Sensor Fusion Algorithm

BACKGROUND

[0001] Mobile computing devices have been developed to increase the functionality that is made available to users in a mobile setting. For example, a user may interact with a mobile phone, tablet computer, or other mobile computing device to check email, surf the web, compose texts, interact with applications, and so on. Some mobile computing devices may connect to and interact with various accessory devices to provide different input techniques, extend functionality, and so forth. One challenge that faces developers of mobile computing devices is managing behaviors and interaction with accessory devices. For instance, a host computing device may have limited control over how an accessory device behaves and thus actions of the accessory may sometimes interfere with operation of the host computing device. Moreover, the user experience may be adversely affected by accessory devices that do not respond in a manner that is consistent with the host computing device. Thus, integrated management of behaviors and interaction for accessory devices may be a challenging consideration for developers of mobile computing devices.

SUMMARY

[0002] Sensor fusion algorithm techniques are described. In one or more embodiments, behaviors of a host device and accessory devices are controlled based upon an orientation of the host device and accessory devices, relative to one another. A combined spatial position and/or orientation for the host device may be obtained based on raw measurements that are obtained from at least two different types of sensors. In addition, a spatial position and/or orientation for an accessory device is ascertained using one or more sensors of the accessory device. An orientation (or position) of the accessory device relative to the host computing device may then be computed based on the combined spatial position/orientation for the host computing device and the ascertained spatial position/orientation for the accessory device. The relative orientation that is computed may then be used in various ways to control behaviors of the host computing device and/or accessory device.

[0003] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not

intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The detailed description is described with reference to the accompanying figures. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The use of the same reference numbers in different instances in the description and the figures may indicate similar or identical items. Entities represented in the figures may be indicative of one or more entities and thus reference may be made interchangeably to single or plural forms of the entities in the discussion.

[0005] FIG. 1 is an illustration of an environment in an example implementation that is operable to employ the techniques described herein.

[0006] FIG. 2 depicts an example implementation of a computing device of FIG. 1 in greater detail.

[0007] FIG. 3 depicts an example implementation of an accessory device of FIG. 1 as showing a flexible hinge in greater detail.

[0008] FIG. 4 depicts an example orientation of the accessory device in relation to the computing device in accordance with one or more embodiments.

[0009] FIG. 5 depicts an example orientation of the accessory device in relation to the computing device in accordance with one or more embodiments.

[0010] FIG. 6 depicts an example orientation of the accessory device in relation to the computing device in accordance with one or more embodiments.

[0011] FIG. 7 depicts an example orientation of the accessory device in relation to the computing device in accordance with one or more embodiments.

[0012] FIG. 8 depicts an example orientation of the accessory device in relation to the computing device in accordance with one or more embodiments.

[0013] FIG. 9 depicts an example orientation of the accessory device in relation to the computing device in accordance with one or more embodiments.

[0014] FIG. 10 depicts illustrates some example rotational orientations of the computing device in relation to the input device in accordance with one or more embodiments.

[0015] FIG. 11 is a flow diagram that describes an example procedure in accordance with one or more embodiments.

[0016] FIG. 12 is a flow diagram that describes an example procedure in accordance with one or more embodiments.

[0017] FIG. 13 illustrates an example system including various components of an example device that can be implemented as any type of computing device as described with reference to FIGS. 1-12 to implement embodiments of the techniques described herein.

DETAILED DESCRIPTION

Overview

[0018] Traditionally, a host computing device may have limited control over how an associated accessory device behaves. Thus actions of the accessory may sometimes interfere with operation of the host computing device, which may detract from the user experience. Accordingly, integrated management of behaviors and interaction for accessory devices may be a consideration for developers of mobile computing devices.

[0019] Sensor fusion algorithm techniques are described. In one or more embodiments, behaviors of a host device and accessory devices are controlled based upon an orientation of the host device and accessory devices, relative to one another. A combined spatial position and/or orientation for the host device may be obtained based on raw measurements that are obtained from at least two different types of sensors. In addition, a spatial position and/or orientation for an accessory device is ascertained using one or more sensors of the accessory device. An orientation (or position) of the accessory device relative to the host computing device may then be computed based on the combined spatial position/orientation for the host computing device and the ascertained spatial position/orientation for the accessory device. The relative orientation that is computed may then be used in various ways to control behaviors of the host computing device and/or accessory device.

[0020] In the following discussion, an example environment and devices are first described that may employ the techniques described herein. Example procedures are then described which may be performed in the example environment and by the devices as well as in other environments and by other devices. Consequently, performance of the example procedures is not limited to the example environment/devices and the example environment/devices are not limited to performance of the example procedures.

Example Operating Environment

[0021] FIG. 1 is an illustration of an environment 100 in an example implementation that is operable to employ the techniques described herein. The illustrated environment 100 includes an example of a computing device 102 that is physically and communicatively coupled to an accessory device 104 via a flexible hinge 106. The computing device 102 may be configured in a variety of ways. For example, the computing device 102 may be

configured for mobile use, such as a mobile phone, a tablet computer as illustrated, and so on. Thus, the computing device 102 may range from full resource devices with substantial memory and processor resources to a low-resource device with limited memory and/or processing resources. The computing device 102 may also relate to software that causes the computing device 102 to perform one or more operations.

[0022] The computing device 102, for instance, is illustrated as including an input/output module 108. The input/output module 108 is representative of functionality relating to processing of inputs and rendering outputs of the computing device 102. A variety of different inputs may be processed by the input/output module 108, such as inputs relating to functions that correspond to keys of the input device, keys of a virtual keyboard displayed by the display device 110 to identify gestures and cause operations to be performed that correspond to the gestures that may be recognized through the accessory device 104 and/or touchscreen functionality of the display device 110, and so forth. Thus, the input/output module 108 may support a variety of different input techniques by recognizing and leveraging a division between types of inputs including key presses, gestures, and so on.

