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

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

(43) **International Publication Date**
16 December 2010 (16.12.2010)

PCT

(10) **International Publication Number**
WO 2010/144050 A1

(51) **International Patent Classification**:
G06F 3/01 (2006.01) ***G06K 9/00*** (2006.01)

(21) **International Application Number**:
PCT/SG2009/000203

(22) **International Filing Date**:
8 June 2009 (08.06.2009)

(25) **Filing Language**: English

(26) **Publication Language**: English

(71) **Applicant** *(for all designated States except US)*: **AGENCY FOR SCIENCE, TECHNOLOGY AND RESEARCH** [SG/SG]; 1 Fusionopolis Way, #20-10, Connexis, Singapore 138632 (SG).

(72) **Inventors; and**
(75) **Inventors/Applicants** *(for US only)*: **MANDERS, Corey, Manson** [CA/SG]; IPTO, Institute for Infocomm Research, 1 Fusionopolis Way, #21-01, Connexis, South Tower, Singapore 138632 (SG). **FARZAM, Farbiz** [IR/SG]; IPTO, Institute for Infocomm Research, 1 Fusionopolis Way, #21-01, Connexis, South Tower, Singapore 138632 (SG). **TANG, Ka Yin, Christina** [MY/SG]; IPTO, Institute for Infocomm Research, 1 Fusionopolis Way, #21-01, Connexis, South Tower, Singapore 138632 (SG). **CHUA, Gim, Guan** [SG/SG]; IPTO, Institute for Infocomm Research, 1 Fusionopolis Way, #21-01, Connexis, South Tower, Singapore 138632 (SG).

(74) **Agent**: **ELLA CHEONG SPRUSON & FERGUSON (SINGAPORE) PTE LTD**; P.O. Box 1531, Robinson Road Post Office, Singapore 903031 (SG).

(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, 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, MK, 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).

Declarations under Rule 4.17:

— *as to non-prejudicial disclosures or exceptions to lack of novelty (Rule 4.17(v))*

Published:

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

(54) **Title**: METHOD AND SYSTEM FOR GESTURE BASED MANIPULATION OF A 3-DIMENSIONAL IMAGE OF OBJECT

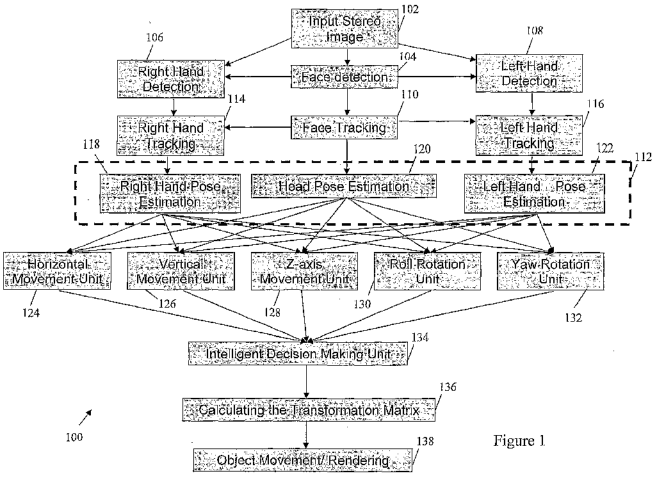


Figure 1

(57) **Abstract**: A method and system for gesture based manipulation of a 3-dimensional image or object. The method comprises the steps of tracking the user's face, right and left hands using stereo image processing; defining a 3-dimensional gesture coordinate system based on the tracked user's face; determining the user's gesture based on the tracked movement of the user's right and left hands within the 3-dimensional gesture coordinate system; and manipulating the image or object based on the detected user's gesture.

WO 2010/144050 A1

Method And System For Gesture Based Manipulation Of A 3-Dimensional Image Or Object

FIELD OF INVENTION

The present invention relates broadly to a method and system for gesture based manipulation of a 3-dimensional image or object, and to a data storage medium comprising computer code means for instructing a computing device to execute a method of gesture based manipulation of a 3-dimensional image or object.

BACKGROUND

Current gesture technology is typically used to control objects restricted to two-dimensional movement. In [Kai Nickel, Rainer Stiefelhagen, "*Pointing Gesture Recognition based on 3Dtracking of Face, Hands and Head Orientation*", Proceedings of the Fifth International Conference on Multimodal Interfaces, Vancouver, Canada, Nov. 5-7, 2003], a gesture system is presented that uses both face and hand positions for the interaction. The face and hand positions are used to construct a vector originating from the hand at which the user is pointing. This type of control only allows for two degrees of freedom, and therefore is incapable of controlling an object in three-dimensional space.

Another system that uses face position for gesture interaction is presented in [Narcio C. Cabral, Carlos H. Morimoto, Marcelo K. Zuffo, "*On the usability of gesture interfaces in virtual reality environments*", CLIHC'05, pp. 100-108, Cuernavaca, Mexico, 2005]. The 2D position of a user's hands and face are detected using a single camera. The left hand is used for several operations that are activated depending on the position of the left hand relative to head. The 2D position of the hands and face in the images are identified by the coordinate of their center of mass, defined by X, Y. For example, the North position of the left hand is used to switch the mode that controls the cursor/screen behavior, for example, switching the navigation mode such as translations or rotation in a 3D exploration application. The right hand is basically used for pointing, i.e., to control the position of the cursor. Thus, all the gesture interactions are still in 2 dimensions, and activation of 3-dimensional control requires un-intuitive separate movements of the left and right hands.

In [R. Voyles, J. Morrow, P. Khosia, "Gesture-Based Programming for Robotics: Human-Augmented Software Adaptation", IEEE Intelligent Systems, Vol 14, No. 6, pp. 22-29, 1999] a system is presented for gesture based 3 dimensional control, however the user must wear a special apparatus (for example a glove containing sensors). In [T. Grossman, D. Wigdor, R. Balakrishnan, "Multi-Finger Gestural Interaction with 3-D Volumetric Displays", in the proceeding of User Interface Software and Technology (UIST), 2004], finger tracking is used to control a volumetric display. The user has infrared reflectors attached to his fingers and the system uses a tracking system to acquire the position of the user's fingers.

Other systems deal with visual manipulation in virtual environments, for example O'Hagan and Zelinsky in [R. O'Hagan and A. Zelinsky, "Visual gesture interfaces fro virtual environments", User Interface Conference (AUIC 2000), pp. 73-80, 2000] used stereo vision for hand tracking and gesture interaction for manipulation of virtual objects (for example moving and resizing virtual objects in a virtual scene) in 3 dimensions. However, to use the system proposed in this work, a single hand must be in front of a known background.

SoftKinetic [http://www.softkinetic.net/Public/eng/Technology/] uses depth-sensing technology along with a Software Development Kit (SDK). The depth information is transformed into a collection of spheres, giving the segmented depth information mass. The system employs determining the Silhouette of a person or object in front of a depth-sensing camera which is invariant to the background. The technology also allows for mass estimation of the 3D object being sensed. The system uses a time-of-flight based depth-sensing camera which is invariant to the background. The technology also allows for mass estimation of the 3D object being sensed. However, no tracking and interpretation of the user's hands interaction is described.

A need therefore exists to provide a method and system for gesture based manipulation of a 3-dimensional image or object which seek to address at least one of the above mentioned problems.

SUMMARY

In accordance with a first aspect of the present invention there is provided a method of gesture based manipulation of a 3-dimensional image or object, the method comprising the steps of tracking the user's face, right and left hands using stereo image processing; defining a 3-dimensional gesture coordinate system based on the tracked user's face; determining the user's gesture based on the tracked movement of the user's right and left hands within the 3-dimensional gesture coordinate system; and manipulating the image or object based on the detected user's gesture.

