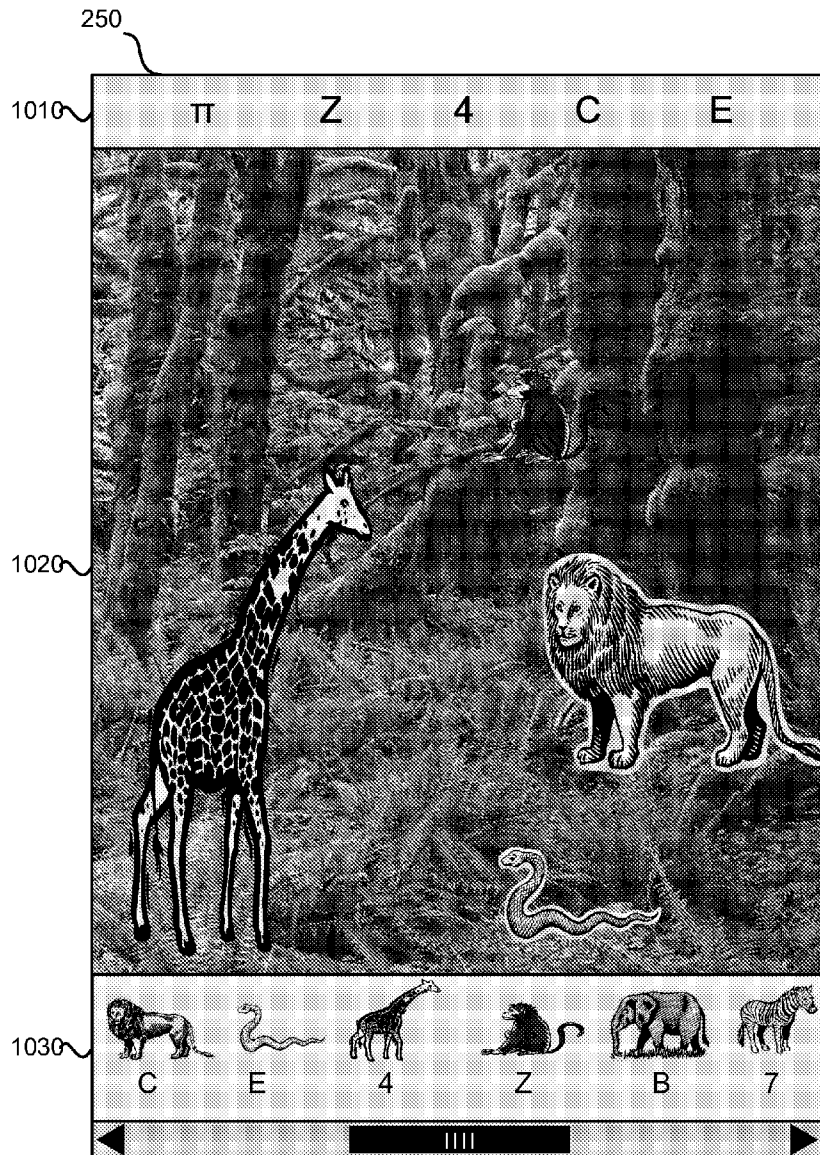




US 20110096997A1

(19) **United States**(12) **Patent Application Publication**
Marciszko et al.(10) **Pub. No.: US 2011/0096997 A1**(43) **Pub. Date: Apr. 28, 2011**(54) **GRAPHICAL IMAGE AUTHENTICATION****Related U.S. Application Data**(75) Inventors: **Tobias Marciszko, Solna (SE);
David De Leon, Lund (SE)**(63) Continuation of application No. 11/837,568, filed on
Aug. 13, 2007.(73) Assignee: **Sony Ericsson Mobile
Communications AB, Lund (SE)****Publication Classification**(21) Appl. No.: **12/669,008**(51) **Int. Cl.**
G06K 9/68 (2006.01)(22) PCT Filed: **Feb. 11, 2008**(52) **U.S. Cl.** **382/218**(86) PCT No.: **PCT/IB08/50491**(57) **ABSTRACT**§ 371 (c)(1),
(2), (4) Date:**Jan. 10, 2011**

A method may include receiving input associated with constructing a first image; generating the first image based on the input; comparing the first image with a second image; and providing access to at least one of a device, a service, or a function when the first image matches the second image.



