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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 August 2012 (16.08.2012)

WIPO | PCT

(10) International Publication Number
WO 2012/109636 A2

(51) International Patent Classification:
G06F 3/041 (2006.01) *G06F 9/44* (2006.01)

(21) International Application Number:
PCT/US2012/024781

(22) International Filing Date:
11 February 2012 (11.02.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/442,219 12 February 2011 (12.02.2011) US
13/099,288 2 May 2011 (02.05.2011) US

(71) Applicant *(for all designated States except US)*: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **ZHAO, Weidong**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **STEVENS, David A.**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **UZELAC, Aleksandar**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **SHIGEMITSU, Takahiro**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **WILSON, Andrew David**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **KEAM, Nigel Stuart**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

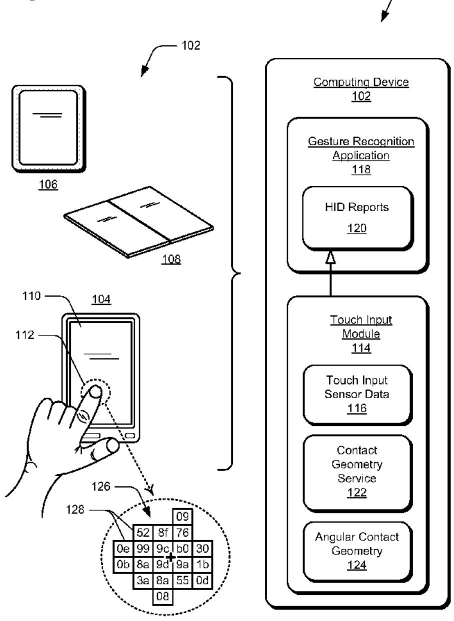
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG,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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU,

[Continued on next page]

(54) Title: ANGULAR CONTACT GEOMETRY

FIG. 1



(57) Abstract: In embodiments of angular contact geometry, touch input sensor data is recognized as a touch input on a touch-screen display, such as a touch-screen display integrated in a mobile phone or portable computing device. A sensor map is generated from the touch input sensor data, and the sensor map represents the touch input. The sensor map can be generated as a two-dimensional array of elements that correlate to sensed contact from a touch input. An ellipse can then be determined that approximately encompasses elements of the sensor map, and the ellipse represents a contact shape of the touch input.

WO 2012/109636 A2

LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

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

ANGULAR CONTACT GEOMETRY

BACKGROUND

[0001] Portable computing devices, such as mobile phones, portable and tablet computers, entertainment devices, handheld navigation devices, and the like increasingly offer more functions and features which can make it difficult for a user to navigate and select commands that are relevant to a function the user wants to initiate on a device. In addition to the traditional techniques used to interact with computing devices, such as a mouse, keyboard, and other input devices, touch sensors and touch-screen displays are commonly integrated in mobile phones and tablet computers, and are utilized both for display and user-selectable touch and gesture inputs. A continuing design challenge with these types of portable devices having touch sensors and/or touch-screen displays is the inherent inaccuracy of touch and gesture inputs, due primarily to the size of users' fingers. While users typically want smaller devices to conveniently carry, the challenges to accurately process touch and gesture inputs on touch-screen displays continue with standard touch processing techniques.

SUMMARY

[0002] This Summary introduces simplified concepts of angular contact geometry, and the concepts are further described below in the Detailed Description and/or shown in the Figures. This Summary should not be considered to describe essential features of the claimed subject matter, nor used to determine or limit the scope of the claimed subject matter.

[0003] Angular contact geometry is described. In embodiments, touch input sensor data is recognized as a touch input on a touch-screen display, such as may be integrated in a mobile phone or portable computing device. A sensor map is generated from the touch input sensor data, and the sensor map represents the touch input. The sensor map can be generated as a two-dimensional array of elements that correlate to sensed contact from the touch input. An ellipse can then be determined that approximately encompasses the elements of the sensor map, and the ellipse represents a contact shape of the touch input.

[0004] In other embodiments, the contact shape of a touch input is generally irregular in shape, and the determined ellipse is of a size and rotation angle that approximately encompass the elements of the sensor map. An inaccuracy ratio (IAR) can be determined to validate the ellipse, where the IAR of the ellipse is determined from regions that are within the bounds of the ellipse, but are not part of the sensor map. These regions are also referred to as non-overlapping regions, and are the regions within the bounds of the ellipse that do not overlap an element of the sensor map.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] Embodiments of angular contact geometry are described with reference to the following Figures. The same numbers may be used throughout to reference like features and components that are shown in the Figures:

FIG. 1 illustrates an example system in which embodiments of angular contact geometry can be implemented.

FIG. 2 illustrates an example of a sensor map and an ellipse that approximately encompasses the elements of the sensor map in accordance with one or more embodiments of angular contact geometry.

FIG. 3 illustrates example method(s) of angular contact geometry in accordance with one or more embodiments.

FIG. 4 illustrates various components of an example device that can implement embodiments of angular contact geometry.

DETAILED DESCRIPTION

[0006] Embodiments of angular contact geometry are described. As noted above, touch and gesture inputs on a touch-screen display of a computing device, such as a mobile phone or portable computer, may not be accurately recognized due to the size of users' fingers when initiating a user-selectable input. This may be particularly noticeable to a user when attempting to touch smaller selectable controls that are displayed in a user interface on a touch-screen display, such as selectable links in a Web page or when trying to resize a user interface with a selectable edge sizing control. Angular contact geometry is implemented to represent a contact shape of a touch input, such as on a touch-screen display of a device, to infer a user-intended touch location on the touch-screen display.

