



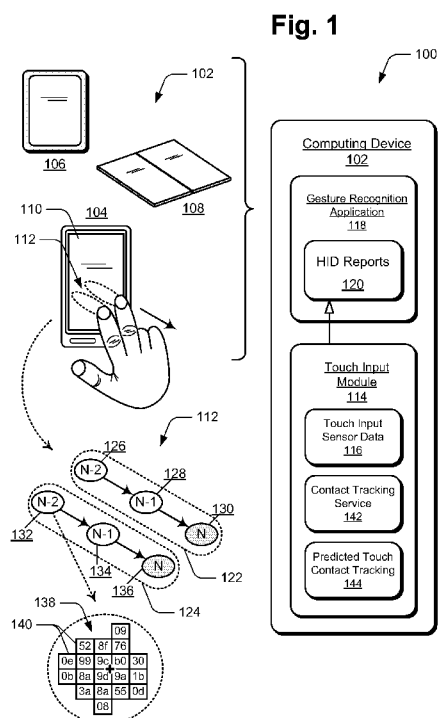
- (51) **International Patent Classification:**
G06F 3/03 (2006.01) *G06F 3/041* (2006.01)
- (21) **International Application Number:**
PCT/US2012/024780
- (22) **International Filing Date:**
11 February 2012 (11.02.2012)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
61/442,222 12 February 2011 (12.02.2011) US
13/152,991 3 June 2011 (03.06.2011) US
- (71) **Applicant (for all designated States except US):** MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, WA 98052-6399 (US).
- (72) **Inventors:** ZHAO, Weidong; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, WA 98052-6399 (US). STEVENS, David, A.; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, WA 98052-6399 (US). UZELAC, Aleksandar; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, WA

98052-6399 (US). BENKO, Hrvoje; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, WA 98052-6399 (US). MILLER, John L.; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, WA 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, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,

[Continued on next page]

(54) **Title:** PREDICTION-BASED TOUCH CONTACT TRACKING

(57) **Abstract:** In embodiments of prediction-based touch contact tracking, touch input sensor data is recognized as a series of components of a contact on a touch-screen display. A first component of the contact can be identified, and a second component can be determined to correlate to the contact. The first component and the second component can then be associated to represent a tracking of the contact. Subsequent components of the contact can be determined and associated with the previous components of the contact to further represent the tracking of the contact.



SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

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

Declarations under Rule 4.17:

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

Published:

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

PREDICTION-BASED TOUCH CONTACT TRACKING**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 touch signal processing to track touch and gesture inputs that are identified from successive frames of sensor image data. Touch contacts on a touch-screen display represent the motion trace of a gesture, such as when a user uses his or her fingers to contact a touch-screen display and gesture while maintaining the contact with the display. A failure to correctly track and interpret the motion trace of a touch contact for a gesture input can lead to the failure of gesture recognition operations and gesture tracking processing.

[0002] For a gesture motion that is a relatively small or short gesture, conventional tracking processing may statically match the spatially co-located touch contacts from successive frames. However, this approach is not effective for a gesture motion that is a relatively large or long gesture, such as may be typically used on a tablet computer or other type of slate form factor where fast gestures such as flicker or panning are involved. The tracking processing may not be sensitive to the complete motion of a gesture, which can result in a gesture “break”, which may then be recognized and processed as much shorter motion ranges than the actual gesture motion. Alternatively, if the tracking processing is overly sensitive to the gesture motion, this can lead to mis-tracking the touch contacts, such as when a user inputs text with a soft or virtual keyboard that is displayed on a touch-screen display.

SUMMARY

[0003] This Summary introduces simplified concepts of prediction-based touch contact tracking, 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.

[0004] Prediction-based touch contact tracking is described. In embodiments, touch input sensor data is recognized as a series of components of a contact on a touch-screen display. A first component of the contact can be identified, and a second component can be determined to correlate to the contact. The first component and the second component can then be associated to represent a tracking of the contact. Subsequent components of the contact can be determined and associated with the previous components of the contact to further represent the tracking of the contact.

[0005] In other embodiments, a same identifier is assigned to the first component and to the second component of the contact, as well as to the subsequent components that represent the tracking of the contact to associate all of the components with the contact. A predicted component position that is correlated to a subsequent component of the contact can be validated based on a nearest-neighbor criteria that evaluates distance from the predicted component position to one or more additional components of the contact. Additionally, a component can be determined as a component of the contact based on a min-max distance determination between the components of the contact.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] Embodiments of prediction-based touch contact tracking 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 prediction-based touch contact tracking can be implemented.

FIG. 2 illustrates example method(s) of prediction-based touch contact tracking in accordance with one or more embodiments.

FIG. 3 illustrates an example architecture of a contact tracking service in accordance with one or more embodiments.

FIG. 4 illustrates an example architecture for processing prediction-based touch contact tracking in accordance with one or more embodiments.

FIG. 5 illustrates a high-level flow diagram for prediction-based touch contact tracking in accordance with one or more embodiments.

FIG. 6 illustrates an example of a merge-on-landing in accordance with one or more embodiments.

FIG. 7 illustrates an example of motion prediction for a decision region in accordance with one or more embodiments.

FIG. 8 illustrates an example of min-max distance determination for a multiple level maximum distance determination in accordance with one or more embodiments.

FIG. 9 illustrates an example of crossing trajectories that may indicate unwanted component associations in accordance with one or more embodiments.

FIG. 10 illustrates an example of how to compute and account for crossing trajectories as shown in FIG. 9 in accordance with one or more embodiments.

FIG. 11 illustrates example method(s) of prediction-based touch contact tracking in accordance with one or more embodiments.

FIG. 12 illustrates example method(s) of prediction-based touch contact tracking in accordance with one or more embodiments.

FIG. 13 illustrates example method(s) of prediction-based touch contact tracking in accordance with one or more embodiments.

FIG. 14 illustrates various components of an example device that can implement embodiments of prediction-based touch contact tracking.

DETAILED DESCRIPTION

[0007] Embodiments of prediction-based touch contact tracking 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 tracked and/or processed. Prediction-based touch contact tracking uses the prediction of components at touch positions from previous frames, and the tracking can be reliably achieved in the case of high motion while at the same time, mis-tracking in the case of a soft keyboard can be avoided. Touch contact tracking can be based on the predicted positions of previously identified components of a contact, with a set of rules defining the scope of validity of the tracking obtained by the prediction. If a prediction does not exist, a solution is implemented to initiate the prediction via a min-max distance determination (*e.g.*, an algorithm or procedure), and optionally with a penalty term for a tracking crossing.

[0008] Embodiments of prediction-based touch contact tracking can include a prediction based two-level procedure for component identification and contact tracking. A first-level of procedure is used to establish an initial association of components of a contact for gesture input tracking based on a prediction, and a second level of procedure is used to validate based on a nearest-neighbor contact mapping

criteria to generate a final association of the components of a contact. Also related is a set of rules defining the operations for multi-finger touch and gesture recognition. This may be implemented for any slate based device, tablet device, mobile phone or computer with a touch-screen display, as well as for other similar technologies such as surface, indirect touch, etc.

[0009] Embodiments also include a min-max distance determination on contact mapping to both generate the association of components of touch contacts for the prediction residuals, as well as to produce the initial prediction of component position when a prediction is not available. The procedure also accounts for penalizing gesture motion tracks that cross over or through the selection of an initial gesture input tracking. Embodiments also include an approach to process and resolve the case of multiple touch positions that alias to a single component, as well as an approach to process and resolve merge-on-landing when two or more touch positions alias to a single, larger component. An application of the techniques described herein may also be implemented to reduce touch latency by performing look-ahead prediction.

[0010] While features and concepts of prediction-based touch contact tracking can be implemented in any number of different devices, systems, environments, networks, and/or configurations, embodiments of prediction-based touch contact tracking are described in the context of the following example devices, systems, and methods.

[0011] FIG. 1 illustrates an example system 100 in which various embodiments of prediction-based touch contact tracking 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 a gesture input 112, such as a user-initiated gesture in a user interface that is displayed on the touch-screen display. In this example, the gesture input is a two finger gesture across the touch-screen display in an approximate direction indicated by the arrow, but may be a single finger gesture input, or a multi-finger gesture input (*e.g.*, three or more finger gesture input). 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. 14 to implement embodiments of prediction-based touch contact tracking.

[0012] 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 gesture 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 the touch input sensor data from the touch input module as HID reports 120 (*i.e.*, human interface device reports). The HID reports include a time and position data, as well as determined touch contact tracking, that correlates to gesture 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 gesture 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.

[0013] 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).

[0014] As described herein, a gesture input may be recognized as a user input with one or more fingers on a touch-screen display of a device, and the gesture input includes one or more contacts that each correlate to the input of a finger on the touch-screen display. In the FIG. 1 example, the two-finger gesture input 112 includes two contacts identified as a first contact 122 that correlates to a first finger of the gesture input, and a second contact 124 that correlates to a second finger of the gesture input.

[0015] The gesture input data is received as a series of frames, and a frame includes a component that represents one touch position of a contact (*e.g.*, along a

gesture input that is one finger). For a two-finger gesture input, a frame can include a component of a first contact that correlates to the input of a first finger, and include a component of a second contact that correlates to the input of a second finger (and so on for more than a two-finger gesture input).

