

[0014] FIG. 5 illustrates an example layout of virtual keys on a touch screen where each of at least one of the virtual keys has a set of greater than one key location corresponding to that virtual key.

[0015] FIG. 6 is a flowchart illustrating a portion of the FIG. 2 flowchart for a virtual key such as is shown in FIG. 5, having more than one key location corresponding to it.

[0016] FIG. 7 illustrates an example layout of key locations corresponding to virtual keys on a touch screen based on an initial “touch” by a user of fingers simultaneously in a “home key” position on the touch screen.

[0017] FIG. 8 illustrates an example arc-shaped keyboard, particularly suited for activation of the virtual keys using a thumb.

[0018] FIG. 9 illustrates an example of virtual GUI items on a touch screen display where, in this example, the virtual GUI items are not virtual keys of a keyboard.

DETAILED DESCRIPTION

[0019] We now describe a method of operating a touch screen to activate one of a plurality of virtual keys. The touch screen is an input device of a computing system which, for example, operates under program control (where the program is stored as instructions in a memory, for example) based at least in part on user input. The computing system may be, for example, a personal computer, a computing system embedded within an electronic device (such as, for example, a cellular telephone, media player or other consumer electronic device.) In operation, a user interacts with the touch screen and, in response, the touch screen (and or hardware/software associated with the touch screen) provides location data pertaining to touch input on the touch screen.

[0020] The touch screen provides a graphical user interface that includes GUI items that can be activated by the user as input to the program operating on the computer system. As can be seen from FIG. 1-1A, 1-1B and 1-1C, the touch area of a user’s finger, to activate a GUI item on a touch screen, typically does not match a visual target associated with that GUI item. FIG. 1-1A shows a finger 12 approaching the touch-sensitive layer 14 of the touch screen. FIG. 1-1A also shows a visual target 10 displayed on the screen 16 of the touch screen. FIG. 1-1B shows the finger 12 touching the touch sensitive layer 14. It can be seen from FIG. 1-1B that the finger 12 contacts the touch sensitive layer 14 at a touch area 18. The mid-point 20 of the touch area 18 does not match the visual target 10 displayed on the screen 16 of the touch screen. FIG. 1-1C shows a plan view of the finger 12, visual target 10 and mid-point 20 of the touch area 18. FIG. 9 illustrates, in plan view, a situation where a finger 902 is being used to potentially activate two somewhat small GUI items, a keep GUI item 904 and a delete GUI item 906.

[0021] It is desirable to process touches on a touch screen in a way that does not necessarily depend on a match between the visual target 10 and a touch area of a touch to activate a GUI to which the visual target 10 corresponds. It is desirable to process touches on a touch screen in a way that “makes sense,” which may include considering factors beyond (or instead of) a correspondence of the visual target and the touch area of a touch to activate the GUI to which the visual target corresponds.

[0022] In one example, the GUI items are virtual keys of a virtual keyboard. The GUI items are not limited to being virtual keys of a virtual keyboard, however. FIG. 1 illustrates a portion of a touch screen, where the gridlines 104 are displayed to the user, visually defining “keys” that are present on a conventional qwerty keyboard. The shapes and locations of the visually-defined keys need not be operationally correlated to the shapes of the virtual keys. This will be seen later, particularly with reference to FIG. 7. The visually-defined keys are labeled in FIG. 1 as 102_q, 102_w, 102_e, etc., to indicate the letter to which each key corresponds on a conventional qwerty keyboard.

[0023] Each of the plurality of virtual keys has a set of at least one key location corresponding to it. The dots 106_q, 106_w, 106_e, etc indicate key locations. In the FIG. 1 example, each set of key locations (corresponding to a separate virtual key) is a set of one key location. Later, sets of key locations are illustrated that include more than one key location.

[0024] In FIG. 1, a touch location is denoted by the “X”108. In accordance with a broad aspect, a parameter is determined for each virtual key that relates the touch location to the set of at least one key location corresponding to that virtual key. The determined parameters are processed to determine one of the virtual keys, and a signal is generated indicating activation of the determined one of the virtual keys.

[0025] We now reference FIG. 2, which is a flowchart illustrating a particular example of a virtual key activation method. For simplicity of illustration, we initially describe a situation like in FIG. 1 where each set of key locations is a set of one key location. At step 202, the touch location is determined, based on location data pertaining to touch input on the touch screen. The location data may be provided, for example, by low level program code being executed by a processor associated with the touch screen. Referring to the FIG. 1 example, the touch location is denoted by the “X”108.

[0026] At step 204, for each virtual key, a distance is determined between the touch location “X”108 and the key location 106 (generically) corresponding to that virtual key. Step 206 is a step included in some examples, and step 206 is discussed later. At step 208, it is determined to which of the virtual keys the shortest distance determined in step 204 corresponds. At step 210, a signal is generated to indicate activation of the virtual key determined in step 208.

[0027] Having described a particular example of a virtual key activation method, we now refer to the illustration in FIG. 3. The hexagonal shapes 302_q, 302_w, 302_e, etc. illustrate an example of the shapes of the virtual keys, superimposed on the visually-defined keys 102. The example touch location “X”108 shown in FIG. 1 is also shown in FIG. 3. Typically, the hexagonal (or other) shapes of the virtual keys would not be displayed to the user. Generically, the particular configuration of each virtual key 302 is based on the number and configuration of the key locations in the set of key locations corresponding to the virtual key. Other example configurations will be described later.

[0028] FIG. 3-1 illustrates an example in which, in contrast to FIG. 3, there is not such a regular correspondence between the visual targets and the key locations.