[0007] In embodiments, the angular contact geometry is determined from a sensor map that is generated from touch input sensor data, and the sensor map represents a touch input. An ellipse can be determined that approximately encompasses elements of the sensor map, and the ellipse represents a contact shape of the touch input. Generally, the contact shape of a touch input is irregular in shape, and the ellipse of a size and rotation angle approximately encompasses the elements of the sensor map. A quality metric for implementations of angular contact geometry can also be determined.

[0008] The quality metric is an inaccuracy ratio (IAR) that can be determined to justify or validate an ellipse. The IAR for an ellipse of a particular size and rotation angle is determined from regions that are within the bounds of the ellipse, but are not part of the sensor map. These regions are also referred to as non-overlapping regions, and are the regions within the bounds of the ellipse that do not overlap an element of the sensor map. A best-fit ellipse has an optimal IAR of the smallest value, where the ellipse with the smallest value IAR encompasses most of the sensor map with the least non-overlapping region area.

[0009] While features and concepts of angular contact geometry can be implemented in any number of different devices, systems, environments, networks, and/or configurations, embodiments of angular contact geometry are described in the context of the following example devices, systems, and methods.

[0010] FIG. 1 illustrates an example system 100 in which various embodiments of angular contact geometry can be implemented. The example system includes a computing device 102, which may be any one or combination of a mobile phone 104, entertainment device, navigation device, user device, wireless device, portable device, tablet computer 106, dual-screen folding device 108, and the like. The computing device includes an integrated touch-screen display 110, which is implemented to sense the position of a touch input 112, such as a user-selectable input in a user interface that is displayed on the touch-screen display. Any of the computing devices can be implemented with various components, such as one or more processors and memory devices, as well as any number and combination of differing components as further described with reference to the example device shown in FIG. 4.

[0011] In the example system 100, the computing device 102 includes a touch input module 114 (*e.g.*, a lower layer component) that is implemented to recognize touch input sensor data 116 as the touch input 112 on the touch-screen display 110. The computing device also includes a gesture recognition application 118 (*e.g.*, a higher layer component) that receives HID reports 120 (*i.e.*, human interface device reports) when generated and output by the touch input module 114. The HID reports include a time and position data, as well as determined angular contact geometry data, such as ellipse axis vectors, that correlates to touch inputs on the touch-screen display of the computing device. The gesture recognition application 118 is implemented to recognize and generate various gestures as determined from touch input data (*e.g.*, the HID reports 120) associated with inputs or combinations of inputs, such as the touch input 112. The gesture recognition application can generate various gestures, such as select gestures, hold gestures, motion gestures, tap gestures, and other types of gestures from various user-selectable inputs.

[0012] An input recognition system of the computing device 102 may include any type of input detection features and/or devices to distinguish the various types of inputs, such as sensors (capacitive or resistive), light sensing pixels, touch sensors, cameras, and/or a natural user interface that interprets user interactions, gestures, inputs, and motions. In implementations, the input recognition system can detect motion inputs from discernable variables, such as from a direction variable, from start region position variables and end region position variables, and/or from a motion rate variable (*e.g.*, a particular number of pixels per second).

[0013] The computing device 102 also includes a contact geometry service 122 that is implemented to determine the angular contact geometry 124 that corresponds to a touch input on the touch-screen display 110, such as the touch input 112. The contact geometry service can be implemented as computer-executable instructions, such as a software application, and executed by one or more processors to implement the various embodiments described herein. The contact geometry service can also be implemented as firmware on dedicated sensor device hardware in the computing device. In this example, the contact geometry service is shown implemented as a component of the touch input module 114. Alternatively, the contact geometry service may be implemented as an independent software application or service to determine the angular contact geometry.

[0014] In embodiments, the contact geometry service 122 is implemented to generate a sensor map 126 from the touch input sensor data 116, and the sensor map represents the touch input 112, such as when a user initiates a touch input with a finger on the touch-screen display 110. In this example, the sensor map includes elements 128 shown as 8-bit hex values that represent the signal strength at an element position in the sensor map. A stronger sensor signal of the touch input sensor data indicates more contact with an element in the sensor map. The sensor map is generated as a two-dimensional array s[x][y], where x and y are the indices of the elements in the two-dimensional grid that correlate to sensed contact from the touch input on the touch-screen display. The stationary baseline level can be subtracted out so that the elements in an area around the sensor map that are not detected as part of the touch input are normalized to a zero level.

[0015] The contact geometry service 122 is also implemented to model the sensor map 126 that correlates to the touch input 112 as a Gaussian distribution, with a probabilistic distribution function as in equation (1):

$$p(\boldsymbol{x}) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \qquad (1)$$

[0016] The variable x = (x,y) is an index vector into the two-dimensional sensor map. The parameter μ is the mean, and the parameter Σ is a 2x2 matrix of the covariance matrix. The contact geometry service 122 can determine the parameters μ and Σ so that the probability density function (Gaussian PDF) best fits the sensor map s[x][y] that represents the contact shape of the touch input 112. To do so, the contact geometry service is implemented to perform the maximum likelihood estimation (MLE) to derive the following equations (2) and (3):

$$\widehat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=0}^{N-1} \boldsymbol{x_i} \qquad (2)$$

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{i=0}^{N-1} (\boldsymbol{x_i}-\widehat{\boldsymbol{\mu}})(\boldsymbol{x_i}-\widehat{\boldsymbol{\mu}})^{\mathrm{T}} \qquad (3)$$

