



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO	TOT CLAIMS	IND CLAIMS
61/320,663	04/02/2010		220	16113-2219P01		

CONFIRMATION NO. 5375

26192
FISH & RICHARDSON P.C.
PO BOX 1022
MINNEAPOLIS, MN 55440-1022

FILING RECEIPT



OC000000041090423

Date Mailed: 04/16/2010

Receipt is acknowledged of this provisional patent application. It will not be examined for patentability and will become abandoned not later than twelve months after its filing date. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Applicant(s)

Jaime Guillermo Ruiz, Kitchener, CANADA;
Yang Li, Sunnyvale, CA;

Power of Attorney:

John Dragseth--42497

If Required, Foreign Filing License Granted: 04/13/2010

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 61/320,663**

Projected Publication Date: None, application is not eligible for pre-grant publication

Non-Publication Request: No

Early Publication Request: No

Title

FLIPPING FOR MOTION-BASED INPUT

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international

patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and

Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 C.F.R. §1.53(c).

INVENTOR(S)					
Given Name (first and middle [if any])		Family Name or Surname		Residence (City and either State or Foreign Country)	
Jaime Guillermo Yang		Ruiz Li		Canada Sunnyvale, CA	
Additional inventors are being named on the <u>0</u> separately numbered sheets attached hereto					
TITLE OF THE INVENTION (500 characters max)					
Flipping for Motion-Based Input					
CORRESPONDENCE ADDRESS					
Direct all correspondence to:					
☒ Customer Number: 26192					
OR					
☐ Firm or Individual Name					
Address					
Address					
City		State		ZIP	
Country		Telephone		Fax	
ENCLOSED APPLICATION PARTS (check all that apply)					
☒ Application Data Sheet.		See 37 C.F.R. § 1.76.		☐ CD(s), Number	
☒ Specification		Number of Pages <u>38</u>		☐ Other (specify):	
☒ Drawing(s)		Number of Sheets <u>7</u>			
Application Size Fee: If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$270 (\$135 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. § 41(a)(1)(G) and 37 C.F.R. § 1.16(s)					
METHOD OF PAYMENT OF FILING FEES AND APPLICATION SIZE FEE FOR THIS PROVISIONAL APPLICATION FOR PATENT					
☐ Applicant Claims small entity status. See 37 C.F.R. § 1.27.				TOTAL FEE AMOUNT (\$) 220	
☐ A check or money order is enclosed to cover the filing fees.					
☐ Payment by credit card. Form PTO-2038 is attached.					
☒ The Director is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number <u>06-1050</u> . A copy of this form is enclosed for fee processing.					
The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.					
☒ No.					
☐ Yes, the name of the U.S. Government agency and the Government contract number are: _____					

Respectfully submitted,

Signature: /John A. Dragseth, Reg. No. 42,497/ Date: April 2, 2010

John A. Dragseth, Reg. No. 42,497

Telephone No: (612) 335-5070

Docket No: 16113-2219P01

60631224.doc

CERTIFICATE OF MAILING BY EFS-WEB FILING

I hereby certify that this paper was filed with the Patent and Trademark Office using the EFS-WEB system on this date: April 02, 2010

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	16113-2219P01
		Application Number	
Title of Invention	Flipping for Motion-Based Input		
The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76. This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.			

Secrecy Order 37 CFR 5.2

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Applicant Information:

Applicant 1					
Applicant Authority		☒ Inventor		☐ Legal Representative under 35 U.S.C. 117	
				☐ Party of Interest under 35 U.S.C. 118	
Prefix	Given Name	Middle Name	Family Name	Suffix	
	Jaime	Guillermo	Ruiz		
Residence Information (Select One) ☐ US Residency ☒ Non US Residency ☐ Active US Military Service					
City	Kitchener	Country Of Residence	CA		
Citizenship under 37 CFR 1.41(b)		US			
Mailing Address of Applicant:					
Address 1		499 Shanley Street			
Address 2		Kitchener			
City	Ontario	State/Province			
Postal Code	N2H 5P1	Country	CA		
Applicant 2					
Applicant Authority		☒ Inventor		☐ Legal Representative under 35 U.S.C. 117	
				☐ Party of Interest under 35 U.S.C. 118	
Prefix	Given Name	Middle Name	Family Name	Suffix	
	Yang		Li		
Residence Information (Select One) ☒ US Residency ☐ Non US Residency ☐ Active US Military Service					
City	Sunnyvale	State/Province	CA	Country of Residence	US
Citizenship under 37 CFR 1.41(b)		CN			
Mailing Address of Applicant:					
Address 1		395 Ano Neuvo Ave. #310			
Address 2					
City	Sunnyvale	State/Province	CA		
Postal Code	94085	Country	US		
All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the Add button. Add 					

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

☐ An Address is being provided for the correspondence Information of this application.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	16113-2219P01
		Application Number	
Title of Invention	Flipping for Motion-Based Input		
Customer Number	26192		
Email Address		Add Email	Remove Email

Application Information:

Title of the Invention	Flipping for Motion-Based Input		
Attorney Docket Number	16113-2219P01	Small Entity Status Claimed	☐
Application Type	Provisional		
Subject Matter	Utility		
Suggested Class (if any)		Sub Class (if any)	
Suggested Technology Center (if any)			
Total Number of Drawing Sheets (if any)	7	Suggested Figure for Publication (if any)	

Publication Information:

☐	Request Early Publication (Fee required at time of Request 37 CFR 1.219)
☐	Request Not to Publish. I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application has not and will not be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32). Enter either Customer Number or complete the Representative Name section below. If both sections are completed the Customer Number will be used for the Representative Information during processing.			
Please Select One:	☒ Customer Number	☐ US Patent Practitioner	☐ Limited Recognition (37 CFR 11.9)
Customer Number	26192		

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, or 365(c) or indicate National Stage entry from a PCT application. Providing this information in the application data sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78(a)(2) or CFR 1.78(a)(4), and need not otherwise be made part of the specification.			
Prior Application Status		Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the Add button.			

Foreign Priority Information:

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	16113-2219P01
		Application Number	
Title of Invention	Flipping for Motion-Based Input		

This section allows for the applicant to claim benefit of foreign priority and to identify any prior foreign application for which priority is not claimed. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55(a).

Remove			
Application Number	Country ⁱ	Parent Filing Date (YYYY-MM-DD)	Priority Claimed
			☐ Yes ☒ No
Additional Foreign Priority Data may be generated within this form by selecting the Add button.			

Assignee Information:

Providing this information in the application data sheet does not substitute for compliance with any requirement of part 3 of Title 37 of the CFR to have an assignment recorded in the Office.

Assignee 1

If the Assignee is an Organization check here. ☐

Prefix	Given Name	Middle Name	Family Name	Suffix

Mailing Address Information:

Address 1			
Address 2			
City		State/Province	
Country		Postal Code	
Phone Number		Fax Number	
Email Address			
Additional Assignee Data may be generated within this form by selecting the Add button.			

Signature:

A signature of the applicant or representative is required in accordance with 37 CFR 1.33 and 10.18. Please see 37 CFR 1.4(d) for the form of the signature.

Signature	/John A. Dragseth, Reg. No. 42,497/		Date (YYYY-MM-DD)	2010-04-02
First Name	John A.	Last Name	Dragseth	Registration Number
				42497

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

ABSTRACT OF THE DISCLOSURE

A computer-implemented method for mode switching on an electronic device includes determining that the electronic device has been rotated by a lengthwise axial rotation in a first direction of rotation past a first threshold orientation, determining that the electronic device has been rotated in a second direction of rotation that is substantially the opposite of the first direction of rotation, past a second threshold, and switching a mode of the electronic device based on the two determinations.

60631353.doc

Flipping for Motion-Based Input

TECHNICAL FIELD

[0001] This document relates to systems and techniques for interpreting motion-based user inputs to a mobile computing device.

BACKGROUND

[0002] The use of motion-based inputs in computing devices has become more and more pervasive over the last several years. Motion-based inputs are input that involve movement of an entire device housing, such as by a user's hands, as distinguished from typed, spoken, or touch screen inputs. Primary interaction with the Nintendo Wii gaming console is motion-based (moving the entire WiiMote device), and many smartphones include accelerometers and other motion-sensing components such as magnetometers. Generally, sensor-based interactions in mobile phones have occurred with respect to specific application contexts, such as tilt detection in games and determining screen orientation.