[0023] In the illustrated example, the accessory device 104 is a device configured as a keyboard having a QWERTY arrangement of keys although other arrangements of keys are also contemplated. Further, other non-conventional configurations for an accessory device 104 are also contemplated, such as a game controller, configuration to mimic a musical instrument, a power adapter, and so forth. Thus, the accessory device 104 may assume a variety of different configurations to support a variety of different functionality. Different accessory devices may be connected to the computing device at different times. Moreover, functionally of a particular accessory device may also be adapted to assume different configurations and capabilities, such as through different selectable modes, software/firmware updates, modular add-on devices/components, and so forth. This may cause changes in the way keys or other controls for an accessory are laid-out and also change the way on which inputs from the accessory are handled by the host and applications. For example, an accessory device may be operable as keyboard and as a game controller by adaptively switching the kinds of keys/controls, displayed labels, and positions of controls to assume different configurations at different times.

[0024] As previously described, the accessory device 104 is physically and communicatively coupled to the computing device 102 in this example through use of a flexible hinge 106. The flexible hinge 106 represents one illustrative example of an

interface that is suitable to connect and/or attach and accessory device to a host computing device 102. The flexible hinge 106 is flexible in that rotational movement supported by the hinge is achieved through flexing (e.g., bending) of the material forming the hinge as opposed to mechanical rotation as supported by a pin, although that embodiment is also contemplated. Further, this flexible rotation may be configured to support movement in one direction (e.g., vertically in the figure) yet restrict movement in other directions, such as lateral movement of the accessory device 104 in relation to the computing device 102. This may be used to support consistent alignment of the accessory device 104 in relation to the computing device 102, such as to align sensors used to change power states, application states, and so on.

[0025] The flexible hinge 106, for instance, may be formed using one or more layers of fabric and include conductors formed as flexible traces to communicatively couple the accessory device 104 to the computing device 102 and vice versa. This communication, for instance, may be used to communicate a result of a key press to the computing device 102, receive power from the computing device, perform authentication, provide supplemental power to the computing device 102, and so on. The flexible hinge 106 or other interface may be configured in a variety of ways to support multiple different accessory devices 104, further discussion of which may be found in relation to the following figure.

[0026] As further illustrated in FIG. 1 the computing device 102 may include various applications 112 that provide different functionality to the device. A variety of applications 112 typically associated with computing devices are contemplated including, but not limited to, an operating system, a productivity suite that integrates multiple office productivity modules, a web browser, games, a multi-media player, a word processor, a spreadsheet program, a photo manager, and so forth. The computing device 102 further includes multiple host sensors 114 that are configured to sense corresponding inputs responsive to manipulation of the computing device 102. Likewise, the accessory device 104 includes one or more accessory sensors 116 that are configured to sense corresponding inputs generated responsive to manipulation of the accessory device 104.

[0027] In accordance with techniques described herein, input obtained from the host sensors 114 and accessory sensors 116 may be processed and/or combined according to a suitable sensor fusion algorithm to resolve an orientation of the accessory device 104 and computing device 102 one to another. In general, input regarding position and/or orientation from multiple different types of sensors is processed in combination to

compute the orientation. The computed orientation may then be used to control behaviors of the host and accessory and perform various corresponding operations. A variety of different types of sensors and algorithms suitable to resolve the orientation may be employed as discussed in greater detail in relation to the following figures.

[0028] To further illustrate, consider FIG. 2 which depicts generally at 200 an example computing device 102 of FIG. 1 in greater detail. In the depicted example, the computing device 102 is shown in a stand-alone configuration without an accessory device 104 being attached. In addition to the components discussed in relation to FIG. 1, the example computing device of FIG. 2 further includes a processing system 202 and computer-readable media 204 that are representative of various different types and combinations of processing components, media, memory, and storage components and/or devices that may be associated with a computing device and employed to provide a wide range of device functionality. In at least some embodiments, the processing system 202 and computer-readable media 204 represent processing power and memory/storage that may be employed for general purpose computing operations. More generally, the computing device 102 may be configured as any suitable computing system and/or device that employ various processing systems and computer-readable media, additional details and examples of which are discussed in relation to the example computing system of Fig. 13.

[0029] The computing device 102 may also implement selected device functionality through one or more microcontrollers 206. The microcontrollers 206 represent hardware devices/systems that are designed to perform a predefined set of designated tasks. The microcontrollers 206 may represent respective on-chip systems/circuits having self-contained resources such as processing components, I/O devices/peripherals, various types of memory (ROM, RAM, Flash, EEPROM), programmable logic, and so forth. Different microcontrollers may be configured to provide different embedded applications/functionality that are implemented at least partially in hardware and perform corresponding tasks. The microcontrollers 206 enable performance of some tasks outside of operation of a general purpose processing system and other applications/components of the computing device or accessory device. Generally, power consumption of the microcontrollers is low in comparison with operating a general purpose processing system for a device.

[0030] As further depicted, the computing device 102 may further include a sensor fusion module 208, a behavior module 210, and a sensor fusion application programming interface (API) 212 to implement aspects of sensor fusion algorithm techniques described

herein. The sensor fusion module 208 generally represents functionality to apply a suitable sensor fusion algorithm as described above and below to derive an orientation that is based on input from multiple sensors. The sensor fusion module 208 may operate to collect inputs regarding positions/orientation/etc. supplied via the various sensors, process the inputs, and compute a corresponding orientation that describe the spatial relationship of the computing device 102 and an accessory device 104.