Defining the 3-dimensional coordinate system may comprise defining an origin, an x-axis and a y-axis of the 3-dimensional gesture coordinate system based on the tracked user's face; and defining a z-axis of the 3-dimensional gesture coordinate system based on an optical centre of one camera of a stereo camera system used for the stereo video image processing.

Determining the user's gesture may comprise determining whether both the user's right and left hands are above or below the y-axis for determining an intended movement of the image or object in an upward or downward direction.

The method may further comprise determining an intended upwards or downward movement based on an average y-component of the right and left hands.

Determining the user's gesture may comprise determining whether the user's right and left hands are equidistant from the x-axis for determining an intended movement of the image or object in a right or left direction.

The method may further comprise determining an intended right or left movement based on an average x-component of the right and left hands.

Defining the 3-dimensional coordinate system may further comprise defining a reference position along the z-axis between the origin and the stereo camera.

Determining the user's gesture may comprise determining whether an average z-component of the user's right and left hands is greater than or smaller than the reference position for determining an intended movement of the image or object in a forward or backward direction.

The method may further comprise determining an intended forward or backward movement based on an average z-component of the right and left hands.

Determining the user's gesture may comprise determining whether a z-component of the user's right hand is smaller than the reference position and a z-component of the left hand is greater than the reference position for determining an intended clockwise rotational yaw movement of the image or object.

Determining the user's gesture may comprise determining whether a z-component of the user's left hand is smaller than the reference position and a z-component of the right hand is greater than the reference position for determining an intended counter-clockwise rotational yaw movement of the image or object.

The method may further comprise determining an intended rotational yaw movement based on a sum of the z-components of the right and left hands.

Determining the user's gesture may comprise determining whether a y-component of the user's right hand is below the y-axis and a y-component of the left hand is above the y-axis for determining an intended clockwise rotational roll movement of the image or object.

Determining the user's gesture may comprise determining whether a y-component of the user's left hand is below the y-axis and a y-component of the right hand is above the y-axis for determining an intended counter-clockwise rotational roll movement of the image or object.

The method may further comprise determining an intended rotational roll movement based on a difference of the y-components of the left and right hands.

Tracking the user's face may comprise using marker-less stereo video image processing.

Tracking the user's right and left hands may comprise utilising a joint probability map based on skin colour and depth information.

In accordance with a second aspect of the present invention there is provided a system for gesture based manipulation of a 3-dimensional image or object, the method comprising the steps of means for tracking the user's face, right and left hands using stereo image processing; means for defining a 3-dimensional gesture coordinate system based on the tracked user's face; means for determining the user's gesture based on the tracked movement of the user's right and left hands within the 3-dimensional gesture coordinate system; and means for manipulating the image or object based on the detected user's gesture.

In accordance with a third aspect of the present invention there is provided a data storage medium comprising computer code means fro instructing a computing device to execute a method of gesture based manipulation of a 3-dimensional image or object, the method comprising the steps of tracking the user's face, right and left hands using stereo image processing; defining a 3-dimensional gesture coordinate system based on the tracked user's face; determining the user's gesture based on the tracked movement of the user's right and left hands within the 3-dimensional gesture coordinate system; and manipulating the image or object based on the detected user's gesture.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention will be better understood and readily apparent to one of ordinary skill in the art from the following written description, by way of example only, and in conjunction with the drawings, in which:

Figure 1 shows a schematic drawing illustrating a system structure according to an example embodiment.

Figure 2 illustrates a schematic block diagram of a range compression process in typical cameras which may affect the RGB to HSV computation.

Figure 3 illustrates a schematic block diagram of a range decompression process prior to the HSV computation and further processing in an example embodiment.

Figure 4 shows a flowchart illustrating a method for tracking hands used in a preferred embodiment of the present invention.

Figures 5 (a) to (d) show gestures (or hand positions) of a preferred embodiment used to manipulate a 3-dimensional image or object.

Figures 6 (a) to (d) show other gestures (or hand positions) of a preferred embodiment used to manipulate a 3-dimensional image or object.

Figures 7 (a) and (b) show other gestures (or hand positions) of a preferred embodiment used to manipulate a 3-dimensional image or object.

Figure 8 shows regions that are sampled in another example embodiment implementation for skin colour detection.

Figure 9 shows a flow chart illustrating a method of gesture based manipulation of a 3-dimensional image or object according to an example embodiment.

Figure 10 shows a schematic drawing illustrating a computer system for implementing the method and system of an example embodiment.

DETAILED DESCRIPTION

The described embodiments provide a system for interacting with 3D objects in a 3D virtual environment. In one embodiment, a fiducial marker is placed on a head-mounted display (HMD) to locate the user's exposed facial skin. Using this information, a skin model is built and combined with the depth information obtained from a stereo camera. The information when used in tandem allows the position of the user's hands to be detected and tracked in real time. Once both hands are located, such a system advantageously allows the user to manipulate the object with five degrees of freedom (translation in x, y, and z axis with roll and yaw rotations) in virtual three-dimensional space or in the real world (non-virtual space) using a series of intuitive hand gestures. In another preferred embodiment, face (and hand) detection can be performed without the use of a marker, further improving the usability of the system.

Some portions of the description which follows are explicitly or implicitly presented in terms of algorithms and functional or symbolic representations of operations on data within a computer memory. These algorithmic descriptions and functional or symbolic representations are the means used by those skilled in the data processing arts to convey most effectively the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities, such as electrical, magnetic or optical signals capable of being stored, transferred, combined, compared, and otherwise manipulated.

Unless specifically stated otherwise, and as apparent from the following, it will be appreciated that throughout the present specification, discussions utilizing terms such as "calculating", "determining", "replacing", "generating", "initializing", "outputting", or the like, refer to the action and processes of a computer system, or similar electronic device, that manipulates and transforms data represented as physical quantities within the the computer system into other data similarly represented as physical quantities within the computer system or other information storage, transmission or display devices.

The present specification also discloses apparatus for performing the operations of the methods. Such apparatus may be specially constructed for the required purposes, or may comprise a general purpose computer or other device selectively activated or reconfigured by a computer program stored in the computer. The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general purpose machines may be used with programs in accordance with the teachings herein. Alternatively, the construction of more specialized apparatus to perform the required method steps may be appropriate. The structure of a conventional general purpose computer will appear from the description below.

In addition, the present specification also implicitly discloses a computer program, in that it would be apparent to the person skilled in the art that the individual steps of the method described herein may be put into effect by computer code. The computer program is not intended to be limited to any particular programming language and implementation thereof. It will be appreciated that a variety of programming languages and coding thereof may be used to implement the teachings of the disclosure contained herein. Moreover, the computer program is not intended to be limited to any particular control flow. There are many other variants of the computer program, which can use different control flows without departing from the spirit or scope of the invention.

Furthermore, one or more of the steps of the computer program may be performed in parallel rather than sequentially. Such a computer program may be stored on any computer readable medium. The computer readable medium may include storage devices such as magnetic or optical disks, memory chips, or other storage devices suitable for interfacing with a general purpose computer. The computer readable medium may also include a hard-wired medium such as exemplified in the Internet system, or wireless medium such as exemplified in the GSM mobile telephone system. The computer program when loaded and executed on such a general-purpose computer effectively results in an apparatus that implements the steps of the preferred method.

The invention may also be implemented as hardware modules. More particular, in the hardware sense, a module is a functional hardware unit designed for use with other components or modules. For example, a module may be implemented using discrete electronic components, or it can form a portion of an entire electronic circuit such as an Application Specific Integrated Circuit (ASIC). Numerous other possibilities exist. Those skilled in the art will appreciate that the system can also be implemented as a combination of hardware and software modules.

