When the keyboard is displayed on a small touch screen, keystrokes that are off-center from a character are detected. Software compares the possible text strings of probable sequences of two or three typed characters against known combinations, such as a history of previously typed text or a lexicon of text strings rated for their frequency within a context. When the character generated by the system is not the character intended by the user, the user must correct the character before going on to select the a following character because the generated character is used to determine probabilities for the following keystroke.

[0010] Recently, various input devices have been introduced that provide new opportunities for user interaction with computers, PDAs, video games, cell phones, and the like.

[0011] For example the laser-projection keyboard offered by companies such as Virtual Keyboard (see http://www.vkb.co.il/) and Canesta (see http://www.canesta.com/) is a projection keyboard that is capable of being fully integrated into smart phones, cell phones. PDAs, or other mobile or wireless devices. The laser-projection keyboard uses a tiny laser pattern projector to project the image of a full-sized keyboard onto a convenient flat surface, such as a tabletop or the side of a briefcase, between the device and the user. The user can then type on this image and the associated electronic perception technology instantly resolves the user's finger movements into ordinary serial keystroke data that are easily used by the wireless or mobile device.

[0012] Also known are muscle-sensing keyboards, such as the Senseboard® virtual keyboard (see, for example, http://www.senseboard.com/), which typically consist of a pair of hand modules with a pad that is placed in the palm of the user's hand. A muscle-sensing keyboard enables a user to type without the physical limitations of a standard keyboard. This type of virtual keyboard typically uses sensor technology and artificial intelligence, such as pattern recognition, to recognize the characters that a user is typing. The keyboard detects the movements of the fingers and relates them to how a touch typist would use, for example, a standard QWERTY keyboard. The information thus generated is then transferred to, for example: a mobile device, such as a personal digital assistant (PDA) or a smart phone using, for example, a cable or a Bluetooth wireless connection.

[0013] Yet another virtual keyboard is the fabric keyboard (see, for example, http://www.electrotextiles.com/). Such keyboards provide three axes (X, Y and Z) of detection within a textile fabric structure approximately 1 mm thick. The technology is a combination of a fabric sensor and electronic and software systems. The resulting fabric interface delivers data according to the requirements of the application to which it is put. The three modes of sensor operation include position sensing (X-Y positioning), pressure measurement (Z sensing), and switch arrays. Thus, a keyboard can be constructed that detects the position of a point of pressure, such as a finger press, using the interface's X-Y positioning capabilities. The system works even if the fabric is folded, draped, or stretched. A single fabric switch can be used to provide switch matrix functionality. Interpreting software is used to identify the location of switch areas in any configuration, for example to implement keyboard functionality.

[0014] Unfortunately, a major obstacle in integrating such virtual keyboards into various data receptive devices is fact that it is very difficult to type accurately when there are no physical keys on which to touch-type. In this regard, the user must rely entirely on hand-eye coordination while typing. Yet most touch typists are taught to type without looking at the keys, relying on tactile feedback instead of such hand-eye coordination. In such virtual keyboards there is literally no point of registration for the user's hands, and thus no tactile feedback to guide the user as he types.

[0015] For all of the preceding systems, the fundamental problem is that the specific activations that result from a users attempts to activate the keys of a keyboard do not always precisely conform to the intentions of the user. On a touch screen keyboard, the user's finger or stylus may hit the wrong character or hit between keys in a boundary area not associated with a specific character. With a miniaturized mechanical keyboard, a given key press may activate the wrong key, or may activate two or more keys either simultaneously or with a roll-over motion that activates adjacent keys in a rapid sequence. And with a virtual keyboard, the lack of tactile feedback allows the user's fingers to drift away from the desired key registrations. Other examples include common keyboards operated by users with limited ranges of motion or motor control, where there is a limited ability to consistently strike any particular space or key; or where the limb, such as in the case of an amputee, or the gloved hand or finger, or the device used to make the entry, such as a stylus, is far larger than the targeted key or character space.

[0016] It would be advantageous to provide an enhanced text entry system that uses word-level disambiguation to correct inaccuracies in user keystroke entries automatically, especially with regard to virtual keyboards.

SUMMARY OF THE INVENTION

[0017] The invention provides an enhanced text entry system that uses word-level disambiguation to correct inaccuracies in user keystroke entries automatically, especially with regard to virtual keyboards.

[0018] Specifically, the invention provides a text entry system comprising:

[0019] a user input device comprising a virtual keyboard including an auto-correcting region comprising a plurality of the characters of an alphabet, wherein one or more of the plurality of characters corresponds to a location with known coordinates in the auto-correcting region, wherein a location associated with the user interaction is determined when a user interacts with the user input device within the auto-correcting region, and the determined interaction location is added to a current input sequence of interaction locations;

[0020] a memory containing a plurality of objects: wherein one or more objects comprise a string of one or a plurality of characters forming a word or a part of a word;

[0021] an output device; and

[0022] a processor coupled to the user input device, memory, and output device, said processor comprising:

[0023] a distance value calculation component which, for a determined interaction location in the input sequence of interactions, calculates a set of distance