[0031] The behavior module 210 represents functionality to control and/or modify a variety of different behaviors associated with the computing device 102 and/or accessory devices 104 based on the computed orientation. This may include but is not limited to managing power states/consumption, selecting operational modes or device states, adjusting sensitivity of one or more sensors, controlling interaction between the host, accessory, and/or peripheral devices, modifying device functionality, enabling/disabling network connections, activating/deactivating applications, and/or setting application states, to name a few examples. These and other examples of behaviors that may be controlled based on a computed orientation are described in greater detail in relation to the example procedures discussed herein below.

[0032] The sensor fusion application programming interface (API) 212 represents functionality to expose information regarding the computer orientation for use by applications 112. For example, applications 112 may utilize the sensor fusion API to request orientation information on demand and/or subscribe to orientation updates from the sensor fusion module 208 and/or an associated notification system. The sensor fusion API may then interact with the sensor fusion module 208 on behalf of the application 112 to cause orientation information to be conveyed to the application 112. Applications 112 may use orientation information in various ways, example of which may be found in the discussion of an example procedure 1200 of FIG. 12 below.

[0033] As previously mentioned, various different types of sensors may be employed to implement the techniques described herein. A host computing device may include an array of sensors used to provide orientation information. By way of example and not limitation, the host sensors 114 for the example computing device 102 of FIG. 2 are depicted as including a gyroscope 214, an accelerometer 216, a magnetometer 218, and a Hall Effect sensor 220. Various other sensors 222 suitable to derive information regarding the position and/or orientation may also be employed.

[0034] FIG. 3 depicts an example implementation 300 of the accessory device 104 of FIG. 1 as showing the flexible hinge 106 in greater detail. In this example, the accessory device

104 is depicted as being detached from the computing device. Here, a connection portion 302 of the input device is shown that is configured to provide a communicative and physical connection between the accessory device 104 and the computing device 102. In this example, the connection portion 302 has a height and cross section configured to be received in a channel in the housing of the computing device 102, although this arrangement may also be reversed without departing from the spirit and scope thereof. The connection portion 302 provides an interface through which attachment/connection of the accessory device 104 to the computing device may be detected. In at least some embodiments, this interface also enables communications for interaction and/or control of the accessory device 104 as described herein. For example, the computing device 102, sensor fusion module 208, and/or behavior module 210 may communicate with the accessory device through the interface to obtain input from various accessory sensors 116 and to direct behaviors of the accessory device.

[0035] The connection portion 302 is flexibly connected to a portion of the accessory device 104 that includes the keys through use of the flexible hinge 106. Thus, when the connection portion 302 is physically connected to the computing device the combination of the connection portion 302 and the flexible hinge 106 supports movement of the accessory device 104 in relation to the computing device 102 that is similar to a hinge of a book. Naturally, a variety of orientations may be supported some examples of which are described in the following section.

[0036] The connecting portion 302 is illustrated in this example as including magnetic coupling devices 304, 306, mechanical coupling protrusions 308, 310, and a plurality of communication contacts 312. The magnetic coupling devices 304, 306 are configured to magnetically couple to complementary magnetic coupling devices of the computing device 102 through use of one or more magnets. In this way, the accessory device 104 may be physically secured to the computing device 102 through use of magnetic attraction. The connecting portion 302 also includes mechanical coupling protrusions 308, 310 to form a mechanical physical connection between the accessory device 104 and the computing device 102. The communication contacts 212 are configured to contact corresponding communication contacts of the computing device 102 to form a communicative coupling between the devices to facilitate various kinds of communications.

[0037] Having discussed an example environment in which embodiments may operate, consider now some example device orientations in accordance with one or more embodiments.

Example Device Orientations

[0038] The following discussion presents some example device orientations. As detailed, different device orientations can be associated with different device power states, different application states, trigger different behaviors, and so forth. The example orientations as well as other orientations may be determined using sensor fusion algorithm techniques described above and below. A determined orientation may then be used to drive different behaviors for the host and/or the accessory.

[0039] FIG. 4 illustrates that the accessory device 104 may be rotated such that the accessory device 104 is placed against the display device 110 of the computing device 102 to assume an orientation 400. In the orientation 400, the accessory device 104 may act as a cover such that the accessory device 104 can protect the display device 110 from harm. In implementations, the orientation 400 can correspond to a closed position of the computing device 102.

[0040] FIG. 5 illustrates that the input device 104 has rotated away from the computing device 102 such that the computing device assumes an orientation 500. The orientation 400 includes a gap 502 that is introduced between the computing device 102 and the accessory device 104. In implementations, the orientation 500 can be caused unintentionally by a user, such as by inadvertent contact with the computing device 102 and/or the accessory device 104 that causes the computing device 102 to sag slightly away from the accessory device 104 such that the gap 502 is introduced.

[0041] FIG. 6 illustrates an example orientation 600 of the computing device 102. In the orientation 600, the accessory device 104 is laid flat against a surface and the computing device 102 is disposed at an angle to permit viewing of the display device 110, e.g., such as through use of a kickstand 602 disposed on a rear surface of the computing device 102. The orientation 600 can correspond to a typing arrangement whereby input can be received via the accessory device 104, such as using keys of a keyboard, a track pad, and so forth.

[0042] FIG. 7 illustrates a further example orientation of the computing device 102, generally at 700. In the orientation 700, the computing device 102 is oriented such that the display device 110 faces away from the accessory device 104. In this example, the kickstand 602 can support the computing device 102, such as via contact with a back

surface of the accessory device 104. Although not expressly illustrated here, a cover can be employed to cover and protect a front surface of the accessory device 104. In the depicted orientation, an angle 702 between the device and host is established. Various different angles corresponding to different positions/orientation may be established, as discussed above and below.

