(12) STANDARD PATENT (11) Application No. **AU 2008299579 B2**

(19) AUSTRALIAN PATENT OFFICE

(54) Title
A system and method for manipulating digital images on a computer display

(51) International Patent Classification(s)
G06F 3/03 (2006.01) ***G06F 3/042*** (2006.01)
G06F 3/041 (2006.01)

(21) Application No: **2008299579** (22) Date of Filing: **2008.09.11**

(87) WIPO No: **WO09/033219**

(30) Priority Data

(31) Number	(32) Date	(33) Country
2007904927	**2007.09.11**	**AU**

(43) Publication Date: **2009.03.19**
(44) Accepted Journal Date: **2014.03.27**

(71) Applicant(s)
Smart Internet Technology CRC Pty Ltd

(72) Inventor(s)
Apted, Trent

(74) Agent / Attorney
Griffith Hack, GPO Box 4164, Sydney, NSW, 2001

(56) Related Art
US 2007/0124699 A1
US 8111239 B2
US 2006/0125799 A1
US 2005/0052427 A1

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) **World Intellectual Property Organization**
International Bureau

(43) **International Publication Date**
19 March 2009 (19.03.2009)

PCT

(10) **International Publication Number**
WO 2009/033219 A1

(51) **International Patent Classification:**
G06F 3/03 (2006.01) ***G06F 3/042*** (2006.01)
G06F 3/041 (2006.01)

(21) **International Application Number:** PCT/AU2008/001345

(22) **International Filing Date:** 11 September 2008 (11.09.2008)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
2007904927 11 September 2007 (11.09.2007) AU

(71) **Applicant** *(for all designated States except US)*: **SMART INTERNET TECHNOLOGY CRC PTY LTD** [AU/AU]; Bay 8, Locomotive Workshop, Australian Technology Park, Eveleigh, New South Wales 1430 (AU).

(72) **Inventor; and**
(75) **Inventor/Applicant** *(for US only)*: **APTED, Trent** [AU/AU]; 6/1-3 See Street, Kingsford, New South Wales 2032 (AU).

(74) **Agent: GRIFFITH HACK**; Level 29, Northpoint, 100 Miller Street, North Sydney, New South Wales 2060 (AU).

(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, 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, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(54) **Title:** A SYSTEM AND METHOD FOR MANIPULATING DIGITAL IMAGES ON A COMPUTER DISPLAY

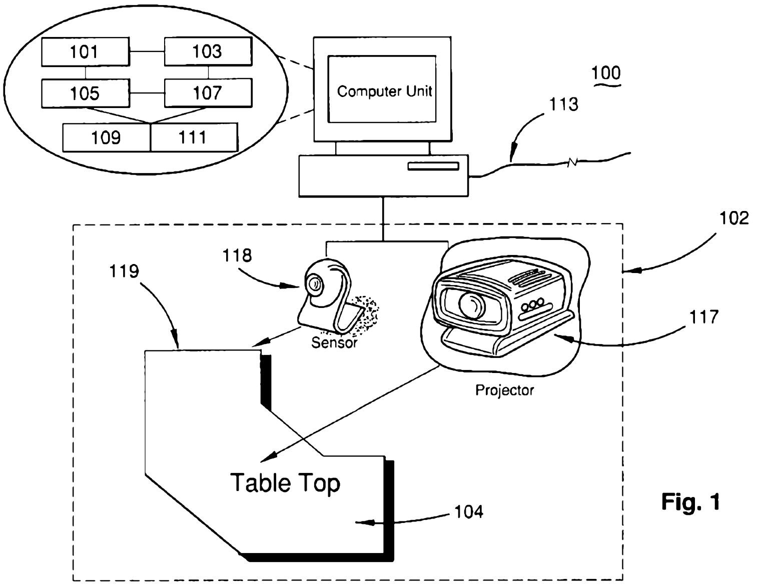

Fig. 1

(57) **Abstract:** The present invention provides a user interface for manipulating digital images. The interface includes a display arranged to display at least one digital image. The at least one image has a front side and at least one opposing side displayable to a user, the front and each one of the opposing sides having at least one detection portion. The detection portion is arranged to detect a manipulation request by the user and, on detection of the request, initiate a routine whereby a displayed side is rotated from view and at least one opposing side is displayed.

A SYSTEM AND METHOD FOR MANIPULATING DIGITAL IMAGES ON A COMPUTER DISPLAY

Technical Field

The present invention relates to a system and method for manipulating digital images on a display of a computing device.

Background of the Invention

Computer users often find the process of inputting and manipulating computer data to be tedious and difficult. One reason for this is that computer interfaces are very artificial and require complex operations to achieve a desired outcome.

