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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 April 2010 (15.04.2010)

PCT

(10) International Publication Number
WO 2010/042661 A2

(51) International Patent Classification:
G08B 21/00 (2006.01)

(21) International Application Number:
PCT/US2009/059900

(22) International Filing Date:
7 October 2009 (07.10.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/247,950 8 October 2008 (08.10.2008) US

(71) Applicant *(for all designated States except US)*: **DP TECHNOLOGIES, INC.** [US/US]; 245-m Mt. Hermon Road, #363, Scotts Valley, CA 95066 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **KAHN, Philippe** [US/US]; 777 Hudson Lane, Aptos, CA 95003 (US). **KINSOLVING, Arthur** [US/US]; 101 Soquel Ave. #403, Santa Cruz, CA 95060 (US). **VOGEL, David** [US/US]; 600 Beel Drive, Santa Cruz, CA 95060 (US). **CHRISTENSEN, Mark, Andrew** [NZ/US]; 107 Brookwood Drive, Santa Cruz, CA 95065 (US).

(74) Agents: **VINCENT, Lester, J.** et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 1279 Oakmead Parkway, Sunnyvale, CA 94085-4040 (US).

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

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

Published:
— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

WO 2010/042661 A2

(54) **Title**: METHOD AND SYSTEM FOR WAKING UP A DEVICE DUE TO MOTION

(57) **Abstract**: A method comprises determining an idle sample value for a dominant axis of a device in an idle state. The method further comprises registering a motion of the device, and evaluating the motion. The method further comprises waking up the device when the analysis of the motion indicates a change in the dominant axis of the device and/or a level of acceleration beyond a threshold.

METHOD AND SYSTEM FOR WAKING UP A DEVICE DUE TO MOTION

FIELD OF THE INVENTION

[001] This invention relates to a method and system for waking up a device from an idle state.

BACKGROUND

[002] Technological progress has led to the proliferation of commercial electronic devices such as portable computers, game controllers, GPS devices, digital cameras, cellular telephones, and personal media players. Continuous improvements have allowed the users to enjoy many features and possible uses from a single mobile device. However, generally, the more applications a mobile device has, the faster the battery of the mobile device depletes. Therefore, it can be difficult to maximize battery life and provide a great user experience at the same time.

SUMMARY OF THE INVENTION

[003] The present invention provides a method and system to wake up a device due to motion. The system determines a dominant axis of a device. The device is placed in an idle state, after a period of inactivity or lack of motion. A sensor, such as an accelerometer, registers a motion of the device. A computation logic analyzes the motion data to determine if the motion data indicates a real motion. If so, the device is woken up.

BRIEF DESCRIPTION OF THE DRAWINGS

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

[005] Figure 1 is an illustration of one embodiment of moving a device that may require waking up the device.

[006] Figure 2 is a block diagram of one embodiment of a system.

[007] Figure 3 is a flowchart of one embodiment of determining whether to wake up a device based on motion data.

[008] Figure 4 is a flowchart of one embodiment of a process to create a long average of accelerations.

[009] Figure 5 is a flowchart of one embodiment of a process for determining whether a device should be woken up from an idle state.

[0010] Figure 6 is a flowchart of one embodiment of a process to detect and correct glitches in motion data.

[0011] Figure 7 is a block diagram of one embodiment of a computer system that may be used with the present invention.

DETAILED DESCRIPTION

[0012] A method and system for waking up a device due to motion of the device is described. Embodiments of the present invention are designed to determine if a device should be woken up from an idle state based on the analysis of motion data. In one embodiment, motion data for the dominant axis is analyzed and the device is woken up from idle state if the motion data analysis points to the motion being “real” motion as opposed to a mere jostle or glitch.

[0013] The following detailed description of embodiments of the invention makes reference to the accompanying drawings in which like references indicate similar elements, showing by way of illustration specific embodiments of practicing the invention. Description of these embodiments is in sufficient detail to enable those skilled in the art to practice the invention. One skilled in the art understands that other embodiments may be utilized and that logical, mechanical, electrical, functional and other changes may be made without departing from the scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims.

[0014] Figure 1 is an illustration of one embodiment of moving an idle device that may result in waking up the device. The idle state is defined, in one embodiment, as a state in which the device is not moving, and there is no active application which includes user interaction/display. In one embodiment, there may be multiple levels of idle state, e.g. where various subsystems are placed in a power-reduced state or not. When the device is in the idle state, the device is placed in low-power mode. In this state, there is sufficient power maintained to monitor at least one sensor. However, other elements and applications are turned off to extend the battery life of the device. In one embodiment, some applications may remain active. For example, the device may be in the idle state, but continue a download, utilizing a network and memory store. In one embodiment, if at least one subsystem is turned off due to lack of device motion, this may be considered an "idle state."

[0015] In one embodiment, after a device 110 is placed on a horizontal surface 115 such as a desk or chair, after a period of inactivity the device 110 goes to the idle state to conserve the battery. In one embodiment, the device is placed into the pocket, purse, bag, or any other non-moving location, the device enters the idle state.

[0016] The system, in one embodiment, is designed to ensure that when the device is picked up by a user, the device is moved from the idle state to an active state rapidly. By initiating the transition from the idle state to the active state without requiring user input, the user wait is reduced. For example, when a user 100 picks up the device 110 from its position on the horizontal surface 115, the device is designed to wake up. In one embodiment, the device 110 is woken up from idle state and the user is presented the last active state of the device. In one embodiment, this may be sufficiently rapid that by the time the device is being viewed by the user, the prior state has been restored. In contrast, if the table on which the device is resting is shaken, or the purse is jostled, the device should not wake up. This reduces power usage, because the device is not continuously being woken up from small motions which occur when someone walks near a table, sits down, or similarly causes small motions.

[0017] Figure 2 is a block diagram illustrating one embodiment of a system 200 of the present invention. In one embodiment, the system 200 is a portable electronic device. The system 200 in one embodiment comprises motion sensor logic 210, sample period logic 230, glitch correcting logic 235, long average logic 240, dominant axis logic 245, memory 250, computation logic 255, and configuration logic 260.

[0018] In one embodiment, the motion sensor logic 210 comprises an accelerometer 220. In one embodiment, the motion sensor logic 210 also includes one or more additional sensors, such as orientation sensor 215.

