wheels, knobs, etc. Each UI element represents a component or control of the user interface. The application behind the UI element(s) has access to the parameter data calculated in block **108**. In one implementation, the application ranks the relevance of the touch data to the UI element corresponding there to. The ranking may be based on some predetermine criteria. The ranking may include producing a figure of merit, and whichever UI element has the highest figure of merit, giving it sole access to the group. There may even be some degree of hysteresis as well (once one of the UI elements claims control of that group, the group sticks with the UI element until another UI element has a much higher ranking). By way of example, the ranking may include determining proximity of the centroid (or features) to the GUI object associated with the UI element.

[**0090**] Following block **110**, the multipoint processing method **100** proceeds to blocks **112** and **114**. The blocks **112** and **114** can be performed approximately at the same time. From the user perspective, in one embodiment, the blocks **112** and **114** appear to be performed concurrently. In block **112**, one or more actions are performed based on differences between initial and current parameter values as well as the UI element to which they are associated. In block **114**, user feedback pertaining to the one ore more action being performed is provided. By way of example, user feedback may include display, audio, tactile feedback and/or the like.

[**0091**] **FIG. 5** is a parameter calculation method **150**, in accordance with one embodiment of the present invention. The parameter calculation method **150** may, for example, correspond to block **108** shown in **FIG. 2**. The parameter calculation method **150** generally begins at block **152** where a group of features is received. Following block **152**, the parameter calculation method **150** proceeds to block **154** where a determination is made as to whether or not the number of features in the group of features has changed. For example, the number of features may have changed due to the user picking up or placing an additional finger. Different fingers may be needed to perform different controls (e.g., tracking, gesturing). If the number of features has changed, the parameter calculation method **150** proceeds to block **156** where the initial parameter values are calculated. If the number stays the same, the parameter calculation method **150** proceeds to block **158** where the current parameter values are calculated. Thereafter, the parameter calculation method **150** proceeds to block **150** where the initial and current parameter values are reported. By way of example, the initial parameter values may contain the average initial distance between points (or Distance (AVG) initial) and the current parameter values may contain the average current distance between points (or Distance (AVG) current). These may be compared in subsequent steps in order to control various aspects of a computer system.

[**0092**] The above methods and techniques can be used to implement any number of GUI interface objects and actions. For example, gestures can be created to detect and effect a user command to resize a window, scroll a display, rotate an object, zoom in or out of a displayed view, delete or insert text or other objects, etc. Gestures can also be used to invoke and manipulate virtual control interfaces, such as volume knobs, switches, sliders, handles, knobs, doors, and other widgets that may be created to facilitate human interaction with the computing system.

[**0093**] To cite an example using the above methodologies, and referring to **FIGS. 6A-6G**, a rotate gesture for controlling a virtual volume knob **170** on a GUI interface **172** of a display **174** of a tablet PC **175** will be described. In order to actuate the knob **170**, the user places their fingers **176** on a multipoint touch screen **178**. The virtual control knob may already be displayed, or the particular number, orientation or profile of the fingers at set down, or the movement of the fingers immediately thereafter, or some combination of these and other characteristics of the user's interaction may invoke the virtual control knob to be displayed. In either case, the computing system associates a finger group to the virtual control knob and makes a determination that the user intends to use the virtual volume knob. This association may also be based in part on the mode or current state of the computing device at the time of the input. For example, the same gesture may be interpreted alternatively as a volume know gesture if a song is currently playing on the computing device, or as a rotate command if an object editing application is being executed. Other user feedback may be provided, including for example audible or tactile feedback.

[**0094**] Once knob **170** is displayed as shown in **FIG. 6A**, the user's fingers **176** can be positioned around the knob **170** similar to if it were an actual knob or dial, and thereafter can be rotated around the knob **170** in order to simulate turning the knob **170**. Again, audible feedback in the form of a clicking sound or tactile feedback in the form of vibration, for example, may be provided as the knob **170** is "rotated." The user may also use their other hand to hold the tablet PC **175**.

[**0095**] As shown in **FIG. 6B**, the multipoint touch screen **178** detects at least a pair of images. In particular, a first image **180** is created at set down, and at least one other image **182** is created when the fingers **176** are rotated. Although only two images are shown, in most cases there would be many more images that incrementally occur between these two images. Each image represents a profile of the fingers in contact with the touch screen at a particular instant in time. These images can also be referred to as touch images. It will be understood that the term "image" does not mean that the profile is displayed on the screen **178** (but rather imaged by the touch sensing device). It should also be noted that although the term "image" is used, the data may be in other forms representative of the touch plane at various times.

[**0096**] As shown in **FIG. 6C**, each of the images **180** and **182** is converted to a collection of features **184**. Each feature **184** is associated with a particular touch as for example from the tips each of the fingers **176** surrounding the knob **170** as well as the thumb of the other hand **177** used to hold the tablet PC **175**.

[**0097**] As shown in **FIG. 6D**, the features **184** are classified, i.e., each finger/thumb is identified, and grouped for each of the images **180** and **182**. In this particular case, the features **184**A associated with the knob **170** are grouped together to form group **188** and the feature **184**B associated with the thumb is filtered out. In alternative arrangements, the thumb feature **184**B may be treated as a separate feature by itself (or in another group), for example, to alter the input or operational mode of the system or to implement another gesture, for example, a slider gesture associated with an equalizer slider displayed on the screen in the area of the thumb (or other finger).