[0017] The parameter N is the total number of sample points when performing the MLE. In this implementation, the value of s[x][y] is treated as a histogram of all the samples at a particular index point (x,y). As such, the contact geometry service can derive N as in equation (4):

$$\boldsymbol{N} = \sum_{y=0}^{H-1}\sum_{x=0}^{W-1} s[x][y] \qquad (4)$$

[0018] The equations (2) and (3) can be rewritten in terms of a weighted sum with s[x][y] as the weight, as in the following equations (5) and (6):

$$\widehat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{y=0}^{H-1}\sum_{x=0}^{W-1} s[x][y]\, \boldsymbol{x} \qquad (5)$$

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{y=0}^{H-1}\sum_{x=0}^{W-1} s[x][y]\, (\boldsymbol{x}-\widehat{\boldsymbol{\mu}})(\boldsymbol{x}-\widehat{\boldsymbol{\mu}})^{\mathrm{T}} \qquad (6)$$

[0019] Although the summations are now over the entire two-dimensional grid, the summation can be processed and determined quickly since s[x][y] of the sensor map is non-zero only near the touch input. Note that the parameter $\widehat{\boldsymbol{\mu}}$ is the center of mass of the touch input, and the covariance matrix $\widehat{\boldsymbol{\Sigma}}$ designates the constant-level contours of the Gaussian distribution, which is the shape of an ellipse. In embodiments, the ellipse represents the contact shape of the touch input. Generally, the contact shape of the touch input is irregular in shape, and the contact geometry service 122 is implemented to determine the ellipse of a size and rotation angle that approximately encompass the elements 128 of the sensor map 126. The contact geometry service determines the ellipse (also referred to as the “best-fit ellipse”) from the Gaussian distribution.

[0020] In embodiments, the contact geometry service 122 is implemented to determine the ellipse shape from the covariance matrix $\hat{\Sigma}$, recognizing that the two main axis (*e.g.*, minor axis and major axis) of the ellipse correspond to the two Eigenvectors of $\hat{\Sigma}$ that each have a length proportional to the square root of the corresponding Eigen values. Accordingly, the contact geometry service solves the following Eigen problem in equation (7):

$$\hat{\boldsymbol{\Sigma}}\,\boldsymbol{\varphi} = \boldsymbol{\Lambda}\boldsymbol{\varphi} \qquad (7)$$

[0021] The parameter Λ=diag(λ0,λ1) is the 2x2 diagonal matrix of Eigen values, and the parameter φ is the Eigen vector matrix of columns that correspond to λ0 and λ1. For this 2x2 Eigen problem, there exists an exact solution, and the two Eigen values can be determined by solving the following quadratic equation (8):

$$\lambda^2 - \boldsymbol{Tr}(\hat{\boldsymbol{\Sigma}})\lambda + |\hat{\boldsymbol{\Sigma}}| = 0 \qquad (8)$$

[0022] FIG. 2 illustrates examples 200 of the sensor map 126 and a determined ellipse that approximately encompasses the elements 128 of the sensor map. As shown at 202, an ellipse 204 that corresponds to the contact shape of the touch input 112 is defined by the two axis vectors 206, 208 that are determined by scaling the two Eigen vectors by the squared root of the corresponding Eigen values $\Lambda^{1/2}\varphi$. The contact geometry service 122 is implemented to globally scale the Eigen values $\Lambda^{1/2}\varphi$ so that the resulting angular contact geometry fits the actual contact shape of the touch input, and an appropriate constant-level contour is selected for the shape matching. In practice, a scaling factor $\boldsymbol{\alpha}$ can also be selected so that the area of the scaled ellipse numerically matches to the actual contact shape of the touch input from s[x][y] of the sensor map 126. As shown at 210, the ellipse 204 can also be represented as a rectangle 212 that bounds the ellipse, where the rectangle is defined by a height, a width, and a rotation angle.

[0023] When the two Eigen values are very close, the resulting Eigen vectors may be less stable in their respective directions, and a small perturbation in the sample configuration could substantially change the directions of the main axis. In an implementation, such a case may be noticeable when the contact shape of a touch input is near a round shape and any noise on the edge can change the angle of the contact geometry. Accordingly, it should be noted that for embodiments of angular contact geometry, the shape of the ellipse is the primary consideration, rather than the respective directions of the axis of the ellipse. In the case of a near round shape, the respective directions of the two axis become much less of a consideration, and therefore their instability is much less of a factor, or is not an issue when determining the angular contact geometry.

[0024] In embodiments, a quality metric for angular contact geometry can be implemented to justify or validate the determined ellipse that approximately encompasses the elements 128 of the sensor map 126. The contact geometry service 122 can be implemented to determine an inaccuracy ratio (IAR) for an ellipse. As shown at 214, the IAR for an ellipse 216 of a particular size and rotation angle is determinable from regions 218, 220 that are within the bounds of the ellipse, but are not part of the sensor map. The regions are also referred to as non-overlapping regions, and are the regions within the bounds of the ellipse that do not overlap an element of the sensor map.

[0025] Theoretically, an ellipse can be selected based on the ellipse that has an optimal IAR of the smallest value, where the ellipse with the smallest value IAR encompasses most of the sensor map with the least non-overlapping region area. For example, an ellipse may not be a good fit to represent the contact shape of a touch input based on the rotation angle of the ellipse. Alternatively or in addition, an ellipse may not be a good fit to represent the contact shape of a touch input based on the size of the ellipse being either too large or too small. The quality metric inaccuracy ratio (IAR) is defined in equation (9) as:

$$\text{InAccuracy Ratio} = \frac{\text{Total non overlapping areas between actual contact and the model}}{\text{Total Area of the model}}$$

