a second and a second. If not, processing returns to **770** to await either finger liftoff or the hold setup condition to be met during future scans of the sensor array. When the hold setup condition is met, decision diamond **802** checks whether all other fingers on the hand of the given finger keypress lifted off the surface more than a half second ago. If they did, step **804** will initialize typematic for the given keypress element. The combination of decision diamonds **800** and **802** allow the user to have other fingers of the hand to be resting on the surface when a finger intended for typematic touches down. But typematic will not start unless the other fingers lift off the surface within half a second of the desired typematic finger's touchdown, and typematic will also not start until the typematic finger has a continued to touch the surface for at least half a second after the others lifted off the surface. If these stringent conditions are not met, the keypress element will not start typematic and will eventually be deleted through either tap timeout **782** when the finger lifts off or through tap timeout **796**) if another touches down after it.

Step **804** simply sets a flag which will indicate to decision diamond **798** during future scan cycles that typematic has already started for the element. Upon typematic initialization, step **810** sends out the key symbol for the first time to the host interface communication queue, along with any modifier symbols being held down by the opposite hand. Step **812** records the time the key symbol is sent for future reference by decision diamond **808**. Processing then returns to step **770** to await the next proximity image scan.

Until the finger lifts off or another taps asynchronously, processing will pass through decision diamond **798** to check whether the key symbol should be sent again. Step **806** computes the symbol repeat interval dynamically to be inversely proportional to finger proximity. Thus the key will repeat faster as the finger is pressed on the surface harder or a larger part of the fingertip touches the surface. This also reduces the chance that the user will cause more repeats than intended since as finger proximity begins to drop during liftoff the repeat interval becomes much longer. Decision diamond **808** checks whether the dynamic repeat interval since the last typematic symbol send has elapsed, and if necessary sends the symbol again in **810** and updates the typematic send time stamp **812**.

It is desirable to let the users rest the other fingers back onto the surface after typematic has initiated **804** and while typematic continues, but the user must do so without tapping. Decision diamond **805** causes typematic to be canceled and the typematic element deleted **778** if the user asynchronously taps another finger on the surface as if trying to hit another key. If this does not occur, decision diamond **182** will eventually cause deletion of the typematic element when its finger lifts off.

The typing recognition process described above thus allows the multi-touch surface to ergonomically emulate both the typing and hand resting capabilities of a standard mechanical keyboard. Crisp taps or impulsive presses on the surface generate key symbols as soon as the finger is released or decision diamond **792** verifies the impulse has peaked, ensuring prompt feedback to the user. Fingers intended to rest on the surface generate no keys as long as they are members of a synchronized finger press or release subset or are placed on the surface gently and remain there along with other fingers for a second or two. Once resting, fingers can be lifted and tapped or impulsively pressed on the surface to generate key symbols without having to lift other resting fingers. Typematic is initiated ether by impulsively pressing and maintaining distinguishable force on a key, or by holding a finger on a key while other fingers on the hand are lifted. Glancing motions of single fingers as they tap key regions are easily tolerated since most cursor manipulation must be initiated by synchronized slides of two or more fingers.

Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope and spirit of the invention being indicated by the following claims.

What is claimed is:

1. A method for mapping gestures performed on a multi-touch surface to simulate mouse manipulations, the method comprising:

detecting a proximity image of a plurality of fingertip contacts on the multi-touch surface;

determining one or more of a size, shape and orientation associated with each of the plurality of fingertip contacts based on the proximity image;

generating mouse pointer motion signals in response to translational slides of a first predetermined number of fingertip contacts;

generating a single mouse click signal in response to a synchronized tap of a second predetermined number of fingertip contacts; and

generating mouse drag signals in response to translational slides of a third predetermined number of fingertip contacts,

wherein the gestures simulating mouse manipulations are performed with a first hand and gestures corresponding to text cursor manipulations are performed with an opposite hand.

2. The method of claim **1** wherein the text cursor manipulations include generating text cursor motion signals in response to translational slides of a fourth predetermined number of fingertip contacts.

3. The method of claim **2** wherein the text cursor manipulations include generating text selection signals in response to translational slides of a fifth predetermined number of fingertip contacts.

4. The method of claim **1** wherein the text cursor manipulations include generating text selection signals in response to translational slides of a fourth predetermined number of fingertip contacts.

5. A method for mapping gestures performed on a multi-touch surface to simulate mouse manipulations, the method comprising:

detecting a proximity image of a plurality of fingertip contacts on the multi-touch surface;

determining one or more of a size, shape and orientation associated with each of the plurality of fingertip contacts based on the proximity image;

generating mouse pointer motion signals in response to translational slides of a first predetermined number of fingertip contacts;

generating a single mouse click signal in response to a synchronized tap of a second predetermined number of fingertip contacts;

generating mouse drag signals in response to translational slides of a third predetermined number of fingertip contacts; and

generating a double mouse click signal in response to a synchronized tap of a fourth predetermined number of fingertip contacts,

wherein the gestures simulating mouse manipulations are performed with a first hand and gestures corresponding to text cursor manipulations are performed with an opposite hand.