[0019] In one embodiment, accelerometer 220 may be used to determine orientation of the device. The orientation may be determined using long averages of accelerations. The sample period logic 230 determines how frequently the motion sensor logic 210 obtains data. In one embodiment, the sample period is preconfigured. In one embodiment, the sample period is adjusted based on the application(s) using the sensor data.

[0020] The accelerometer 220 periodically samples motion data. The long average logic 240 calculates an average of the acceleration data over the sample period. In one embodiment, the long average logic 240 calculates the average of the accelerations over a number of measurements, rather than over a time period. In one embodiment, the long average logic 240 calculates accelerations over 5 minutes. In one embodiment, the long average logic 240 calculates accelerations over 20 measurements.

[0021] In one embodiment, the acceleration data is sent to the glitch correcting logic 235, where the data is analyzed to determine if any it represents a glitch, i.e., data outside a pre-determined range of acceptable data. For example, it is extremely unlikely if not impossible for motion data to go from zero acceleration to 10m/s acceleration in one reading. In one embodiment, the pre-determined range of data is a predetermine change in acceleration from a current acceleration. For example, if the device is idle – e.g. not moving – the range of accelerations possible for the device is fairly limited. In one embodiment, glitch correcting logic 235 further may be used to discard non-human motions. For example, if the device is not being used but is in a moving vehicle, in one embodiment the vehicle’s motion can be discarded as not fitting the signature of human motion.

[0022] In one embodiment, the glitch correcting logic 235 discards any abnormal accelerometer reading(s). In one embodiment, the non-glitch data is then passed on to the long average logic 240. In another embodiment, the glitch data is from the long average by glitch correcting logic 235. In one embodiment, if a certain number of glitch data points have been discarded, glitch notifier logic 237 notifies the user. In one embodiment, glitch notifier logic 237 may also notify the manufacturer. The glitches generally are indicative that the accelerometer or sensor is malfunctioning.

[0023] The long average logic 240 calculates one or more long averages of acceleration based on the received motion data. In one embodiment, the long average logic 240 utilizes a ring buffer memory 250, discarding older data as new data is added to the long average. In one embodiment, the long average logic 240 creates a long average of accelerations along a single axis. In one embodiment, the dominant axis – defined as the axis most impacted by gravity -- is used by the long average logic 240. In one embodiment, the axis corresponds to one of the axes of the accelerometer. In one embodiment, the axis is defined as the orientation experiencing the most pull from gravity. In one embodiment, the long average logic 240 creates long averages of accelerations along multiple axes.

[0024] Determining the orientation of an electronic device may include identifying a gravitational influence. The axis with the largest absolute long average may be the axis most influenced by gravity, which may change over time (e.g., as the electronic device is rotated). Therefore, a new dominant axis may be assigned when the orientation of the electronic device and/or the inertial sensor(s) attached to or embedded in the electronic device changes.

[0025] In one embodiment, the actual axis with the largest absolute long average over a sample period is assigned as the dominant axis. In alternative embodiment, the dominant axis does not correspond to one of the actual axes of the inertial sensor(s) in a current orientation, but rather to an axis that is defined as approximately aligned to gravity. In one embodiment, the dominant axis corresponds to a virtual axis that is a component of a virtual coordinate system. In one embodiment, a true gravity assessment, such as by doing trigonometric calculations on the actual axes based on the gravitational influence is performed to determine orientation.

[0026] In one embodiment, a long average of accelerations is computed by the long average logic 240 when the device goes into idle state after a period of inactivity. In one embodiment, the long average and the dominant axis for which it is computed are received by computation logic 255. The computation logic 255 also receives, based on a new sample of motion data, a current dominant axis and an updated current long average for the current dominant axis.

[0027] If the prior and current dominant axes are the same, the computation logic 255 determines if the long average has changed by more than a predetermined threshold. In one embodiment, when the change in the dominant axis is larger than the threshold value, the computation logic 255 communicates with the power logic 265 and the device state logic 270, to power up the device and restore the last active device state. If the change in the dominant axis is not larger than the threshold value, the device is maintained in the idle state.

[0028] In one embodiment, if the new dominant axis is different from the prior dominant axis, the computation logic 255 communicates with the power logic 265 and configuration logic 260 to restore the device to the last active device state.

[0029] Figure 3 is a flowchart of one embodiment of determining whether to wake up a device based on motion data. At block 305, the process starts. In one embodiment, the process runs continuously. In one embodiment, the user may initiate the auto-wake-up system, or set a preference to have the auto-wake-up system on.

[0030] At block 310, the process determines if it is time to sample motion data. In one embodiment, the motion data is sampled periodically. If it is time to sample motion data, the process continues to block 315. Otherwise, the process returns block 310.

[0031] At block 315, the process gets sample motion data. In one embodiment, based on the sample motion data, at least one current/updated long average of accelerations is calculated. In one embodiment, the long average is based on a preset number of measurements, or on a preset time. The process continues to block 320.

[0032] At block 320, the process determines whether the device is in idle state. In one embodiment, the device is placed in idle state after the device has been inactive for a period of time. Inactive, in one embodiment, means that the device is not moving and that there are no user-interactive applications active on the device. In one embodiment, when the device is placed in idle state, a long average is initialized. If the device is not in idle state, the process returns to block 310. If the process determines that the device is in idle state, the process continues to block 325.

[0033] At block 325, the process determines if the device has experienced any motion, e.g. there is a difference between the readings of the accelerometer that are larger than a minimum threshold. In one embodiment, this determination is made by using a filter to remove accelerometer motions below the minimum threshold. If the process determines that no motion has been detected, the process returns to block 310. If the process determines that the accelerometer data indicates a movement of the device, the process continues to block 330.

[0034] At block 330, the process determines if the movement is a “real” motion and not a mere jostle or bump. The device may move, for example, as a result of a little jostle of a desk or table on which the device is laying, a heavy step nearby, or something else that creates a very small motion, but which does not warrant waking up the device. In contrast, the device may move as a result of being picked up by a user intending to use the device. In this case, the movement is a “real” motion which warrants awakening the device.

[0035] If the motion is not a “real” motion, the process returns to block 310. If the movement is a “real” motion, the process continues to block 335. At block 335, the process wakes up the device. The process continues to block 340.

[0036] At block 340, the process in one embodiment configures the device to restore the last device state when the device was active. In one embodiment, the system allows the user to customize the wake-up restoration of the device. For example, the user may customize the system not to start the previously-active applications, but to present a home screen. The process then ends.