[0026] Generally, a poor fit of the model ellipse to the actual contact shape of the touch input may result in areas and/or orientations that don't match, and the metric inaccuracy ratio of equation (9) reflects both cases. For the case of an approximate round shape, the non-overlapping area is typically small regardless of the orientation of the axis. The shape of the ellipse may not always fit exactly to the actual contact shape. As such, use of the inaccuracy ratio as a practical quality measure for implementations of angular contact geometry is offset with an optimal inaccuracy ratio (optimal IAR), which is defined in equation (10) as:

$$OptimalIAR = \min_{\text{model} \in \text{All Models}}(IAR(model)) \quad (10)$$

[0027] The "model" in this example is the ellipse, and the process of determining the optimal IAR can be performed off-line when searching for possible ellipses to find the optimal value. This quantity reflects the inherent discrepancy between the actual contact shape of the touch input and the model ellipse, and thus, can be subtracted out from the inaccuracy ratio. A normalized inaccuracy ratio as defined in equation (11) below can then be utilized, and a smaller value implies higher quality in implementations of angular contact geometry:

$$NormalizedIAR = OptimizedIAR - IAR \qquad (11)$$

[0028] Example method 300 is described with reference to FIG. 3 in accordance with one or more embodiments of angular contact geometry. Generally, any of the services, functions, methods, procedures, components, and modules described herein can be implemented using software, firmware, hardware (*e.g.*, fixed logic circuitry), manual processing, or any combination thereof. A software implementation represents program code that performs specified tasks when executed by a computer processor. The example methods may be described in the general context of computer-executable instructions, which can include software, applications, routines, programs, objects, components, data structures, procedures, modules, functions, and the like. The program code can be stored in one or more computer-readable storage media devices, both local and/or remote to a computer processor. The methods may also be practiced in a distributed computing environment by multiple computer devices. Further, the features described herein are platform-independent and can be implemented on a variety of computing platforms having a variety of processors.

[0029] FIG. 3 illustrates example method(s) 300 of angular contact geometry. The order in which the method blocks are described are not intended to be construed as a limitation, and any number of the described method blocks can be combined in any order to implement a method, or an alternate method.

[0030] At block 302, touch input sensor data is recognized as a touch input on a touch-screen display. For example, the touch input module 114 at the computing device 102 (FIG. 1) recognizes the touch input sensor data 116 as various inputs or combinations of inputs, such as the touch input 112 on the touch-screen display 110 of the computing device.

[0031] At block 304, a sensor map is generated from the touch input sensor data, where the sensor map represents the touch input. For example, the contact geometry service 122 at the computing device 102 generates the sensor map 126 from the touch input sensor data 116, and the sensor map represents the touch input 112. The sensor map can be generated as a two-dimensional array s[x][y] of elements that correlate to sensed contact from the touch input on the touch-screen display. A stronger sensor signal of the touch input sensor data indicates more contact with an element 128 in the sensor map.

[0032] At block 306, the sensor map is modeled as a Gaussian distribution with variables computed based on a weighted average. For example, the contact geometry service 122 at the computing device 102 models the sensor map 126 that correlates to the touch input 112 as a Gaussian distribution, with a probabilistic distribution function having variables computed based on a weighted average.

[0033] At block 308, an ellipse is determined that approximately encompasses elements of the sensor map. For example, the contact geometry service 122 at the computing device 102 determines an ellipse 204 as generated by the Gaussian distribution, and the ellipse approximately encompasses the elements 128 of the sensor map 126. Generally, the contact shape of a touch input may be irregular in shape, and the contact geometry service determines the ellipse of a size and rotation angle that approximately encompasses the elements of the sensor map.

[0034] At block 310, an inaccuracy ratio (IAR) is determined to validate the determined ellipse. For example, the contact geometry service 122 at the computing device 102 determines an inaccuracy ratio (IAR) to validate the determined ellipse size and rotation angle. The IAR for an ellipse of a particular size and rotation angle is determined from one or more regions that are within the bounds of the ellipse, but are not part of the sensor map. The non-overlapping regions are the regions within the bounds of the ellipse that do not overlap an element of the sensor map. A best-fit ellipse has an optimal IAR of the smallest value and encompasses most of the sensor map with the least non-overlapping region area, and the best-fit ellipse represents the contact shape of the touch input.

[0035] At block 312, the ellipse is represented as a rectangle that bounds the ellipse, where the rectangle is defined by a height, a width, and a rotation angle. For example, the contact geometry service 122 at the computing device 102 represents the best-fit ellipse 204 as the rectangle 212 that bounds the ellipse, and the rectangle is defined by a height, a width, and a rotation angle that correlates with the ellipse.

[0036] At block 314, a minor axis and a major axis of the rectangle are determined that represents the ellipse and, at block 316, the axis vectors of the minor axis and the major axis are scaled to correlate a geometry of the ellipse with the contact shape of the touch input.

[0037] At block 318, HID reports are generated from the touch input sensor data and the determined angular contact geometry. For example, the touch input module 114 at the computing device 102 generates the HID reports 120 (*i.e.*, human interface device reports). The HID reports include a time and position data, as well as determined angular contact geometry data, that correlates to touch inputs on the touch-screen display of the computing device.

[0038] FIG. 4 illustrates various components of an example device 400 that can be implemented as any of the devices, or services implemented by devices, described with reference to the previous FIGs. 1-3. In embodiments, the device may be implemented as any one or combination of a fixed or mobile device, in any form of a consumer, computer, portable, user, communication, phone, navigation, television, appliance, gaming, media playback, and/or electronic device. The device may also be associated with a user (*i.e.*, a person) and/or an entity that operates the device such that a device describes logical devices that include users, software, firmware, hardware, and/or a combination of devices.

