S210, where a set of words is generated using the dictionary database. The set contains the current letter string in the message line with each candidate symbol at the end of it (except for the combination that is already displayed in step S108) and every possible word allowed by the insertion of each candidate symbol in the current letter string. In step S212 a weighting process is used to give scores to each possible member of the set. These scores are compared with each other in step S214 and a list of scoring members is generated in score order in step S216. In one embodiment, the list of scoring members will be a list of six alphanumeric characters that is typically the top six scoring members. However, the number in this list can vary and usually depends on the display area and font size.

[0039] In more detail, the weighting process in step S212, mentioned above, awards a score $W_{final}$ to each member of the set according to the following formula:

$$W_{final}=a*W_{freq}+b*W_{distance} \quad (1)$$

[0040] where $W_{freq}$ is a score awarded to a word based upon the likelihood of that word or combination, which is usually attendant on its frequency of use, and $W_{distance}$ is a score which is the inverse of the distance from the selected position 52 to the representative position for the key that would be required for that word or combination to be the correct one. In formula (1), "a" and "b" are precept constants which are set to give a good balance between selection based on word frequency and selection based on the distance of the selected position to the representative position of a key.

[0041] In variant embodiments, there can be a learning programme to vary these constants "a" and "b" so that the more accurate the user's selection history tends to be, the higher the value "b" becomes relative to the value "a" and the greater the weighting given to the distance score over the likelihood score.

[0042] Every word in the dictionary database is given a likelihood score, $W_{freq}$ on a scale of 1-10, which is also maintained in the memory 40. The dictionary database may not necessarily include every word in a particular language and size of the dictionary database depends the memory space allocated by the memory 40. The most frequently used words such as "the" have a score of 10 , whilst less frequently used words like "theomachy" have a score of 1, with most words in between. For the purposes of formula (1), combinations that do not appear in the dictionary database are treated as having a likelihood score, $W_{freq}$ of 0.

[0043] The word scores are preset in the factory but are automatically modified through use, so that words used more frequently by the user get a higher $W_{freq}$ score and words used less frequently get a lower $W_{freq}$ score. New words can also be added through a learning process. The predictive word input technology can usefully automatically track the frequency of word use. For instance: if a non-dictionary word is selected even once, it is added to the dictionary and every five times a word is used, it gains a higher score. In this example, there may be no more than a predetermined number of words with any one $W_{freq}$ score; when one word moves up or down a score, taking the number of words with that score over the maximum, the least frequently used word from that score moves down. Individual user's habits can also be learned. Thus, if more than one user uses any one device, then the different users can be identified and their habits learned separately.

[0044] In further variants, the predictive word input technology can also take advantage of grammar checking technology as an extra factor in deciding scores.

[0045] Normally the dictionary only contains words containing letters. However, alternative embodiments provide a dictionary database with symbol strings containing symbols other than letters, and/or the ability to learn such strings (for instance telephone numbers). In such embodiments, various steps, such as steps S202 and S206 are adjusted to allow through non-letter symbols.

[0046] Step S116, mentioned above, relates to re-calibration of representative positions of the keys. This aspect is based on the fact that people tend not to be random in where they touch a screen to select a particular key. They tend to hold the device in a similar position throughout each use and from one use to another, with the same parallax error in each case. Thus they are likely to touch the screen at roughly the same position, each time when they want a particular key, even though that position may not be directly above the desired key.

[0047] As is mentioned above, initially the representative position of a key is at its centre. Whilst that is where it starts, it is not fixed there and can be re-calibrated based on use. More particularly, the system learns from the confirmation of previous key selections and moves the representative position of each key towards where the user tends to touch the screen when selecting that key. Thus, during symbol and word selection, the X and Y offset from the key centre, for each key that is input, is collected and, once a candidate word is selected or a symbol confirmed (e.g. by way of a return or space input), those offsets are used to calculate new positions for the respective representative positions or their respective keys to recalibrate the touch panel.

[0048] For each input symbol, there is an X offset (Xoff-cent) between the selected position 52 and the centre of the symbol key and a Y offset (Yoff-cent) between the selected position 52 and the centre of the symbol key. During the re-calibration process in step S116, those offsets are used to calculate a new representative position for the respective key. This is calculated based on an average.

[0049] More particularly, the new representative positions for each key, Xnew and Ynew, in terms of the X and Y offset from the centre of each key are determined by the following formulae:

$$X\text{new}=(X\text{off-cent}+\Sigma X\text{off-cent-old})/n \quad (2)$$

$$Y\text{new}=(Y\text{off-cent}+\Sigma Y\text{off-cent-old})/n \quad (3)$$

[0050] where "ΣXoff-cent-old" is the sum of all previous "Xoff-cent" used in recalculating the representative position for this key, "ΣYoff-cent-old" is the sum of all previous "Yoff-cent" used in recalculating the representative position for this key, and "n" is the number of times the representative position for this key has been recalculated, including the current time.

[0051] So that initial inputs do not skew the results, "ΣXoff-cent-old" and "Yoff-cent-old" are originally set at "0" and "n" is precept to a large figure such as 100. This therefore gives weight given to the existing representative position.

[0052] This calculation means that the original setting will always be a factor in Xnew and Ynew. This can avoided, for