[0037] Figure 4 is a flowchart of one embodiment of a process to create a long average of accelerations. The process 400 starts at block 405. In one embodiment, this process is continuously running when the device is powered.

[0038] At block 410, the long average logic, in one embodiment, receives motion data from the accelerometer. In one embodiment, the long average logic receives the data from a glitch correcting logic which removes abnormal data from the motion data before the motion data is passed to the long average logic. The process continues to block 415.

[0039] At block 415, the long average logic adds the sampled motion data to the long average, to create an updated long average of accelerations. In one embodiment, the long average logic maintains a long average only for the dominant axis (e.g., the axis on which the gravitational effect is detected). In another embodiment, the long average logic maintains an average for one or more axes. The process continues to block 420.

[0040] At block 420, the long average logic, in one embodiment, optionally sends the long averages of accelerations for a plurality of axes to the dominant axis logic for determination of the dominant axis. In an alternative embodiment, the dominant axis logic retrieves the long averages of accelerations for a plurality of axes from memory to determine the dominant axis. The process then returns to block 410, to continue receiving motion data.

[0041] Figure 5 is a flowchart of one embodiment of a process 500 for determining whether a device should be woken up from an idle state. The process starts at block 505. In one embodiment, the process is activated when a preset period with no motion has been detected.

[0042] At block 510, the process places the device in idle state after the device has been inactive for a period of time. The process continues to block 515.

[0043] At block 515, the computation logic receives data for the dominant axis DA1 of the idle device and accelerations along DA1 over a sampling period, computed by the long average logic after the device becomes idle. The process continues to block 520.

[0044] At block 520, the computation logic assigns the long average of accelerations along DA1 over a period to Idle Sample (IS). In one embodiment, IS is saved to memory. The process continues to block 525.

[0045] At block 525, the process receives new dominant axis data DA2 and the new acceleration data along DA2. The process continues to block 530.

[0046] At block 530, the computation logic adds the new data to the long average of accelerations along DA2 to generate a Current Sample (CS). Also at block 530, in one embodiment, the computation logic saves CS to memory. The process continues to block 535.

[0047] At block 535, the computation logic compares the idle dominant axis DA1 with the current dominant axis DA2. If the current dominant axis DA2 is different from the idle dominant axis DA1, the process continues to block 545. In one embodiment, the comparison is within a range, e.g. a minimum change of one degree has to occur to identify DA2 as being different from DA1. In one embodiment, if the dominant axis has changed, then the orientation of the device has changed, and that warrants waking up the device. If DA2 is substantially the same as DA1, then the computation logic continues to block 540.

[0048] At block 540, the computation logic determines if the long average along the dominant axis has changed by more than a threshold value, i.e., if the difference between the Current Sample value and the Idle Sample value is larger than the threshold value. In one embodiment, the threshold value is set to 30, which is approximately a 10^{th} of a g. If the difference between IS and CS is less than the threshold value, the process returns to block 510, to continue monitoring the idle state. CS becomes IS, for the next calculation.

[0049] If the computation logic determines that the change in the long average of accelerations along the dominant axis is greater than the threshold, then the computation logic continues to block 545. At block 545, the computation logic communicates with the power logic of the configuration logic to start up the device. The process then ends.

[0050] Figure 6 is a flowchart of an embodiment of a process 600 to detect and correct glitches in motion data. In one embodiment, this process is always active. In one embodiment, this process is active when the device is in the idle state. In one embodiment, the glitch correction takes place before the motion data is added to the long average. The process starts at block 605.

[0051] At block 610, the glitch correcting logic receives motion data from an accelerometer.

[0052] At block 615, the glitch correcting logic determines if the received motion data contains a glitch. In one embodiment, a glitch is a datum that indicates a motion outside an acceptable range. For example, it is extremely unlikely that a device would go from idle (e.g., no motion) to moving at an acceleration of 64 feet per second squared (equivalent to 2 g). The correcting logic examines each datum against a range of acceptable data to determine if the datum falls within this range and, therefore, should be used in calculating the long average of accelerations. In one embodiment, the glitch correction logic utilizes the change in acceleration between two readings to determine whether there is a glitch.

[0053] If the glitch correcting logic determines that the motion data is not a glitch, the glitch correcting logic continues to block 625.

[0054] At block 625, the glitch correcting logic sends the motion data to the long average logic. The process then returns to block 610, to continue monitoring the acceleration data.

[0055] If at block 615, the glitch correcting logic determines that the motion data is outside the allowable range, the glitch correcting logic continues to block 635.

[0056] At block 635, the glitch correcting logic discards the unacceptable motion data. At block 640, the process determines whether there have been an excessive number of glitches. In one embodiment, the glitch correcting logic uses the motion data to detect a possible problem with the accelerometer. In one embodiment, an excessive number of glitches may indicate a problem with the accelerometer. If the process determines that there have been an excessive number of glitches, the process, at block 645, generates an alert regarding the problem. In one embodiment, the alert may be a message to alert the user of the device. In one embodiment, the alert may be a notification to one or more recipients via a network connection. For example, the system may notify a service provider, manufacturer, or other appropriate notification target.

[0057] The process then returns to block 610, to continue monitoring the acceleration data.

[0058] Figure 7 is a block diagram of one embodiment of a computer system that may be used with the present invention. It will be apparent to those of ordinary skill in the art, however that other alternative systems of various system architectures may also be used. The computer system may include a bus or other internal communication means 715 for communicating information, and a processor 710 coupled to the bus 715 for processing information. The system further comprises a random access memory (RAM) or other volatile storage device 750 (referred to as memory), coupled to bus 715 for storing information and instructions to be executed by processor 710. Main memory 750 also may be used for storing temporary variables or other intermediate information during execution of instructions by processor 710. In one embodiment, the system also comprises a read only memory (ROM) and/or static storage device 720 coupled to bus 715 for storing static information and instructions for processor 710, and a data storage device 725 such as a flash memory, magnetic disk, optical disk and its corresponding disk drive. Data storage device 725 is coupled to bus 715 for storing information and instructions.

