form a horizontal line, and the detected fingers form a line that is not quite horizontal, then the output from step **506** may include an indication that the detected gesture was not a perfect match. For example, the system may generate a similarity value (S_T) indicating how similar the detected gesture actually was to the gesture's template. Each template may define threshold values and/or ranges for determining similarity. For example, the template for the two-finger line example may indicate that a horizontal line has a perfect similarity value of 1.0; a line askew by less than 10 degrees may have a similarity value of 0.8; and a line askew by 11 to 20 degrees may have a similarity value of 0.5. The output can be data written to memory, a control signal to another application or device, viewable results of a command executed in response to the gesture (e.g., displaying a message confirming the command was executed, a prompt associated with the command, etc.), or any other desired form.

If no match is found in step **506** between a template and the gesture region of the image, the process may provide an output indicating this fact. Then, the process may conclude, or it may return to step **502** to capture and process another image.

The discussion of FIG. **5** is in the context of touch-based gestures, but the same or a similar process may be used for non-touch gestures using camera **203**. For example, non-touch gestures made near the screen **201** may still reflect enough infrared light to be detected by camera **203**.

Additionally, camera **204** may also be used in a similar manner to capture images from above screen **201**, and to determine whether a touch or non-touch gesture has been performed. For example, camera **204** may provide sequential frames of images, and the system **100** may analyze the frames to separate out the user's hands from a background image, to recognize the positioning of the hands, and to compare them with predetermined templates of known gestures to identify a gesture made above or on the screen **201**. A given gesture's template may also include template information for the camera **204**. For example, the template for a hand press gesture may include a standard image, from above screen **201**, showing a generic hand touching the screen **201** in a hand pressing gesture. The detected image from camera **204** may be compared against this standard image to determine whether it appears, from camera **204**'s point of view, that a hand press gesture was made, and the system may generate a similarity value (S_H) indicating the strength of this identification.

FIG. **6** illustrates an example process for an application that uses both touch and non-touch gesture detection systems for user input. This may be, for example, the sand simulation described above. In this process, the two cameras **203**, **204** may each provide their own determination of whether a given touch or non-touch gesture has been detected.

In step **601**, the process begins by capturing and processing camera data to detect gestures, as discussed above with respect to FIG. **5**. In step **602**, a determination may be made as to whether a start command has been detected. A start command may be any desired input (e.g., gesture, keyboard, etc.) signaling the application that the combined touch/non-touch gesture inputs are about to begin. For example, the sand program discussed above may define a start command as a waving of the hand over the screen **201**, or by pressing a key on a keyboard input **106**, or any other desired input. In some embodiments, the start command may be configured to only use one type of input (e.g., touch, non-touch, keyboard, etc.), to minimize risk of ambiguity in gesture detection. If no start command has been detected, the process may return to step **601**. Note that even if no start command has been detected, a detected gesture may still have meaning to a different application running in parallel (other than the one following the example FIG. **6** process).

If a start command has been detected, then in step **603**, the system may perform the process from FIG. **5** to determine whether a gesture has been detected from camera **203**'s point of view. The camera **203** may be most suited for detecting touch gestures, but it may also attempt to detect non-touch gestures as well, as discussed above with respect to FIG. **5**. As output, the system may generate a gesture identification and similarity value (S_T) from camera **203**'s point of view.

Then, in step **604**, the system may perform the same (or similar) gesture detection process, but using the image from camera **204**. That camera **204** may be most suited for detecting non-touch gestures occurring above the screen **201**, but it may also attempt to detect touch gestures as well. As output, the system may generate a gesture identification and similarity value (S_H) from camera **204**'s point of view.

Then, in step **605**, the determinations from these systems are considered together to resolve the gesture into a command. This combined consideration may be performed according to the more detailed example shown in FIG. **7**, discussed below. After the gesture is detected, the system may determine, in step **606**, whether the gesture has resulted in an end command. The end command, like the start command, may be another predetermined input (e.g., gesture and/or keyboard) that signals to the application that the user wishes to end the application.

If the end command has been detected, then the process may proceed to step **607**, and the application operation may be finalized. The finalization may include any step needed to conclude the application, such as closing files opened by the application, releasing reserved memory resources, printing and/or saving a final image displayed on the screen **201**, etc.

If the end command was not detected, then the process may proceed to step **608**, and a command corresponding to the detected gesture may be executed. Various commands may each be assigned to one or more gestures, such as by storing a template file associating gestures with system/application commands. When a detected gesture corresponds to one of those commands, then the corresponding command may be executed in step **608**. In step **608**, if the detected gestures do not correspond to a command, then the system need not execute any command in step **608**.

After executing any corresponding command in step **608**, the process may return to step **603**, to attempt to detect the next touch and non-touch gestures. This process may repeat continuously while the application is executing. Additionally, it is noted above that the detection of a gesture need not always result in a corresponding command being executed. In some situations, a detected gesture may simply be a part of a longer gesture required for a corresponding command. To use the sand example, the command for picking up sand may be mapped to a sequence of gestures, such as a hand press touch gesture, followed by a non-touch grasp gesture (or making a first). The system may detect the hand press touch gesture, and pass through steps **606** and **608**, returning to **603** to detect the ensuing grasp gesture before executing the command for picking up the sand.

In the discussion of FIG. **6** above, the touch and non-touch gesture detections were considered together in step **605**, resulting in a determination of a recognized touch and/or non-touch gesture. This "fusion" of touch and non-touch gesture detection can be achieved in a variety of ways, to allow both touch and non-touch gestures to be integrated into a user interface scheme for the system **100**. FIG. **7** illustrates one example process for this fusion of step **605**.