provide such measurements (and others) directly. The control processor computes the desired feedback force for each haptel and drives the actuators to generate the appropriate forces. The haptic response of each haptel may be configured to be essentially arbitrary within a certain range. The range of available effects depends on the type of sensors employed, the bandwidth and precision of the sensors and effectors, the resolution of the analog-to-digital and digital-to-analog conversion performed, the amount of available processing power and the update frequency of the control loop, among other factors. These tradeoffs would be apparent to one skilled in the art of force feedback design.

[**0023**] Because the touchable area is comprised of many haptels, each of which can function independently, the device allows multiple touches at once. Each haptel responds to only one touch at a time, so that there is a lower bound on the distance between two touches which do not interfere with each other. The worst-case value of this minimum distance is approximately the diagonal size of a haptel. However, in a specific instance the minimum distance can be substantially smaller depending on the locations of the two touches. Smaller haptels allow touches to be closer to one another.

[**0024**] A typical interaction is a user pressing a graphical button displayed as part of a GUI. The finger touches the device, landing on a specific haptel. The overall location of the touch is determined by the touch location sensor of the haptel in combination with the location of that haptel within the haptel grid. The touch location is communicated to a processor (e.g., a computer) which discovers that a graphical button is "underneath" the touch, and therefore communicates this information to the control processor to use a "button" haptic response for this touch. As the user presses down on the haptel, the control processor responds with a feedback force which increases as the surface is depressed until the position reaches a certain threshold, at which point the feedback force is quickly reduced. This causes the applied force to momentarily exceed the feedback force, which results in the quick downward movement of the haptel surface. In this way a "clicking" sensation is conveyed to the user. Preferably, the computer is continually informed of the state of the touch so that when the haptel reaches the bottom of its travel, the computer executes the action represented by the graphical button and displays the button in its activated state.

[**0025**] If the graphical button is disabled, the computer has the control processor use a "disabled button" haptic response. In this response the feedback force increases with position at a higher rate than the "button" response with no force drop-off. This creates the sensation of an unyielding surface which informs the user than the action represented by the graphical button cannot be initiated.

[**0026**] The preceding descriptions assume that each touch falls within the bounds of a single haptel, but this need not be the case. If the touchable area of the device is mapped to a GUI in which interface elements can be placed anywhere, some will happen to be located on the edge between two haptels or the vertex where four haptels meet. A touch on such a control is therefore likely land on more than one haptel. Such "border touches" can be transparently handled by the device. The first step is to merge related touches. If two touches appear simultaneously on adjacent haptels a short distance apart, the device can safely infer that the touches are really a single touch on the border between those two haptels. Similar inferences can be made for touches that appear simultaneously near the vertex of any number of haptels.

[**0027**] Once the set of haptels is determined, the haptels are managed in a coordinated fashion. The center of the touch is computed, preferably by weighting each touch location by the force applied to that haptel, and then dividing by the total force applied to the haptels involved. Likewise, the collective surface position, velocity, and acceleration are computed, preferably by weighted average of the haptels involved. Other weightings are possible, including equal weighting of values. The applied force measurements of the haptels involved may be summed to compute the total force applied. The haptic response is then computed from these collective measurements in much the same way they would be computed for a single haptel, resulting in a collective feedback force. This feedback force is distributed across the haptels involved in the touch in proportion to the amount of the total applied force lands on each haptel. In addition, a restoring force pulls the haptels towards the collective position to prevent surfaces from drifting apart due to measurement errors and other factors. As a result, the total feedback force is effectively distributed across the haptels involved in the touch, and the haptel's surfaces will have similar position, velocity, and acceleration. This provides the illusion that a single surface was pressed, making the coordinated nature of the touch undetectable by the user.

[**0028**] Not only can such device coordinate a fixed set of haptels, but it can also transparently add and remove haptels from the coordination set over time. This is necessary during "dragging" operations in which touches move across the device. When a touch gets close to another haptel, the newly-added haptel is added to the coordination set. This has the effect of causing its surface to become flush with the haptels already involved in the touch. Preferably, this is done without affecting the feel of the touch in progress. When the touch moves far enough away from a given haptel, that haptel is removed from the coordination set, leaving it free to participate in another touch.

[**0029**] This coordination effectively makes the haptels' gridded nature invisible to the user and to software applications. The computer specifies the response for a touch in a declarative fashion, and the device ensures that this response will be generated regardless of where the touch falls, how many haptels are involved in the touch, or whether the touch moves. Device-specific information provided to the computer might include the minimum allowed distance between independent touches, so that the computer can separate controls designed for simultaneous use appropriately or give feedback to the user when one touch ventures too close to another.

[**0030**] The foregoing is a summary and thus contains, by necessity, simplifications, generalizations and omissions of detail; consequently, those skilled in the art will appreciate that the summary is illustrative only and is not intended to be in any way limiting. Other aspects, inventive features, and advantages of the present invention, as defined solely by the claims, will become apparent in the non-limiting detailed description set forth below.