Figure 1 shows a schematic drawing illustrating the system structure 100 according to an example embodiment. Images captured by an input stereo image unit 102, such as a stereo camera, are provided to a face detection unit 104, as well as right hand and left hand detection units 106, 108. Outputs from the face detection unit 104 are also provided to the right hand and left hand detection units 106, 108, to facilitate right and left hand detection respectively. Additionally, the output from the face detection unit 104 is provided to a face tracking unit 110 for tracking the face location, which serves as the origin of a three dimensional gesture coordinate system used by the system 100 in determining and processing the various gestures in the gesture estimation unit 112, which will be described in more detail below.

Output from the face tracking unit 110 is also provided to right hand and left hand tracking units 114, 116 respectively. In the right and left hand tracking units 114, 116, the face tracking input together with respective inputs from the right and left hand detection units 106, 108 are utilized for tracking the right and left hand respectively.

The gesture estimation unit 112 in this example embodiment comprises three elements, more particular a right hand pose estimation element 118, a head pose estimation unit 120, and a left hand pose estimation unit 122. Each of the estimation units 118, 120, 122 perform pose estimation based on respective inputs from the right hand tracking unit 114, the face tracking unit 110, and the left hand tracking unit 116.

Outputs from each of the pose estimation units 118, 120 and 122 are provided to a set of movement determination units, more particular horizontal movement unit 124, vertical movement unit 126, z axis movement unit 128, roll rotation unit 130, and yaw rotation unit 132. Details of the movement determination units will be described below. Based on respective outputs from the movement determination units 124, 126, 128, 130 and 132, a decision unit 134 determines an intended control function based on the user's estimated/detected gesture. The output from the decision unit 134 is fed to a processing unit 136 for calculating a corresponding transformation matrix, which in turn is used in an object movement/rendering unit 138 to implement the user control.

In a preferred example embodiment, the Hue-Saturation based colour space (HS space) is used to detect the user's hand motion instead of the RGB colour space. Hue defines the dominant colour of an area, while saturation measures the purity of the dominant colour in proportion to the amount of "white" light. The luminance component of the colour space, is not used in the example embodiments as our aim is to model what can be thought of "skin tone", which is more controlled by the chrominance than the luminance components.

Typically, cameras capture images using three colour channels, red, green and blue. If there is access to the raw sensor data from the camera, the values collected are typically linear. Hence, if two images are taken of the same object but the sensor array is exposed to the incoming light for twice as long in the second image, one would expect that the raw values would be twice that of the first exposure. The red, green and blue pixel values observed are the result of the linear sensor values r, g and b with the range compression function, f, applied. These red, green and blue pixel values may respectively be referred to as R = f(r), G = f(g) and B = f(b).

When performing skin detection tasks, a color space transformation from RGB to HSV is usually performed. Thus, the pixel values f(r), f(g) and f(b) are transformed to hue, saturation and intensity component, with the purpose of separating the intensity component in the color space. Once completed, the intensity component will be dropped to allow for intensity differences, while retaining the color information.

Assuming that a face has been successfully detected and that the range compressed pixel values are used, Equations (1) – (4) show the typical output from a camera, with the output consequentially used for image processing tasks. In Equations (1) – (4), max = max(f(r), f(g), f(b)) and min = min(f(r), f(g), f(b)).

In Equation (1), the hue component is calculated for a first image. When f(max) = f(r), f(g) ≥ f(b) and f(min) = f(b), the hue component for the first image is given by Equation (2).

Assuming a second image, which is the same image, but differs in exposure time by the ratio k, Equation (3) gives the hue component for this second image when f(max) = f(r), f(g) ≥ f(b) and f(min) = f(b). In Equation (3), k is the ratio of the exposure time of the second image to the exposure time of the first image. For example, if the exposure time of the first image is 100ms and the exposure time of the second image is 200ms, k = 2. Hence, given a pixel in the first image, f(g), not accounting for noise, the same pixel in the second image will be f(kg).

$$H_1 = \begin{cases} 0 & \text{if max = min} \\ 60^o \times \dfrac{f(g)-f(b)}{f(\max)-f(\min)} + 0^o & \text{if max = f(r) and f(g) ≥f(b)} \\ 60^o \times \dfrac{f(g)-f(b)}{f(\max)-f(\min)} + 360^o & \text{if max = f(r) and f(g)< f(b)} \\ 60^o \times \dfrac{f(g)-f(r)}{f(\max)-f(\min)} + 120^o & \text{if max = f(g)} \\ 60^o \times \dfrac{f(r)-f(g)}{f(\max)-f(\min)} + 240^o & \text{if max = f(b)} \end{cases} \qquad (1)$$

$$H_1 = 60^o \times \frac{f(g)-f(b)}{f(\max)-f(\min)} = 60^o \times \frac{f(g)-f(b)}{f(r)-f(b)} \qquad (2)$$

$$H_2 = 60^o \times \frac{f(kg)-f(kb)}{f(kr)-f(kb)} \qquad (3)$$

H_1 would only be equal to H_2 if f is linear as shown in Equation (4).

$$H_2 = 60^o \times \frac{f(kg)-f(kb)}{f(kr)-f(kb)} = 60^o \times \frac{kf(g)-kf(b)}{kf(r)-kf(b)}$$

$$= 60^o \times \frac{f(g)-f(b)}{f(r)-f(b)}$$

$$= H_1 \qquad (4)$$

Figure 2 illustrates a schematic block diagram of a range compression process 200 which transform images from the RGB space to the HSV space prior to processing. In Figure 2, light rays from a subject 202 passes through the lens 204 of a camera 214. The light rays are then detected by the sensor 206 in the camera 214 to form an image. The image captured by the sensor 206 is subject to sensor noise (nq). Range compression of the image is then carried out in compressor 208. Before the image is output, file compression of the image, for example, JPEG compression may be performed giving rise to image noise (nf). The dynamically range compressed image is then stored, transmitted or processed in unit 210. The image is then transmitted to a display unit 212.

In most display units such as the cathode ray tube, the image is inherently distorted whereby this distortion can be modelled by a non-linear function such as an exponential function. Such a distortion, also known as a non-linear gamma correction, invariably exists in order to maintain backward compatibility with previous types of display units. Although the non-linearity may be adjusted in LCD displays, instead of using this control to correct for the non-linearity, the non-linearity is typically amplified to improve the contrast in most LCD displays. Hence, it is preferable that the camera 214 includes the compressor 208 which applies a range compression function f to the image so as to offset the inherent distortion of the image in the display unit 212. As mentioned before, the hue value for two images having different exposure times would be the same only if f is linear as shown in Equation (4).

However, f is typically not linear, as mentioned above. Because of the non-linearity in the camera's output and that the data recorded from the camera is a non-linear representation of the photometric quantity of light falling on the sensor array, the notion of separating the luminance from the chrominance (which motivates an RGB to HSV type of transformation) is lost. Since the exposure time should simply change the luminance of the observed image and not affect its chrominance, the saturation and the hue components should remain unchanged. The inventors have recognised that this is not the case with the presence of the non-linear range compression function f, as described above. Example embodiments of the invention exploit a linearization of the camera's output data.

Figure 3 illustrates a schematic block diagram of a range compression process 300 according to an embodiment of the present invention. In Figure 3, light rays from a subject 302 passes through the lens 304 of a camera 314. The light rays are then detected by the sensor 306 in the camera 314 to form an image. The image captured by the sensor 306 is subject to sensor noise (nq). Range compression of the image is then carried out in compressor 308. Before the image is output, file compression of the image, for example, JPEG compression may be performed giving rise to image noise (nf). The range of the image is then expanded in the estimated expander 310 before linear processing in unit 312.