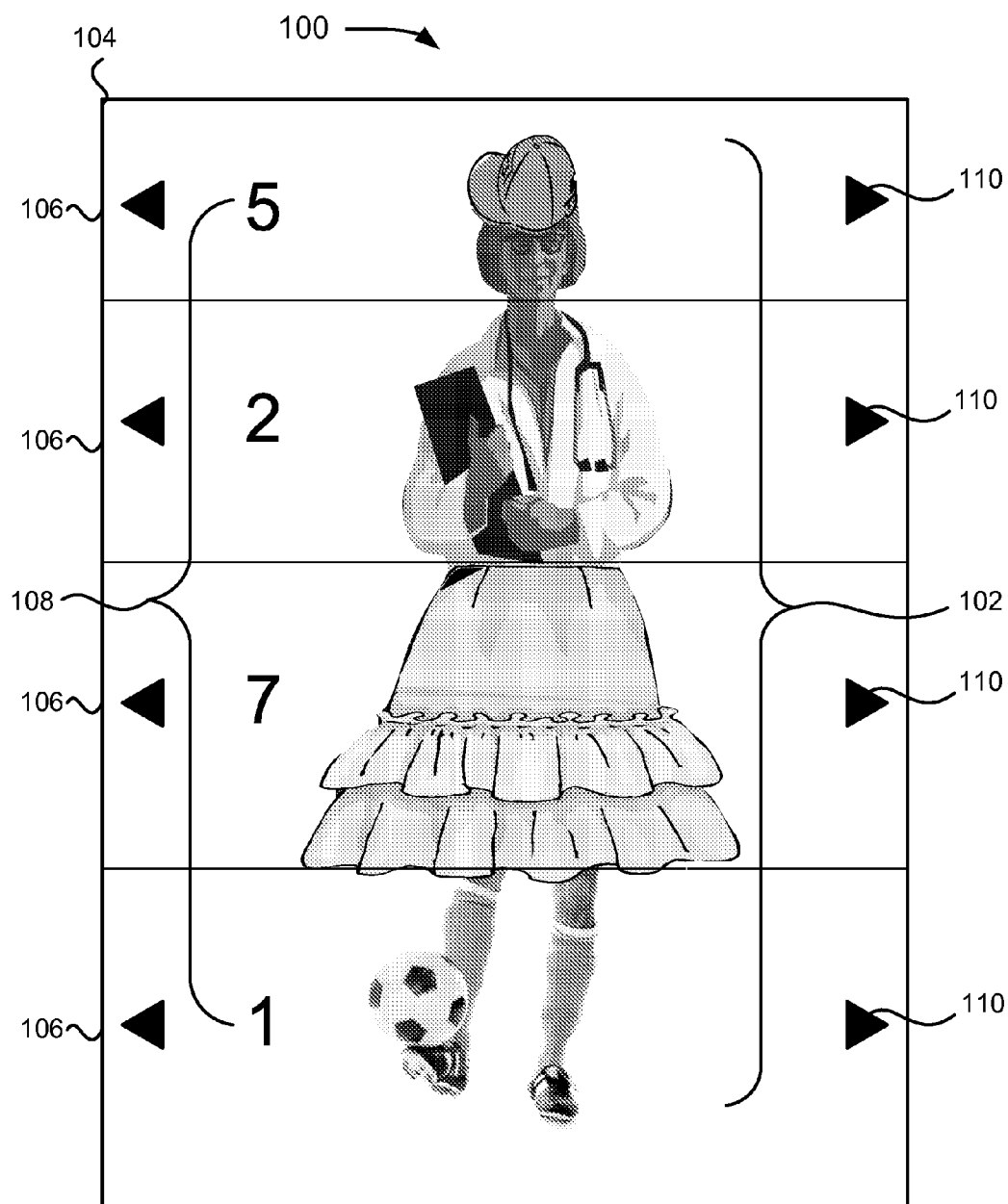


Fig. 1

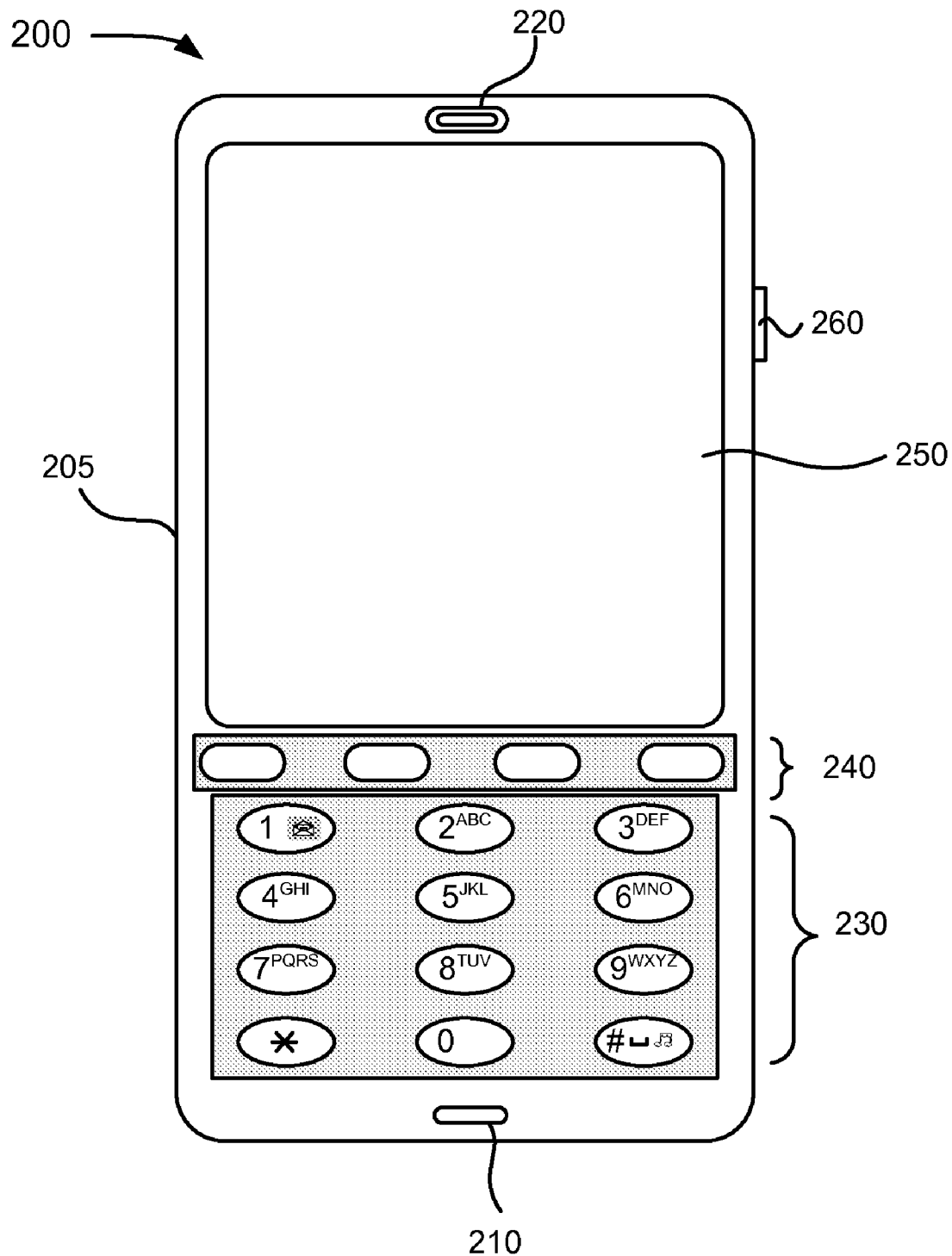


Fig. 2

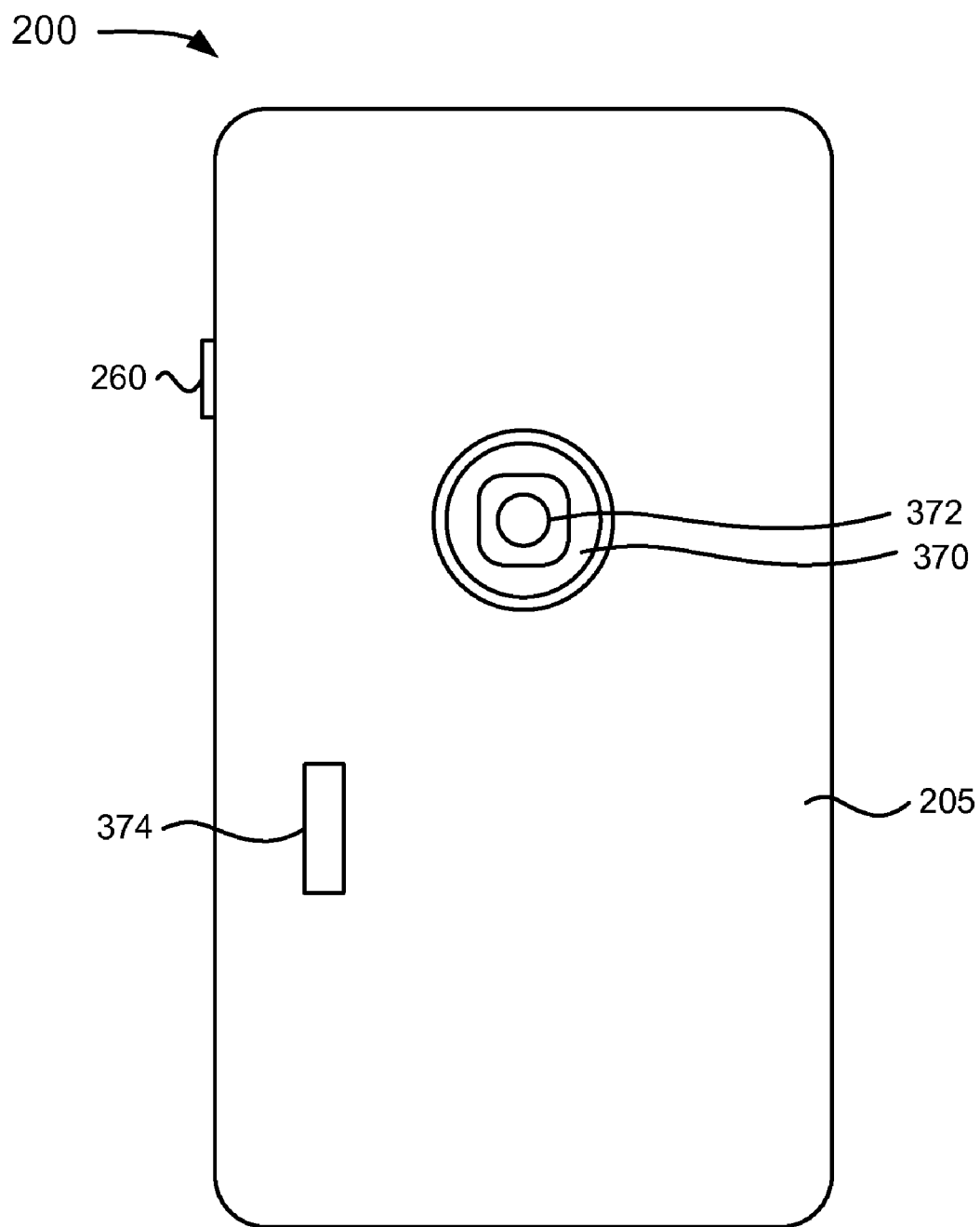


Fig. 3