[0059] The system may include various input/output devices, such as a screen, audio output, keyboard, button, mouse, etc. These I/O devices may also be coupled to bus 715 through bus 765 for communicating information and command selections to processor 710. Another device, which may optionally be coupled to computer system 700, is a communication device 790 for accessing other nodes of a distributed system via a network. The communication device 790 may include any of a number of commercially available networking peripheral devices such as those used for coupling to an Ethernet, token ring, Internet, or wide area network. The communication device 790 may further be a null-modem connection, a wireless connection mechanism, or any other mechanism that provides connectivity between the computer system 700 and the outside world. Note that any or all of the components of this system and associated hardware may be used in various embodiments of the present invention. It will be appreciated by those of ordinary skill in the art that any configuration of the system may be used for various purposes according to the particular implementation.

The control logic or software implementing the present invention can be stored in main memory 750, mass storage device 725, or other storage medium locally or remotely accessible to processor 710.

[0060] It will be apparent to those of ordinary skill in the art that the system, method, and process described herein can be implemented as software stored in main memory 750 or read only memory 720 and executed by processor 710. This control logic or software may also be resident on an article of manufacture comprising a computer readable medium having computer readable program code embodied therein and being readable by the mass storage device 725 and for causing the processor 710 to operate in accordance with the methods and teachings herein.

[0061] The present invention may also be embodied in a handheld or portable device containing a subset of the computer hardware components described above. For example, the handheld device may be configured to contain only the bus 715, the processor 710, and memory 750 and/or 725. The present invention may also be embodied in a special purpose appliance including a subset of the computer hardware components described above. For example, the appliance may include a processor 710, a data storage device 725, a bus 715, and memory 750, and only rudimentary communications mechanisms, such as a small touch-screen that permits the user to communicate in a basic manner with the device. In general, the more special-purpose the device is, the fewer of the elements need be present for the device to function. In some devices, communications with the user may be through a touch-based screen, or similar mechanism.

[0062] It will be appreciated by those of ordinary skill in the art that any configuration of the system may be used for various purposes according to the particular implementation. The control logic or software implementing the present invention can be stored on any machine-readable medium locally or remotely accessible to processor 710. A machine-readable medium includes any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer). For example, a machine readable medium includes read-only memory (ROM), random access memory (RAM), magnetic disk storage media, optical storage media, flash memory devices. In one embodiment, the system may be embodied in a signal, such as an electrical, optical, acoustical or other forms of propagated signal (e.g., carrier waves, infrared signals, digital signals, etc.).

[0063] In the foregoing specification, the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A method comprising:

determining an idle sample value for a dominant axis of a device;

registering a motion of the device; and

waking up the device when the motion of the device indicates a change in the dominant axis of the device.

2. The method of claim 1, wherein determining the idle sample value for the dominant axis comprises:

receiving motion data from a motion sensor;

processing the motion data to establish an idle sample value; and

processing the idle sample value to establish the dominant axis.

3. The method of claim 2, wherein the motion sensor comprises an accelerometer.

4. The method of claim 2, wherein the idle sample value comprises a long-average of accelerations over a sample period along the dominant axis recorded when the device goes to idle mode after a period of inactivity.

5. The method of claim 1, further comprising determining the idle sample value for each of the other axes of the device.

6. The method of claim 1, wherein registering the motion of the device comprises:

receiving motion data from a motion sensor; and

processing the motion data to determine a current sample value of the dominant axis of the device.

7. The method of claim 1, further comprising comparing a difference between a current sample value along the dominant axis determined based on the motion of the device and the idle sample value of the dominant axis against a threshold value.

8. The method of claim 1, wherein the change in the dominant axis comprises a change in acceleration along the dominant axis.

9. The method of claim 1, wherein waking up the device further comprises configuring the device to return to a last active device state.

10. The method of claim 6, wherein the current sample value is a long average of accelerations.

11. The method of claim 6, further comprising determining the current sample value for each of the other axes of the device.

12. The method of claim 6, wherein the motion sensor comprises an accelerometer.

13. The method of claim 6, wherein processing the motion data further comprises

verifying whether the motion data comprises one or more glitches; and

removing the one or more glitches in the motion data from the motion data before calculating the long average.

14. The method of claim 6, further comprising determining that the device is to be woken up based on the difference between the current sample value and the idle sample value being greater than a threshold value.

15. The method of claim 8, further comprising:

determining a new dominant axis based on the motion data received from the motion sensor;

computing a difference between the current sample value along the new dominant axis and an idle sample value along the new dominant axis determined when the device goes to idle mode after a period of inactivity; and

comparing the difference against a threshold value to establish whether to wake the device up.

16. A system comprising:

a long average logic to create one or more long averages of accelerations as measured by a motion sensor over a period of time;

a dominant axis logic to determine a dominant axis of a device based on motion data; and

a computation logic to determine if the long averages of accelerations indicate a true motion of the device.

17. The system of claim 16, further comprising a motion sensor logic to detect motion data.

18. The system of claim 17, wherein the motion sensor logic comprises an accelerometer to detect acceleration along one or more axes.

19. The system of claim 16, further comprising a sample period logic to set the period over which motion data is collected to compute the one or more long averages of accelerations.

20. The system of claim 16, further comprising a power logic to activate the device when the motion data indicates the device should be woken up.