In one example, the estimated expander 310 uses the inverse of the range compression function f i.e. f^{-1}, assuming that this inverse exists. f^{-1} is applied to the pixel values prior to the hue and saturation computations. Using this approach, the hue calculation for a first image is as shown in Equation (5).

$$H_1 = \begin{cases} 0 & \text{if max = min} \\ 60^o \times \dfrac{f^{-1}(f(g)) - f^{-1}(f(b))}{f^{-1}(f(\max)) - f^{-1}(f(\min))} + 0^o & \text{if max = f(r) and f(g) \geq f(b)} \\ 60^o \times \dfrac{f^{-1}(f(g)) - f^{-1}(f(b))}{f^{-1}(f(\max)) - f^{-1}(f(\min))} + 360^o & \text{if max = f(r) and f(g)< f(b)} \\ 60^o \times \dfrac{f^{-1}(f(g)) - f^{-1}(f(r))}{f^{-1}(f(\max)) - f^{-1}(f(\min))} + 120^o & \text{if max = f(g)} \\ 60^o \times \dfrac{f^{-1}(f(r)) - f^{-1}(f(g))}{f^{-1}(f(\max)) - f^{-1}(f(\min))} + 240^o & \text{if max = f(b)} \end{cases} \tag{5}$$

Assuming that f(max) = f(r), f(g) ≥ f(b) and f(min) = f(b), H_1 is calculated according to Equation (6).

$$H_1 = 60^o \times \frac{f^{-1}(f(g)) - f^{-1}(f(b))}{f^{-1}(f(r)) - f^{-1}(f(b))} = 60^o \times \frac{g-b}{r-b} \tag{6}$$

In a second image with a different exposure time whereby k is the ratio of the second image's exposure time to the first image's exposure time, the hue component of the second image is given by Equation (7). With the introduction of the inverse function f^{-1}, H$_1$ is equal to H$_2$ i.e. the hue components of two images with different exposure times are the same.

$$\begin{aligned} H_2 &= 60^o \times \frac{f^{-1}(f(kg)) - f^{-1}(f(kb))}{f^{-1}(f(kr)) - f^{-1}(f(kb))} \\ &= 60^o \times \frac{kg - kb}{kr - kb} \\ &= 60^o \times \frac{k(g-b)}{k(r-b)} \\ &= 60^o \times \frac{g-b}{r-b} \\ &= H_1 \end{aligned} \qquad (7)$$

Similarly, Equation (8) shows the saturation component for an image when the inverse of the range compression function f, i.e. f^{-1}, for each pixel is used.

$$S_1 = \begin{cases} 0 & \text{if } f^{-1}(f(\max))=0 \\ \left[\dfrac{f^{-1}(f(\max)) - f^{-1}(f(\min))}{f^{-1}(f(\max))} \right] * \max Value & \end{cases} \qquad (8)$$

It must be noted that for light intensity invariance in skin detection tasks, the intensity calculation is not required and the intensity component is dropped after the RGB to HSV computation. The dimensionality of the colour space of the original RGB image hence reduced from I^3 to R^2 by dropping the intensity component V.

Equations (6) – (8) show that the saturation and hue components remain the same for two images with different exposure times after the estimated expander 310 is included prior to the computation of the hue and saturation components. This in turn shows that the expansion of the rearrange compression function i.e. inclusion of the inverse function f^{-1} and the linearization of the pixels to be brought to photometric values can effectively separate the luminance component from the chrominance in the RGB to HSV transformation. With the luminance component effectively separated from the chrominance, the consequent colour analysis can be advantageously simplified in the example embodiments.

Figure 4 shows a flowchart illustrating a method 400 for tracking hands according to a preferred embodiment of the present invention. The system in the example embodiment starts running without a skin-colour model. A new frame is acquired in step 402. In order to initialize and maintain the skin-colour model automatically, Intel's open source library (OpenCV) is used to search and detect the user's face in each frame in step 404. In one example, step 404 uses the Viola and Jones classification algorithm.

To extract the skin colour information, the output of the camera is first linearized using a look-up table (LUT) derived from the tonal calibration of the camera as described earlier on in Equations (7) – (8). When this calibration is not possible, an approximate response function is used. Then the data from the detected face region is converted from the RGB format to the HSV format.

If a face is detected in step 404, a mask is then built for the face regions in step 406 based on the HS ranges in these regions. For example, in the detected face region, there are naturally some non-skin colour areas, such as the eyes, eyebrows, mouth, hair and the background. The HS values of these areas are far away from the HS values in the skin colour area within the HS space and are hence applied a mask. This leaves only the "fleshy" areas of the face to contribute to the HS histogram.

In step 408, the hand Region of Interest (ROI) is defined based on the face position. In one example, when the user is facing the system, it is assumed that the right hand is on the right side of the image and the left hand is on the left side of the image. The horizontal aspect of the ROI for the right (or left) hand is then taken to be the right (or left) side of the image starting slightly from the right (or left) of the face to the right(or left) edge of the camera's image. The vertical aspect of the ROI of both hands starts from just above the user's head to the bottom of the image. In one example, with these ROIs defined, the subsequent joint probability map computed can be reduced by masking the region outside the ROIs to zero.

In step 410, the HS histogram of the skin on the face is obtained. In one example, the HS histogram is computed according to the description below.

Assuming that the image, I, has width = w and height = h and that after the hue and saturation computation, each pixel in the image has an illumination invariant hue and saturation value, $S(I_{x,y})$ is the saturation component for the pixel in the location row = x and column = y and correspondingly, $H(I_{x,y})$ is the hue component for the same pixel location. Considering a subset of pixels ψ from the image I which are from a part of the image detected as the face, a HS histogram can be constructed by using an appropriate quantization of the hue and saturation values. In one example, the histogram is of size 120 x 120 whereby this size has been proven to be effective via testing. However, this quantization can easily be changed. By setting maxValue in the calculation of the saturation to an appropriate value, the hue and the saturation components can be quantized into discrete values whereby the number of discrete values is equal to maxValue. The quantization of the hue (H) component to give the quantized value $\hat{H}$ is given in Equations (9). For example, choosing maxValue as 120 will quantize each of the hue values into one of 120 discrete values.

$$\hat{H} = \frac{H}{360} \times \max Value \qquad (9)$$

In the example embodiments, to construct a histogram K of dimension maxValue x maxValue, an indicator function δ is first defined in Equation (10).

$$\delta(x,x',y,y') = \begin{cases} 1 & \text{if } x = x' \text{ and } y = y' \\ 0 & \text{otherwise} \end{cases} \qquad (10)$$

Then the two-dimensional histogram K with indices 0 ≤ i < maxValue and 0 ≤ j < maxValue may be defined according to Equation (11). In Equation (11), w is the width and h is the height of the image I.

$$K_{i,j} = \sum_{s=0}^{\omega-1}\sum_{t=0}^{h-1} \delta(\hat{H}(\psi_{s,t}),i,S(\psi_{s,t}),j) \qquad (11)$$

In one example, the HS histogram of the new frame obtained according to Equation (11) is added to a set of previously accumulated histograms. In this example, to provide added robustness and stability to the model, a record of previous histograms is kept. Furthermore, when one (or both) of the users' hands have been previously detected, this information can be used to supplement the skin-tone model, in order to increase the robustness of the system. The final histogram can then be an aggregate of the previous histograms collected. For example, this history can extend back over a finite and small region, approximately 10 frames, allowing for adaptability and changes in users, lighting, changes in camera parameters, etc., while still gaining performance benefits from signal averaging and increased sample data.