[0003] Some mobile devices depend on users to press a button and/or hold down a button while making a motion input, in order to indicate an intent that they want their movement of a device to be interpreted as a purposeful input rather than as meaningless general motion of the device. Such a mechanism gives the user control over how the device interprets the user's inputs, but it may be difficult for a user to maintain force on a button while performing various motion-based input actions.

SUMMARY

[0004] This document describes systems and techniques that may be used to permit mode switching with respect to motion-based input. For example, a motion of turning a device over and then turning in back to about its original orientation can be determined to be a user input to switch into a motion-based input mode on the device. As such, after a flip/un-flip, or double flip, input motion is provided to a device, subsequent motion of the device may be sensed and interpreted according to a set of predetermined motion-based inputs. For example, while the mode is changed, tilting of the device may be interpreted to move a cursor on a display of the device in the direction of the tilting. A user may repeat the double flip or perform another action (including by inaction, such as by not moving the device for a predetermined time period) to change the device back to its original mode in which motion-based input is not active. Also, a motion-based command may be provided by a user, and followed by a delimiter input such as a double-flip, where the command is recognized and executed only after the delimiter input is received.

[0005] The handling of mode switching and motion-based inputs may be handled by a universal component, such as a component of an operating system on a device, to which various applications on the device may subscribed and may then receive information. For example, an input method editor (IME) on a device may provide functionality for intercepting all inputs to a device (e.g., via physical keyboard, virtual keyboard, softkeys, trackballs, device motion, and voice) and converting those inputs into a common form for provision to applications running on the device. As one example, an application may be open to receiving directional

inputs from a user of the device on which the application is executing. The application may be unconcerned with whether the user enters the information via swiping on a touch screen, rolling a trackball, pressing a D pad, or using motion-based input. Rather, an IME may receive the inputs and may manage whether the device is in a motion-based input mode, and may supply the application with information regarding the user's directional input, where the supplied informational is essentially independent of the mechanism by which the user provided the input.

[0006] In certain implementations, such systems and technique may provide one or more advantages. For example, motion-based input may be more intuitive than keyboard or touch screen input in many instances. However, a user may not want to have their device be open to motion-based input at all times (lest they accidentally provide input that they did not intend). Therefore, the techniques here can provide an input that is natural, but is not likely to be occur by accident (i.e., by mere jostling of the device), so that a user can easily switch to an input mechanism that may be more convenient for them at the moment. In addition, in certain implementations, the gesture gives users the control to activate motion gestures without any hardware modifications to existing devices. The gesture can be quick to perform and can be performed in a limited amount physical space. In addition, a motion to activate a motion-based input mode, and subsequent motions to provide such additional input, may be made without a user having to pause at all.

[0007] In one implementation, a computer-implemented method for mode switching on an electronic device is disclosed. The process comprises determining that the electronic device has been rotated by a lengthwise axial rotation in a first

direction of rotation past a first threshold orientation, determining that the electronic device has been rotated in a second direction of rotation that is substantially the opposite of the first direction of rotation, past a second threshold, and switching a mode of the electronic device based on the two determinations.

[0008] In another implementation, a computer-implemented method for mode switching on an electronic device comprises obtaining data that characterizes a user's motion of the electronic device in free space, determining whether the obtained data matches a model for a particular gesture that corresponds to mode switching, and switching the electronic device, based on the determination, into a mode in which motion of the device in free space is interpreted as intentional user motion-based input and is used to control one or more applications executing on the electronic device.

[0009] In yet another implementation, a computer-implemented system for mode switching on an electronic device is disclosed. The system comprises one or more motion sensors mounted inside a housing of the electronic device, and a mode switching module. The mode switching module is programmed to use information from the motions sensors to determine whether the electronic device has been (a) rotated by a lengthwise axial rotation in a first direction of rotation past a first threshold orientation, and (b) rotated in a second direction of rotation that is substantially the opposite of the first direction of rotation, past a second threshold. The mode switching module is also programmed to cause an operational mode of the electronic device to be switched based on the two determinations.

[0010] The details of one or more embodiments are set forth in the accompanying drawings and the description below. Other features and advantages will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

[0011] FIGs. 1A and 1B show motion-based inputs to a wireless computing device.

[0012] FIG. 2 is a schematic diagram of a smart phone system that supports motion-based inputs.

[0013] FIG. 3 is a flow chart of a process for training a motion-based input model employing a group of mobile device users.

[0014] FIGs. 4A and 4B are flow charts of methods for processing motion-based input to a mobile electronics device.

[0015] FIG. 5 shows an example of a computer device and a mobile computer device that can be used to implement the techniques described here.

[0016] Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0017] This document describes systems and techniques for handling motion-based input to a computing device such as a smart phone or app phone. As described here, a particular motion-based input may be assigned as a delimiter to indicate that a user would like to “open” the device to additional motion-based inputs – i.e., to switch input modes on the device so that the other motion-based inputs that were previously disabled, will be enabled. The delimiting input

discussed here is a double flip, whereby a user makes a first motion with the device, and then backtracks through the motion, where the particular motion here involves rotating the device approximately 180 degrees about an axis of the device and then rotating it back substantially to its original orientation. The delimiting motion could be other motions, though will generally be selected from motions that are not commonly created accidentally, so that a user's device will change modes only when the user intends that it changes modes.

[0018] FIGs. 1A and 1B show motion-based inputs to a wireless computing device. Referring to FIG. 1A, there are shown three chronological instances of a user's interaction with a touch screen smart phone that they are holding in their right hand. In each instance, the user is holding the device horizontally above a table where they may have just picked up the device. In instance (a), the user is cradling the device in their palm, with a touch screen of the device facing upward. The user may have just picked the device up off the table, or may have just finished interacting with the device, such as by contacting a surface of a touch screen on the device with their left forefinger.

[0019] In instance (b), the user has rotated their hand while gripping the smart phone in a normal manner, approximately 180 degrees counterclockwise (as viewed from the user's arm). As a result, the face of the smart phone is now pointing substantially downward toward the table or floor. The rotation in this example has been substantially about a longitudinal axis that passes from the top of the smart phone to the bottom (though the axis may translate some during the rotating motion).

[0020] The user may hold the device in this orientation for a time period, which may be as little as a split instant. Then the user may turn their wrist back clockwise, placing the device in the orientation shown in instance (c), where the touch screen on the device is again facing upward, presumably so that the user can easily view the screen.

[0021] The effect of this motion-based input may be to change an operating mode of the device. For example, the change may involve changing from one operating application on the device to a second operating application (where the device supports multi-tasking). The direction that the user tilted the device may affect the direction that the application on the display is shifted

[0022] Aside from changing an operating mode, the motion-based input may be used to affect the display of a particular application on the device. For example, the input may be used to change pages in a document being displayed on the device. It may also be used to shift through photographs, record albums, songs, or other items on the device. Again, the direction (right or left) that the user tilts the device may affect the direction that such a shift is made.

[0023] The threshold level of motion that is required in order to affect a change in the operation of the device like those discussed here, may be determined in a number of manners. For example, a certain level of inclination of the device against a base plane (which may be a horizontal plane or a slanted plane) may trigger the change. Alternatively, a certain amount of change in inclination may be the trigger, even if a certain level of inclination is not reached. Also a combination of absolute change and relative change may be required for a trigger to occur. In addition, as

explained below, certain users may be asked to perform various motion-based gestures, the orientation information from their actions may be recorded, and a model may be constructed, and that model may be matched to motions by subsequent users in order to determine whether those subsequent users intend a particular motion. Such training can be used for multiple motions, including for a delimiter like that just discussed, and subsequent motion inputs that may be received after the delimiter causes the device to be changed to a motion-based input mode.

[0024] In addition to a threshold for the level of the initial motion, a system may include a threshold for a level of backtracking motion before a delimiter will be triggered. For example, a threshold number of degree of rotation in the backtracking direction may be used, or a certain orientation relative to the starting orientation may be used as a trigger threshold. And again, comparison to models of training user behavior (and subsequent gathered motion/orientation data) may be made such as in the manners discussed in detail below.

[0025] Referring now to FIG. 1B, a similar chronological sequence is shown, though here in figure separate images. In this example, there is no background so as to indicate an absolute frame of reference for the motion. That is because the comparisons described here may be made if the longitudinal axis of the device is horizontal, vertical, or at an angle in between. Also, although there are references here to axes, rotations about the axes, and rotating a device 180 degrees, it should be understood that substantial compliance, within an understanding of acceptable repeatability by a user of a device so that a proper level of false positives or false

negatives is maintained. In particular, absolute values may be assigned as predetermined values to make a device determine when an intended delimiter-creating input is provided (e.g., flipping a device more than 160 degrees total or within 15 degrees from flat horizontal), or acceptable exceptions from model data may be used (e.g., so that 95% of motion inputs that a training group intended to be inputs are considered to be inputs in the model).