[0039] The device 400 includes communication devices 402 that enable wired and/or wireless communication of device data 404, such as received data, data that is being received, data scheduled for broadcast, data packets of the data, etc. The device data or other device content can include configuration settings of the device, media content stored on the device, and/or information associated with a user of the device. Media content stored on the device can include any type of audio, video, and/or image data. The device includes one or more data inputs 406 via which any type of data, media content, and/or inputs can be received, such as user-selectable inputs and any other type of audio, video, and/or image data received from any content and/or data source.

[0040] The device 400 also includes communication interfaces 408, such as any one or more of a serial, parallel, network, or wireless interface. The communication interfaces provide a connection and/or communication links between the device and a communication network by which other electronic, computing, and communication devices communicate data with the device.

[0041] The device 400 includes one or more processors 410 (*e.g.*, any of microprocessors, controllers, and the like) which process various computer-executable instructions to control the operation of the device. Alternatively or in addition, the device can be implemented with any one or combination of software, hardware, firmware, or fixed logic circuitry that is implemented in connection with processing and control circuits which are generally identified at 412. In embodiments, the device 400 can also include a touch input module 414 that is implemented to recognize touch input sensor data. Although not shown, the device can include a system bus or data transfer system that couples the various components within the device. 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.

[0042] The device 400 also includes one or more memory devices 416 (*e.g.*, computer-readable storage media) that enable data storage, such as random access memory (RAM), non-volatile memory (*e.g.*, read-only memory (ROM), flash memory, etc.), and a disk storage device. A disk storage device may be implemented as any type of magnetic or optical storage device, such as a hard disk drive, a recordable and/or rewriteable disc, and the like. The device may also include a mass storage media device.

[0043] Computer readable media can be any available medium or media that is accessed by a computing device. By way of example, and not limitation, computer readable media may comprise storage media and communication media. Storage media include volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information, such as computer-readable instructions, data structures, program modules, or other data. Storage media 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, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store information and which can be accessed by a computer.

[0044] Communication media typically embody computer-readable instructions, data structures, program modules, or other data in a modulated data signal, such as carrier wave or other transport mechanism. Communication media also include any information delivery media. A modulated data signal 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.

[0045] A memory device 416 provides data storage mechanisms to store the device data 404, other types of information and/or data, and various device applications 418. For example, an operating system 420 can be maintained as a software application with the memory device and executed on the processors. The device applications may also include a device manager, such as any form of a control application, software application, signal processing and control module, code that is native to a particular device, a hardware abstraction layer for a particular device, and so on. In this example, the device applications 418 include a gesture recognition application 422 and a contact geometry service 424 that implement embodiments of angular contact geometry as described herein.

[0046] The device 400 also includes an audio and/or video processing system 426 that generates audio data for an audio system 428 and/or generates display data for a display system 430. The audio system and/or the display system may include any devices that process, display, and/or otherwise render audio, video, display, and/or image data. Display data and audio signals can be communicated to an audio device and/or to a display device via an RF (radio frequency) link, S-video link, composite video link, component video link, DVI (digital video interface), analog audio connection, or other similar communication link. In implementations, the audio system and/or the display system are external components to the device. Alternatively, the audio system and/or the display system are integrated components of the example device, such as an integrated touch-screen display.

[0047] Although embodiments of angular contact geometry have been described in language specific to features and/or methods, the subject of the appended claims is not necessarily limited to the specific features or methods described. Rather, the specific features and methods are disclosed as example implementations of angular contact geometry.

CLAIMS

1. A method, comprising:

recognizing touch input sensor data as a touch input on a touch-screen display;

generating a sensor map from the touch input sensor data, the sensor map representing the touch input; and

determining an ellipse that approximately encompasses the sensor map, the ellipse representing a contact shape of the touch input.

2. A method as recited in claim 1, wherein:

the contact shape of the touch input is irregular in shape; and

the ellipse is of a size and rotation angle that approximately encompasses the sensor map.

3. A method as recited in claim 2, further comprising determining an inaccuracy ratio (IAR) to validate the ellipse, the IAR of the ellipse determined from one or more regions that are within the bounds of the ellipse, but are not part of the sensor map.

4. A method as recited in claim 1, wherein the sensor map is generated as a two-dimensional array of elements that correlate to sensed contact from the touch input on the touch-screen display.

5. A method as recited in claim 4, wherein a stronger sensor signal of the touch input sensor data indicates more contact with an element in the sensor map.

6. A method as recited in claim 4, further comprising modeling the sensor map as a Gaussian distribution, and wherein the ellipse that approximately encompass the sensor map is determined from the Gaussian distribution.

7. A method as recited in claim 6, further comprising:

determining a minor axis and a major axis of the ellipse from a covariance matrix; and

scaling axis vectors of the minor axis and the major axis to correlate a geometry of the ellipse with the contact shape of the touch input.

8. A computing device, comprising:

a touch-screen display;

a touch input module configured to recognize touch input sensor data as a touch input on the touch-screen display;

at least a memory and a processor to implement a contact geometry service configured to:

generate a sensor map from the touch input sensor data, the sensor map representing the touch input; and

determine an ellipse that approximately encompasses the sensor map, the ellipse representing a contact shape of the touch input.

9. A computing device as recited in claim 8, wherein:

the contact shape of the touch input is irregular in shape; and

the ellipse is of a size and rotation angle that approximately encompasses the sensor map.

10. A computing device as recited in claim 9, wherein the contact geometry service is further configured to determine an inaccuracy ratio (IAR) to validate the ellipse, the IAR of the ellipse being determinable from one or more regions that are within the bounds of the ellipse, but are not part of the sensor map.

