[0270] In the preferred embodiment, the chord motion recognizers for each hand function independently and the input events for each chord can be configured independently. This allows the system to allocate tasks between hands in many different ways and to support a variety of bimanual manipulations. For example, mouse cursor motion can be allocated to the fingertip pair chord on both hands and mouse button drag to a triple fingertip chord on both hands. This way the mouse pointer can be moved and drug with either hand on either half of the surface. Primary mouse clicks would be generated by a tap of a fingertip pair on either half of the surface, and double-clicks could be ergonomically generated by a single tap of three fingertips on the surface. Window scrolling could be allocated to slides of four fingers on either hand.

[0271] Alternatively, mouse cursor manipulations could be allocated as discussed above to the right hand and right half of the surface, while corresponding text cursor manipulations are allocated to chords on the left hand. For instance, left fingertip pair movement would generate arrow key commands corresponding to the direction of motion, and three fingertips would generate shift arrow combinations for selection of text.

[0272] For host computer systems supporting manipulations in three or more degrees of freedom, a left hand chord could be selected to pan, zoom, and rotate the display background while a corresponding chord in the right hand could translate, resize and rotate a foreground object. These chords would not have to include the thumb since the thumb can touch down anytime after initiating chord motion without changing the selected chord. The user then need add the thumb to the surface when attempting rotation or scaling.

[0273] Finger chords which initially include the thumb can be reserved for one-shot command gestures, which only generate input events once for each slide of a chord rather than repeating transmission each time an additional unit of motion is detected. For example, the common editing commands cut, copy and paste can be intuitively allocated to a pinch hand scaling, chord tap, and anti-pinch hand scaling of the thumb and an opposing fingertip.

[0274] FIG. **41** shows the steps within the key layout definition and morphing process, which is part of the typing recognition module **12**. Step **700** retrieves at system startup a key layout which has been pre-specified by the user or manufacturer. The key layout consists of a set of key region data structures. Each region has associated with it the symbol or commands which should be sent to the host computer when the region is pressed and coordinates representing the location of the center of the region on the surface. In the preferred embodiment, arrangement of those key regions containing alphanumeric and punctuation symbols roughly corresponds to either the QWERTY or the Dvorak key layouts common on mechanical keyboards.

[0275] In some embodiments of the multi-touch surface apparatus it is advantageous to be able to snap or morph the key layout to the resting positions of the hands. This is especially helpful for multi-touch surfaces which are several times larger than the standard keyboard or key layout, such as one covering an entire desk. Fixing the key layout in one small fixed area of such a surface would be inconvenient and discourage use of the whole available surface area. To provide feedback to the user about changes in the position of the key layout, the position of the key symbols in these embodiments of the multi-touch surface would not be printed permanently on the surface. Instead, the position of the key symbols would be reprogrammably displayed on the surface by light emitting polymers, liquid crystal, or other dynamic visual display means embedded in the multi-touch surface apparatus along with the proximity sensor arrays.

[0276] Given such an apparatus, step **702** retrieves the current paths from both hands and awaits what will be known as a layout homing gesture. If decision diamond **704** decides with the help of, a hand's synchronization detector that all five of the hand's fingers have just been placed on the surface synchronously, step **706** will attempt to snap the key layout to the hand such that the hand's home row keys lie under the synchronized fingertips, wherever the hand is on the surface. Step **706** retrieves the measured hand offsets from the hand position estimator and translates all key regions which are normally typed by the given hand in proportion to the measured hand offsets. Note the currently measured rather than filtered estimates of offsets can be used because when all five fingers are down there is no danger of finger misidentification corrupting the measured offsets. This procedure assumes that the untranslated locations of the home row keys are the same as the default finger locations for the hand.

[0277] Decision diamond **708** checks whether the fingers appear to be in a neutral, partially closed posture, rather closed than outstretched or pinched together. If the posture is close to neutral, step **710** may further offset the keys normally typed by each finger, which for the most part are the keys in the same column of the finger by the measured finger offsets. Temporal filtering of these finger offsets over several layout homing gestures will tend to scale the spacing between columns of keys to the user's hand size. Spacing between rows is scaled down in proportion to the scaling between columns.

[0278] With the key layout for the hand's keys morphed to fit the size and current position of the resting hand, step **712** updates the displayed position of the symbols on the surface, so that the user will see that the key layout has snapped to the position of his hand. From this stage the user can begin to type and the typing recognizer **718** will use the morphed key region locations to decide what key regions are being pressed. The layout will remain morphed this way until either the user performs another homing gesture to move it somewhere else on the surface, or until the user takes both hands off the surface for a while. Decision diamond **714** will eventually time out so that step **716** can reset the layout to its default position in readiness for another user or usage session.

[0279] For smaller multi-touch surfaces in which the key layout is permanently printed on the surface, it is advantageous to give the user tactile feedback about the positions of key regions. However, any tactile indicators placed on the surface must be carefully designed so as not to impede smooth sliding across the surface. For example, shallow depressions made in the surface near the center of each key mimicking the shallow depressions common on mechanical keyboard keycaps would cause a vibratory washboard effect as the hand slides across the surface. To minimize such washboard effects, in the preferred embodiment the multi-touch surface provides for the fingertips of each hand a single, continuous depression running from the default index fingertip location to the default pinky fingertip location. This corresponds on the QWERTY key layout to shallow, slightly arched channels along home row from the "J" key to the ";" key for the right hand, and from the "A" key to the "F" key for the left hand. Similarly, the thumbs can each be provided