[0026] In the figure, the user is first holding their device (e.g., a Nexus One telephone) in a right hand so that its screen is facing them. They then start rotating their hand counterclockwise until the telephone has been rotated through an angle of about 170 degree and then they rotate it back to substantially its original orientation. Note that, with the input completed, the screen is again facing the user, so that the user is in a good position to continue using the phone in a manner that is particular useful.

[0027] FIG. 2 is a schematic diagram of a smart phone system 200 that supports motion-based inputs. In general, the system is represented by a mobile device 202, such as a smart phone, that has a touch screen display 208. In addition, the device 202 may have alternative input mechanisms, such as a track ball 210, D Pad, and other selectable buttons. A number of components within the device 202 may provide for interaction by the device 202 in various manner, including by motion-based inputs (i.e., where the device itself is moved to provide inputs). Only certain example components are shown here, for purposes of clarity. In this example, the display 208 is showing a first-person shooting spaceship game, where, when the device is in a motion-based input mode, the user can steer their spaceship by

rotating the device right or left, and tilting it forward and backward in a familiar manner.

[0028] The device 202 may communicate via a wireless interface 216, through a network 206 such as the internet and/or a cellular network, with servers 204. For example, the device 202 may carry telephone calls through a telephone network or through a data network using VOIP technologies in familiar manners. Also, the device 202 may transmit other forms of data over the internet, such as in the form of HTTP requests that are directed at particular web sites, and may receive responses, such as in the form of mark-up code for generating web pages, as media files, as electronic messages, or in other forms.

[0029] A number of components running on one or more processors installed in the device 202 may enable a user to have simplified motion-based input on the device 202 and touch screen input on the display 208. For example, an interface manager 216 may manage interaction with the touchscreen interface 204, and may include a display controller 212 and a touchscreen input controller 214.

[0030] The display controller 212 may manage what information is shown to a user via interface 208. For example, an operating system on the device 202 may employ display controller 212 to arbitrate access to the interface 202 for a number of applications 218 running on the device 202. In one example, the device 202 may display a number of applications, each in its own window, and the display manager may control what portions of each application are shown on the interface 202.

[0031] The input controller 214 may control the handling of data that is received from a user via the touch screen display 208 or other input mechanisms. For

example, the input controller 214 may coordinate with the display controller 212 to identify where, on the display 208, a user is entering information (e.g., where the cursor is or where the user's finger is contacting the display) so that that the device may understand the context of the input. In addition, the input controller 214 may determine which application or applications 218 should be provided with the input. For example, when the input is provided within a text entry box of an active application, data entered in the box may be made available to that application. Likewise, applications 218 may subscribe with the input controller 214 so that they may be passed information entered by a user in appropriate circumstances. In one example, the input controller 214 may be programmed with an alternative input mechanism like those shown in FIGs. 1A and 1B, and may manage which application or applications 218 are to receive information from the mechanism.

[0032] An input method editor (IME) 220 may also be provided for similar purposes. In particular, the IME 220 may be a form of operating system component that serves as an intermediary between other applications on a device and the interface controller 214. The IME 220 generally is provided to convert user inputs, in whatever form, into textual formats or other formats required by applications 218 that subscribe to receive user input for a system. For example, the IME 220 may receive voice input, may submit that input to a remote server system, may receive back corresponding textual data, and may pass the textual data to an active application. Similarly, the IME 220 may receive input in Roman characters (e.g., A, B, C ...) in pinyin, and may provide suggested Chinese characters to a user (when the pinyin maps to multiple such characters) and then pass the user-selected

character to a subscribing application. Relevant to the input techniques discussed above, the IME 220 may also interpret motion-based inputs sensed by motion sensors 222 (which as used here can include orientation sensors that operate without depending on motion of the device 202), such as accelerometers and magnetometers. For example, an application that wishes to take advantage of a motion-based interface may register (e.g., when it is executed) with the IME 220, designating certain motions as corresponding to standard control inputs, such as a downward sweeping motion corresponding to the downward directional portion of a D-pad control.

[0033] The IME 220 or other form of motion input processing module may make use of user data 226 and motion models 224. The user data 226 may include various information by which the operation of device 202 may be customized for a user. For example, a user may select particular motion-based gestures to correspond to certain commands on device 202, and such information may be used by IME 220 when determining how to interpret received motion-based input from the user. The motion models may contain data generated from testing various users in providing motion-based input, so that the models can be compared to motion-based inputs of a user of device 202 in order to interpret what the user of device 202 intends by their inputs.

[0034] Along with motion-based inputs that correspond to control functions, in certain implementations, the IME 220 or another form of gesture interface can register certain motions or activities by the user as being indicative of activating the motion-based input mode on the device 202. For example, the user may indicate a

desire to use the motion-based input mechanisms described here by moving the device in a certain manner in space, such as by rotating it in a certain manner and then rotating it back to its initial orientation. Upon receipt of such an event, for example, the input controller 214, using the IME 220, may report subsequent inputs as if they were received on a different sort of input mechanism (e.g., a trackball 210 or D Pad). For example, the input controller 214 may report a certain amount of input movement in a particular direction based on how far a user sweeps the device 202 in open space in a particular direction. The input controller 214, via the IME, may also report coordinated inputs that are provided by a user on the surface of the display 208.

[0035] The IME 220 or other gesture interface may also reformat data in other manners. In one example, a music player application may not have been written to work with the motion-based input mechanisms described here. Instead, the music player may receive information to scroll as an indicator that a user has slid their finger on the screen. The IME 220 may be programmed to reside in a communication channel between the input controller 214 and the application, and may convert motion-based input (e.g., tipping the device 202 forward) into the form of messages that the music player application expects to see from the input controller 214 (e.g., sliding of a finger upward on a list of songs on the screen so as to scroll them upward). In this manner, the system 200 can provide a number of different manners in which to provide quick and convenient user input to one or more applications running on device 202.

[0036] FIG. 3 is a flow chart of a process for training a motion-based input model employing a group of mobile device users. The process generally involves providing an adequately large number of users software that they may execute on their mobile devices that include motion or orientation sensors, and that records particular information about the motion of those devices under use, so as to capture data that represents motion by the users in carrying out one or more particular gestures. Such information is then used to determine whether motions by other users on other devices is intended to represent the particular gestures.

[0037] Such an approach can assist in permitting devices to distinguish normal motion from motion-input, with a high recognition rate and low false negative rate.

[0038] The process begins at box 302, where a study group is enrolled. The study group may include volunteers who respond to a request for study participants via an on-line mechanism. Such enrollment may involve storing identification information for the members of the group, demographic information (e.g., age, gender, and technical skills), along with information about each user's mobile device (e.g., ID number and device model).

[0039] For members who fit the requirements of the study and have devices capable of performing in the study, tracking software is downloaded to the user's computing devices at box 304. The software is programmed to access data from the device accelerometer and orientation sensor(s), to format and store such data, and to subsequently upload it so that it can be analyzed. Thus, at box 306, data on the user motion is capture for certain defined gestures. For example, an image showing a gesture may be displayed on a user's screen, and the user may be

asked to perform the motion, such as after their device beeps to indicate that their input is being tracked and recorded. Other motions may then be displayed, and the stored motion data may be associated with the particular gestures that the user was asked to perform. Alternatively, the user may be provided with applications that respond to motion-based input, and the user may interact with the applications in such a way. The tracking software may know the gestures a user is to perform for each of various functions, and may assign motion input to a particular gesture when such input can be distinguished from other possible inputs that the user may be trying to provide to the device.

[0040] In this study, the motion recognizer application was implemented in Java using the Android SDK on Nexus One's with a Qualcomm 8250 1 GHz processor, a three-axis accelerometer, and an orientation sensor. The Nexus One's accelerometer measures the acceleration applied to the phone minus the force of the Earth's gravity along three axes. The orientation sensor measures the degrees of azimuth, pitch and roll of the phone. Azimuth is defined as the angle between the magnetic north direction and the y-axis, around the z-axis. Pitch is the measurement of rotation around the x-axis while roll the measurement of rotation around the y-axis. While the Nexus One also includes proximity, ambient light and magnetic sensors, only the accelerometer and orientations sensors were used in this study because they are common in all Android based devices.