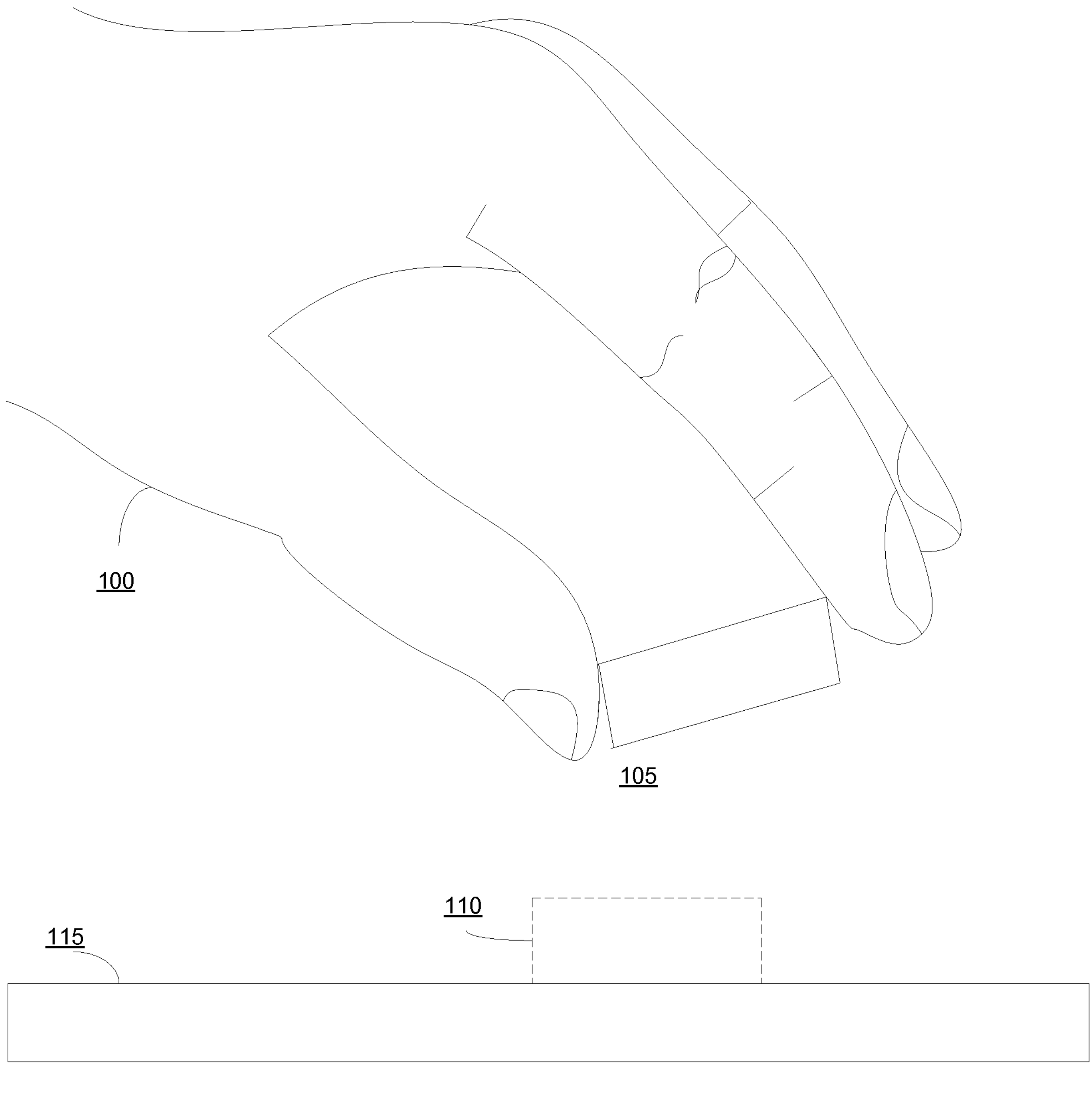
21. The system of claim 16, further comprising a device state logic to restore the device to a last active state.

22. The system of claim 21, wherein the device state logic allows user interaction to customize applications to be displayed when the device is woken up.

23. The system of claim 16, further comprising a glitch corrector logic to correct one or more glitches in the motion data.

24. The system of claim 23, wherein the glitch corrector removes the one or more glitches before the one or more long averages are calculated.

Figure 1



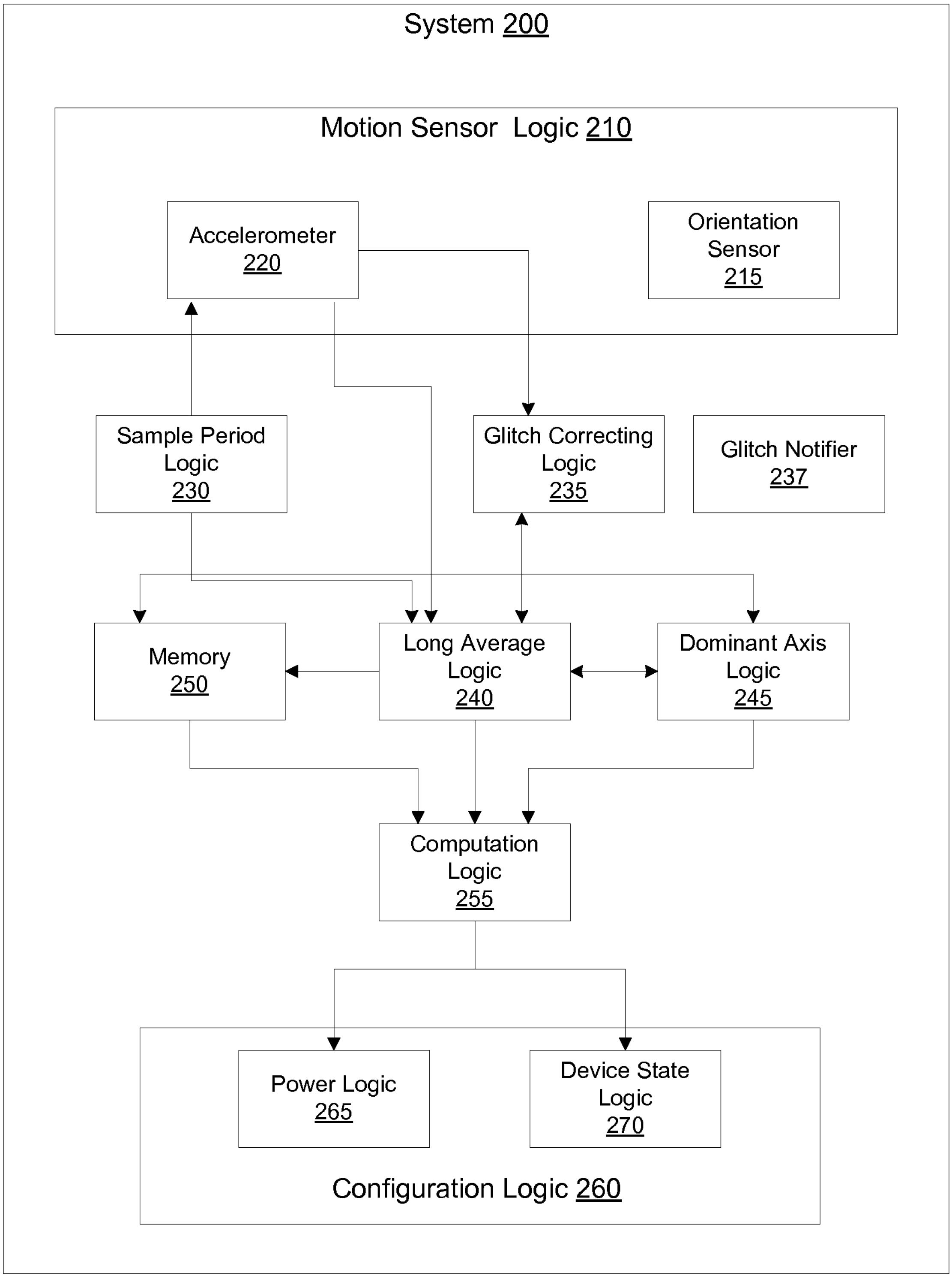

System 200
Motion Sensor Logic 210
Accelerometer 220
Orientation Sensor 215
Sample Period Logic 230
Glitch Correcting Logic 235
Glitch Notifier 237
Memory 250
Long Average Logic 240
Dominant Axis Logic 245
Computation Logic 255
Power Logic 265
Device State Logic 270
Configuration Logic 260