[0043] FIG. 8 illustrates an example orientation 800, in which the accessory device 104 may also be rotated so as to be disposed against a back of the computing device 102, e.g., against a rear housing of the computing device 102 that is disposed opposite the display device 110 on the computing device 102. In this example, through orientation of the connection portion 202 to the computing device 102, the flexible hinge 106 is caused to "wrap around" the connection portion 202 to position the accessory device 104 at the rear of the computing device 102.

[0044] This wrapping causes a portion of a rear of the computing device 102 to remain exposed. This may be leveraged for a variety of functionality, such as to permit a camera 802 positioned on the rear of the computing device 102 to be used even though a significant portion of the rear of the computing device 102 is covered by the accessory device 104 in the example orientation 800. Further to the example illustrated in FIG. 8, the display device 110 of the computing device 102 may be determined to be oriented at an angle 804 relative to the accessory device 104. In general, the angle 804 may change as the accessory device 104 is manipulated into different positions. For example, the angle 804 as shown in FIG.8 can be determined to be approximately 360 degrees. Other orientations may correspond to other angles, and angle ranges may be established and associated with defined modes or states that may trigger different behaviors. Thus, behaviors may be controlled based on the particular mode/state that correspond to the current angle between the host and accessory.

[0045] FIG. 9 illustrates a further example orientation of the computing device 102, generally at 900. In the orientation 900, the computing device 102 is rotated sideways, e.g., in a portrait orientation relative to a surface 902 on which the computing device 102 is disposed. The display device 110 is visible, with the accessory device 104 rotated away from the display device 110. In at least some implementations, a width of the accessory device 104 can be narrower than a width of the computing device 102. Additionally or alternatively, the width of the accessory device 104 can be tapered such that the edge closest to the hinge 106 is wider than the outermost edge. This can enable the face of the

display device 110 to recline back in the orientation 900, to provide for a suitable viewing angle.

[0046] FIG. 10 illustrates that the computing device 102 may be rotated within a variety of different angle ranges with respect to the accessory device 104. As detailed herein, different angle ranges can be associated with different power states, different application states, and so on.

[0047] An angle range 1000 is illustrated, which corresponds to a closed position for the computing device 102. Thus, if the computing device 102 is positioned at an angle within the angle range 1000 relative to the accessory device 104, the computing device 102 can be determined to be in a closed position. A closed position can include an associated closed state where various functionalities/behaviors for the computing device 102 and accessory device 104 can be modified accordingly based on the closed state.

[0048] Further illustrated is an angle range 1002, which may correspond to a typing orientation for the computing device 102. Thus, if the computing device 102 is positioned at an angle within the angle range 1002 relative to the accessory device 104, the computing device 102 can be determined to be in a typing orientation. Within this orientation, the computing device 102 and/or the accessory device 104 can placed in a typing power state where functionalities/behaviors for the computing device 102 and accessory device 104 can be customized accordingly based on the typing state.

[0049] FIG. 10 further illustrates an angle range 1004, which corresponds to a viewing position for the computing device 102. Thus, if the computing device 102 is positioned at an angle within the angle range 1004 relative to the accessory device 104, the computing device 102 can be determined to be in a viewing orientation. In this orientation, functionalities/behaviors for the computing device 102 and accessory device 104 can be controlled accordingly based on the viewing state.

[0050] The orientations, angle ranges, power states, and so forth discussed above are presented for purposes of illustration only. It is contemplated that a wide variety of different orientations, device states, and angle ranges may be implemented within the spirit and scope of the claimed embodiments.

[0051] Having discussed some example device orientations, consider now some example procedures in accordance with one or more embodiments.

Example Procedures

[0052] The following discussion describes sensor fusion algorithm techniques that may be implemented utilizing the previously described systems and devices. Aspects of each of

the procedures may be implemented in hardware, firmware, software, or a combination thereof. The procedures are shown as a set of blocks that specify operations performed by one or more devices and are not necessarily limited to the orders shown for performing the operations by the respective blocks. In portions of the following discussion, reference may be made to the example operating environment 100 of FIG.1, the example devices of FIGS. 2-3, and the example orientation shown in FIGS. 4-10, respectively.

[0053] FIG. 11 depicts an example procedure 1100 in which an orientation of an accessory relative to a host is computed. In at least some embodiments, the procedure may be performed by a suitably configured computing device, such as the example computing device 102 of FIG. 2 that includes or otherwise make use of a sensor fusion module 208 and/or behavior module 210.

[0054] Raw spatial positions for a host computing device are calculated independently using at least two different types of sensors (block 1102). The raw spatial positions are processed to obtain a combined spatial position for the host computing device (block 1104).

[0055] For example, the sensor fusion module 208 may be configured to implement a designated sensor fusion algorithm. Generally, the sensor fusion algorithm is configured to aggregate information from an array of different kinds of host sensors 114 employed by a computing device 102. The aggregation of multiple different sensing techniques and types of sensors may provide improved resolution of positions and may smooth errors that may be introduced by individual techniques and sensors. In at least some embodiments, the sensor fusion algorithm is configured to calculate at least two independent computations of the raw spatial position of the computing device 102 using different respective sensors. Multiple independent computations of the raw position may then be used to produce a combined spatial position. Each of the independent computations may employ one or more of the various types of host sensors 114 described above and below. At least some of the sensors used for different independent computations are of different types. Thus, the sensor fusion algorithm obtains input from a variety of different host sensors 114 and combines this information to resolve the position of the computing device 102.

[0056] In one approach, the computing device 102 includes a gyroscope 214 that may be used to obtain one of the independent computations of the raw position. Generally, a gyroscope uses principles of angular momentum to calculate orientation and rotation. The gyroscope 214 can be used to recognize movement within three-dimensional space

and may enable determination of position with respect to a reference object/point, such as the earth. Using input obtained from the gyroscope 214, the sensor fusion module 208 may operate to compute a raw spatial position for the computing device. The raw spatial position may be expressed as coordinates in a three dimensional coordinate system defined with x, y, and z axes relative to the reference object/point (e.g., the earth).

