US 20120249461A1

(19) **United States**

(12) **Patent Application Publication**

FLANAGAN et al.

(10) **Pub. No.: US 2012/0249461 A1**

(43) **Pub. Date: Oct. 4, 2012**

(54) **DEDICATED USER INTERFACE CONTROLLER FOR FEEDBACK RESPONSES**

(75) Inventors: **Adrian FLANAGAN**, Raheen (IE); **Kenneth M. FEEN**, Hospital (IE); **Mark J. Murphy**, Kilmore (IE); **Mel J. CONWAY**, Broadford (IE); **Susan Michelle PRATT**, Caherconlish (IE)

(73) Assignee: **ANALOG DEVICES, INC.**, Norwood, MA (US)

(21) Appl. No.: **13/433,069**

(22) Filed: **Mar. 28, 2012**

Related U.S. Application Data

(60) Provisional application No. 61/470,764, filed on Apr. 1, 2011.

Publication Classification

(51) **Int. Cl.**

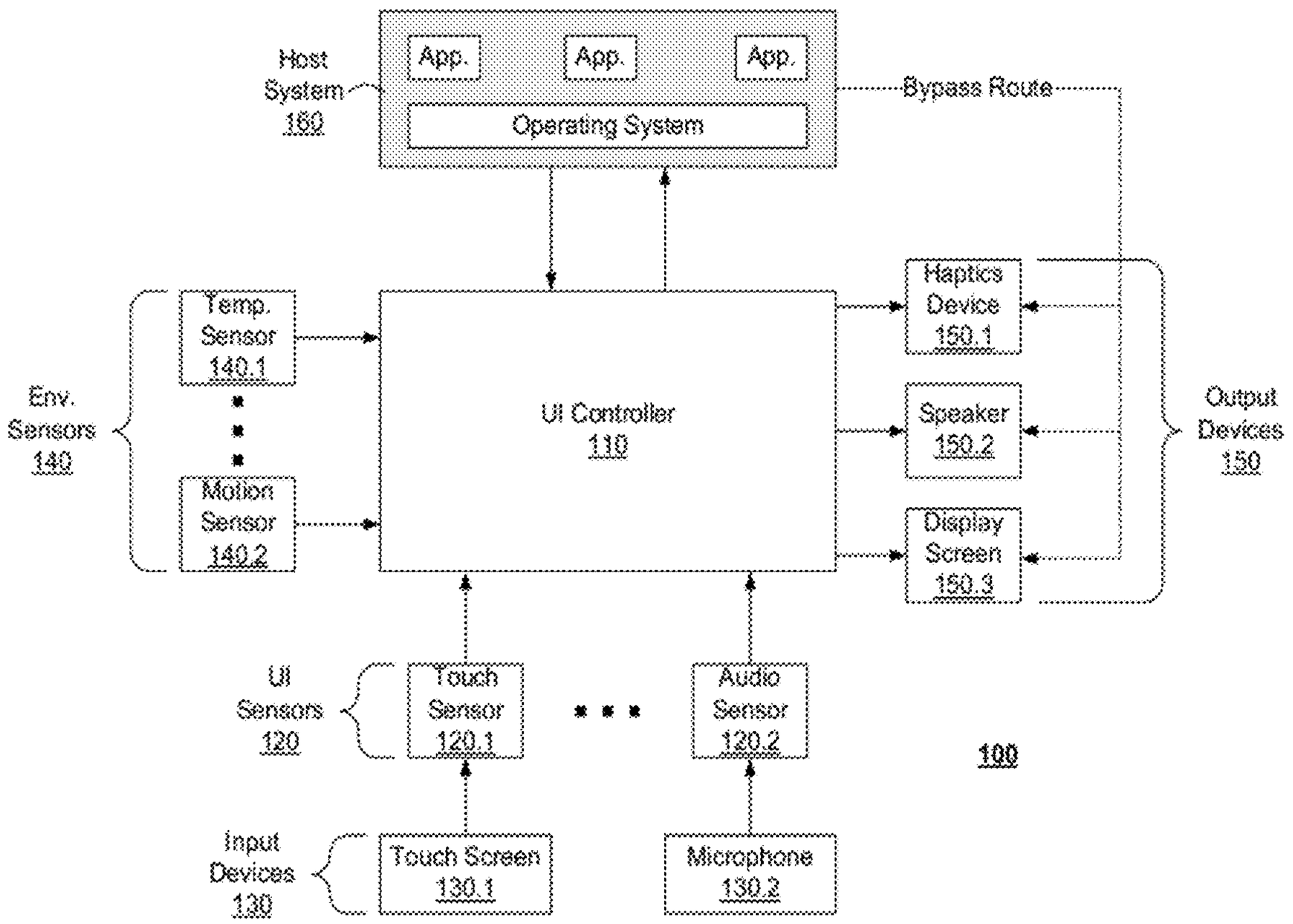
G06F 3/041 (2006.01)

G06F 3/01 (2006.01)

(52) **U.S. Cl.** **345/173**; 345/156

(57) **ABSTRACT**

Embodiments of the present invention provide a user interface processing system for a device that may include at least one sensor, at least one output device, and a controller. The controller may include a memory, which may store instructional information, and a processor. The processor may be configured to receive sensor data from the at least one sensor and to interpret sensor data according to the instructional information. The processor may also generate a user interface feedback command and transmit the command to the at least one output device. Furthermore, the processor may report the sensor data to a host system of the device. By processing the sensor data and generating a corresponding feedback response, for example a haptic response, without the need for host system processing, the user interface controller may decrease latency in providing the feedback response to the user.



