



(43) International Publication Date
13 September 2012 (13.09.2012)

(51) International Patent Classification:

G06F 3/048 (2006.01) G06F 3/033 (2006.01)
G06F 3/01 (2006.01) G06F 3/038 (2006.01)

(21) International Application Number:

PCT/JP2012/056218

(22) International Filing Date:

6 March 2012 (06.03.2012)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

1103833.8 7 March 2011 (07.03.2011) GB

(71) Applicant (for all designated States except US): **SHARP KABUSHIKI KAISHA** [JP/JP]; 22-22, Nagaike-cho, Abeno-ku, Osaka-shi, Osaka, 5458522 (JP).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **KAY, Andrew. JUVONEN, Matti Pentti Taavetti. BROWN, Christopher James.**

(74) Agent: **HARAKENZO WORLD PATENT & TRADE-MARK**; Daiwa Minamimorimachi Building, 2-6, Tenjinbashi 2-chome Kita, Kita-ku, Osaka-shi, Osaka, 5300041 (JP).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) Title: A USER INTERFACE, A DEVICE INCORPORATING THE SAME AND A METHOD FOR PROVIDING A USER INTERFACE

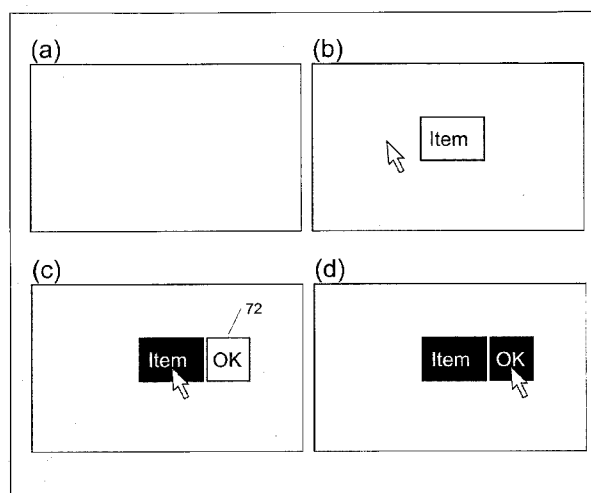


Figure 7

(57) Abstract: A user interface for a device, the device having a display or being networked with a device having a display, the interface comprising: a tracking system capable of locating a predetermined part of a user, in space; an object tracking system capable of locating an object in space when the object is pointing at the display; a unit for calculating a cursor position at which a user is pointing, using information from the tracking system and the object tracking system; a GUI control unit to create a graphical user interface (GUI) using information from the cursor position, to overlay the GUI and the cursor on the display, and to control the GUI; wherein the GUI control unit is designed to initialise when the object points at a predetermined start target; and wherein the GUI control unit responds to the cursor passing over an element of the GUI.

DESCRIPTION

TITLE OF INVENTION:

A USER INTERFACE,

5

A DEVICE INCORPORATING THE SAME AND

A METHOD FOR PROVIDING A USER INTERFACE

TECHNICAL FIELD

10

The present invention relates to a user interface for controlling a device. In addition, the invention relates to a device incorporating such a method and to a method of providing a user interface. It may relate in particular to a TV set, a set-top box, a PVR, DVD or Blu-ray player, radio, hi-fi, multimedia player, internet multimedia device or home network controller.

15

BACKGROUND ART

The television remote control has not changed significantly since its invention. Meanwhile, televisions and other display devices have acquired new functionality.

20

A recent trend has been to introduce alternative methods of controlling a television. Such methods can be used either in addition to, or instead of, the traditional remote control. One reason for the trend is the perceived need to provide a user experience that is both easier to use and richer, that is, affording a greater amount of control than previous methods.

25

Compared to many other devices such as computers and

mobile phones, TV user interfaces are often quite crude. Richer user interfaces are required due to the changing role of the television as a home 'media hub' providing functions such as web browsing and interactive television.

5 Although gesture interfaces have been studied for decades, mainstream commercial interest in such interfaces is relatively new. In the absence of a standardised set of gestures, different companies have implemented a variety of interfaces with custom gestures for interacting with the
10 device and a custom set of graphical objects, or 'widgets', displayed on screen. These systems have a steep learning curve as the user must be taught a set of gestures specific to each implementation.

 All gesture interfaces require some method of
15 recognising the user's hand position and reacting to it, but how this is done can vary considerably between implementations. These systems aim to try to find the compromise between low computational complexity and a sufficiently rich set of gestures. A high-end computer vision
20 system may be able to analyse images and provide a real-time three-dimensional model of the position of the user's arm, hand and fingers. Such a high-end system can recognise a large palette of subtle hand gestures and react to them, but it may be very expensive. A simpler gesture-based system that is
25 able only to track the position of the user's hand can

distinguish only a few large gestures, which may limit the way in which the user interacts with the system.

Commercial gesture tracking systems have predominantly been of the latter, cheaper variety. A typical set of gestures that these systems can recognise includes moving the hand to left, right, up or down to navigate through menus and occasionally a separate gesture to browse through lists of items such as TV channels or music albums. A gesture-based system will also need a way to bring up the interface. For this purpose, such an interface may require yet another gesture, such as waving at the system.

In many systems, the behaviour of the user determines a virtual cursor position on (or around) the display (which we abbreviate to simply cursor). The cursor may indicate a region as precise as a pixel (as with a conventional computer mouse pointer), or a whole (virtual) object (such as a highlighted text character, virtual button, text box or picture.) In any case, the cursor indicates on the display a position or object that the user is interacting with, for example to select, move or modify.

A common problem for simple gesture-based systems is that of selection: in a system that can track a two-dimensional point, how does one implement the very common action of selecting an item? Two common solutions exist. The first is trying to recognise a special select gesture. This can

mean moving the hand towards the camera, or simulating some kind of a mouse click type gesture, perhaps with the index finger. Both options are hard to recognise, particularly with just one camera, and may be hard to perform accurately.

5 The second common select action is that of dwelling, that is, holding the cursor still on the selection for a predetermined length of time. Holding the cursor still for long enough may require the user to exercise fine motor control, and may therefore be difficult. It may also be frustrating for the user
10 as it introduces an enforced a delay into to the system.

The publicly available website <http://www.dontclick.it> interactively introduces a computer interface system which allows the user to interact with graphical user interface elements by moving a cursor, without the need to click. The
15 website introduces a number of different user interface elements. The system is designed for computer user interface. Making selections in the system requires a fine control of the cursor position, using a computer pointing device such as a mouse.

20 US patent publication 20080123937 A1 describes a system for controlling a user interface. In order to facilitate selection, the system can alternate between 'point' and 'click' modes.

25 US patent number 5,594,469 (expired) describes a system for control using an open hand, described therein as

'the "how" position,' to enter the control mode, and to select and manipulate user interface elements by moving a cursor. In the system, the user selects an item by dwelling, that is, keeping the cursor on top of it for a predetermined length of time. The user moves the cursor by gesturing in the corresponding direction. The system uses a separate 'exit' gesture to exit the control mode. The system must be calibrated for the hand position.

US patent publication 20100277412 A1 describes a camera-based system for tracking objects, including optical methods to determine a finger position in the proximity of a mobile device, for the purpose of controlling that device.

US patent publication 20100079374 A1 describes a tracking system where the user holds a pointing device incorporating a camera. The position of the pointing device is calculated from the camera image. In this system, the user must always hold the special tracking device.

US patent publication 20060098873 A1 describes a system where an object is tracked using two cameras. The object to be tracked is segmented from the background by subtracting the background from each image.

CN 1904806 proposes a "position input system" for use with a computer, to replace the current mouse or touch-sensitive screen. The proposed position input system has a pair of CCD cameras which determine the mid-point of the

user's eyes and the position of the user's finger, and these are used to estimate a target location at which the user's finger points. The position input system requires an initial calibration process using a "standardisation template".

5 WO 02/007073 proposes a method of determining a direction in which a particular participant is pointing. It specifically addresses the problem that different people have different ways of pointing, which it asserts can lead to inaccuracies in the determination of the direction in which a
10 person is pointing – in particular in a situation (such as a video conference) where there are a plurality of participants. It therefore proposes a calibration technique in which an user points at a known target, and a reference point is determined for that user from the position of the known target and the
15 determined position of the user's finger. This reference point is then used in subsequent determinations of where that user is pointing.

SUMMARY OF INVENTION

20 Most existing camera-based systems for user interaction have significant problems. Commonly used tracking systems are either expensive or low quality, must be calibrated before use, and can only track predetermined objects, and they may perform poorly in changing lighting conditions. Meanwhile, if the user interface is based on gestures, it requires the user to
25 be taught a set of gestures, which may vary between

implementations. Implementing all functions with gestures may be difficult, requiring the use of dwell gestures for selection, the use of which make operating the system slow and cumbersome.

5 A first aspect of the invention provides a user interface for a device, the device having a display or being networked with a device having a display, the interface comprising: a tracking system capable of locating a predetermined part of a user, in space; an object tracking system capable of locating
10 an object in space when the object is pointing at the display; a unit for calculating a cursor position at which a user is pointing, using information from the face tracking system and the object tracking system; a GUI control unit to create a graphical user interface (GUI) using information from the
15 cursor position, to overlay the GUI and the cursor on the display, and to control the GUI; wherein the GUI control unit is designed to initialise the GUI when the object points at a predetermined start target; and wherein the GUI control unit responds to the cursor passing over an element of the GUI.

20 A second aspect of the present invention provides a device having a user interface according to the first aspect.

 A third aspect of the present invention provides a method of providing a user interface for a device, the device having a display being networked with a device having a
25 display, the method comprising: locating a predetermined part

of a user in space; locating an object in space when the object is pointing at the display; calculating a cursor position at which a user is pointing, using information from the face tracking system and the object tracking system; creating a graphical user interface (GUI) using information from the cursor position, and overlaying the GUI and the cursor on the display; and controlling the GUI using a GUI control unit; wherein the GUI control unit is designed to initialise the GUI when the object points at a predetermined start target; and wherein the GUI control unit responds to the cursor passing over an element of the GUI.

The present invention comprises, at least, an electronic display device (on which a graphical user interface (GUI) is to be displayed), a tracking unit for determining a cursor position on the display based on the pointing action and position of a user and a GUI control unit to generate and control the GUI on the display using information from the tracking unit and also to control a related function of the system. The GUI control unit places elements on the display, and may overlay them on of another display function (such as the normal TV image). These elements may display information, or may be interactive, as is common in a GUI.

For example, the invention may be a TV. In such case the display would normally be the familiar TV display, over which the GUI elements are occasionally overlaid, by the GUI

control unit, when the user wishes to exercise control over the TV. Further, user interaction with the GUI would lead to, for example, changing of channel, selection of input device or control of parameters such as volume, contrast etc.

5 Interaction with the GUI could also, optionally, exercise control over other devices directly connected to the TV (such as PVR, DVD or Blu-ray player), or only indirectly connected (such as lighting, heating or other home devices). The invention may also be realised as a set-top box, to be
10 connected to a TV, making use of the TV's own display to show the GUI elements. In this case the invention may control the operation of the TV through some remote control mechanism; or it may also provide the video images to be displayed. In the latter case, the TV itself may merely
15 display images, in a somewhat dumb mode, and the invention itself may include its own tuning or playback components.

According to the invention, a GUI is initialised when the user points at one of a predetermined set of positions, called the *start target*, determined by the tracking system.
20 Typically, the user points at the camera which may be above or below the display (or elsewhere), but this is not essential. Ideally, the location of this initialisation pointing event is chosen to minimise inadvertent actions, so for example simply pointing at an actor on the screen in a TV drama ought not to
25 initialise the GUI.

The position of a cursor is then controlled by the user via the tracking unit until the GUI control unit determines that the interaction is over, and the GUI is terminated until the next initialisation occurs. The user controls the cursor position simply by pointing, in a natural way, at or near the display. So long as the GUI is active a cursor is drawn on the display at the position corresponding to the pointing, to give assurance to the user that their commands are being recognised. Graphical user interface elements, commonly referred to as widgets, are arranged on the display screen with which the user may interact by moving the cursor over. When a user moves the cursor onto a widget a confirmation region may appear beside it. If the cursor subsequently passes over a confirmation region the GUI control unit may possibly change the displayed elements, possibly change the state of the system (such as changing channel) or possibly terminate the GUI.

The GUI may be terminated (to allow the system to return to its normal function, such as TV viewing) when a final action, such as channel selection, is confirmed; or when an explicit cancel operation is selected; or after a predetermined timeout of no operations; or if the tracking unit detects the user is no longer pointing at (or near to) the display; or if the tracking unit fails to determine that there is a pointing user present.

The present invention is not a gesture-based interface. Instead, it is a style of interaction whereby the user controls the system by pointing. Pointing at the start target brings up a set of selections of which the user can select one by pointing at it. Every step is interactive: as the user selects an item a confirmation button appears next to it. To make the selection, the use can point at the button or 'slide' over it, which instantly confirms the selection. No separate click action is required to operate the system. Only the sequence of positions of the pointer is significant.

The present invention is based on the concept of pointing which is universal in human societies (although with some regional variations in hand shape). In contrast, gesture-based interfaces vary between implementations and may require changing to accommodate cultural variations.

Advantageously, the present invention overcomes the problem of custom gestures that must be learned. Because it is based on pointing, the user interface is interactive and effectively tells the user what to do next. In contrast, gesture-based interfaces require training. Gesture-based interfaces may have extensive variations between implementations, so the training may not be transferable between implementations.