[0016] In the FIG. 1 example, the first contact 122 of the gesture input 112 includes successive components, such as component 126, component 128, and component 130 at different touch positions along the first contact. Similarly, the second contact 124 of the gesture input 112 includes successive components, such as component 132, component 134, and component 136 at different touch positions along the second contact. Accordingly, a first frame of the two-finger gesture input includes the component 126 and the component 132 of the respective first and second contacts at N-2 in the series of components. Similarly, a next frame of the gesture input at N-1 includes the component 128 and the component 134 of the respective first and second contacts, and a current frame of the gesture input at N includes the component 130 and the component 136 of the respective first and second contacts.

[0017] Therefore, a contact of a gesture input spans multiple frames and includes the components from each successive frame that have been identified as correlating to the contact, or to a section of the contact. A component represents a touch position of a contact in a frame (*e.g.*, after the component has been identified as correlating to the contact). As described in embodiments, a component can be identified as correlating to a particular contact based on a nearest-neighbor contact mapping criteria that evaluates distance between component positions. However, if the nearest-neighbor contact mapping does not identify a component to one of the existing contacts, then a new contact of the gesture input can be generated to represent the tracking of an additional finger used to gesture on the touch-screen display.

[0018] The touch input module 114 recognizes the touch input sensor data 116 as the series of components of the two contacts 122, 124 of the gesture input 112 on the touch-screen display 110 of the computing device 102. In embodiments, the touch input module 114 is implemented to generate a sensor map 138 from the touch input sensor data 116 for each component of each contact. A sensor map represents an individual component of a contact, such as when a user initiates the gesture input 112 on the touch-screen display 110. In this example, the sensor map includes elements 140 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 touch contact with an element in the sensor map. The sensor map can be generated as a two-dimensional array, and array indices of the elements in the two-dimensional grid correlate to sensed touch contact from the gesture 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 contact are normalized to a zero level.

[0019] The computing device 102 also includes a contact tracking service 142 that is implemented to determine predicted touch contact tracking 144 that corresponds to one or more contacts of a gesture input on the touch-screen display 110, such as the gesture input 112. The contact tracking 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 tracking service can also be implemented as firmware on dedicated sensor device hardware in the computing device. In this example, the contact tracking service is shown implemented as a component of the touch input module 114. Alternatively, the contact tracking service may be implemented as an independent software application or service to predict touch contact tracking.

[0020] In embodiments, the contact tracking service 142 is implemented to perform various procedures and/or algorithms of prediction-based touch contact tracking. The contact tracking service can identify and predict components of the gesture input that are mapped (*e.g.*, correlated, associated) as the two contacts. The components 126-130 represent a tracking of the first contact 122, and the components 132-136 represent a tracking of the second contact 124. The components that are identified as correlating to a particular contact are all assigned the same identifier. For example, the components 126-130 of the first contact 122 are all assigned the same first identifier, and the components 132-136 of the second contact 124 are all assigned the same second identifier, where the first and second identifiers are different to distinguish the separate contacts. As further described below, the contact tracking service can validate that a predicted component position correlates to a subsequent component of a contact based on a nearest-neighbor criteria that evaluates distance from the predicted component position to the additional components of the contact. Additionally, the contact tracking service can determine that components correlate to a particular contact based on a min-max distance determination between the components of the contact.

[0021] Example methods 200, 1100, 1200, and 1300 are described with reference to respective FIGs. 2, 11, 12, and 13 in accordance with one or more embodiments of prediction-based touch contact tracking. 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.

[0022] FIG. 2 illustrates example method(s) 200 of prediction-based touch contact tracking. 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.

[0023] At block 202, a gesture input is recognized on a touch-screen display. For example, the touch input module 114 (FIG. 1) at the computing device 102 recognizes various inputs or combinations of inputs, such as the gesture input 112 on the touch-screen display 110 of the example device 104. In this example, the gesture input 112 is a two-finger gesture input that includes the two contacts identified as the first contact 122 that correlates to a first finger of the gesture input, and the second contact 124 that correlates to the second finger of the gesture input.

[0024] At block 204, prediction-based touch contact tracking is determined that corresponds to the gesture input. At block 206, a next component position of a contact is predicted based on one or more previous components of the contact. At block 208, a next component of the contact is predicted based on a min-max distance determination between components of the contact. At block 210, the components of the contact of the gesture input are mapped to represent a tracking of the contact. At block 212, an initial association of the components is predicted to establish the

tracking of the contact. At block 214, a final association of the components of the contact is validated based on a nearest-neighbor criteria. At block 216, multiple touch positions that alias to a single component of the contact are resolved. At block 218, a merge-on-contact is resolved for an additional touch position that merges with a component of the contact.

[0025] For example, the contact tracking service 142 (FIG. 1) at the computing device 102 determines prediction-based touch contact tracking that corresponds to the gesture input 112 on the touch-screen display 110 of the example device 104. The contact tracking service also predicts a next component position of the contact of the gesture input based on one or more previous components of the contact of the gesture input; predicts a next component of the contact of the gesture input based on a min-max distance determination between components of the contact; maps (*e.g.*, correlates, associates) the components of the contact of the gesture input to represent a tracking of the contact; predicts an initial association of the components to establish the tracking of the contact; validates a final association of the components of the contact based on a nearest-neighbor criteria; resolves multiple touch positions that alias to a single component of the contact; and/or resolves a merge-on-contact for an additional touch position that merges with a component of the contact.

[0026] FIG. 3 illustrates an example implementation and various components of the contact tracking service 142 that is described with reference to FIG. 1, and which is implemented for touch signal processing. FIG. 4 illustrates an example architecture 400 for processing prediction-based touch contact tracking and is described with reference to the touch input module 114 shown in FIG. 1, and with reference to the contact tracking service 142 shown in FIGs. 1 and 3. FIG. 5 illustrates a high-level flow diagram 500 of prediction-based touch contact tracking implemented by a touch contact tracker module 300 (also referred to as the “contact tracker”) that is shown as a component of the contact tracking service 142 in FIG. 3.

[0027] Touch input sensor data 116 is input at 402 (FIG. 4) (*e.g.*, received at the touch input module 114), and a sensor unit of the touch input module generates sensor maps 302 (FIG. 3) from the touch input sensor data for each component of a contact of a gesture input. A baseline level can be subtracted out, or otherwise removed, for elements around a sensor map that are normalized to a zero level at 404. For example, the sensor map 138 can be generated as a two-dimensional grid with a pitch of 5mm and an element depth of eight to sixteen bits. A connected components

analyzer 304 is implemented for connected component analysis at 406. The connected components analyzer receives the normalized sensor maps 420 as an input and generates an array 306 of the connected contact components that represent the touch contacts replaced by components of a gesture input. The touch contact tracker module 300 is implemented for component identification and contact tracking at 408 to associate (*e.g.*, correlate or map) the components from previous frames and the components at touch positions in a current frame that represent the traces (*e.g.*, tracking) of a continuous gesture motion, such as the tracking of the first contact 122 and the tracking of the second contact 124.

[0028] In FIG. 5, a set of touch contacts from a previous frame are examined for predictability when each new frame is received as input at block 502. Touch contacts that are still in the initial tracking stage (*i.e.*, “yes” from block 502), and before a sufficient history of touch contact association has been established, are input to a touch contact map module 308 (also referred to as the “contact mapper”) for min-max contact mapping at block 504. The min-max contact mapping generates an initial association of mapped contact components 310, which are then output and buffered as determined components 312 with a history of association for future frames.

[0029] Touch contacts that are not initial touch contacts (*i.e.*, “no” from block 502), such as when at least two previous frames have been received, are input to a motion prediction module 314 for motion prediction at block 506 to generate touch contact predicted positions 316. These touch contact predicted positions, along with the connected components 508 of the current frame, are input into the touch contact map module 308 for nearest-neighbor contact mapping at block 510 based on the nearest-neighbor distance matching algorithm. The result of the nearest-neighbor distance matching is checked against a criterion at 512 to determine whether the components of the current frame have been successfully mapped to previous components of a contact.

[0030] If the mapping of the components is successful (*i.e.*, “yes” from block 512), then the mapped association is input to a touch contact merger module 318 at block 514. If the mapping of the components is not successful (*i.e.*, “no” from block 512), then the components are input to the touch contact map module 308 for min-max contact mapping at block 516 and a two-level combinatory mapping is invoked. With the input from the motion prediction module 314 (*i.e.*, as

output at block 506) and the connected components 508, the min-max contact mapping attempts to establish a first level association between these two sets of component positions based on a min-max distance determination, along with a set of rules involving hand and/or finger kinematic dynamics. The first level mapping association is then processed at block 518 for nearest-neighbor distance validation of the final determined contact association.

[0031] For those contacts that do not have a mapping established at the first level (*e.g.*, at block 518), they are routed over all of the components for a contact aliasing check at block 520 to determine possible aliasing, which may be merged contacts as determined at block 514 by the touch contact merger module 318. A single component may associate to multiple touch contact positions, which can occur when multiple fingers of a gesture input motion move closer to appear as a single component based on the touch input sensor data. To detect a touch contact merger, for any unmapped component after a first-level contact association, a nearest-neighbor verification can be initiated for contact associations for all of the components, and any match indicates an aliased association between a single component and multiple touch positions.

[0032] The touch contact merger module 318 processes and resolves the merged components, such as to independently check whether two touch positions have the same [x,y] grid point coordinates. A contact merger may include multiple touch positions aliased to one component, as well as a scenario of merge-on-landing when a first touch position is already sensed or detected as a first finger touches on a touch-screen display and a user lands a second finger closely next to the first one. The two touch positions may then be detected as merged together in one larger component.