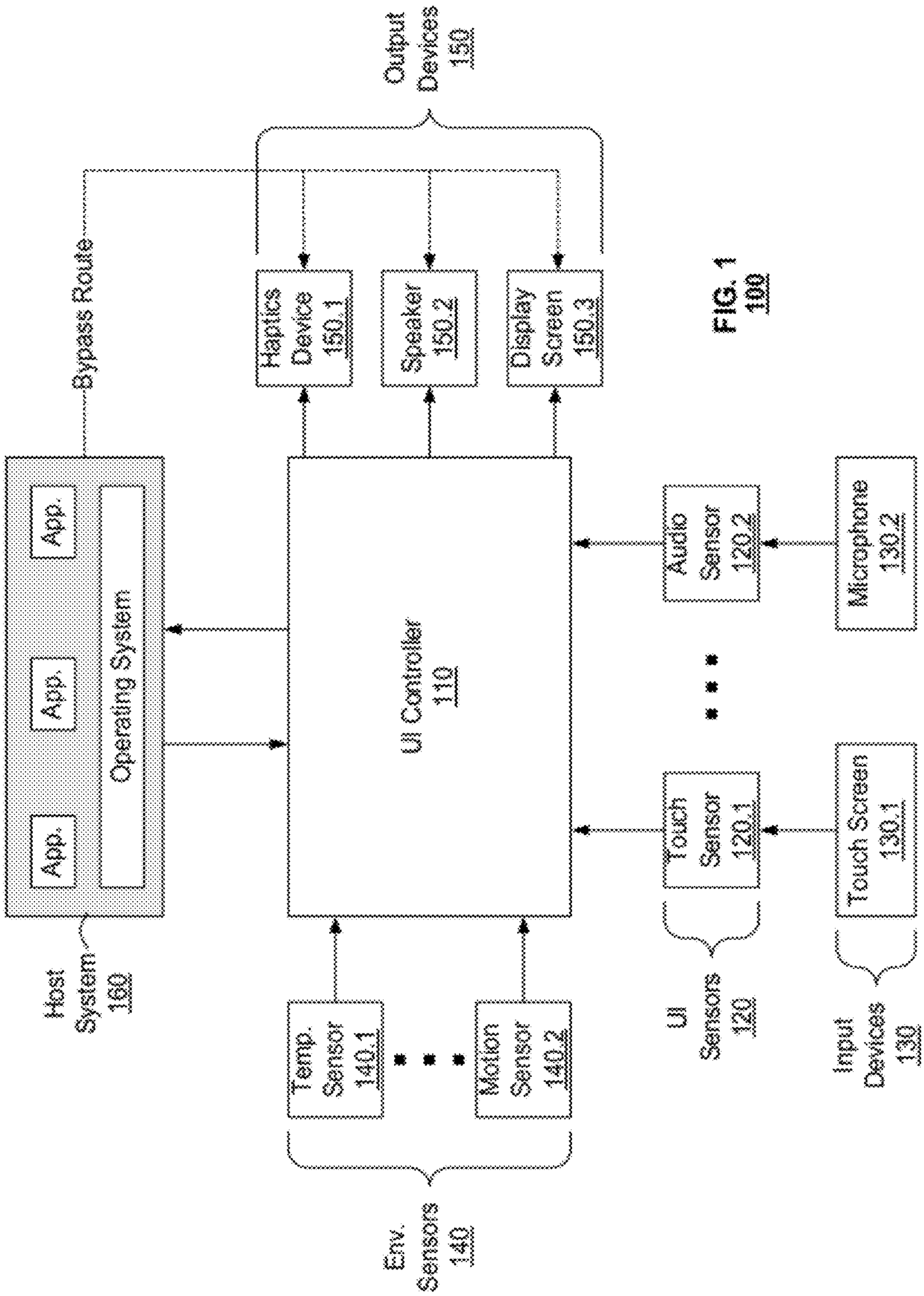

Output Devices 150
Haptics Device 150.1
Speaker 150.2
Display Screen 150.3
Bypass Route
App.
App.
App.
Operating System
UI Controller 110
Audio Sensor 120.2
Microphone 130.2
Touch Sensor 120.1
Touch Screen 130.1
Temp. Sensor 140.1
Motion Sensor 140.2
Env. Sensors 140
UI Sensors 120
Input Devices 130
Host System 160

FIG. 1
100

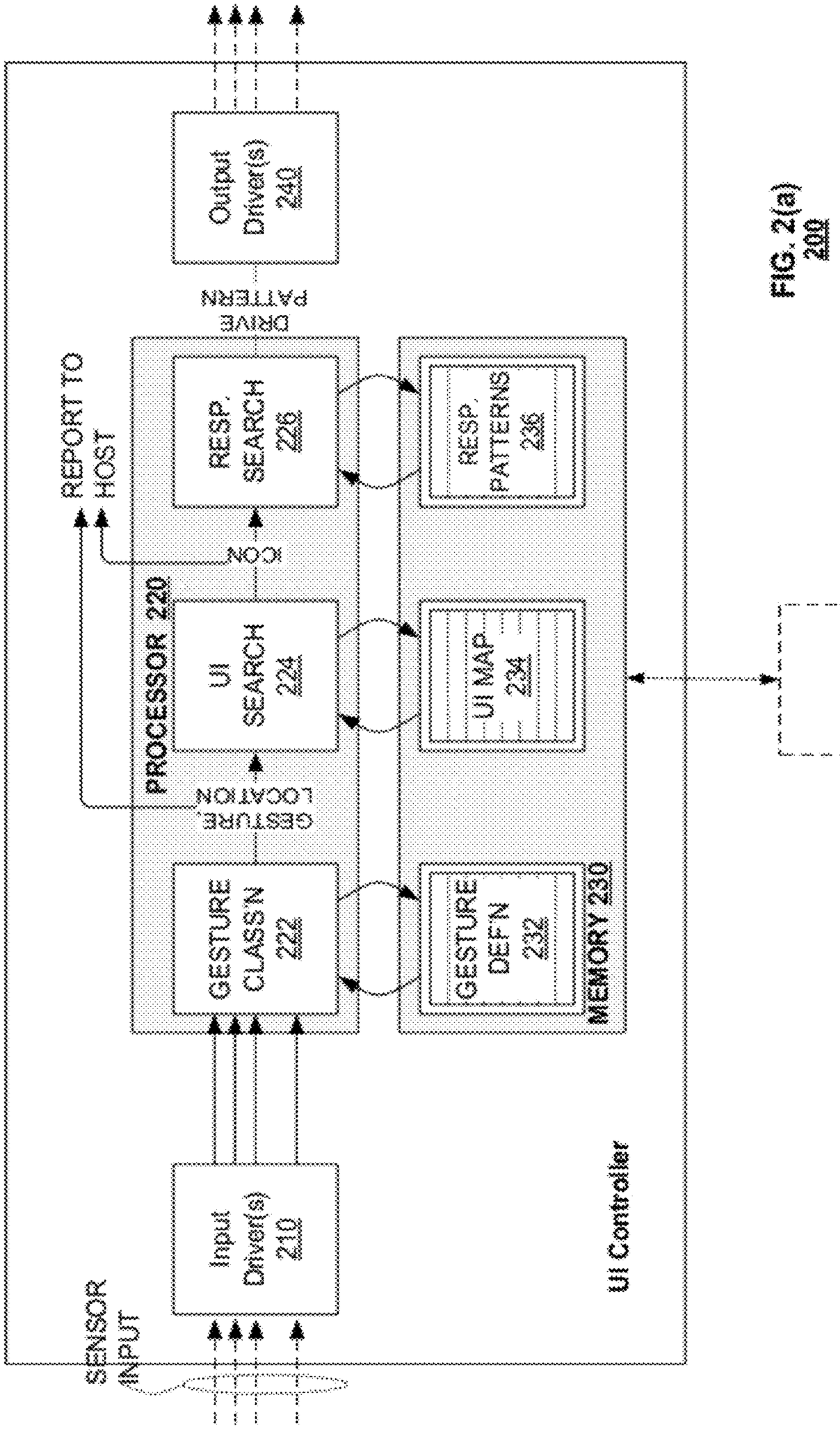

UI Controller
Input Driver(s) 210
SENSOR INPUT
MEMORY 230
GESTURE DEFN 232
UI MAP 234
RESP. PATTERNS 236
PROCESSOR 220
GESTURE CLASSN 222
UI SEARCH 224
RESP. SEARCH 226
GESTURE LOCATION
ICON
REPORT TO HOST
DRIVE PATTERN
Output Driver(s) 240
HOST