The present invention provides a user interface which requires no selection gesture and no 'dwell' delay, so it feels fast and responsive. Selection is easier than dwelling where

the hand or other pointing device must be kept sufficiently still for a period of time. There is no separate start gesture; instead, the user may start the interface by pointing at a defined part of the system, typically at the camera. This has the advantage of being intuitive and easy to learn, but also easy to recognise.

In this disclosure we use the term "finger" as a shorthand for any convenient extremity, object or device that the user moves into the start region to act as or in place of a finger in the role of pointing. It may for example include a drinks can, a cup, an item of food, a pencil, a magazine, a book etc, or perhaps an object specially designed for this role (such as a conventional remote controller, with or without modification) or a novelty item such as a "magic wand", or whatever happens to be in the user's hand at the time.

BRIEF DESCRIPTION OF DRAWINGS

In the annexed drawings, like references indicate like parts or features:

Figure 1 shows a typical scenario for the use of the invention, and illustrates the relationship between pointing and the cursor position.

Figure 2 shows a possible architecture of the tracker subsystem.

Figure 3 shows the location of the start region.

Figure 4 shows a possible architecture of the finger

finder unit.

Figure 5 shows a possible architecture of the finger detection unit.

Figure 6 shows details of finger template generation.

5 Figure 7 shows a set of possible states for a user interface using a simple button type graphical element (widget).

10 Figure 8 shows a set of possible states for a user interface using a button type widget, where the confirmation region appears in an alternative position.

Figure 9 shows a set of possible states for a user interface using a selection widget, where two confirmation regions appear alongside the widget.

15 Figure 10 shows a set of possible states for a user interface where the user selects an item from a hierarchical menu.

Figure 11 shows a set of possible states for a user interface where the user selects an item from a radial menu.

20 Figure 12 shows a set of possible states for a user interface where the user adjusts a value using a linear adjustment widget.

Figure 13 shows an example of clipping the cursor position to the visible screen area if the pointer points outside the visible screen area.

25 DESCRIPTION OF REFERENCE NUMERALS

- 1 a typical user
- 2 the user's face, or eye position
- 3 the pointer (which may be a finger or other object)
- 4 indicates the imaginary line passing through the
- 5 eye(2) and the pointer (3)
- 6 the cursor position
- 7 the display
- 8 TV
- 9 tracking sensors
- 10 9 example elements of the GUI
- 10 the imaginary region in which the user is expected to
- be situated
- 41 the user's arm
- 42 allowed set of cursor starting positions
- 15 43 cone-like shape
- 44 start region
- 45 start target
- 61 rectified left camera view
- 62 rectified right camera view
- 20 63 disparity map
- 64 face features in disparity map
- 65 hand and finger features in disparity map
- 66 blobs within the relevant range for the start region
- 67 the highest-scoring blob
- 25 68 a template

	71	item region
	72	confirmation region
	75	adjustment region
	76	adjustment indicator
5	81	visible screen area
	82	area around the visible screen area which can be tracked
	83	the position where the user is pointing outside the visible screen area
10	84	the clipped cursor position corresponding to the pointer
	700	GUI control unit
	801	rectification unit
	802	disparity unit
15	803	face finding unit
	804	finger finding and tracking unit
	805	cursor position calculation unit
	810	processing unit for the tracking unit
	811	rectified image data from the rectification unit 801
20	812	disparity data from the disparity unit 802
	813	face position data from the face finding unit 803
	814	finger detection unit
	815	template generation unit
	816	tracking unit
25	817	filtering unit

8141 start region calculation unit

8142 blob detection unit

8143 blob filtering unit

8144 blob validation unit

5 DESCRIPTION OF EMBODIMENTS

Figure 1 illustrates a device including a user interface in accordance with an exemplary embodiment of the invention. A display device 6 forming part of a TV set 7 is equipped with tracking sensors 8 and a tracking system (in this example a tracker processing unit 810) arranged to track at least one user 1 in a certain (imaginary) region of space 10, at least in front of the TV set at a reasonable viewing distance and viewing angle. The TV set 7 incorporates a GUI control unit. The tracker processing unit 810 determines the position of a user's face (or eye) 2 relative to the display surface, and also determines the presence in space of an object (such as a finger 3 of the user or a part (eg a tip) of a finger 3 of the user) in a start region. The tracker processing unit 810 determines, at least some of the time, a cursor position 5 as the approximate point of intersection between the plane of the display 6 and an imaginary straight line 4 through the user's face (or eye) and finger. Note that the cursor position need not always be inside the displayable area. When suitable conditions are met a GUI control unit 700 connected to or part of the TV set 7 draws GUI elements 9 on the display,

possibly including a cursor at the cursor position 5. The GUI control unit 700 and the tracker processing unit 810 may, as is obvious, be arranged in a physically different way, for example inside the TV casing, or in a box separate from the TV; together or separately and connected by a signalling medium.

Figure 2 illustrates the preferred embodiment of the tracking subsystem, consisting of sensors 8 and processing unit 810. A pair of sensors, in this example cameras 8, are arranged to face the region of space 10 in which the user 1 is likely to be, and are connected to a processing unit 810 (which naturally need not be in the same housing as the cameras, so long as it can receive the image data without significant delay). Preferably the cameras are joined rigidly (but perhaps adjustably) so they cannot move relative to each other by accident. In addition, the cameras should preferably be firmly attached relative to the display, for the same reason. The cameras are calibrated in such a way that the relative positions of the cameras as well as any optical distortions are known (or could be calculated in a separate calibration step) and can be corrected for, and such that the fields of view overlap in such a way as to contain the region in which the user may be found. This way, as is well known, each frame captured from the camera views can be mapped by a rectification unit 801 into a common coordinate system in a

standard process widely known as stereo rectification. This process is well described in chapter 12 of "Learning OpenCV – Computer Vision with the OpenCV Library" by Gary Bradski & Adrian Kaehler, published in 2008 by O'Reilly Media, Inc.

5 The rectification unit may additionally provide other services, such as partial removal of digital noise and automatic gain control for the cameras (to ensure that the image is neither under-exposed nor over-exposed) which would lead to poor tracking results.

10 The requirements for the cameras depend on the quality of the optics and sensor, on their spacing and alignment, on the number and position(s) of the user(s), on the lighting levels and the complexity of the surrounding visual field. In practice, good results can be obtained with average quality
15 640x480 monochrome sensors sampling synchronised frames at up to 48Hz, 6mm lenses and well-aligned (that is, near to parallel optical axes and near to parallel vertical axes), with a baseline stereo separation of 12cm.

20 Given views from two cameras 8 that look in roughly the same direction, some distance apart, it is possible (as is well known) to calculate a depth map. A disparity unit 802 does this by matching regions in the rectified camera views and calculating the disparity between corresponding object pixels in the two views. For example good results may be obtained
25 using the OpenCV software library function

cvFindStereoCorrespondenceBM as described in chapter 12 of "Learning OpenCV - Computer Vision with the OpenCV Library" by Gary Bradski & Adrian Kaehler, published in 2008 by O'Reilly Media, Inc.

5 For convenience in the following description we use the convention that a disparity of 0 corresponds to an object at great distance, and larger (positive) values of disparity correspond to nearer objects (if this is not the case, it is easy to correct for a non-zero infinite disparity by a simple horizontal offset and perhaps reversal of sign in the calculation). Given such a stereo pair of images, and a depth map, the two can be combined so that a substantial part of each (two-dimensional) image can be identified with a particular depth, that is, distance from the camera. In this way three-dimensional information can be reconstructed from the pair of cameras. A face finder unit 803 detects the position of any user face in the scene. Using the disparity information from the rectification unit 801, the disparity unit 802 and the face position a finger finder unit 804 locates and tracks a finger over time. A cursor position unit calculates the approximate position of the cursor in the plane of the display 6, or else reports that it cannot detect any pointing activity in the scene.

25 Strictly speaking, the rectification unit 801 is not necessary, however, as is well known, it may be commonly

used because it allows a much computationally cheaper algorithm to be used in the disparity unit 802. It also allows greater accuracy in mapping image points to real points in 3D, since it accounts for camera distortion. However, as an
5 alternative it would be easy, and cheaper, to correct the resulting finger and head coordinates after tracking using uncorrected image data.

Human face detection is a well understood problem in computer vision that can be solved in various ways. A
10 common method is to create beforehand a generalised face template from a number of face photographs, and search for regions of the frame that bear resemblance to the face template. The template need not be a simple bitmap; one method is to represent the face as a weighted sum of simple
15 signals, or wavelets, which can then be matched against regions of the frame. Many of the face matching algorithms are illumination invariant, that is, robust against changes in illumination. One suitable and very well known method for face detection is the method of Haar classifiers, described by
20 Lienhart and Maydt in "An Extended Set of Haar-like Features for Rapid Object Detection" (ICIP02, pp. I: 900-903, 2002) and implemented in the OpenCV machine vision library, available from <http://sourceforge.net/projects/opencvlibrary/>.

Some face recognition systems return a list of face
25 candidates with coordinates and perhaps a confidence

measure. If the face recognition algorithm is applied on both camera views independently, the results can be merged together, thereby improving match confidence and furthermore, a good estimate for face disparity can also be found in this case.

The face finder unit 803 will preferably predict face position when the face is temporarily obscured or tilted away, and keep track of several users in the vicinity. In this way the system can be made to handle several users at once, as will be explained later.

Figure 3 shows the geometry relevant to calculating the start region 44, which is the region of space in which the finger should be found in order for detection to occur and the GUI to start up. The start region obviously depends on the position of the user's face. In this figure the position of the finger depends on the amount of extension of the arm 41. It also depends on the choice of left or right arm, the choice of left or right eye (usually the dominant eye), the distance and elevation of the camera system 8 and the way the user is feeling. If the user is using another object (such as a can or book) to point with, this will be held in a similar position. The allowed set of cursor starting positions 42 may be fairly large, and subtends an irregular cone-like shape 43 at the user's eye. Due to the amount of arm extension there will be a range within the cone representing the start region 44.

The position that the user is told to point to start is the start target 45. Note that the cone-like shape 43 need not include the camera: this is simply a convenience in case the starting instruction for the user is "point at the camera". For example, the cameras may be below the display, but the user is told to point to the top of the display to start interaction. This does not significantly change the method of calculation of the start region. It is preferable, but not essential, to position each camera so that its image plane is roughly parallel to the near and far faces of the start region 44.

Figure 4 shows the finger finder unit 804 in more detail. It inputs rectified image data 811 from the rectification unit 801, disparity data 812 from the disparity unit 802 and face position data 813 from the face finder unit 803. The finger finder can use continuous information from these sources, however for efficiency it will allow significantly cheaper processing if the finger finder 804 requests information only for relevant parts of the scene. For example, the finger detection unit 814 step needs disparity information only in a small area of the image near to the face region (the projection of the start region 44 onto the image plane as seen through the camera lens); thus the disparity unit need calculate only a much smaller region of disparity than the whole image.

The finger detection unit 814 uses the face position to determine the start region, and uses disparity information to

perform the search. When a finger is found a template generation unit 815 records information about the finger image and disparity, in the form of a template, for further tracking. Once this is done the finger finder unit 804 enters
5 a tracking phase. During the tracking phase the tracking unit 816 searches each successive camera frame for the template stored earlier. So long as the tracking unit 816 continues to find the finger with sufficient frequency and confidence the finger finder unit 804 remains in tracking
10 mode. The found finger coordinates are passed to an optional filtering unit 817. The purpose of this unit is to remove unwanted effects such as jitter, and generally to condition the coordinates reported by the finger finder unit 804 as a whole. If the tracking unit 816 determines that
15 tracking is lost the finger finder unit 804 returns to finger detection mode for the finger detection unit 814 to wait again for a finger in the start region 44.

The finger detection unit 814 is explained more fully in figure 5. A start region calculation unit 8141 uses the face
20 position to estimate the start region. The distance from the camera to the face may be estimated in one of several ways. The preferred method is to consider the disparity between face positions in the left and right images and to use triangulation. Other methods include estimating face distance by its
25 apparent size (and assuming a "normal" face size), or by

calculating some kind of maximum of the disparity in the face region of the disparity data 802. A combination of these methods may be used to improve the estimate. Next the blob detection unit 8142 looks for contiguous regions within the disparity information 812 which are inside the projection of the start region 44 onto the image plane. For example, the disparity information may be compared with threshold values to determine which pixels lie at about the right distance from the camera, and the pixels which pass the test are grouped into contiguous or nearly contiguous areas, which we call blobs. Grouping may be done using simple morphological operators (such as dilation) and simple graph connectivity algorithms, as is well known. Each blob represents a possible candidate for the position of a finger. A blob filtering unit 8143 examines all the blobs and chooses likely candidates to be the blob corresponding to a finger. A convenient way to do this is to compute a likelihood score for each blob, and discard blobs whose score lies below a threshold, determined by trial and error. For example, blobs which are too small may be due to noise in the disparity graph, or represent only part of a larger object, and so receive a low score. Blobs which are too large may represent another person walking in front of the detected face (or perhaps a child on a lap), and may also receive a low score. Higher scores may be given to blobs nearer to the centre of

the start region, and so on. The precise values and scoring functions may be worked out by simple geometry, but depend on the physical properties of the particular cameras, their arrangement in space, and viewer distance, so it is not possible to give universal equations. Finally, a blob validation unit 8144 is optionally employed to remove false positives. This unit may wait for several contiguous frames containing a successful blob before allowing the best one through. A reasonable value might be three consistent blobs in a row, depending on the frame rate of the system. Finally, if validation is passed, the finger detection unit 814 outputs the representation of the blob, preferably including data to determine which pixels form the blob, and the coordinates of the bounding box containing the blob. Note that this method will detect and locate any object of approximately the right size inside the start region, and does not require any particular hand shape or skin colour, or even that the hand is empty.