[0033] FIG. 6 illustrates an example 600 of a merge-on-landing with a component 602 from a current frame in a first contact tracking 604 that associates with a component 606 from a previous frame. The example also illustrates an aliasing touch position 608 that is detected as having merged together with the other component 602 to form the one larger component 610. The merge-on-landing can be detected by evaluating conditions and characteristics of the merge-on-landing. For example, the contact-to-component mapping is 1-to-1 (*i.e.*, no contact aliasing exists); the projected touch position is within the area of the component; the size of the component is substantially larger than the size of the contact; and the predicted

position is off from the center of mass of the component, but is still within the area of the component.

[0034] Alternatively, when the contact has a high velocity, the span of the component is substantially greater than the span of the contact when both are projected to the direction normal to the velocity; and when the contact has a high velocity, the predicted position is off from the center of mass of the component – all after being projected to the direction normal to the velocity. When these conditions are all met, another contact can be generated with its center positioned diagonally to the original contact across the center of mass of the component, and this new contact is the aliasing to the component.

[0035] To resolve a merge-on-landing, the contact tracking service 142 can attempt to resolve the two touch positions in the one component 610 and track them in the merged state, as described below in a probabilistic approach. Alternatively, the contact tracking service can determine the first touch position of the component 602 based on initial touch input sensor data for the merged component and resolve the second touch position 608 as being any new touch input sensor data of the merged component.

[0036] FIG. 7 illustrates an example 700 of motion prediction for a decision region 702 that represents a predicted component position 704 of a contact given the tracking of the contact, such as a contact of the gesture input 112 on the touch-screen display 110 of the example device 104 (FIG. 1). In embodiments, the motion prediction module 314 (FIG. 3) of the contact tracking service 142 is implemented for motion prediction when the gesture input begins on the touch-screen display at position $N-2$ and continues the tracking of the contact to position $N-1$ and then to position N . The component at position N is determined as the component that is associated with the components at $N-2$ and $N-1$ based on proximity to the predicted component position 704. When there is an association for a component identified as i in previous N -frames, then the position of the component in the current frame of X_i can be predicted thru a linear combination of the previous component positions of the contact, as in the following equation:

$$\widehat{X}_i(t) = \sum_{k=1}^N a_k X_i(t-k)$$

[0037] If $N=3$ for example, one solution is a constant $a_1=2.5$, $a_2=-2$, and $a_3=0.5$ determined via the simple constant acceleration condition. If $N=2$ for

example, then a solution is $a_1 = 2$ and $a_2 = -1$. In general, these coefficients may be time-dependent variables and a more advanced technique, such as a Kalman filter, can be utilized to determine the parameters through an iterative procedure.

[0038] For each component $\hat{X}_i(t)$ of the i^{th} touch position generated in the prediction stage above, the nearest-neighbor contact mapping attempts to associate a component $X(t)$ of the current frame. A nearest-neighbor contact mapping can be resolved as described herein and/or with other techniques and algorithms. A mapped association can be established when all of the components of a current frame are considered to determine a component with an $X(t)$ that is within the two-dimensional decision region 702 centered around the predicted position $\hat{X}_i(t)$. The decision region can be constructed first with a round shape of radius r which corresponds to an area matching the actual touch shape of the i^{th} contact at the frame $t-1$. The round shape region is then modified with a velocity related expansion along the direction of the velocity, and with an expansion factor λ proportional to the norm of the velocity. This expansion accounts for the error introduced by the inaccuracy of the velocity prediction. In general, λ can have an upper bound λ_{max} to avoid erroneous association between fingers of a gesture input motion that are close together along the direction of the velocity. In one implementation, $\lambda = (1 + \lambda_{\text{max}}|v|)/(1 + |v|)$, however other choices for λ are also possible.

[0039] In practice, the procedure of verifying the nearest-neighbor contact mapping criterion can be performed in a reverse procedure. A difference vector: $d = X(t) - \hat{X}_i(t)$ is computed first, and then a reverse scaling on d is performed with the factor $1/\lambda$ along the direction of velocity vector: $v = X(t-1) - X(t-2)$. The norm of the resultant vector $\tilde{d}$ is then checked against the radius r of the decision region, and a value smaller than r indicates that an association has been determined. Another extension of the nearest-neighbor contact mapping is to implement a probabilistic approach, where a probability distribution function of the vector d is defined, and instead of passing a hard decision of which touch the component $X(t)$ belongs to, a probability distribution is passed among all of the components. This information can be passed through the HID reports so that the gesture recognition application 118 (FIG. 1) (e.g., a higher layer) can utilize the information.

[0040] In embodiments, criteria that nearest-neighbor contact mapping is successful can be established, such as to determine a successful mapping at block 512 (FIG. 5). Generally, the nearest-neighbor contact mapping generates a reliable touch

component association. However, in some distributions of touch contact velocity, an unreliable association may be determined, such as when the velocities of the touch positions involved are fast and with substantial acceleration in certain directions. The resulting predicted component locations become closer to that of neighbored components rather than the tracking contact component that should be associated with the predicted component.

[0041] A simple condition to detect this instance of an unreliable association is to determine that the total number of unassociated components after the nearest-neighbor mapping is greater than the difference between the number of components and the number of touches. Note that this is a sufficient condition in that, theoretically, there may be good mappings classified as a bad mapping. Since the majority of frames have reliable association predictions, the instances of an unreliable association (*e.g.*, a failure case) will likely be of a very small percentage (*e.g.*, less than 5%) with a negligible computational cost. If a failure case is detected, an algorithm or procedure for combinatory contact mapping can be invoked to determine a correct association, as implemented by the touch contact map module 308 of the contact tracking service.

[0042] The combinatory contact mapping can establish a reliable mapping between a given set of contacts established in a previous frame and a given set of components in the current frame. When a prediction is determined and the nearest-neighbor contact mapping fails, the combinatory contact mapping can be used to establish a first-level association between the two sets (*e.g.*, the touch positions and the components) by matching the prediction residuals between the prediction positions and that of the components. When a prediction is not determined at the initial phase of a contact life span, the combinatory contact mapping can also be used to establish the initial association. If an initial association of a contact does not exist, as may happen in the first two frames of a new contact, the touch contact map module 308 can then set the original position as the predicted position, and no prediction residual is present. The touch contact map module can include the combinatory contact mapping algorithm, the min-max distance determination algorithm for distance mapping, and a cross-trajectory suppressor for penalizing trajectories of gesture input motions that cross each other. The combinatory contact mapping algorithm can be implemented as described herein via a min-max algorithm and/or with other techniques and algorithms, such as the Hungarian Algorithm.

[0043] In embodiments, a nearest-neighbor verification for contact and component association can be implemented. The first-level association that is established with a min-max distance determination (as further described below) can be evaluated with a second-level verification process, which is similar to nearest-neighbor contact mapping. Specifically the component $X(t)$ of the current frame, after the establishment of the initial association to $\hat{X}_i(t)$ at the min-max mapping stage, is confirmed to belong to the i^{th} contact if $X(t)$ falls into a two-dimensional predicted region centered around the predicted position $\hat{X}_i(t)$ of the i^{th} contact. Generally, a difference to the nearest-neighbor contact mapping is that instead of evaluating all of the possible components for a given $\hat{X}_i(t)$, a component with the first-level association is selected for the verification process.

[0044] FIG. 8 illustrates an example 800 of min-max distance determination for a multiple level maximum distance determination. In embodiments, the touch contact map module 308 (FIG. 3) of the contact tracking service 142 is implemented to determine an optimal mapping to associate (*e.g.*, determine, correlate) the components of a previous frame to the components of a current frame such that the maximum distance between all mapping pairs within the mapping is minimized. Depending on how the min-max distance determination is applied, the position of a component of the previous frame that is being mapped from may be the predicted position of a component in a subsequent frame.

[0045] The min-max distance determination seeks to determine the mapping that has the smallest maximum distance between the components of a previous frame and the components of a current frame. In this example, a maximum distance 802 from a component 804 to a touch position 806 of a subsequent component has already been identified. A mapping is then determined from components 808 in a current frame to components 810 in a previous frame. For example, the determination for a first contact is whether to select the component association represented by the dashed line or the solid line at 812, and the determination for a second contact is whether to select the component association represented by the dashed line or the solid line at 814.

[0046] In an equation, $P(i)$ defines a mapping from N to M where N represents the set of numerical labels on the contacts from a previous frame, and M represents the labels of the components in a current frame. More specifically, $P(i)$ is a function of $i \in N$ (range $0 \dots N-1$) taking the value in M (range $0 \dots M-1$), such that $P(i) \neq$

$P(j)$ for $i \neq j$. Furthermore, P denotes the entire set of all possible $P(i)$, and the best $P(i)$ within P is determined so that the mapping defines an association of current components with previous touch positions that makes the most sense.