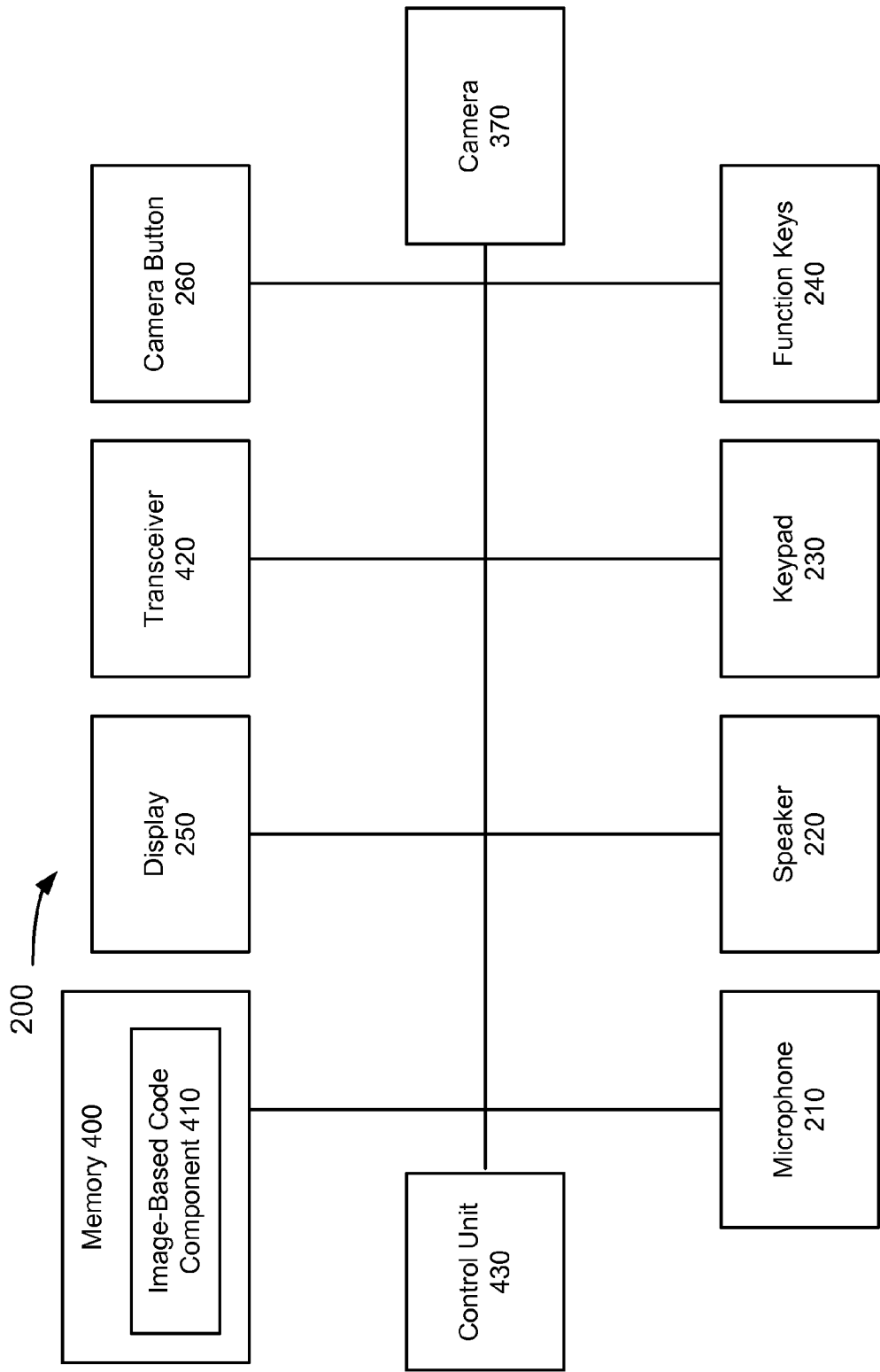


Fig. 4

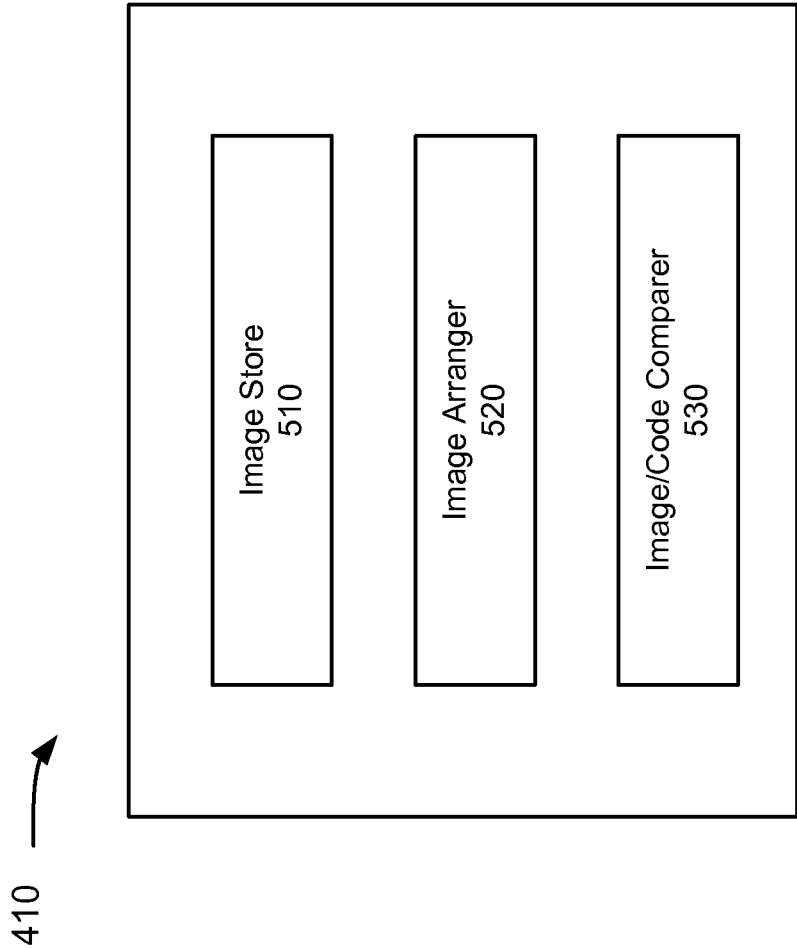


Fig. 5

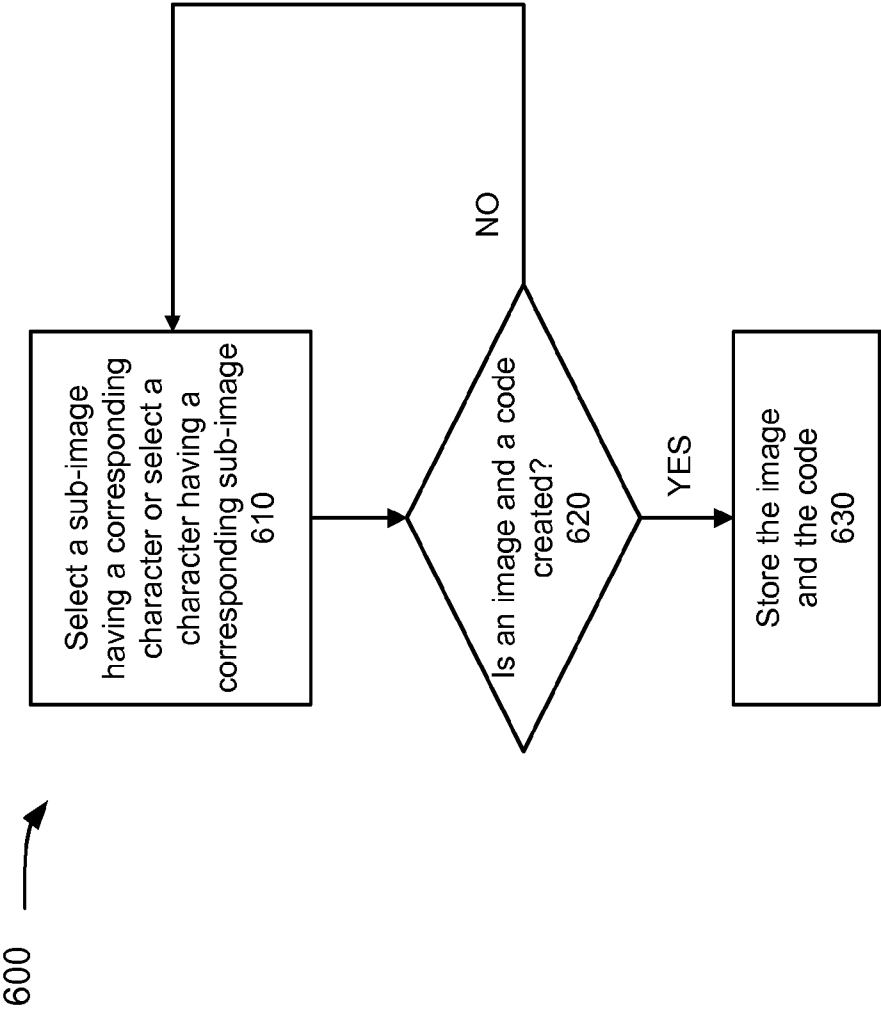
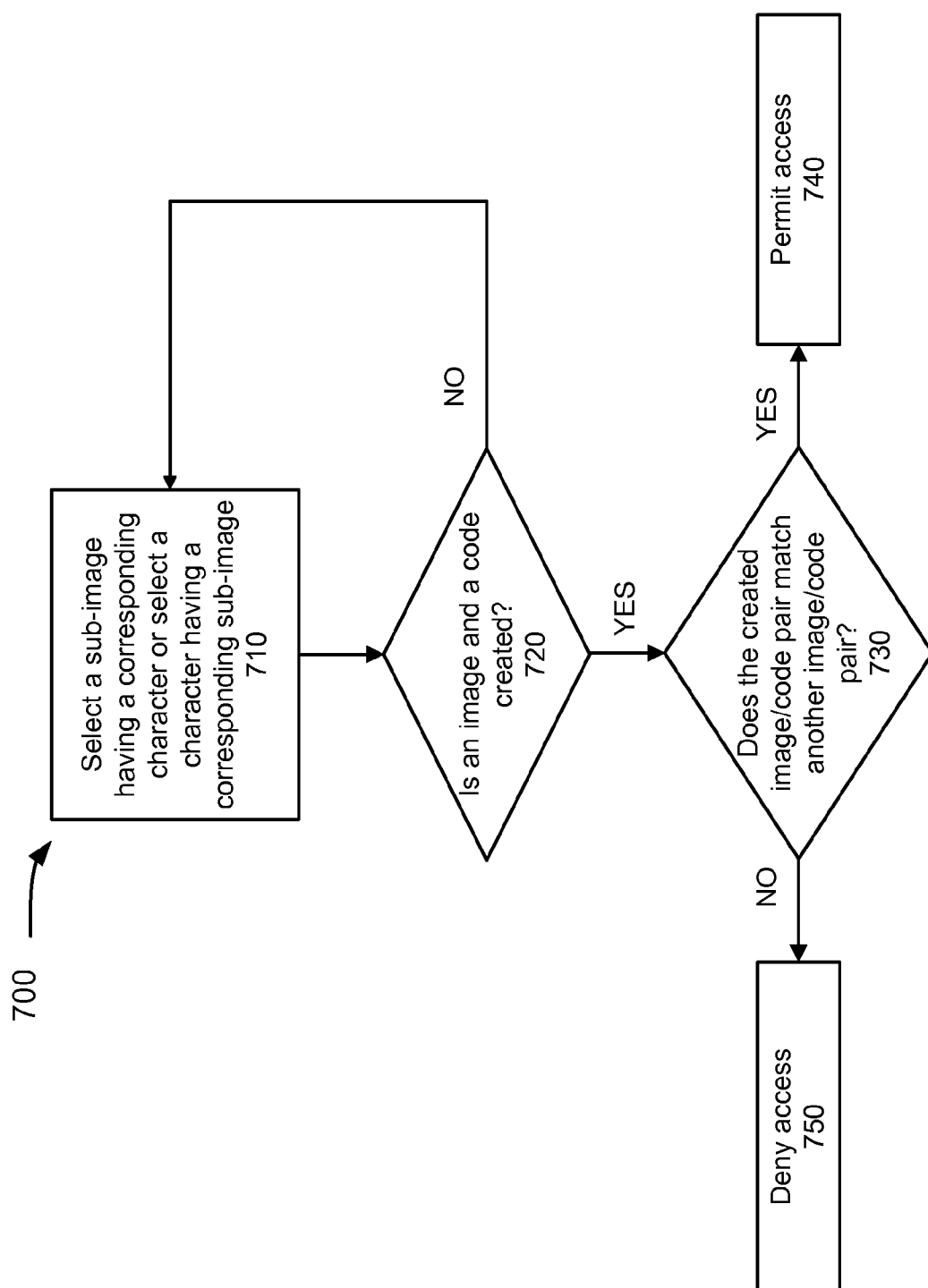


