extra corner area along the x-axis for the dominant-hand side of the square to account for users who hold their styluses at steep angles. A property of this iteration was that it did not negatively impact users who held their styli more vertically.

[0050] An alternative embodiment would use other shapes for the corner regions. One example is to use triangular corner regions rather than rectangles. **FIG. 6** shows the design using rectangles changing to triangles **601-602**, and providing extra corner area on the dominant-hand side for both right-handed **603-604** and left-handed **605-606** users. This design was shown to lessen accidental corner hits even more. Many other alternatives are possible. For example, inflation/deflation can be modified so that the inflated and deflated corners are both triangles.

[0051] Though Graffiti is popular, it also has problems. We took some lessons from studies of Graffiti in an effort to alleviate some of its problems, or at least to avoid reproducing them. Certain letters in Graffiti have specific problems. For example, many people handwrite an n beginning at the top-left of the letter and initially going down. In Graffiti, this almost always results in an h. The present invention supports an optional initial down stroke on letters that commonly have them: b, d, m, n, p, q, and r. Similarly, many people make a down stroke at the end of u, and in Graffiti this almost always produces an h or w. The present invention allows this down stroke on u. Graffiti also often produces a u when novices make a v but forget to add an unnatural serif on the right. The present invention avoids this u-v confusion, as every form is more than just subtly different from every other form.

[0052] Another problem for novices is confusion between the x and k in Graffiti, as these are mirror images of each other. The present invention removes this similarity by redesigning the k so that it starts at the top-left, where a handwritten k starts, not at the top-right, where a Graffiti k starts. The preferred embodiment does leave the top-right k as an alternate form for current Graffiti users.

[0053] As mentioned above, motor impaired users sometimes "bounce" inadvertently on the screen. One embodiment of the present invention has characters that are entered by taps in the corners. We removed all of these except period (.) to reduce the likelihood of entering an accidental mode or character by inadvertent "bounces" on the screen.

[0054] In the present invention, we differentiate position based on the known location of the square. Hence, we can tell i from l even though they are the same stroke because they are on different edges. This is a powerful concept, as it allows for input in a very small area. It also means we do not need separate regions of the screen devoted to letters, numbers, capitals (e.g., as Jot does), and so on.

[0055] The implementation of the preferred embodiment for the present invention enables fast character recognition. With reference to **FIG. 9**, the recognizer does nothing until it detects a pen-down event **1001**. If the down event is in an inflated corner **1003**, then it deflates the corners **1005**, and begins queuing up all the points over which the stylus moves until the stylus is lifted **1006-1007**. No recognition or filtering is done during the stylus movement to maximize the number of movement points queued. Once the stylus is lifted, the recognizer notes the first corner where the recognition started **1008**, and then loops through the point queue and hit-tests the points against the deflated corner regions, collecting the sequence of corners **1009-1012**. The result of this loop is a 32-bit integer value representing the sequence in which the corners were hit. This integer is assembled efficiently: when a new corner is hit, the existing integer sequence is bit-shifted to the left and the new corner is "appended" with bitwise-OR. This sequence is then sent to a lookup function **1013** that finds the character corresponding to the corner sequence, if any **1014**. If a result is a defined character, the character is output at **1015** and the process ends at **1016**.

[0056] Another embodiment eliminates the enqueuing of the points while the mouse is moving **1005-1008** by testing the points for being in corners **1003**, **1012** while the pen is moving, and then queuing only the corner ids. This results in simpler code, but code which takes more processor time for each input point.

[0057] This recognition algorithm is fast in linear time O(n), and it could be implemented on a weak processor with a poor digitizer sampling rate and a noisy digitization of stylus coordinates. Anecdotally, it was not possible for us to move the stylus faster than the present invention could recognize the stroke on a Palm Vx, which polls its screen for the pen every 20 ms.

[0058] Another aspect of the present invention is implemented not in software but in plastic in the preferred embodiment. The template **103** is important for the present invention to work well, and designing and fabricating this plastic piece involved just as iterative a process as developing the software. We have numerous prototypes. Some are small and sit on the PDA's screen. We found this to work fine for able-bodied people, but users with motor impairments sometimes put pressure with their fingers on the template, causing it to press against the screen and confuse the digitizer. We designed another model to avoid putting pressure on the screen **102**. It sits on top of the Palm chassis and therefore cannot touch the screen (as shown in **FIG. 1**).

[0059] As an alternative embodiment, **FIG. 7** shows that the square hole **703** might be on the faceplate of a wrist watch **701** which has a touch-sensitive screen. Then the user could use a stylus **702** to enter text on the watch using the character set of the present invention. Another alternative embodiment is shown in **FIG. 8**, where a slightly larger touch sensitive surface **801** is used, and the pointing is performed with a finger **802** on the surface **803** instead of a stylus. The finger can feel the edges and corners. The touch sensitive surface might be mounted on the front or back of a mobile device, or somewhere in an automobile to enable text entry while driving. In all cases, the same alphabet is used (for example, as shown in **FIG. 2**).

[0060] The present invention's recognition technology and methodology for generating character recognition data can be implemented on other types of electronic devices, be they handheld or wearable, including those without the luxury of a fully-digitized touch screen. All that is required are four corner sensors and either one other sensor or a timer. These sensors could be crude: they do not have to determine coordinates, only whether the stylus is in contact with them or not. Thus, the invention is a reliable character recognizer without being a pattern-matcher that depends on the whole path of movement.

[0061] Another advantage of the present invention is that it can be implemented using a simple, low-cost joystick