[0047] For any $P(i)$ in P , an array $D(i,P)$: $i \in N$ denotes the distances for each pair i in the mapping. More specifically, for each pair i in the mapping, the distance $D(i,P)$ is defined as the L^2 distance between the position of the component of the current frame and the predicted position of the component of the previous frame if the prediction exists, or the position of the component of the previous frame. A descending sort of array of $D(i,P)$ is then initiated and the result is denoted as Sorted $D(k,P)$, where $0 \leq k < N$ and:

$$\text{Sorted}D(0,P) \geq \text{Sorted}D(1,P) \dots \geq \text{Sorted}D(N-1,P)$$

[0048] The best P can be obtained when solving the following minimization problem: $\text{Best}P = \text{argmin}_{P \in P} (ED(P) + \lambda_0 * EC(P))$; where:

$$ED(P) \stackrel{\text{def}}{=} \sum_{k=0}^{N-1} (\text{Sorted}D(k,P) * \text{MAX}D^{N-k})$$

[0049] The value ED is the contribution from the maximum distance matching, and the value $\text{MAX}D$ is the maximum distance on the display screen (typically the diagonal distance). The layered maximum distance matching in this example accounts for the degeneracy of the configuration once the components with larger distances have been matched.

[0050] FIG. 9 illustrates an example 900 of crossing trajectories that may indicate unwanted associations between components 902 in a current frame and components 904 from a previous frame. A min-max distance determination may incorrectly associate a current component to a previous component. A value $EC(P)$ is defined as being proportional to the number of crossings in trajectory for all pairs of labels in the mapping P , and at a sampling rate of approximately 100Hz, a crossing can be regarded as spurious mapping. In this example, the matching of component A to component C, and component B to component D is preferred over the crossing trajectories of components A to D and B to C. However, without $EC(P)$, a pure $ED(p)$ would likely incorrectly associate the crossing trajectories of components A to D and B to C, which is clearly undesired.

[0051] FIG. 10 illustrates an example 1000 of how to compute and account for crossing trajectories, such as shown in FIG. 9. A computation of EC considers that

for the pair of trajectories $\overline{x_0x_1}$ and $\overline{x_2x_3}$, the condition where they cross each other is when the following inequalities hold true simultaneously:

$$0 < (b_1 * c_0 - b_0 * c_1) / det < 1$$

$$0 < (a_0 * c_1 - a_1 * c_0) / det < 1$$

$$(det = b_1 * a_0 - b_0 * a_1) ; \text{ where}$$

$$\mathbf{a} = (a_0, a_1) = \mathbf{x}_1 - \mathbf{x}_0$$

$$\mathbf{b} = (b_0, b_1) = \mathbf{x}_2 - \mathbf{x}_3$$

$$\mathbf{c} = (c_0, c_1) = \mathbf{x}_2 - \mathbf{x}_0$$

[0052] FIG. 11 illustrates example method(s) 1100 of prediction-based touch contact tracking. 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.

[0053] The aliased touch positions of two contacts shown in the example 600 (FIG. 6) of a touch merger can be resolved utilizing a probability distribution to partition the one larger component 610 so that each contact (e.g., the contact 602 and the contact 608) can be associated to a separate portion of the component. In embodiments, the contact tracking service 142 can resolve the first component to a first contact (i.e., centroid) of a frame (N), and resolve the second component, distinct from the first component, to a second contact of the frame (N). When the first and second contacts merge into the larger component in a subsequent frame (N+1), the larger merged component is resolved to a first contact and a second contact. Any number of merged contacts may be resolved.

[0054] At block 1102, tracking methods are utilized to resolve a first touch position to a first centroid(N) for a frame (N). At block 1104, a second touch position, distinct from the first touch position, is resolved to a second centroid(N) for the frame(N). The tracking continues to a subsequent frame (N+1) if the first touch position and second touch position remain distinct. However, if the first touch position and the second touch position merge into a larger component in the subsequent frame (N+1), as determined at block 1106, the merged touch positions are resolved to a first centroid (N+1) and a second centroid (N+1) at block 1108.

[0055] At block 1108, a method can be initiated to identify a continuous component in frame (N+1) having a combined statistic approximately equal to a sum of an individual statistic of the first contact and an individual statistic of the second contact from frame (N). A non-limiting example of such a statistic is the area of the

overall component. For example, if the area of a merged component in frame (N+1) is approximately equal to the combined area of the separate contacts in frame (N), the merged component may be verified. Alternatively or in addition, a continuous contact may be verified if the covariance of the merged component is substantially equal to the combined covariance of the individual contacts in a previous frame. The method 1100 may also be initiated for a merge-on-landing when two contacts are aliased to a single, larger component, and block 1104 is skipped when the second contact is not identified as a separate contact, but rather is aliased in the larger component to the first contact.

[0056] FIG. 12 illustrates example method(s) 1200 of prediction-based touch contact tracking, such as for a single contact of a gesture input. 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.

[0057] At block 1202, touch input sensor data is recognized as a series of components of a contact on a touch-screen display. For example, the touch input module 114 (FIG. 1) at the computing device 102 recognizes the touch input sensor data 116 as the series of components of a contact (*e.g.*, either of the two contacts 122, 124) of the gesture input 112 on the touch-screen display 110, such as on the example device 104.

[0058] At block 1204, a first component of the contact is identified. At block 1206, a second component is determined that correlates to the contact. At block 1208, the first component and at least the second component are associated to represent a tracking of the contact. For example, the contact tracking service 142 (FIG. 1) at the computing device 102 identifies a first component 126 of the contact 122 from a first frame, predicts a position for a subsequent component of the contact initiate a second frame and determines a second component 128 that correlates to the contact from the second frame, and associates the first component and at least the second component to represent a tracking of the contact if a nearest-neighbor criterion is passed. If the nearest-neighbor criterion is not passed, then the subsequent component from the second frame is initiated as a touch position of a new contact, and the previous contact is terminated. In an embodiment, the contact tracking service 142 determines that the second component correlates to the contact based on a

min-max distance determination between the first and second components of the contact.

[0059] At block 1210, a component position of the contact is predicted from the tracking of the contact. For example, the contact tracking service 142 predicts a component position of the contact from the tracking of the contact (*e.g.*, based on previous components of the contact that represent the contact tracking). At block 1212, the predicted component position is correlated to a subsequent component of the contact. At block 1214, the predicted component position is validated as correlating to the subsequent component of the contact based on a nearest-neighbor criteria. For example, the contact tracking service 142 validates the predicted component position as correlating to the subsequent component of the contact based on a nearest-neighbor contact mapping that evaluates distance from the predicted component position to one or more additional components of the contact. At block 1216, a same identifier is then assigned to all of the components that are associated with the contact. For example, the contact tracking service 142 also assigns the same identifier to the subsequent components that represent the tracking of the contact.

[0060] FIG. 13 illustrates example method(s) 1300 of prediction-based touch contact tracking, such as for multiple contacts of a gesture input. 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.

[0061] At block 1302, touch input sensor data is recognized as a series of components of a gesture input that includes at least two contacts on a touch-screen display. For example, the touch input module 114 (FIG. 1) at the computing device 102 recognizes the touch input sensor data 116 as a series of components of the gesture input 112 that includes the two contacts 122, 124 on the touch-screen display 110, such as on the example device 104. In this example, a user initiates a two-finger gesture input that includes the first contact 122 and includes the second contact 124.

[0062] At block 1304, a first component is determined as correlating to a second component of a first contact of the gesture input. At block 1306, the first component is mapped to the second component to represent a first contact tracking. At block 1308, a first identifier is assigned to both of the first and second components

of the first contact. For example, the contact tracking service 142 (FIG. 1) at the computing device 102 determines that the first component 126 correlates to the second component 128 of the first contact 122 of the gesture input 112, maps the first component to the second component to represent a tracking of the first contact, and assigns an identifier to both of the first and second components of the first contact.

[0063] At block 1310, a third component is determined as correlating to a fourth component of a second contact of the gesture input. At block 1312, the third component is mapped to the fourth component to represent a second contact tracking. At block 1314, a second identifier is assigned to both of the third and fourth components of the second contact. For example, the contact tracking service 142 (FIG. 1) at the computing device 102 predicts that the component 132 correlates to the component 134 of the second contact 124 of the gesture input 112, maps the components to represent a tracking of the second contact, and assigns a second, different identifier to both of the components of the second contact.

[0064] At block 1316, the components of the contacts are validated as correlating correctly based on a nearest-neighbor criteria that evaluates distance from each of the components to one or more additional components of the contacts. For example, the contact tracking service 142 validates that the components of the contacts are correlated correctly based on the nearest-neighbor contact mapping that evaluates distance from each of the components to one or more additional components of the contacts.

[0065] FIG. 14 illustrates various components of an example device 1400 that can be implemented as any of the devices, or services implemented by devices, described with reference to the previous FIGs. 1-13. 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.

[0066] The device 1400 includes communication devices 1402 that enable wired and/or wireless communication of device data 1404, 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 1406 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.

[0067] The device 1400 also includes communication interfaces 1408, 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.

[0068] The device 1400 includes one or more processors 1410 (*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 1412. In embodiments, the device 1400 can also include a touch input module 1414 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.

[0069] The device 1400 also includes one or more memory devices 1416 (*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.

[0070] 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.

[0071] 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.

[0072] A memory device 1416 provides data storage mechanisms to store the device data 1404, other types of information and/or data, and various device applications 1418. For example, an operating system 1420 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 1418 include a gesture recognition application 1422 and a contact tracking service 1424 that implement embodiments of angular contact geometry as described herein.

[0073] The device 1400 also includes an audio and/or video processing system 1426 that generates audio data for an audio system 1428 and/or generates display data for a display system 1430. 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.

[0074] Although embodiments of prediction-based touch contact tracking 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 prediction-based touch contact tracking.