FIG. 2(a)
200

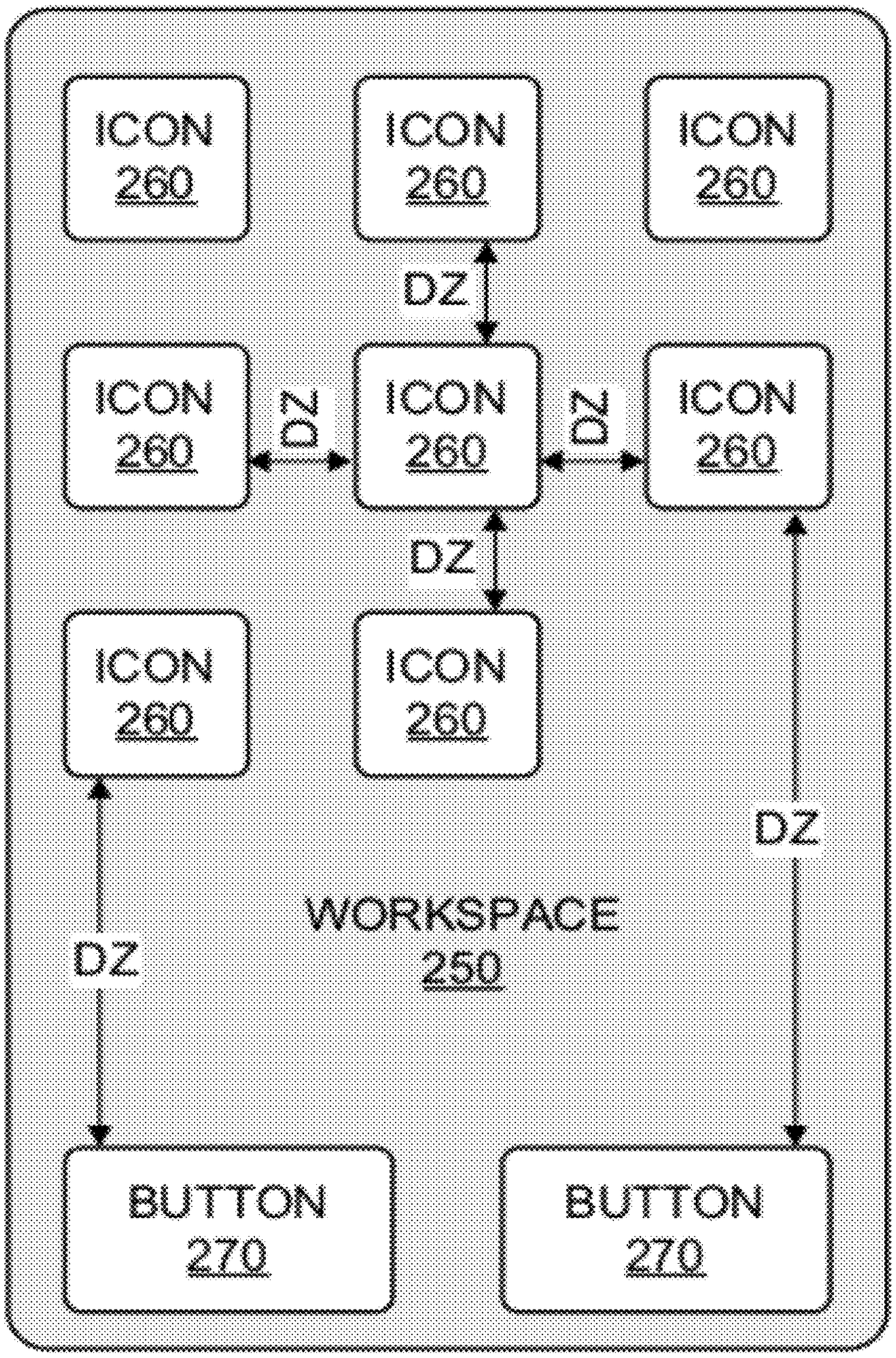

ICON 260
ICON 260
ICON 260
DZ
ICON 260
DZ
ICON 260
DZ
ICON 260
DZ
ICON 260
ICON 260
DZ
DZ
WORKSPACE 250
BUTTON 270
BUTTON 270

FIG. 2(b)