The template generation unit 815 receives successful blob data from the finger detection unit. Depending on the finger tracking unit 816, different information must be stored as a template to represent the blob. In the simplest case, the template is just the region of the image data (from at least one of the cameras) which corresponds exactly to the blob. In this case, the set of positions of pixels in the blob may be

regarded as a mask, so that only pixels of interest to the template are selected by the mask.

Figure 6 shows the finger detection and template acquisition for easier understanding. The rectified left camera view 61 and rectified right camera view 62 of course appear slightly different because each camera has a different perspective. Here they are shown in outline only, for clarity. By measuring how far each element in the image is shifted the disparity map 63 corresponding to a possible start region is calculated. Here it is shown as a contour map for clarity. The large upper feature 64 corresponds to the face, and the large lower feature 65 corresponds to the hand and finger. The fine contour lines within show small gradations of disparity. Other small features may correspond to noise, where the disparity unit has failed to make a good estimate of disparity, and it remains to construct a template without being overly confused by the noise. Connected regions of disparity within the range appropriate to the start region are retained as blobs, whilst other regions are removed, as shown in 66. Finally only the highest scoring blob, in this case the largest blob is retained, as shown in 67. The template generation unit 815 uses the remaining blob 67 as a mask and its position in the (in this case) left image to extract a template 68 containing only the finger and hand.

The tracking unit 816 considers each frame of input

and tries to find the position of the template within the frame. There are many possible tracking algorithms, as is well known. A simple method is to simply search for the best match with the template at each position in the image. If the best match is good enough then tracking succeeds and outputs the coordinates of the best match. Otherwise tracking fails. Many more efficient and suitable algorithms for tracking an object through a sequence of frames are available, including feature tracking using SIFT or SURF features, as is widely known. In any case, once a candidate location for the best tracking position is known it can be checked in the disparity map to confirm that it has the correct 3D position (using similar criteria as were used for blob candidate selection in the blob filtering unit 8143). Again, this requires only a small region of disparity, of the size of the template, and so is efficient to compute.

Preferably, but optionally, the template may be updated during tracking using subsequent frames, in order to take account of changes to the appearance of the finger over time (due for example to changes in lighting, hand shape, orientation or perspective). This is easily done since the point of match is known, and the disparity information has been obtained at that point to check the match. This information can be used to acquire and mask a new template.

The tracker terminates tracking when it can no longer

predict the finger position with sufficient confidence, for example if the block match score is low for too many successive frames, or if the disparity check fails. Additionally the tracker may terminate following a signal from the GUI, for example when the user finalises an interaction by selection cancel or OK. Once the tracker terminates, the finger detector may start searching again for a finger in the start region 44.

It has been found advantageous to apply an edge detection filter to both the template and the search region before running the search. This results in a more robust match of the finger position. A suitable and well known edge detection filter for this purpose is the Canny filter.

In the common case, the cursor is controlled by the user's finger, in a conventional pointing pose. The locus of the pointer on the screen, that is the cursor position, is computed by 3D geometry in the cursor position unit 805; in the preferred embodiment, the cursor lies at the intersection of the display and a line that extends from the user's eye and finger onto the screen. In other words, the cursor appears in the position where the user is pointing, as the user would naively expect.

In practice, the precise relative coordinates of camera, face, finger and display are not known, or known only approximately. As a practical step, the cursor position may

be calibrated as follows: the user points to (at least) the four corners of the display in succession, while the computer records the corresponding face and finger coordinates for each corner. Subsequently in use the actual cursor position is determined by interpolation of these recorded calibration coordinates, for example, using the well known technique of barycentric coordinates for interpolation as described by Bradley in "The Algebra of Geometry: Cartesian, Areal and Projective Co-ordinates" (Bath: Highperception), scaled and offset depending on the relative face position. This calibration, or its equivalent, may be performed at the factory in a case when the camera is bonded immovably to the display panel.

It is important to keep latency (the difference in time between a user's action and the user seeing the movement of the cursor) reasonable small, to help reinforce the user's mental impression that the cursor is directly controlled by the hand movement. As a rough guide, latency of less than 100ms is good, but 50ms is better still. Latency can be reduced by using faster components, such as camera and processing units, and by careful implementation of the tracking algorithms. Additionally, it is well known that jitter (or, apparently random fluctuations) of the cursor position lead to a poor user experience. Vision based tracking algorithms are notoriously prone to jitter, due to noise in the

camera sampling, and due to the amazing complexity of visual scenes. However, it is essential to keep jitter low, to prevent the user from making inadvertent selections within the GUI, which would lead to frustration. Thus the amplitude of jitter is ideally significantly smaller than the size of the GUI elements such as selection boxes.

Therefore it may be preferable to apply a jitter reduction step to the finger position in the filter unit 817. For example, the well known Kalman filter, as described by Kalman in "A new approach to linear filtering and prediction problems" (Journal of Basic Engineering 82 (1): 35-45), may be used to smooth the motion of the finger position without introducing latency. A simple averaging filter may be used too, though this can add latency.

Cursor position jitter can be reduced by directly filtering the output of the cursor position unit 805. Alternatively, the inputs to the cursor position unit 805 (i.e. the finger position and the head position) can be filtered. In addition, since small amounts of jitter in the face detection position can logically result in large cursor movements on the display it may be preferable to capture the face position at the time that a template is acquired, and use that same head position in calculation of the cursor position until the cursor is released, even if the user's head moves (or appears to move, due to inaccuracies in the face finding algorithm).

In another embodiment of the invention it is possible to correct for errors in cursor position due to differences in pointing due to left and right finger, or due to dominance of the left or right eye, or due to pointing with an oddly-shaped object (such as a rolled up newspaper). In this case we observe that on entering the start region 44 the user is probably intending to point at the start target 45. The finger finding unit may therefore compensate by assuming this intention to point, noting the detected finger position and the corresponding offset from the assumed position, and invert the offset to give an adjustment which would compensate for the erroneous cursor position. This adjustment is then applied to the subsequent tracking, with the result that the cursor lies closer to where the user expects it should be.

In the preferred embodiment it is possible for more than one user to be present at a time. To achieve this result one method the system may use is as follows: The face finder unit 803 algorithm is adjusted to return the positions for all the faces in the camera's field of view. It must also keep a predictive model of the face positions, so that the different users are kept distinct, and tracked even if their faces are briefly not detected. However, if a face is not detected for more than a few seconds it should assume the user is no longer present (or has moved to a different location). The

result of this step is at any time there is a list of users with their probable face positions. For each face position there is a corresponding start region 44, thus the finger detection unit 814 must check all of the start regions for a finger, until it
5 finds a finger present. A template is generated for that finger, and the tracking unit 804 then tracks that template until tracking terminates, with the cursor position unit 805 using that tracked position and the corresponding user's face position to calculate the cursor position. The finger
10 detection unit does not begin to detect a new finger until tracking terminates on the previous finger. In this way, only one user at a time controls the cursor.

In another embodiment the face finder unit may also incorporate a face recognition unit, so that different viewers
15 may be distinguished and remembered from session to session. In this case it would be easy to permit individual calibration for each user's preferred style of pointing (such as straight arm, or bent arm for example, and dominant eye). This would allow a more precise calibration per person, and a
20 better user experience.

In another embodiment, using a face recognition unit would allow only certain authorised viewers to control the display, or provide different capabilities for different viewers. This would be of use in a public space, where members of the
25 public might have less authority over the display than

previously registered officials; or for TV where for example children may be given only restricted viewing controls.

Although the invention may be used with virtually no instruction, it may be advantageous to allow users to practise the kind of steady pointing they need to operate such an interface. A face recognition unit could be used so that when a user not seen before by the system starts to operate the interface more help is given. In this "beginner mode" instructions can be printed on the display, and fewer widgets could be available, with a method for enabling the normal, "advanced mode" after a suitable amount of familiarisation. It may be advantageous to present a "doodling mode" in which the user can practise pointing and controlling the cursor without any menu items to clutter the display.

In addition, a sensitivity zone can be defined. The sensitivity zone is a region in front of the user within which all pointing actions take place and therefore within which the finger or pointing device should be tracked. The sensitivity zone encompasses all spatial locations in which a pointing finger could lie whilst pointing either at the display or within an extra margin beyond the display (to allow for tracking and user inaccuracy.) When the finger or pointing device leaves the sensitivity zone, the pointing action may be considered finished. The extent of the sensitivity zone may depend on the 3D location of the detected face. Restricting tracking to

this zone naturally reduces the amount of computation required (since the zone will generally contain fewer pixels than the entire captured image).

There are instances where the face cannot be found in the image. If the user interface is activated by the action of pointing at or near to the camera, the face is likely to be obstructed at this point. In addition, because face detection is a relatively expensive operation, it may not be practical to obtain the face position for each frame. In a typical case the user is sitting down or otherwise relatively still while controlling the television; in these instances, it can be assumed that the face position does not change much between subsequent frames. In these instances, historical or predictively filtered face position data may be used. It is common to use a Kalman filter for predictive filtering in applications of this type.

Although a stereo camera system has been described, it should be clear that any method can be used to obtain a two-dimensional picture along with depth information. Possible ways of doing this include but are not limited to special depth cameras, cameras or other sensors mounted on beside, above or below the viewing area, ultrasonic sensing and so on. In the absence of complete depth information about the scene, certain things may be inferred from a single two-dimensional image using indirect metrics such as apparent eye separation

for head position.

Apart from the considerations around obtaining depth information, a number of options are available for the cameras used in the tracking. Colour cameras are not
5 required but, if they are available, then colour information may be incorporated in the tracker. If the system is required to perform in a dark or dimly lit environment, then a camera sensitive to infrared light may be used, optionally with an infrared light source.

10 An embodiment of the invention makes use of the observation that the user's head moves less than the finger or pointing device. Finding the face, potentially one of the most computationally expensive parts of the system, can be performed less often than finding the finger or pointing device.
15 In an image frame where the face is obscured, or where the face detection information is not available because it has not yet been computed, the face location can be estimated from previous locations. It is also possible to switch from the robust but expensive generic face finding mode to a simpler
20 template finding face tracker once the face has initially been found.

In an embodiment of the invention the position of the face and finger or pointing device are filtered in order to reduce cursor jitter. Applying such filtering should be done in
25 such a way that the latency of the system is not increased.

Predictive filters such as the Kalman filter can be used for this purpose, as is well known.

In an embodiment of the invention, when the user points at the camera to activate the system, a cursor appears on the display. The position of the cursor corresponds to the position of the user's head and finger. As the user moves the finger, the cursor follows the motion. As the position of the user's head and finger or pointing device are known in three dimensions, the cursor can appear on the display surface on the intersection of an imaginary line that extends from the user's eye through the pointing finger.

In a system where the positions of the cameras are fixed and known relative to the display and each other, and the distortions inherent in the camera optics are known and can be characterised, there is no need to calibrate the system. In such a system, a 3D position of any object in the view of the cameras can be computed in a coordinate system that is registered with the display. It is therefore possible to compute in three dimensions the position of the face, and the position of the finger, and find the locus of the cursor on the screen without requiring a separate calibration step.

Figures 7-12 show some possible states for a user interface, including examples of graphical user interface elements or widgets that could be used by such an interface. The figures show only the user interface layer. In a practical

system, the user interface layer would typically be overlaid on other information displayed.

In Figures 7-12, State (a) shows the state where no user interface elements are displayed. This is the normal state when the user is not interacting with the device. Typically the display would be filled with content the user is watching, such as a TV programme (but for clarity this is not shown in the diagram). The cursor is shown as a typical 'mouse pointer' shape but may be of any desired shape or image.

A number of widgets may exist for various types of interaction. A simple selection widget as shown in Figure 7 may be the simplest one. When the user interface is activated, only the item region 71 is shown. Once the user moves the cursor onto the selection widget, in this case by moving to the right, a confirmation region 72 appears beside it. The location of this confirmation region may depend on a variety of factors including the position of other widgets and the direction from which the user approaches the widget. When the user moves the cursor onto the confirmation region the confirmation region is activated and the GUI control unit 700 may perform an action associated with the selection widget. The range of cursor positions required to activate a confirmation region may correspond to the area of the confirmation region shown on the display. Alternatively, said range of cursor positions may be smaller than the area of the confirmation region in

order to mitigate effects such as cursor jitter and unintended motion of the user's finger.

In an embodiment of the invention, the position of the confirmation region of an item is determined by the position and direction that the cursor entered the item. In some cases, it may be advantageous for the confirmation region to be placed opposite or nearly opposite to the point of entry, as shown in Figure 7, so that the cursor continuing its motion traverses the whole of the item before confirmation is decided. In other cases, it may be advantageous for the confirmation region to be placed on the side of the point of entry, as shown in Figure 8, so that the user needs to move the cursor first in one direction to make a selection, and then in a different direction to confirm it, in order to avoid accidental selection.

In an embodiment of the invention, the selection widget may have multiple confirmation regions, each of which selects a different state. This way it is possible to choose from a number of discrete options. A simple example using two confirmation regions for 'Yes' and 'No' is shown in Figure 9.

In an embodiment of the invention, the action of pointing at the camera brings up an on-screen display, an example of which is shown in Figures 10-12 as State (b). If the start target region lies above the display, then the user must move his or her finger downward (into the visible screen area) to make a selection. The on-screen display can display

any number of widgets and the user may select one by pointing at it. Once the cursor is on top of a widget, the on-screen display changes to reflect the selection. In Figures 10-12, State (c) is an example of the user selecting a widget by moving the cursor over it.