[0041] At box 308, a model of motion data for gestures is built. The model may take various forms, including as a composite of measurements processed in the manner described below with respect to FIG. 3A. The building of the data model, in

particular, may match the manner in which particular inputs from later users are processed before being compared to the “baseline” data in the model.

[0042] At box 310, the processed model data is provided to other user devices. In particular, the information learned from the testing group may be used to establish threshold data for triggering the recognition of a delimiter input for changing the mode of a device. Such providing of the model may occur by the model data being built into operating system data that is loaded on new computing devices, so that such devices may be ready to accept motion-based inputs as soon as they are purchased and enabled. The data may also be downloaded to a preexisting device, such as when a user chooses to have motion-based input functionality like that described here added to their device after they purchase it.

[0043] At box 312, motion-based gesture control is executed on the other user devices employing the model that was developed using the study group. The particular control operation may involve receiving data from sensors on a device, quantizing the data in a particular, and using the quantized data for gesture recognition. Particular example processes for such operations are shown in FIGs. 4A and 4B.

[0044] .FIGs. 4A and 4B are flow charts of methods for processing motion-based input to a mobile electronics device. FIG. 4A shows an example method for quantizing sensor values, which may be used in capturing motion data from users in a training group in building a motion-based model, or from later users for comparison to the model, in order to determine the input intent of those later users. The need for quantization was primarily a result of acceleration being measured in

relation to the Earth's gravity and not actual motion of the device. The inclusion of the Earth's gravity results in different accelerometer readings depending on how the user is holding the phone. The goal in this example was to ensure that the recognizer is able to detect a double flip gesture regardless of the angle at which the user is holding their computing device.

[0045] The process begins at box 402, where a device receives azimuth, pitch, and roll data from orientation sensors on the device. At box 404, those readings are converted from degrees to radians, which helps to compress the scale of possible orientation values, such that acceleration readings are not made irrelevant (leading to many false positives). At box 406, the values are normalized by calculating the differences in the readings over prior readings, because all acceleration values are dependent on the current orientation of the device and azimuth is determined by magnetic North.

[0046] At box 408, floating point values are rounded to integers, which lowers the level of computation that is required while maintaining adequate accuracy. At box 410, the input is compressed by using an averaging window across the data. The window is a 200ms window that is moved in 100ms steps. The compression shortens the input gesture for comparison purposes, and also filters noise generated from the sensor readings. By this process then, quantized data is available to describe the motion of a particular gesture by a user.

[0047] FIG. 4B is a flow chart of a process for recognizing gesture from motion or orientation data on a mobile computing device. This process follows the quantization process for users who are employing the model, and is used to

distinguish random unintentional motion from intentional motion, such as the flip flop, or double flip, motion discussed above with respect to FIGs. 1A and 1B.

[0048] As shown at box 420, the motion-based inputs are first quantized. Such inputs may be group in various manners so as to delineate the intended beginning of a motion-based input, and the intended end of such an input. Such action may occur by starting the capture of data when a device previously at rest is placed in motion. A running window may be applied to the data and testing of the data may also take place periodically over the current window, with comparisons made until a match is determined. Various other manners may also be used for sampling the data and determining how to use the data to identify matching motion.

[0049] In this example, at box 422, the captured data for a user's device is compared to model data from a training group, using Dynamic Time Warping (DTW), which is a classical algorithm based on dynamic programming to match two time series with temporal dynamics, given the function for calculating the distance between two time samples. DTW has become increasingly popular in human-computer interaction (HCI) due to its simplicity to implement and limited amount of required training samples compared to other methods.

[0050] The implementation applied to the double flip gesture discussed here uses a weighted Euclidean distance function for calculating the distance between the quantized time series of acceleration and orientation to the DTW templates. Because the display facing away from the user is an important characteristic of the gesture, the process here doubles the weight to the Euclidean distance of pitch and roll compared to that of azimuth and acceleration along the three axes. Giving

higher weight to pitch and roll minimizes the probability that normal motion will be recognized as a double flip gesture. If the calculated distance is less than a given threshold, the recognizer reports a match for the double flip gesture (box 424). Otherwise, the process returns to collecting and quantizing motion-based inputs (which can include orientation inputs).

[0051] One drawback to the DTW algorithm is the need to typically compare all templates with the incoming gesture. Due to the limited computation resources available in mobile phones compared to desktop computers, it can be important to maximize accuracy while minimizing the number of templates required. To determine the appropriate number of templates, the implementation here was benchmarked by varying the number of templates available to the DTW algorithm. The benchmark consisted of performing recognition 200 times on a recorded input stream for different levels of available templates. The time to match increased quickly at first as templates were added, but then decreased. After approximately 30 templates, the time needed to match the incoming gesture was around 50ms/per recognition task. While this is still relatively quick, it is at this point when the sensors are collecting data faster than the recognizer was able to process the incoming data. Therefore, in this example, the number of available templates was limited to 20 templates to provide an adequate temporal buffer in case variation in computation times.

[0052] Referring again to FIG. 4B, at box 426, the device's input mode is switched when there is a match. In this example, the mode is switched to a mode in which other motion-based gestures are accepted on the device. Thus, initially,

the device may have been open only to the particular double flip motion-based gesture shown in this document. However, after such a gesture is provided by a user, the device may make itself open to various other gestures, including gestures that are more subtle than the double flip gesture, and thus would not be appropriate motion-based gestures to leave active on the device at all times.

[0053] At box 428, the user continues to provide motion-based input, and the process identifies (e.g., using the quantizing and matching just discussed) whether that input matches particular defined gestures that were not available until after the double flip gesture was provided to change the device mode. Eventually, the user may change the mode back to its initial state, such as by performing another double flip gesture. The particular mode in which the device is currently set may be indicated, for example, by an icon on a display of the device.

[0054] In this process, a match is indicated when the distance calculated by DTW between a user's input motions and a template from a saved model is below a threshold. Setting the threshold too low (requiring a more exact match) will result in the gesture being difficult to perform, resulting in a high rate of false negatives. In contrast, setting the threshold too high will result in a high rate of false positives.

[0055] Two experiments were conducted in order to find an optimal threshold. A first experiment looked at false positives. To collect data representing the motion a phone sustains during everyday use, custom software was written in Java using the Android SDK to log the data stream received from a device's accelerometer and orientation sensors. The software was also responsible for periodically uploading the recorded data to a secure server whenever the display was off. The ability to

remotely collect the data allowed the recruitment of a large number of participants without needing to be physically present. Recruitment of participants was performed by sending an email to an internal email list of employees at a high-tech company asking for volunteers. The email directed the employees to an internal webpage that provided information on the software and a link to download the application. Participants were required to use a Nexus One as their primary mobile device and were asked to run the software for at least 24 hours. Ninety-nine participants installed and ran the software long enough such that at least one log was uploaded to the server. In total, 2169.2 hours of sensor data was collected.

[0056] To analyze the dataset, software was written in Java to replay the recorded input stream. As the data stream was read, gesture recognition was performed and detected, and double flip gestures were recorded. To determine the effects of threshold on the false positive rate, the program iterated through the complete dataset, incrementing the threshold distance by one for each iteration.

[0057] For results of the first study, the false positive rate followed an s-curve shape for false positive rate as the threshold distance was increase, with the false positive rate maxing at about 10%, which is a relatively low rate.

[0058] In the second experiment, directed to false positives, the effort was aimed at too low of a threshold that will result in a large number of false negatives resulting in user frustration. Ultimately, the experiment looked to find the balance between false positive and false negative conditions.

[0059] At the beginning of the study, the researcher described the procedure and demonstrated the double flip gesture. The researcher then handed the

participant a phone running the same software used in the first experiment and asked to perform the double flip gesture 20 times, pausing between each gesture. Upon completion, the participants were asked to rate how easy the gesture was to perform using a 5-point Likert. The trial took approximately 5 minutes to complete. Twenty volunteers were recruited using the same email list as the first experiment, resulting in a total of 400 gestures being recorded and analyzed. Data was analyzed using the same software and procedure as in our first experiment.

[0060] For results, true positive rates by threshold level rose initially as threshold distance increase but then quickly hit 100% at a threshold around 41. Table 1 shows the true and false positive rates for the six lowest thresholds from both experiments.