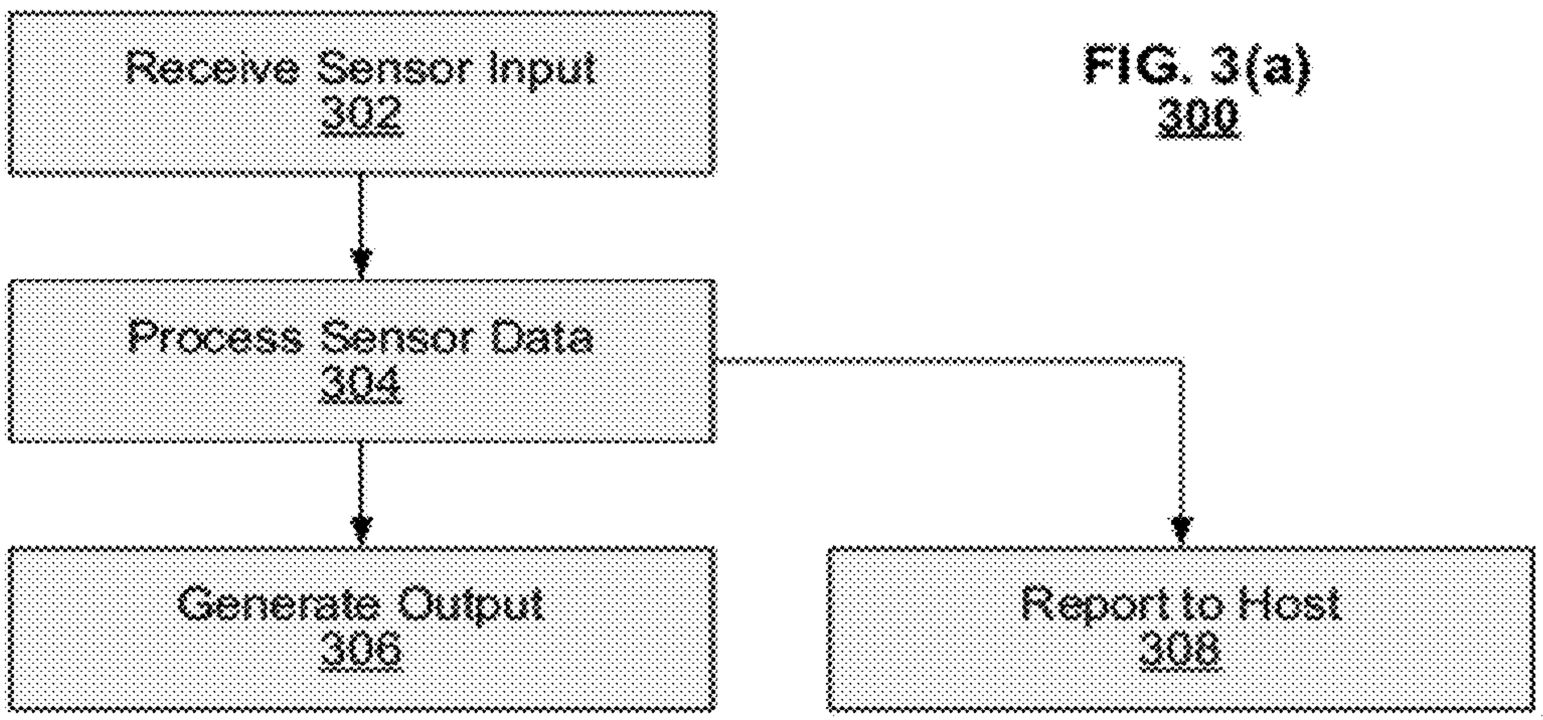


FIG. 3(a)
300

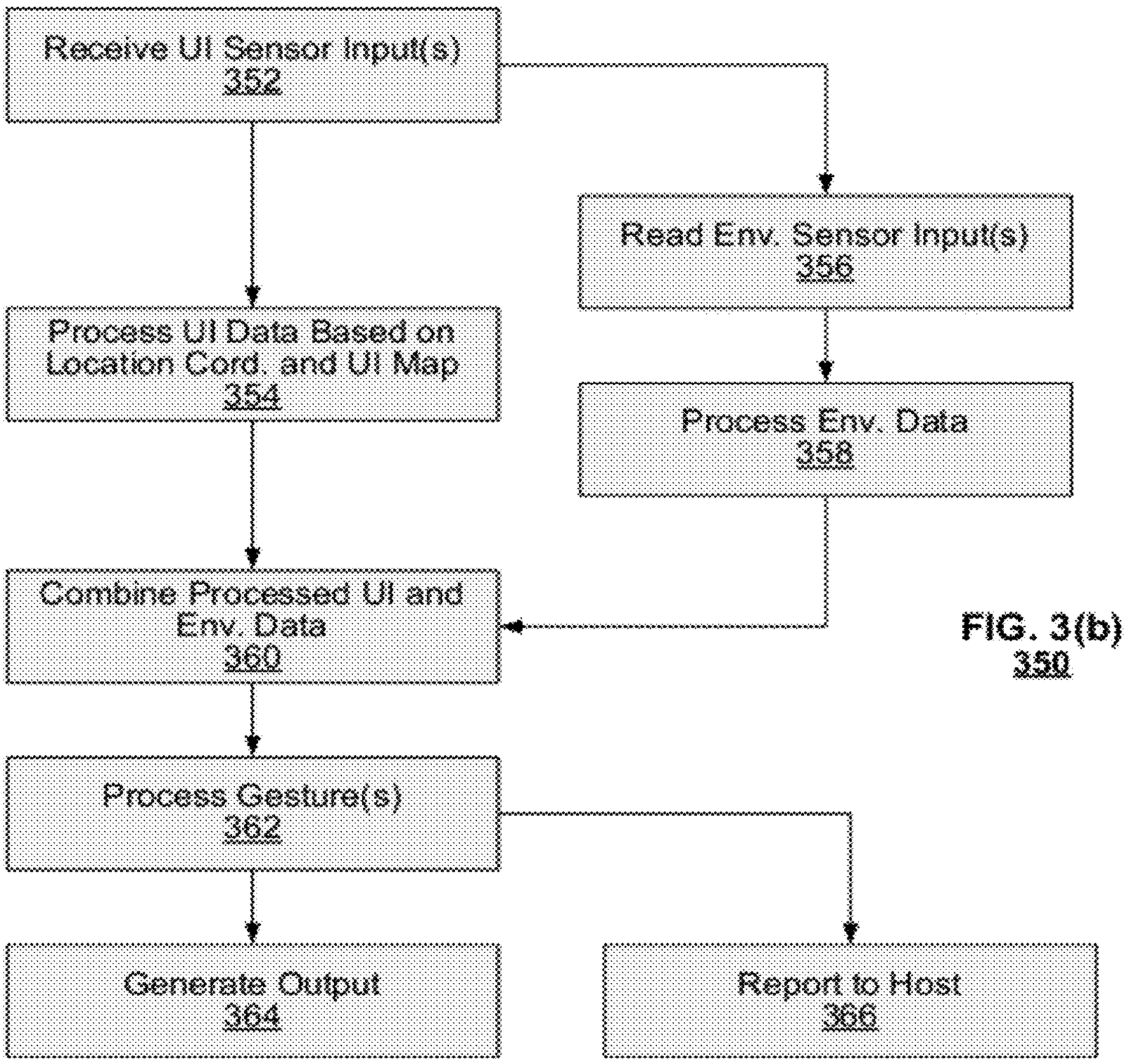


FIG. 3(b)
350

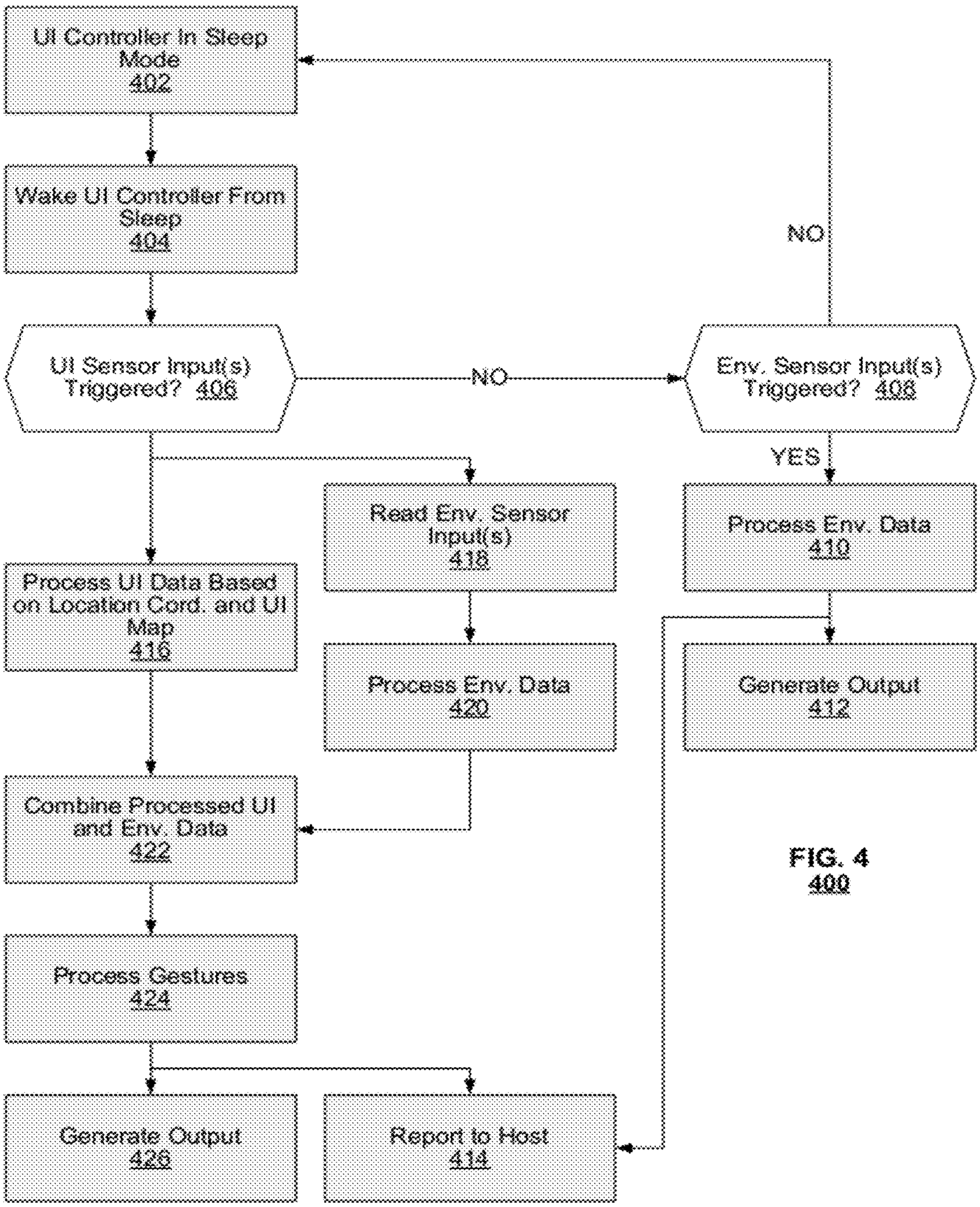

UI Controller In Sleep Mode 402
Wake UI Controller From Sleep 404
UI Sensor Input(s) Triggered? 406
NO
Env. Sensor Input(s) Triggered? 408
YES
Process UI Data Based on Location Cord. and UI Map 416
Read Env. Sensor Input(s) 418
Process Env. Data 420
Process Env. Data 410
Generate Output 412
Combine Processed UI and Env. Data 422
Process Gestures 424
Generate Output 426
Report to Host 414
NO

