, the gaming environment may be rendered on the display 590 (755) (e.g., on the display of the mobile computing device).

[0110] During gameplay, the virtual resource management system 500 can receive various user interactions 594 in connection with the gameplay (760). Such user interactions 594 may be performed by way of touch inputs, mouse interactions, keyboard interactions, interactions using a game controller such as a joystick or a specialized controller device, or a combination of the above. Furthermore, such user interactions 594 may be performed in connection with remote operation of a remotely operated device. In accordance with examples described herein, the gameplay can incorporate a virtual or augmented reality environment in which the user can utilize collected virtual resources. For example, collected virtual resources may be consumed by a virtual character under operative control of the user. Additionally or alternatively, selected virtual resources can be expended to build a virtual building or town. Collected virtual food items may be utilized to feed a virtual colony of gameplay characters. Virtual oil may be utilized by the user during gameplay to modernize a primitive society. Virtual workers may be employed for production or to build infrastructure. Various alternatives and additions in connection with task-oriented operations and gameplay are contemplated.

[0111] Accordingly, based on the user interactions 594, the resource engine 550 of the virtual resource management system 500 can dynamically modify or update the virtual resource log 565 to enable the user to expend, trade, accumulate, invest, earn, etc. virtual resources from the virtual resource log 565 (765).

[0112] When the user wishes to end a gameplay session, the virtual resource management system 500 may receive a user input to deactivate the gaming application, and thus terminate the gaming mode (770). In order to save gameplay progress, the resource engine 550 of the virtual resource management system 500 can compile the virtual resources left in the virtual resource log 565, and transmit a saved list of virtual resources to the host server 570 (775). Thereafter, the virtual resource management system 500 may flush the virtual resource log (777), since the progress is remotely saved.

[0113] Use Case Scenarios

[0114] FIGS. 8A through 8C illustrate unique identifier logs and association tables utilized in connection with virtual resource association, acquisition, and allocation. Various implementations of the disclosed systems and method described herein are contemplated. For example, the detected MAC address of various network devices may be associated with any number of plausible items. Such items may be individuals associated with the network device (e.g., an owner of the network device), or such items may be associated with persons of interest (e.g., characters in connection with a gaming environment), as illustrated in FIG. 8A. Alternatively, such items may be locations of the respective network devices, such that a user is enabled to identify where he/she has traveled, as illustrated in FIG. 8B. Additionally or alternatively, such items may be associated with a time stamp, thereby enabling a user to determine that an interaction has taken place with a particular network device and a time in which the interaction took place, as illustrated in FIG. 8C. Furthermore, the examples provided with respect to FIGS. 8A through 8C may be implemented using the wearable device 400 as described with respect to FIG. 4, and/or using a modified virtual resource management system 500 as discussed with respect to FIG. 5.

[0115] Referring to FIG. 8A, a UID Log 810 of a signal detection system (e.g., a wearable device or wrist-worn device 800 operating in finder mode) can compile UIDs (e.g., UID XYZ 812, UID EFG 814, UID NOP 816, UID TUV 818, UID IJK 819, etc.) associated with respective network devices. The respective network devices may be any one of an access point, a mobile computing device, a tablet computer, a personal computer, a BLUEETOOTH-enabled device, a radio-frequency identification device, a local area network enabled device, etc. Furthermore, a user may carry the wrist-worn device 800, which can come within wireless detection range of any number of the foregoing network devices. In accordance with examples
described herein, the wrist-worn device 800 can compile UIDs corresponding to the network devices in a UID Log 810.

[0116] In many examples, the wrist-worn device 800 only compiles UIDs associated with network devices in which the user comes within wireless range. In such examples, the user may enable a function on a computing device (e.g., a tablet computer, smart phone, PC, etc.), which can pull the UIDs from the wrist-worn device 800 and make associations according to a local association table. In order to transmit the UIDs to the user’s computing device, the wrist-worn device 800 can include connectivity functions to establish a connection with the computing device (e.g., Wi-Fi, BLUETOOTH, etc.). Alternatively, the wrist-worn device 800 can include inductive data communication capabilities in order to transmit the UIDs to the computing device over an inductive link.

[0117] Alternatively, the wrist-worn device 800 can include a local association table 820 in order to make a given association when the wrist-worn device 800 comes within wireless range of a given network device. Upon each detection event, the association table 820 of the wrist-worn device 800 can log an association. For example, as the user carries the wrist-worn device 800 within wireless range of an access point, the wrist-worn device 800 can receive the access point’s advertising beacon, which can include the access point’s unique identifier, UID XYZ 812. The wrist-worn device 800 association table 820 can be referenced to identify that UID XYZ is associated with Ricky H. 822—who may be an owner of the access point. The user may then journey within range of a wireless device with UID EFG 814, which the association table 820 may associate with Jose C. 824—an owner of the wireless device. In such implementations, after an excursion in which the user encounters any number of network devices, the user may thereafter review the association table/log to identify the individuals (i.e., Ricky H., Jose C., etc.) whose devices the user encountered during the excursion.

[0118] Alternatively, for task-oriented implementations (e.g., gaming), the association table may associate such UIDs with characters or places in a gaming environment. As contemplated in one or more examples, each detection event can correspond to a meeting or visitation in which the user meets with a character or visits a point of interest. A task-oriented application may require the user to meet a certain character or visit a certain place before a next achievement is reached. Accordingly, along a physical excursion by the user, the wrist-worn device 800 may detect advertising beacons for respective devices associated with UID NOP 816, UID TUV 818, and UID 819. The association table 820 may be referenced to identify characters and/or places (real or virtual) with which the respective network devices are associated. As an example, the association table/log 820 can identify Mark M. 826 as being associated with UID NOP 816, Dave. S. 828 as being associated with UID TUV 818, and the Coliseum 829 as being associated with UID IJK 819. Upon transmitting such characters and/or places to the user’s computing device, the task-oriented application can input such meetings and visits and record a number of respective achievements.