[0057] In particular, the angular velocity input obtained from the gyroscope can be processed to determine angular positioning of the computing device. Initially, the input from the gyroscope may be filtered to remove a low pass constant offset of the gyroscope. Such a low pass constant offset may be created if the gyroscope is stuck in a non-zero position and is removed to prevent inaccuracy in the computation. The algorithm may integrate over multiple axes of the gyroscope (e.g., x, y, and z axes) to obtain a transform that describes a raw spatial position for the computing device. This processing may involve integrating angular velocity input from the gyroscope through a Runge-Kutta integration algorithm (or other suitable algorithm) to obtain corresponding impulse data. The impulse data may be expressed as quaternions for the different axes, which when multiplied together produce a quaternion that describes a transformation between the computing device 102 and the earth (or other selected reference object/point) with respect to their respective axes/coordinate systems. This provides one independent version of the raw spatial position for the computing device 102.

[0058] Another independent computation of the raw spatial position may be obtained using an accelerometer 216 and a magnetometer 218 in combination. Here, the accelerometer 216 is configured as a three axes accelerometer that may be employed to derive two of the degrees of freedom of the device (e.g., position with respect to the x-axis and y-axis). In the low pass, the vector of acceleration is approximately 1g down pointing to the center of the earth. The components of acceleration measured via the accelerometer 216 may be obtained as distributed across each of the three axes. The components of acceleration can in turn be used to compute angles of the accelerometer /device axes with respect to the low pass vector that points to the center of the earth. This provides two of the three degrees of freedom with respect to tilt or orientation of the device. In particular, the accelerometer processing just described is used to resolve the tilt/orientation of the x-axis and y-axis of the computing device 102.

[0059] Now, the magnetometer 218 may be employed to resolve the remaining degree of freedom with respect to tilt/orientation of the device. The magnetometer 218 may be initialized/configured to act like a compass. In this approach, the magnetometer 218 can

be used to compute a vector that is parallel to the ground (e.g., the earth's surface). This vector points to magnetic north and can be used to determine rotation of the device with respect to the z-axis. Now, the tilt/orientation of the x-axis and y-axis from the accelerometer and the rotation of the device with respect to the z-axis from the magnetometer 218 may be used to construct another quaternion that describes a transformation between the computing device 102 and the earth (or other selected reference object/point) with respect to their respective axes/coordinate systems. This provides another independent way in which a raw spatial position for the computing device 102 may be obtained. Other examples using different sensors and combination of sensors are contemplated. For example, a global positioning satellite (GPS) radio may be used to provide some positioning data that may be used alone or in combination with other kinds of sensor data to compute the position/orientation of the computing device 102.

[0060] Accordingly, at least two different results for the raw spatial position are computed using the foregoing example techniques or other suitable techniques. The sensor fusion algorithm may be further configured to combine multiple independent computations of raw spatial position in various ways. The combining generally involves interpolating between two or more raw spatial positions to reduce or eliminate inaccuracies and/or smooth the results. The interpolation produces a combined spatial position for the computing device that is based on two or more independently obtained raw spatial positions.

[0061] By way of example and not limitation, results obtained using a gyroscope may be more precise in the short term relative to other sensors and position determination techniques. However, small integration errors associated with the gyroscope computations may build up over time creating an increasingly larger offset that may result in inaccurate results in the long term. Thus, interpolating the gyroscope results with other independently obtained results can effectively adjust for expected integration errors in the gyroscope results. In one approach, a normalized linear interpolation is employed that may be biased towards the gyroscope results since these results are initially more precise and subject to less noise. Other independent results, such as the results from the accelerometer/magnetometer, may be included in the interpolation to keep the gyroscope results in check and slowly adjust the bias for the combined result away from the gyroscope results and towards the other results over time. This produces a mathematically smooth transformation as the combined result.

[0062] A spatial position for an accessory device connected to the host computing device is ascertained using one or more sensors of the accessory device (block 1106). The spatial position for the accessory device 104 may be computed in any suitable way, including but not limited to the techniques described in relation to the computing device 102. Accessory sensors 116 for different accessories may include any of the various types of sensors described herein. Accordingly, different corresponding techniques may be used to ascertain spatial position of the accessory based on appropriate input from one or more accessory sensors 116. Different techniques may also be employed for different accessories based on the types of sensors that are included with the accessory. In general, the sensor fusion module 206 may be configured to obtain input from different sensors of the accessory over a suitable interface with the accessory and compute a corresponding spatial position based on the input.

[0063] In one particular example, the sensor fusion module 206 may compute a spatial position using an accelerometer 216 associated with the accessory device 104. In this approach, the accelerometer 216 may be employed to resolve the tilt/orientation with respect to the x-axis and y-axis of the accessory device 104. This may occur in a manner that is comparable to the computation of the same kind of information for the computing device 102 using an associated accelerometer as described above.

[0064] In some arrangements, the accessory device 104 may be configured to connect to the computing device 102 using a connection portion 302 that is connectable to an interface of the computing device via a known location. For instance, in the hinge example previously described, at least some information regarding the position of the accessory device may be established based upon the known location and nature of the connection to the host device. Thus, it may be sufficient to use the two degrees of freedom (e.g., x-axis and y-axis position/pitch and roll) for the accessory device 104 in such cases to resolve the position of the accessory relative to the host. It should be noted though that rotation with respect the z-axis may also be computed for the accessory device 104 in some embodiments, using a magnetometer 218 as discussed previously or using other sensors and techniques. This may be employed in configurations in which an accessory may still be manipulated in three dimensions even when connected to a host device, such as by way of a ball and socket type connection.