FIG. 4
400

DEDICATED USER INTERFACE CONTROLLER FOR FEEDBACK RESPONSES

RELATED APPLICATIONS

[0001] This application claims priority to provisional U.S. Patent Application Ser. No. 61/470,764, entitled "Touch Screen and Haptic Control" filed on Apr. 1, 2011, the content of which is incorporated herein in its entirety.

BACKGROUND

[0002] The present invention relates to user interface control.

[0003] Typically, a single host processor controls robust operating functions for a consumer electronic device. One function generally controlled by the host processor is "haptics," which refers to generating tactile feedback to a user of consumer electronics products, for example, when using a touch screen. When a user interacts with a user interface (UI) such as a touch screen, a haptic system produces a mechanical vibration that simulates a "click" of a mechanical actuator. For a user to accept haptics, the haptic response should follow closely in time with the user action. Thus, prolonged latency in the haptic response, which is the delay between the moment of user contact and the corresponding haptics response, causes a disconnect between the touch and the haptic response.

[0004] Bundling all the operating control for a device increases latency in haptic responses as well as other UI feedback responses. This latency is due to the time the device incurs to sense a user interaction, register and decode the interaction, process it through the operating system and/or an active application, select a response to the interaction, and drive the corresponding output device. When the latency exceeds about 250 ms, the latency becomes noticeable to the user and it can be perceived as device error rather than an event that was triggered by the user's input. For example, a user may touch a first button on a touch screen and move onto another function of the device before feeling the haptic response to the first button. This temporal disconnect results in low user acceptance of haptics leading to a poor user experience.

[0005] Furthermore, bundling all the operating control for a device leads to inefficient power consumption. For example, the host processor when in sleep mode generally wakes regularly to check the various bundled functions. Since the host processor typically is one of the larger power consumers in a device, waking the host processor regularly to check each bundled function on the device significantly drains power.

[0006] Hence, the inventors recognized a need in the art for user feedback responses with low latency and low power consumption.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] FIG. **1** is a simplified block diagram of a display device according to an embodiment of the present invention.

[0008] FIG. **2**(*a*) is a simplified block diagram of a user interface (UI) controller according to an embodiment of the present invention.

[0009] FIG. **2**(*b*) is illustrates a two-dimensional workspace according to an embodiment of the present invention.

[0010] FIG. **3**(*a*) illustrates a simplified flow diagram for generating a UI effect according to an embodiment of the present invention.

[0011] FIG. **3**(*b*) illustrates a simplified flow diagram for generating a UI effect according to an embodiment of the present invention.

[0012] FIG. **4** illustrates a simplified flow diagram for operating in sleep mode and to generate a UI effect according to an embodiment of the present invention.

DETAILED DESCRIPTION

[0013] Embodiments of the present invention provide a user interface processing system for a device that may include at least one sensor, at least one output device, and a controller. The controller may include a memory, which may store instructional information, and a processor. The processor may be configured to receive sensor data from the sensor(s) and to interpret sensor data according to the instructional information. The processor may also generate a user interface feedback command and transmit the command to the at least one output device. Furthermore, the processor may report the sensor data to a host system of the device. By processing the sensor data and generating a corresponding feedback response, for example a haptic response, without the need for host system processing, the user interface controller may decrease latency in providing the feedback response to the user.

[0014] FIG. **1** is a simplified block diagram of a haptic-enabled display device **100** according to an embodiment of the present invention. The device **100** may include a User Interface (UI) controller **110**, UI sensor(s) **120** with corresponding input device(s) **130**, environmental sensor(s) **140**, output device(s) **150**, and a host system **160**.

[0015] The UI controller **110** may be coupled to the UI sensors **120** to receive user inputs and to the environmental sensors **140** to receive environmental conditions. The UI controller **110** also may be coupled to the output devices **150** to generate user feedback in response to the detected user inputs and environmental conditions. Moreover, the UI controller **110** may be coupled to the host system **160** of the device. The UI controller **110** may receive instructions from the host system **160** and may transmit processed data from the UI sensors **120** and environmental sensors **140** to the host system **160**. The structure of the UI controller **110** will be described in further detail below.

[0016] The UI sensors **120** may detect user input from their corresponding input devices **130**. A touch screen **130.1** may be provided as an input device **130**. The touch screen **130.1** may be a capacitive touch screen, a stereoscopic capacitive touch screen, a resistive touch screen. The input devices **130** may also be provided as an audio-pick device such as a microphone **130.2**. Moreover, the input devices **130** may be provided as an optical system including a light emitting and light pick-up device, and/or an infra-red light emitting and light pick-up device. Consequently, the UI sensors **120** may be provided as a corresponding touch sensor **120.1**, audio sensor **120.2**, optical sensor, and/or infra-red sensor.

[0017] In another embodiment, the UI sensor(s) **120** may identify proximity events. For example, the UI sensors **120** may detect user fingers approaching the corresponding input device(s) **130** of a touch screen such as a capacitive touch screen. The UI controller **110** may then calculate a proximity event from the UI sensors **120** data.

[0018] The environmental sensors **140** may detect environmental conditions such as location, position, orientation, temperature, lighting, etc., of the device. For example, the environmental sensors **140** may be provided as a temperature

sensor **140.1**, a motion sensor **140.2** (e.g., digital compass sensor, GPS, accelerometer and/or gyroscope), and/or an ambient light sensor.

[0019] The output devices **150** may generate sensory user feedback. The user feedback may be a haptics response to provide a vibro-tactile feedback, an audio response to provide an auditory feedback, and/or a lighting response to provide a visual feedback in response to a user input. The output devices may be provided as a haptics device **150.1**, a speaker **150.2**, a display screen **150.3**, etc. The haptics device **150.1** may be embodied as piezoelectric elements, linear resonant actuators (LRAs) and/or eccentric rotating mass actuators (ERMs). In another embodiment, multiple haptics actuators may be provided to provide plural haptic responses, for example at different parts of the device simultaneously. The speaker **150.2** may provide an audio response, and the display screen **150.3** may provide a visual response. The display screen **150.3** may be provided as a backlit LCD display with an LCD matrix, lenticular lenses, polarizers, etc. A touch screen may be overlaid on face of the display.

[0020] The host system **160** may include an operating system and application(s) that are being executed by the operating system (OS). The host system **160** may represent processing resources for the remainder of the device and may include central processing units, memory for storage of instructions representing an operating system and/or applications, input/output devices such as display driver (not shown), audio drivers, user input keys and the like. The host system **160** may include program instructions to govern operations of the device and manage device resources on behalf of various applications. The host system **160** may, for example, manage content of the display, providing icons and softkeys thereon to solicit user input thru the output devices **150**. The host system **160** may also control the output devices **150** via the UI controller **110** or directly via the bypass route shown in FIG. **1**.