In step 412, a probability map indicating the probability that each pixel in the image is a part of the skin is calculated by transforming the histogram obtained at the end of step 410. The histogram is first transformed into a probability distribution through normalization, specifically according to Equation (12) whereby T is given by Equation (13). In Equation (12), $\hat{K}_{i,j}$ is the normalized histogram which can also be termed the probability distribution.

$$\hat{K}_{i,j} = \frac{K_{i,j}}{T} \qquad (12)$$

$$T = \sum_{i=0}^{\max Value} \sum_{j=0}^{\max Value} K_{i,j} \qquad (13)$$

After obtaining the normalized histogram in Equation (12), given a light intensity invariant skin model, the probability distribution in the example embodiments can then be back projected onto the image in HS space yielding a probability map according to Equation (14). The ROIs for the left and right hands, where regions outside of the ROIs are masked to zero in the probability map, are used in the example embodiments in the back projection of the normalized histogram. The back projection can be limited to candidate regions of the input image corresponding to the ROIs hence reducing computation time. This back projection of the skin colour region onto the candidate regions of the input image can produce adequate probability maps when used to detect skin regions since the skin colour regions of the face and the hand almost overlap with each other.

The probability map $M_{i,j}$ obtained at the end of step 412 indicates the probability that the pixel i,j in the image corresponds to skin.

$$M_{i,j} = \hat{K}_{H(I(i,j)),S(I(i,j)))} \qquad (14)$$

If no face is detected, in step 414, it is determined if the hands are in front of the face by checking if the ROI of the hands was close to the ROI of the face in a previous frame. If the hands are not detected in the previous frame, this step is omitted and the algorithm starts from step 402 again. If it is determined that the hands are not in front of the face, a new frame is acquired and the algorithm starts from step 402 again. If the hands are in front of the face, histograms of previous frames are extracted in step 416. The hand ROI is then defined based on its ROI position in the previous frame in step 418. Steps 414, 416 and 418 allow the hand ROI to be defined when the face is not detected in situations such as when the hands occlude the face.

Step 412 is performed after step 418. If no face is detected, in step 412, a probability map indicating the probability that each pixel in the image is a part of the skin is calculated using the normalized previous frame histogram as obtained in Equation (12) i.e. the previous frame probability distribution. In step 412, this probability distribution from the previous frame is back projected onto the current image in HS space to yield the probability map. The ROIs for the hands, where regions outside of the ROIs are masked to zero in the probability map, are used in the example embodiments in the back projection of the normalized previous frame histogram. The back projection can be limited to candidate regions of the input image corresponding to the ROIs hence reducing computation time.

Using a stereo camera, for example a Point Gray Research Bumblebee2 [http://www.ptgrey.com/products/bumblebee2/index.asp.] in an example implementation, it is possible to get depth information based on pixel disparities resulting from the two cameras of the system having differing spatial locations. Given the task of tracking hands, and also given that a user is facing an interactive system and the camera system, one assumption made in the example embodiment is that it is likely that the user's hands are in front of his or her face. Another assumption made in an example embodiment is, that the closer an object is to the system, the more likely it is to be one of the user's hands. Thus, in considering an appropriate probability function, two distances are considered in an example embodiment. The first is d_{face}, the distance of the user's face to the stereo camera. The second distance is hardware dependent, d_{min}, the minimum distance an object can be to the camera, such that the system is still able to approximate a depth. Consider a function Δ (D), given a distance detected, D:

$$\Delta(D) = \begin{cases} 0 & \text{if } D > d_{face} \\ 1 & \text{if } D < d_{min} \\ 1 - \frac{D-d_{min}}{d_{face}-d_{min}} & \text{otherwise} \end{cases} \tag{15}$$

Given this function, and an assumption in an example embodiment that negative distances are not achievable, the function may be normalized to become a probability function by multiplying Δ(D) by $1/(0.5\cdot(d_{face}-d_{min})+d_{min})$. Thus, the probability of a point being from a hand (Pr(H|D) given it's distance D can be considered as:

$$Pr(H|D) = \frac{1}{0.5\cdot(d_{face}-d_{min})+d_{min}}\cdot\Delta(D) \tag{16}$$

Considering a depth or disparity image M based on an output of the stereo camera, one can convert this to a probability map N, where each discrete point in the map N is a hand probability given the detected distance:

$$N_{i,j} = Pr(H|\eta_{i,j}). \tag{16}$$

In an example embodiment, the probability map M described previously (see equation (14)) has the dimension w × h. Each point in the map is assigned a probability of being skin. Similarly, depth is used to assign a probability to each point of being from a hand, given it's approximated distance to the camera system. This map is denoted N (see equation (16)) and also has dimension w × h. A joint probability map P of the probability of a point being a hand given it's depth and color can advantageously be created, by taking the dot product of the two distributions:

$$P_{i,j} = M_{i,j} \cdot N_{i,j} \quad (17)$$

This joint probability function has been shown to be effective for the purpose of hand tracking, in particular where a stereo camera system is used in which the camera response function and it's inverse are known. The resulting probability map P can be given to a tracking system such as a Camshift tracker [BRADSKI, G. R. 1998. Computer vision face tracking for use in a perceptual user interface. Intel Technology Journal, Q2, 15] for the purpose of tracking hands. For further details of a probabilistic tracking method for use in an example embodiment, reference is also made to [MANDERS, C., FARBIZ, F., CHONG, J. H., TANG, K. Y., CHUA, G. G., AND LOKE, M. H. 2008. robust hand tracking using a skin-tone and depth joint probability model. to appear in 8th IEEE Intl Conf Automatic Face and Gesture Recognition, published September 17th 2008], the contents of which are hereby incorporated by cross-reference.

In one implementation, the openCV [Intel Open Source Computer Vision Library. http://www.intel.com/research/mrl/research/opencv/] implementation of Camshift is used, and the center of the CvBox2D returned by the tracker is used as the reference as to where the user's hands are located, with two Camshift trackers employed, one to track each of the user's hands. Through this method, it is also possible to acquire the angle of the user's hand. Though this information is fairly accurate, it was not used to manipulate any parameter in regards to the 3D object being controlled in the described example embodiments. However, it will be appreciated that this information may additionally be used in different embodiments to manipulate the 3D object being controlled. Given the user's tracked two hands attributed positions in 3D space, this information is used in an example embodiment to manipulate the 3D object in the 3D environment. The set of gestures (or hand positions) of a preferred embodiment used to manipulate a 3-dimensioanl image or object, e.g. in the for of a virtual object, are shown in Figures 5 to 7. The overall set of gestures is advantageously quite intuitive, and essentially centered around the user's face or part thereof. This position forms the origin of the gesture coordinate system. The z-axis is co-linear with the optical center of one of the stereo cameras, and the x- and y-axes are parallel to the horizontal and vertical axes respectively of the image frame of the stereo cameras, which are preferably aligned with the real-life horizontal and vertical directions.

The x, y, and z origins of the gesture 3D co-ordinate system are set as the center of the user's face. The location of the user's face is determined in real-time (at the camera's frame rate) using e.g. Intel's open source library (OpenCV) as well as depth information from the stereo cameras and moves as the user moves. Locations between the origin and the camera are considered to have a positive z component, and objects further away from the origin to have a negative z component.

To move the object upwards (increase the y-axis value), both of the user's hands are positioned above the origin of the co-ordinate system, as illustrated in Figure 5(a). As the average value of the y-component increases,. the object will be moved towards positive Y-axis. To move the object downward in the y-direction, both hands are positioned below the origin of the co-ordinate system, as illustrated in Figure 5(b). If we consider the two hand positions in this co-ordinate system as $h_L = \{x_l, y_l, z_l\}$ and $h_R = \{x_r, y_r, z_r\}$ for the left and right hands, the term α_y used to move the 3D object along the y-axis is:

$$a_y = \alpha \frac{y_l + y_r}{2}$$

where α is the scale factor.

