data currently outside of the viewing area. The viewing area may be the entire viewing area of the display screen or it may only be a portion of the display screen (e.g., a window frame).

[**0082**] As mentioned above, scrolling may be implemented vertically (up or down) or horizontally (left or right). In the case of vertical scrolling, when a user scrolls down, each new set of data appears at the bottom of the viewing area and all other sets of data move up one position. If the viewing area is full, the top set of data moves out of the viewing area. Similarly, when a user scrolls up, each new set of data appears at the top of the viewing area and all other sets of data move down one position. If the viewing area is full, the bottom set of data moves out of the viewing area.

[**0083**] By way of example, the display screen, during operation, may display a list of media items (e.g., songs). A user is able to linearly scroll through the list of media items by moving his or her finger across a touch screen. As the finger moves across the touch screen, the displayed items from the list of media items are varied such that the user is able to effectively scroll through the list of media items. In most cases, the user is able to accelerate their traversal of the list of media items by moving his or her finger at greater speeds. Some embodiments, which may be related to the above example, are described in greater detail below. See for example **FIGS. 6, 23, 27**.

[**0084**] **FIG. 2** is a multipoint processing method **100**, in accordance with one embodiment of the present invention. The multipoint processing method **100** may for example be performed in the system shown in **FIG. 1**. The multipoint processing method **100** generally begins at block **102** where images are read from a multipoint input device, and more particularly a multipoint touch screen. By way of example, the multipoint touch screen may generally correspond to the multipoint touch screen disclosed in copending U.S. patent application Ser. No. 10/840,862, which is hereby incorporated herein by reference. Although the term "image" is used it should be noted that the data may come in other forms. In most cases, the image read from the touch screen provides magnitude (Z) as a function of position (x and y) for each sensing point or pixel of the touch screen. The magnitude may, for example, be reflect the capacitance measured at each point.

[**0085**] Following block **102**, multipoint processing method **100** proceeds to block **104** where the image is converted into a collection or list of features. Each feature represents a distinct input such as a touch. In most cases, each feature includes its own unique identifier (ID), x coordinate, y coordinate, Z magnitude, angle θ, area A, and the like. By way of example, **FIGS. 3A and 3B** illustrate a particular image **120** in time. In image **120**, there are two features **122** based on two distinct touches. The touches may for example be formed from a pair of fingers touching the touch screen. As shown, each feature **122** includes unique identifier (ID), x coordinate, y coordinate, Z magnitude, angle θ, and area A. More particularly, the first feature **122**A is represented by ID_1, x_1, y_1, Z_1, θ_1, A_1 and the second feature **122**B is represented by ID_2, x_2, y_2, Z_2, θ_2, A_2. This data may be outputted for example using a multitouch protocol.

[**0086**] The conversion from data or images to features may be accomplished using methods described in copending U.S. patent application Ser. No. 10/840,862 which is hereby incorporated herein by reference. As disclosed therein, the raw data is received. The raw data is typically in a digitized form, and includes values for each node of the touch screen. The values may be between 0 and 256 where 0 equates to no touch pressure and 256 equates to full touch pressure. Thereafter, the raw data is filtered to reduce noise. Once filtered, gradient data, which indicates the topology of each group of connected points, is generated. Thereafter, the boundaries for touch regions are calculated based on the gradient data, i.e., a determination is made as to which points are grouped together to form each touch region. By way of example, a watershed algorithm may be used. Once the boundaries are determined, the data for each of the touch regions are calculated (e.g., x, y, Z, θ, A).

[**0087**] Following block **104**, multipoint processing method **100** proceeds to block **106** where feature classification and groupings are performed. During classification, the identity of each of the features is determined. For example, the features may be classified as a particular finger, thumb, palm or other object. Once classified, the features may be grouped. The manner in which the groups are formed can widely varied. In most cases, the features are grouped based on some criteria (e.g., they carry a similar attribute). For example, the two features shown in **FIGS. 3A and 3B** may be grouped together because each of these features is located in proximity to each other or because they are from the same hand. The grouping may include some level of filtering to filter out features that are not part of the touch event. In filtering, one or more features may be rejected because they either meet some predefined criteria or because they do not meet some criteria. By way of example, one of the features may be classified as a thumb located at the edge of a tablet PC. Because the thumb is being used to hold the device rather than being used to perform a task, the feature generated therefrom is rejected, i.e., is not considered part of the touch event being processed.

[**0088**] Following block **106**, the multipoint processing method **100** proceeds to block **108** where key parameters for the feature groups are calculated. The key parameters may include distance between features, x/y centroid of all features, feature rotation, total pressure of the group (e.g., pressure at centroid), and the like. As shown in **FIG. 4**, the calculation may include finding the centroid C, drawing a virtual line **130** to each feature from the centroid C, defining the distance D for each virtual line (D_1 and D_2), and then averaging the distances D_1 and D_2. Once the parameters are calculated, the parameter values are reported. The parameter values are typically reported with a group identifier (GID) and number of features within each group (in this case three). In most cases, both initial and current parameter values are reported. The initial parameter values may be based on set down, i.e., when the user sets their fingers on the touch screen, and the current values may be based on any point within a stroke occurring after set down. As should be appreciated, blocks **102-108** are repetitively performed during a user stroke thereby generating a plurality of sequentially configured signals. The initial and current parameters can be compared in later steps to perform actions in the system.

[**0089**] Following block **108**, the process flow proceeds to block **110** where the group is or associated to a user interface (UI) element. UI elements are buttons boxes, lists, sliders,