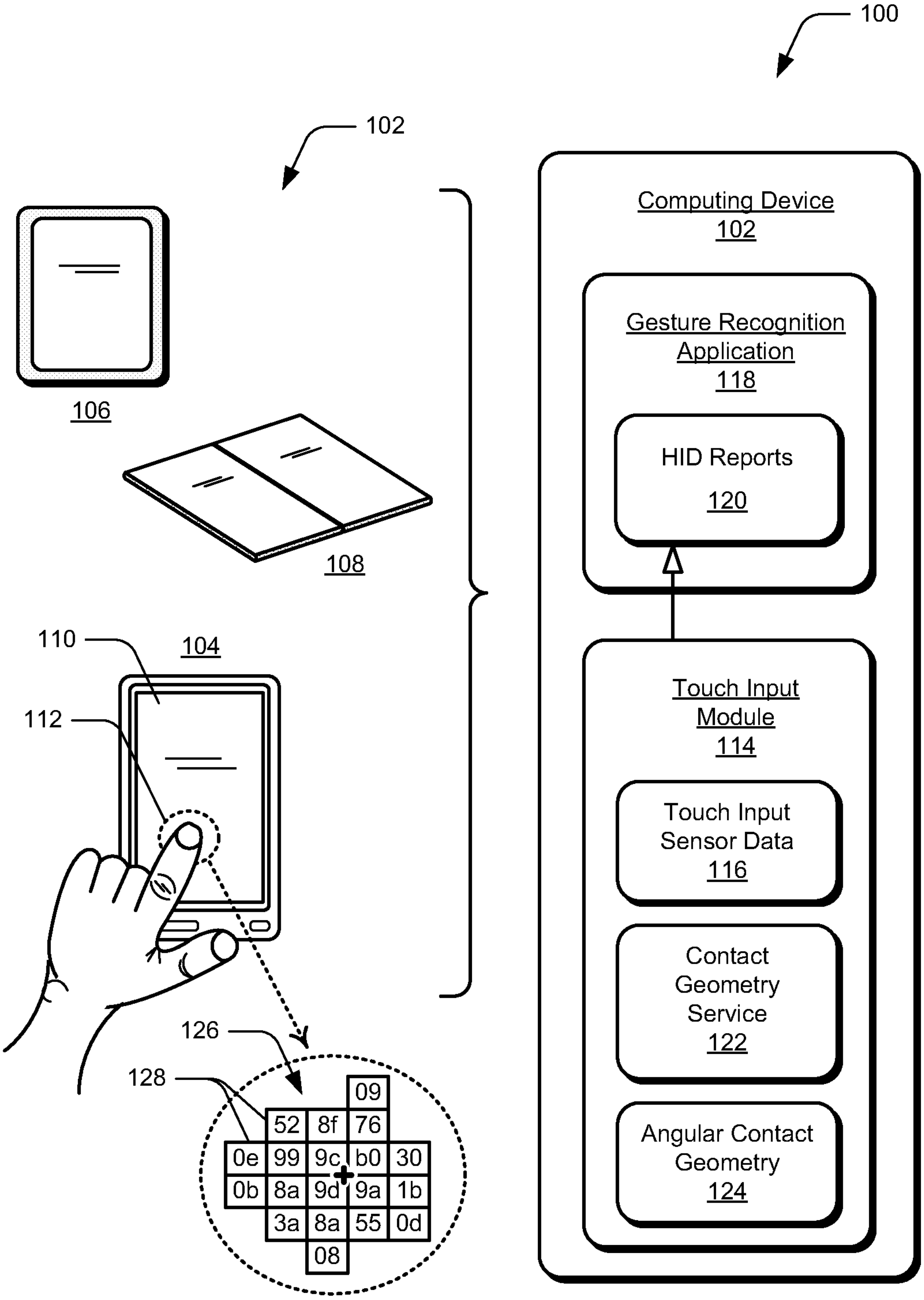

100
102
106
108
110
104
112
126
128
09
52 8f 76
0e 99 9c b0 30
0b 8a 9d 9a 1b
3a 8a 55 0d
08
Computing Device 102
Gesture Recognition Application 118
HID Reports 120
Touch Input Module 114
Touch Input Sensor Data 116
Contact Geometry Service 122
Angular Contact Geometry 124