[0119] FIG. 8B illustrates another usage scenario in which the UIDs of network devices may be associated with a real-world or mock-world environment. As an example, the user may carry the signal detection device 800 to within wireless range of an access point of a coffee shop, where the access point has a unique identifier UID XYZ 832. The wrist-worn device 800 can log UID XYZ in the UID Log 830 and reference the association table/log 840 to determine that UID XYZ 832 is associated with the coffee shop 842. After an excursion of passing through any number of wireless beacons, the user can map the excursion using the physical locations, logged in the association table/log 840, of the network devices along the way.

[0120] In some examples, the physical locations of the network devices may be determined via the detected beacon, which may include location information. Alternatively, the wrist-worn device 800, or mobile computing device to which the wrist-worn device 800 is connected, can include location-based functionality (e.g., GPS resources) to log the location in the association table/log 840. Accordingly, in response to a detection event, the wrist-worn device 800 can be triggered to pinpoint the physical location of the detection event—which can be logged along with the UID of the network device, as shown in FIG. 8B.

[0121] In other examples, the location associated with a UID may be based on a mock-world environment corresponding to, for example, a gaming environment. For example, a user may interact with a task-oriented application on a computing device, which may require the user to visit El Dorado 844. The user may then physically search for or journey to a specified network device having UID EFG 834, which, in accordance with the association table, is associated with El Dorado 844. In such examples, the user may be enabled to reference the association table 840, which may provide a physical location in the real-world of the network device associated with El Dorado 844. Thus, upon wirelessly contacting the network device having UID EFG 834, the association table/log 840 can log the detection event, and the user can accomplish a next achievement in the task-oriented application.

[0122] Additionally, in combination with the above examples with respect to FIGS. 8A and 8B, the wrist-worn device 800 may also include a timer or clock to enter such a timestamp as triggered by a detection event, as illustrated in FIG. 8C. Thus, a user may further review a time in which the wrist-worn device 800 detected the wireless signal of a respective wireless device. For example, during a given excursion, the user may come within wireless range of a network device having UID XYZ 852, which can be logged as being associated with Item 1 862. The detection event of UID XYZ 852 can further trigger a clock or timer to log a timestamp associated with the detection event. The time may reflect a local or universal time, or may reflect an elapsed time from say, the beginning of the excursion or a start time of the task-oriented application.

[0123] Default and Sharing Mode

[0124] FIG. 9 illustrates a wearable device pairing triggering virtual resource data sharing in connection with gameplay, as described herein. The wearable device 900 may be carried or worn by a user to detect advertising beacons or wireless signals from various network devices. The wearable device 900 (e.g., as wrist-worn device) can include a UID Log 901 and automatically log each UID (e.g., MAC address) corresponding to each wireless device detected. According to many examples, upon linking with a computing device 905, wearable device 900 can transmit the UID’s 904 from the UID Log 901 to the computing device 905. The link may be any data connection. For example, the
wearable device 900 can include functionality corresponding to Wi-Fi, radio-frequency, infrared, BLUETOOTH, near-field communication (NFC), and the like. Thus, the UID’s 904 may be transmitted to the computing device 905 over such a communication link.

[0125] In accordance with examples discussed with respect to FIGS. 8A through 8C, the wearable device 900 can collect UIDs corresponding to various network devices, which may be associated with any number of items. For example, an association table of either the computing device 905 or the wearable device 900 can include associations between UIDs of network devices and the registered owners of those network devices. Thus, upon detection of such network devices, the registered owner may be logged in an association log 903. In such examples, a user may scroll through the association log 903 to identify individuals with which the user came into wireless contact.

[0126] In variations, the association table of either the wearable device 900 or the computing device 905 can associate various network devices with real-world or mock-world characters, landmarks, or other places. Such associations may be made in connection with a task-oriented application 902, such as a gaming application providing a gaming environment.

[0127] In some examples, the wearable device 900 can itself include an association table/log and/or a virtual resource log to perform and log associations and collect virtual resources. In such examples, logged association items 906 and/or collected virtual resources 908 may also be transmitted to the computing device over the communication link. Furthermore, timestamps 909 correlated to the detection events and logged associated items 906 may also be transmitted to the computing device 905.

[0128] In similar implementations, the computing device 905 may run a task-oriented application 902, which can trigger the communication link with the wearable device 900. Execution of the task-oriented application 902 can correspond to running a game providing a gameplay environment which utilizes items associated with the UID’s 904 of network devices. The task-oriented application 902 can cause the computing device 905 to receive the UID’s 904 from the wearable device 900 and reference an association log 903 to determine whether a given UID is associated with a given association item. If the given UID is not associated, the computing device 905 can create an association or retrieve an association from a host server. Given an association, the computing device 905 can log a specified amount of collected virtual resources in local virtual resource log 907 for use in the task-oriented application 902.

[0129] In certain aspects, the wearable device 900 may operate in default mode in which the wearable device 900 monitors for wireless signals indicating that another wearable device, (e.g., wearable device 930) is nearby. In the default mode, the wearable devices 900, 930 can each detect wireless signals from each other (e.g., beacon signals) and can generate feedback (e.g., a haptic and audio response) indicating such. The users can visually locate each other and have the option of pairing their wearable devices 900, 930. As discussed herein, the default mode may be executing as a background mode while the wearable device 900 operates in another mode, such as funder mode.

[0130] When the devices 900, 930 are paired, for example, via an inductive link 925, the wearable devices 900, 930 can each initiate a sharing mode that enables the users to view and/or share acquired virtual resources. The computing devices 905, 910 of the users can each run a task-oriented application, or gaming application 912 that enable the users to share information regarding the virtual resources collected, and trade virtual resources accordingly. In the example shown, the user of the wearable device 930 can receive virtual resource data 927 from the wearable device 900 (or the mobile computing device 905), and the user’s mobile computing device 910 can acquire “unlocked” virtual resources 929 based on a trade with the wearable device 900.