Fig. 6

**Fig. 7**

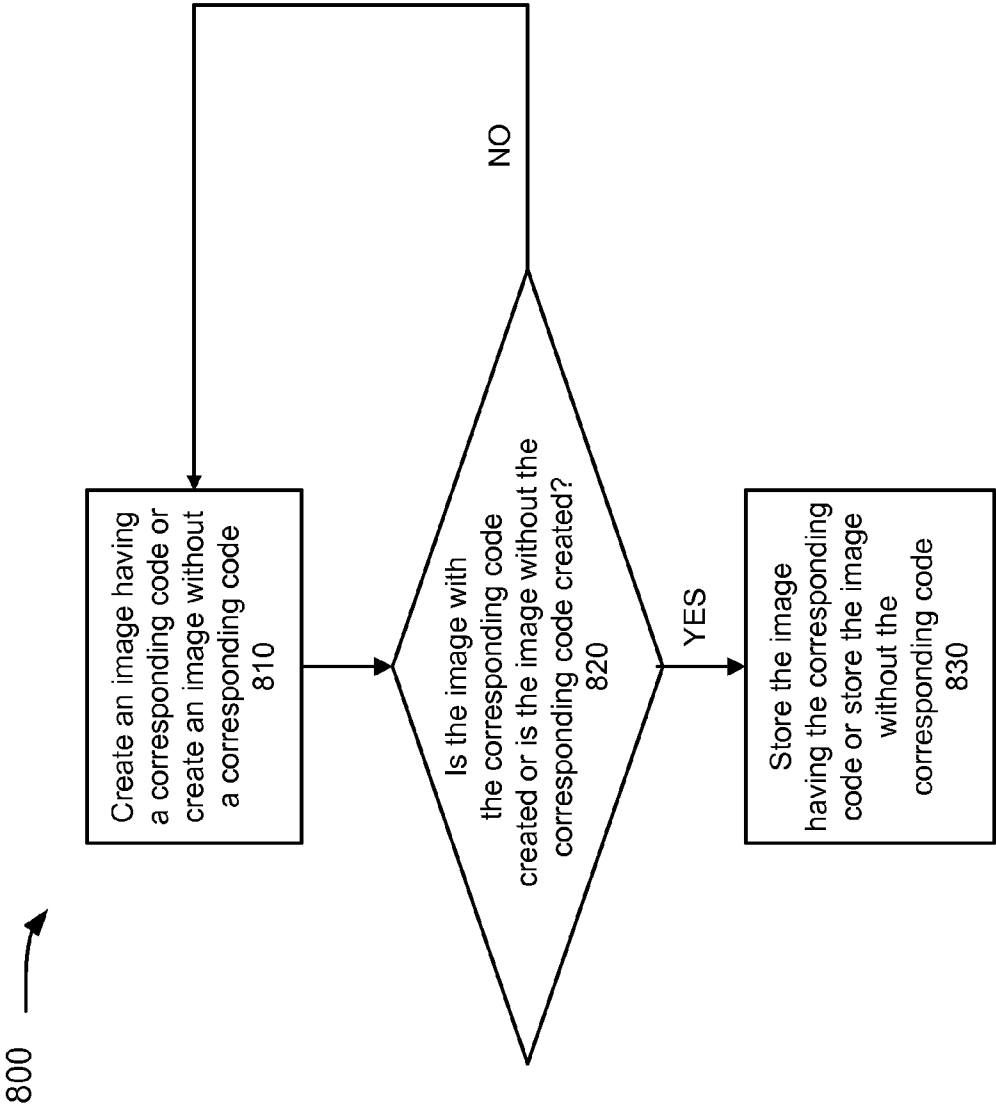


Fig. 8

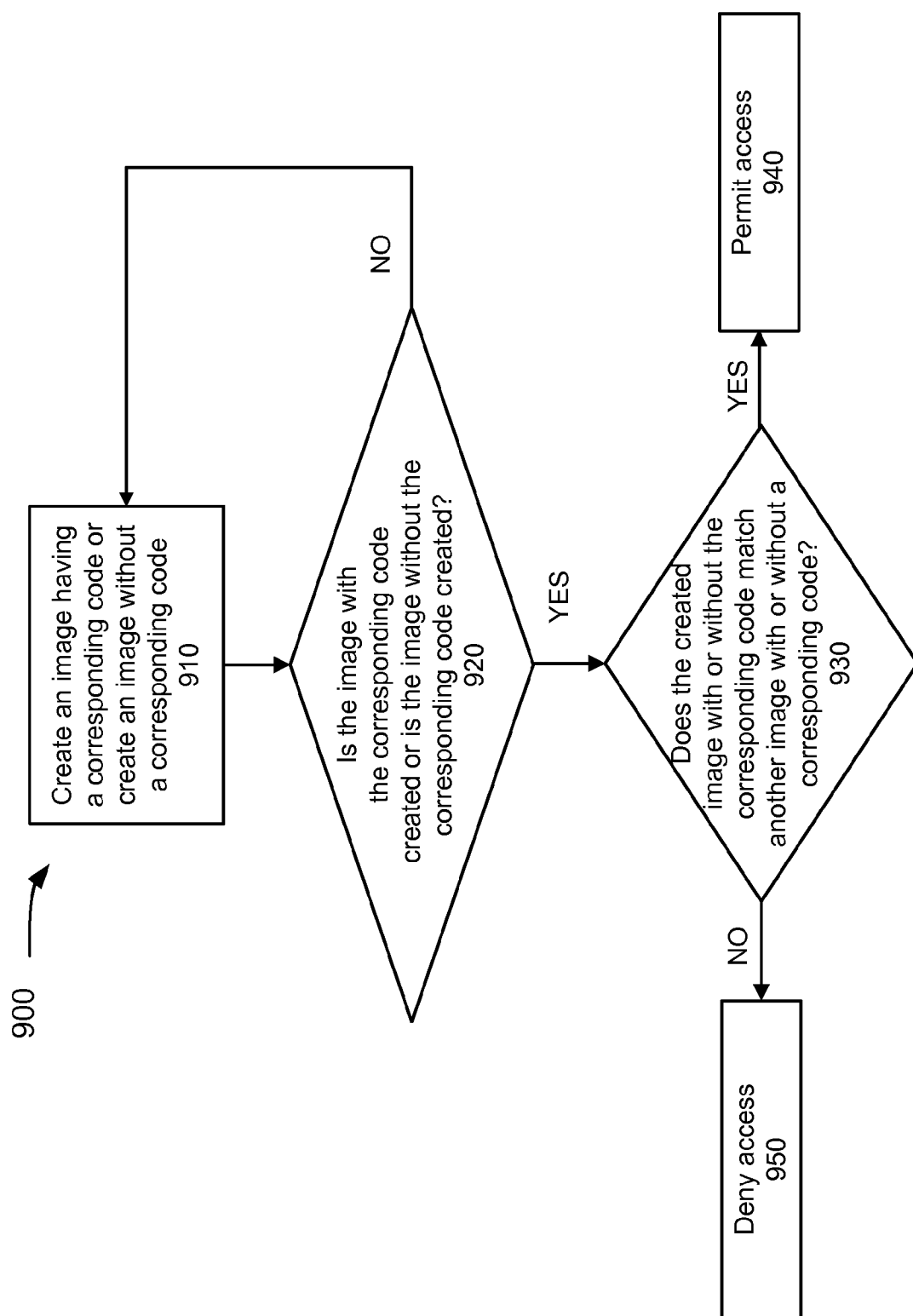


Fig. 9

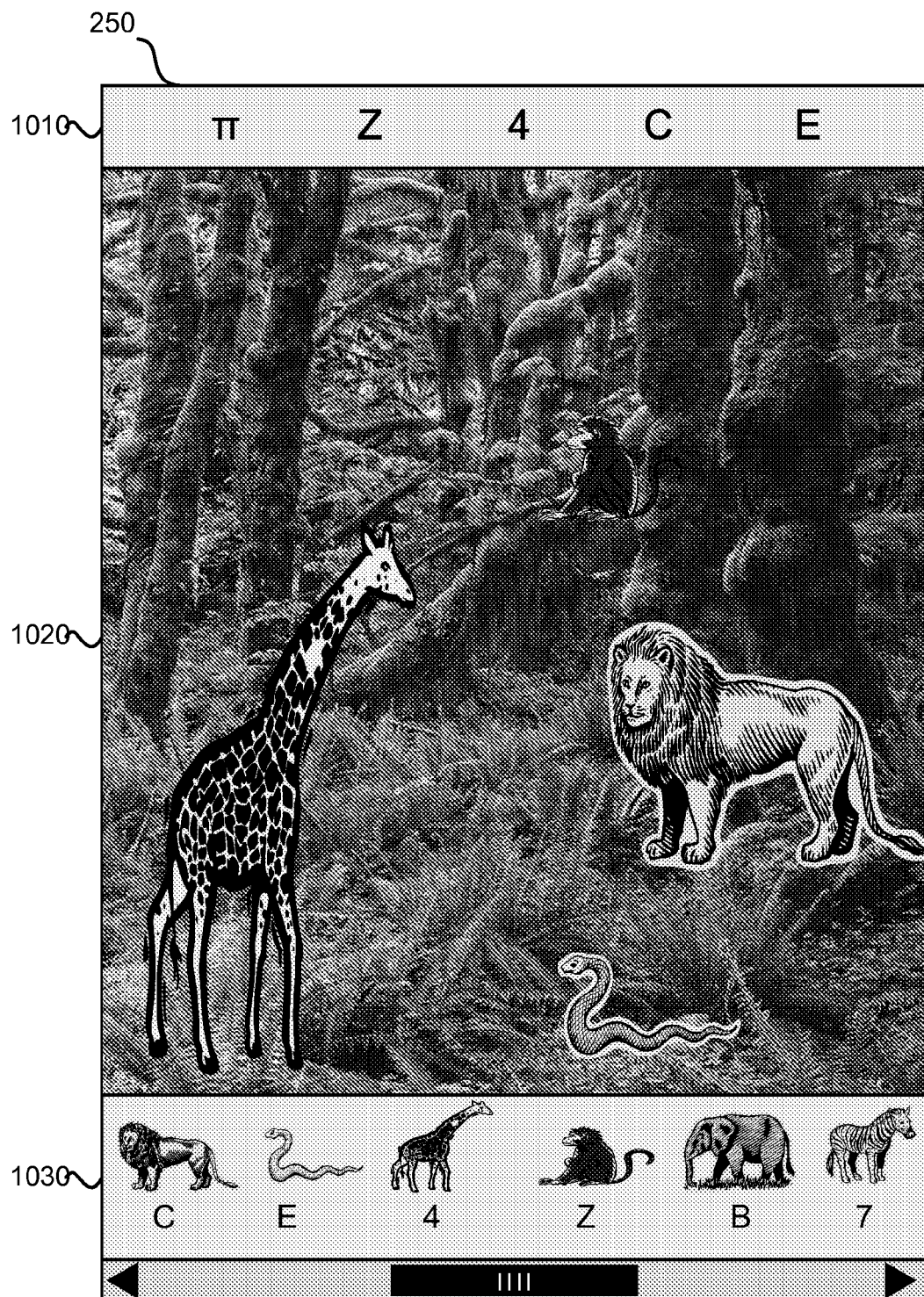


Fig. 10

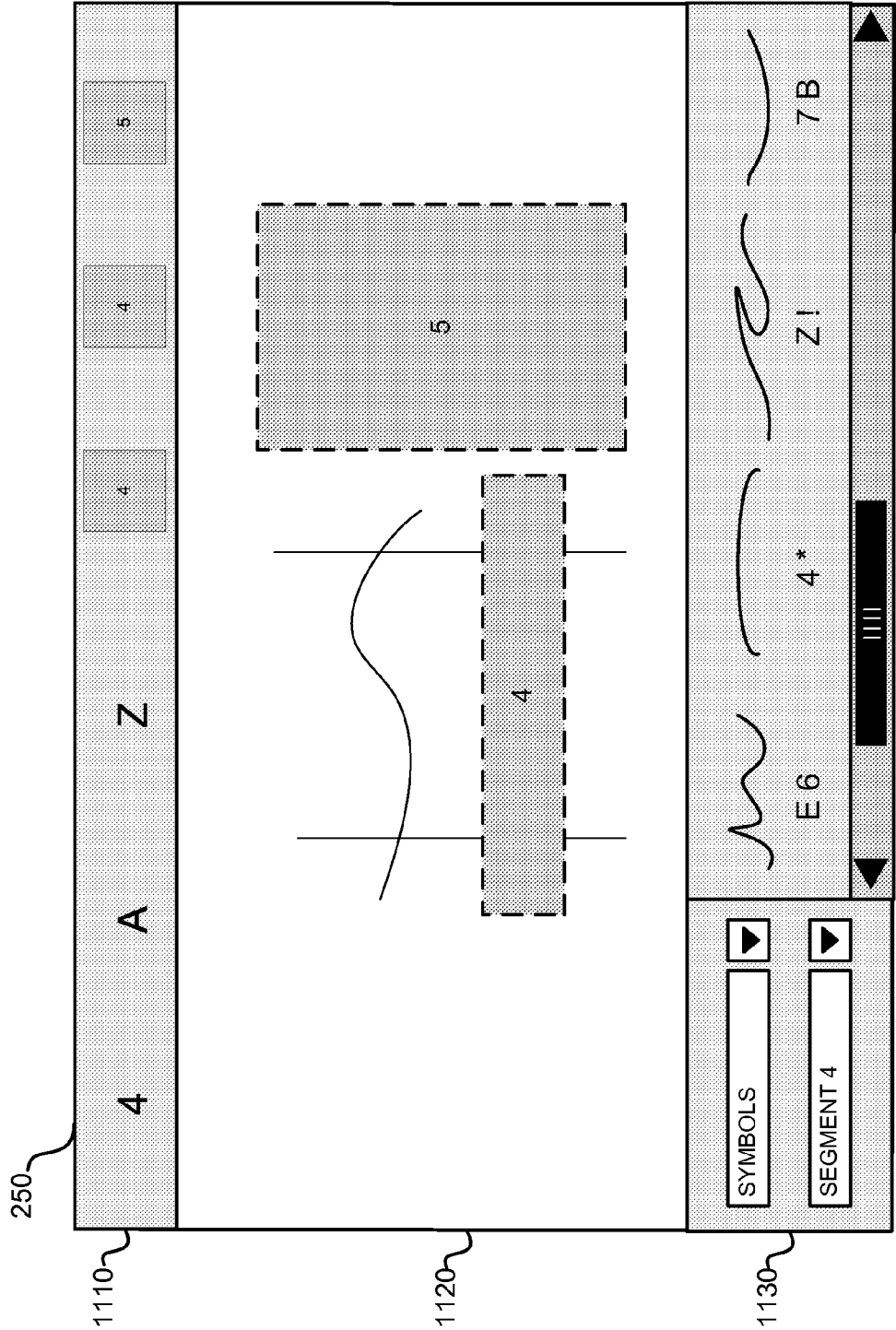


Fig. 11

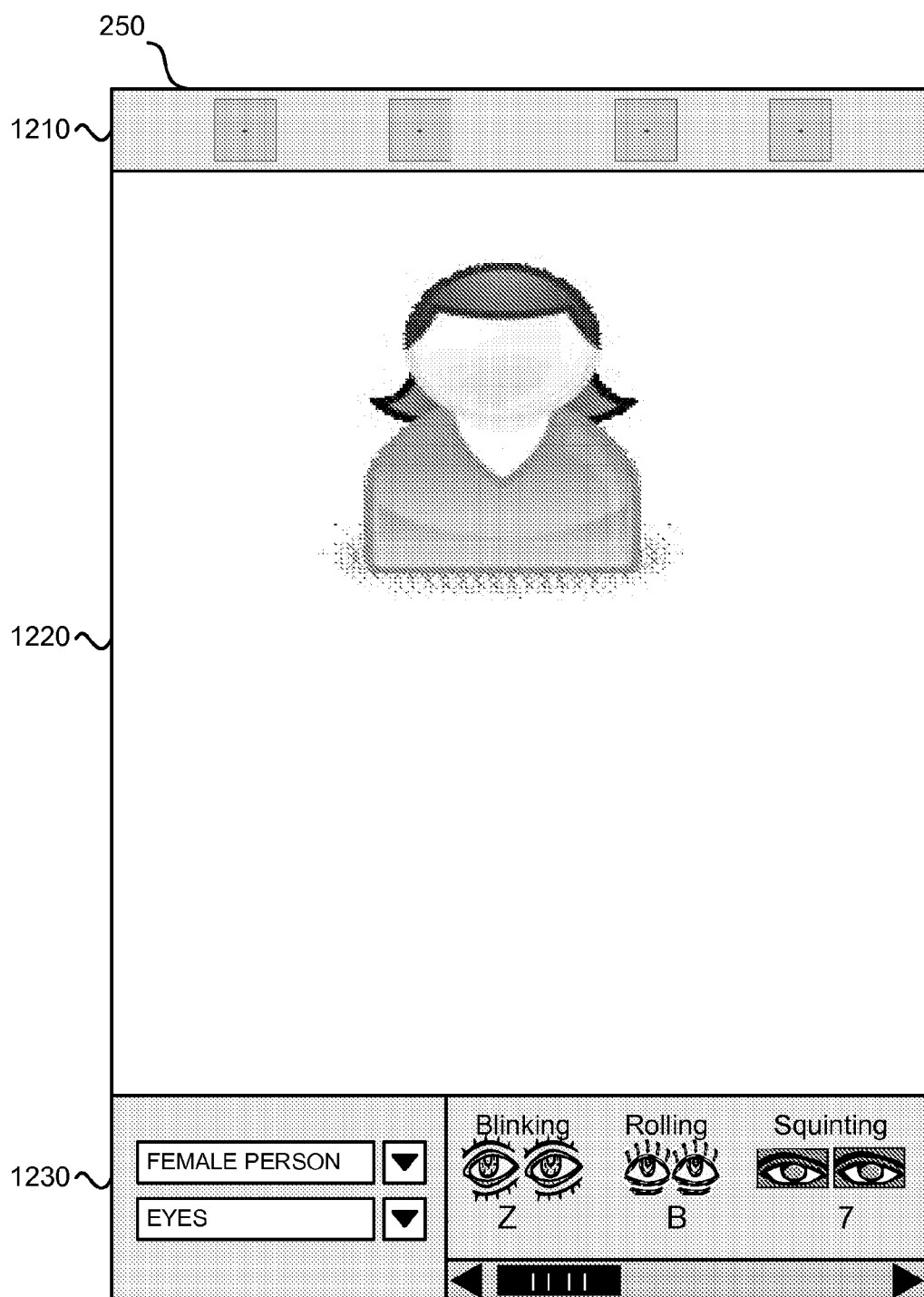


Fig. 12

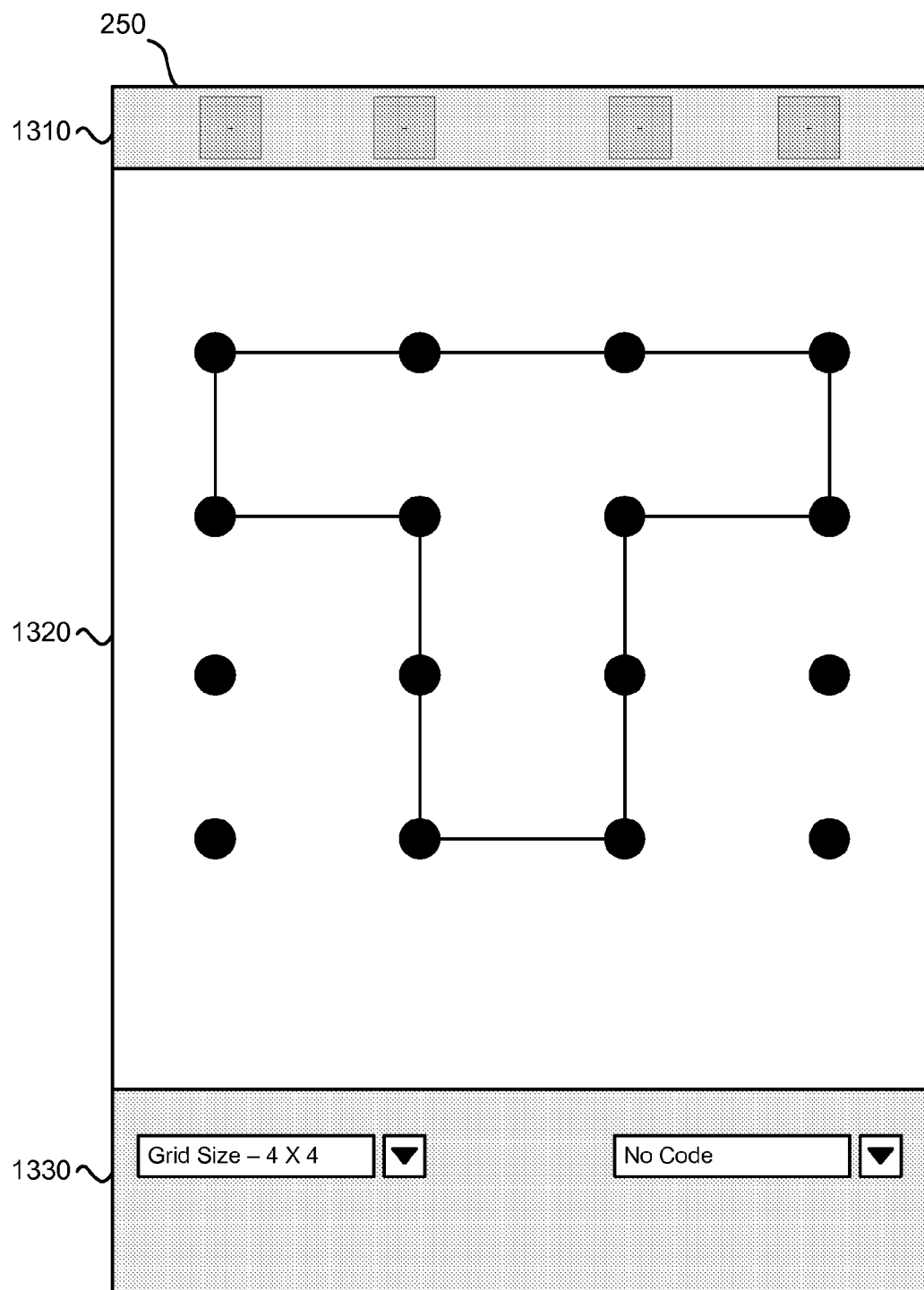


Fig. 13

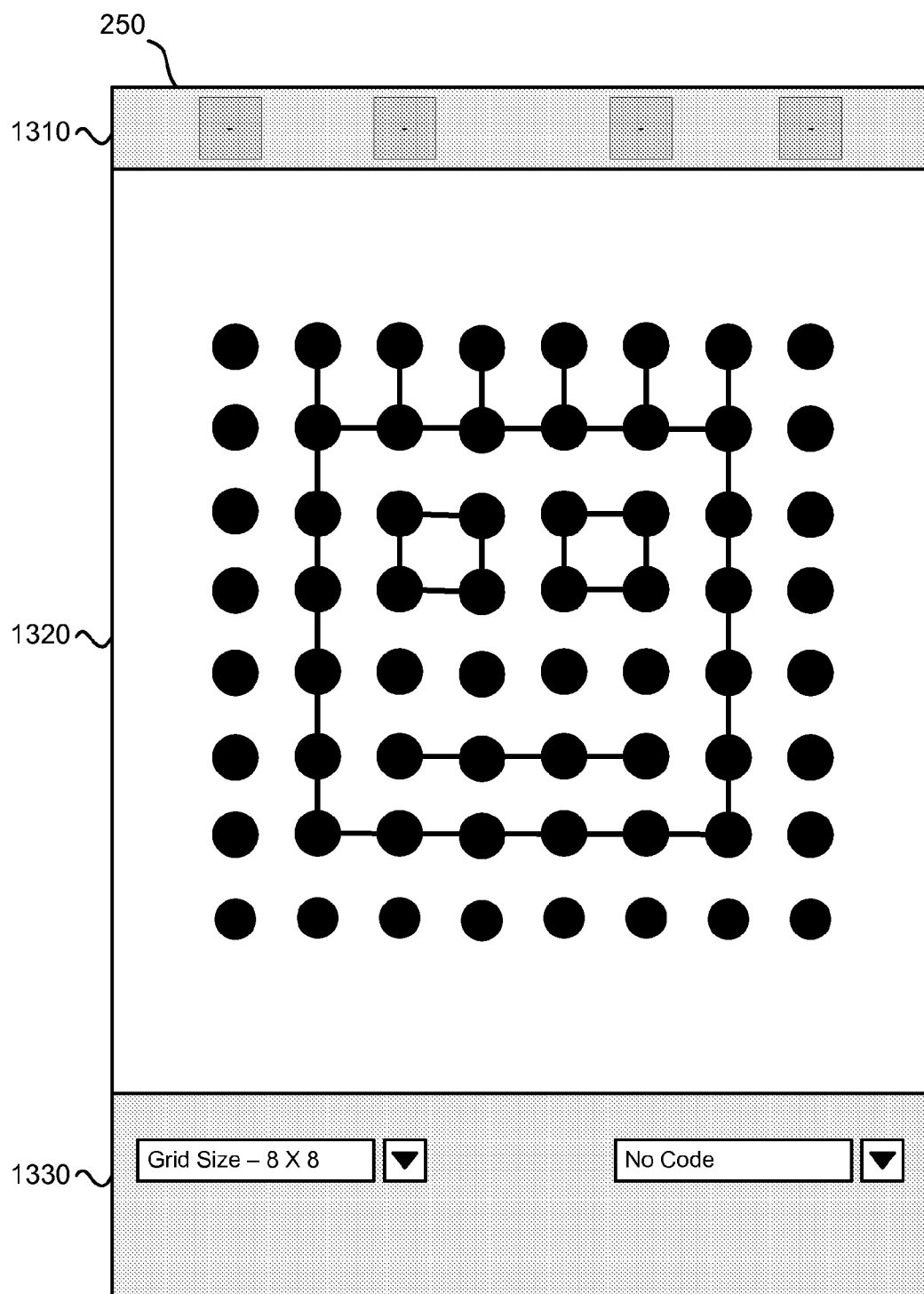


Fig. 14

GRAPHICAL IMAGE AUTHENTICATION

BACKGROUND

[0001] There are countless services, functions and devices that employ some form of password or personal identification number (PIN) for their use. Given the widespread utilization of these types of codes, most users have multiple codes that they have to remember. However, it is not uncommon among users to have difficulty recalling a code. While there are a variety of reasons why users suffer from this difficulty, it would be beneficial to provide users with an alternative approach.

SUMMARY

[0002] According to one aspect, a computer-readable medium may contain instructions executable by at least one processor. The computer-readable medium may include one or more instructions for providing sub-images, one or more instructions for receiving selection of a plurality of sub-images, one or more instructions for constructing a first image, the first image being a unified image including the selected plurality of sub-images, one or more instructions for comparing the selected plurality of sub-images with previously selected sub-images, and one or more instructions for providing access to at least one of a device, a service, or a function when the selected plurality of sub-images match the previously selected sub-images.

[0003] Additionally, the one or more instructions for providing sub-images may include one or more instructions for categorizing the sub-images.

[0004] Additionally, the selected plurality of sub-images may include at least one of two sub-images of a different category or at least one sub-image that is an animation.

[0005] Additionally, the one or more instructions for receiving selection may include one or more instructions for receiving selection of a character associated with the sub-image.

[0006] Additionally, each sub-image includes a corresponding character, and the unified image includes a first code comprising the characters.

[0007] Additionally, the one or more instruction for constructing may include one or more instructions for overlaying at least one of the selected plurality of sub-images onto at least one other of the selected plurality of sub-images.

[0008] Additionally, the selected plurality of sub-images may be segments of the first image, and the one or more instructions for constructing may include one or more instructions for assembling the selected plurality of sub-images to form the first image based on the segmentation of each sub-image.

[0009] Additionally, there may be at least one of the selected plurality of sub-images that has a plurality of corresponding characters.

[0010] Additionally, the sub-images may include sub-images relating to living things, non-living things, and places.

[0011] Additionally, the one or more instructions for constructing may include one or more instructions for overlaying the selected plurality of sub-images to corresponding specific regions on a scenic sub-image.

[0012] Additionally, the plurality of sub-images may include a plurality of characters, and the one or more instruc-

tions for constructing may include one or more instructions for displaying the corresponding sub-images as the plurality of characters are selected.

[0013] According to another aspect, a method may include providing sub-images having a plurality of categories, selecting a plurality of sub-images, constructing a first image that includes the selected plurality of sub-images, displaying the first image in a unified image, comparing information associated with the unified image with information associated with a previously constructed unified image, and providing access to at least one of a device, a service, or a function when the information associated with the unified image matches the information associated with the previously constructed unified image.

[0014] Additionally, the selecting may include selecting the plurality of sub-images having at least two sub-images of a different category.

[0015] Additionally, the method may include providing characters associated with the plurality of sub-images, and where the first image includes the selected plurality of sub-images and associated characters.

[0016] Additionally, the selecting may include selecting the plurality of sub-images based on characters associated with the plurality of sub-images.

[0017] Additionally, the constructing may include overlaying at least one of the selected plurality of sub-images onto another of the selected plurality of sub-images. Additionally, the constructing may include constructing the first image based on the selected plurality of sub-images that correspond to segments of the first image, each segment including a sub-image and at least one character, and each segment having a specific region to occupy in the first image.

[0018] Additionally, the sub-images may include animation and video. Additionally, the constructing may include dragging and dropping the selected sub-image in an unoccupied region of the first image to be constructed.

[0019] According to yet another aspect, a device may include a memory to store instructions, and a processor to execute the instructions to receive selection of a plurality of sub-images and characters, construct a first image that includes the selected plurality of sub-images and a first code that includes the selected characters, compare the received selections with previously stored selections associated with an image and a code, and provide access to a function or a service of the device when the received selections match the previously stored selections.