FIG. 1

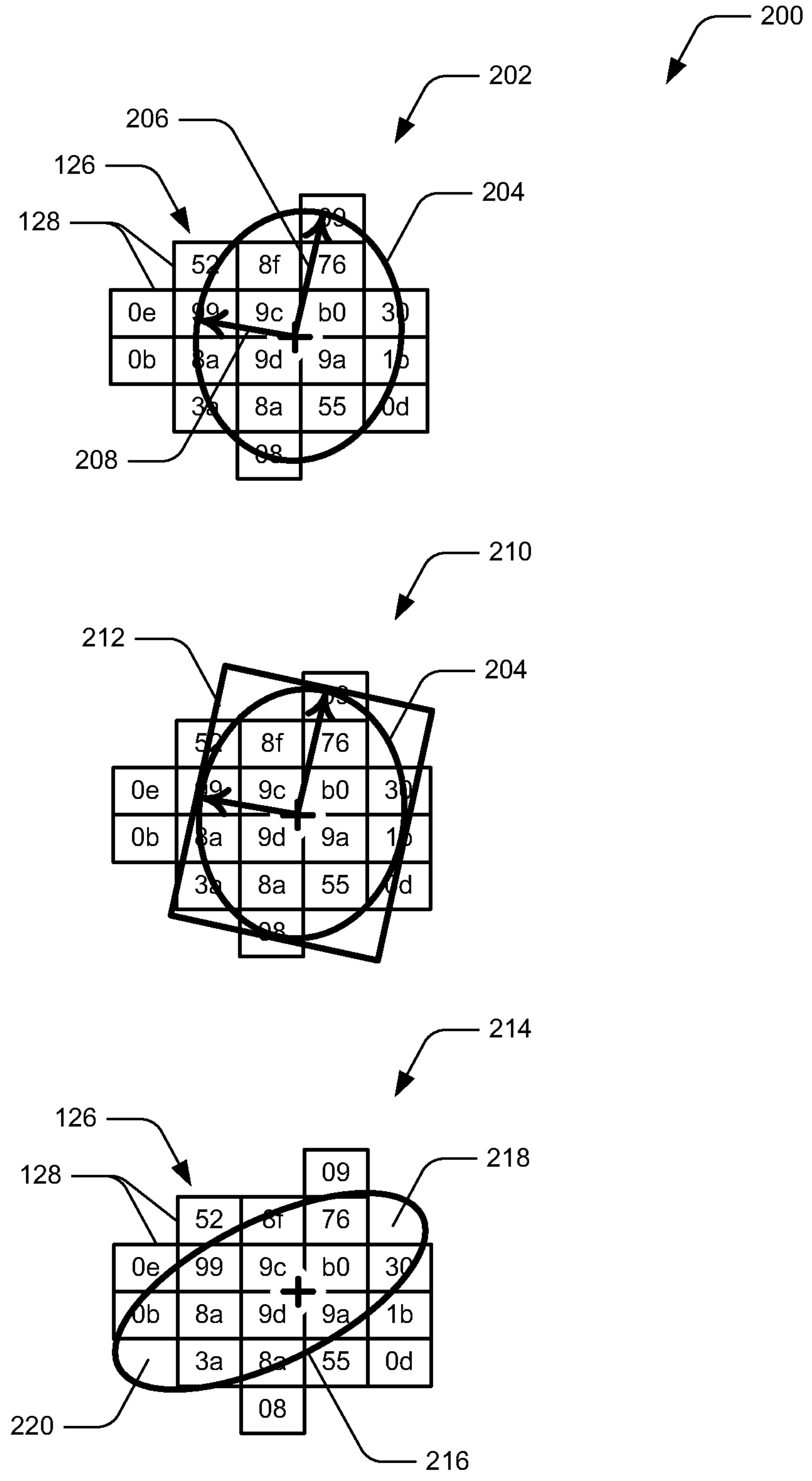

200
202
206
126
128
204
52
8f
76
0e
99
9c
b0
30
0b
8a
9d
9a
1b
3a
8a
55
0d
08
208
210
212
204
214
126
128
09
218
220
216