Threshold	False Positive	True Positive
35	0.00%	73.00%
36	0.00%	82.00%
37	0.00%	87.00%
38	0.00%	89.00%
39	0.00%	96.00%
40	0.00%	98.5%
41	0.01%	100%

Table 1

[0061] A threshold of 41 in this example was thus optimal, providing a 100% true positive rate while only a 0.01% false positive rate. Users in the study rated the gesture very high with an average rating of 4.6 scale (1 being difficult and 5 being easy). As shown in Figure 4, only one participant rated the gesture lower than “somewhat easy” indicating that they felt using the gesture frequently in succession for a long period of time would result in fatigue.

[0062] Study Determining Appropriate Features for a Mode-Changing Gesture

[0063] A study of double flipping motion-based input was performed at a technology company. 2,100 hours of motion data was captured from the phones of ninety-nine volunteers, and it was found that a recognizer was extremely resistant to false positives (less than 0.01%).

[0064] As the first step of the study, twenty volunteers, ten males and ten females, were interviewed. The participants all worked for a high-tech company but did not all hold technical positions. All participants indicated that they used a smartphone as their primary mobile device and ranged between the ages of 21-44 (mean = 28, $\sigma = 5.4$). The volunteers received a \$30 gift certificate to an online bookseller for their participation.

[0065] Study Description and Procedure

[0066] The study consisted of a semi-structured interview designed to explore users' pre-existing mental models of how motion-gesture based interaction should occur. Interviews were videotaped and lasted approximately one hour. At the beginning of the interview, participants were provided with an HTC Nexus One smartphone running custom software developed in Java using the Android SDK. The software was responsible for logging all data received from the phone's accelerometer and orientation sensors.

[0067] The interviewer then instructed the participant to imagine that the device was a "magical device" with the ability to perfectly infer the user's intended action. Next, the interviewer explained that we are researching possible uses of motion to

interact with a phone, for example, when wearing gloves and unable to interact with the touch screen. The participant was then asked to come up of list of tasks that they believed would benefit from the use of physical motion gestures. For each task the user created, the volunteer was asked to brainstorm possible gestures to represent the task using the think aloud protocol. During this time, the researcher would also ask questions as needed for clarification. Once the participant decided on a final gesture design, she was asked to perform the gesture using the provided phone. Finally, the participant was asked to self-rate the gesture she designed for how well it matched its intended purpose.

[0068] Results

[0069] For each interview, a transcript of the recorded video was created to extract individual quotes. The quotes were then clustered to identify common themes using a bottom-up, inductive analysis approach. We now describe the emerging themes relevant to the design of the delimiting gesture described here.

[0070] First: The desire for moded interaction as a method to reduce false positive and false negative conditions. Even though participants were told to assume the phone had perfect recognition rates, some participants still wanted to explicitly activate and deactivate when the phone was listening for a gesture. Participants expressed their concern about false positive conditions on actions which they considered to be detrimental (i.e. hanging up the phone prematurely) and false negative conditions when confirmation that particular action had been completed was desirable.

[0071] To allow a user control over motion gesture mode, a majority of participants suggested the use of a physical button as a method to activate physical motion gesture mode. However, there was no consensus on where the button should be located. In addition, some participants mentioned that holding a button and performing the gesture often made the gesture harder to perform. Another suggestion was the use of a virtual button displayed on the touchscreen. Participants who suggested this method were asked a follow up question on how the mode would be activated when unable to use the screen, such as when wearing gloves. In this situation, participants either mentioned a physical button or wanted to turn off gestures all together.

[0072] Second: The screen should be visible at all times. When brainstorming possible gestures, participants were very mindful with being able to see the screen during and after a gesture. Participants who designed gestures where the screen was not visible justified their design decisions by indicated the gesture represented not needing to interact with the phone. Participants favored quick simple gestures that minimized the amount of physical space needed to perform the gesture. Participants who performed larger movements mentioned concern about either hitting themselves or their neighbor.

[0073] THE DOUBLE FLIP DELIMETER

[0074] The requirement for a gesture delimiter is the result of our participants requesting the ability to specify when the system should enable motion gestures. We also recognize that while participants were told to imagine a perfect recognition device, no such device exists. Therefore, the use of a delimiter seems like a

reasonable method to reduce false positives and still being able to achieve a high recognition rate.

[0075] During our study, participants suggested the use of a physical or a virtual button press as an option to activate motion gesture based interaction. However, our goal is to allow motion gestures to be used in absence of a touch screen. While a hardware button is feasible, participants commented that holding a button and performing the gesture was often cumbersome. In addition, our goal is to create a delimiter that is not dependent on hardware but instead relies on the sensors commonly available in today's smart phones. Therefore, we decided that the delimiter should be a motion gesture that meets the following requirements:

- The gesture should be resistant to false positives. The main goal of having a delimiter is to limit false positives. Therefore, any delimiter must also be resistant to false positives and be a gesture that is unlikely to occur during normal use of the phone.
- The screen must be visible after completing the gesture. This allows the user to interact with the phone after the delimiter has been completed.
- The gesture should be quick and compact. The physical space required to perform the gesture should be limited in order to allow the gesture to be performed in crowded spaces.
- The delimiter should be unique so that it does not collide with a gesture that makes more sense in another context. In other words, the gesture should be distinct from any gestures that a user/developer would likely want to incorporate into an application.

[0076] As shown in Figure 1B, the Double Flip delimiter is performed by quickly rotating the wrist such that the phone's display is facing away from the user, then back to the original position with the display facing the user. Since during normal use, users commonly interact with the front of the device, due to the location of the display, it is unlikely the gesture will be triggered accidentally. The need to flip the phone back so the display is facing the user also allows us to fulfill our second design requirement, in that after the gesture has been performed the user is able to interact with the information presented on the display. Because the gesture can be performed with the simple rotation of the wrist and no movement in the arm, the gesture is also very compact and requires no more physical space than holding the phone such that the screen is visible to the user. In addition, the gesture is easy to perform and can be done quickly.

[0077] Finally, while some participants did suggest rotating the phone such that the screen was pointed away from the individual, no participants suggested the Double Flip gesture. Therefore, the DoubleFlip gesture is unlikely to collide with gestures designed by developers.

[0078] FIG. 5 shows an example of a generic computer device 500 and a generic mobile computer device 550, which may be used with the techniques described here. Computing device 500 is intended to represent various forms of digital computers, such as laptops, desktops, workstations, personal digital assistants, servers, blade servers, mainframes, and other appropriate computers. Computing device 550 is intended to represent various forms of mobile devices, such as personal digital assistants, cellular telephones, smart phones, and other

similar computing devices. The components shown here, their connections and relationships, and their functions, are meant to be exemplary only, and are not meant to limit implementations of the inventions described and/or claimed in this document.

[0079] Computing device 500 includes a processor 502, memory 504, a storage device 506, a high-speed interface 508 connecting to memory 504 and high-speed expansion ports 510, and a low speed interface 512 connecting to low speed bus 514 and storage device 506. Each of the components 502, 504, 506, 508, 510, and 512, are interconnected using various busses, and may be mounted on a common motherboard or in other manners as appropriate. The processor 502 can process instructions for execution within the computing device 500, including instructions stored in the memory 504 or on the storage device 506 to display graphical information for a GUI on an external input/output device, such as display 516 coupled to high speed interface 508. In other implementations, multiple processors and/or multiple buses may be used, as appropriate, along with multiple memories and types of memory. Also, multiple computing devices 500 may be connected, with each device providing portions of the necessary operations (e.g., as a server bank, a group of blade servers, or a multi-processor system).

[0080] The memory 504 stores information within the computing device 500. In one implementation, the memory 504 is a volatile memory unit or units. In another implementation, the memory 504 is a non-volatile memory unit or units. The memory 504 may also be another form of computer-readable medium, such as a magnetic or optical disk.

[0081] The storage device 506 is capable of providing mass storage for the computing device 500. In one implementation, the storage device 506 may be or contain a computer-readable medium, such as a floppy disk device, a hard disk device, an optical disk device, or a tape device, a flash memory or other similar solid state memory device, or an array of devices, including devices in a storage area network or other configurations. A computer program product can be tangibly embodied in an information carrier. The computer program product may also contain instructions that, when executed, perform one or more methods, such as those described above. The information carrier is a computer- or machine-readable medium, such as the memory 504, the storage device 506, memory on processor 502, or a propagated signal.