[0020] Additionally, the device may include a display, and the processor may execute instructions to display the first image and the first code in a unified image on the display.

[0021] According to still another aspect, a device may include means for providing sub-images having different categories, means for receiving selection of a plurality of sub-images, means for constructing a first image that includes the selected plurality of sub-images, means for displaying the first image in a unified image, means for comparing the selections with previously stored selections, and means for providing access to a function or a service of the device when the selections match the previously stored selections.

[0022] According to another aspect, a method may include receiving input associated with constructing a first image, generating the first image based on the input, comparing the first image with a second image, and providing access to at least one of device, a service, or a function when the first image matches the second image.

[0023] Additionally, the method may include generating a first passcode that is associated with the first image, and comparing the first passcode with a second passcode associated with the second image.

[0024] Additionally, the receiving input associated with constructing the first image may include receiving a selection of a base sub-image to facilitate construction of the first image, and the method may further include generating a second sub-image based on the base sub-image.

[0025] Additionally, generating the second sub-image may include receiving drawing input from a user associated with constructing the second sub-image.

[0026] Additionally, the method may include displaying a grid that includes a plurality of coordinates in response to select of the base sub-image.

[0027] Additionally, the providing access may include comparing coordinate information associated with the first image with coordinate information associated with the second image. Additionally, the receiving input associated with constructing the first image may include receiving a selection of a plurality of sub-images and corresponding character associated with each sub-image.

[0028] Additionally, the providing access to at least one of the device, service, or function may include providing access to at least some of applications or functions of a mobile phone.

BRIEF DESCRIPTION OF THE DRAWINGS

[0029] The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate exemplary embodiments described herein and, together with the description, explain these exemplary embodiments. In the drawings:

[0030] FIG. 1 is a diagram illustrating a concept described herein;

[0031] FIG. 2 is a diagram illustrating a front view of exemplary external components of an exemplary device having image-based code capability;

[0032] FIG. 3 is a diagram illustrating a rear view of exemplary external components of the device depicted in FIG. 2;

[0033] FIG. 4 is a diagram illustrating exemplary internal components of the device depicted in FIG. 2;

[0034] FIG. 5 is a diagram of the exemplary image-based code component depicted in FIG. 4;

[0035] FIG. 6 is a flow diagram illustrating exemplary operations for designing an image-based code;

[0036] FIG. 7 is a flow diagram illustrating exemplary operations for providing image-based code to access a device, a service or a function;

[0037] FIG. 8 is a flow diagram illustrating exemplary operations for designing an image to access a device, a service or a function;

[0038] FIG. 9 is a flow diagram illustrating exemplary operations for providing an image to access a device, a service, or a function; and

[0039] FIGS. 10-14 are diagrams illustrating exemplary screenshots for providing an image-based code.

DETAILED DESCRIPTION

[0040] The following detailed description refers to the accompanying drawings. The same reference numbers in different drawings may identify the same or similar elements. Also, the following description does not limit the invention.

The terms image and sub-image, as used herein, are intended to be broadly interpreted to include any representation of graphical information (e.g., a picture, a video, an animation, etc.). The term code, as used herein, is intended to be broadly interpreted to include any character string (e.g., letters, numbers, alphanumeric sequence, symbols, etc.). The term character, as used herein, is intended to be broadly interpreted to include any letter, number, or symbol.

Overview

[0041] Implementations described herein may provide an image-based code. FIG. 1 is a diagram illustrating a concept **100** as described herein. As illustrated, a device may include a display **104** and logic that provides for the design and/or use of an image-based code. For example, an image **102** may be displayed on display **104** of the device. Image **102** may be divided into segments **106**. Each segment **106** may include a sub-image of image **102** and a corresponding sub-character of code **108**. A user may construct image **102** by choosing from an array of segments **106** using a tab **110**. For example, a user may select tab **110** and change the top portion of the image (i.e., a sub-image of a woman's head wearing a baseball cap) to another sub-image (e.g., a man's head wearing a fireman's helmet (not illustrated)). A user may construct a unified image based on the selected segments.

[0042] The user's selection of the predetermined "correct" image **102** and/or code **108** may be used as a basis for security functions for device. For example, the device may require correct selection of image **102** and/or code **108** before the device will allow access to itself.

