

that apparatus whether embodied by a hardwired circuit or embodied by a processor running software that can perform those processes.

[0027] The operation of the processor 38 in this exemplary embodiment is described in more detail with reference to FIG. 4, which is a flow chart for this aspect of the invention. On receiving signals  $S_x$ ,  $S_y$  (input data) in step S100, the processor 38 first determines in step S102 if they correspond to a position in the virtual keyboard 20. If they do not, then the process proceeds to step S104, which decides if the touch corresponded to a position in the list display area 26. If they do correspond to a position in the virtual keyboard 20 the processor 38 decides or determines in step S106 appropriate candidate keys for what the user intended. This determination is based on calculations of the distances from the selected position 52 to the representative positions 50*t*, 50*y*, 50*g*, 50*h* of the adjacent keys 22. Initially at least, as is shown in FIG. 3, the representative position 50 of a key 22 is at the centre of that key, but that may be modified as is discussed later (see Step S116).

[0028] The processor does not work out the distance from the selected position to the representative position for every possible key. It ignores those that are more than a predetermined distance away, which in this embodiment is the distance equal to the distance between the centres of two adjacent keys in the same row (e.g. from the centre of the “t” key to the centre of the “y” key). This leads to the selection of the letter “t”, “y”, “g” and “h” keys as candidates.

[0029] Another possibility is for the predetermined distance to be based on the distance between two adjacent keys in different rows (e.g. from the centre of the “y” key to the centre of the “g” key or from the centre of the “y” key to the centre of the “h” key). Many other possibilities exist. The distance that is used depends upon the sensitivity that the designer (or user) desires.

[0030] An alternative approach to selecting the candidate keys for the key that is pressed is to select the key in which the selected position falls, to work out the two closest sides of that key to the selected position and then to include those other keys that are in contact with any part of those two sides. Alternatively again, each key 22 can be divided into quarters and the candidates are chosen as the key in which the selected position falls and those keys adjacent to the key quarter in which the selected position falls. In these cases, the selected position 52 in FIG. 3 would only lead to the letter “y”, “g” and “h” keys as candidates.

[0031] In step S108 the most likely symbol of the candidate symbols is displayed in the relevant position in the message line 24. The most likely symbol is deemed to be the symbol from the key 22 in whose display area the selected position falls. Thus with the example shown in FIG. 3, the letter “h” would be displayed in the message line 24.

[0032] Alternatively, the processor would display the symbol from the key 22 whose representative position is closest to the selected position 52, in the current position in the message line 24. In the example shown in FIG. 3, although the selected position 52 is in the display area 54*h* of the “h” key, it is closer to the representative position 50*y* of the “y” key than to the representative position 50*h* of the “h” key. Thus the letter “y” would be displayed, and not the letter “h” in the message line 24.

[0033] In step S110 the processor decides upon a list of candidates, either as alternatives to the symbol displayed in step S108 or as a complete word to replace the current string in message line 24. The sub-steps for this process are described later with reference to FIG. 5.

[0034] The following step S112 displays the list generated in step S110 in list display area 26. The process next passes through a decision step S114, where it decides if the preceding input has confirmed any keys, for example if an input symbol has been followed by a space, which has been followed by some other input, which means that the user intended the space and therefore intended what preceded the space. If confirmation has occurred, the next step is S116, where the representative positions of the keys representing the confirmed inputs, may be recalibrated. The process then reverts to step S100, as it also does if the answer to the question of step S114 is negative. Step S100 awaits a new user input. Typically this would be by way of a selection from an item in the displayed list, in which case the selected letter or word would appear in the message line 24, or this may be by way of a new input via the virtual keyboard, in which case the previously assumed symbol put in the message line 24 in step S108 remains there and the above process repeats itself. Alternatively, the user may be selecting some other instruction.

[0035] If step S104 determines that the current selected position 52 is within the list display area 26, the processor enters that selected word or symbol in the message line in step S118. The process then goes straight to step S116 for re-calibration of key representative positions. If step S104 determines that the current selected position 52 is not within the list display area 26, the next step is step S120, in which whatever other processing is necessary is carried out. Step S122 then determines if the process is to leave the virtual keyboard. If it is not leaving the virtual keyboard, the process reverts to step S114 to check if any symbol has been confirmed.

[0036] FIG. 5 shows the sub-steps for step S110 for generating a list. Firstly in step S202, the processor decides if any of the current candidate symbols is a letter. If at least one of them is a letter, then in step S204 the processor decides if the current input is not the first symbol in the current symbol string, i.e. whether it is the second or a later one. If it is not the first symbol in the string, then in step S206 the processor decides if the preceding symbols in the string are all letters. If they all are, then in step S208, the processor decides if any of the current candidate symbols could, if placed in the current letter string, lead to a word in the dictionary database in the memory 40.

[0037] If the answer to the decision in any of steps S202 to S208 is “No”, then the process proceeds to step S210, where a symbol list is generated just containing the symbols for the remaining candidate keys not displayed in the message line by step S108. These other symbols are placed in the list in the order of proximity of the selected position 52 to the representative positions for their corresponding selected candidate keys 22. Thus with the example shown in FIG. 3, when the letter “h” is displayed in the message line 24, the list would contain the letters “y”, “g” and “t”, in that order.

[0038] If the answer to the decision in every one of steps S202 to S208 is “Yes”, then the process proceeds to step