CLAIMS

1. A method, comprising:
recognizing touch input sensor data as a series of components of a contact on a touch-screen display;
identifying a first component of the contact;
determining that a second component correlates to the contact; and
associating the first component and at least the second component to represent a tracking of the contact.
2. A method as recited in claim 1, further comprising:
assigning a same identifier to the first component and to the second component of the contact;
determining that subsequent components correlate to the contact; and
assigning the same identifier to the subsequent components that combine with the first and second components to represent the tracking of the contact.
3. A method as recited in claim 1, further comprising:
predicting a component position of the contact from the tracking of the contact; and
correlating the predicted component position to a subsequent component of the contact.
4. A method as recited in claim 3, wherein the predicted component position is predicted based on at least the previous first and second components that represent the tracking of the contact.
5. A method as recited in claim 3, further comprising validating that the predicted component position correlates to the subsequent component of the contact based on a nearest-neighbor criteria that evaluates distance from the predicted component position to one or more additional components of the contact.
6. A method as recited in claim 1, wherein said determining that the second component correlates to the contact is based on a min-max distance determination between the first and second components of the contact.
7. A method as recited in claim 1, further comprising resolving multiple contacts that alias to a single unassociated component.

- 8.** A computing device, comprising:
- a touch-screen display;
 - a touch input module configured to recognize touch input sensor data as a series of components of a contact on the touch-screen display;
 - at least a memory and a processor to implement a contact tracking service configured to:
 - identify a first component of the contact;
 - determine that a second component correlates to the contact; and
 - associate the first component and at least the second component to represent a tracking of the contact.
- 9.** A computing device as recited in claim 8, wherein the contact tracking service is further configured to:
- predict a component position of the contact from the tracking of the contact;
- and
- correlate the predicted component position to a subsequent component of the contact.
- 10.** A computing device as recited in claim 9, wherein the contact tracking service is further configured to validate that the predicted component position correlates to the subsequent component of the contact based on a nearest-neighbor criteria that evaluates distance from the predicted component position to one or more additional components of the contact.

1/9

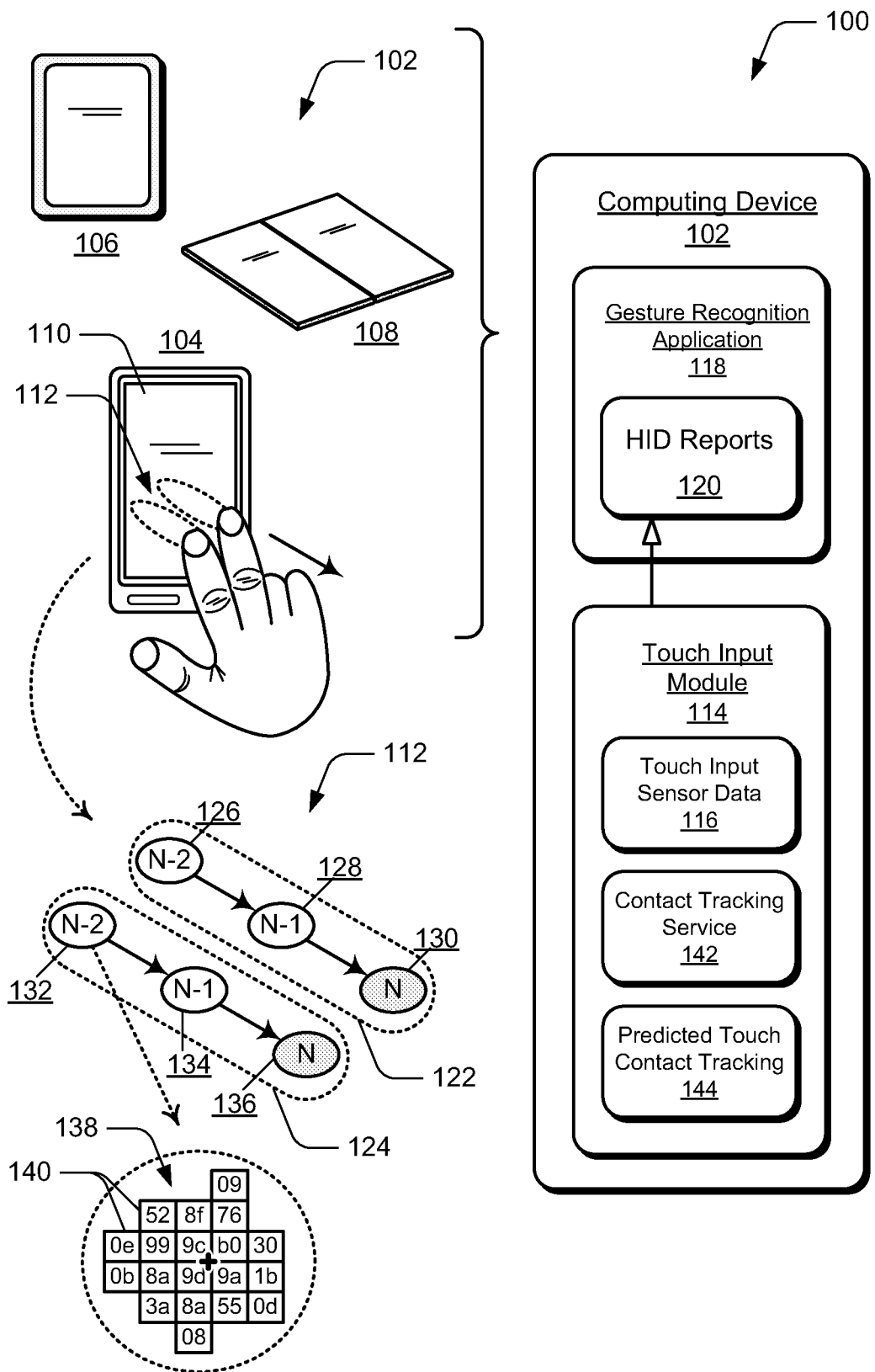
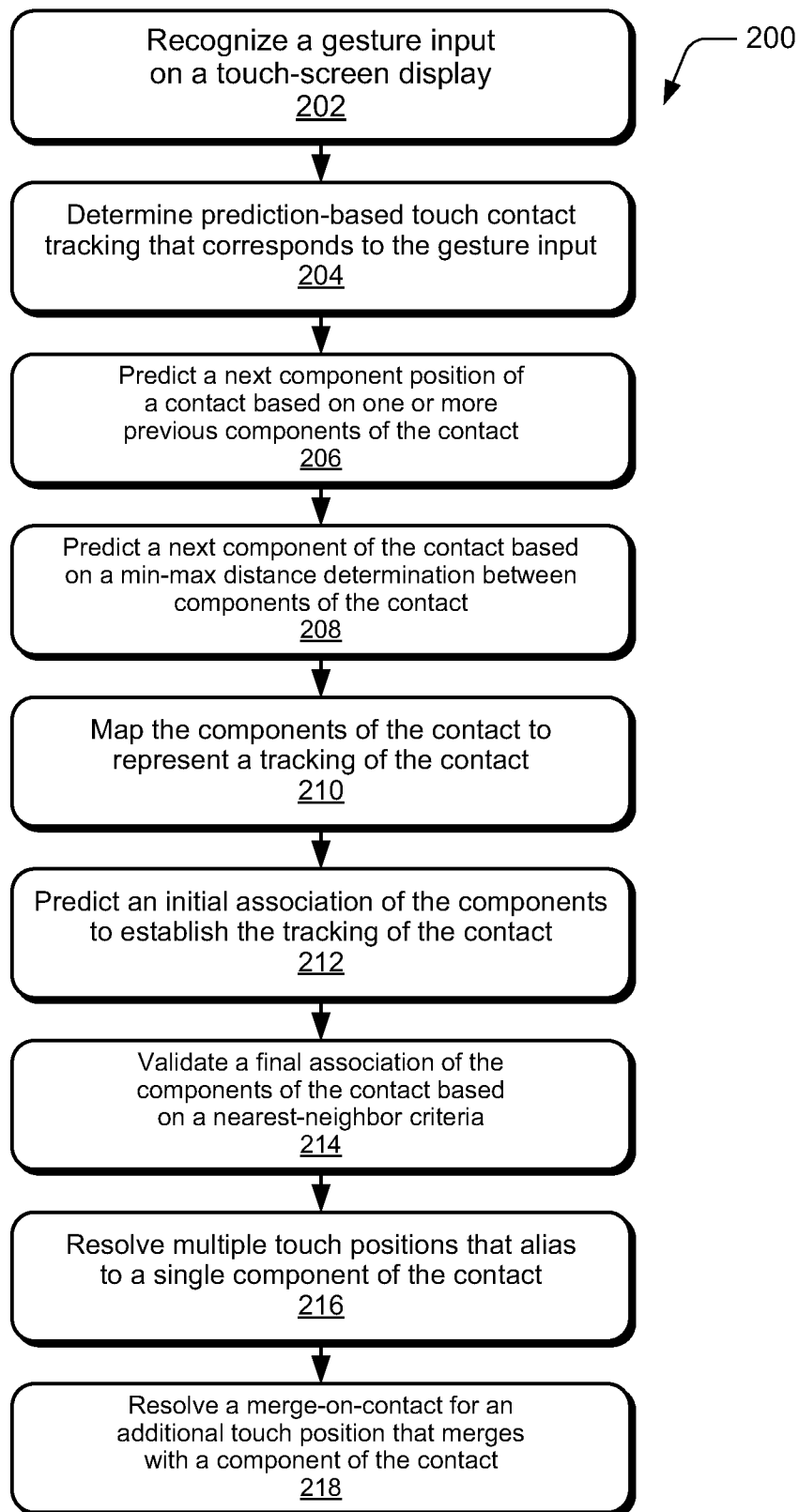
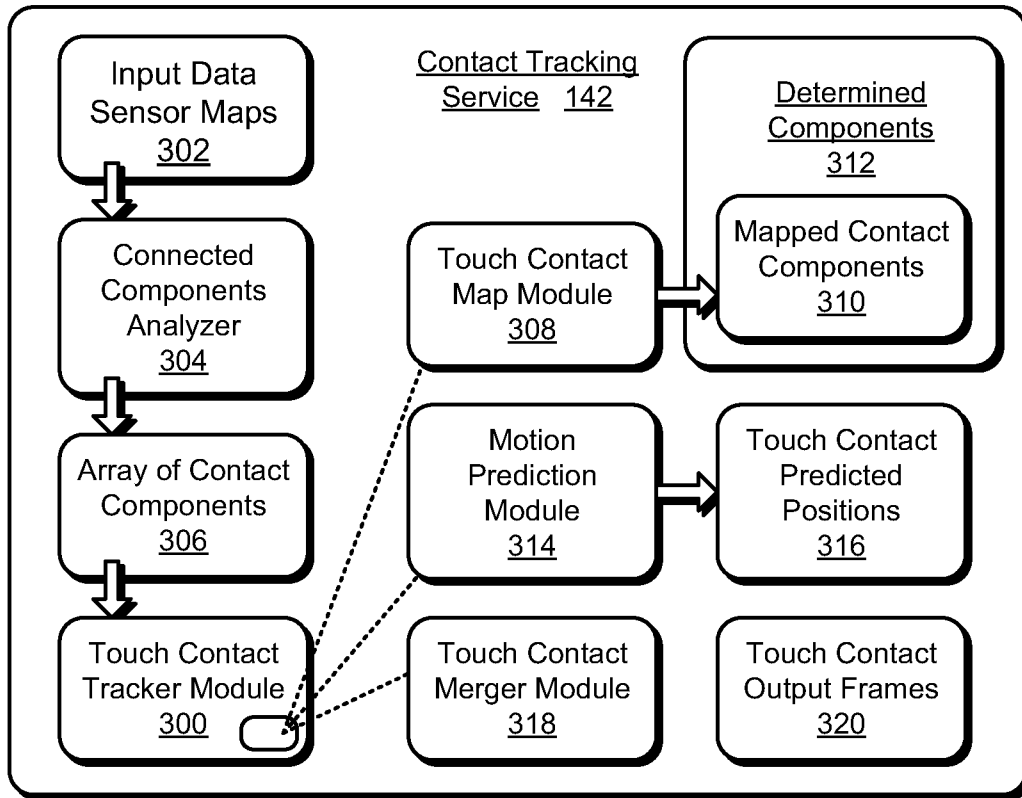
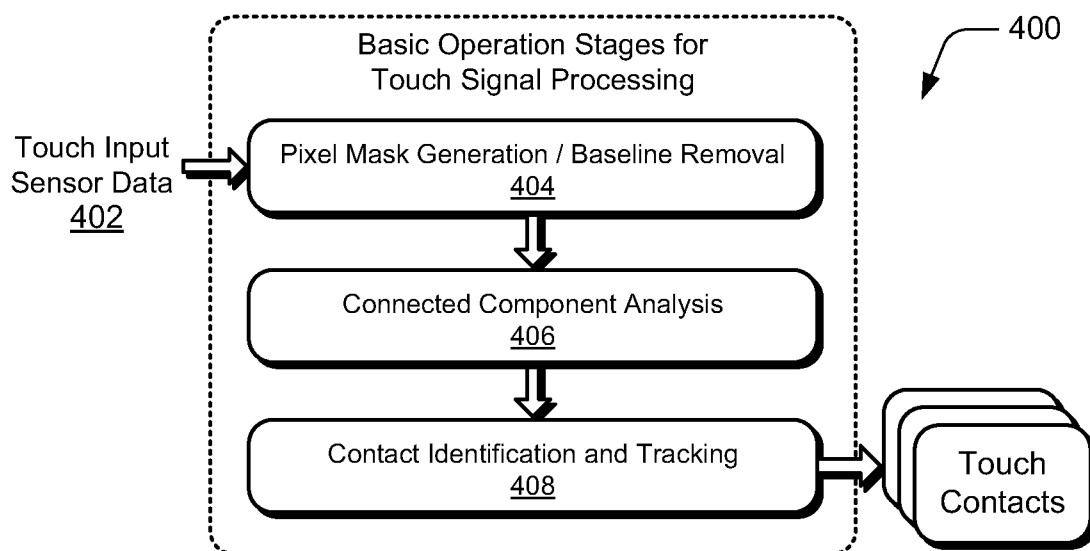