The user interface items may represent choices, such as which channel to watch; or variable parameters, such as volume control; or menus and submenus to allow navigation to display more sets of user interface items. Some items operate by making a separate confirmation items appear. In these cases, the effect of the selection (such as channel changing) is not confirmed until the cursor passes over the confirmation item. In order to confirm the selection, the user moves the cursor into the confirmation region. As soon as the cursor reaches the confirmation region, the selection is made. Optionally, if the cursor leaves the selection item area without passing through the confirmation region then the confirmation region is removed from the display as if the cursor never entered the selection item area in the first place. This allows the user to back out of a decision at a late stage, and also allows for some accidental errors in finger position to be rendered harmless.

It is important that if the user feels that the cursor has passed over an active region of the GUI, then the associated action should occur. However, the finger tracker can deliver

cursor coordinates at only a certain rate, therefore the cursor may cross a widget from one side to the other without appearing at any point on top of that widget. To prevent this causing annoyance to the user it is possible to interpolate
5 intermediate cursor positions between the actual tracked positions. These intermediate positions are then treated in exactly the same way as if they were actual cursor positions.

Figure 10 shows the use of a multi-level menu. State (b) shows an initial state, to be entered when the GUI begins. In
10 State (c), the user selects a menu by moving the cursor over it. This causes a category menu to appear. As the user moves the cursor over one of the category entries, as in State (d), a category submenu appears. The user may select another category, which will be reflected by the submenu. If the user
15 moves the cursor over a submenu element as in State (e), a confirmation box appears. The user may now select the item by moving the cursor above the confirmation box, as in State (f). This style of interaction may be used with any depth of hierarchy.

20 In an embodiment of the invention, one or more of the menu entries may perform a cancellation action, taking the user back to a previous state. This works in the same way as a button widget or a menu entry, except that it is labelled with a label such as "CANCEL" to indicate its function and
25 that when the cursor moves over it and then, optionally, its

confirmation region, any pending actions are cancelled rather than confirmed.

When navigating menus it may be advantageous if moving the cursor backwards over a higher level menu causes any already pending lower level menus to be removed from the display, thus allowing the user to try a different descent.

Similar in function to the multi-level menu discussed earlier, a radial menu is represented as a number of sectors centred around a middle region. To make a selection in a radial menu, the user moves the cursor in a certain direction. The confirmation region may appear along the outer perimeter of the sector so that, when the cursor continues moving along a straight path, the item is first selected and the selection then confirmed.

Figure 11 shows the use of an example radial menu. To use the radial menu, the user selects the menu by moving the cursor over the specified region, as in State (c). This action brings up the radial menu, centred around the region, as shown in State (d). The user may now move the cursor above a menu entry to select it, as in State (e). A confirmation box now appears, the selection of which will confirm the menu selection as in State (f). As with the multi-level menu above, the radial menu may be used with any depth of hierarchy. If further levels of hierarchy are desired, they may appear as concentric menus around, or start new menus of the same or

different type elsewhere on the display. If the cursor in State (e) moves not to the confirmation region but to a different item, the pending item becomes deselected, the new item becomes pending, and the previous confirmation item is replaced by another in the appropriate new position corresponding to the new pending item.

The radial menu may be fully visible, as pictured, or it may be partly visible, for example starting from an edge or a corner of the screen. In an embodiment of the invention, one or more of the menu entries may perform a cancellation action, optionally with a confirmation region, which takes the user back to a previous state. In another embodiment of the invention, one or more of the menu entries may be missing, leaving a gap in the radial menu. This gap may be used as an exit route for the cursor so that widgets may be accessed outside the radial menu.

Figure 12 shows the use of a linear adjustment widget, used to select a value from range which may be either continuous or discrete. For example, such a linear adjustment widget may be used to control the volume of the TV audio. The user selects, State (c), a menu entry which brings up the adjustment widget in State (d). By moving the cursor along the length of the adjustment region 75, the user may make a desired adjustment as in State (e). The adjustment indicator 76 follows the cursor to give feedback. Optionally the

present value of the slider (such as volume level as a number) may be overlaid on or near the adjustment indicator. As before, the user confirms the adjustment by moving the cursor over a confirmation region as in state (f).

5 In the case of a volume adjustment, the value is essentially analogue (although it may be quantised by the system). It should be clear that linear adjustment sliders can also be used to make a selection from a set of discrete entries, particularly when the entries can be placed in an order that
10 makes sense to the user.

 In Figure 12 the adjustment widget is positioned in such a way that it replaces the menu entry, which ensures that the cursor is already in a suitable position. An alternative is to position the widget away from the menu entry such that the
15 user may observe the value first and decide whether it needs adjusting. If the value should be adjusted, the user can move the cursor over the adjustment widget and perform the adjustment normally. If the value should not be adjusted, the user can cancel the action.

20 Operating a slider requires a certain amount of careful control from the user. For commonly used functions it is important to design the GUI to make it as easy for the user as is practical. We have found it advantageous to place a horizontal volume control slider at the bottom of the display.
25 It is best if it is tall, say 10% of the height of the display, so

that the user can easily keep the cursor within the slider region.

It may be advantageous to react to pointer positions which would result in the actual cursor position to lie outside the visible screen area. If the tracking unit is arranged to track positions of the user's finger in a region larger than that which corresponds to the visible screen area - that is to say, the sensitivity zone is sufficiently large - it may be desirable to limit, or clip, the cursor position such that it stays within the visible screen area. Figure 13 shows an example of such clipping. In this case, the visible screen area 81 is smaller than the area which can be tracked 82. If the user points in a position 83 such that the corresponding cursor position would lie outside the visible screen area, it may be desirable to clip the cursor position 84 such that the cursor, or a portion of the cursor, remains within the visible screen area.

As an example of a situation where clipping the cursor position may be advantageous, we allow that even if the cursor dips off the bottom of the screen (as it might if the user's arm sags or jerks a little) it is still considered to lie inside the slider area. In other words, the horizontal coordinate of the cursor is used to control the slider position, so long as the vertical coordinate is such that the cursor lies below the top of the adjustment region.

Another advantage of clipping the cursor position is that it makes it easier to place widgets along the edges of the screen. If the cursor position is limited to the screen area, and a widget is positioned along an edge of the screen, then
5 the user may point to any position beyond the edge to select the widget. As the region where the user can point to select a widget is expanded to outside the visible screen area, selection is easier. Figure 13 shows a menu entry that is active even though the user points outside the visible screen
10 area.

Alongside interactive widgets it may be desirable to display other information, such as programme or channel names or alerts. These may appear within the user interface. They may disappear after a predefined timeout, or they may
15 be cleared by the user by selecting them, optionally with a confirmation step.

Textual or other information may also appear as a result of the user making a selection. For example, placing the cursor over a channel name (but not confirming the selection)
20 may display current and upcoming programme information on the screen.

Although a few example actions have been demonstrated, it should be clear that different widgets may be mixed as required. One example is a linear hierarchical menu where
25 certain entries bring up linear adjustment sliders or radial

menus. Another example is a horizontal linear adjustment slider where the confirmation brings up a vertical linear adjustment slider. This is one way to adjust a two-dimensional value.

5 An example where two slider-type widgets could be used is a channel map, or electronic programme guide. In many television systems such a function is implemented as a grid where the vertical axis represents channel selection and the horizontal axis represents time. In a remote controller based
10 system, the user can use the arrow keys to navigate the channel map, and an 'OK' button to select a channel or record a programme.

 Similar functionality could be implemented in a pointing-based system as follows: the channel map is
15 displayed on the screen with a vertical slider aligned alongside it. Adjusting the position of the vertical slider allows the user to choose a channel. As the channel selection is confirmed, the user may be shown a horizontal slider alongside the channel's programme information so that the
20 user may select a programme. To see programme information for another channel, the user may return to the vertical slider.

 The user may have several options to quit the on-screen display. Cancellation buttons may appear as selections on the menu system, or implicitly outside the visible screen area.
25 Widgets may have their own 'return' or 'cancel' regions in

addition to the confirmation regions. In addition, a predefined timeout may cancel the pointing sequence and quit the on-screen display. Additionally, if the tracker signals to the user interface that it can no longer track the pointer, the quit
5 action may be taken.

It may be advantageous for the pending selection to be highlighted in some way, so that the user's attention is drawn to the effect of what would happen were the confirmation region to be selected. A method of highlighting might be to
10 temporarily change the colour of the pending selection. If the user's following actions are such as to back out of a pending selection, the pending selection highlight should be removed.

When items change state, (for example to appear, to be
15 erased, to be highlighted, to be selected, pending, confirmed or cancelled) it may be advantageous to provide a gradual transition to the new state, as for example a gradual fading instead of a sudden deletion. It may also be advantageous to provide an audible cue, such as a beep or other sound effect,
20 when items change state.

It may be advantageous to provide an undo function, so that if the user inadvertently makes some change, then the next time the GUI is started choosing the undo function undoes that change and reverts to an earlier state.

25 Some confirmation regions cause the pending selection

or value to be accepted and also end the GUI session. However, when several selections or values are required it may be preferable not to end the GUI session without a further action from the user. For example, after adjusting
5 "brightness" the user will perhaps like also to adjust "contrast", and so the GUI should not suddenly end when brightness has been confirmed.

It may be advantageous to support user interfaces which require a large number of entries from which to make a
10 selection. A common example of such an interface is text entry. One method to enable this is to have widgets describing a keyboard on the display. However, it may be awkward to implement this in the straightforward way, since there must be space for the cursor to pass between unwanted
15 letter keys on the way to the required letter key. Since there are so many keys the gaps may be unfeasibly small. There are other methods which could be adapted to this system of cursor motion without a click. We mention just two: the Swype method, described in patent US7098896, where
20 passing over wrong keys is not a problem, because they are discounted by the intelligence of the system to "know" what the user is trying to type, using language analysis; and the Dasher method, described by Ward, Blackwell and MacKay in "Dasher - a Data Entry Interface Using Continuous Gestures
25 and Language Models" (Proc UIST 2000, pp. 129-137), which

uses a gesture-based user interface where the vertical axis to select letters and the horizontal axis to descend into a tree structure denoting subsequent letters, the system determining the most probable next letters and displaying them more prominently.

Although the invention has been shown and described with respect to a certain embodiment or embodiments, equivalent alterations and modifications may occur to others skilled in the art upon the reading and understanding of this specification and the annexed drawings. In particular regard to the various functions performed by the above described elements (components, assemblies, devices, compositions, etc.), the terms (including a reference to a "means") used to describe such elements are intended to correspond, unless otherwise indicated, to any element which performs the specified function of the described element (i.e., that is functionally equivalent), even though not structurally equivalent to the disclosed structure which performs the function in the herein exemplary embodiment or embodiments of the invention. In addition, while a particular feature of the invention may have been described above with respect to only one or more of several embodiments, such feature may be combined with one or more other features of the other embodiments, as may be desired and advantageous for any given or particular application.

In a user interface of the invention, the GUI control unit may respond to an interpolated course of the cursor passing over an element of the GUI.

The tracking system may be a face tracking system capable of locating a user's face, or a part thereof, in space.

The face tracking system may comprise a stereo camera.

When the cursor or the interpolated course of the cursor passes over an element of the GUI, the GUI control unit may generate a confirmation area that is displaced from the element of the GUI in the direction of travel of the cursor.

When the cursor or the interpolated course of the cursor passes over an element of the GUI, the GUI control unit may generate a confirmation area that is , displaced from the element of the GUI in a direction different from the direction of travel of the cursor.

When the cursor or the interpolated course of the cursor passes over an element of the GUI, the GUI control unit may generate a confirmation area that is displaced from the element of the GUI in a direction different from a direction of travel of the cursor required to actuate the element of the GUI.

The GUI control unit may be adapted to initialise when the object tracking system detects an object in a first region of space, the first region of space defined on the basis of a determined location of the user's face and the start target.

The user interface may be adapted to derive a correction

from a detected position of the object when the user points at the start target.

The unit for calculating the cursor position may be adapted to calculate the cursor position using the determined
5 correction.

The user interface may be adapted to determine the first region of space.

The start target may be outside an active area of the display.

10 The object tracking system may be adapted to track an object within a second region of space, the second region of space determined on basis of a determined location of the user's face.

The tracking system may be capable of locating a
15 respective predetermined part of a plurality of users, the object tracking system may be capable of locating a plurality of objects in space, and GUI control unit may be adapted to initialise when one of the objects points at the predetermined start target.

20 The GUI control unit may be adapted to initialise when the object tracking system detects an object in one of a plurality of first regions of space, each first region of space defined on the basis of a determined location of the face of a respective user and the start target.

25 INDUSTRIAL APPLICABILITY

The present invention provides a low cost and convenient user interface to an electronic product with a display.

For a fuller understanding of the nature and advantages
5 of the invention, reference should be made to the ensuing
detailed description taken in conjunction with the
accompanying drawings.

The invention being thus described, it will be obvious
that the same way may be varied in many ways. Such
10 variations are not to be regarded as a departure from the
spirit and scope of the invention, and all such modifications
as would be obvious to one skilled in the art are intended to
be included within the scope of the following claims.

CLAIMS

1. A user interface for a device, the device having a display or being networked with a device having a display, the interface comprising:

5 a tracking system capable of locating a predetermined part of a user, in space;

an object tracking system capable of locating an object in space when the object is pointing at the display;

10 a unit for calculating a cursor position at which a user is pointing, using information from the tracking system and the object tracking system;

a GUI control unit to create a graphical user interface (GUI) using information from the cursor position, to overlay the GUI and the cursor on the display, and to control the GUI;

15 wherein the GUI control unit is designed to initialise the GUI when the object points at a predetermined start target;

and wherein the GUI control unit responds to the cursor passing over an element of the GUI.