[0131] According to one or more examples, the user may then run the gaming application 912 on the computing device 910 and interact with the computing device 910 via the gaming application 912, which can provide a gaming environment 920 that requires the use directly or indirectly of an association table 914 in connection with wearable device 930 and the gaming application 912. For example, the gaming environment 920 may require user interactions with real-world or mock-world characters and/or places, which may be accomplished by coming within wireless range of specified network devices associated with such real-world or mock-world characters and/or places in an association table 914 of the computing device 910. Upon detecting such network devices, the computing device 910 can log associated items (e.g., real-world characters, mock-world characters, real-world locations, mock-world locations, etc.). In accordance with the gaming environment 920, upon logging such associations, various tasks may be achieved.

[0132] In similar examples, the gaming environment 920 may require the collection and use of virtual resources, as described above. Thus, the user may run the signal detector/ gaming application 912 and interact with the gaming environment 920, which may require the user to collect a number and amount of virtual resources. In the sharing mode, the wearable device 930 enables the user to acquire unlocked virtual resources 929 from another user, which can be compiled in a virtual resource log 918 associated with the gaming application 912. Accordingly, the user can perform a physical excursion to enable the computing device 910 to wirelessly interact or otherwise detect various network devices to collect such virtual resources for utilization in the gaming environment 920. The gaming environment 920 can provide the virtual resource log 918 which can inform the user of which virtual resources and how many of each virtual resource the user has collected.

[0133] Interactive or Battle Mode

[0134] FIGS. 10A and 10B illustrate a wearable device pairing triggering an interactive mode, or battle mode, between a pair of proximate users 1002, 1012 as shown in FIG. 10B. The interactive mode can be triggered upon selection of the mode on each of a pair of wearable devices 1010, 1030. Additionally or alternatively, an inductive link 1025 between the devices 1010, 1030 can enable the users to select the interactive mode, as shown in FIG. 10A. For example, the devices 1010 may each be connected to a respective mobile computing device, such as mobile computing device 1050 and mobile computing device 1060 shown in FIG. 10B. Upon performing the inductive link 1025, the users 1002, 1012 can each select the interactive mode to perform a series of actions and alternate between offensive and defensive sub-modes in the interactive mode.
Referring to FIG. 10B, users 1002 and 1012 operate their wearable devices 1010, 1030 in interactive mode, which transmits action data 1048 between the devices 1010, 1030. The action data 1048 can correspond to user gestures 1041 performed by the users 1002, 1012 while the devices 1010, 1030 are in interactive mode. In many examples, one device may be in an offensive sub-mode in which the user 1002 can perform a number of offensive actions 1042, such as attack actions, using the wearable device 1010. Action data 1048 corresponding to the offensive action 1042 can be generated by the wearable device 1010, and transmitted to the wearable device 1030. The wearable device 1010 can generate feedback to the user indicating whether or not the offensive action 1042 was successful.

At the same time, the proximate user 1012 can perform user gestures 1041 corresponding to a defensive action 1044 based on the offensive action 1042 performed by the user 1002. Whether the defensive action 1044 is effective or ineffective against the offensive action 1042 (e.g., a blocking action), the wearable device 1030 can generate feedback reflecting such. The wearable devices 1010, 1012 can alternate between offensive and defensive sub-modes giving each user successive opportunities to perform offensive 1042 and defensive actions 1044. Furthermore, sensor patterns can be preconfigured corresponding to any number of offensive actions 1042 and defensive actions 1044. Still further, only certain sensor patterns for defensive actions 1044 (e.g., a specified type of blocking action) may be effective against a given offensive action 1042. Each sensor pattern for each offensive 1042 or defensive action 1044 can be preconfigured in accordance with a set of rules (e.g., gaming rules) that utilize the wearable devices 1010, 1030 in connection with such physical gameplay. Thus, the users 1002, 1012 may previously “train” to perfect the defensive actions 1044 in light of the offensive actions 1042 and vice versa in a training mode described herein.

During an interactive or battle session, the mobile computing devices 1050, 1060 connected to each wearable device 1010, 1030 can maintain a tally or score based on any number of metrics in light of the offensive and defensive interactions between the users 1002, 1012. Successful or partially successful offensive 1042 or defensive actions 1044 may be given respective scores, and unsuccessful actions 1042, 1044 may be given other scores, or negative scores. A predetermined threshold score 1052 may be set by the interactive session, for example, as agreed upon by the users 1002, 1012, or in accordance with a particular game selected in the interactive mode. Once reached, the interactive session may end and a winner may be declared. A final result can be displayed on the respective mobile computing devices 1050, 1060, and the user’s 1002 may then choose to initiate another interactive session or conclude the experience.

FIG. 11 is a flow chart describing an example method of implementing an interactive mode between a pair of user utilizing a corresponding pair of wearable devices. In the example described with respect to FIG. 11, reference may be made to various features shown and described in FIG. 10A and FIG. 10B for illustrative purposes. Referring to FIG. 11, the wearable device 1010 can detect an inductive pairing 1025 with a proximate wearable device 1030 (1100). In response to a user selection, the wearable device 1010 can initiate an interactive mode (1105). Initially, the wearable device 1010 can determine an ordering or sequence between offensive and defensive sub-modes (1110), and then monitor sensor data for user gestures 1041 performed depending on a present sub-mode.

In the offensive sub-mode (1175), the wearable device 1010 can identify user gestures 1041 indicating an offensive attack 1042 (1120). The wearable device 1010 can do so by active monitoring or via state machine monitoring, as described herein. Sensor patterns corresponding to each of any number of offensive attack actions can be detected, and cause the wearable device 1010 to generate and transmit action data 1048 indicating the particular offensive action 1042 performed by the user 1002 to the proximate device 1030 (1125). The wearable device 1010 may thereafter receive data from the proximate wearable device 1030 indicating whether or not the offensive action was successful (1130) and generate a feedback output accordingly (1135). If the offensive action 1042 was successful, then the wearable device 1010 can generate positive feedback (e.g., audio, visual, etc.) (1137). If the offensive action 1042 was unsuccessful or a failure, then the wearable device 1010 can generate negative feedback (1139). Upon generating the feedback, the wearable device 1010 can switch to the defensive sub-mode (1180).

