[0023] FIG. **2** illustrates an example gesture detection system **200** that may be used with the computing system **100**. The gesture detection system **200** may include a horizontal display screen **201**, which may serve as display **104** to display graphical images to be seen by the user, as well as input **106** to receive input from the user (as will be described below). The screen **201** may be composed of a clear panel, such as glass or acrylic, to allow an image to be displayed via projection from below.

[0024] The system **200** may be configured to detect different types of input, such as touch-based inputs and non-touch-based inputs. Touch-based inputs may include placing an object, such as a user's finger, on the screen **201**. Non-touch-based inputs may include gestures made above the screen **201**, such as moving a user's hand over the system screen **201**. The systems for detecting these gestures may involve a series of cameras. For touch-based gestures, the system **200** may include a touch-based gesture detection system, having a series of light emitters **202** (which may be infrared) and camera **203**. Light from the emitters **202** may illuminate, and pass through, the underside of screen **201**.

[0025] When an object, such as the user's finger, is placed on top of the screen **201** from above, the object will reflect the light from emitters **202**. That reflected light may be detected by one or more cameras **203**. Images detected from camera **203** may be supplied to processor **101** for processing to determine where the screen **201** has been touched. Some touch-based gestures may incorporate movement (e.g., swiping a finger across a portion of screen **201** in a predetermined pattern), and in those cases the images from camera **203** may be sequential in time. FIG. **4**, discussed further below, provides an example process by which such gestures may be detected.

[0026] For non-touch-based gestures, the system may include a non-touch based gesture detection system, having a second camera **204**, positioned above the top of screen **201**. The second camera **204** may be a more traditional, visible light camera. Images captured from camera **204** may also be supplied to processor **101** for processing to determine where gestures were made above the screen **201**, and what types of gestures were made.

[0027] Although the two cameras **203**, **204** are described above as being useful for touch and non-touch gestures, respectively, they are each useful for both gesture types. In other words, the infrared camera **203** may also be used to detect non-touch gestures that occur near the screen **201** and are visible. Similarly, the visible light camera **204** may also detect touch gestures, for example, by determining whether the user's hand appears to be touching the screen **201**. In some embodiments herein, the combination of these two input detection approaches helps provide an effective input gesture detection mechanism.

[0028] FIG. **3** illustrates examples of touch and non-touch gestures that may be made with a user's hand. For example, non-touch gestures may involve finger configurations (such as the "Start" and "End" configurations illustrated) and/or movements (such as the "Grasping" and "Dropping" movements illustrated). Touch gestures may involve an object making contact with the screen **201**, such as one or more fingers pressing on the screen **201** at a location (or in a predetermined pattern), or a hand pressing on the screen **201**. Touch gestures may also involve movements across the screen **201**, such as a finger mopping motion of dragging the finger. In some embodiments described herein, touch and non-touch gestures can be combined, and used together to provide predetermined commands to the computing system.

[0029] FIGS. **4***a* and **4***b* illustrate an example application that can combine touch and non-touch gestures. The example application mimics the behavior of sand on the screen **201**. In FIG. **4***a*, the user may make a closed-first gesture above the screen **201** (signaling to the application that the first is holding sand), and the user may then open the hand to mimic dropping sand on the screen **201**. The computing system **100** may detect this gesture, and may display an image of a pile of sand under the user's hand. Then, the user may touch the screen **201** to clear away sand at the location of the touch; swipe/mop a finger across the surface to clear away patterns of sand; pick up sand from the screen **201** (e.g., a hand pressing touch gesture followed by a first);

[0030] and/or drop sand on the screen **201**, to create patterns and images in the sand, as illustrated in FIG. **4***b*.

[0031] FIG. **5** illustrates an example process by which a touch gesture may be detected using camera **203**. Steps in the process are shown on the left, and corresponding example images from camera **203** are shown on the right, to help illustrate the process. First, in step **501**, the camera **203** may be initialized to capture a background image in which no objects are placed on the screen **201**. Once the system has been initialized, it is ready to begin detecting touch gestures. In step **502**, the camera **203** may capture a new image or sequence of images, and in step **503**, the background image may be subtracted from the new image to yield a foreground image of the object(s) that were not present in the background during initialization. This subtraction may be performed using any desired image processing method.

[0032] In step **504**, the foreground image may be binarized to identify, on a per-pixel basis, whether an object is, or is not, present at that pixel. For example, the binarization may generate a binary (e.g., black and white) version of the foreground image, by comparing color or brightness values of a pixel with a predetermined threshold value, so that pixels having values above the threshold are indicated as white, and pixels having values below the threshold are indicated as black (or vice-versa).

[0033] In step **505**, the binarized foreground image may then be examined to isolate the region in which an object appears. In the example shown in FIG. **5**, the lower portion of the image shows the user's two hands, so this gesture region is extracted in step **505** (the gesture region is highlighted for explanation by a surrounding box in FIG. **5**). Any desired image process can be used to extract this region. For example, a connected-component method may be used, in which the brightest part of the image (e.g., the strongest point of actual contact) is first selected, and then nearby pixels having a similar brightness (or having a brightness that is within a predetermined threshold of the first point) are added, and the process continues, spreading to nearby pixels until there is a sufficient dropoff in image brightness between neighboring pixels. This method may define regions, such as the highlighted rectangular region illustrated beside step **505**.

[0034] When the gesture region has been identified, the image in that region may be compared in step **506** with predetermined templates to determine whether a gesture has been detected. This comparison may be performed using any desired image matching algorithm. For motion-based gestures, the template may comprise a series of predefined gesture images through the motion, and the steps above may be repeated to account for the various portions of the motion.