α_y will can be interpreted differently depending on the application. It can be considered as the positional value, velocity, or acceleration in y-the direction depending on the application. For instance, in an application to control a virtual car (or a spaceship), it may be more intuitive that α_y refers to the acceleration similar to the acceleration pedal in a real car.

Moving the object in the direction of the x-axis is preferably also quite intuitive. The average value of the right and left hand positions in terms of their x-axis is used. If the user's hands are equidistant from his or her head, the object will stay stationary in terms of the x-axis. As the user moves his or her hands toward the left or right, the 3D object will move in that direction related to the term α_x that increases as the mean value of the x-coordinates increases:

$$a_x = \beta \frac{x_l + x_r}{2}$$

In this case, β is used to scale the velocity, acceleration, or change in position along the x-axis. This is shown in Figure 5(c) and (d).

Moving the object along the z-axis will now be described for the example embodiment. A reference point at some distance δ_z positioned on the z-axis is defined. If the average value of the z-components of the hands is greater than δ_z, the 3D object increases its z-axis component, related to the term α_z defined below that is proportional to the distance away from the point δ_z. As the user moves their hands toward his or her body (i.e. average value of the z-components of the hands is smaller than δ_z), the 3D object decreases its z component. The acceleration δ_z can be expressed as:

$$a_z = \gamma \frac{(z_l + z_r) - 2\delta_z}{2}$$

Again, δ is a constant scalar and α_z can be the actual position, the velocity, or the acceleration in the z-direction, depending and the application. This is depicted in Figures 6(a) and (b).

For rotational yaw movements, given the co-ordinate system in the example embodiment described, if the user wants to yaw the 3D object clockwise, i.e. rotate about the y-axis in a clockwise manner, the user positions the right hand toward the face (less than δ_z), and the left hand away from the face (greater than δ_z). The opposite is done to yaw the 3D object counter-clockwise. This can be expressed as a rotational yaw v_{yaw} as:

$$v_{yaw} = \psi(z_l + z_r - 2\delta_z)$$

Again here, how the v_{yaw} value relates to the yaw rotation depends on the application. It can be considered as the actual rotational angle, the angular velocity, or the angular acceleration.

As in the translational computations, ψ is a constant used to scale v_{yaw}. This movement is depicted in Figures 6(c) and (d).

To roll (rotation about the z-axis), the user positions one hand above the y-axis of the co-ordinate system and the other hand below the y-axis. Placing the right hand below the origin and the left hand above will cause a clockwise rotation about the z-axis. Similarly, moving the right hand above y-axis and the left hand below the y-axis will roll the 3D object in a counter-clockwise manner. This can be expressed as a rotational roll term v_{roll} which will be related to the actual rotation in roll depending on the application in a similar way as v_{yaw}.

$$v_{roll} = \xi(y_l - y_r)$$

The constant ξ is used to scale v_{roll}. This is presented in Figures 7(a) and (b).

In the described example embodiments, translations and rotations can advantageously be combined. For example moving the 3D object along the z-axis while rolling or yawing, or moving the object simultaneously in the x- and y-axis directions.

In an alternative embodiment, a marker may be used to facilitate face detection for extracting the skin colour information. In one such embodiment, the user views a virtual 3D space by means of a head mounted display (HMD) that presents a 3D image of the content. Using a marker that is affixed to the HMD, the users head location and position is tracked by means of using the ARToolkit [KATO, H., AND BILLINGHURST, M. 1999. Marker tracking and hmd calibration for a video-based augmented reality conferencing system] [BILLINGHURST, M., AND KATO, H. 2002. Collaborative augmented reality. Communications of the ACM 45, 7 (July), 64–70]. The HMD is positioned such that the lower region, i.e. the chin, of the users face is visible to the stereoscopic camera.. The z-axis is co-linear with the optical center of one of the stereo cameras, and the x- and y-axes are parallel to the horizontal and vertical axes respectively of the image frame of the stereo cameras, which are preferably aligned with the real-life horizontal and vertical directions.

Given that the user is wearing an HMD to experience the immersive environment, the tracking marker may be placed on the front of the HMD. Having determined the exact location and orientation of the tracking marker, one can readily locate the user's chin, i.e. the portion of the user's face not obstructed from the camera's view. Regions that were sampled in an example embodiment implementation are shown in Figure 8. The z-, x- and y-axes origins are determined automatically from the position and depth of the ARToolkit tracking marker, with a minor correction for the y-axis to position the origin of the coordinate system on the user's chin.

As shown in Figure 8, if the position of the marker 800 is known, an accurate sample of the user's skin tone may be acquired by sampling from the regions 801 to 804 shown in Figure 8. This sampling, in tandem with the depth information gained from the stereo camera as described above for the preferred embodiment, is used to track the user's hands in two dimensions. In brief again, the histogram of the sampled skin patches is first converted to HSV space from the original RGB space. Before applying this color transformation, the pixel values are first range-decompressed to correct for the nonphotometric characteristic of pixel data typically recovered from commercial cameras. Using the range decompression in addition to using an HSV transformation and disregarding the luminance component can greatly increases the illumination invariance when tracking the user's hand.

After producing a 2D histogram from the chrominance and hue components of the HSV colorspace, given the skin-colored pixels found in the exposed skin area, the histogram is normalized to become a probability distribution. Given each pixel in a video frame, the probability function recovered is back-projected to yield a skin probability for each pixel in the frame. We refer to this 2D back-projected probability map as the skin probability image. Similarly, depth is used to assign a probability to each point/pixel of being from a hand, given its approximated distance to the camera system. A joint probability map of the probability of a pixel belonging to hand given its depths and color is then created, by taking the dot product of the two distributions.

Figure 9 shows a flow chart 900 illustrating a method of gesture based manipulation of a 3-dimensional image or object according to an example embodiment. At step 902, the user's face, right and left hands are tracked using stereo image processing. At step 904, a 3-dimensional gesture coordinate system is defined based on the tracked user's face. At step 906, the user's gesture is determined based on the tracked movement of the user's right and left hands within the 3-dimensional gesture coordinate system. At step 908, the image or object are manipulated based on the detected user's gesture.

The method and system of the example embodiment can be implemented on a computer system 1000, schematically shown in Figure 10. It may be implemented as software, such as a computer program being executed within the computer system 1000, and instructing the computer system 1000 to conduct the method of the example embodiment.

The computer system 1000 comprises a computer module 1002, input modules such as a keyboard 1004 and stereo camera 1006 and a plurality of output devices such as a display 1008, and a 3-D manipulation robot 1010.

The computer module 1002 is connected to a computer network 1012 via a suitable transceiver device 1014, to enable access to e.g. the Internet or other network systems such as Local Area Network (LAN) or Wide Area Network (WAN).

The computer module 1002 in the example includes a processor 1018, a Random Access Memory (RAM) 1020 and a Read Only Memory (ROM) 1022. The computer module 1002 also includes a number of Input/Output (I/O) interfaces, for example I/O interface 1024 to the display 1008, and I/O interface 1026 to the keyboard 1004.

The components of the computer module 1002 typically communicate via an interconnected bus 1028 and in a manner known to the person skilled in the relevant art.

The application program is typically supplied to the user of the computer system 1000 encoded on a data storage medium such as a CD-ROM or flash memory carrier and read utilising a corresponding data storage medium drive of a data storage device 1030. The application program is read and controlled in its execution by the processor 1018. Intermediate storage of program data maybe accomplished using RAM 1020.