[0082] The high speed controller 508 manages bandwidth-intensive operations for the computing device 500, while the low speed controller 512 manages lower bandwidth-intensive operations. Such allocation of functions is exemplary only. In one implementation, the high-speed controller 508 is coupled to memory 504, display 516 (e.g., through a graphics processor or accelerator), and to high-speed expansion ports 510, which may accept various expansion cards (not shown). In the implementation, low-speed controller 512 is coupled to storage device 506 and low-speed expansion port 514. The low-speed expansion port, which may include various communication ports (e.g., USB, Bluetooth, Ethernet, wireless Ethernet) may be coupled to one or more input/output devices, such as a keyboard, a pointing device, a scanner, or a networking device such as a switch or router, e.g., through a network adapter.

[0083] The computing device 500 may be implemented in a number of different forms, as shown in the figure. For example, it may be implemented as a standard server 520, or multiple times in a group of such servers. It may also be implemented as part of a rack server system 524. In addition, it may be implemented in a personal computer such as a laptop computer 522. Alternatively, components from computing device 500 may be combined with other components in a mobile device (not shown), such as device 550. Each of such devices may contain one or more of computing device 500, 550, and an entire system may be made up of multiple computing devices 500, 550 communicating with each other.

[0084] Computing device 550 includes a processor 552, memory 564, an input/output device such as a display 554, a communication interface 566, and a transceiver 568, among other components. The device 550 may also be provided with a storage device, such as a microdrive or other device, to provide additional storage. Each of the components 550, 552, 564, 554, 566, and 568, are interconnected using various buses, and several of the components may be mounted on a common motherboard or in other manners as appropriate.

[0085] The processor 552 can execute instructions within the computing device 550, including instructions stored in the memory 564. The processor may be implemented as a chipset of chips that include separate and multiple analog and digital processors. The processor may provide, for example, for coordination of the other components of the device 550, such as control of user interfaces, applications run by device 550, and wireless communication by device 550.

[0086] Processor 552 may communicate with a user through control interface 558 and display interface 556 coupled to a display 554. The display 554 may be, for example, a TFT LCD (Thin-Film-Transistor Liquid Crystal Display) or an OLED (Organic Light Emitting Diode) display, or other appropriate display technology. The display interface 556 may comprise appropriate circuitry for driving the display 554 to present graphical and other information to a user. The control interface 558 may receive commands from a user and convert them for submission to the processor 552. In addition, an external interface 562 may be provide in communication with processor 552, so as to enable near area communication of device 550 with other devices. External interface 562 may provide, for example, for wired communication in some implementations, or for wireless communication in other implementations, and multiple interfaces may also be used.

[0087] The memory 564 stores information within the computing device 550. The memory 564 can be implemented as one or more of a computer-readable medium or media, a volatile memory unit or units, or a non-volatile memory unit or units. Expansion memory 574 may also be provided and connected to device 550 through expansion interface 572, which may include, for example, a SIMM (Single In Line Memory Module) card interface. Such expansion memory 574 may provide extra storage space for device 550, or may also store applications or other information for device 550. Specifically, expansion memory 574 may include instructions to carry out or supplement the processes described above, and may include secure information also. Thus, for example, expansion memory 574 may be provide as a security module for device 550, and may be programmed with

instructions that permit secure use of device 550. In addition, secure applications may be provided via the SIMM cards, along with additional information, such as placing identifying information on the SIMM card in a non-hackable manner.

[0088] The memory may include, for example, flash memory and/or NVRAM memory, as discussed below. In one implementation, a computer program product is tangibly embodied in an information carrier. The computer program product contains instructions that, when executed, perform one or more methods, such as those described above. The information carrier is a computer- or machine-readable medium, such as the memory 564, expansion memory 574, memory on processor 552, or a propagated signal that may be received, for example, over transceiver 568 or external interface 562.

[0089] Device 550 may communicate wirelessly through communication interface 566, which may include digital signal processing circuitry where necessary. Communication interface 566 may provide for communications under various modes or protocols, such as GSM voice calls, SMS, EMS, or MMS messaging, CDMA, TDMA, PDC, WCDMA, CDMA2000, or GPRS, among others. Such communication may occur, for example, through radio-frequency transceiver 568. In addition, short-range communication may occur, such as using a Bluetooth, WiFi, or other such transceiver (not shown). In addition, GPS (Global Positioning System) receiver module 570 may provide additional navigation- and location-related wireless data to device 550, which may be used as appropriate by applications running on device 550.

[0090] Device 550 may also communicate audibly using audio codec 560, which may receive spoken information from a user and convert it to usable digital information. Audio codec 560 may likewise generate audible sound for a user, such as through a speaker, e.g., in a handset of device 550. Such sound may include sound from voice telephone calls, may include recorded sound (e.g., voice messages, music files, etc.) and may also include sound generated by applications operating on device 550.

[0091] The computing device 550 may be implemented in a number of different forms, as shown in the figure. For example, it may be implemented as a cellular telephone 580. It may also be implemented as part of a smartphone 582, personal digital assistant, or other similar mobile device.

[0092] Various implementations of the systems and techniques described here can be realized in digital electronic circuitry, integrated circuitry, specially designed ASICs (application specific integrated circuits), computer hardware, firmware, software, and/or combinations thereof. These various implementations can include implementation in one or more computer programs that are executable and/or interpretable on a programmable system including at least one programmable processor, which may be special or general purpose, coupled to receive data and instructions from, and to transmit data and instructions to, a storage system, at least one input device, and at least one output device.

[0093] These computer programs (also known as programs, software, software applications or code) include machine instructions for a programmable processor, and can be implemented in a high-level procedural and/or object-oriented

programming language, and/or in assembly/machine language. As used herein, the terms “machine-readable medium” “computer-readable medium” refers to any computer program product, apparatus and/or device (e.g., magnetic discs, optical disks, memory, Programmable Logic Devices (PLDs)) used to provide machine instructions and/or data to a programmable processor, including a machine-readable medium that receives machine instructions as a machine-readable signal. The term “machine-readable signal” refers to any signal used to provide machine instructions and/or data to a programmable processor.

[0094] To provide for interaction with a user, the systems and techniques described here can be implemented on a computer having a display device (e.g., a CRT (cathode ray tube) or LCD (liquid crystal display) monitor) for displaying information to the user and a keyboard and a pointing device (e.g., a mouse or a trackball) by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback (e.g., visual feedback, auditory feedback, or tactile feedback); and input from the user can be received in any form, including acoustic, speech, or tactile input.

[0095] The systems and techniques described here can be implemented in a computing system that includes a back end component (e.g., as a data server), or that includes a middleware component (e.g., an application server), or that includes a front end component (e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the systems and techniques described here), or any combination of such back end,

middleware, or front end components. The components of the system can be interconnected by any form or medium of digital data communication (e.g., a communication network). Examples of communication networks include a local area network ("LAN"), a wide area network ("WAN"), and the Internet.

[0096] The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

[0097] A number of embodiments have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. For example, much of this document has been described with respect to television advertisements, but other forms of future, viewership-based advertisements may also be addressed, such as radio advertisements and on-line video advertisements.

[0098] In addition, the logic flows depicted in the figures do not require the particular order shown, or sequential order, to achieve desirable results. In addition, other steps may be provided, or steps may be eliminated, from the described flows, and other components may be added to, or removed from, the described systems. Accordingly, other embodiments are within the scope of the following claims.

WHAT IS CLAIMED IS:

1. A computer-implemented method for mode switching on an electronic device,
the method comprising:

- determining that the electronic device has been rotated by a lengthwise axial rotation in a first direction of rotation past a first threshold orientation;

- determining that the electronic device has been rotated in a second direction of rotation that is substantially the opposite of the first direction of rotation, past a second threshold; and

- switching a mode of the electronic device based on the two determinations.

2. A computer-implemented method for mode switching on an electronic device,
the method comprising:

- obtaining data that characterizes a user's motion of the electronic device in free space;

- determining whether the obtained data matches a model for a particular gesture that corresponds to mode switching; and

- switching the electronic device, based on the determination, into a mode in which motion of the device in free space is interpreted as intentional user motion-based input and is used to control one or more applications executing on the electronic device.

3. A computer-implemented system for mode switching on an electronic device,
the system comprising:

one or more motion sensors mounted inside a housing of the electronic
device; and

a mode switching module programmed to:

use information from the motions sensors to determine whether the
electronic device has been:

rotated by a lengthwise axial rotation in a first direction of
rotation past a first threshold orientation, and

rotated in a second direction of rotation that is substantially
the opposite of the first direction of rotation, past a second
threshold; and

cause an operational mode of the electronic device to be switched
based on the two determinations.