In the defensive sub-mode (1180), the wearable device 1010 can receive data indicating an offensive attack action 1042 from the proximate wearable device 1030 (1150). The wearable device 1010 can monitor sensor data for sensor patterns corresponding to user gestures 1041 indicating a defensive action 1044, or blocking move, performed by the user 1002 (1155). The wearable device 1010 can further determine, using timing and sensor data, whether the defensive action 1044 was successful (1160). For example, the user 1002 may be given a predetermined amount of time to perform a defensive action 1042 in response to an offensive action 1042 once the action data 1048 (indicating the offensive action 1042) is received. Accordingly, reception of the action data 1048 can trigger a timer on the wearable device 1010 giving the user limited time to perform an appropriate defensive action 1044.

After transmitting action data 1048 corresponding to the defensive action 1044, the wearable device 1010 can generate a feedback output (1165). If the defensive action 1044 is successful and performed within the time threshold, the wearable device 1010 can generate positive feedback (1167). However, if the defensive action 1044 was unsuccessful, the wearable device 1010 can generate negative feedback (1169), and then transmit successful/unsuccessful data to the proximate device 1030 (1170), and tally the score for the interactive session.

During the interactive session, the wearable device 1010 can determine whether the threshold score has been achieved by either the user 1002 or the proximate user 1012 (1185). If not (1187), then the users 1002, 1012 can continue in the interactive mode, alternating between offensive (1175) and defensive (1180) sub-modes. However, if the threshold score has been achieved by one of the users 1002, 1012 (1189), then the wearable device 1010 can exit the interactive mode and transmit the final scores to the connected mobile computing device 1050 (1195).

Use of multiple wearable devices 1010, 1030 may be further expanded into various proximity-based activities. Accordingly, in some aspects of interactive mode, multiple users can join and engage in an interactive session that utilizes proximity sensing resources on the wearable
devices. Proximity sensors can include components of an inertial measurement unit (IMU) incorporated into each wearable device 1010, 1030 (e.g., an infrared sensor). Additionally or alternatively, GPS resources on the wearable devices 1010, 1030 themselves or on connected mobile computing devices 1050, 1060 can be utilized for proximity-based activities. Such activities can include sporting activities to determine, for example, finish line positions of a running race or physical game activities for children (e.g., hide-and-seek or capture the flag). Feedback can be provided to each user in the interactive session, providing information connected to the proximity-based activity (e.g., information indicating a team-leader, a ranking, a team score, positional information, and the like).

[0144] Training Mode

[0145] FIG. 12 is a flow chart describing an example method of initiating a training mode on a wearable device in connection with a self-propelled device. In the below discussion of FIG. 12, reference may be made to like reference characters representing various features described with respect to FIG. 1A for illustrative purposes. Referring to FIG. 12, the wearable device 102 can detect a user input placing the wearable device 102 in training mode (1200). The user input can be detected via a mode selector on the wearable device 102 (1202), or via launch of a designated application on a connected mobile computing device (1204). Alternatively, the training mode can be initiated via a combination of user inputs on the mode selector and an inductive link with a remotely operated device 140. Accordingly, the wearable device 102 may also detect an inductive pairing with a remotely operated device 140 (1205). In response to the inductive pairing, the wearable device 102 can transmit data to initiate the training mode on the remotely operated device 140 as well (1210). The transmitted data can cause the remotely operated device 140 to execute instructions to aid the user in training for a series of actions. For example, the series of actions can correspond to offensive and defensive actions that the user can implement when the wearable device 102 is in interactive or battle mode. Additionally or alternatively, the series of actions can get progressively more difficult as the user successively accomplishes each action.

[0146] Initially, the wearable device 102 can synchronize directionally with the remotely operated device 140 (1218). In some aspects, the user can manually synchronize the gyroscopic sensors of the devices 102, 140 by physically pointing a forward operational direction of the remotely operated device 140 away from the wearable device 102 and providing a calibration input (e.g., an input on the mode selector of the wearable device 102) (1217). In other aspects, the gyro synchronization may be performed automatically (1219).

[0147] Automatic synchronization can be initiated by the wearable device 102 by generating and transmitting a spin command to the remotely operated device 140, which can execute a spin accordingly (1220). Using a signal detector, the wearable device 102 detect an asymmetry in the radiation pattern of the remotely operated device 140 as it spins, indicating the direction towards the remotely operated device 140 (1225). With a known direction, the wearable device 102 can transmit a direction calibration command to the remotely operated device 140 indicating the direction, which the remotely operated device 140 can process to align its internal drive system accordingly (1230).

[0148] In many aspects, the wearable device 102 can track the location of the remotely operated device 140 as it traverses and maneuvers (1240). The remotely operated device 140 can include surface features or an accessory (e.g., a magnetically coupled attachment) that indicates a forward “looking” direction of the remotely operated device 140. In certain examples, the user is instructed to walk or run around in a circle until the user is directly facing the forward looking direction of the remotely operated device 140. In many aspects, the wearable device 102 can include sensors to determine an orientation of the user. For example, the wearable device 102 can determine whether the user is facing an instructed direction in connection with the training mode, such as facing the remotely operated device 140. Additionally or alternatively, the wearable device 102 can generate an output, via the feedback mechanism, instructing the user to perform a set of actions (1245). The output may be in the form of audio instructions, and can be based on data received from the remotely operated device 140 (1247), or from the wearable device 102 utilizing a local routine set (1249), which may be randomized or sequenced in accordance with the executing training mode instructions.

[0149] Once the instructions are outputted to the user, the wearable device 102 can initiate a timer (1250). The timer can be initiated for each instruction outputted to the user, and a threshold time limit can be set for each instruction. Thus, the user can be instructed to perform the set of actions within the predetermined time period. The wearable device 102 can monitor the sensor data to determine whether the user successfully performs the set of actions (1255). Specifically, for each instruction output, the wearable device 102 can determine whether the user has performed the instructed set of actions within the established threshold time limit (1260). If so (1261), then the wearable device 102 can generate another output instructing the user to perform another set of actions (1245). However, if the user fails to perform the set of actions (1263), then the wearable device 102 can terminate the training session and generate a fail score (1265), which may be displayed on the user’s mobile computing device.