It will be appreciated by a person skilled in the art that numerous variations and/or modifications may be made to the present invention as shown in the specific embodiments without departing from the spirit or scope of the invention as broadly described. The present embodiments are, therefore, to be considered in all respects to be illustrative and not restrictive.

For example, to make the interaction more effective for 3D virtual environments, changing the camera angle in relation to the user's head movements may be played.

CLAIMS

1. A method of gesture based manipulation of a 3-dimensional image or object, the method comprising the steps of:

tracking the user's face, right and left hands using stereo image processing;

defining a 3-dimensional gesture coordinate system based on the tracked user's face;

determining the user's gesture based on the tracked movement of the user's right and left hands within the 3-dimensional gesture coordinate system; and

manipulating the image or object based on the detected user's gesture.

2. The method as claimed in claim 1, wherein defining the 3-dimensional coordinate system comprises:

defining an origin, an x-axis and a y-axis of the 3-dimensional gesture coordinate system based on the tracked user's face; and

defining a z-axis of the 3-dimensional gesture coordinate system based on an optical centre of one camera of a stereo camera system used for the stereo video image processing.

3. The method as claimed in claim 2, wherein determining the user's gesture comprises determining whether both the user's right and left hands are above or below the y-axis for determining an intended movement of the image or object in an upward or downward direction.

4. The method as claimed in claim 3, further comprising determining an intended upwards or downward movement based on an average y-component of the right and left hands.

5. The method as claimed in any one of claims 2 to 4, wherein determining the user's gesture comprises determining whether the user's right and left hands are equidistant from the x-axis for determining an intended movement of the image or object in a right of left direction.

6. The method as claimed in claim 5, further comprising determining an intended right or left movement based on an average x-component of the right and left hands.

7. The method as claimed in any one of claims 2 to 7, wherein defining the 3-dimensional coordinate system further comprises defining a reference position along the z-axis between the origin and the stereo camera.

8. The method as claimed in claim 7, wherein determining the user's gesture comprises determining whether an average z-component of the user's right and left hands is greater than or smaller than the reference position for determining an intended movement of the image or object in a forward, or backward direction.

9. The method as claimed in claim 8, further comprising determining an intended forward or backward movement based on an average z-component of the right and left hands.

10. The method as claimed in any one of claims 7 to 9, wherein determining the user's gesture comprises determining whether a z-component of the user's right hand is smaller than the reference position and a z-component of the left hand is greater than the reference position for determining an intended clockwise rotational yaw movement of the image or object.

11. The method as claimed in any one of claims 7 to 10, wherein determining the user's gesture comprises determining whether a z-component of the user's left hand is smaller than the reference position and a z-component of the right hand is greater than the reference position for determining an intended counter-clockwise rotational yaw movement of the image or object.

12. The method as claimed in claims 10 or 11, further comprising determining an intended rotational yaw movement based on a sum of the z-components of the right and left hands.

13. The method as claimed in any one of claims 7 to 12, wherein determining the user's gesture comprises determining whether a y-component of the user's right hand is below the y-axis and a y-component of the left hand is above the y-axis for determining an intended clockwise rotational roll movement of the image or object.

14. The method as claimed in any one of claims 7 to 13, wherein determining the user's gesture comprises determining whether a y-component of the user's left hand is below the y-axis and a y-component of the right hand is above the y-axis for determining an intended counter-clockwise rotational roll movement of the image or object.

15. The method as claimed in claims 13 or 14, further comprising determining an intended rotational roll movement based on a difference of the y-components of the left and right hands.

16. The method as claimed in any one of the preceding claims, wherein tracking the user's face comprises using marker-less stereo video image processing.

17. The method as claimed in any one of the preceding claims, wherein tracking the user's right and left hands comprises utilising a joint probability map based on skin colour and depth information.

18. A system for gesture based manipulation of a 3-dimensional image or object, the method comprising the steps of:

means for tracking the user's face, right and left hands using stereo image processing;

means for defining a 3-dimensional gesture coordinate system based on the tracked user's face;

means for determining the user's gesture based on the tracked movement of the user's right and left hands within the 3-dimensional gesture coordinate system; and

means for manipulating the image or object based on the detected user's gesture.

19. A data storage medium comprising computer code means fro instructing a computing device to execute a method of gesture based manipulation of a 3-dimensional image or object, the method comprising the steps of:

tracking the user's face, right and left hands using stereo image processing;

defining a 3-dimensional gesture coordinate system based on the tracked user's face;

determining the user's gesture based on the tracked movement of the user's right and left hands within the 3-dimensional gesture coordinate system; and

manipulating the image or object based on the detected user's gesture.