20 2. A user interface as claimed in claim 1 wherein, when the cursor or the interpolated course of the cursor passes over an element of the GUI, the GUI control unit generates a confirmation area, the confirmation area being displaced from the element of the GUI in the direction of
25 travel of the cursor.

3. A user interface as claimed in claim 1 wherein,
when the cursor or the interpolated course of the cursor
passes over an element of the GUI, the GUI control unit
5 generates a confirmation area, the confirmation area being
displaced from the element of the GUI in a direction different
from the direction of travel of the cursor.

4. A user interface as claimed in claim 1 wherein,
10 when the cursor or the interpolated course of the cursor
passes over an element of the GUI, the GUI control unit
generates a confirmation area, the confirmation area being
displaced from the element of the GUI in a direction different
from a direction of travel of the cursor required to actuate the
15 element of the GUI.

5. A user interface as claimed in any preceding claim
wherein the GUI control unit responds to an interpolated
course of the cursor passing over an element of the GUI.

20 6. A user interface as claimed in any preceding claim
wherein the tracking system comprises a face tracking system
capable of locating a user's face, or a part thereof, in space.

25 7. A user interface as claimed in any preceding claim

wherein the face tracking system comprises a stereo camera.

8. A user interface as claimed in any preceding claim wherein the GUI control unit is adapted to initialise when the object tracking system detects an object in a first region of space, the first region of space defined on the basis of a determined location of the user's face and the start target.

9. A user interface as claimed in any preceding claim and adapted to derive a correction from a detected position of the object when the user points at the start target.

10. A user interface as claimed in claim 9 wherein the unit for calculating the cursor position is adapted to calculate the cursor position using the determined correction.

11. A user interface as claimed in claim 8, or in claim 9 or 10 when dependent from claim 8, wherein the interface is adapted to determine the first region of space.

12. A user interface as claimed in any preceding claim wherein the start target is outside an active area of the display.

13. A user interface as claimed in any preceding claim

wherein the object tracking system is adapted to track an object within a second region of space, the second region of space determined on basis of a determined location of the user's face.

5

14. A user interface as claimed in any preceding claim wherein the face tracking system is capable of locating a respective face, or a part thereof, of a plurality of users, the object tracking system is capable of locating a plurality of
10 objects in space, and GUI control unit is adapted to initialise when one of the objects points at the predetermined start target.

15. A user interface as claimed in claim 14 wherein the
15 GUI control unit is adapted to initialise when the object tracking system detects an object in one of a plurality of first regions of space, each first region of space defined on the basis of a determined location of the face of a respective user and the start target.

20

16. A device comprising a user interface as defined in any one of claims 1 to 15.

17. A method of providing a user interface for a device,
25 the device having a display or being networked with a device

having a display, the method comprising:

locating a predetermined part of a user in space;

locating an object in space when the object is pointing at the display;

5 calculating a cursor position at which a user is pointing, using information from the face tracking system and the object tracking system;

 creating a graphical user interface (GUI) using information from the cursor position, and overlaying the GUI
10 and the cursor on the display; and

 controlling the GUI using a GUI control unit;

 wherein the GUI control unit is designed to initialise the GUI when the object points at a predetermined start target;

 and wherein the GUI control unit responds to the cursor
15 passing over an element of the GUI.