[0043] As will be described herein, numerous variations to FIG. 1 may be employed. For example, instead of the image being segmented, the image may include a layering of sub-images. Additionally, or alternatively, a sub-image may include animation and/or video. Additionally, or alternatively, a user may select a character of a code to display a sub-image. Still further, additional variations will be described below.

[0044] A sub-image may be of any type, such as living things, non-living things, places, shapes, symbols, etc. However, a user may construct a unique, single image based on a plurality of sub-images that may be more memorable compared to a code. In addition, a user may still be provided with a corresponding code of the image that the user may remember.

Exemplary Device

[0045] FIG. 2 is a diagram illustrating a front view of exemplary external components of an exemplary device having image-based code capability. As illustrated, device **200** may include a housing **205**, a microphone **210**, a speaker **220**, a keypad **230**, function keys **240**, a display **250**, and a camera button **260**. The terms device and component, as used herein, is intended to be broadly interpreted to include hardware, software, and/or a combination of hardware and software.

[0046] Housing **205** may include a structure configured to contain components of device **200**. For example, housing **205** may be formed from plastic and may be configured to support microphone **210**, speaker **220**, keypad **230**, function keys **240**, display **250**, and camera button **260**.

[0047] Microphone **210** may include any component capable of transducing air pressure waves to a corresponding electrical signal. For example, a user may speak into micro-

phone 210 during a telephone call. Speaker 220 may include any component capable of transducing an electrical signal to a corresponding sound wave. For example, a user may listen to music through speaker 220.

[0048] Keypad 230 may include any component capable of providing input to device 200. Keypad 230 may include a standard telephone keypad. Keypad 230 may also include one or more special purpose keys. In one implementation, each key of keypad 230 may be, for example, a pushbutton. A user may utilize keypad 230 for entering information, such as text or a phone number, or activating a special function.

[0049] Function keys 240 may include any component capable of providing input to device 200. Function keys 240 may include a key that permits a user to cause device 200 to perform one or more operations. The functionality associated with a key of function keys 240 may change depending on the mode of device 200. For example, function keys 240 may perform a variety of operations, such as placing a telephone call, playing various media, setting various camera features (e.g., focus, zoom, etc.) or accessing an application. Function keys 240 may include a key that provides a cursor function and a select function. In one implementation, each key of function keys 240 may be, for example, a pushbutton.

[0050] Display 250 may include any component capable of providing visual information. For example, in one implementation, display 250 may be a liquid crystal display (LCD). In another implementation, display 250 may be any one of other display technologies, such as a plasma display panel (PDP), a field emission display (FED), a thin film transistor (TFT) display, etc. Display 250 may be utilized to display, for example, text, image, and/or video information.

[0051] Display 250 may also operate as a view finder, as will be described later. Camera button 260 may be a pushbutton that enables a user to take an image.

[0052] Device 200 is exemplary; device 200 is intended to be broadly interpreted to include any type of electronic device where an image-based code may be utilized. For example, device 200 may include a communication device, such as a wireless telephone or a personal digital assistant (PDA), a computational device, such as a computer, an entertainment device, such as a game system, a stationary device, such as a security system, or any other type of device that includes a display in which an image-based code may be utilized. Accordingly, although FIG. 2 illustrates exemplary external components of device 200, in other implementations, device 200 may contain fewer, different, or additional external components than the external components depicted in FIG. 2. Additionally, or alternatively, one or more external components of device 200 may include the capabilities of one or more other external components of device 200. For example, display 250 may be an input component (e.g., a touch screen). Additionally, or alternatively, the external components may be arranged differently than the external components depicted in FIG. 2.

[0053] In other instances, a user may access a function or service via a network (e.g., the Internet, a private network, a wireless network, a television network, etc.) where an image-based code may be utilized. For example, a user may visit a Web server to gain access to a credit card account, a banking account, an e-mail account, a video rental service account, etc. based on an image-based code. Accordingly, the concept described herein may be applied to various platforms and schemes.

[0054] FIG. 3 is a diagram illustrating a rear view of exemplary external components of the device depicted in FIG. 2. As illustrated, in addition to the components previously described, device 200 may include a camera 370, a lens assembly 372, and a flash 374.

[0055] Camera 370 may include any component capable of capturing an image. Camera 370 may be a digital camera. Display 250 may operate as a view finder when a user of device 200 operates camera 370. Camera 370 may provide for automatic and/or manual adjustment of a camera setting. In one implementation, device 200 may include camera software that is displayable on display 250 to allow a user to adjust a camera setting. For example, a user may be able to adjust a camera setting by operating function keys 240 and/or camera button 260.

[0056] Lens assembly 372 may include any component capable of manipulating light so that an image may be captured. Lens assembly 372 may include a number of optical lens elements. The optical lens elements may be of different shapes (e.g., convex, biconvex, plano-convex, concave, etc.) and different distances of separation. An optical lens element may be made from glass, plastic (e.g., acrylic), or plexiglass. In one implementation, lens assembly 372 may be permanently fixed to camera 370. Lens assembly 372 may provide for a variable aperture size (e.g., adjustable f-number). Flash 374 may include any type of light-emitting component to provide illumination when camera 370 captures an image. For example, flash 374 may be a light-emitting diode (LED) flash (e.g., white LED) or a xenon flash.

[0057] Although FIG. 3 illustrates exemplary external components, in other implementations, device 200 may include fewer, additional, and/or different components than the exemplary external components depicted in FIG. 3. For example, device 200 may not include camera 370 and other components associated therewith. In still other implementations, one or more external components of device 200 may be arranged differently.

[0058] FIG. 4 is a diagram illustrating exemplary internal components of the device depicted in FIGS. 2 and 3. As illustrated, device 200 may include microphone 210, speaker 220, keypad 230, function keys 240, display 250, camera button 260, camera 370, a memory 400, a transceiver 420, and a control unit 430. No further description of microphone 210, speaker 220, keypad 230, function keys 240, display 250, camera button 260, and camera 370 is provided with respect to FIG. 4.

[0059] Memory 400 may include any type of storing component to store data and instructions related to the operation and use of device 200. For example, memory 400 may include a memory component, such as a random access memory (RAM), a read only memory (ROM), and/or a programmable read only memory (PROM). Additionally, memory 400 may include a storage component, such as a magnetic storage component (e.g., a hard drive) or other type of computer-readable medium. Memory 400 may also include an external storing component, such as a Universal Serial Bus (USB) memory stick, a digital camera memory card, and/or a Subscriber Identity Module (SIM) card.

[0060] Memory 400 may include an image-based code component 410. Image-based code component 410 may include instructions to cause device 200 to provide image-based code capability as described herein. Image-based code component 410 will be described in greater detail below.

[0061] Transceiver 420 may include any component capable of transmitting and receiving information. For example, transceiver 420 may include a radio circuit that provides wireless communication with a network or another device.

[0062] Control unit 430 may include any logic that may interpret and execute instructions, and may control the overall operation of device 200. Logic, as used herein, may include hardware, software, and/or a combination of hardware and software. Control unit 430 may include, for example, a general-purpose processor, a microprocessor, a data processor, a co-processor, and/or a network processor. Control unit 430 may access instructions from memory 400, from other components of device 200, and/or from a source external to device 200 (e.g., a network or another device).

[0063] Control unit 430 may provide for different operational modes associated with device 200. Additionally, control unit 430 may operate in multiple modes simultaneously. For example, control unit 430 may operate in a camera mode, a walkman mode (e.g., a music playing mode), and/or a telephone mode. In one implementation, a user may prevent access of device 200 by employing an image-based code. The image-based capability of device 200 will be described in greater detail below.

[0064] Although FIG. 4 illustrates exemplary internal components, in other implementations, device 200 may include fewer, additional, and/or different components than the exemplary internal components depicted in FIG. 4. For example, in one implementation, device 200 may not include transceiver 420. In still other implementations, one or more internal components of device 200 may include the capabilities of one or more other components of device 200. For example, transceiver 420 and/or control unit 430 may include their own on-board memory 400.

[0065] FIG. 5 is a diagram of the exemplary image-based code component depicted in FIG. 4. Image-based code component 410 may include an image store 510, an image arranger 520, and/or an image/code comparer 530. Although not illustrated, image-based code component 410 may include a graphical user interface (GUI). The GUI may include various graphical interfaces, such as icons, menus, tabs, drag-and-drop interface, etc. to permit the design and selection of an image/code pair and/or an image.

[0066] Image store 510 may allow a user to display various sub-images, such as living objects (e.g., people, animals, plants, etc.) or non-living objects (e.g., places, things, shapes, symbols, etc.) and characters.

[0067] Each of the sub-images may co-exist with one or more characters. In one implementation, a sub-image/character pair may be fixed. For example, as illustrated in FIG. 1, one of the sub-image/character pairs is a woman's head with a baseball cap (sub-image) and the number five (character). In this instance, image store 510 may not allow, for example, the number five to be changed to a different number because the sub-image/character pair (i.e., the woman's head with a baseball cap and the number five) is a fixed sub-image character/pair. In other implementations, however, image store 510 may provide for the customization of sub-image/character pairs. For example, image store 510 may provide a GUI to change, for example, one sub-image/character pair to a new and different sub-image/character pair. In this way, a user may design a sub-image/character pair that includes, perhaps, a favorite sub-image with a favorite character (e.g., lucky number). In still other implementations, image store 510 may

provide a GUI where sub-images may not co-exist with one or more characters. For example, a user may create an image without a corresponding code.

[0068] Image store 510 may also allow a user to import sub-images. For example, the GUI may import a sub-image from a resource external to device 200, such as the Internet, or from a resource internal to device 200, such as memory 400. For example, an image captured by camera 370 and stored in memory 400 may be added to the image store 510. Additionally, or alternatively, image store 510 may provide a GUI to import new characters. For example, a user may import unique symbols (e.g., Chinese characters or abstract symbols) to be associated with a sub-image.

[0069] Image store 510 may also allow a user to create a sub-image and/or a character. For example, image store 510 may provide tools (e.g., drawing tools, painting tools, etc.) to create a sub-image and/or a character. Image store 510 may also provide a GUI to manage the size, shape and/or orientation of an imported sub-image so that the sub-image may be utilized to form an image. In other instances, a user may create a sub-image and/or an image utilizing, for example, a stylus, his or her finger, keys on keypad 230, a joystick, touchpad, etc.

[0070] Image arranger 520 may allow the user to construct an image and a code based on the sub-images and the characters from image store 510. That is, a user may construct a unified image based on the sub-images. Described below are two exemplary implementations that may be employed to construct an image and a code; however, other implementations may be realized. In a first implementation, an image/code pair may include a plurality of segments, where each segment includes at least a sub-image and a corresponding character. The GUI of image arranger 520 may include, for example, tabs, such as tabs 110, to select a segment, or may include a drag-and-drop interface to select a segment.

[0071] The segmentation of the image/code pair may or may not be uniform. For example, each segment of the image/code pair may or may not be of similar size, shape and/or orientation.

[0072] Additionally, or alternatively, each segment of the image/code pair may or may not differ in contribution to the overall image and/or code. That is, for an image/code pair, one segment may include a sub-image that contributes up to fifty percent of the image, and may contribute, for example, three characters of a six character code. Conversely, one segment may include a sub-image that contributes up to ten percent of the image, and may contribute, for example, one character of a five character code.

[0073] Additionally, or alternatively, the image/code pair may not be completely segmented. For example, an image/code pair may include an initial, static sub-image with no corresponding character, and the remaining portion of the image and the code may be constructed based on segments having a sub-image and a character.

[0074] In a second implementation, an image/code pair may be configured as a layering of sub-images having co-existing characters. For example, image-based code component 410 may include a GUI that provides selection of a base sub-image, such as a scenic sub-image (e.g., a jungle, outer space, a room in a house, underwater in an ocean, etc.) and various sub-images to overlay (e.g., by drag-and-drop) on the scenic sub-image. The selection of overlay sub-images may range from an overlay sub-image (e.g., a space ship or an exotic animal) that relates to the scenic sub-image (e.g., outer

space or jungle), to an overlay sub-image (e.g., baseball bat or washing machine) that is relatively unrelated to the scenic sub-image (e.g., outer space or jungle). In one implementation, the overlay sub-images may be dispersed in various regions of the base sub-image. Additionally, or alternatively, a base sub-image (e.g., an abstract image, such as a colored circle) may provide for overlaying multiple overlay sub-images (e.g., concentric circles of different colors) in one region.