Fig. 1

2/9

**Fig. 2**

3/9

**Fig. 3****Fig. 4**

4/9

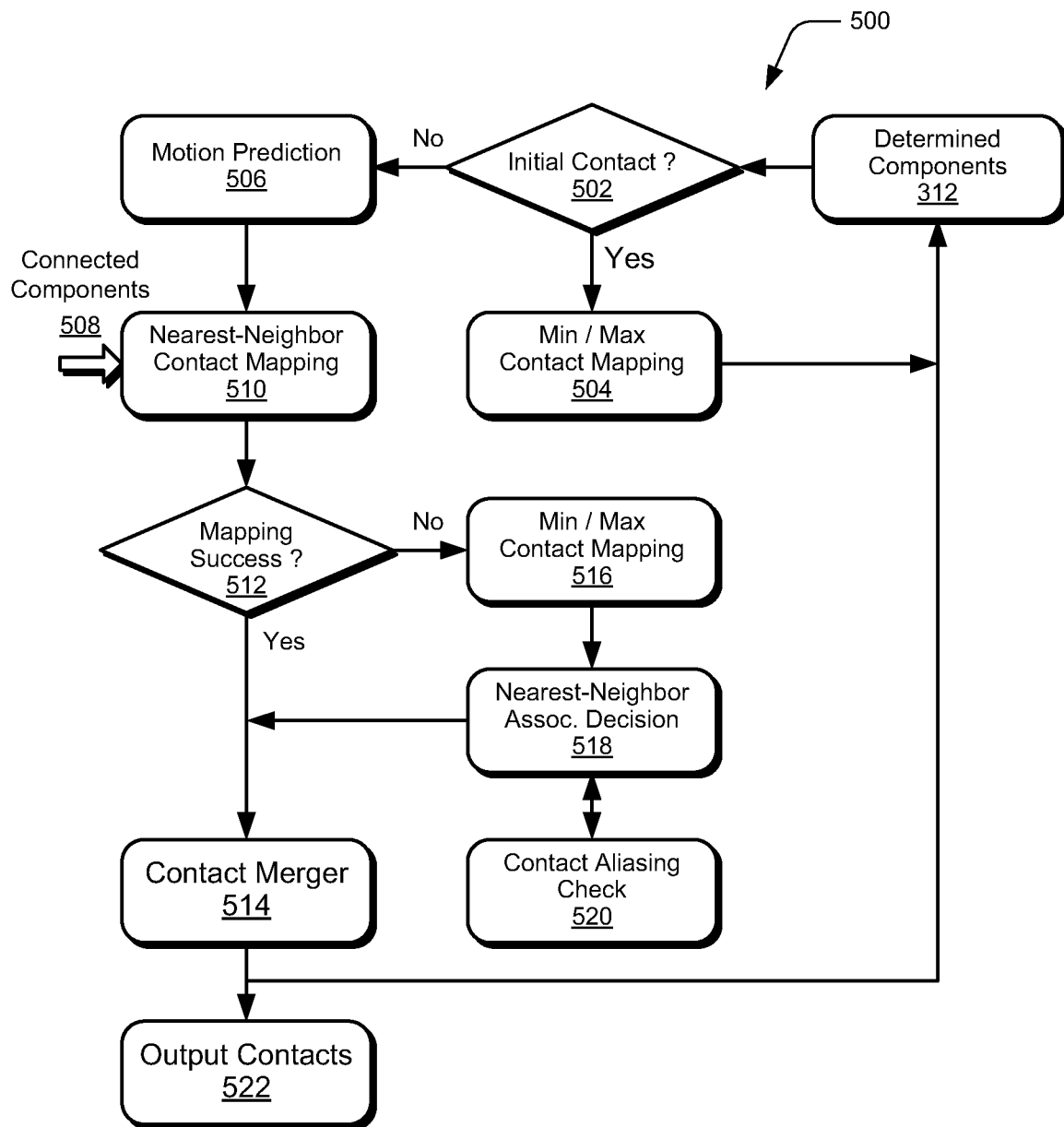


Fig. 5

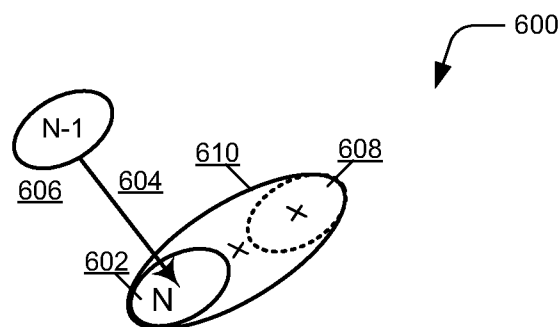
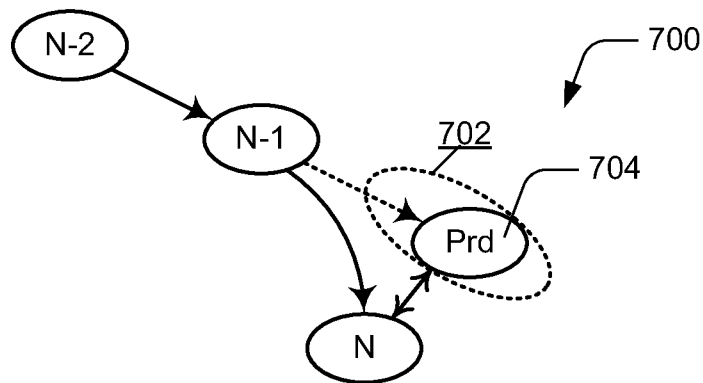
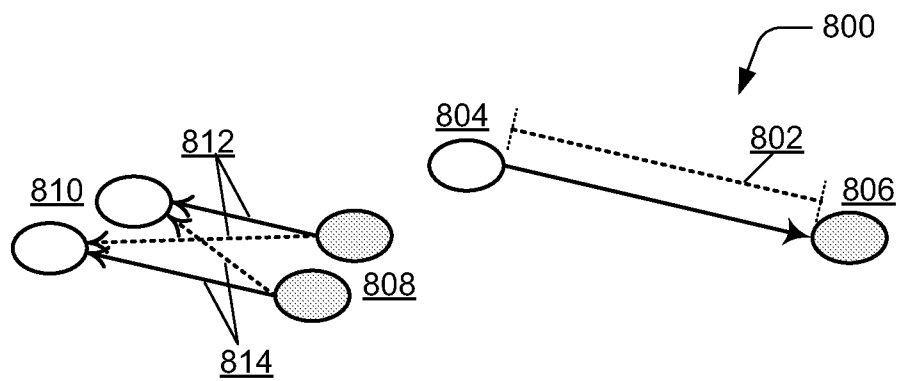
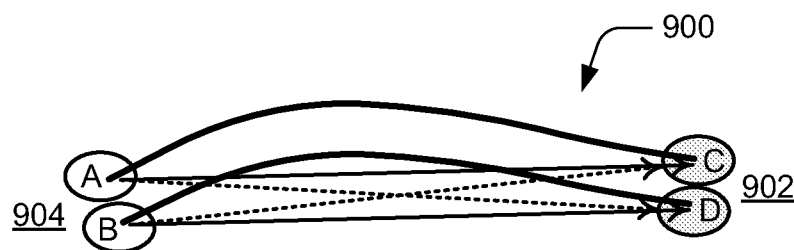
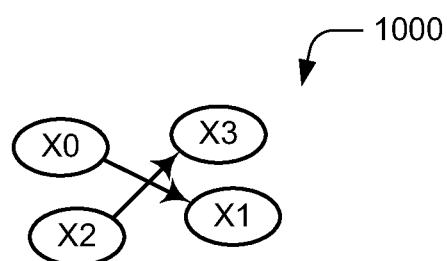
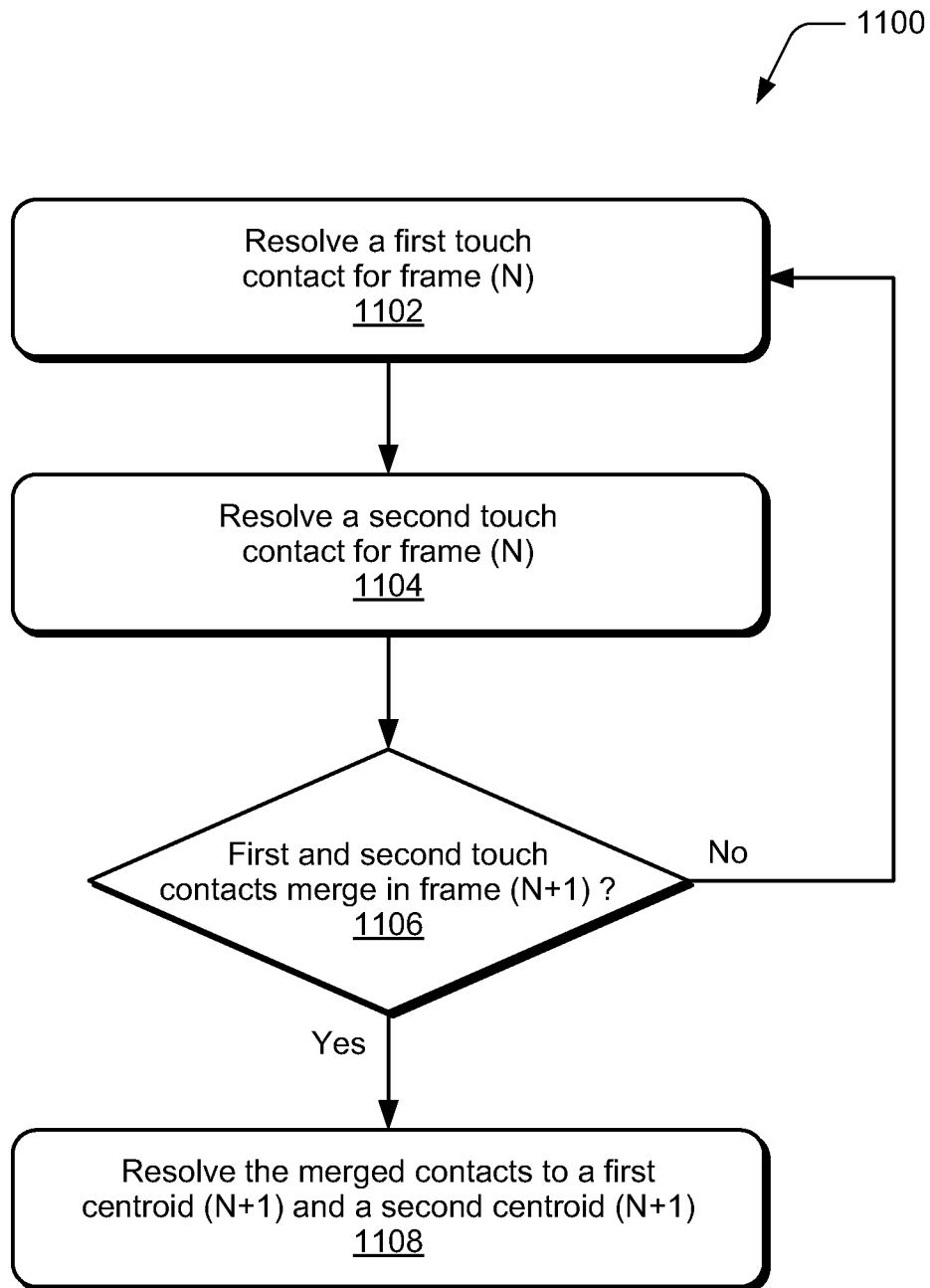