Figure 2

300

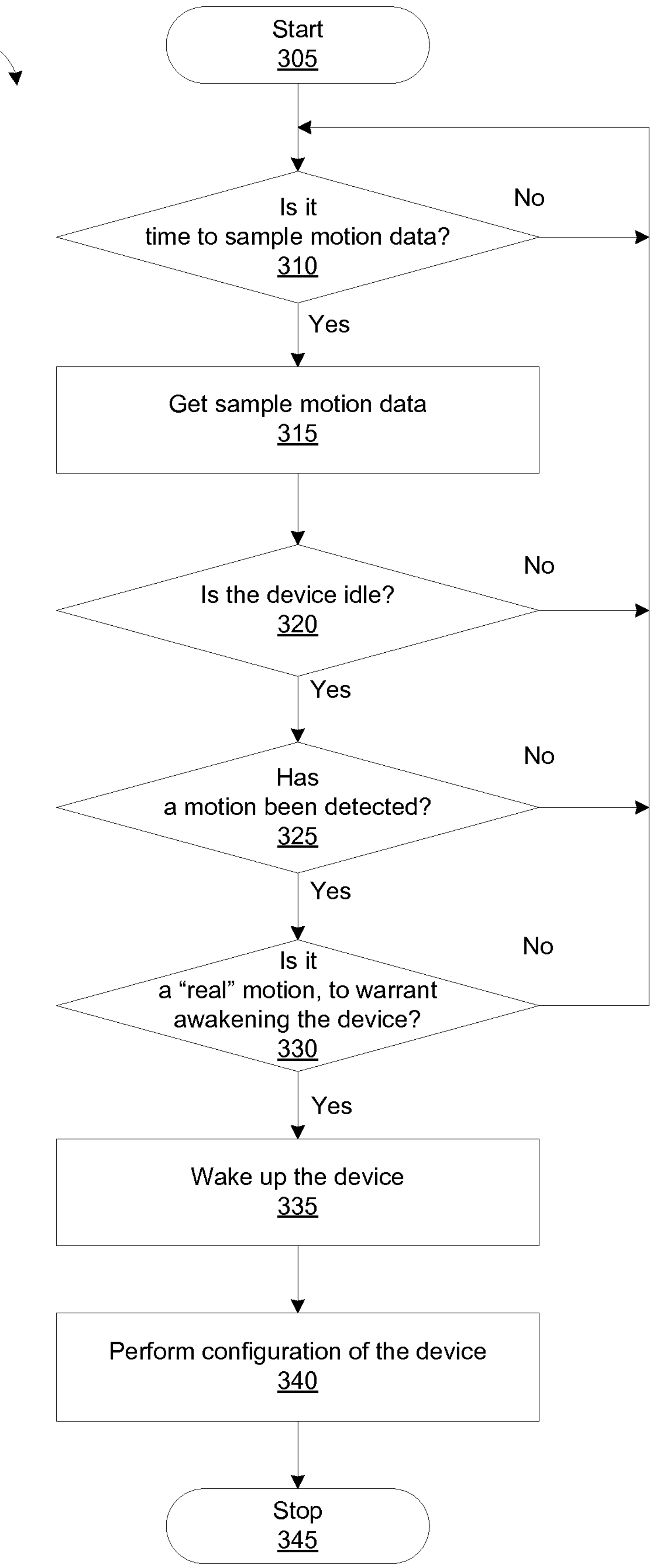

Start
305
Is it
time to sample motion data?
310
No
Yes
Get sample motion data
315
Is the device idle?
320
No
Yes
Has
a motion been detected?
325
No
Yes
Is it
a “real” motion, to warrant
awakening the device?
330
No
Yes
Wake up the device
335
Perform configuration of the device
340
Stop
345

Figure 3

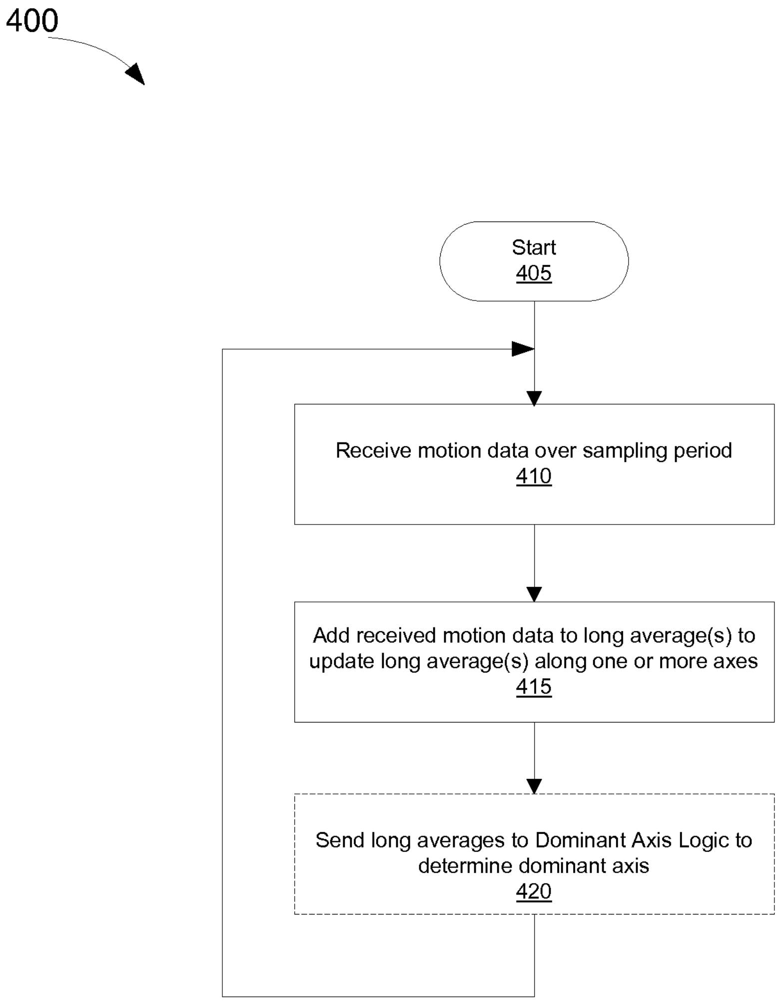

400
Start
405
Receive motion data over sampling period
410
Add received motion data to long average(s) to update long average(s) along one or more axes
415
Send long averages to Dominant Axis Logic to determine dominant axis
420