[0065] An orientation of the accessory device relative to the host computing device is computed based on the combined spatial position for the host computing device and the ascertained spatial position for the accessory device (block 1108). The computed

orientation may correspond to any of the different orientations discussed in relation to FIGS 4-10 as wells as other possible orientations. Here, a comparison can be made between the combined spatial position for the computing device 102 and the ascertained spatial position of the accessory device 104 to derive information regarding the orientation of the device one to another. In particular, the combined spatial position indicates a transformation between how axes in a coordinate system for the computing device 102 are oriented relative to axes associated with a reference coordinate system for the earth or other reference. Similarly, the ascertained spatial position of the accessory device 104 indicates a transformation between how axes in a coordinate system for the accessory device are oriented relative to axes of the reference coordinate system. Accordingly, these two positions may be used to compute a transformation of the accessory device 104 relative to the computing device 102 that is independent of the reference coordinate system.

[0066] By way of example, in some cases, the orientation may be defined as an angle of the accessory device 104 with respect the computing device 102 as represented in FIG. 10. As also discussed previously, different angles may be associated with different interaction states, such as the closed state, typing state, and viewing state examples given above. The orientation may alternatively be expressed in another suitable manner, such as using x, y, z coordinates.

[0067] Optionally, the computed orientation may be verified using a Hall Effect sensor 220 of the computing device 102. The Hall Effect sensor 220 may be configured to utilize magnetic force to detect proximity between the computing device 102 and the accessory device 104. For example, the Hall Effect sensor 220 may measure proximity based upon one or more magnets that are included with the computing device 102 and/or the accessory device 104. When the computing device 102 is rotated to a closed position, the Hall Effect sensor 220 may be configured to align with and detect a magnet of the accessory device 104. When the computing device 102 is positioned away from the accessory device 104 in an open position, the Hall Effect sensor 220 may be unable to detect the magnet or the detected magnetic force may change as the computing device 102 is rotated at different angles relative to the accessory device 104. The Hall Effect sensor 220 provides another way in which the orientation may be determined. Thus, the Hall Effect sensor 220 may be used as an additional check on whether the orientation computed using other sensors is accurate. This additional check may be made before causing and/or

controlling some kinds of behaviors, such as powering down the devices or switching off different components based on orientation.

[0068] One or more behaviors of the host computing device and accessory device are controlled based on the orientation that is computed (block 1110). Various behaviors and responsive actions may be driven based on a computed orientation of an accessory with respect to the host. The behavior module 210 may be configured to obtain orientation results from the sensor fusion module 208 and control various behaviors accordingly.

[0069] Controlling the behaviors may include at least power management operations for the computing device 102 and/or host device. Generally, power management operations are configured to control power consumption and prolong battery life. For example, the behavior module 210 may cause changes in power modes/states to occur based on particular orientations. This may include toggling the devices and/or selected components on/off according to a determined orientation. For example, in a closed state both the host and accessory may be powered down or placed into a sleep mode. In another example, the accessory may be powered down when the orientation corresponds to a viewing state. The accessory device 104 may also automatically wake-up in particular orientation, such as when a typing state is detected. A variety of other power management examples are also contemplated that may occur in response to a computed orientation.

[0070] In another example, controlling the behaviors may include selectively adjusting and/or enabling/disabling different sensors for the device according to the orientation. By way of example, rotation of the accessory fully around to cover the backside of the host may be indicative of a game play state. In this arrangement, it may be likely that an accelerometer 216 may be used for gameplay whereas use of touch functionality for keyboard/typing input from the accessory may be unlikely. According, in this arrangement sensitivity of an accelerometer 216 may be increased/turned-on and touch sensitivity may be decreased or disabled. In a typing state, the opposite may be true and the accelerometer 216 may be disabled or adjusted to less sensitivity and the touch sensitivity may be increased or re-enabled. Thus, sensitivity of sensors may be adjusted and particular sensors may be turned on/off based on orientation. It should be noted that sensors that are controlled may include sensors involved in computation of the orientation as well as other sensors of the host or accessory.

[0071] In yet another example, functionality that is activated for the accessory and/or host may be modified based on the orientation. For example, an accessory may be configured to act as game controller when wrapped around to the backside and transform to provide

keyboard type inputs when in a typing orientation. In a further example, reading gestures to scroll or turn pages via the accessory may be enabled by input across the accessory device in a viewing orientation and may be disabled for other states/orientation. These kinds of changes in the functionality provided by an accessory may occur by selectively exposing, enabling, configuring or otherwise activating different controls, functions, and gestures according to different orientations.

[0072] Comparable changes to activate gestures, touch keys, and other functionality of the host computing device based on the orientation may also occur. For example, gestures for manipulation of media content on the display 110 may be active in some orientations (e.g., viewing state or gaming state) and deactivated in other scenarios. Some additional examples of modifications that may be made to functionality that is activated/available for the computing device based on orientation include selectively enabling/disabling network connections and/or controlling interactions of the host with accessory devices and/or peripheral devices (e.g., printers, streaming media devices, storage devices) based upon the computed orientation.

[0073] Additionally, behaviors of applications 112 may also be controlled based on a computed orientation. For example, the behavior module 210 may be configured to selectively activate or deactivate different applications 112 based on the orientation. This may include toggling between applications operating in foreground and background processes, launching and closing particular applications, minimizing/maximizing, and so forth. Applications 112 may also retrieve and/or subscribe to receive updates of computed orientation that the applications may make use of in various ways, some details of which are provided in relation to the following figure. Accordingly, a wide variety of behaviors may be controlled based on a computed orientation, of which the particular behaviors enumerated above are but as few illustrative examples.