[0021] FIG. **2**(*a*) is a functional block diagram of a UI controller **200** according to an embodiment of the present invention. The UI controller **200** may be implemented in the device **100** of FIG. **1**. The UI controller **200** may include input driver(s) **210**, a processor **220**, a memory **230**, and output driver(s) **240**. The input driver(s) **210** may receive sensor inputs (environmental and/or user interface sensors) and may generate a corresponding input signal. The sensor inputs may be coupled to the input driver(s) **210** via a serial interface such as a high speed I2C interface. The input driver(s) **210** may also control the coupled sensor operations such as when to power on, read data, etc.

[0022] The processor **220** may control the operations of the UI controller **110** according to instructions saved in the memory **230**. The memory **230** may be provided as a non-volatile memory, a volatile memory such as random access memory (RAM), or a combination thereof. The processor **220** may include a gesture classification module **222**, a UI search module **224**, and a response search module **226**. The memory **230** may include gesture definition data **232**, UI map data **234**, and response patterns data **236**. The data may be stored as look-up-tables (LUTs). For example, the gesture definition data **232** may include a LUT with possible input value(s) and corresponding gesture(s). The UI map data **234** may include a LUT with possible input value(s) and corresponding icon (s). Furthermore, the response patterns **236** may include a LUT with possible gesture and icon value(s), and their corresponding response drive pattern(s). Also, the data may be written into the memory **230** by the host system (e.g., OS and/or applications) or may be pre-programmed.

[0023] The gesture classification module **222** may receive the input signal from the input driver(s) **210** and may calculate a gesture from the input signal based on the gesture definition data **232**. For example, the gesture classification module **222** may compare the input signal to stored input value(s) in the gesture definition data **232** and may match the input signal to a corresponding stored gesture value. The gesture may represent a user action on the touch screen indicated by the input signal. The calculated gesture may be reported to the host system.

[0024] The UI search module **224** may receive the input signal from the input driver(s) **210** and may calculate a UI interaction such as an icon selection from the input signal based on the UI map data **232**. For example, the UI search module **224** may compare the input signal to stored input value(s) in the UI map data **232** and may match the input signal to a corresponding UI interaction. The UI interaction may represent a user action on the touch screen indicated by the input signal. The calculated UI interaction may be reported to the host system.

[0025] Further, the response search module **226** may receive the calculated gesture and UI interaction, and may generate a response drive pattern based on the response patterns data **236**. For example, the response search module **226** may compare the stored gesture and UI interaction to stored gesture and UI interaction values, and may match them to a corresponding response drive pattern. The response drive pattern may received by output driver(s) **240**, which, in turn, may generate corresponding drive signals that are outputted to respective output device(s) (i.e., haptic device, speaker, and/or display screen). For example, the drive pattern may correspond to a haptic effect, audio effect, and/or visual effect in response to a user action to provide quick feedback to the user because the UI map data **232**, the UI search module **234**, the response patterns data **236** are available in the UI controller. Thus, the device can output response faster than if OS and application are involved.

[0026] According to an embodiment of the present invention, a haptic-enabled display device may establish interactive user interface elements and provide a haptic response only when user input spatially coincides with a registered element. In another embodiment, a haptics enabled device may register specific haptics response patterns with each of the interactive elements and, when user input indicates interaction with an element, the device responds with a haptic effect that is registered with it.

[0027] FIG. **2**(*b*) illustrates a two-dimensional workspace **250** (i.e., UI map) for use in accordance with embodiments of the present invention. The workspace **250** is illustrated as including a plurality of icons **260** and buttons **270** that identify interactive elements of the workspace **250**. The workspace **250** may include other areas that are not designated as interactive. For example, icons **260** may be spaced apart from each other by a certain separation distance. Further, other areas of the display may be unoccupied by content or occupied with display data that is non-interactive. Thus, non-interactive areas of the device may be may be designated as "dead zones" (DZs) for purposes of user interaction (shown in gray in the example of FIG. **2**(*b*)).

[0028] In an embodiment, the device may output haptics responses when a touch is detected in a spatial area of the workspace that is occupied by an interactive user element. In

an embodiment, the device may be configured to avoid outputting a haptics response when a user interacts with a dead zone of the workspace, even though the device may register a touch at the position. By avoiding outputting of haptics responses for user touches that occur in dead zones, the device improves user interaction by simulating clicks only for properly registered user interactivity.

[0029] FIG. **3**(*a*) illustrates a method **300** of generating a UI effect according to an embodiment of the present invention. In step **302**, the UI controller **110** may receive sensor input(s). The sensor input(s) may be from UI sensor(s) or from environmental sensor(s) or a combination thereof.

[0030] In step **304**, the UI controller **110** may process the sensor data by decoding the data according to instructions stored in its memory. The instructions may be sent from the host system **160** and may include gesture definitions, UI map information, response patterns corresponding to a currently application running on the device **100**. For example, the UI map information may relate to a specific display level/stage in the running application. The UI map may identify spatial areas of the touch screen that are displaying interactive user interface elements, such as icons, buttons, menu items and the like. The UI controller may calculate a gesture and/or user interaction representing the sensor data.

[0031] The instructions may also include user feedback profiles corresponding to the current display level/stage of the running application. For example, the user feedback profiles may define different UI effects such as haptic effects, sound effects, and/or visual effects associated with various sensor inputs.

[0032] In step **306**, the UI controller **110** may generate a UI effect drive pattern, which may be based on the processed sensor data and the stored instructions. The UI controller **110** may transmit the drive pattern to one or more of the output devices **150**, which, in turn, may generate the desired UI effect. As described above, the UI effect may be a sensory feedback to the user such as a haptic effect, sound effect, and/or visual effect. For example, in response to sensed input of user input event of touching an icon, the UI controller **110** may generate a vibrating haptic effect accompanied with a clicking sound to provide the user confirmation of the specific user input event. Thus, the UI controller **110** may generate user feedback response in the form of a UI effect such as a haptic response without the need to involve the host system **160**.

[0033] The UI controller **110** may also report the processed sensor data to the host system **160** in step **308**. The host system **160** may update the running application on device according to the processed sensor data. The host system **160** may then send updated gesture definitions, UI maps, and/or response patterns to the UI controller **110** if the display level/stage of the running application has changed or the running application has ended in response to the processed sensor data. In another embodiment, all instruction data may be sent to the UI controller **110** at the initiation of an application.

[0034] Having a direct sensor-to-output communication path in a device advantageously reduces latency of feedback responses such as haptic events. As noted, during operation, delays of 250 ms between a touch and a haptics response can interfere with satisfactory user experience. Such delays can be incurred in systems that require a host system **160** to decode a user touch and generate a haptics event in response. During high volume data entry, such as typing, texting or cursor navigation, users enter data so quickly that their fingers may have touched and departed a given touch screen location before a 250 ms latency haptics event is generated. Thus, a dedicated UI controller according to embodiments of the present invention as described herein may reduce feedback response latency to improve user experience satisfaction.

