any left hand finger would freeze all the key positions in the left-hand zone but will not freeze the right hand zone keys. This embodiment will allow the operator to type any intended key easily by lifting the hands entirely or partially, or sliding. Although, there are recommended keys for certain fingers, one can type 'C' with the left index finger. However, this may be difficult depending on the initial distance between the middle finger and the index finger of the left hand before the freeze occurred.

[0102] The freeze will timeout in a designated period if no finger presents, and no interaction occurs. The timeout period may vary and/or be designated by the user. When both hands are no longer on the sensing surface (**1**), the soft keyboard disappears.

[0103] The operator can perform the virtual touch-typing mode with one hand (four fingers present or in the process of typing) and perform active space with another hand (browsing letter with one or two fingers), as shown in **FIGS. 18A** to **18**C.

[0104] Every time the operator rests the four fingers on one hand back to or near to all the reference keys positions where they were last frozen, all key positions (keys mapping) of that hand-zone will be recalibrated. In fact, according to one exemplary embodiment, recalibration may occur every time the operator places his/her fingers back to the reference positions in order to ensure a smooth typing experience.

[0105] The soft keyboard (**35**) may be displayed merely as a reminder of the position of each key. The soft keyboard (**35**) does not intend to show the actual size or distance between the keys, although according to one exemplary embodiment, the soft keyboard (**35**) can be set to do so. For a skilled touch-type operator, the soft keyboard (**35**) can be set to display in a very small size or set to be removed after the feedback has indicated which reference keys the user's fingers are on.

[0106] Returning now to **FIGS. 7C and 7D**, these FIGS. illustrate an exemplary method for the logical sequences that occur during a virtual touch-typing interaction method. As shown in **FIG. 7C**, the active space interaction mode illustrated in **FIGS. 7A and 7B** are performed (a-h). Once performed, the computing device determines if four fingers are detected in the resting regions (aa). If not, the active space process illustrated in steps (i-s) of **FIG. 7B** are performed. If, however, four fingers are detected in the resting regions, a reference key position for each reference finger is determined, the appropriate keys are highlighted on the soft keyboard, and the associated keys location is determined for each reference finger (bb). Once these key locations are determined, the computing device determines whether any keys are missing due to area constraints (cc). If any keys are missing, the active space process illustrated in **FIG. 7B** is performed. If, however, there are not any keys missing, key selections are detected (dd). If the selection of a key is detected, the input data is recorded. If, however, no key selection is detected, the computing device senses for the movement of fingers (ff), the movement of fingers outside the resting region (gg), or the removal of fingers (hh) from the sensing surface as described above. If the computing device senses the removal of a finger (hh), all the key positions are frozen (ii) and the computing device determines if a key selection has been made (jj). If so, the data input is recorded, interactive feedback is performed, and any clocks are deactivated (kk). If however, the computing device does not determine that a key selection has been made, the computing device then determines if the fingers have been moved, lifted off of the sensing surface, and then touched down on a different location (ll). If such a selection is sensed by the computing device, the newly selected keys are determined, the appropriate keys on the soft keyboard are highlighted, and any active clocks are deactivated as the computing device returns again to block (jj). If, however, the fingers have not been moved, lifted off of the sensing surface, and then touched down on a different location (ll), the computing device checks for a first (nn) or second (pp) clock time out, which if detected will restart the present method (oo, qq). If, however, neither clock time out is detected, the computer checks to see if all four fingers are present in the resting regions and if a first clock is dormant (rr). If so, the first clock is activated (ss) and the present method begins again at block (jj) (uu). If, however, block (rr) is negative, the computing device then determines if all four fingers are missing and a second clock is dormant (tt). If so, the method returns to block (jj). If not, the four fingers are checked for their last reference key position (ww). If they are there, the process begins again by deactivating the clocks and returning to block (bb). The method illustrated above and in **FIGS. 7C and 7D** are merely exemplary embodiments of the present system and method and in no way limit the present system and method to the embodiments described.

[0107] Moreover, according to one exemplary embodiment, a password typing mode may be presented. According to this exemplary embodiment, a number of visual feedbacks (e.g. inputting highlight) may be omitted when typing a password. The computer will recommend typing in the touch-type mode since browsing letters with the active space mode may reveal the password to an onlooker (e.g. when the display is large).

[0108] Moreover, the present virtual touch-type and active space modes are well suited for use on a handheld PC, since its small size will not allow touch-typing with the normal mechanical keyboard. Additionally, the software hosting the present system and method will dynamically adjust positions of the keys according to the current operator's finger position and hand-size. According to this exemplary embodiment, the software can learn to adapt to all kinds of hands during word processing, this is contrary to other existing systems where the operator is forced to adapt to the system.

[0109] The present system and method also allows an operator to focus only on the display screen while interacting with a computing device. Consequently, those who do not know how to touch-type can type faster since they no longer need to search for keys on the keyboard, and eventually will learn to touch-type easily. Those who are touch-typists can also type more pleasantly since the software can be customized for their unique desires.

[0110] The present user interface models, active space methods, and virtual touch-typing methods may also be applied to simulate various kinds of traditional switch panels. For example, numeric keypads, calculator panels, control panels in the car, remote controller panels, and some musical instrument panels such as piano keyboards. More-