[0074] FIG. 12 depicts an example procedure 1200 in which a computed orientation is exposed for use by applications. In at least some embodiments, the procedure may be performed by a suitably configured computing device, such as the example computing device 102 of FIG. 2 that includes or otherwise make use of a sensor fusion application programming interface (API) 212.

[0075] An orientation of an accessory device relative to a host computing device is computed based on a combined spatial position for the host computing device and an ascertained spatial position for the accessory device (block 1202). This may occur in

accordance with a designated sensor fusion algorithm as discussed in relation to the example procedure 1100 of FIG. 11 above.

[0076] An interface is exposed that is operable by one or more applications to obtain the computed orientation (block 1204). The computed orientation is supplied to an application in response to receiving a request from the application via the interface (block 1206). In particular, a computing device 102 may include a sensor fusion application programming interface (API) 212 that is operable to supply computed orientation information to applications 112. In one approach, the sensor fusion API may provide orientation information on demand responsive to individual requests. In addition or alternatively, the sensor fusion API may be configured to facilitate registration of applications 112 to subscribe to receive orientation updates. In response to a request to subscribe, the API may register an application with the sensor fusion module 208 and/or an associated notification system configured to supply notification messages to registered applications when orientation changes occur. The applications 112 may then receive notification messages sent via the notification system that describe updates to the orientation.

[0077] The sensor fusion API may supply the orientation and/or related information to application in various formats. For example, the orientation may be in the form of a transform of the accessory device 104 relative to the computing device 102 as computed in the manner described above. In this case, an application may process the supplied orientation information to obtain information in an appropriate format for the application, such as an orientation angle or a defined orientation state corresponding to the computed orientation. In addition or alternatively, the sensor fusion module 208 may operate to compute an orientation state on behalf of applications. Thus, information supplied via the sensor fusion API may include a state name or identifier that may be directly usable by the applications.

[0078] Applications 112 may make use of orientation information supplied through the API in various ways. For instance, an application 112 may selectively modify a user interface and/or functionality of the user interface for the application based on the orientation. This may include activating different controls, menus, gestures, and/or input modes for different respective orientations. For example, a navigation menu that appears in one orientation (typing/keyboard input orientation) may disappear in a viewing orientation. Further, an application 112 may be configured to include various modes and switch between the modes based on orientation. For example, a messaging application

may switch from a text input mode to a video mode in accordance with the computed orientation. In another example, the application may modify the manner in which particular inputs are interpreted in different orientations. For instance, a button press in a typing orientation may be used for alphanumeric entry whereas the same button may be used for content control functions in a viewing orientation. Other buttons, keys, and other controls may also be selectively enabled or disabled as the orientation changes. A variety of other examples are also contemplated.

[0079] Having considered the foregoing example procedures, consider now a discussion of example systems and devices that may be employed to implement aspects of techniques in one or more embodiments.

Example System and Device

[0080] FIG. 13 illustrates an example system generally at 1300 that includes an example computing device 1302 that is representative of one or more computing systems and/or devices that may implement the various techniques described herein. The computing device 1302 may be, for example, be configured to assume a mobile configuration through use of a housing formed and size to be grasped and carried by one or more hands of a user, illustrated examples of which include a mobile phone, mobile game and music device, and tablet computer although other examples are also contemplated.

[0081] The example computing device 1302 as illustrated includes a processing system 1304, one or more computer-readable media 1306, and one or more I/O interface 1308 that are communicatively coupled, one to another. Although not shown, the computing device 1302 may further include a system bus or other data and command transfer system that couples the various components, one to another. A system bus can include any one or combination of different bus structures, such as a memory bus or memory controller, a peripheral bus, a universal serial bus, and/or a processor or local bus that utilizes any of a variety of bus architectures. A variety of other examples are also contemplated, such as control and data lines.

[0082] The processing system 1304 is representative of functionality to perform one or more operations using hardware. Accordingly, the processing system 1304 is illustrated as including hardware element 1310 that may be configured as processors, functional blocks, and so forth. This may include implementation in hardware as an application specific integrated circuit or other logic device formed using one or more semiconductors. The hardware elements 1310 are not limited by the materials from which they are formed or the processing mechanisms employed therein. For example, processors may be comprised

of semiconductor(s) and/or transistors (e.g., electronic integrated circuits (ICs)). In such a context, processor-executable instructions may be electronically-executable instructions.

[0083] The computer-readable storage media 1306 is illustrated as including memory/storage 1312. The memory/storage 1312 represents memory/storage capacity associated with one or more computer-readable media. The memory/storage component 1312 may include volatile media (such as random access memory (RAM)) and/or nonvolatile media (such as read only memory (ROM), Flash memory, optical disks, magnetic disks, and so forth). The memory/storage component 1312 may include fixed media (e.g., RAM, ROM, a fixed hard drive, and so on) as well as removable media (e.g., Flash memory, a removable hard drive, an optical disc, and so forth). The computer-readable media 1306 may be configured in a variety of other ways as further described below.

[0084] Input/output interface(s) 1308 are representative of functionality to allow a user to enter commands and information to computing device 1302, and also allow information to be presented to the user and/or other components or devices using various input/output devices. Examples of input devices include a keyboard, a cursor control device (e.g., a mouse), a microphone, a scanner, touch functionality (e.g., capacitive or other sensors that are configured to detect physical touch), a camera (e.g., which may employ visible or non-visible wavelengths such as infrared frequencies to recognize movement as gestures that do not involve touch), and so forth. Examples of output devices include a display device (e.g., a monitor or projector), speakers, a printer, a network card, tactile-response device, and so forth. Thus, the computing device 1302 may be configured in a variety of ways to support user interaction.