[0035] FIG. **3**(*b*) illustrates a method **350** of generating a UI effect according to another embodiment of the present invention. In step **352**, the UI controller **110** may receive UI sensor input(s). The UI sensor input(s) may correspond to a user input event relating to the device **100**. For example, the UI sensor input(s) may come from a capacitive touch sensor, resistive touch sensor, audio sensor, optical sensor, and/or infra-red sensor. In one embodiment, the user input event may identify a proximity event such as when the user's finger(s) approach a touch screen.

[0036] In step **354**, the UI controller **110** may generate location coordinates for the user event and may process the UI sensor data based on the location coordinates and instructions stored in the memory **220**. Typically, location coordinates may be resolved as X,Y coordinates of touch along a surface of the touch screen. Additionally, according to an embodiment of the present invention, location coordinates may also be include a Z coordinate corresponding to the distance from the touch screen, for example in relation to a proximity event.

[0037] As described above, the instructions may be sent from the host system **160** and may include UI map information corresponding to a currently application running on the device **100**. In particular, the UI map information may relate to a specific display level/stage in the running application. The UI map may identify spatial areas of the touch screen that are displaying interactive user interface elements, such as icons, buttons, menu items and the like. The instructions may also include user feedback profiles corresponding to the current display level/stage of the running application. For example, the user feedback profiles may define different UI effects such as haptic effects, sound effects, and/or visual effects associated with various sensor inputs.

[0038] Further in response to receiving UI sensor input(s), the UI controller **110** may read environmental sensor input(s) in step **356**. The environmental sensor input(s) may be indicative of device environmental conditions such as location, position, orientation, temperature, lighting, etc. For example, the environmental sensor input(s) may be provided by an ambient light sensor, digital compass sensor, accelerometer and/or gyroscope.

[0039] In step **358**, the environmental sensor input(s) may be processed based on instructions stored in the memory **220**. As shown in FIG. **3**(*b*), the UI controller **110** may process the UI sensor data while reading and processing environmental sensor data. The parallel processing may further reduce latency issues.

[0040] In step **360**, the processed UI data and environmental data may be combined. In step **362**, the UI controller **110** may process the combined data to interpret user actions such as a gesture. For example, tap strengths may be distinguished by the UI controller if the application uses tap strength levels as different user input events. The UI sensor data may correspond to the location of the tap, and environmental data may correspond to force from an accelerometer measurement. For example, a light tap may be identified by the touch screen as a normal touch while a hard tap may be identified by the accelerometer measurements over a certain threshold level. Thus, a light tap may be distinguished from a hard tap. More-

over, different tap strengths as well as other input variances may designate different gestures.

[0041] Based on the interpreted user action, the UI controller **110** may generate a corresponding UI effect drive pattern in step **364**. The UI controller **110** may generate an effect command for the drive pattern based on the processed sensor data and the stored instructions. The UI controller **110** may transmit the drive pattern to one or more of the output devices **150** to produce the UI effect. As described above, the UI effect may be a sensory feedback to the user such as a haptic effect, sound effect, and/or visual effect.

[0042] Furthermore, the UI controller **110** may also report the interpreted user action to the host system **160** in step **366**. The host system **160** may update the running application on device according to the interpreted user action. The host system **160** may then send updated gesture definitions, UI maps, and/or response patterns to the UI controller **110** if the display level/stage of the running application has changed or the running application has ended in response to the interpreted user action. In another embodiment, all instruction data may be sent to the UI controller **110** at the initiation of an application.

[0043] A dedicated UI controller separate from the host system according to embodiments of the present invention described herein may also advantageously reduce power consumption. Having the host system process UI sensor and environmental sensor inputs is inefficient especially during sleep cycles. Typically, a host system must wake from sleep mode on a regular basis to read the coupled sensor inputs. However, according to an embodiment of the present invention the UI controller may service the sensor inputs and allow the host system to remain in sleep mode. Allowing the host system, generally a large power consumer, to remain in sleep mode for longer periods of time may reduce the overall power consumption of the device.

[0044] FIG. **4** illustrates a method **400** of sleep mode operations and generating a UI effect according to an embodiment of the present invention. In step **402**, the UI controller **110** may be in sleep mode. The host system **160** may also be in sleep mode at this time.

[0045] At step **404**, the UI controller **110** may wake from sleep mode. For example, the UI controller **110** may wake based on a wake up timer trigger or the like. The host system **160** may remain in sleep mode at this time.

[0046] In step **406**, the UI controller **110** may check if any UI sensor inputs are triggered. For example, the UI controller **110** may check if the user has interacted with a selected object to wake the device from sleep mode.

[0047] If no UI sensor inputs are triggered in step **406**, the UI controller **110** may check if any environmental sensor inputs are triggered in step **408**. If no environmental sensor inputs are triggered either, the UI controller **110** may return to sleep mode. However, if an environmental sensor input is triggered, the UI controller **110** may read and process the environmental data in step **410**. If necessary, a feedback output may be generated based on the environmental data in step **412**. Also, if necessary, the environmental data may be reported to the host system in step **414** in turn waking the host system. Alternatively, after processing the environmental data, the UI controller **110** may return to sleep mode if a feedback output is not deemed necessary.

[0048] If a UI sensor input(s) is triggered in step **406**, the UI controller **110** may read and process the UI data. The UI sensor input(s) may correspond to a user event relating to the device **100**. For example, the UI sensor input(s) may come from a capacitive touch sensor, resistive touch sensor, audio sensor, optical sensor, and/or infra-red sensor. In one embodiment, the user event may identify proximity event such as when the user's finger(s) approach a touch screen.

[0049] In step **416**, the UI controller **110** may generate location coordinates for the user event and may process the UI sensor data based on the location coordinates and instructions stored in the memory **220**. Typically, location coordinates may be resolved as X,Y coordinates of touch along a surface of the touch screen. Additionally, according to embodiments of the present events, location coordinates may also be resolved as a Z coordinate corresponding to the distance from the touch screen for a proximity event.

[0050] As described above, the instructions may be sent from the host system **160** and may include The instructions may also include UI map information corresponding to an application running concurrently on the device **110**, in particular to a current display level/stage in the running application. The UI map may identify spatial areas of the touch screen that are displaying interactive user interface elements, such as icons, buttons, menu items and the like. The instructions may also include user feedback profiles corresponding to the current display level/stage of the running application. For example, the user feedback profiles may define different UI effects such as haptic effects, sound effects, and/or visual effects associated with various sensor inputs.

[0051] Further in response to receiving UI sensor input(s), the UI controller **110** may read environmental sensor input(s) in step **418**. The environmental sensor input(s) may be indicative of environmental conditions of the device such as location, position, orientation, temperature, lighting, etc. For example, the environmental sensor input(s) may be provided by an ambient light sensor, digital compass sensor, accelerometer and/or gyroscope.

[0052] In step **420**, the environmental sensor input(s) may be processed based on instructions stored in the memory **220**. As shown, the UI controller **110** may process the UI sensor data while reading and processing environmental sensor data. The parallel processing may further reduce latency issues.

[0053] In step **422**, the processed UI data and environmental data may be combined. In step **324**, the UI controller **110** may process the combined data to interpret use actions such as gesture(s) as described above.

[0054] Based on the interpreted user action, the UI controller **110** may generate a corresponding UI effect drive pattern in step **426**. The UI controller **110** may generate an effect command for the drive pattern based on the processed sensor data and the stored instructions. The UI controller **110** may transmit the drive pattern to one or more of the output devices **150** to produce the UI effect. As described above, the UI effect may be a sensory feedback to the user such as a haptic effect, sound effect, and/or visual effect.