1/13

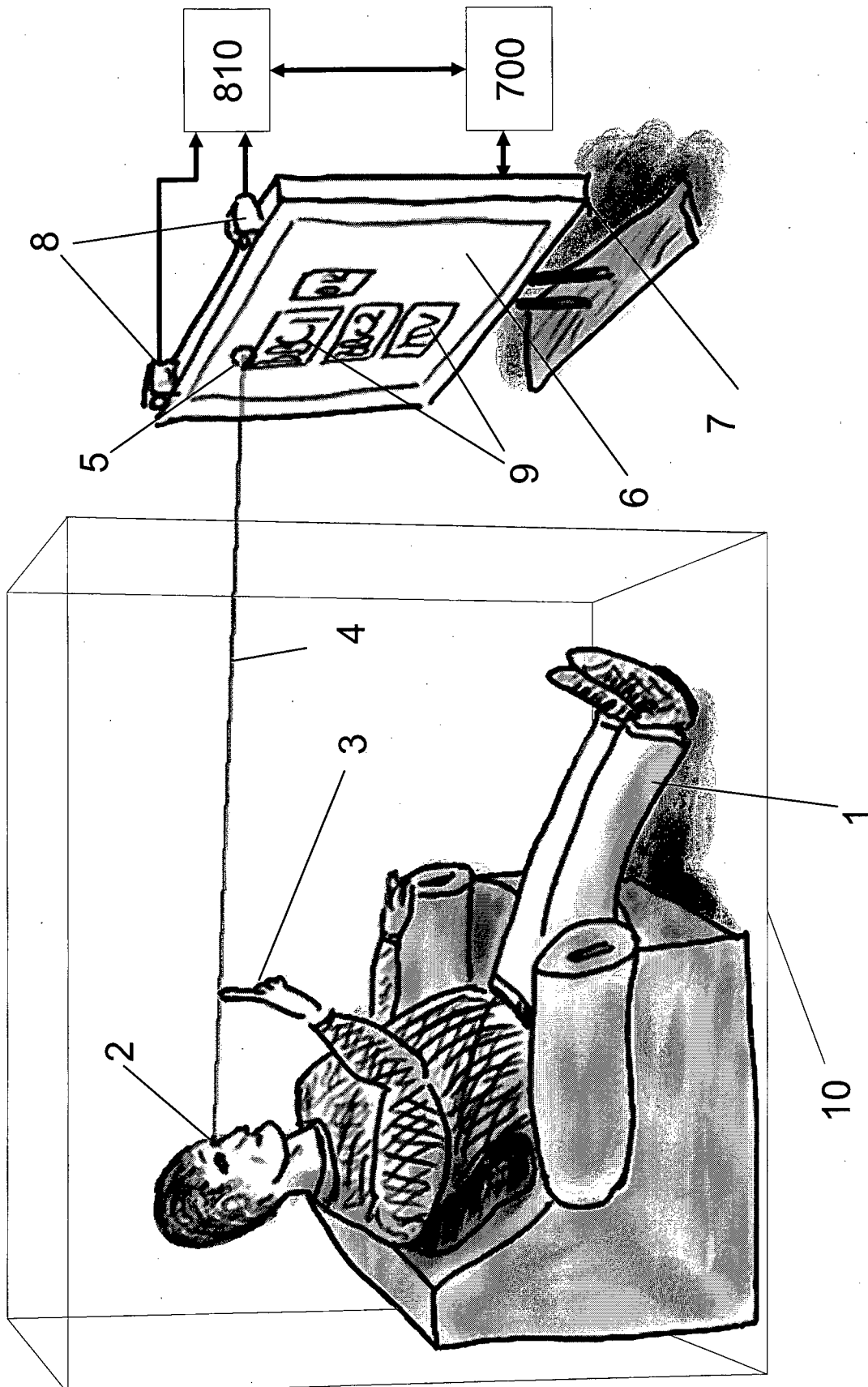


Figure 1

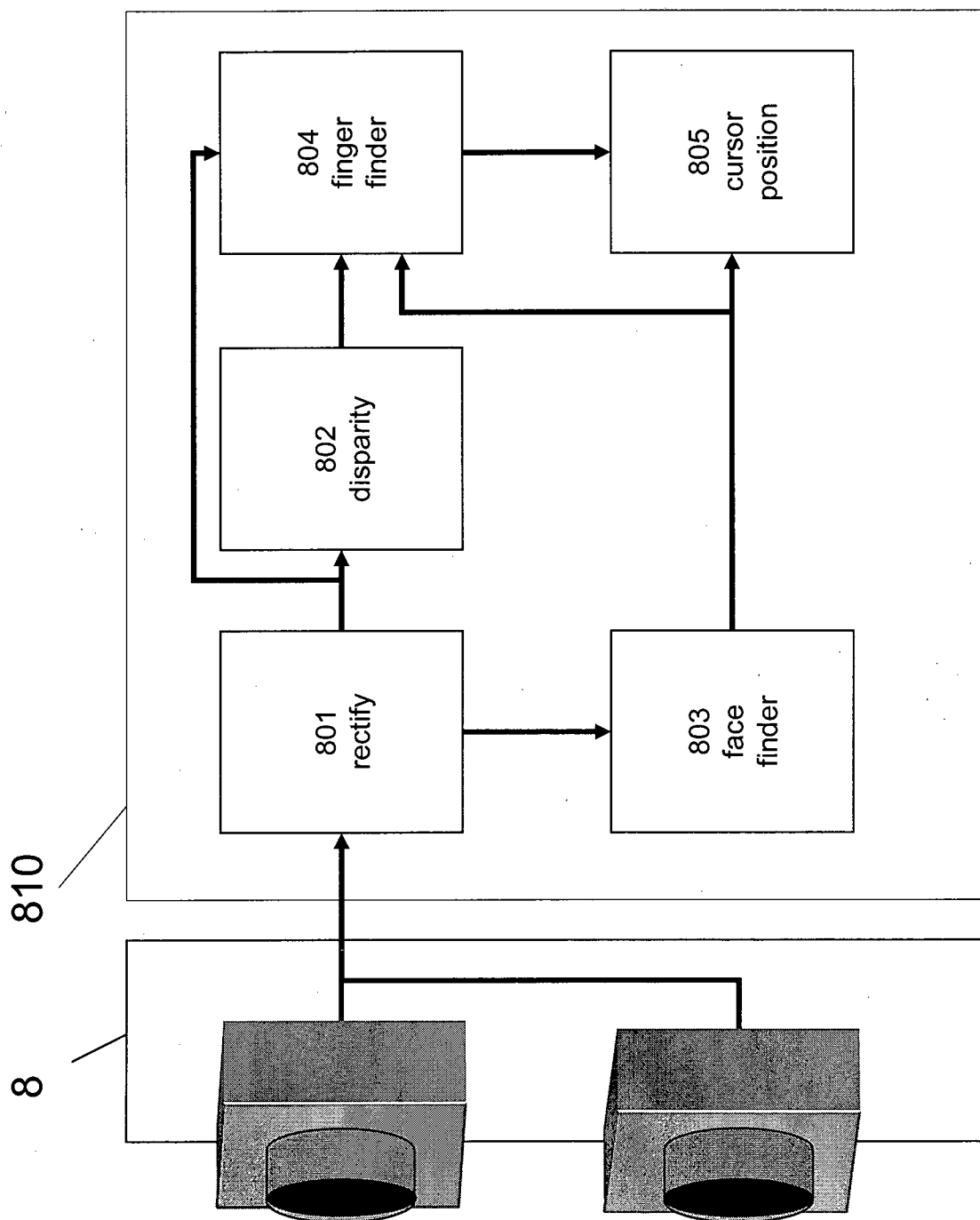


Figure 2

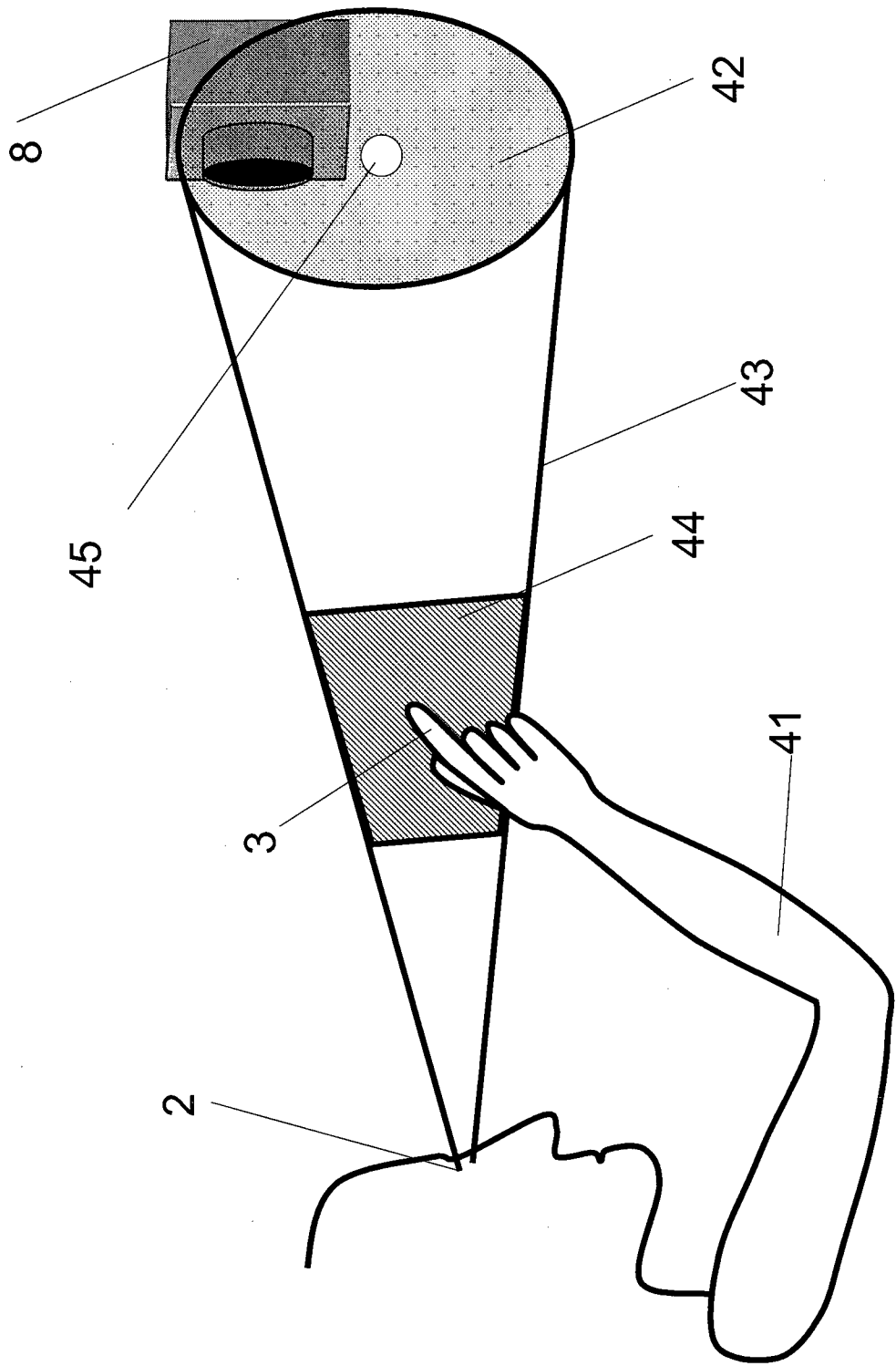


Figure 3

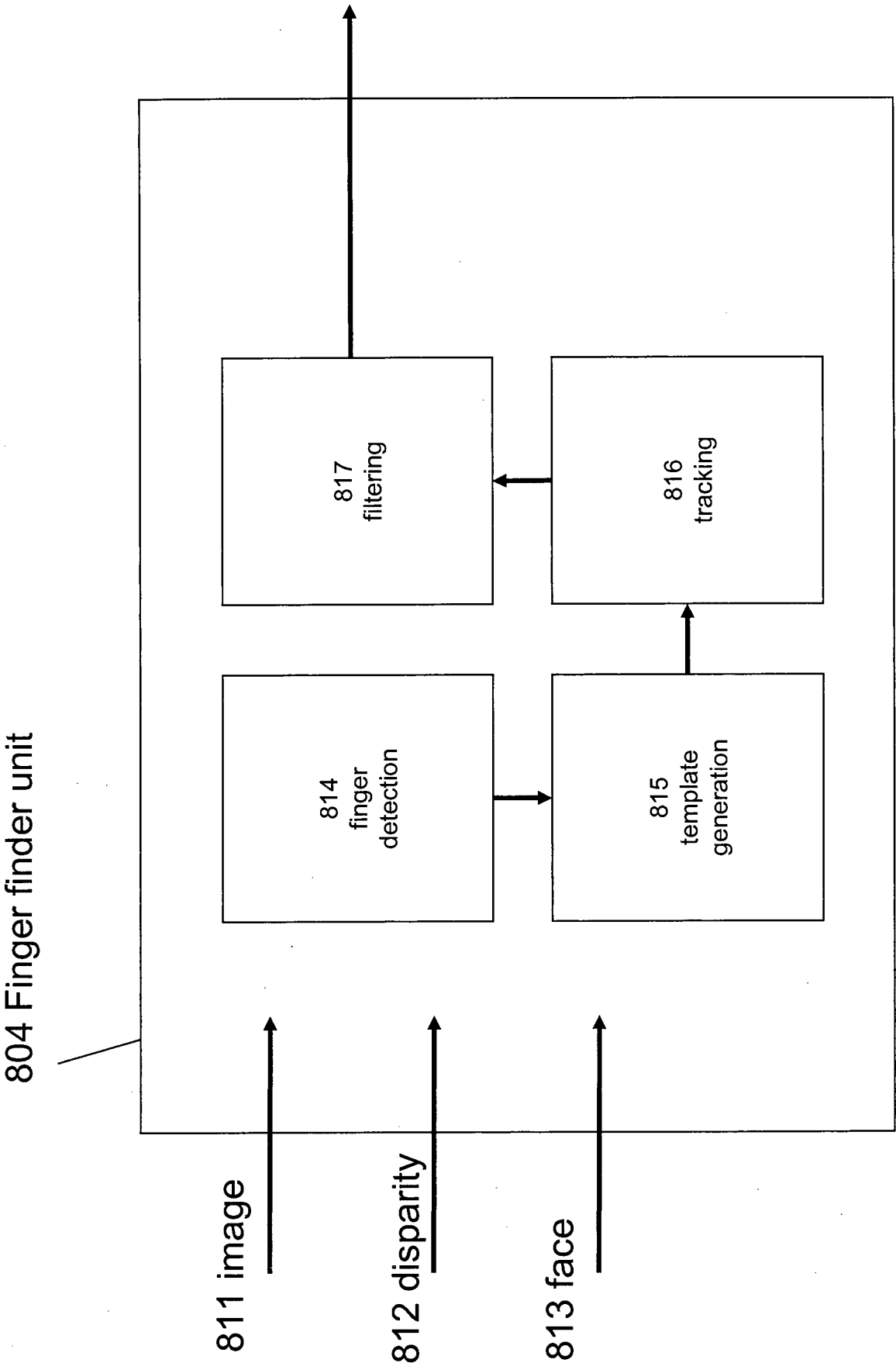


Figure 4

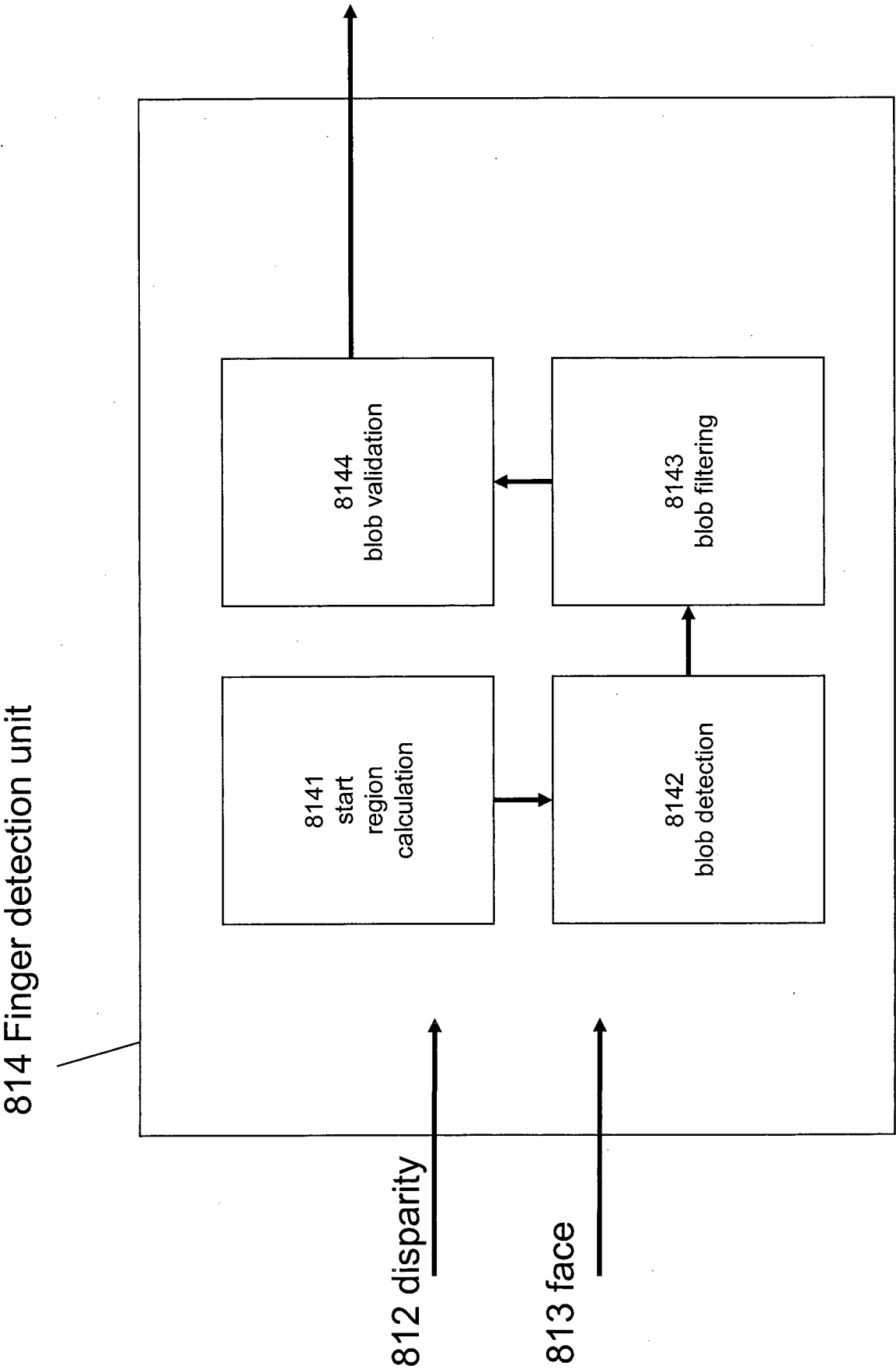


Figure 5

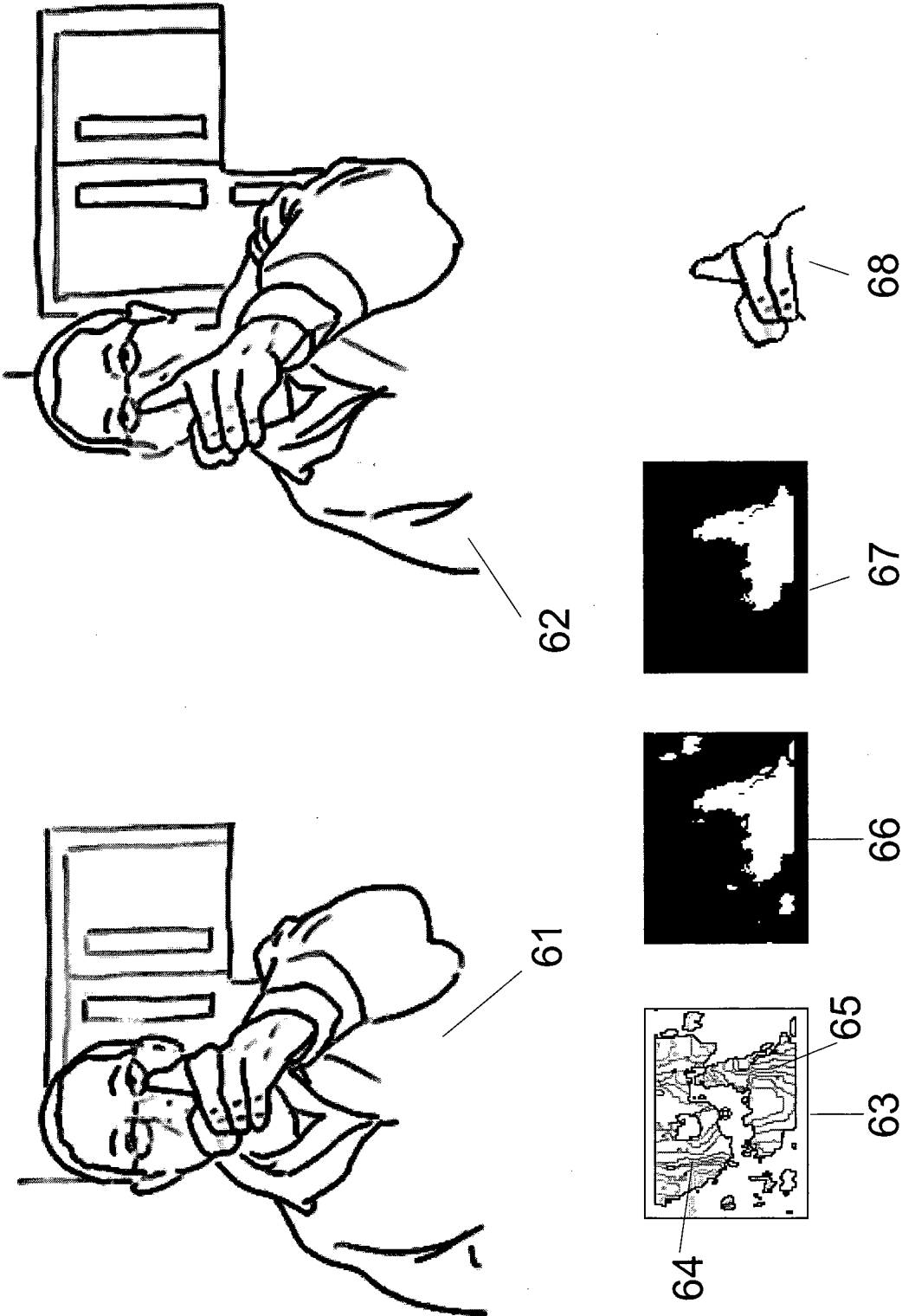


Figure 6

7/13

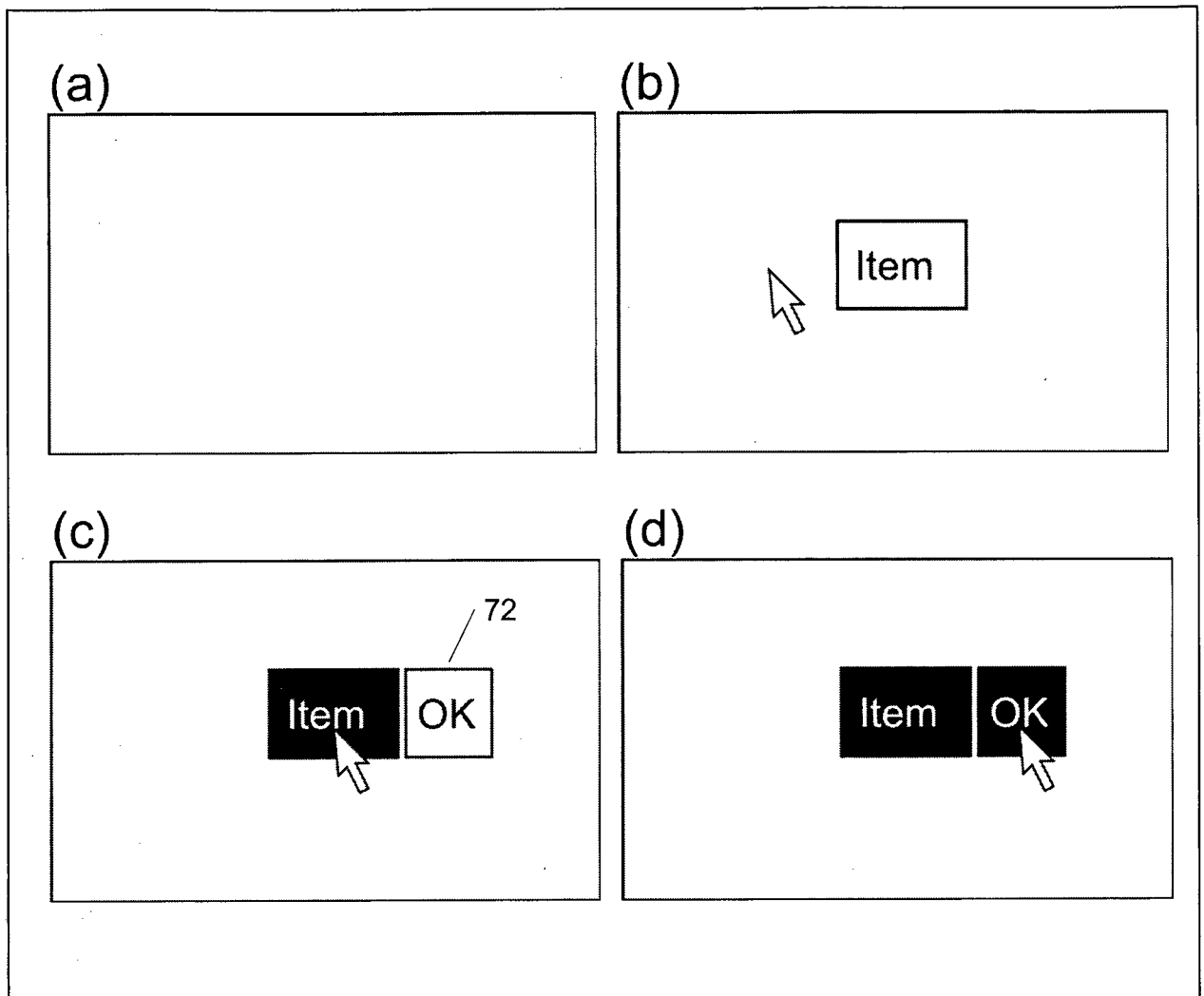


Figure 7

8/13

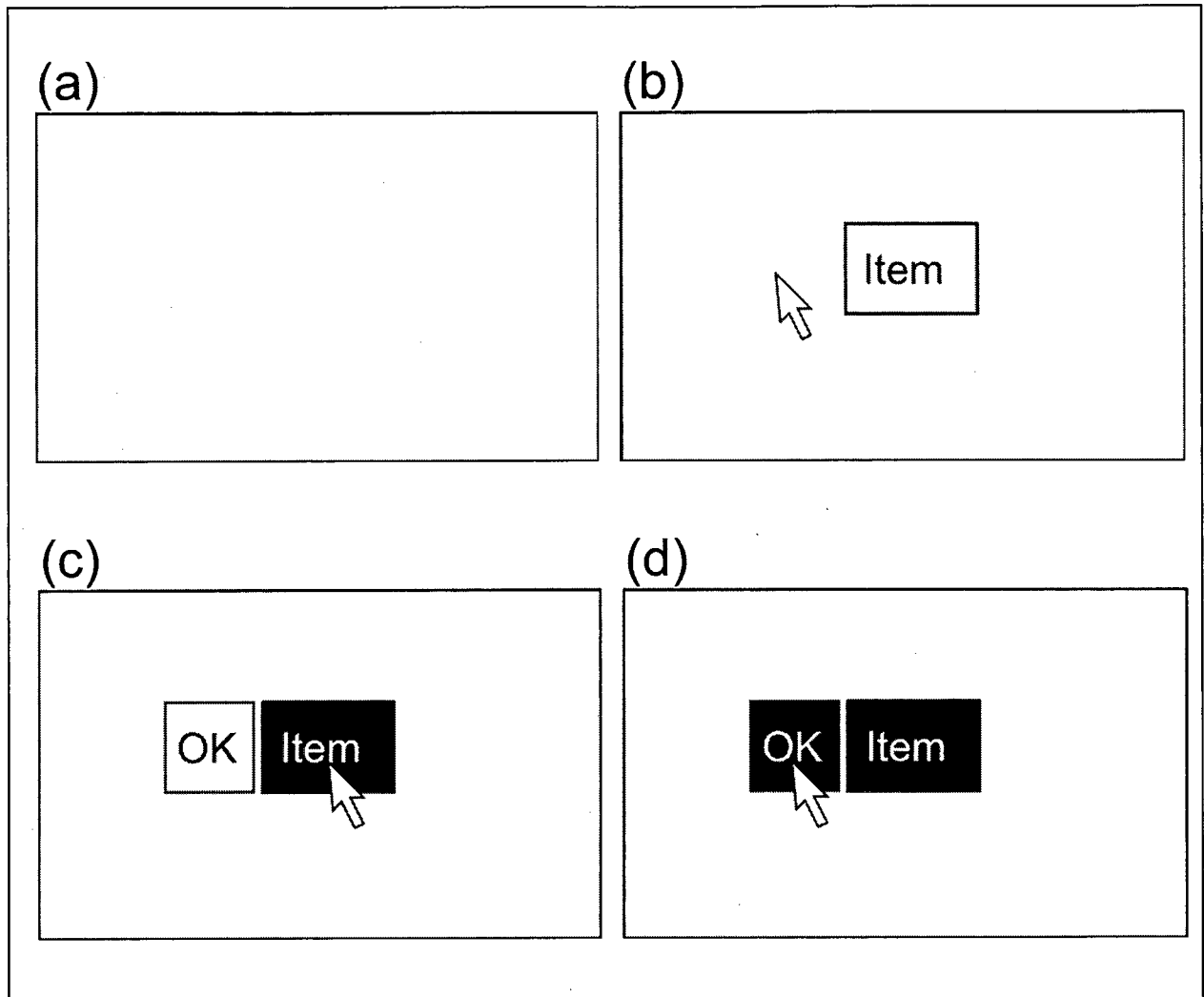


Figure 8

9/13

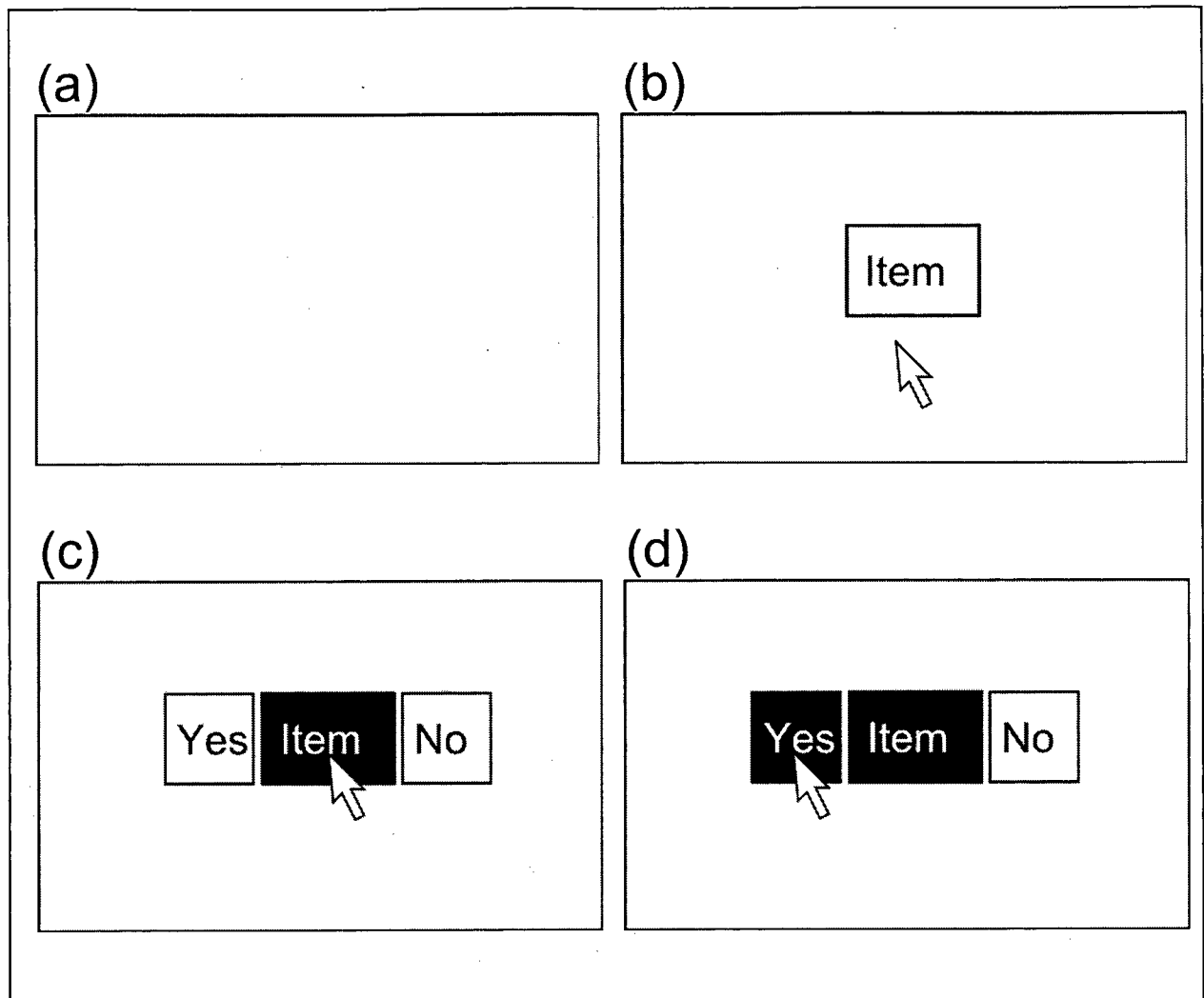


Figure 9

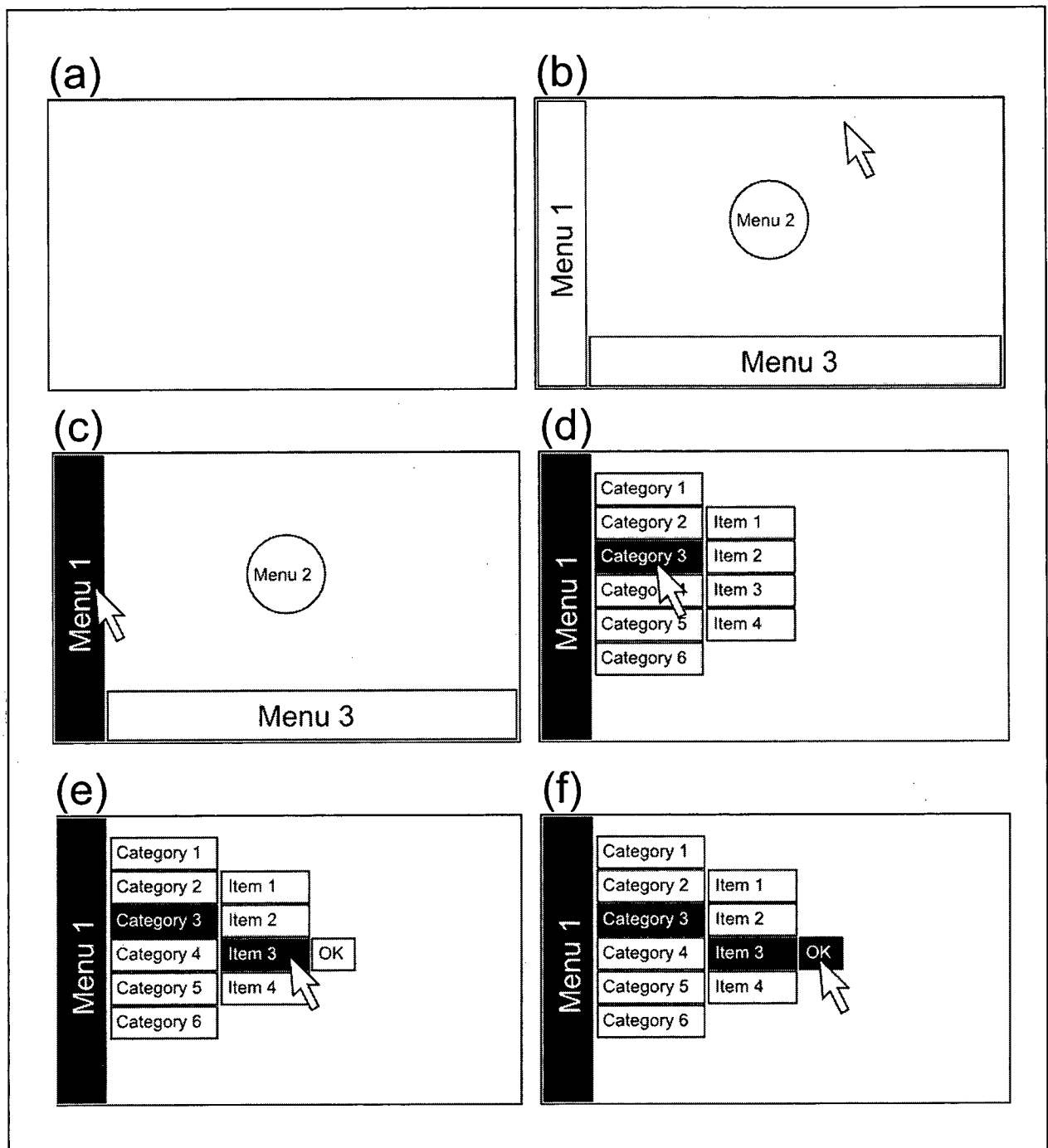


Figure 10

11/13

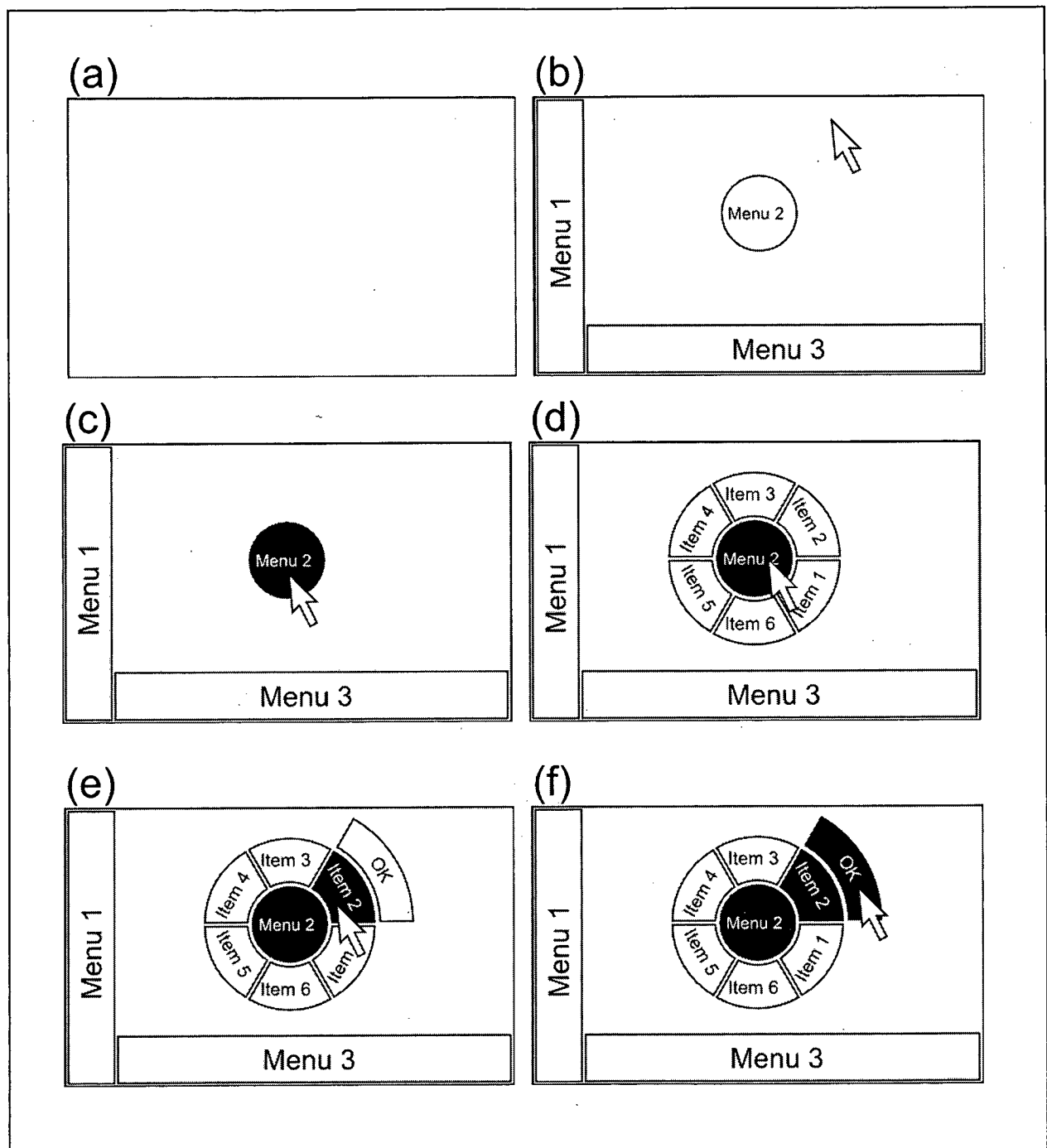


Figure 11

12/13

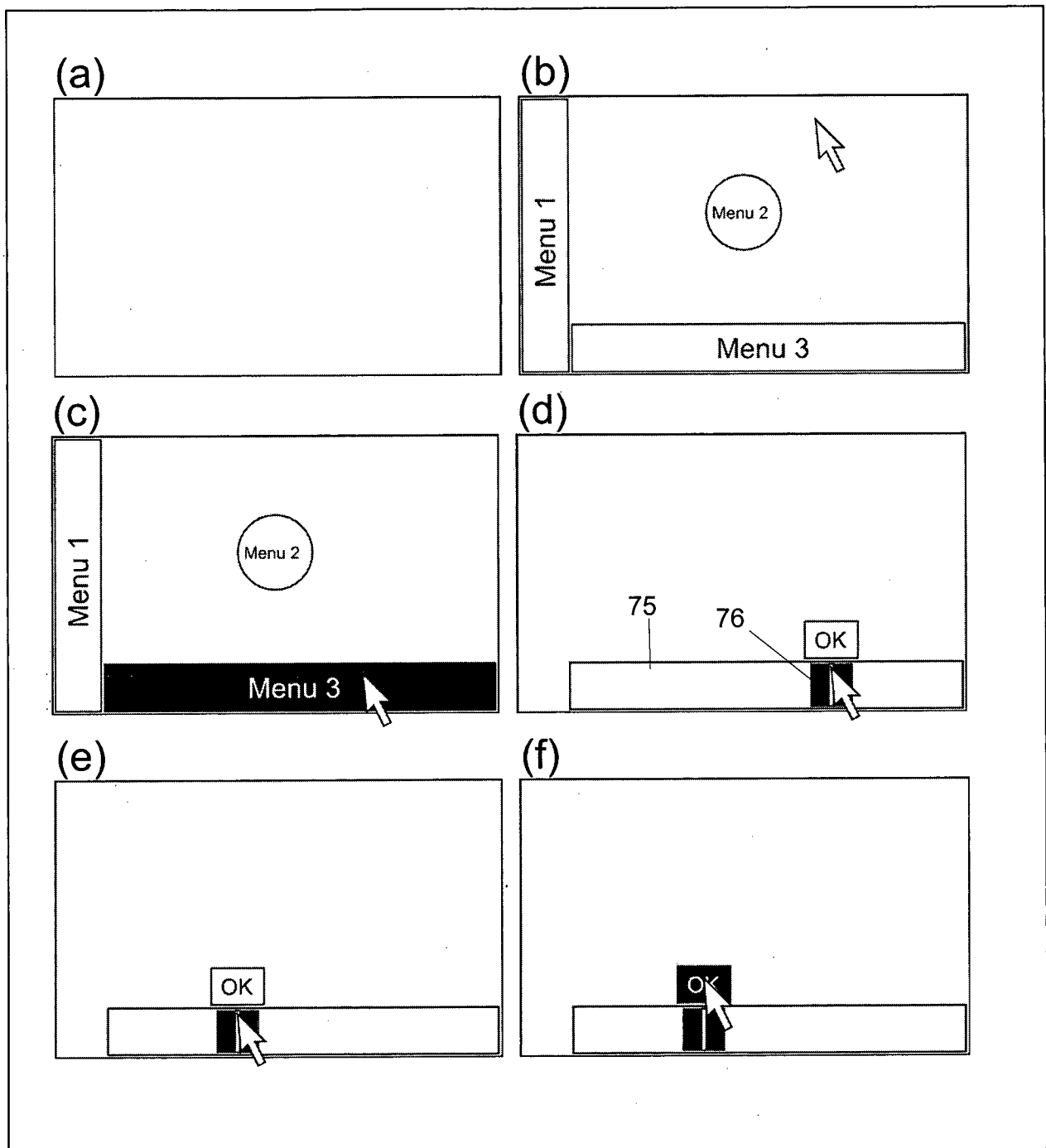


Figure 12

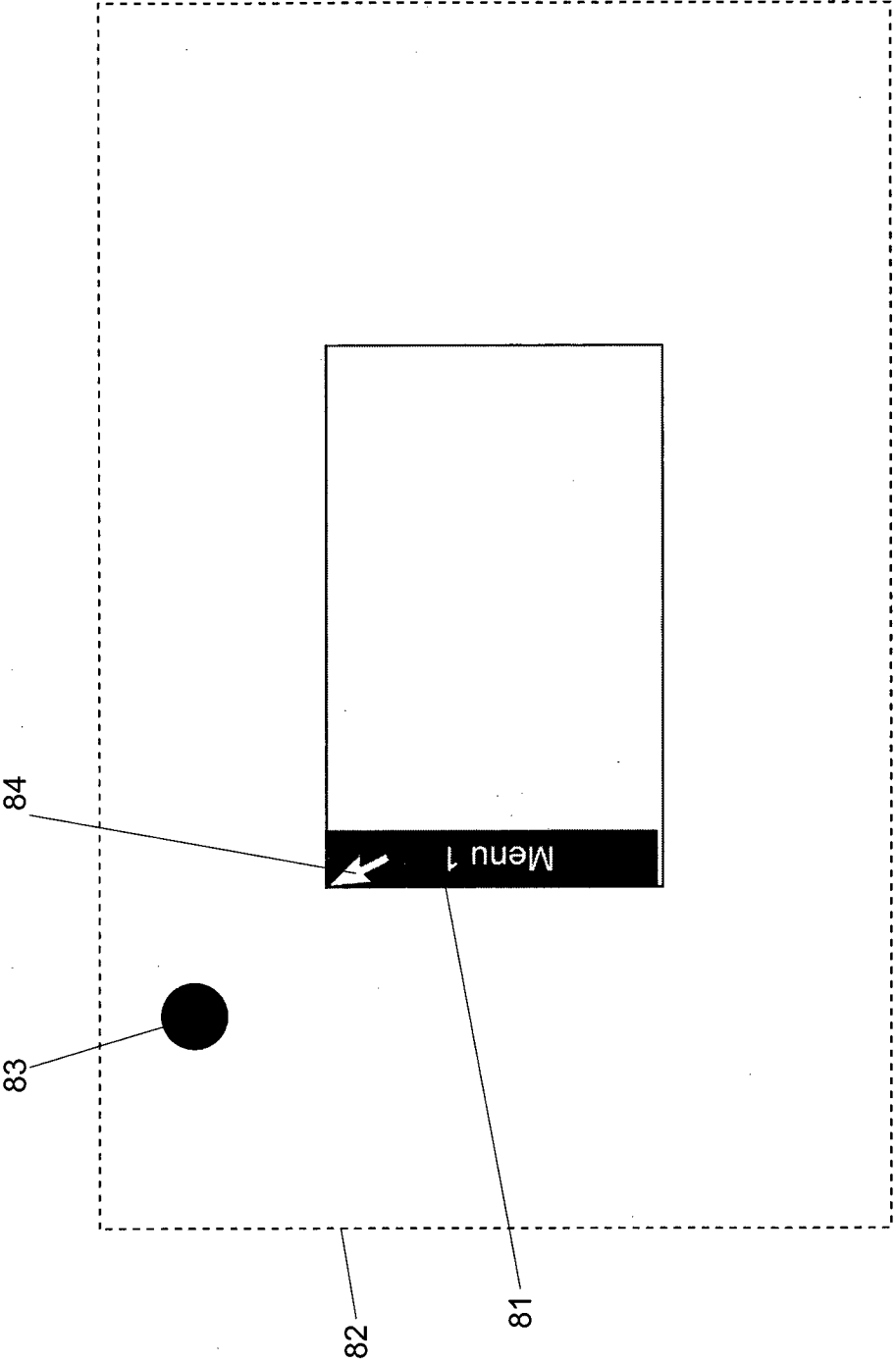


Figure 13

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2012/056218

