

currently touching the surface, as happens when a finger touches down much earlier than the other fingers yet remains touching as they simultaneously touch down. An ordered list of finger release times is similarly maintained and searched separately. Alternative embodiments may require that a finger still be touching the surface to be included in the synchronized press subset.

[0253] Decision diamond 602 checks whether a synchronization marker is pending from a previous image scan cycle. If not, decision diamond 604 checks whether the search 603 found a newly synchronized press subset in the current proximity image. If so, step 606 sets the temporal synchronization marker to the oldest press within the new synchronized subset. Additional finger presses may be added to the subset during future scan cycles without affecting the value of this temporal synchronization marker. If there is currently no finger press synchronization, decision diamond 605 determines whether three or more fingers have just been released simultaneously. Simultaneous release of three or more fingers should not occur while typing with a set of fingers but does occur when lifting fingers off the surface from rest. Therefore simultaneous release of three or more fingers reliably indicates that the released fingers are not intended as keypresses and should be deleted from the keypress queue 605, regardless of whether these same fingers touched down synchronously. Release synchronization of two fingers is not by itself a reliable indicator of typing intent and has no effect on the keypress queue. The keypress queue is described later with FIGS. 42-43B.

[0254] Once a press synchronization marker for the hand is pending, further processing checks the number of finger presses which are synchronized and waits for release of the synchronized fingers. If decision diamond 608 finds three or more fingers in the synchronized press subset the user cannot possibly be typing with these fingers. Therefore step 612 immediately deletes the three or more synchronized presses from the keypress queue. This way they cannot cause key symbol transmission to the host, and transmission of key symbols from subsequent asynchronous presses is not blocked waiting for the synchronized fingers to be released.

[0255] However, when the synchronization only involves two finger presses 608, it is difficult to know whether the user intended to tap a finger pair chord or intended to type two adjacent keys and accidentally let the key presses occur simultaneously. Since such accidental simultaneous presses are usually followed by asynchronous releases of the two fingers, but finger pair chords are usually released synchronously, the decision whether the presses are asynchronous key taps or chord taps must be delayed until finger release can be checked for synchronization. In the meantime, step 610 places a hold on the keypress queue to prevent transmission of key symbols from the possible finger chord or any subsequent finger presses. To prevent long backups in key transmission, decision diamond 614 will eventually release the queue hold by having step 615 delete the synchronized presses from the keypress queue if both fingers remain touching a long time. Though this aborts the hypothesis that the presses were intended as key taps, the presses are also less likely to be key taps if the fingers are not lifted soon after touchdown.

[0256] If the synchronized fingers are not lifting, decision diamond 616 leaves the synchronization marker pending so synchronization checks can be continued with updated path parameters 600 after the next scan cycle. If the synchronized fingers are lifting, but decision diamond 618 finds with the

help of the synchronization release search 603 that they are doing so asynchronously 618, step 622 releases any holds on the keypress queue assuming any synchronized finger pair was intended to be two keypresses. Though the synchronized finger presses are not deleted from the keypress queue at this point, they may have already been deleted in step 612 if the pressed subset contained more than two. Also, step 624 clears the temporal synchronization marker, indicating that no further synchronization tests need be done for this subset.

[0257] Continuing to FIG. 39B, if the fingers synchronized during touchdown also lift simultaneously, step 618 removes them and any holds from the keypress queue in case they were a pair awaiting a positive release synchronization test. Further tests ensue to determine whether the synchronized fingers meet additional chord tap conditions. As with single finger taps, the synchronized fingers cannot be held on the surface more than about half a second if they are to qualify, as a chord tap. Decision diamond 626 tests this by thresholding the time between the release of the last remaining synchronized finger and the temporal press synchronization marker. A chord tap should also exhibit a limited amount of lateral finger motion, measured either as an average of peak finger speeds or distance traveled since touchdown in decision diamond 628. If the quick release and limited lateral motion conditions are not met, step 624 clears the synchronization marker with the conclusion that the synchronized fingers were either just resting fingers or part of a chord slide.

[0258] If the chord tap conditions are met, step 630 looks up, using the synchronized subset bitfield, any input events such as mouse clicks or keyboard commands assigned to the combination of fingers in the chord tap. Some chords such as those including all four fingertips may be reserved as resting chords 634, in which case decision diamond 632 will find they have no associated input events. If the chord does have tap input events, step 636 appends these to the main outgoing event queue of the host communication interface 20. Finally step 624 clears the synchronization marker in readiness for future finger synchronizations on the given hand.

[0259] As a further precaution against accidental generation of chord taps while typing, it is also useful for decision diamond 632 to ignore through step 634 the first chord tap which comes soon after a valid keypress without a chord slide in between. Usually after typing the user will need to reposition the mouse cursor before clicking, requiring an intervening chord slide. If the mouse cursor happens to already be in place after typing, the user may have to tap the finger chord a second time for the click to be sent, but this is less risky than having an accidental chord tap cause an unintended mouse button click in the middle of a typing session.

[0260] FIG. 40A shows the detailed steps of the chord motion recognizer module 18. The chord motion recognition process described below is repeated for each hand independently. Step 650 retrieves the parameters of the hand's identified paths 250 and the hand's extracted motion components from the motion extraction module 16. If a slide of a finger chord has not already started, decision diamond 652 orders slide initiation tests 654 and 656. To distinguish slides from glancing finger taps during typing, decision diamond 654 requires at least two fingers from a hand to be touching the surface for slide mode to start. There may be some exceptions to this rule, such as allowing a single finger to resume a previous slide within a second or so after the previous slide chord lifts off the surface.