Figure 4

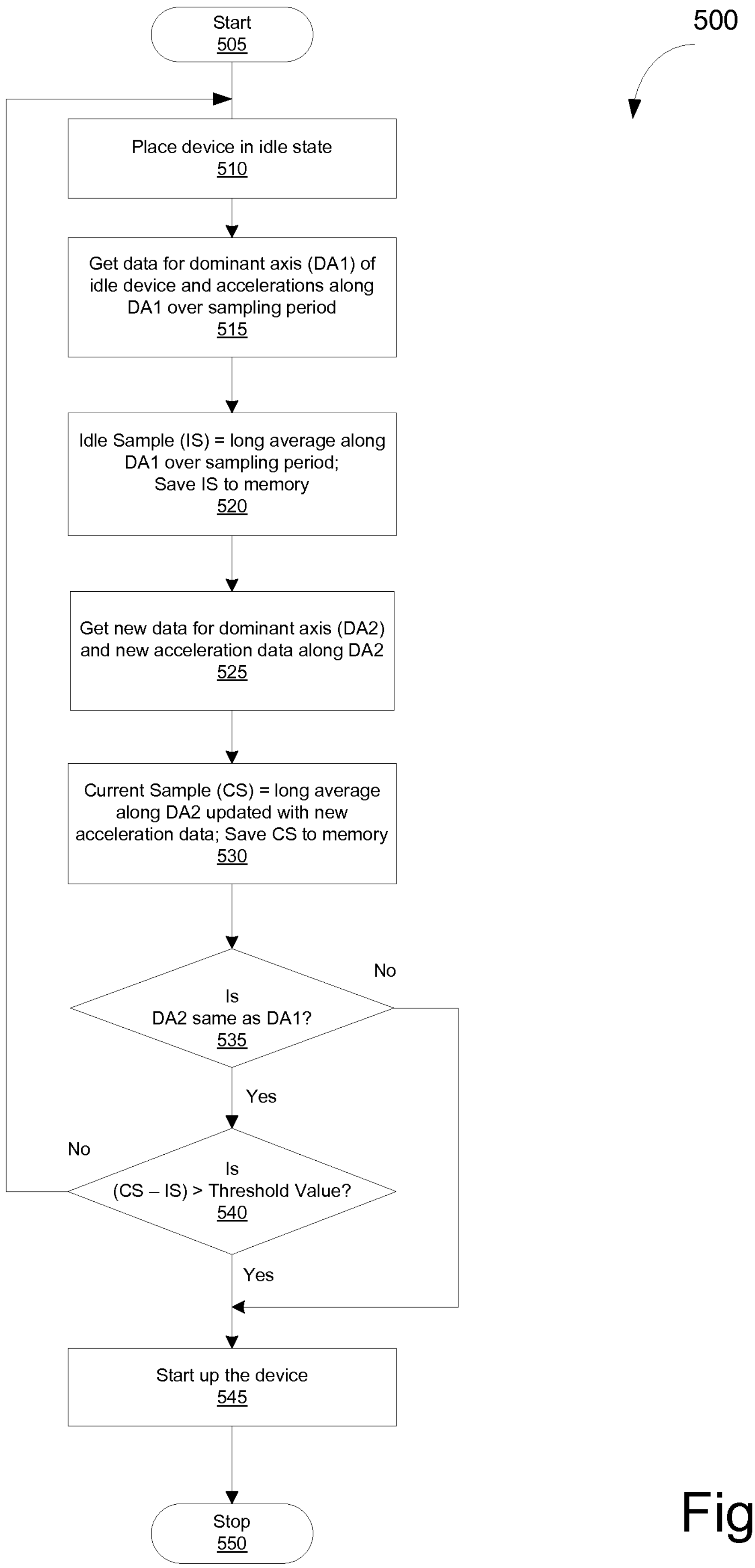

500
Start
505
Place device in idle state
510
Get data for dominant axis (DA1) of idle device and accelerations along DA1 over sampling period
515
Idle Sample (IS) = long average along DA1 over sampling period;
Save IS to memory
520
Get new data for dominant axis (DA2) and new acceleration data along DA2
525
Current Sample (CS) = long average along DA2 updated with new acceleration data; Save CS to memory
530
Is
DA2 same as DA1?
535
No
Yes
Is
(CS – IS) > Threshold Value?
540
No
Yes
Start up the device
545
Stop
550