Most computer users will, over time, become familiar with any interface. However, there remain those users who experience difficulty irrespective of how much interaction they have with the interface. For example, computer users who have had little training with computers, or those who access their computer infrequently, can find it difficult to remember what input is required to have the computer perform a desired task. This is particularly the case with manipulating objects such digital images which are displayed on the computer screen. Digital images often have many layers of abstraction, requiring the user to work with many layers and be very coordinated and precise with the computer interface in order to manipulate the image accurately and correctly.

Summary of the Invention

In a first aspect the present invention provides a user interface for manipulating digital images comprising a display arranged to display at least one digital image, the at least one image having a front side and at least one opposing side displayable to a user, the front and each one of the opposing sides having at least one detection portion, wherein the detection portion is arranged to detect a manipulation request by the user and, on detection of the request, initiate a routine whereby a displayed side is rotated from view and at least one opposing side is displayed.

In an embodiment of a user interface the detection portion is a defined segment of one of the front and each one of the opposing sides.

In an embodiment of a user interface the rotation occurs only if the manipulation request is sustained by the user during the execution of the routine.

In an embodiment, the routine provides feedback to the user during the execution of the routine.

In an embodiment, the feedback is an animation of the rotation of the image.

In an embodiment, the animation depicts the rotation of the image on an axis that is perpendicular to the surface

In an embodiment, the manipulation request is one of a hand gesture and a movement of a stylus, pen or electronic peripheral.

In an embodiment, the manipulation request comprises a user input including one of a hand gesture, movement of a stylus, pen and electronic peripheral input to the user interface.

In an embodiment the manipulation request is a rotation movement performed by the user input.

In an embodiment each one of the opposing sides may display a representation of a digital object.

In an embodiment the digital object is an audio file.

In an embodiment the representation is an icon associated with the digital object.

In accordance with a second aspect the present invention provides a method for manipulating digital images comprising: the steps of displaying at least one digital image on a display, rendering the at least one image to have a front side and at least one opposing side, displaying the front side on the display, placing at least one detection portion on the front and each one of the opposing sides, detecting a manipulation request by the user with the detection portion and, on detection of the request, initiate a routine whereby a displayed side is rotated from view and at least one opposing side is displayed.

In accordance with a third aspect the present invention provides a system for manipulating digital images comprising a computer unit, the computer unit having an interface according to the first aspect.

In accordance with a fourth aspect the present invention provides a computer program comprising at least one instruction for controlling a computer system to implement a user interface according to the first aspect.

In accordance with a fifth aspect the present invention provides a computer program comprising at least one instruction for controlling a computer system to implement a method according to the second aspect.

In accordance with a sixth aspect the present invention provides a computer readable medium providing a computer program in accordance with any one of the fourth or fifth aspects.

Brief Description of the Drawings

Notwithstanding any other forms which may fall within the scope of the present invention, a preferred embodiment will now be described, by way of example only, with reference to the accompanying drawings in which:

Fig. 1 is a schematic of a computer system utilised to implement an embodiment of the present invention;

Figs. 2a and 2b are flow diagrams showing method steps for flipping and attaching objects to a digital image, in accordance with an embodiment of the present interface; and

Figs. 3A to 3D are a series of screen shots of a flipping animation, in accordance with an embodiment of the present invention.

Detailed Description

With reference to Fig. 1, there is shown a computing system 100 arranged to implement a method in accordance with an embodiment of the present invention. The system 100 is in the form of a personal computer (hereafter "tabletop computer") including a surface-mount display 102. The display 102 is arranged to display a user interface 104 and receive user input for manipulating digital images and other information displayed thereon. The digital images have a front side and at least one opposing side displayable to a user of the tabletop computer 100. The front and opposing sides have at least one detection portion which is arranged to detect a manipulation request by the user. On detection of the request, the tabletop computer 100 initiates a routine whereby a currently displayed side of the image is rotated from view and at least one opposing side is displayed.

To achieve the aforementioned functionality, the tabletop computer 100 comprises computer hardware including a motherboard 101, central processing unit 103, read only memory (ROM) 105, random access memory (RAM) 107, and input/output devices such as disk drives 109, Ethernet ports 111, etc. There may also be provided a plurality of communication links 113 which connect to one or more other computing devices, such as servers, personal computers, terminals, wireless or handheld computing devices etc. At least one of the plurality of communications links may be connected to an external computing network through a telephone line or other type of communications link.

The tabletop computer 100 also includes a display 102 including a projector 117 connected to the tabletop computer 100. The projector 117 projects the user interface 104 onto a surface 119. The display 102 also includes a sensor 118 capable of detecting a user input for manipulating objects displayed on the user interface 104. In an embodiment, the user input may comprise hand movements (or gestures) and/or input from a stylus or pen. In addition to the hardware, the tabletop computer 100 includes an operating system (such as the Microsoft Windows™ XP operating system, which is provided by Microsoft Corporation) that resides on the hard disk and which co-operates with the hardware to provide an environment in which the software applications can be executed. In this regard, the tabletop computer 100 is loaded with a detection module which detects manipulation requests input by a user.