1/7

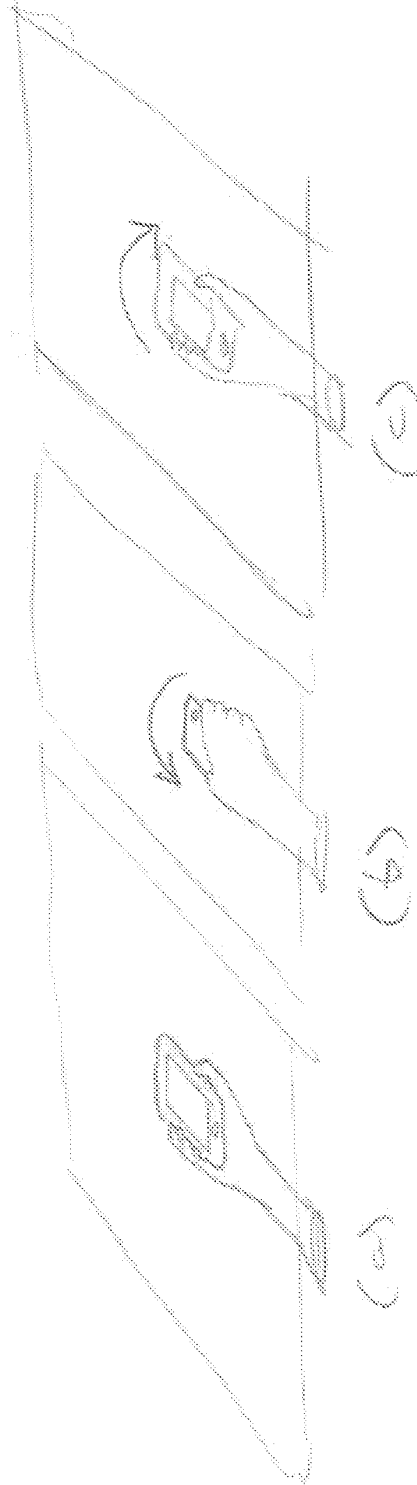


FIG. 1A

2/7

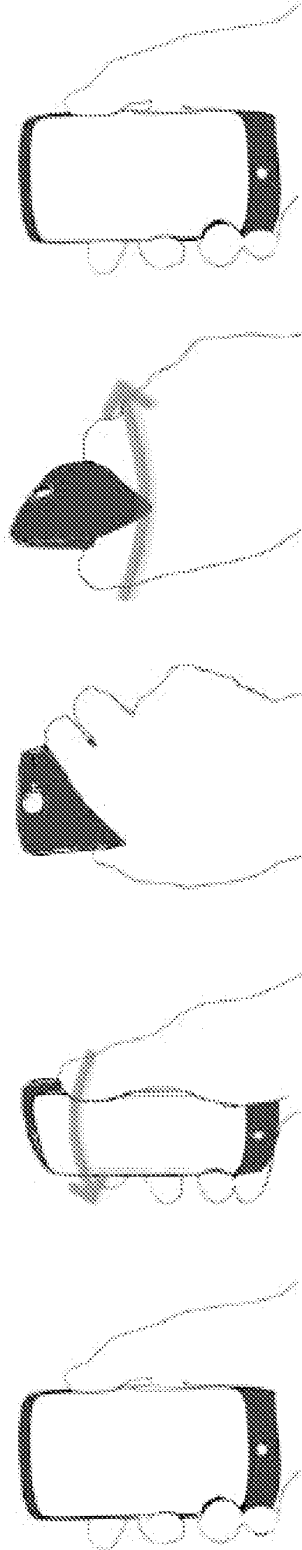


FIG. 1B

3/7

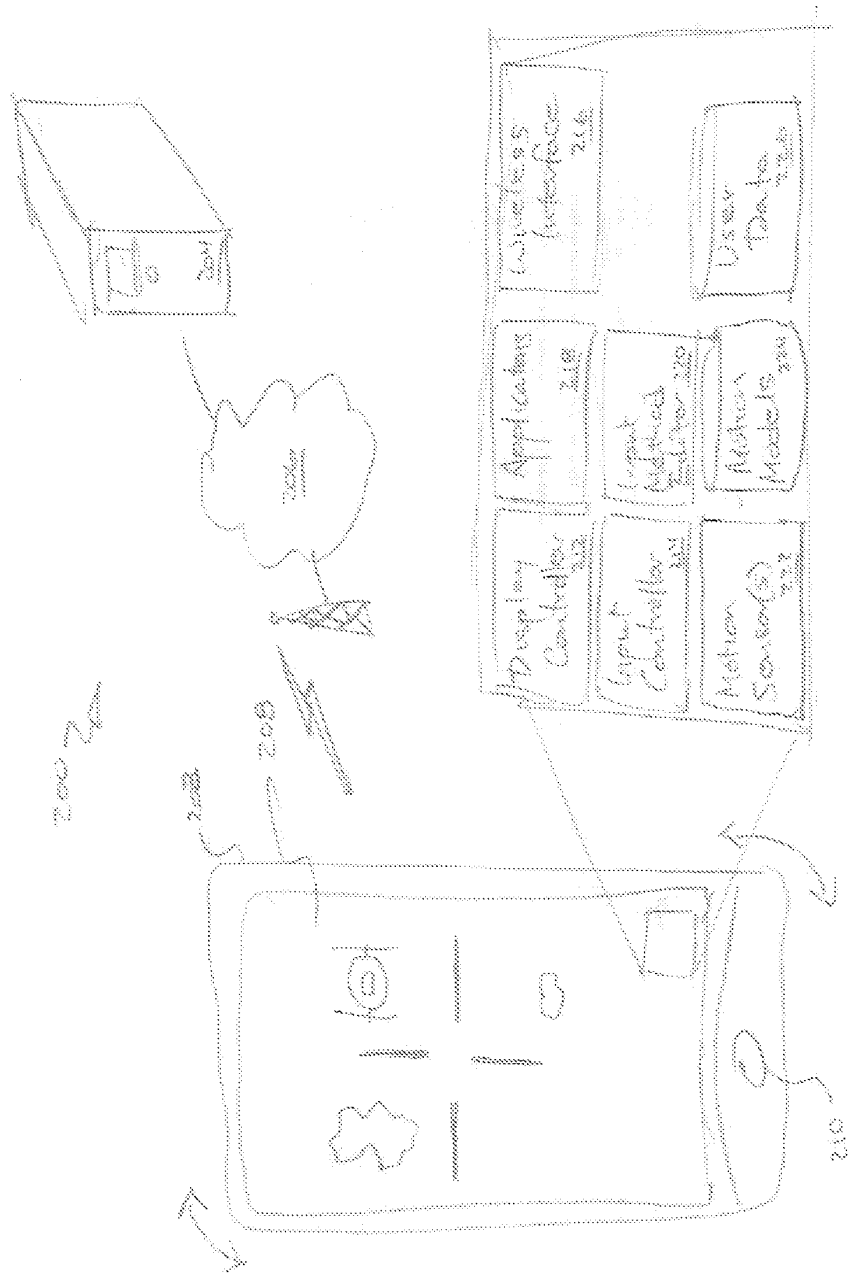


FIG. 2

4/7

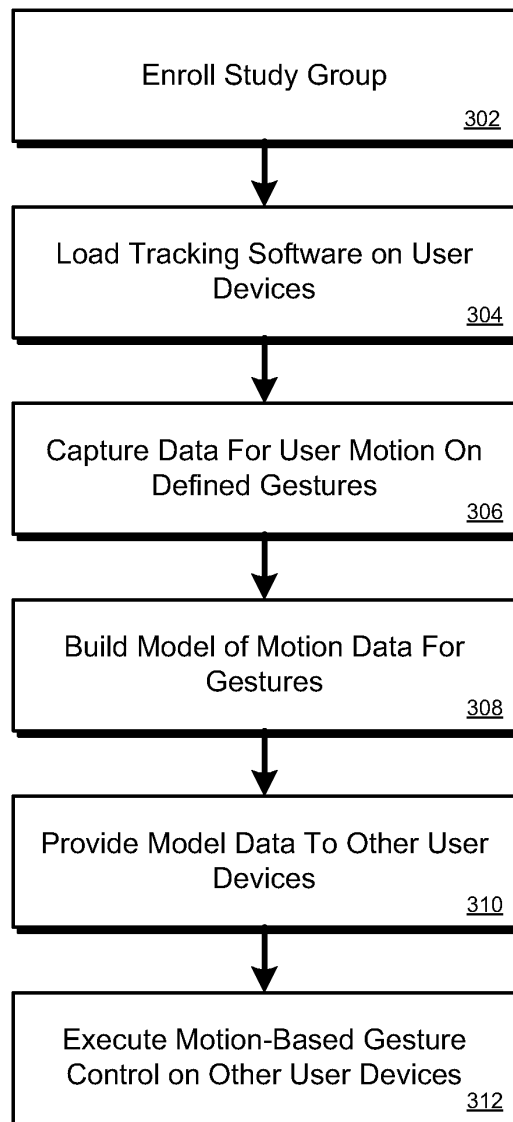


