[0093] **FIG. 11** illustrates an example of a third "pop-up" general usage mode of the touch screen of the present invention. In the pop-up mode, the touch screen displays a special image much as in the auxiliary mode. The pop-up mode allows all the same display elements on the touch screen and all the same alternative interpretations of finger actions on the touch sensor as in the auxiliary mode. However, the pop-up image appears in response to a user command or other event in the host computer and is not associated with any particular software application on the main display.

[0094] In the example of **FIG. 11**, the pop-up image is an application launcher. When the application launcher is invoked, image **1100** replaces the previous image on the touch screen. Image **1100** includes various icons **1102** representing commonly used tools and software applications on the computer. The set of applications shown may be predetermined or may be chosen by the user. When the user taps the finger on one of icons **1102**, image **1100** disappears and is replaced by the original touch screen image, and the selected application software is launched. Typically, this application would be a conventional software application such as a word processor running on the main display of the computer, but some of icons **1102** may represent system commands (such as shutting down the computer), other tools (such as another pop-up application on the touch screen), or links to additional application launcher screens. The user can also tap on icon **1104** to exit the application launcher screen without invoking any application.

[0095] Pop-up screens such as the application launcher of **FIG. 11** may be invoked by any of various well-known means for invoking applications, such as a keyboard key, an icon like icon **420** or corner tap region like region **410** of **FIG. 4**, or the "Start" menu of Microsoft Windows®.

[0096] Pop-up screens may be implemented as regular applications as viewed by the operating system; in this case, the application would not create a visible window on the main display, but it would create a touch screen image using the same mechanisms that other applications would use to create an auxiliary touch screen image. In an alternate embodiment, pop-up screens like that of **FIG. 11** could be implemented specially within the touch screen driver software, or they could be implemented in the touch screen controller hardware such as controller **216** of **FIG. 2**.

[0097] **FIG. 12** illustrates a pop-up calculator application that operates entirely within the touch screen. Image **1200** includes the familiar numeric display **1202** and a matrix of buttons **1204** of a calculator. The user taps on the button icons to operate the calculator in the usual fashion. The user taps on button **1206** to close the calculator and restore the touch screen to its previous image. The calculator operates autonomously with respect to the applications visible on the main display of the computer. This autonomous behavior is particularly valuable when the calculator is being used in tandem with an application on the main display, such as a database application looking up numeric data. In the example of **FIG. 12**, buttons **1208** and **1210** are provided to allow numbers to be pasted back and forth between the calculator and the active application on the main display.

[0098] Computer keyboards traditionally include a numeric keypad, but portable computer keyboards rarely have room for a conventional keypad. Portable computer system designers are forced to adopt awkward solutions such as the "Fn" key. A pop-up numeric keypad screen very similar to the calculator of **FIG. 12** could serve the role of the numeric keypad in a portable computer. This keypad screen could be invoked by the "NumLock" key already provided on computer keyboards for activating the numeric keypad.

[0099] Many computer operating systems provide a magnification tool to assist the visually impaired. This tool typically creates a window on the main screen that displays a magnified copy of the display image surrounding the cursor. This magnifier window can obstruct useful information on the main display. According to the present invention, as illustrated in **FIG. 13A**, the magnifier can instead take the form of a pop-up image **1302** on touch screen **1300**, leaving the main display undisturbed. Unlike the examples of **FIGS. 11 and 12**, the magnifier pop-up would probably be left displayed much of the time that the computer is used. This pop-up application would therefore leave the touch sensor operating as a conventional pointing device; hence, corner tap regions **1304** and **1306** are retained. When the touch screen is in the activated state, the magnifier application can offer additional accessibility features on the touch screen. In the example of **FIG. 13B**, in the activated state, touch screen **1320** replaces image **1302** with an image of controls such as magnification level adjustment **1322**. Also, close box **1324** appears in the activated state to allow the user to turn off the magnification feature. In an alternate embodiment, in the activated state, the magnifier activates features to assist operation of small on-screen controls.

[0100] In an alternative magnification mode, the main display image is reduced and moved to the touch screen display, and then a magnified view of a portion of the image is shown on the main display. This has the advantage that the main display is larger and likely to have greater clarity and color depth than the touch screen, and will thus be a better detail viewing device for the visually impaired.

[0101] Debugging is a task that greatly benefits from a secondary display. Computer programmers today sometimes attach a second display monitor to their computers so that the program under debugging can operate undisturbed on the first display monitor. These second displays are costly and inconvenient, particularly on portable computers. As shown in **FIG. 13C**, a debugger could be implemented instead as a pop-up application on the touch screen of the present invention, providing the benefits of a secondary debugging display with no extra cost or bulk. In the example of **FIG. 13C**, image **1340** includes command buttons **1342** and source code display window **1344**.

[0102] Users of ideographic languages like Chinese and Japanese typically rely on input methods beyond the simple direct keystroke mapping used in Western languages. A variety of input methods are in use for ideographic languages, many of which require or benefit greatly from providing visual feedback to the user through a special window. This window can obstruct the application for which the input is intended. According to the present invention, the input method dialog can be implemented as a pop-up image on the touch screen. One popular input method is handwriting recognition, in which case the touch screen can also serve as the handwriting input device for added benefit.

[0103] **FIG. 14** illustrates an example Chinese handwriting entry system on touch screen **1400**. Handwriting entry