In the embodiment described herein, the digital image is in the form of a digital photograph having two displayable sides, akin to a physical photograph. Manipulation requests include, among others, requests to move or drag the image from one part of the user interface 104 to another, as well as requests to "flip" the digital image so that another side of the image may be presented to the user. To carry out a flipping action, there is provided a flipping module which co-operates with the user interface 104 and the sensor 118. Responsive to receiving a flipping request, the flipping module carries out an animation process whereby the digital image is animated on the display as if it were rotated from one view such that the displayed view is hidden and the opposing side of the digital image is displayed. This rotation may be on an axis perpendicular to any straight edge of the digital image, or it may be a rotation that may be a combination on both the perpendicular and parallel axis to any straight edge of the digital image. This rotation provides a visual effect akin to a physical card, photo or object being turned over by hand. An attachment module is also provided by the tabletop computer 100 which allows data objects, including, but not limited to files such as audio or text files, or other digital data to be associated with the flipped side of the digital image. In such an embodiment, the user can, for example, attach a media file to the digital image which can be executed once the image has been flipped. This functionality provides the user with a familiar experience as it allows the user to virtually flip an image, using the same action and skill as would be required to interact with physical objects.

Referring to Fig. 2a, there is provided a flow chart which outlines a methodology in accordance with an embodiment of the present invention. At step 200, the software application in accordance with an embodiment of the invention detects (periodically) the presence of a flip request. If a flip request is detected, step 202 is invoked and a flipping action is initiated. Once the flipping action at step 202 has been initiated, a determination is made as to whether the flipping action is sustained (step 204). If the flipping action is sustained, step 206 is invoked and the flipping action is completed. After the flipping action is completed, the method returns to awaiting the next flip request (step 200).

Conversely, if the flipping action is not sustained, the methodology progresses to step 208 where a partial flipping action is initiated. Once the partial flipping action is completed, the methodology returns to step 200, namely awaiting a flip request.

In more detail, the detection module detects a user input request to flip the digital image. The flip request is provided by the user in the form of a hand movement and/or the movement of a stylus, pen or other electronic peripheral device. To detect a flip request, the detection module determines the location of user input touch points received on the user interface 104 by monitoring for such touch points with the sensor 118. The touch points are calculated by determining absolute values of the components of the image coordinates of the touch points. If the detection module determines that one or more touch points are located in the prescribed detection portions (also referred to as "hot spots"), the determination module instructs the flipping module to initiate the flipping action.

The hot spots are configurable based on user behaviour or hardware configuration. In an embodiment, the hot spots include both the top-left corner and bottom-right corner of the digital image. To instruct a flipping action, the user simultaneously touches the hot spots using their thumb and middle finger, and carries out a pinching action such that the thumb and middle finger are brought together. In this manner, the user input is effectively the same action as the action that the user would perform to flip a physical photograph. A similar movement may also be performed by using a stylus or pen, where the user first touches a hot spot using the stylus, and without lifting the stylus, moves the stylus towards the other hot spot, thereby simulating the flipping hand gesture. As a way of illustrating an implementation of this embodiment, without limitation, a listing of code for one method of implementation is provided in Annexure 1.

Still referring to Fig. 2a, the flipping module receives the arguments of pixel coordination (x,y) on the user interface 104 corresponding to the control point used for the flip action 202 and the coordinates are converted using the cached inverse transformation matrices used to position the object being flipped in the virtual environment. The x coordinate is normalised to make it comparable with the y coordinate so as to give equal weighting to determine the initial direction of the flip. As a way of illustrating an implementation of this embodiment, without limitation, a listing of code for one method of implementation is provided in Annexure 2.

The direction of the flip is determined by the location of the initial control point of the digital image. As discussed earlier, each control point has a pixel coordination (x,y). In determining the direction of flip, if the user only touches one control point the absolute values of each x and y components are considered, and the largest component (i.e. the component with the largest distance from the centre of the image) determines the direction of the flip. In the case where the user touches a plurality of points on the digital image, the average of all coordinates of the control points is first calculated and the resultant coordinate is utilised to determine the largest component and therefore the direction of flip. The distance to flip is twice the position component (as points initially extend from -0.5 to 0.5), so the distance lies in the range [-1.0, 1.0]. The angle to flip is the arccosine of the distance. The module maintains the state to continue a flip in the same direction once initiated, and to determine whether the module is flipping to the reverse side, or back to the "front" side. The flipping of the image itself occurs in the draw procedure and this communicates with the user interface to animate the flipping process. Unlike traditional computer user interfaces, whereby the feedback is lacking, delayed, unrealistic or non existent, this function allows a feedback to be provided to the user thereby enhancing the user's experience in using the interface. In other words, the provision of feedback by detecting the direction of flipping or the cancellation of a flip assists in further teaching the user in using the interface by providing real time feedback for the user, much like the user's physical interaction with a real life environment. As a way of illustrating an implementation of this embodiment, without limitations, a listing of code for one method of implementation is provided in Annexure 3.