[0055] Furthermore, the UI controller **110** may also report the interpreted user action to the host system **160** in step **414** in turn waking the host system. The host system **160** may update the running application on device according to the interpreted user action. The host system **160** may send updated gesture definitions, UI maps, and/or response patterns to the UI controller **110** if the display level/stage of the running application has changed or the running application has ended in response to the interpreted user action. In another embodiment, all instruction data may be sent to the UI controller **110** at the initiation of an application.

[0056] Those skilled in the art may appreciate from the foregoing description that the present invention may be implemented in a variety of forms, and that the various embodiments may be implemented alone or in combination. Therefore, while the embodiments of the present invention have been described in connection with particular examples thereof, the true scope of the embodiments and/or methods of the present invention should not be so limited since other modifications will become apparent to the skilled practitioner upon a study of the drawings, specification, and following claims.

[0057] Various embodiments may be implemented using hardware elements, software elements, or a combination of both. Examples of hardware elements may include processors, microprocessors, circuits, circuit elements (e.g., transistors, resistors, capacitors, inductors, and so forth), integrated circuits, application specific integrated circuits (ASIC), programmable logic devices (PLD), digital signal processors (DSP), field programmable gate array (FPGA), logic gates, registers, semiconductor device, chips, microchips, chip sets, and so forth. Examples of software may include software components, programs, applications, computer programs, application programs, system programs, machine programs, operating system software, middleware, firmware, software modules, routines, subroutines, functions, methods, procedures, software interfaces, application program interfaces (API), instruction sets, computing code, computer code, code segments, computer code segments, words, values, symbols, or any combination thereof. Determining whether an embodiment is implemented using hardware elements and/or software elements may vary in accordance with any number of factors, such as desired computational rate, power levels, heat tolerances, processing cycle budget, input data rates, output data rates, memory resources, data bus speeds and other design or performance constraints.

[0058] Some embodiments may be implemented, for example, using a computer-readable medium or article which may store an instruction or a set of instructions that, if executed by a machine, may cause the machine to perform a method and/or operations in accordance with the embodiments. Such a machine may include, for example, any suitable processing platform, computing platform, computing device, processing device, computing system, processing system, computer, processor, or the like, and may be implemented using any suitable combination of hardware and/or software. The computer-readable medium or article may include, for example, any suitable type of memory unit, memory device, memory article, memory medium, storage device, storage article, storage medium and/or storage unit, for example, memory, removable or non-removable media, erasable or non-erasable media, writeable or re-writeable media, digital or analog media, hard disk, floppy disk, Compact Disc Read Only Memory (CD-ROM), Compact Disc Recordable (CD-R), Compact Disc Rewriteable (CD-RW), optical disk, magnetic media, magneto-optical media, removable memory cards or disks, various types of Digital Versatile Disc (DVD), a tape, a cassette, or the like. The instructions may include any suitable type of code, such as source code, compiled code, interpreted code, executable code, static code, dynamic code, encrypted code, and the like, implemented using any suitable high-level, low-level, object-oriented, visual, compiled and/or interpreted programming language.

We claim:

1. A user interface processing system for a device, comprising:
- at least one sensor;
- at least one output device; and a
- a controller, including
  - a memory to store instructional information, and
  - a processor to receive sensor data from the at least one sensor, interpret the sensor data according to the instructional information, generate a user interface feedback command, transmit the command to the at least one output device, and to report the sensor data to a host system of the device.

2. The user interface processing system of claim **1**, wherein the instructional information includes gesture definitions and a UI map, and wherein to interpret the sensor data includes calculating a user interaction from the sensor data based on the UI map and calculating a gesture from the sensor data based on the gesture definitions.

3. The user interface processing system of claim **2**, wherein to report the sensor data includes transmitting the calculated user interaction and calculated gesture to the host system.

4. The user interface processing system of claim **2**, wherein the UI map includes an active area and a deadzone where the user interface feedback command is generated if the user interaction is with the active element and no user interface feedback command is generated if the user interaction is with the deadzone.

5. The user interface processing system of claim **1**, wherein the at least one sensor is a touch screen sensor.

6. The user interface processing system of claim **5**, further comprising an input device coupled to the touch screen sensor.

7. The user interface processing system of claim **5**, wherein the touch screen sensor is a proximity sensor.

8. The user interface processing system of claim **5**, wherein the touch screen sensor is a force sensor.

9. The user interface processing system of claim **1**, wherein the at least one sensor is an environmental sensor.

10. The user interface processing system of claim **9**, wherein the environmental sensor is an ambient light sensor.

11. The user interface processing system of claim **9**, wherein the environmental sensor is a digital compass sensor.

12. The user interface processing system of claim **9**, wherein the environmental sensor is an accelerometer or gyroscope.

13. The user interface processing system of claim **1**, wherein the at least one output device is a haptics driver.

14. A display controller for a device, comprising:
- an interface;
- a memory storing data processing instructions; and
- a processor configured to:
  - receiving a sensor input from the interface;
  - processing the sensor input based on the stored data processing instructions;
  - generating a user interface output command;
  - transmitting the user interface output command to an effect generating device; and
  - transmitting the processed sensor input data to a host system of the device.

15. The display controller of claim **14**, wherein the serial interface includes a plurality of I2C connectors that are configured to be read in parallel.

16. A method of generating user interface effects, comprising:

receiving at least one sensor input;

processing the at least one sensor input;

generating a user interface effect based on the processed at least one sensor input and stored instructions; and

reporting the processed at least one sensor input to a host processor.

17. The method of claim **16**, wherein the at least one sensor input includes a user interaction sensor input and an environmental sensor input.

18. The method of claim **17**, wherein the processing includes separately processing the user interaction sensor input based on location coordinates and a user interface map received from the host processor, and the environmental sensor input.

19. The method of claim **16**, further comprises processing a gesture based on the at least one sensor input.

20. The method of claim **16**, wherein the user interface effect is a haptics effect.

21. A method of operating a user interface system in an electronic device, comprising:

placing a user interface controller in sleep mode;

after a predetermined time, waking the user interface controller from sleep mode;

checking a user interface sensor input trigger,

if triggered, reading an environmental sensor input, generating a user interface effect output, and reporting to a host system of the electronic device;

if not triggered, returning to sleep mode unless the environmental sensor input is triggered.

22. The method of claim **21**, further comprises if the environmental sensor input is triggered, generating a user interface effect output and reporting to the host system.

23. The method of claim **21**, wherein the user interface effect command is a haptics effect.

24. A haptics driver, comprising:

an input interface for connection to a touch screen sensor;

an output interface for connection to a haptic device;

a host device interface for connection to a host device;

a memory for storage of:

a UI map representing displayable user interface element(s) associated with the touch screen sensor,

a response pattern(s) associated with the user interface elements, the user interface element(s) and response pattern(s) to be received via the host device interface and stored in the memory; and

a processor to interpret sensor input data received at the input interface, identify whether a user interface element is indicated by the sensor input data and, when a user interface element is so indicated, output an associated response pattern to the output interface.

25. The haptic driver of claim **24**, further comprising, when a user interface element is so indicated, outputting data identifying the indicated user interface element via the host interface.

* * * * *