[0150] Sword or Wield Mode

[0151] FIG. 13 is a flow chart describing an example method of implementing a wearable device in a sword mode. In the below description of FIG. 13, reference may be made to like reference characters representing various features of FIG. 1A and FIG. 1B. Referring to FIG. 13, the wearable device 102 can detect a user input placing the wearable device 102 in a sword mode (1300). When in the sword mode, the wearable device 102 can initially monitor sensor data 137 to determine whether the user has grabbed or grasped an object (1305). In some implementations, a state machine in the memory 180 to the wearable device 102 specific to the grabbing/grasping action can provide a state machine report to the output generator 160 indicating that the user is holding an object.

[0152] In many examples, the wearable device 102 can determine a series of actions performed by the user with the object from the sensor data 137 (1310). Such actions can be actively determined by the output generator 160 in the sword mode, or via state machine reports from respective state machines that indicate each specific action. In any case, in response to each action, the wearable device 102 can generate a feedback response or output 132 using a feedback mechanism (1315). The feedback output 132 for each action
can be distinct audio responses 196 (1317), haptic responses 192 (1316), and/or visual responses 194 (1318).

[0153] In many examples, based on the action performed by the user, the feedback mechanism can generate a corresponding output, such as sword fighting sounds, or other weapon-like or wand-like sounds based on the executing sword mode instructions. Thus, the feedback output 132 based on the sensor patterns detected when the user is grasping an object may be any such sounds corresponding to the actions performed by the user. In this sense, the “sword mode” may rather be a “wand mode,” a “magic mode,” a “conductor mode,” a “sporting mode” (e.g., for a tennis-like game), and the like. When the user wishes to end the sword mode, the wearable device 102 can detect a user input, such as an input on the mode selector 110 (1320). In response to the input, the wearable device 102 can deactivate or terminate the sword mode (1325).

[0154] Hardware Diagrams

[0155] FIG. 14 is a block diagram that illustrates a computer system upon which examples described may be implemented. For example, one or more components discussed with respect to the remotely operated device of FIG. 1A, and the methods described herein, may be performed by the system 1400 of FIG. 14.

[0156] In one implementation, the computer system 1400 includes processing resources 1410, a main memory 1420, ROM 1430, a storage device 1440, and a communication interface 1450. The computer system 1400 includes at least one processor 1410 for processing information and a main memory 1420, such as a random access memory (RAM) or other dynamic storage device, for storing information and instructions to be executed by the processor 1410. The main memory 1420 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by the processor 1410. The computer system 1400 may also include a read only memory (ROM) 1430 or other static storage device for storing static information and instructions for the processor 1410. A storage device 1440, such as a magnetic disk or optical disk, is provided for storing information and instructions. For example, the storage device 1440 can correspond to a computer-readable medium that stores instructions performing sensor data processing and translation operations as discussed herein.

[0157] The communication interface 1450 can enable computer system 1400 to communicate with a computing device and/or wearable device (e.g., via a cellular or Wi-Fi network) through use of a network link (wireless or wired). Using the network link, the computer system 1400 can communicate with a plurality of devices, such as the wearable device, a mobile computing device, and/or other self-propelled devices. The main memory 1420 of the computer system 1400 can further store the drive instructions 1424, which can be initiated by the processor 1410. Furthermore, the computer system 1400 can receive control commands 1462 from the wearable device and/or mobile computing device. The processor 1410 can execute the drive instructions 1424 to process and/or translate the control commands 1462—corresponding to user gestures performed by the user—and implement the control commands 1452 on the drive system of the self-propelled device.

[0158] Additionally, the main memory 1420 can further include mode instructions 1424, which the processor 1410 can execute to place the self-propelled device in one or multiple modes to interact with the wearable device. In some examples, execution of the mode instructions 1422 can place the self-propelled device in an operational mode that provides feedback 1452 and/or instructions 1454 to the wearable device over the network 1480 (e.g., in training mode).

[0159] Examples described herein are related to the use of computer system 1400 for implementing the techniques described herein. According to one example, those techniques are performed by computer system 1400 in response to processor 1410 executing one or more sequences of one or more instructions contained in main memory 1420. Such instructions may be read into main memory 1420 from another machine-readable medium, such as storage device 1440. Execution of the sequences of instructions contained in main memory 1420 causes processor 1410 to perform the process steps described herein. In alternative implementations, hard-wired circuitry and/or hardware may be used in place of or in combination with software instructions to implement examples described herein. Thus, the examples described are not limited to any specific combination of hardware circuitry and software.

[0160] FIG. 15 is a block diagram that illustrates a mobile computing device upon which examples described herein may be implemented, such as the mobile computing device 495 of FIG. 4. In one example, the computing device 1500 may correspond to, for example, a cellular communication device (e.g., feature phone, smartphone, etc.) that is capable of telephony, messaging, and/or data services. In variations, the computing device 1500 can correspond to, for example, a tablet or wearable computing device.

[0161] In an example of FIG. 15, the computing device 1500 includes a processor 1510, memory resources 1520, a display device 1530 (e.g., such as a touch-sensitive display device), one or more communication sub-systems 1540 (including wireless communication sub-systems), input mechanisms 1550 (e.g., an input mechanism can include or be part of the touch-sensitive display device), and one or more location detection mechanisms (e.g., GPS component) 1560. In one example, at least one of the communication sub-systems 1540 sends and receives cellular data over data channels and voice channels.

[0162] The memory resources 1520 can store a designated control application 1522, as one of multiple applications, to initiate the communication sub-system 1540 to establish one or more wireless communication links with the self-propelled device and/or a wearable device. Execution of the control application 1522 by the processor 1510 may cause a graphical user interface (GUI) 1535 to be generated on the display 1530. Interaction with the GUI 1535 can enable the user to calibrate the forward directional alignment between the self-propelled device and the computing device 1500. Furthermore, the GUI 1535 can allow the user to initiate a task-oriented operation (e.g., a game) to be performed by the user in conjunction with operating the self-propelled device with user gestures using the wearable device, as described herein.