Referring now to Fig. 2b, there is described a methodology which an object may be attached to the "back" (i.e. other side) of a flipped image. At step 210, a determination is made as to whether a user has initiated an attachment request. If an attachment request is initiated, the system proceeds to determine whether the image has been flipped (step 212). If the image has been flipped, the object is attached (step 214). Thereafter, the methodology returns to the periodic determination of whether an attachment request has been initiated.

Conversely, if the image is not flipped, an attachment cannot be made to the object and the methodology returns to step 210, namely a periodic determination of whether a new attachment request has been made.

In more detail, whenever a digital object is moved on the user interface 104, a checklink function selects the object directly below the touch point on the object being moved. If there is a digital image present and the image has been flipped by the flip module, then an attachment occurs between the digital data and the image. Once the digital objects are attached or released from an attachment, the objects are subject to a layout algorithm. Two example layout algorithms are a "FlowLayout" and "Space-Filling Thumbnails" layout, but potentially any layout can be used including a "Stacked Layout" which stacks objects of the same "type" in a pile. This embodiment further enhances the user's experience with the user interface 104 as it allows the user to actively add attachments such as messages, information, clips, descriptions or other items to the images. The attachments are easily accessible, removable and manageable even for those without computing skills, as the graphical display and the structure of the presentation in the embodiment provides an easy file management of the objects. Upon the addition or removal of any attachments, the user can simply "flip" the image over again and thereby allow for easy storage or filing. The entire process of image flipping, object attachment and filing is, on traditional computer interface, a difficult and cumbersome task. However in this presently described embodiment, the entire process can be achieved with normal intuitive actions that a normal user, (particularly a user unfamiliar with computers) would be able to complete with little training. As a way of illustrating an implementation of the embodiment, without limitation, a listing of code for one method of implementation is provided in Annexure 4.

EXAMPLE

With reference to Figs. 3A to 3D, a user chooses the image 300 that they wish to manipulate. The user may then touch the digital image 300 projected onto the table as if the user were manipulating a real object. The user may bring the digital image closer or further away, and move the image to any new position on the user interface 102.

To flip the digital image 300, the user simulates a flipping gesture (as described above) over the digital image 300 using the prescribed areas 302. In an embodiment, this is achieved by the user placing two fingers on the image, one each in adjacent photo corners. The user than drags two fingers across the image to flip it, in one embodiment, a "corner" triangle is rendered on the edges of the digital image and a user's finger or stylus or pen when placed in a "corner" triangle will begin a flip action, and as it is dragged over the image, the image is seen to rotate on a horizontal axis, perpendicular to the straight line joining the two edges. If the flipping movement from the user is withdrawn less than half way during the flipping of the image, the image will return to its previous state as the flipping movement is cancelled. During the flipping action, the image 300 may be displayed as an animation of a simulated turning movement as if they were physical photographs placed on a table (see Figs. 3B & 3C).

In Fig. 3D, the flipped image is rendered with a semi transparent white colouring 304, although in another embodiment, the rendering can be completed in different graphical details as to portray the image as having been flipped 306. The image once flipped behaves differently when other items on the display are dragged over it. The user may attach various digital data to the image once it is flipped. Such data may include, but not limited to audio files, texts, multimedia files or other digital data. The user can initiate an attachment of digital data by selecting the digital data object and positioning this object over a flipped image. This can be done by either the use of a menu or with a dwell motion when the object is over the flipped image. Once the attachment is made a circle with a number appears on the back of the flipped image.

The user may record pieces of audio objects and attach these objects onto the image, which will be displayed as a circle on the image. The audio links may be deleted by dragging the circle out of the image, thereby breaking this link.

Although the display described above was in the form of a surface displaying a projected image, it will be understood that the display can take any form. For example, the display may take the form of a touch-screen display or may simply be a conventional display where inputs to the display are received by traditional inputs such as a keyboard and mouse.

Although not required, the embodiments described with reference to the Figures can be implemented via an application programming interface (API) or as a series of libraries, for use by a developer, and can be included within another software application, such as a terminal or personal computer operating system or a portable computing device operating system. Generally, as program modules include routines, programs, objects, components, and data files that perform or assist in the performance of particular functions, it will be understood that the functionality of the software application may be distributed across a number of routines, objects and components to achieve the same functionality as the embodiment described and the broader invention claimed herein. Such variations and modifications are within the purview of those skilled in the art.