[0075] In one implementation, a region for an overlay sub-image may be fixed. For example, a base sub-image may include specific regions that accept an overlay sub-image. In one implementation, for example, as a specific region is occupied, the next region to accept an overlay sub-image may be highlighted. In one implementation, the specific regions to be occupied may have a particular order, while in other implementations the specific regions to be occupied may not have a particular order.

[0076] Additionally, or alternatively, a base sub-image may not include specific regions to be occupied. That is, an overlay sub-image may be placed anywhere on the base sub-image. In one implementation, the order in which the overlay sub-image is placed with respect to the base sub-image may change the order of the corresponding code. For example, a scenic sub-image (e.g., a jungle “X”) may include three animals (e.g., a lion “5”, a giraffe “4”, and a snake “H”).

[0077] When the order of overlaying the overlay sub-images is lion, snake, and giraffe, the code may be “X5H4”; however when the order of placing the overlay sub-images is snake, lion, and giraffe, the code may be “XH54”. Thus, although the image may be same (i.e., the image contains the same animals with corresponding characters and may have been placed in the same regions), the code may be different.

[0078] According to another implementation, as previously described, image-based code component 410 may provide a selection of a base sub-image. However, instead of a user selecting various sub-images to overlay (e.g., by drag-and-drop), a user may create an image by, for example, drawing the image on display 250 utilizing a stylus, tools of image store 510, or some other input mechanism (e.g., a joystick, a touch pad, etc.). In one implementation, a user may create and/or select a code to correspond to the image. Additionally, or alternatively, image-based code component 410 may automatically generate a code to correspond to the image. Additionally, or alternatively, a user may create an image without having a corresponding code. That is, a user may only utilize the image, for example, to gain access to device 200, by drawing the image on display 250.

[0079] Image/Code comparer 530 may include a component to compare one image-based code to another image-based code. For example, image/code comparer 530 may compare an image-based code previously stored in memory 400 to an image-based code entered by a user when trying to use device 200. Image/Code comparer 530 may compare an image, a sub-image, a character, a code and/or information (e.g., identifiers or coordinates) associated therewith of the image-based code stored in memory 400 with an image, a sub-image, a character, a code and/or information (e.g., identifiers or coordinates) associated therewith of the image-based code entered.

[0080] In one implementation, image/code comparer 530 may provide, for example, an indication (e.g., a visual or an auditory cue) corresponding to the result of the comparison. Additionally, or alternatively, for example, image/code com-

parer 530 may not provide any indication of the result; rather, device 200 will permit access or deny access depending on the result of the comparison.

[0081] Although FIG. 5 illustrates exemplary components to provide the image-based code capability as described herein, in other implementations, device 200 may include fewer, different, or additional components than the exemplary components depicted in FIG. 5. As described above, image-based code component 410 may provide various functional capabilities for designing and employing an image-based code with, for example, device 200. However, in other applications, for example, a Web server employing image-based code component 410 for accessing a bank account, some of the functions described above may or may not be provided. For example, in such an instance, image-based code component 410 may not provide for importing sub-images and/or characters, or altering a sub-image/character pair.

[0082] FIG. 6 is a flow diagram illustrating exemplary operations for designing an image-based code. Process 600 may begin with a selection of a sub-image having a corresponding character, or a selection of a character having a corresponding sub-image (Block 610). Image-based code component 410 may provide a GUI on display 250. The GUI may provide for the selection of a sub-image and a corresponding character. In other instances, a character may be entered from keypad 230 and the corresponding sub-image may be displayed on display 250. As mentioned above, the GUI of image-based code component 410 may include, for example, various menus, icons, tabs, drag-and-drop interface, etc. to permit selection of a sub-image and/or character.

[0083] In Block 620, image-based code component 410 may determine whether an image/code pair is created. For example, image-based code component 410 may determine whether the number of sub-images or characters is insufficient or whether certain sub-image regions are unoccupied. If an image/code pair is not created (Block 620—NO), then additional selections may be needed to complete an image/code pair. When an image/code pair is created (Block 620—YES), the image/code pair may be stored in memory 400 (Block 630).

[0084] FIG. 7 is a flow diagram illustrating exemplary operations for providing image-based code access to a device, such as device 200. Process 700 may begin with a selection of a sub-image having a corresponding character, or a selection of a character having a corresponding sub-image (Block 710). Image-based code component 410 may provide a GUI on display 250. The GUI may provide for the selection of a sub-image and a corresponding character. In other instances, a character may be entered from keypad 230 and the corresponding sub-image may be displayed on display 250. As mentioned above, the GUI of image-based code component 410 may include, for example, various menus, icons, tabs, drag-and-drop interface, etc., to permit selection of a sub-image and/or a character.

[0085] In Block 720, image-based code component 410 may determine whether an image/code pair is created. For example, image-based code component 410 may determine whether the number of sub-images or characters is insufficient, or whether certain sub-image regions are unoccupied. If an image/code pair is not created (Block 720—NO), then additional selections may be needed to complete an image/code pair. When an image/code pair is created (Block 720—YES), image-based code component 410 may compare the entered image/code pair with another image/code pair to

determine whether a match exists (Block 730). For example, image-based code component 410 may make a comparison with an image/code pair stored in memory 400. When the comparison is successful (Block 730—YES), access to device 200 may be granted (Block 740); however, when the comparison is not successful (Block 730—NO), access to device 200 may be denied (Block 750).

[0086] FIG. 8 is a flow diagram illustrating exemplary operations for designing an image-based code. Process 800 may begin with creating an image having a corresponding code, or creating an image without a corresponding code (Block 810). In such an instance, image-based code component 410 may provide a GUI on display 250. The GUI may include various menus, icons, tabs, and/or drawing tools, etc. to create an image and include a region on display 250 to create one or more images. Additionally, or alternatively, a user may create an image on display 250 utilizing his/her finger, a stylus, one or more keys of keypad 230, a joystick, a touchpad, etc.

[0087] In each instance, a user may create an image without selecting each sub-image to create the image. For example, a user may create an image merely by using his or her finger, a stylus, one or more keys, a joystick, a touchpad, etc. Additionally, or alternatively, a user may select a base sub-image (e.g., a grid) to draw on and to serve as a guide to create an image. As described above, an image may be created with or without a corresponding code.

[0088] In one implementation, a code corresponding to an image may be selected by a user. Additionally, or alternatively, a code may be automatically generated by image-based code component 410.

[0089] In Block 820, image-based code component 410 may determine whether the image with the corresponding code is created or the image without the corresponding code is created.

[0090] For example, image-based code component 410 may determine whether the image with or without the corresponding code is created based on whether a user enters the image or code (e.g., by pressing an Enter key, etc.). If the image with or without the corresponding code is not created (Block 820—NO), then additional information (i.e., image creation or code creation) may be needed to complete the image with or without the corresponding code. When the image with or without the corresponding code is created (Block 820—YES), the image with or without the corresponding code may be stored in memory 400 (Block 830). In one implementation, the stored image may include, among other things, coordinate information corresponding to the base sub-image (e.g., a grid) and/or drawing data.

[0091] FIG. 9 is a flow diagram illustrating exemplary operations for providing image-based code access to a device, such as device 200. Process 900 may begin with creating an image having a corresponding code, or creating an image without a corresponding code (Block 910). Image-based code component 410 may provide a GUI on display 250. The GUI may include various menus, icons, tabs, and/or drawing tools, etc. to create an image and include a region on display 250 to create an image. Additionally, or alternatively, a user may create an image on display 250 utilizing his/her finger, a stylus, one or more keys of keypad 230, a joystick, a touchpad, etc. In each instance, a user may create an image without selecting each sub-image to create the image, as described above with respect to FIG. 8. Additionally, or alternatively, a

user may select a base sub-image (e.g., a grid) to draw on and to serve as a guide to create an image.

[0092] As described above, an image may be created with or without a corresponding code. In one implementation, a code corresponding to an image may be selected by a user. Additionally, or alternatively, a code may be automatically generated by image-based code component 410.

[0093] In Block 920, image-based code component 410 may determine whether the image with the corresponding code is created or the image without the corresponding code is created. For example, image-based code component 410 may determine whether the image with or without the corresponding code is created based on whether a user enters the image or code (e.g., by pressing an Enter key, etc.). If the image with or without the corresponding code is not created (Block 920—NO), then additional information (i.e., image creation or code creation) may be needed to complete the image with or without the corresponding code. When the image with or without the corresponding code is created (Block 920—YES), image-based code component 410 may compare the entered image with or without the corresponding code to another image with or without a corresponding code to determine whether a match exists (Block 930). For example, image-based code component 410 may make a comparison with image information and/or code information stored in memory 400. When the comparison is successful (Block 930—YES), access to device 200 may be granted (Block 940); however, when the comparison is not successful (Block 930—NO), access to device 200 may be denied (Block 950).

[0094] FIGS. 6-9 illustrate exemplary operations for permitting access to device 200; however, as previously mentioned, in other instances, a user may access a function or service via a network (e.g., the Internet, a public switched telephone network, a private network, a wireless network, a television network, etc.) where an image-based code may be utilized. For example, a user may visit a Web server to gain access to a credit card account, a banking account, an e-mail account, a video rental service account, etc. based on an image-based code. Accordingly, the concept described herein may be applied to various platforms and schemes. Additionally, it will be appreciated that a user may enter only the code (when a code is created) to access device 200 in correspondence to the flow diagrams of FIGS. 6-9, or use a code one access is granted to device 200 based on an image-based code. For example, a two-level code may be employed for access to some functions. For example, the image-based code may be used to access certain applications and/or functions (e.g., placing a call), and a second code (e.g., an alphanumeric code) may be used to access other applications and/or functions (e.g., modifying a phonebook contact list).

EXAMPLES

[0095] FIGS. 10-14 are diagrams illustrating exemplary screenshots for providing an image-based code. The description below omits discussion relating to a user's selection of, for example, various menus, prompts, and/or graphical links to arrive at the screenshots depicted in FIGS. 10-14. As illustrated, each screenshot is displayed on display 250 of device 200.

[0096] FIG. 10 is a diagram illustrating an image/code pair based on layering of sub-images. In this example, the GUI of image-based code component 410 includes a code region 1010, an image region 1020, and a sub-image selector region

1030. Code region **1010** indicates the corresponding characters of the sub-images in image region **1020**. Sub-image selector region **1030** provides a GUI component to select various sub-images.

[0097] For example, image region **1020** includes a base sub-image, such as a jungle, with a corresponding character of π (pi). Various sub-images may be placed on the scenic sub-image using, for example, drag-and-drop. For example, a user may drag the sub-image of the monkey onto the sub-image of the jungle, and the letter Z may appear in the code region **1010**. Next, the user may drag the sub-image of the giraffe onto the sub-image of the jungle, and the number four may appear in the code region **1010**. Subsequently, the user may drag the lion and the snake to image region **1020** and the corresponding letters C and E may appear in the code region **1010**. In this example, the scenic sub-image does not include any specific regions where an overlay sub-image may be placed. However, in other implementations, a base image may include specific regions where a sub-image may be placed.

[0098] FIG. **11** is a diagram illustrating a partial image/code pair based on segmentation. In this example, the GUI of image-based code component **410** includes a code region **1110**, an image region **1120**, and a sub-image selector region **1130**. As illustrated, code region **1110** include three characters (i.e., 4, A, and Z) corresponding to the three strokes in image region **1120**, and three unoccupied character regions numbered four and five. Image region **1120** includes two straight lines and one curved line corresponding to the three characters, and two unoccupied segment regions numbered four and five corresponding to unoccupied character regions numbered four and five. The size and orientation of each unoccupied segment region is different, and each unoccupied segment region is highlighted. In this example, unoccupied segment region four may have two corresponding characters in code region **1110**, while unoccupied segment region five has one corresponding character in code region **1110**.

[0099] Sub-image selector region **1130** includes a pull-down menu to select the type of image to be constructed. In this example, the image to be constructed is of a symbol. So, for example, a user may select the category of the image to be designed and/or entered by using the pull-down menu in sub-image selector region **1130**. Additionally, a pull-down menu indicates the next segment to be occupied. Sub-image selector region **1130** may automatically provide sub-images that correspond with the size, shape, and orientation of, for example, the next unoccupied segment. So, for example, when a user selects symbol as a category of the image to be designed and/or entered, the pull-down menu may indicate segment one, and sub-image selector region **1130** may automatically provide sub-images that correspond with the first segment to be occupied. A user may select a sub-image and drag the sub-image into image region **1120** and the corresponding code, such as the number 4 may appear in code region **1110**.