[0163] FIG. 16 is a block diagram of an example portable sensing device upon which examples described herein may be implemented, such as the wearable device 102 of FIG. 1A.

[0164] In an example of FIG. 16, the portable sensing device 1600 includes a processor 1610, memory resources 1620, a feedback mechanism 1630 (e.g., audio 1632, haptic 1633, visual 1631 devices), a communication sub-systems
(e.g., wireless communication sub-systems such as BLUETOOTH low energy), one or more sensors 1660 (e.g., a gyroscope sensor or accelerometer) and an input mechanism 1650 (e.g., an analog or digital mode selector). In one example, the communication sub-system 1640 sends and receives data over one or more channels.

[0165] The memory resources 1620 can store mode instructions 1623 corresponding to a plurality of control modes 1622, as described herein, which can be executed by the processor 1610 to initiate a particular mode. Certain executing mode instructions 1623 can initiate the communication sub-system 1640 to establish one or more wireless communication links with the self-propelled device and/or the mobile computing device. Execution of a control mode 1622 by the processor 1610 may cause output of the portable to generate distinct feedback responses using the feedback mechanism 1630 based on sensor data from the sensor(s) 1660 indicating user gestures performed by the user.

[0166] In some examples, the memory resources 1620 can comprise a number of state machines 1624 which can provide state machine reports 1627 to the processor 1610. Each state machine 1624 can identify certain sensor patterns are identified by respective state machines 1624. Each state machine 1624 may monitor for a single sensor pattern which, if identified by that state machine 1624, can cause the state machine 1624 to transition states, thereby providing a state machine report 1627 to the processor 1610 identifying the user gesture performed. The processor 1610 can translate the state machine reports 1627—which indicate the user gestures—in accordance with an executing set of mode instructions 1623 in order to generate a corresponding output via the feedback mechanism 1630 and/or control commands 1612 to be communicated to the self-propelled device via the communication sub-system 1640.

[0167] While examples of FIG. 14, FIG. 15, and FIG. 16 provide for a computing system 1400, a computing device 1500, and a portable sensing device 1600 for implementing aspects described, in some variations, other devices of the three can be arranged to implement some or all of the functionality described with the processing resources of the remotely operated device 1400 of FIG. 1A, the mobile computing device 495 of FIG. 4, or the wearable device 102 of FIG. 1A, as shown and described throughout.

[0168] With further reference to examples of FIG. 14, FIG. 15, and FIG. 16, some examples include functionality for projecting an orientation and/or perspective of a user onto a gaming environment via a sensing output of a portable sensing device 1600. For example, when the portable sensing device 1600 is worn, the orientation and perspective of the user can be inferred from sensors 1660 (e.g., IMU). and this sensor information can be virtualized for the gaming environment. For example, the gaming environment can show on a computing device (e.g., display screen of a computer, mobile computing device etc.). The gaming environment can have a perspective that is based on the orientation of the user (e.g., user is standing north), as determined by the portable sensing device 1600. The perspective can change as the user changes orientation, moves in a particular direction etc. In some examples, the portable sensing device 1600 can be used to control a virtual or actual object (e.g., self-propelled device or remotely operated device), and the orientation and direction of the controlled object may be with reference to a reference frame of the user.

[0169] In variations, a reference frame of the self-propelled device may be used, and the user’s orientation can be used to influence control of the virtual or actual device in motion. For example, the user’s movement or motion can influence a change of direction. Alternatively, both orientations can be used concurrently. For example, if the device under control is a virtual vehicle that carries the user, the user may turn his head (e.g., when wearing a necklace carrying the portable sensing device 2000) to see a view to a particular side while the orientation of the vehicle is used for the motion of the vehicle.

[0170] Multi-Device Usage

[0171] FIG. 17 illustrates an embodiment of multiple sensing devices that concurrently provide input for a program or application which utilizes the inputs, along with inferences which can be made about a person or object that carries the device. In one example, the combination of sensing devices can include a hat, a ring, eyeglasses or a device which the user can carry in his or her hand (e.g., FOB, mobile computing device). In variations, more than two wearable devices can be employed by one user.

[0172] FIG. 18 illustrates a system which concurrently utilizes input from multiple modular sensing devices in connection with execution of an application or program. With reference to an example of FIG. 18, a multi-device system 1800 includes a first modular sensing device 1810, a second modular sensing device 1820, and a controller 1830. Each of the first and second modular sensing devices 1810, 1820 includes a respective inertial measurement unit (IMU) 1812, 1822, a processor 1814, 1824 and memory 1816, 1826. The IMU 1812, 1822 of each modular sensing device 1810, 1820 can include sensors such as an accelerometer 1815, 1825 and gyroscope sensor 1817, 1827. The first and second modular sensing devices 1810, 1820 may also include additional sensing resources, such as a magnetometer and/or proximity sensor.

[0173] The controller 1830 can include a processor 1832 and a memory 1834. The processor 1832 can execute instructions 1835 for a program or application which can execute and process inputs 1811, 1813 from each of the respective modular sensing devices 1810, 1820. In some variations, the controller 1830 is a mobile computing device, such as a multi-purpose wireless communication device which can wirelessly communicate with each of the first and second modular sensing devices 1810, 1820.

[0174] While an example of FIG. 18 illustrates the controller 1830 as a separate device from the first and second modular sensing devices 1810, 1820, variations provide that the controller 1830 is integrated or otherwise combined with at least one of the first or second modular sensing devices 1810, 1820. For example, the controller 1830 can include a multi-purpose wireless communication device that is equipped with a gyroscope sensor and accelerometer. Thus, for example, variations can provide the second modular
sensing device 1820 to be a local resource of the controller 1830, which communicates with the first modular sensing device 1810.

[0175] With further reference to FIG. 18, the controller 1830 can receive inputs 1811, 1813 from respective first and second modular sensing devices 1810, 1820. The inputs 1811, 1813 can be received in connection with an application 1839 or program that is executed by the processor 1832 of the controller 1830. The processor 1832 can execute the instructions 1845 in order to implement an inference engine 1835 for determining inferences about the person or object with one or both of the modular sensing devices 1810, 1820. For example, the application 1839 can correspond to a game or simulation, and the inference engine 1835 can be specific to the application 1839. Among other applications, the inference engine 1835 can be used to determine when the motions of two modular sensing devices 1810, 1820 are separate and distinct from one another, or continuous and/or part of the same input motion.