[0085] The computing device 1302 is further illustrated as being communicatively and physically coupled to an accessory device 1314 that is physically and communicatively removable from the computing device 1302. In this way, a variety of different accessory devices may be coupled to the computing device 1302 having a wide variety of configurations to support a wide variety of functionality. In this example, the accessory device 1314 includes one or more controls 1316, which may be configured as press-sensitive keys, mechanically switched keys, buttons, and so forth.

[0086] The accessory device 1314 is further illustrated as including one or more modules 1318 that may be configured to support a variety of functionality. The one or more modules 1318, for instance, may be configured to process analog and/or digital signals received from the controls 1316 to determine whether an input was intended, determine

whether an input is indicative of resting pressure, support authentication of the accessory device 1314 for operation with the computing device 1302, and so on.

[0087] Various techniques may be described herein in the general context of software, hardware elements, or program modules. Generally, such modules include routines, programs, objects, elements, components, data structures, and so forth that perform particular tasks or implement particular abstract data types. The terms "module," "functionality," and "component" as used herein generally represent software, firmware, hardware, or a combination thereof. The features of the techniques described herein are platform-independent, meaning that the techniques may be implemented on a variety of commercial computing platforms having a variety of processors.

[0088] An implementation of the described modules and techniques may be stored on or transmitted across some form of computer-readable media. The computer-readable media may include a variety of media that may be accessed by the computing device 1302. By way of example, and not limitation, computer-readable media may include "computer-readable storage media" and "computer-readable signal media."

[0089] "Computer-readable storage media" may refer to media and/or devices that enable persistent and/or non-transitory storage of information in contrast to mere signal transmission, carrier waves, or signals per se. Thus, computer-readable storage media refers to non-signal bearing media. The computer-readable storage media includes hardware such as volatile and non-volatile, removable and non-removable media and/or storage devices implemented in a method or technology suitable for storage of information such as computer readable instructions, data structures, program modules, logic elements/circuits, or other data. Examples of computer-readable storage media may include, but are not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, hard disks, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or other storage device, tangible media, or article of manufacture suitable to store the desired information and which may be accessed by a computer.

[0090] "Computer-readable signal media" may refer to a signal-bearing medium that is configured to transmit instructions to the hardware of the computing device 1302, such as via a network. Signal media typically may embody computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as carrier waves, data signals, or other transport mechanism. Signal media also include any information delivery media. The term "modulated data signal" means a signal that has one

or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media include wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared, and other wireless media.

[0091] As previously described, hardware elements 1310 and computer-readable media 1306 are representative of modules, programmable device logic and/or fixed device logic implemented in a hardware form that may be employed in some embodiments to implement at least some aspects of the techniques described herein, such as to perform one or more instructions. Hardware may include components of an integrated circuit or on-chip system, microcontroller devices, an application-specific integrated circuit (ASIC), a field-programmable gate array (FPGA), a complex programmable logic device (CPLD), and other implementations in silicon or other hardware. In this context, hardware may operate as a processing device that performs program tasks defined by instructions and/or logic embodied by the hardware as well as a hardware utilized to store instructions for execution, e.g., the computer-readable storage media described previously.

[0092] Combinations of the foregoing may also be employed to implement various techniques described herein. Accordingly, software, hardware, or executable modules may be implemented as one or more instructions and/or logic embodied on some form of computer-readable storage media and/or by one or more hardware elements 1310. The computing device 1302 may be configured to implement particular instructions and/or functions corresponding to the software and/or hardware modules. Accordingly, implementation of a module that is executable by the computing device 1302 as software may be achieved at least partially in hardware, e.g., through use of computer-readable storage media and/or hardware elements 1310 of the processing system 1304. The instructions and/or functions may be executable/operable by one or more articles of manufacture (for example, one or more computing devices 1302 and/or processing systems 1304) to implement techniques, modules, and examples described herein.

Conclusion

[0093] Although the example implementations have been described in language specific to structural features and/or methodological acts, it is to be understood that the implementations defined in the appended claims is not necessarily limited to the specific features or acts described. Rather, the specific features and acts are disclosed as example forms of implementing the claimed features.

CLAIMS

What is claimed is:

1. A method implemented by a host computing device comprising:

calculating multiple raw spatial positions for the host computing device independently using at least two different types of sensors of the host computing device;

processing the multiple raw spatial positions to obtain a combined spatial position for the host computing device;

ascertaining a spatial position for an accessory device connected to the host computing device using one or more sensors of the accessory device; and

computing an orientation of the accessory device relative to the host computing device based on the combined spatial position for the host computing device and the ascertained spatial position for the accessory device.

2. A method as described in claim 1, further comprising exposing the computed orientation for use by one or more applications of the host computing device via an application programming interface (API).

3. A method as described in claim 1, wherein calculating the multiple raw spatial positions for the host computing device comprises using a gyroscope to compute one of the multiple raw spatial positions.

4. A method as described in claim 1, wherein calculating the multiple raw spatial positions for the host computing device comprises using an accelerometer and a magnetometer in combination to compute one of the multiple raw spatial positions.

5. A method as described in claim 1, wherein the spatial position for the accessory device is ascertained via an accelerometer of the accessory device.

6. A method as described in claim 1, wherein the different types of sensors include a gyroscope, an accelerometer, and a magnetometer of the host computing device.

7. A method as described in claim 1, wherein processing the multiple raw spatial positions to obtain the combined spatial position comprises interpolating between the multiple raw spatial positions to reduce inaccuracy of the combined spatial position.

8. A method as described in claim 1, wherein the calculating, processing ascertaining, and computing are performed via one or more microcontrollers of the host computing device configured to implement a sensor fusion module at least partially in hardware.

9. A method as described in claim 1, further comprising controlling one or more behaviors of the host computing device based on the orientation that is computed.

10. A method as described in claim 9, wherein controlling the one or more behaviors of the host computing device comprises selectively adjusting sensitivity of one or more sensors based on the orientation that is computed.