ANNEXURE 1

Procedure for deciding whether to initiate a flip

```
bool Image::flipSelected() {
    GLfloat x = fabsf(currMPos.x), y = fabsf(currMPos.y);
    return !inBlackhole &&
        (y - x > RConfig::CORNER_PROP ||
         x - y > 0.5*aspect - 0.5 + RConfig::CORNER_PROP);
}
```

ANNEXURE 2

Procedures for flipping an object, given the current contact position on the screen

```
bool Image::flipto(int screen_x, int screen_y) {
    P3<GLfloat> avgPos;
    //determine in texture coords where the control point is
    setObject(avgPos, screen_x, screen_y, true);
    avgPos.x /= aspect; //normalise between 0 and 1
    //(we always flip along the bigger direction)

    float cosval, dist;
    if (!flipping) { //not already flipping
        xflip = fabs(avgPos.y) > fabs(avgPos.x);
        dist = xflip ? avgPos.y*2.0 : avgPos.x*2.0;
        backflip = dist < 0;
        wasflipped = flipped;
        flipping = true;
    } else {
        dist = xflip ? avgPos.y*2.0 : avgPos.x*2.0;
    }

    //now the flipping state toggles if the distance has changed in sign
    flipped = wasflipped != (backflip != dist < 0);

    //determine the angle to rotate the image
    cosval = fabs(dist);
    if (cosval < 1.0) {
        flipangle = acosf(cosval); //between 0 and 90

        //now, if we started on the 'wrong' side, we rotate the other way
        //we also do a magical flip at the centre if flipped
        if (xflip == backflip)
            flipangle *= -1.0;

        //now we are here, whenever we have a change from
        // the start, we want to subtract from 180 degrees
        if (flipped != wasflipped)
            flipangle = -flipangle;

    } else {
        //dist is > 1.0, so touch point moved away — don't flip
        flipangle = 0.0;
    }

    pc.makeDirty();
    return true;
}
```

ANNEXURE 3

```
void Image::drawPartialFlip() {
    glPushMatrix();
    glRotatef(flipangle*RAD2DEG,
              xflip ? 1.0 : 0.0,
              xflip ? 0.0 : 1.0,
              0.0);
    if (flipped) {
        drawReverse();
        //also draw the back if we are flipping
        if (flipping) {
            glRotatef(180,
                      xflip ? 1.0 : 0.0,
                      xflip ? 0.0 : 1.0,
                      0.0);
            glCallList(*model);
        }

    } else {
        drawTexture();
        if (flipping) {
            glRotatef(180,
                      xflip ? 1.0 : 0.0,
                      xflip ? 0.0 : 1.0,
                      0.0);
            drawReverse(); /* if rendering ?? */
        }
    }
    glPopMatrix();
}
```

ANNEXURE 4

Procedure, called each time an object is moved, to check whether it should be attached to another object

```
void Resource::checkLink() {
    // get the topmost object in the environment at the 'touch' location
    P3<int> scr(pc.getScreen() + clickPos);
    PResource top = env->selectTop(scr.x, scr.y, this, true);

    PResource cur = 0; // current parent, if any
    if (link)
        cur = link->getParent();

    if (cur != top && canDetach()) { //if changed
        // remove old dependencies, if any
        removeDependenciesOnParent();

        if (top && top->isFlipped()) {
            if (canLink(top) && top->canLink(this)) {
                Layout *lay = top->getChildLayout();
                /* if new parent has a layout manager, add to that */
                if (lay) {
                    lay->add(this);
                    lay->layout();
                    link = lay;
                } else { /* else use 'basic' linking method */
                    link = PDAdapter::link(this, top);
                }
                pushFlasher(top); // give feedback
            }
        }
    }
}
```

CLAIMS:

1. A user interface for manipulating digital images comprising:

a display arranged to display at least one digital image, the at least one image having a front side and at least one opposing side displayable to a user, the front and each one of the opposing sides having at least one detection portion, wherein the detection portion is arranged to detect a manipulation request by the user and, on detection of the request, initiate a routine whereby a displayed side is rotated from view and at least one opposing side is displayed, wherein different sides of the image have different interaction behaviours with digital objects and upon receiving a request to attach a digital object to the image the object is attached in dependence on the image having been rotated.

2. A user interface according to Claim 1, wherein the detection portion is a defined segment of one of the front and each one of the opposing sides.

3. A user interface according to Claim 1 or 2, wherein rotation occurs only if the manipulation request is sustained by the user during the execution of the routine.

4. A user interface according to Claim 1, 2 or 3, wherein the routine provides feedback to the user during the execution of the routine.

5. A user interface according to Claim 4, wherein the feedback is an animation of the rotation of the image.

6. A user interface according to Claim 5, wherein the animation depicts the rotation of the image on an axis that is perpendicular to the surface.

7. A user interface according to any one of the preceding claims, wherein the manipulation request comprises a user input including one or more of a hand gesture and movement of a stylus, pen or electronic peripheral input to the user interface.

8. A user interface according to Claim 7, wherein the manipulation request is a rotation movement performed by the user.

9. A user interface according to any one of the preceding claims, wherein each one of the opposing sides may display a representation of a digital object.