[0176] According to one implementation, each input 1811, 1813 can correspond to one or more of a position, height, orientation, velocity, linear and/or rotational acceleration. Each of the first and second sensing devices 1810, 1820 generate a set of measured (or sensed data) corresponding to, for example, a movement (e.g., gesture) made with the respective sensing device 1810, 1820. Additionally, the controller 1830 can process input 1811, 1813 corresponding to each of the respective data sets in order to determine a third data set of inferences. In this way, the inferences reflect information determined from sensed data, rather than directly measured data. The inferences which can be output from the inference engine 1835 can be deterministic or probabilistic, depending on implementation.

[0177] With reference to an example of FIG. 17, user 1701 can wear two modular sensing devices, and the inference engine 1835 can assume some inferences based on anatomical constraints and/or context (e.g., such as provided from execution of the application 1839). For example, each of the first and second modular sensing devices 1810, 1820 can correspond to a wearable wrist device. Alternatively, the second modular sensing device 1820 can correspond to the pendent 1712 or neck-worn device. By way of example, if the first modular sensing device 1810 (wrist device 1710) is detected to be in motion, the inference engine 1835 can be used to determine additional position data for the movement of that device along a third axis based on orientation, position, and motion of second modular sensing device 1820 (wrist device 1711 or pendent device 1712). For example, if the first modular sensing device 1810 (wrist device 1711) measures arc motion, and the second modular sensing 1820 is the pendent, then the orientation of the second modular sensing device can indicate whether, for example, the arc motion is in front of the user or to the user’s side. Alternatively, if the second modular sensing device 1820 is the second wrist device 1712, the information sensed from the second wrist device can identify the corresponding hand or device as being in front of the body. In such an orientation, the inference engine 1835 can determine the inference to be that the user is making the arc of motion in front of his body. Similarly, if the height of the second sensing device 1820 is determined to be belt high and the device is held by the user, the orientation of the user’s torso can be inferred (along with the direction of the arc).

[0178] In examples in which the second modular sensing device 1820 is a pocket device (e.g., mobile computing device, FOB), information can be determined from, for example, the height of the device (e.g., user standing, crouching or jumping) and the rotation of the device. For example, if the second modular sensing device 1820 is pocket word, a change in the orientation of the device from vertical to horizontal, in combination with a downward acceleration can indicate the user is crouching. If the user is crouching, for example, the type of motion that is likely by the first modular sensing device 1810 may be limited (e.g., motion of the wrist device 1710 is likely in front of user when user is moving up or down). The examples described with respect to FIGS. 17 and 18 can enable the user to utilize the modular sensing device(s) in connection with a real-world gameplay environment (or other task-oriented activities) executed by one or more of the modular sensing devices 1810, 1820. control a remotely operated device using gestures sensed by the modular sensing devices 1810, 1820, interact with other users, and perform various tasks in which the modular sensing devices 1810, 1820 can provide feedback and response output.

[0179] Modular Sensing Device Implementations

[0180] FIG. 19 illustrates an example of a modular sensing device insertable into a wrist worn apparatus. In particular, a modular sensing device 1900 can be constructed in accordance with examples provided herein, in order to implement operations and functionality such provided by any of the examples described. The modular sensing device 1900 can include a housing 1910 for containing a processor, memory (e.g., a controller implementing a plurality of state machines), and one or more sensors (e.g., IMU, gyroscope, accelerometer, proximity sensor, magnetometer, etc.). The modular sensing device 1900 can also include a wireless communication resource for communicating with other devices, including devices which may be controlled in movement (e.g., self-propelled device) or other processes.

[0181] In some examples, the modular sensing device 1900 can communicate sensor data, including output from the IMU, to another device for purpose of controlling movement of the other device. In some variations, the modular sensing device 1900 can include processing capabilities to process raw sensor data into higher data forms of communication. For example, the modular sensing device 1900 can generate output in the form of commands, or input for command selection from a receiving device.

[0182] According to some examples, the housing 1910 of the modular sensing device 1900 can include securement features for enabling the modular sensing device 1900 to fasten onto another compatible structure 1920. The compatible structure 1920 can include an opening that is shaped to receive and secure the modular sensing device 1910. As shown with an example of FIG. 19, the securement features can include, for example, structural or shaped features of the housing 1910. For example, the housing 1910 can be dimensioned and/or structured (e.g., housing may be biased) to snap-fit into the compatible structure 1920. Alternatively, at least one of the housing 1910 or compatible structure 1920 can include an integrated and/or mechanical fastener to secure the modular sensing device 1900.

[0183] FIG. 20 illustrates an implementation of the modularized sensing device 2000. As shown, the sensing device 2000 can be retained by the compatible structure 2020 (e.g., wrist-worn strap), and then removed and placed in an
opening of a wielded device 2010 (e.g., play sword). The placement of the modular sensing device 2000 in different compatible structures 2020, 2010 for retention and use can be coordinated with different functionality being enabled through the sensing device. For example, the modular sensing device 2000 in the wrist-worn strap 2020 can be used in conjunction with a first program running on a mobile computing device (controller), self-propelled device and/or other computer system (e.g., virtual gaming system). When placed in the wielded device 2010 (e.g., a wand), the modular sensing device 2000 can be operated in conjunction with a mobile computing device, self-propelled device and/or other computer system (e.g., virtual gaming system) which executes a second program or application. In each context, the orientation of the modular sensing device 2000 can be used to determine a perspective, such as a virtual field of view for gameplay. The perspective can refer to the orientation, direction and/or position of the user, and/or of the user’s body part with respect to the sensing device. With the wand, the orientation and direction of the sensing device can be used to project a corresponding virtual object in a virtual environment (e.g., sword). The modular sensing device 2000 may also be able to read an identifier of the compatible structure 2020, 2010 in order to determine information about the structure, such as its dimension, and whether the structure is word or carried. Based on known information, inferences can be determined for purpose of virtualization, etc. (e.g., length of sword).