Fig. 6

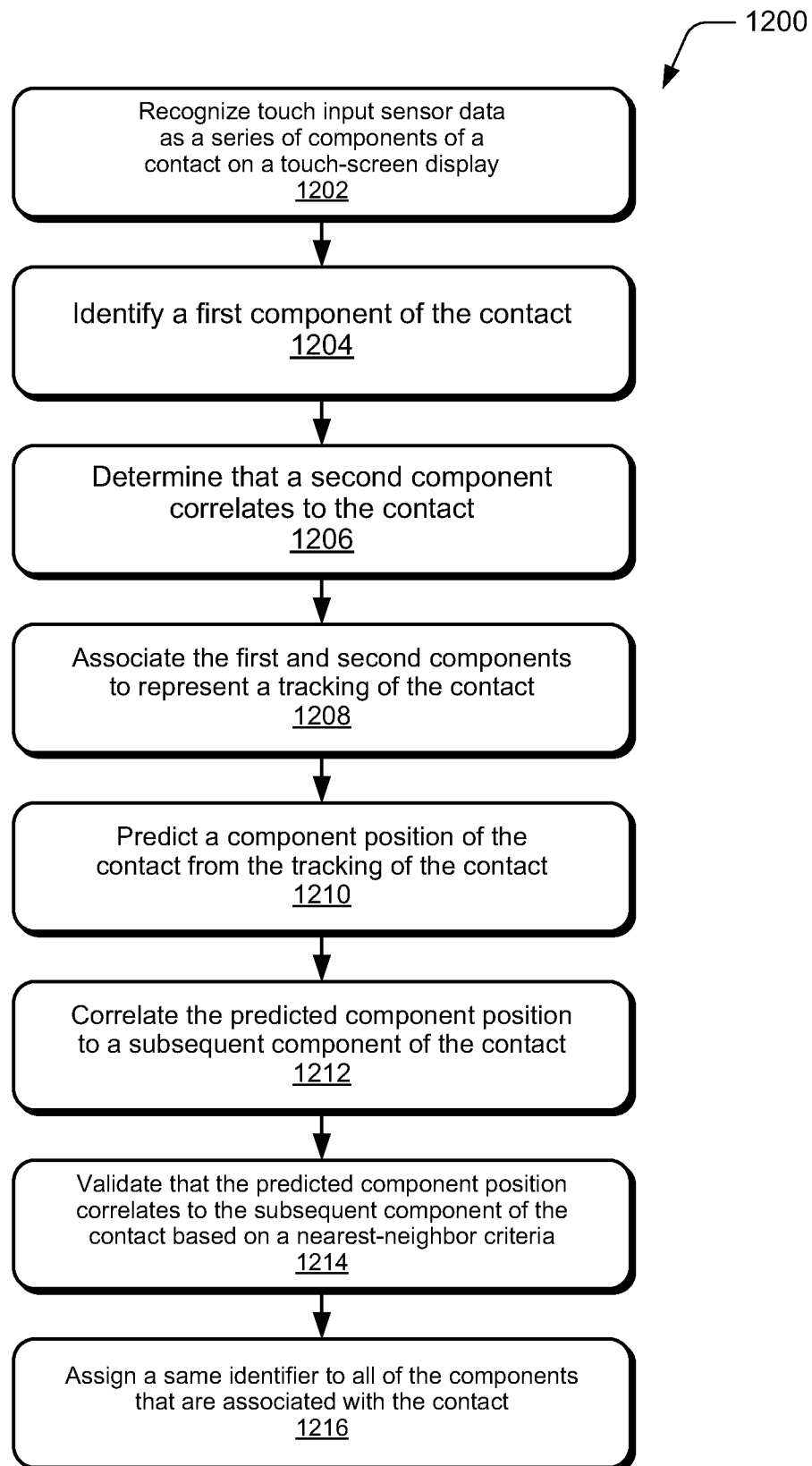
5/9

**Fig. 7****Fig. 8****Fig. 9****Fig. 10**

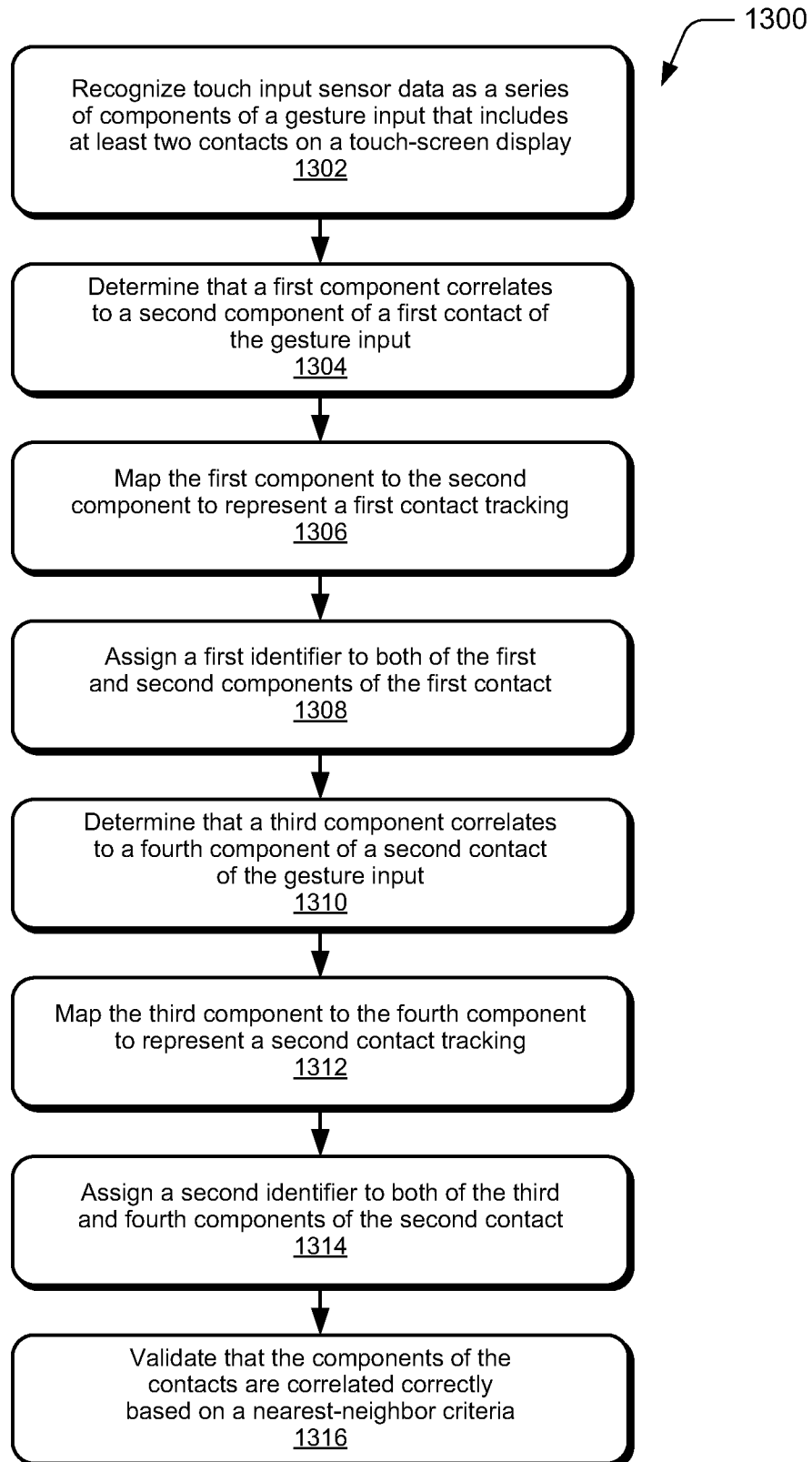
6/9

**Fig. 11**

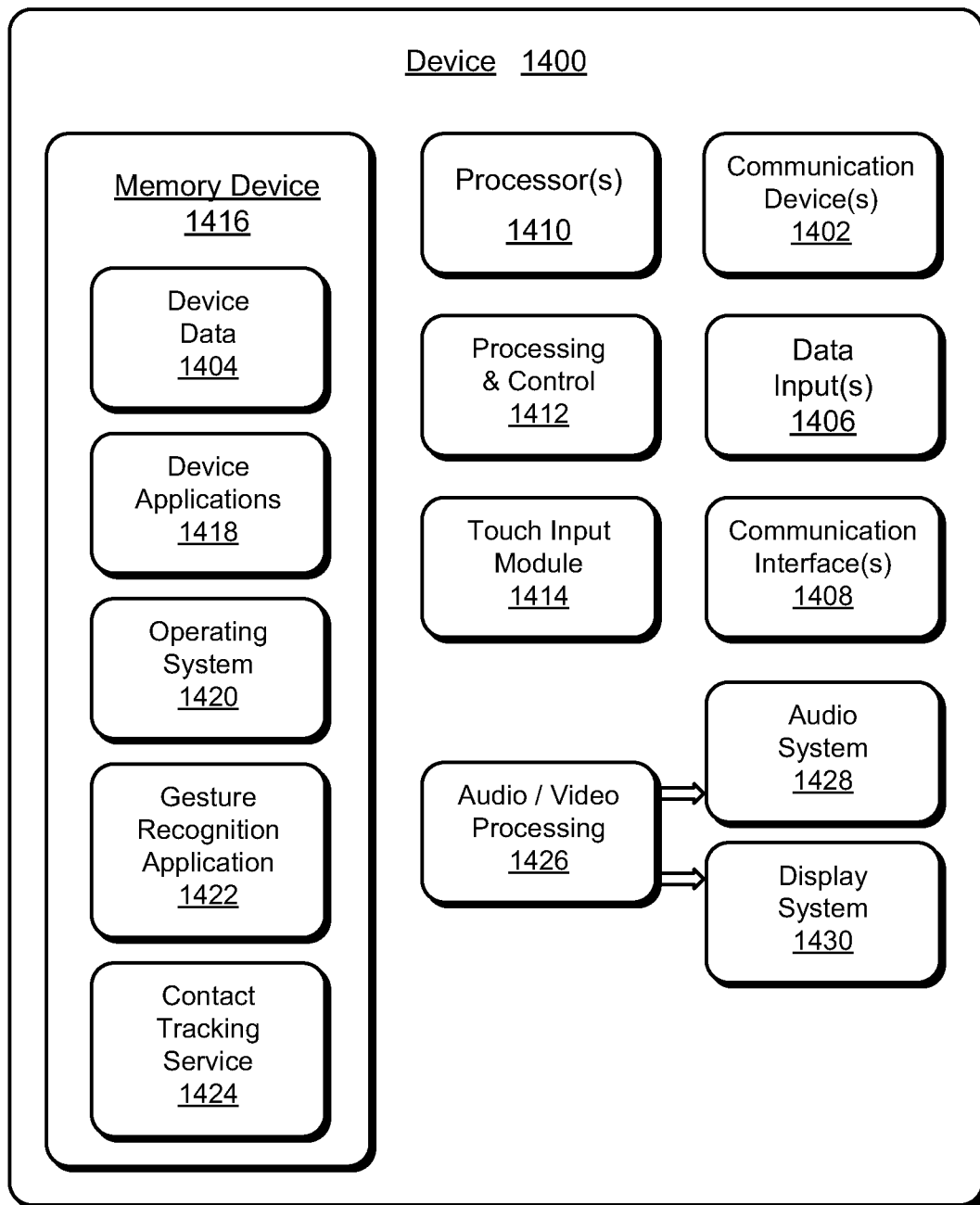
7/9

**Fig. 12**

8/9

**Fig. 13**

9/9

**Fig. 14**