10. A user interface according to Claim 9, wherein the digital object is an audio file.

11. A user interface according to Claim 10, wherein the representation is an icon associated with the digital object.

12. A method for manipulating digital images comprising the steps of:

displaying at least one digital image on a display, rendering the at least one image to have a front side and at least one opposing side, displaying the front side on the display, placing at least one detection portion on the front and each one of the opposing sides, detecting a manipulation request by the user with the detection portion and, on detection of the request, initiate a routine whereby a displayed side is rotated from view and at least one opposing side is displayed, wherein different sides of the image have different interaction behaviours with digital objects and upon receiving a request to attach a digital object to the image attaching the object in dependence on the image having been rotated.

13. A method according to Claim 12, wherein detection portion is a defined segment of at least one of the front and each one of the opposing sides of the image.

14. A method according to Claim 13, wherein rotation occurs only if the manipulation request is sustained by the user during the execution of the routine.

15. A method according to Claim 12 or 14, wherein feedback is provided by the routine to the user during the execution of the routine.

16. A method according to Claim 15, wherein the feedback is an animation of the rotation of the image.

17. A method according to Claim 16, wherein the animation depicts the rotation of the image on an axis that is perpendicular to the surface.

18. A method according to any one of the preceding Claims 12 to 17, wherein the manipulation request comprises a user input including one or more of a hand gesture and movement of a stylus, pen or electronic peripheral input to the user interface.

19. A method according to Claim 18, wherein the manipulation request is initiated by a rotation movement performed by the user input.

20. A method according to any one of the preceding Claims 12 to 19, wherein each one of the opposing sides may display a representation of a digital object.

21. A method according to Claim 20, wherein the digital object is an audio file.

22. A method according to Claim 21, wherein the digital object is an image file.

23. A method according to either one of Claims 21 or 22, wherein the representation is an icon associated with the digital object.

24. A system for manipulating digital images comprising a computer unit, the computer unit implementing a user interface according to any one of Claims 1 to 11.

25. A computer program comprising at least one instruction for controlling a computer system to implement a method according to any one of Claims 1 to 11.

26. A computer program comprising at least one instruction for controlling a computer system to implement a method according to Claims 12 to 23.

27. A computer readable medium providing a computer program in accordance with Claim 25.

28. A computer readable medium providing a computer program in accordance with Claim 26.

29. A user interface for manipulating digital images substantially as herein described with reference to the accompanying drawings.

30. A method for manipulating digital images substantially as herein described with reference to the accompanying drawings.

31. A system for manipulating digital images substantially as herein described with reference to the accompanying drawings.

32. A computer program for manipulating digital images substantially as herein described with reference to the accompanying drawings.

33. A computer readable medium providing a computer program for manipulating digital images substantially as herein described with reference to the accompanying drawings.