FIG. 3

5/7

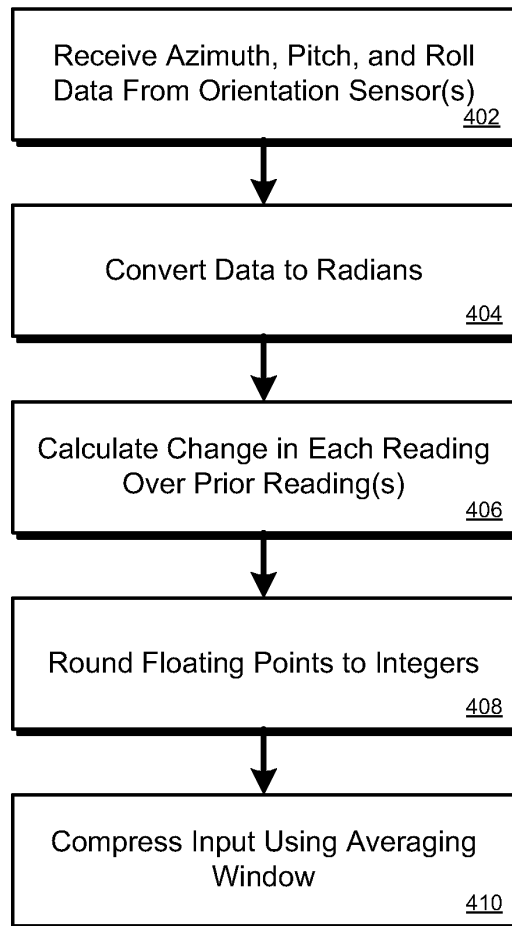


FIG. 4A

6/7

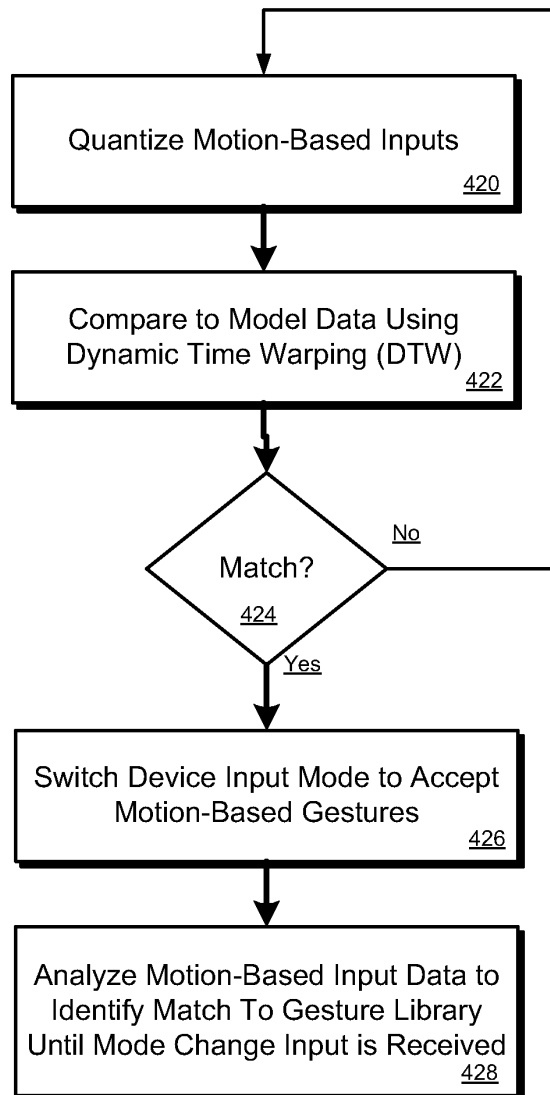


FIG. 4B

7/7

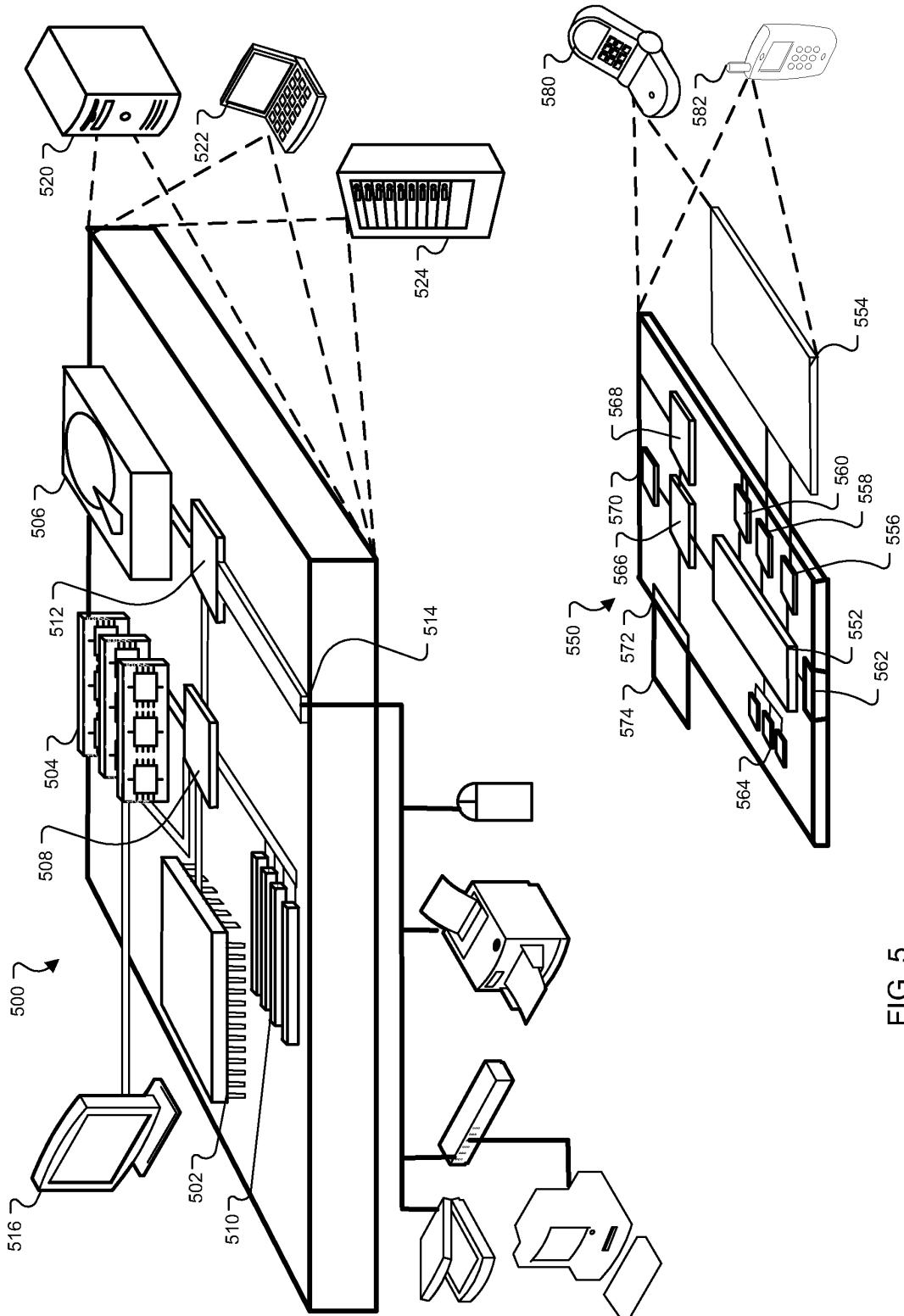


FIG. 5