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. G06F3/048 (2006.01) i, G06F3/01 (2006.01) i, G06F3/033 (2006.01) i,
G06F3/038 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl. G06F3/048, G06F3/01, G06F3/033, G06F3/038

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan 1922-1996
Published unexamined utility model applications of Japan 1971-2012
Registered utility model specifications of Japan 1996-2012
Published registered utility model applications of Japan 1994-2012

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JP 2011-39844 A (Shimane Prefecture) 2011.02.24, Column[0006]-[0007], [0038]-[0057], Fig4, Fig8, Fig9, Fig17 & WO 2011/018901 A1	1-17
X	JP 2010-45658 A (Sony Corporation) 2010.02.25, Column[0088], [0103]-[0104], [0134]-[0139], Fig9-15 & US 2010/0039378 A1 & CN 101651804 A	1-17
A	JP 2008-90640 A (Toyota Motor Corporation) 2008.04.17, Column [0064]-[0072], Fig13-16 (No Family)	2-5
A	JP 2010-534895 A (GestureTeck, Inc.) 2010.11.11, Column[0068]-[0073], [0124], Fig5-14 & US 2009/0031240 A1 & US 2009/0027337 A1 & WO 2009/018161 A1 & CN 101810003 A	6-15

☐ Further documents are listed in the continuation of Box C.
☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

05.06.2012

Date of mailing of the international search report

12.06.2012

Name and mailing address of the ISA/JP

Japan Patent Office

3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan

Authorized officer

Akiko NISHITANI

Telephone No. +81-3-3581-1101 Ext. 3521

5E

4 5 3 6