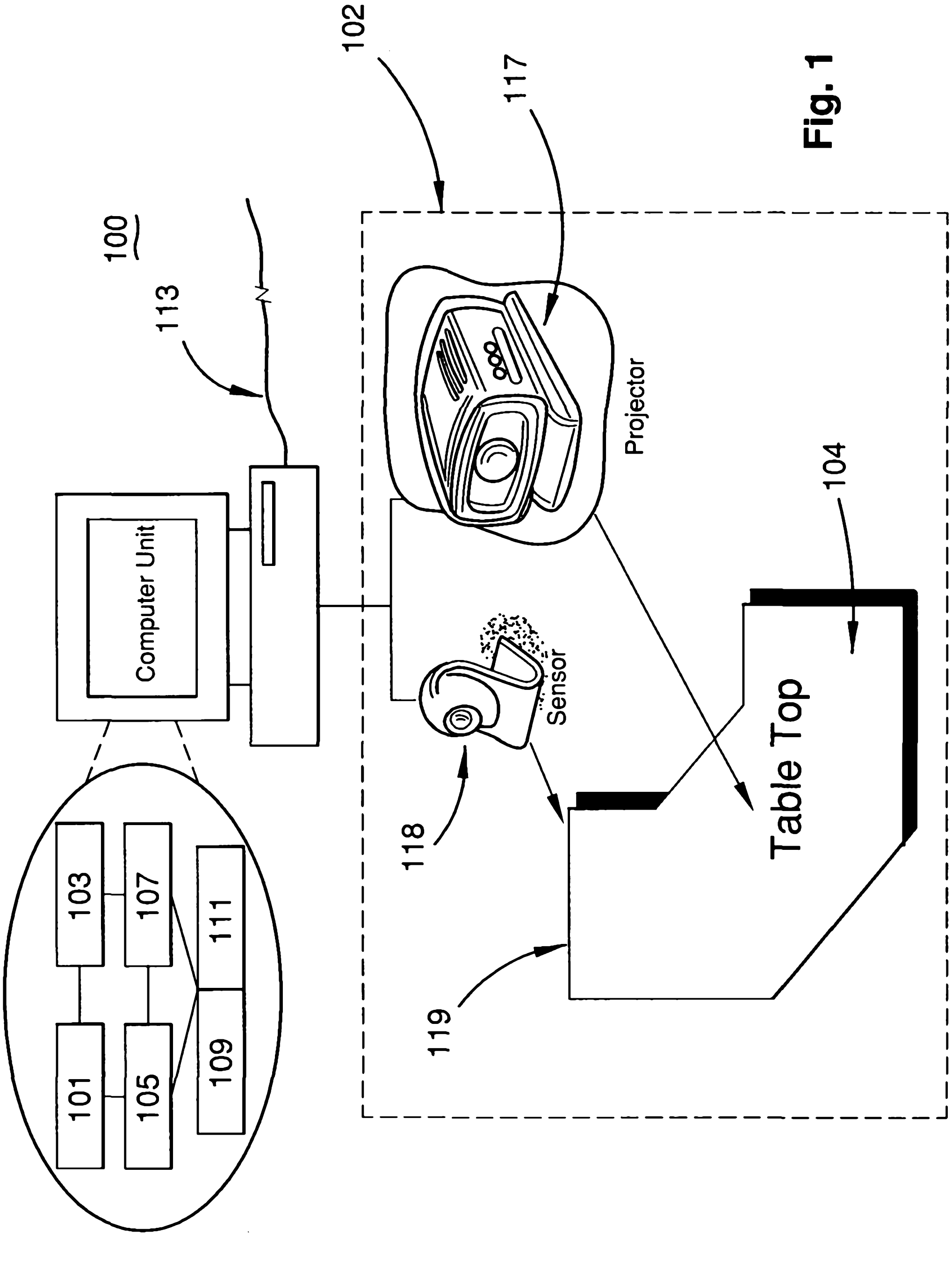
100
113
102
117
Projector
Computer Unit
Sensor
118
104
Table Top
119
101
103
105
107
109
111

Fig. 1

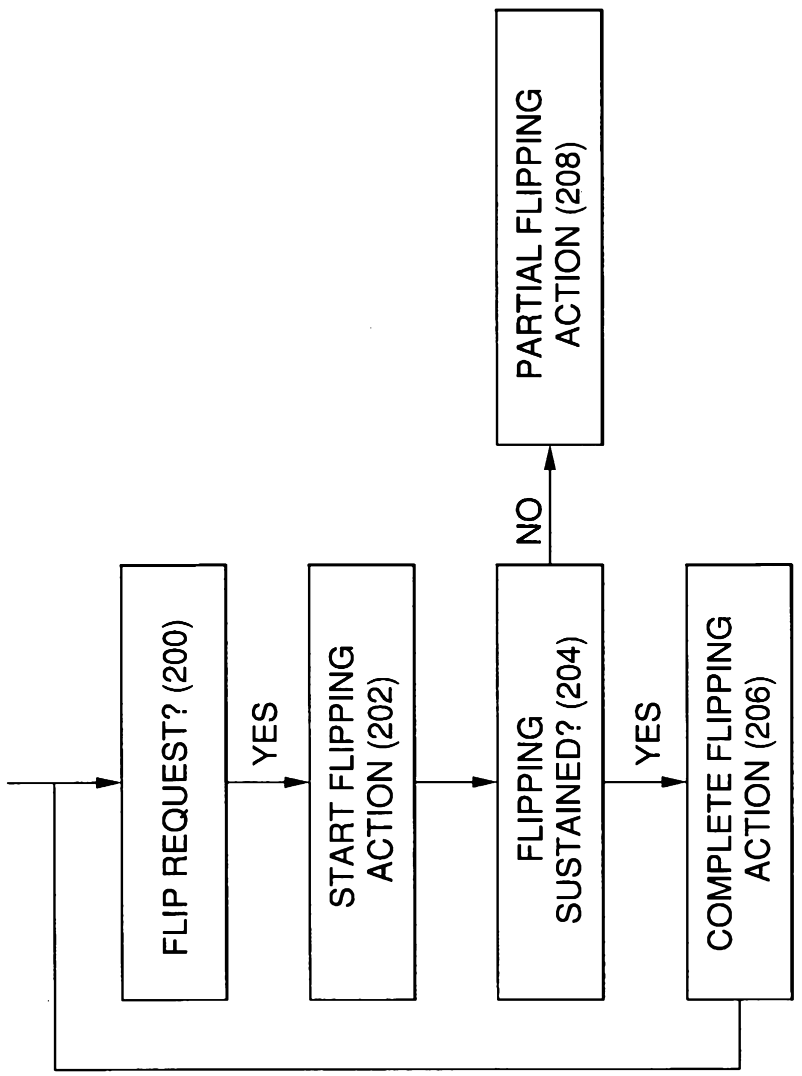
FLIP REQUEST? (200)
YES
START FLIPPING ACTION (202)
FLIPPING SUSTAINED? (204)
YES
COMPLETE FLIPPING ACTION (206)
NO
PARTIAL FLIPPING ACTION (208)