[0100] As illustrated in FIG. **11**, a user may have already selected sub-image/character pairs for segments one, two and three (i.e., the two straight lines and the one curved line in image region **1120**). Segment four appears in the pull-down menu of sub-image selector region **1130**, and unoccupied segment four may be highlighted as the next unoccupied segment in image region **1120**. However, any segment may be selected using the pull-down menu. For example, if a user selects an incorrect sub-image/character pair and/or wishes to

change the sub-image/character pair, the user may re-select a different sub-image/character pair.

[0101] FIG. **12** is a diagram illustrating a partial image/code pair based on segmentation and layering. In this example, the GUI of image-based code component **410** includes a code region **1210**, an image region **1220**, and a sub-image selector region **1230**. In this example, image region **1220** includes a static sub-image (i.e., a female head) without a corresponding character. Sub-image selector region **1230** provides a pull-down menu indicating that a female person is selected. A user may select from various categories of images to be designed and/or entered. Sub-image selector region **1230** also indicates the first of the sub-images to be used. For example, sub-image selector region **1230** indicates that sub-images of eyes, which are animated, may be selected. The sub-image of eyes may be placed on the face portion of the female person. That is, the sub-image of eyes is an overlay sub-image. In one implementation, when eyes are selected, the next type of sub-image may be provided, such as different animations of a torso (e.g., twisting, using a hula hoop, etc.). In this instance, the sub-image of the torso may be a segment sub-image. So, for example, when a sub-image of eyes is selected, the pull-down menu may indicate that sub-images of a torso may be selected.

[0102] In this example, the image/code pair may include a combination of overlay sub-images and segment sub-images. In other instances, the image/code pair may include only overlay sub-images or only segment sub-images.

[0103] FIG. **13** is a diagram illustrating a user-created image. In this example, the GUI of image-based code component **410** may include a code region **1310**, an image region **1320**, and/or a menu region **1330**. In this example, image region **1320** may allow a user to create an image. For example, if display **250** is a touch screen, a user may utilize his/her finger, a stylus, or some other instrument to create an image, such as the T-shaped image illustrated in image region **1320**. As illustrated in menu region **1330**, a user may select a grid to use as a guide to create an image. In this way, a user may create an image simply by connecting the dots.

[0104] Although, the T-shaped image is illustrated in FIG. **13**, the types of images, symbols, abstract shapes, etc., may be created via the grid or without the grid. Menu region **1330** may provide a selection of a grid size (e.g., 2x2 or 4x4). However, in other instances, a user may desire to select a larger grid (e.g., a 5x5 or a 6x6). In such instances, a larger grid may allow the user to create a more complex image, which may translate into a greater level of security. For example, as illustrated in FIG. **14**, a more complex image may be created using an 8x8 grid (e.g., a smiley face). Additionally, or alternatively, other grid patterns may be utilized to provide a greater complexity and/or sophistication of the image. In this regard, it will be appreciated that different images (e.g., in terms of complexity) may be used to provide access to a device, a function, and/or a services. For example, to access credit card information may require a more complex image-based code compared to other information that may be deemed less sensitive.

[0105] Additionally, or alternatively, menu region **1310** may allow a user to select whether a code is to be created with the image. In the examples of FIGS. **13** and **14**, a user may select that no code is to be created. Thus, in such an instance, a user may create the image without creating a corresponding code. Additionally, or alternatively, a user may select that a code is created. Since the user is creating the image, without,

for example, a drag-and-drop of sub-images, a user may type in a code after the image is created. In other instances, image-based code component **410** may generate, for example, a code of random characters, and display the characters in code region **1310**. Additionally, or alternatively, similar to that previously described, an order of strokes by which the image is created by a user may determine the code and/or whether the image is correct. That is, a user may create the same image multiple ways; however, if the order of the strokes is used by image-based code component **410** when determining whether to allow access to device **200**, only one image may allow a user to access device **200**. In each case, the user may know whether the order of strokes is being used by image-based code component **410** to allow access to device **200**. In one implementation, the order of strokes may be based on coordinates corresponding to the dots of the grid. In other words, an order by which a user draws, for example, some symbol, shape, etc., using a series of dots may allow image-based code component **410** to determine an order of the strokes by which an image is created. Additionally, or alternatively, for example, image/code comparer **530** may utilize coordinate information corresponding to the dots of the grid to determine whether an image matches a pre-stored image.

[0106] Although not specifically mentioned with respect to FIGS. **10-14**, it will be appreciated that a user may enter a letter and/or a number in code regions **1010**, **1110**, **1210** or **1310** without selecting a sub-image or creating an image. In such an instance, when a user enters a character in a code region the corresponding sub-image may be provided in image regions **1020**, **1120**, **1220**, or **1320**.

[0107] Additionally, or alternatively, it will be appreciated with respect to FIGS. **10-14** that selections from menus, etc., as described above, may be implemented as user preferences.

CONCLUSION

[0108] The foregoing description of implementations provides illustration, but is not intended to be exhaustive or to limit the implementations to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practice of the teachings. For example, while it has been described that different images could be utilized based on different levels of security, other criteria may be utilized. For example, different days of the week may be a basis for providing a different image. In this regard, for example, an image of someone playing soccer could be the image used to provide access to device **200** during the weekend, and the image of the smiley face could be the image used during the weekdays. Thus, a user may customize different settings associated with an image-based code to provide added security, entertainment, etc.

[0109] It should be emphasized that the term “comprises” or “comprising” when used in the specification is taken to specify the presence of stated features, integers, steps, or components but does not preclude the presence or addition of one or more other features, integers, steps, components, or groups thereof.

[0110] In addition, while a series of processes and/or acts have been described herein, the order of the processes and/or acts may be modified in other implementations. Further, non-dependent processes and/or acts may be performed in parallel.

[0111] It will be apparent that aspects described herein may be implemented in many different forms of software, firmware, and hardware in the implementations illustrated in the

figures. The actual software code or specialized control hardware used to implement aspects does not limit the invention. Thus, the operation and behavior of the aspects were described without reference to the specific software code—it being understood that software and control hardware can be designed to implement the aspects based on the description herein. No element, act, or instruction used in the present application should be construed as critical or essential to the implementations described herein unless explicitly described as such. Also, as used herein, the article “a” and “an” are intended to include one or more items. Where only one item is intended, the term “one” or similar language is used. Further, the phrase “based on” is intended to mean “based, at least in part, on” unless explicitly stated otherwise. As used herein, the term “and/or” includes any and all combinations of one or more of the associated list items.

1.-30. (canceled)

31. A method comprising:

providing, using a device, a base image comprising image items, the image items being associated with coordinate information;

receiving, using the device, selection of a first plurality of the image items, the first plurality of the image items corresponding to first coordinate information;

creating, using the device, a first image based on the received selection of the first plurality of the image items;

comparing, using the device, first information, associated with the created first image, with second information associated with a second image,

where the comparing comprises at least one of:

comparing the first coordinate information, associated with the first image, with second coordinate information associated with the second image, or

comparing first code information associated with the created first image with second code information associated with the second image; and

providing, using the device, access to the device based on a result of comparing the first information, associated with the created first image, with the second information, associated with the second image.

32. The method of claim **31**, further comprising:

receiving selection of a second plurality of the image items, the second plurality of the image items corresponding to the second coordinate information;

creating the second image based on the received selection of the second plurality of the image items; and

storing the second information associated with the created second image, the second information including the second coordinate information,

where the comparing comprises:

comparing the first information, associated with the created first image, with the stored second information of the created second image.

33. The method of claim **31**, further comprising:

generating the first code information based on the received selection of the first plurality of the image items; and generating the second code information based on a selected second plurality of the image items,

where comparing the first code information with the second code information includes:

comparing the generated first code information with the generated second code information.

34. The method of claim **31**, where creating the first image comprises creating the second image based on a first order of the selection of the first plurality of the image items, and where the second image is created based on a second order of selection of a second plurality of the image items, the method further comprising:
providing access to the device based on the first order, of the selection of the first plurality of the image items, and the second order, of the selection of the second plurality of the image items.

35. The method of claim **34**, where the first plurality of the image items corresponds to the second plurality of the image items, and

where the first order, of the selection of the first plurality of the image items, differs from the second order, of the selection of the second plurality of the image items, the method further comprising:

denying access to the device based on the first order, of the selection of the first plurality of the image items, being different than the second order, of the selection of the first plurality of the image items.

36. The method of claim **31**, where providing access to the device comprises at least one of:

providing access to one or more functions of the device;
providing access to information provided by the device; or
providing access to one or more services associated with the device.

37. The method of claim **31**, where the second image corresponds to a first level of complexity, the first level of complexity being associated with a first level of security, and where providing access to the device comprises providing access to a first type of information including confidential information,

the method further comprising:

creating a third image corresponding to a second, different level of complexity, the second level of complexity being associated with a second, different level of security; and

providing, based on the created third image, access to a second, different type of information excluding the confidential information.

38. The method of claim **37**, where a first size of the base image corresponds to the first level of complexity, the method further comprising:

receiving selection of a second base image having a second, different size, the second size corresponding to the second level of complexity; and

creating the third image using the selected second base image.

39. A system comprising:

a device to:

receive selection of a base image including image items;
receive selection of a first plurality of the image items, the first plurality of the image items corresponding to first coordinate information;

create a first image based on the received selection of the first plurality of the image items;

compare first information of the created first image with stored second information of a second image,

where, when comparing, the device is to at least one of:
compare the first coordinate information with second coordinate information included in the stored second information of the second image, or

compare first code information, associated with the created first image, with second code information included in the stored second information of the second image; and

provide access to the device based on comparing the first information, of the created first image, with the stored second information, of the second image.

40. The system of claim **39**, where, when providing access, the device is to at least one of:

provide access to information provided by the device;

provide access to at least one service associated with the device; or

provide access to at least one function of the device.

41. The system of claim **39**, where, when providing access, the device is to:

provide access to a first function of the device;

receive the first code information; and

provide, based on the received first code information, access to a second function of the device, the second function being associated with the first function.

42. The system of claim **39**, where the second image corresponds to a first level of security, and

where, when providing access, the device is to provide access to a first type of information corresponding to the first level of security,

the device further to:

create a third image corresponding to a second, different level of security; and

provide, based on the created third image, access to a second, different type of information, the second type of information including information that is more sensitive than information included in the first type of information.

43. The system of claim **42**, where a quantity of the image items of the base image is associated with the first level of security,

the device further to:

receive selection of a second base image including a second, different quantity of image items, the second quantity of image items being associated with the second level of security; and

create the third image using the second base image.

44. The system of claim **42**, where the second quantity of image items included the second base image is greater than the quantity of the quantity of the image items included in the base image.

45. The system of claim **39**, where, when creating the first image, the device is to create the second image based on a first order of the selection of the first plurality of the image items, the second image being created, based on a second order of selection of a second plurality of the image items, prior to the first image,

the device further to:

generate the first code information based on the first order of the selection of the first plurality of the image items;

generate the second code information based on the second order of the selection of the second plurality of the image items; and

providing access to the device based on comparing the generated first code information with the generated second code information.

46. The system of claim **39**, where the first order of the selection of the first plurality of the image items is different than the second order of the selection of the second plurality

of the image items, and where the generated first code information is different than the generated second code information,

the device further to:

deny access to the device based on the generated first code information being different than the generated second code information.

47. A computer-readable memory device incorporating one or more instructions, executable by at least one processor, to perform a method comprising:

creating a first image based on a received user input;

comparing first information of the created first image with stored second information of a stored second image, where the comparing comprises at least one of:

comparing first coordinate information of the created first image with second coordinate information included in the stored second information of the stored second image, or

comparing first code information generated based on the received user input with second code information included in the stored second information of the stored second image; and

providing access to the device based on a result of comparing the first information, of the created first image, with the stored second information, of the stored second image.

48. The computer-readable memory device of claim **47**, where the stored second image corresponds to a first level of security, and

where, when providing access, the device is to provide access to a first type of information,

the one or more instructions to perform a method further comprising:

creating a third image corresponding to a second, different level of security; and

providing, based on the created third image, access to a second, different type of information, the second type of information including information that is less sensitive than information included in the first type of information.

49. The computer-readable memory device of claim **47**, the one or more instructions to perform a method further comprising:

receiving selection of a fourth image including a first quantity of image items, the first quantity of image items corresponding to the first level of security;

creating the second image based on a selection of one or more image items of the first quantity of image items;

receiving selection of a fifth image including a second, different quantity of image items, the second quantity of image items corresponding to the second level of security; and

creating the third image based on a selection of one or more image items of the second quantity of image items.

50. The computer-readable memory device of claim **47**, where creating the first image comprises creating the second image based on receiving selection of a first plurality of image items in a first order,

where creating the second image comprises receiving selection of the first plurality of image items in a second, different order,

the one or more instructions to perform a method further comprising:

denying access to the device based on the first order being different than the second order.

* * * * *