FIG. 2

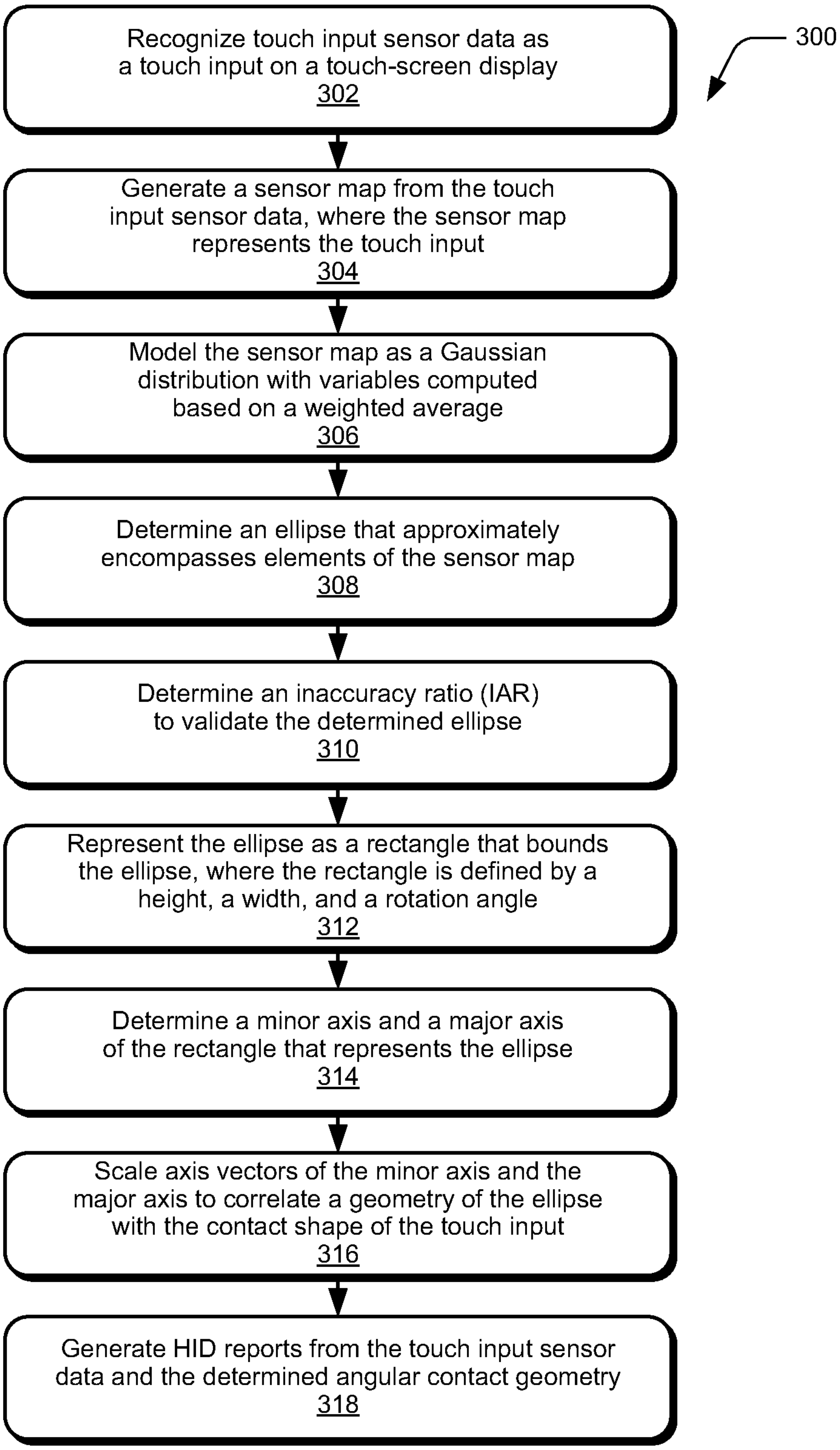

300
Recognize touch input sensor data as a touch input on a touch-screen display 302
Generate a sensor map from the touch input sensor data, where the sensor map represents the touch input 304
Model the sensor map as a Gaussian distribution with variables computed based on a weighted average 306
Determine an ellipse that approximately encompasses elements of the sensor map 308
Determine an inaccuracy ratio (IAR) to validate the determined ellipse 310
Represent the ellipse as a rectangle that bounds the ellipse, where the rectangle is defined by a height, a width, and a rotation angle 312
Determine a minor axis and a major axis of the rectangle that represents the ellipse 314
Scale axis vectors of the minor axis and the major axis to correlate a geometry of the ellipse with the contact shape of the touch input 316
Generate HID reports from the touch input sensor data and the determined angular contact geometry 318

FIG. 3

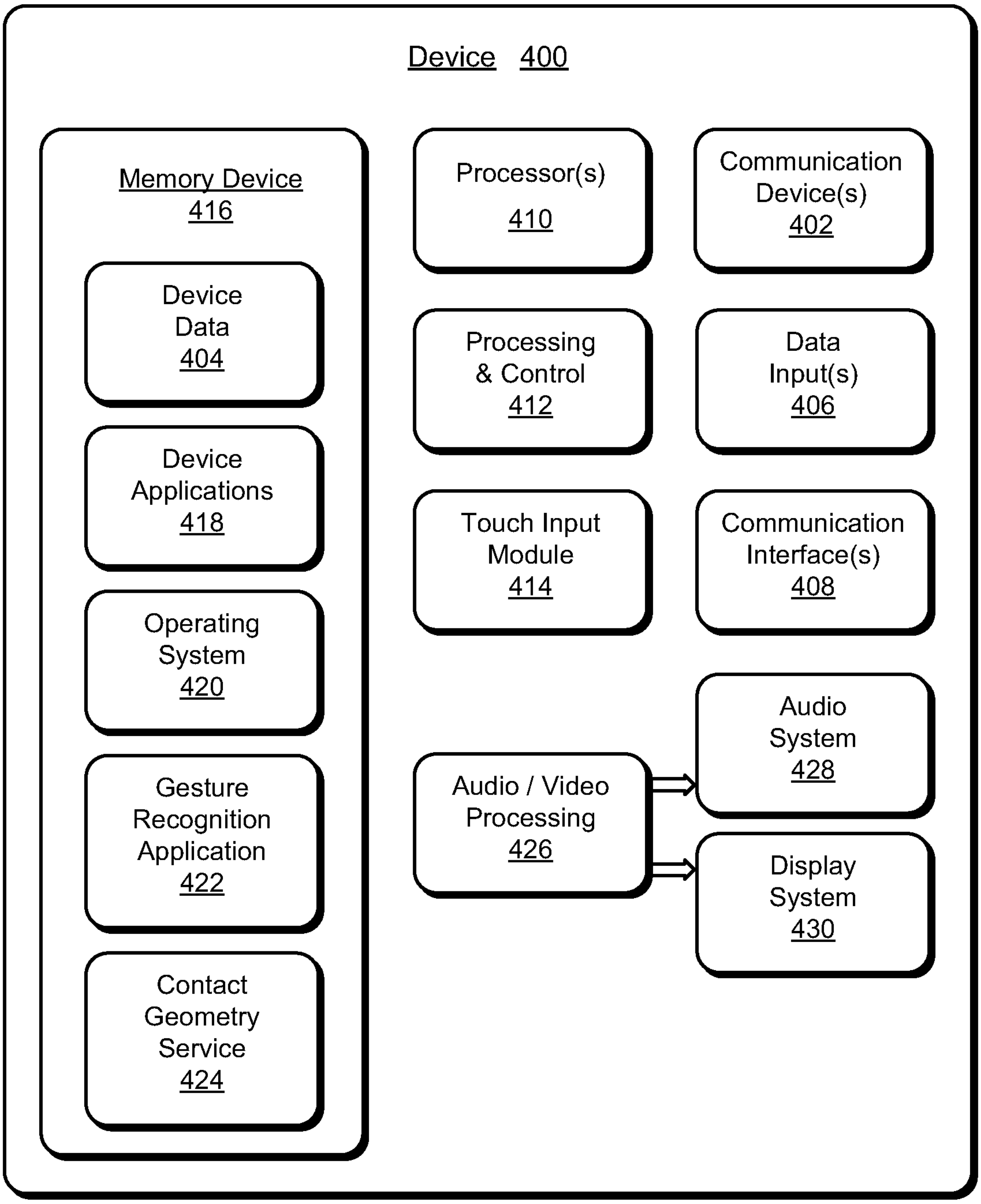

Device 400
Memory Device 416
Device Data 404
Device Applications 418
Operating System 420
Gesture Recognition Application 422
Contact Geometry Service 424
Processor(s) 410
Processing & Control 412
Touch Input Module 414
Audio / Video Processing 426
Communication Device(s) 402
Data Input(s) 406
Communication Interface(s) 408
Audio System 428
Display System 430

FIG. 4