Fig. 2a

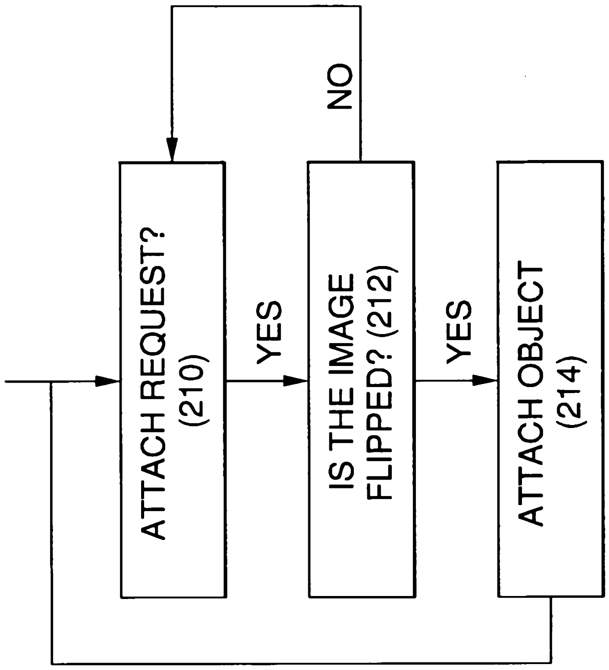
ATTACH REQUEST?
(210)
YES
IS THE IMAGE
FLIPPED? (212)
NO
YES
ATTACH OBJECT
(214)

Fig. 2b

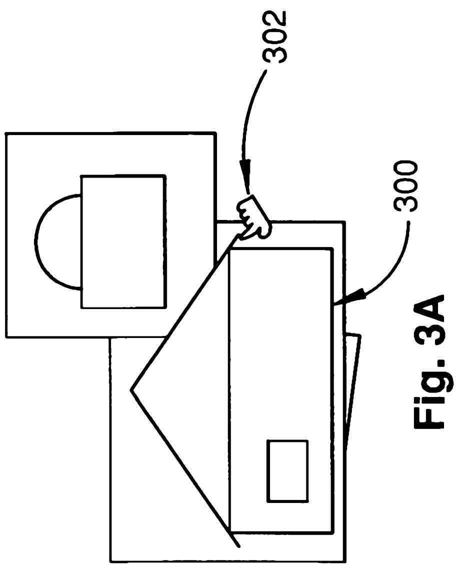

Fig. 3A

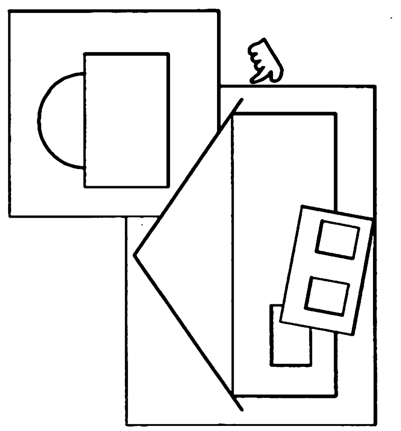

Fig. 3B

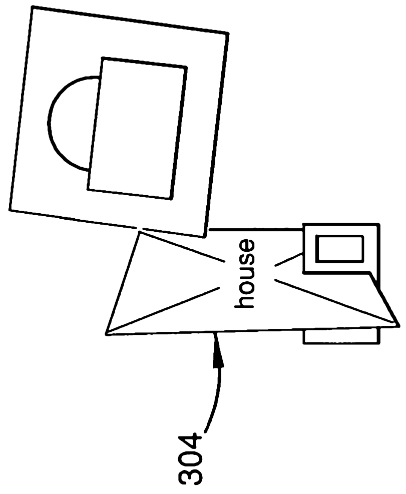

Fig. 3C

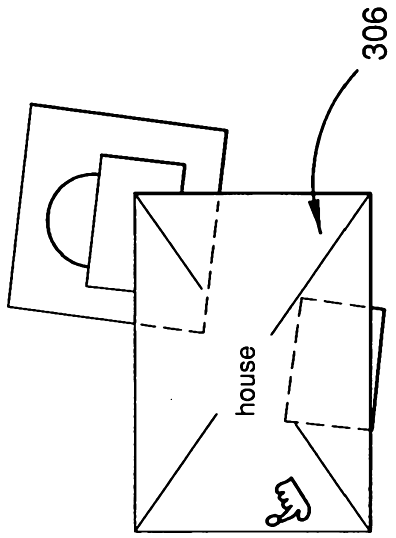

Fig. 3D