Figure 5

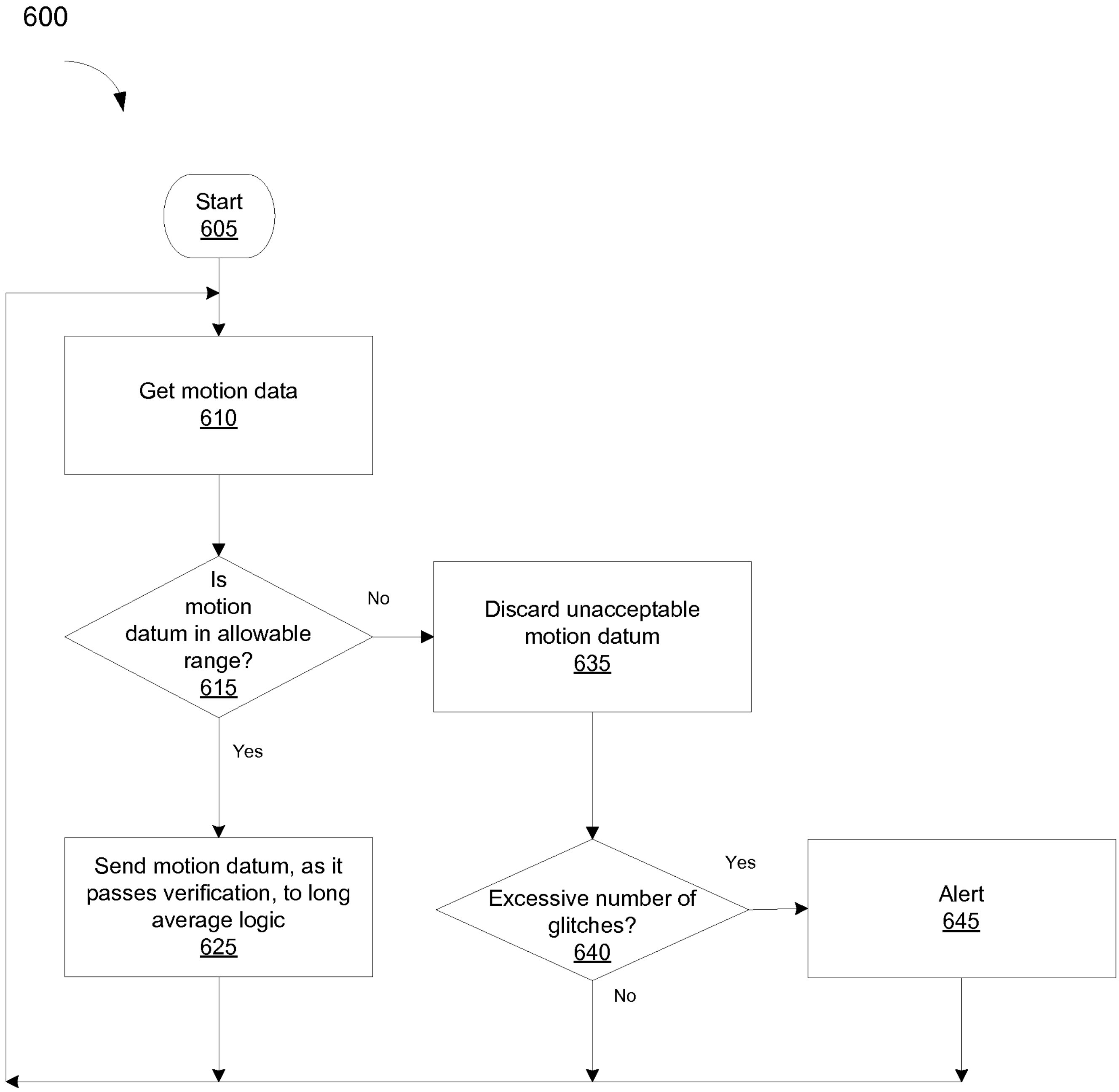

600
Start
605
Get motion data
610
Is
motion
datum in allowable
range?
615
No
Discard unacceptable
motion datum
635
Yes
Send motion datum, as it
passes verification, to long
average logic
625
Yes
Excessive number of
glitches?
640
Alert
645
No

Figure 6

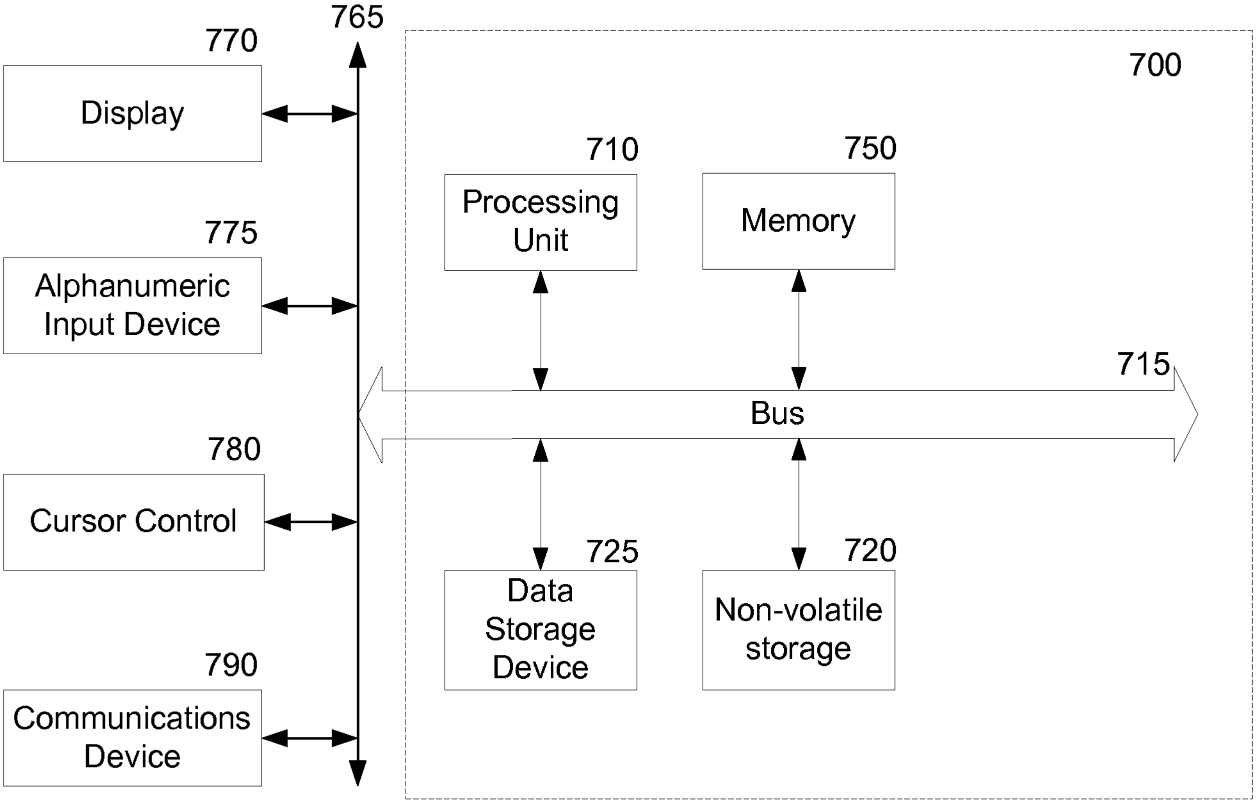

765
770
Display
775
Alphanumeric Input Device
780
Cursor Control
790
Communications Device
700
710
Processing Unit
750
Memory
715
Bus
725
Data Storage Device
720
Non-volatile storage

Figure 7