[0184] Conclusion

[0185] It is contemplated for examples described herein to extend to individual elements and concepts described herein, independently of other concepts, ideas or system, as well as for examples to include combinations of elements recited anywhere in this application. Although examples are described in detail herein with reference to the accompanying drawings, it is to be understood that this disclosure is not limited to those precise examples. As such, many modifications and variations will be apparent to practitioners skilled in this art. Accordingly, it is intended that the scope of this disclosure be defined by the following claims and their equivalents. Furthermore, it is contemplated that a particular feature described either individually or as part of an example can be combined with other individually described features, or parts of other examples, even if the other features and examples make no mentioned of the particular feature. Thus, the absence of describing combinations should not preclude the inventor from claiming rights to such combinations.

What is claimed is:

1. A wearable device to be worn by a user, the wearable device comprising:
   a mode selector;
   one or more sensors;
   a signal detector to detect wireless signals;
   a feedback mechanism configured to provide feedback to the user, the feedback mechanism comprising at least one of a haptic system, an audio system, or a visual system; and
   a controller configured to:
   in response to a user input on the mode selector, initiate a selected mode from a plurality of modes on the wearable device, the selected mode causing the controller to execute a real-world game; and
   in accordance with the selected mode, interpret sensor data from the one or more sensors and inputs from the signal detector, the sensor data corresponding to a series of actions performed by the user utilizing the wearable device; and
   based on interpreting the sensor data and the detected signals in accordance with the selected mode, generate feedback responses via a feedback mechanism.

2. The wearable device of claim 1, wherein the wearable device comprises a wrist-worn device, and wherein the selected mode comprises a wield mode that causes the controller to:
   detect sensor data corresponding to a series of actions performed by the user while the user is holding a wielded device; and
   generate a series of feedback responses unique to the wield mode in response to the series of actions performed by the user.

3. The wearable device of claim 1, wherein the selected mode comprises a finder mode that causes the controller to:
   using the signal detector, detect a wireless signal from a network device, the wireless signal including a unique identifier for the network device, and determine whether the unique identifier for the network device correlates to any of a plurality of digital resources.

4. The wearable device of claim 3, wherein the finder mode further causes the controller to:
   in response to determining that the unique identifier for the network device does correlate to a respective digital resource, (i) generate a feedback response via the feedback mechanism indicating that the network device comprises a source of the respective digital resource, and (ii) enable the user to select a mining mode on the wearable device to cause the controller to store an amount of the respective digital resource.

5. The wearable device of claim 4, wherein the mining mode causes the controller to:
   generate variable feedback outputs via the feedback mechanism based on a direction towards the network device, based on the wireless signal, to enable the user to locate the respective digital resource.

6. The wearable device of claim 5, wherein the mining mode causes the controller to generate the variable feedback outputs by (i) dynamically increasing an intensity of the feedback mechanism as the user points the wearable device towards the direction of the network device, and (ii) dynamically decreasing an intensity of the feedback mechanism as the user points the wearable device away from the direction of the network device.

7. The wearable device of claim 5, wherein the mining mode further causes the controller to:
   when the user points the wearable device in the direction of the network device for a predetermined amount of time, generate and store a predetermined amount of the respective digital resource in a memory of the wearable device.

8. The wearable device of claim 7, wherein the selected mode enables the user to utilize the respective digital resource in connection with the real-world game.

9. The wearable device of claim 7, wherein the selected mode further causes the controller to:
   wirelessly connect with a computing device of the user; and
transfer the respective digital resource to a memory of the computing device, the respective digital resource being utilizable in connection with gameplay on the computing device.

10. The wearable device of claim 1, wherein the selected mode further causes the controller to:
while executing the real-world game, detect a proximate wearable device using the signal detector;
wirelessly connect with the proximate wearable device via a communication interface of the wearable device; and
in response to wirelessly connecting with the proximate wearable device, execute a sub-mode to cause the controller to:
identify, in the sensor data, a second set of actions performed by the user indicating a blocking move;
translate the second set of actions into offensive data corresponding to the blocking move; and
transmit the defensive data to the proximate device via the wireless communication interface, wherein the defensive data indicates whether the blocking move was successful.

16. The wearable device of claim 15, wherein the blocking sub-mode causes the controller to:
identify, in the sensor data, a second set of actions performed by the user indicating a blocking move;
translate the second set of actions into offensive data corresponding to the blocking move; and
transmit the defensive data to the proximate device via the wireless communication interface, wherein the defensive data indicates whether the blocking move was successful.

17. The wearable device of claim 16, wherein the interactive mode further causes the controller to:
generate a respective score for each successful attacking move and each successful blocking move;
maintain a score count based on the respective scores for each of the successful attacking moves and each of the successful blocking moves; and
when the score count reaches a predetermined threshold, generate a feedback response using the feedback mechanism indicating that the simulated fighting session has ended.

18. The wearable device of claim 17, wherein the interactive mode further causes the controller to:
connect with a mobile computing device; and
transmit a final score count to the mobile computing device for display.

19. The wearable device of claim 1, wherein the controller implements a plurality of state machines, each respective state machine being associated with a sensor data signature corresponding to a specified user action in the series of actions, and wherein the respective state machine executes a state transition when the sensor data correlates with the sensor data signature; the state transition causing the controller to generate an associated feedback response via the feedback mechanism.

20. A computer-implemented method of operating a wearable device, the method being performed by one or more processors and comprising:
in response to a user input on a mode selector of the wearable device, initiating a selected mode from a plurality of modes on the wearable device, the selected mode causing the wearable device to execute a real-world game; and
in accordance with the selected mode, interpreting sensor data from one or more sensors of the wearable device and detected signals from a signal detector of the wearable device, the sensor data corresponding to a series of actions performed by the user utilizing the wearable device; and
based on interpreting the sensor data and the detected signals in accordance with the selected mode, generating feedback responses via a feedback mechanism.

* * * * *