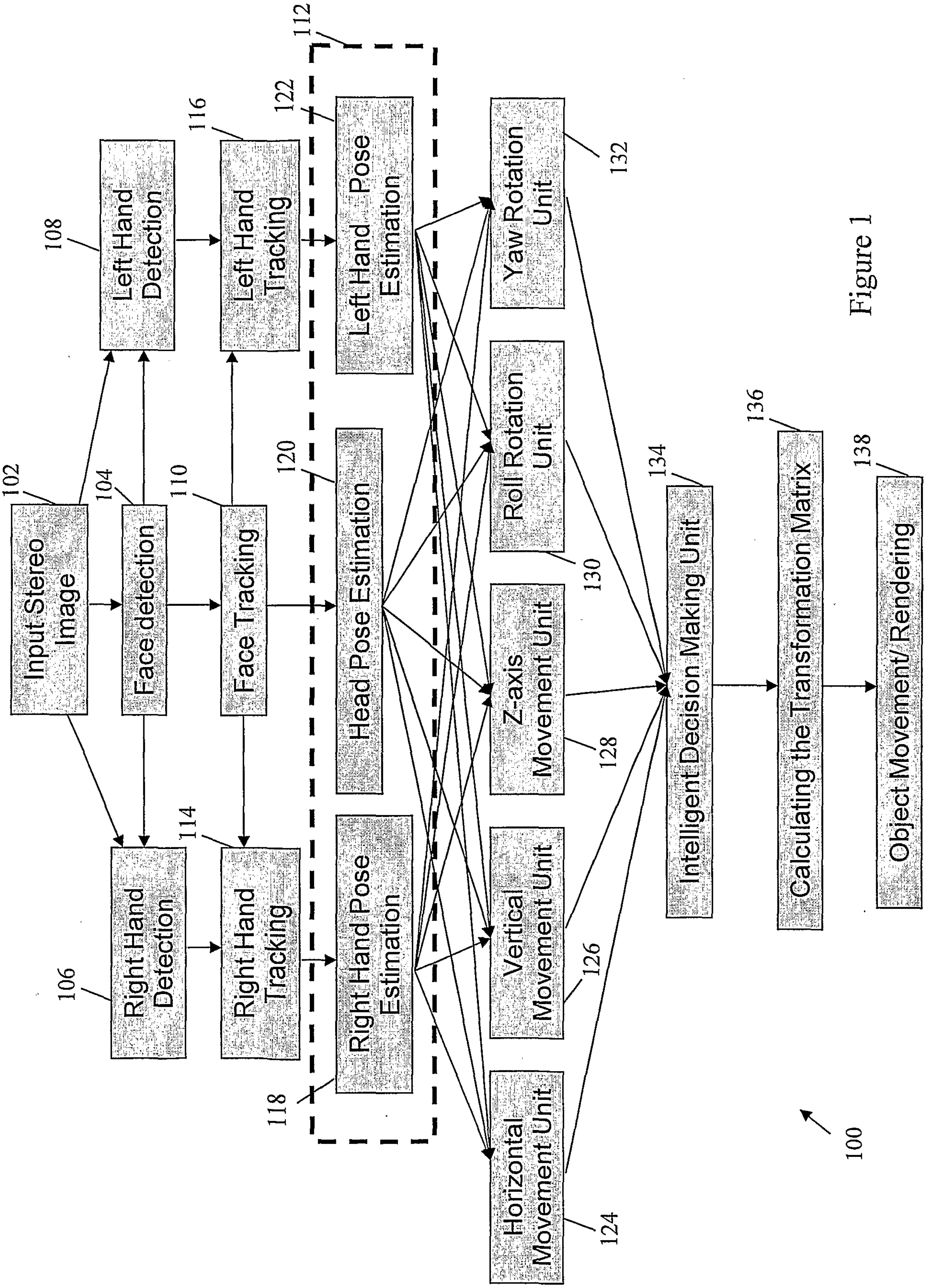

Input Stereo Image
102
Face detection
104
Right Hand Detection
106
Left Hand Detection
108
Face Tracking
110
112
Right Hand Tracking
114
Left Hand Tracking
116
Right Hand Pose Estimation
118
Head Pose Estimation
120
Left Hand Pose Estimation
122
Horizontal Movement Unit
124
Vertical Movement Unit
126
Z-axis Movement Unit
128
Roll Rotation Unit
130
Yaw Rotation Unit
132
Intelligent Decision Making Unit
134
Calculating the Transformation Matrix
136
Object Movement/ Rendering
138
100

Figure 1

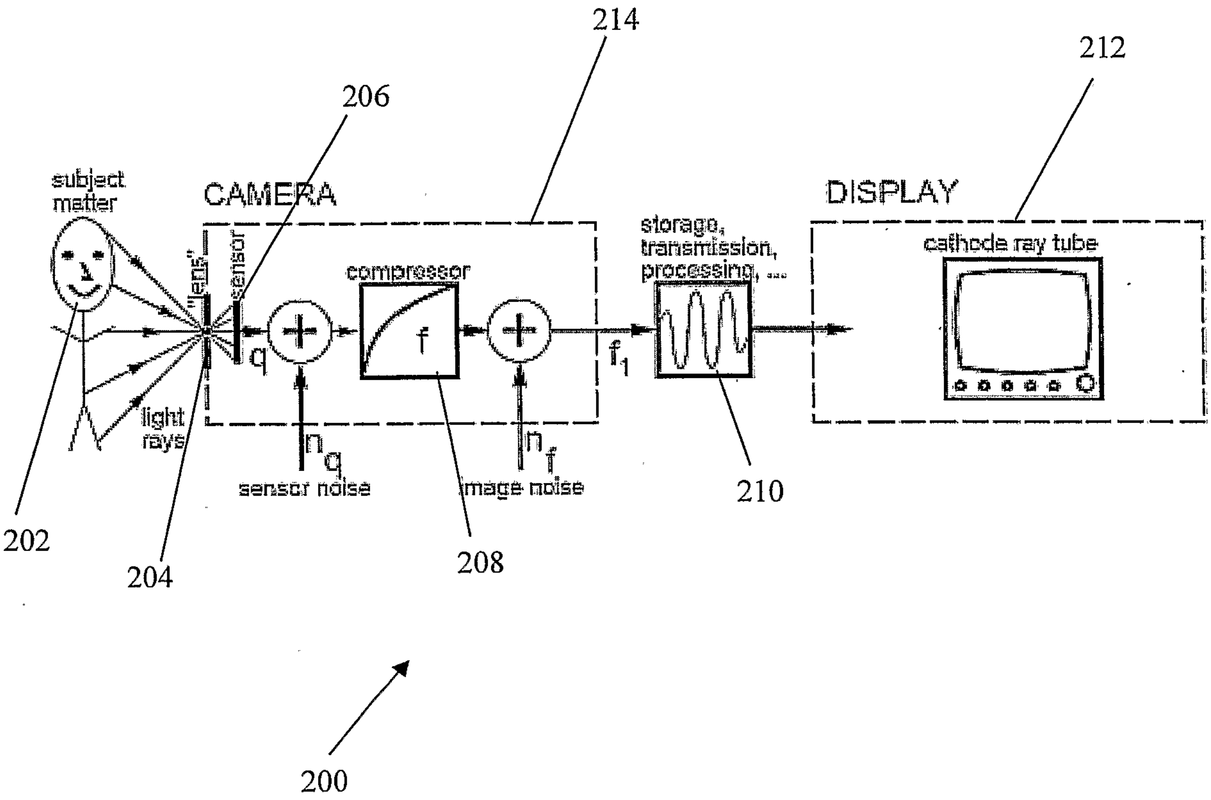

214
212
206
subject matter
CAMERA
DISPLAY
"lens"
sensor
compressor
storage, transmission, processing, ...
cathode ray tube
q
f
f_1
light rays
n_q
sensor noise
n_f
image noise
210
202
204
208
200

Figure 2

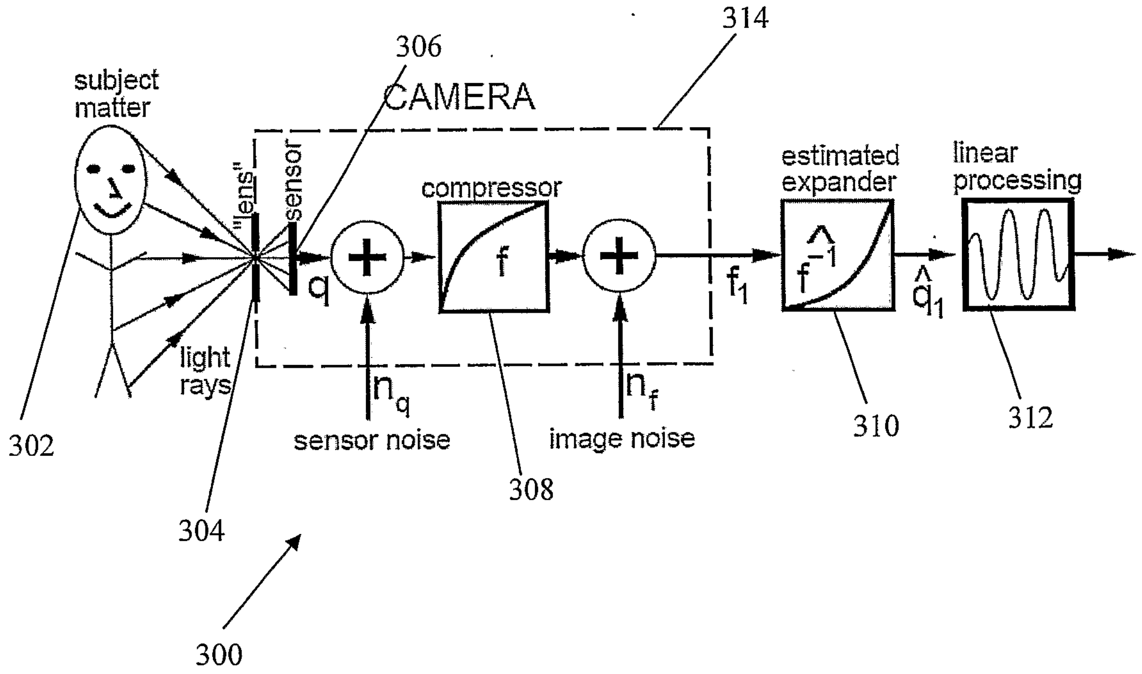

314
306
subject
matter
CAMERA
"lens"
sensor
compressor
f
q
estimated
expander
$\hat{f}^{-1}$
linear
processing
f_1
$\hat{q}_1$
light
rays
n_q
sensor noise
n_f
image noise
302
304
308
310
312
300

Figure 3

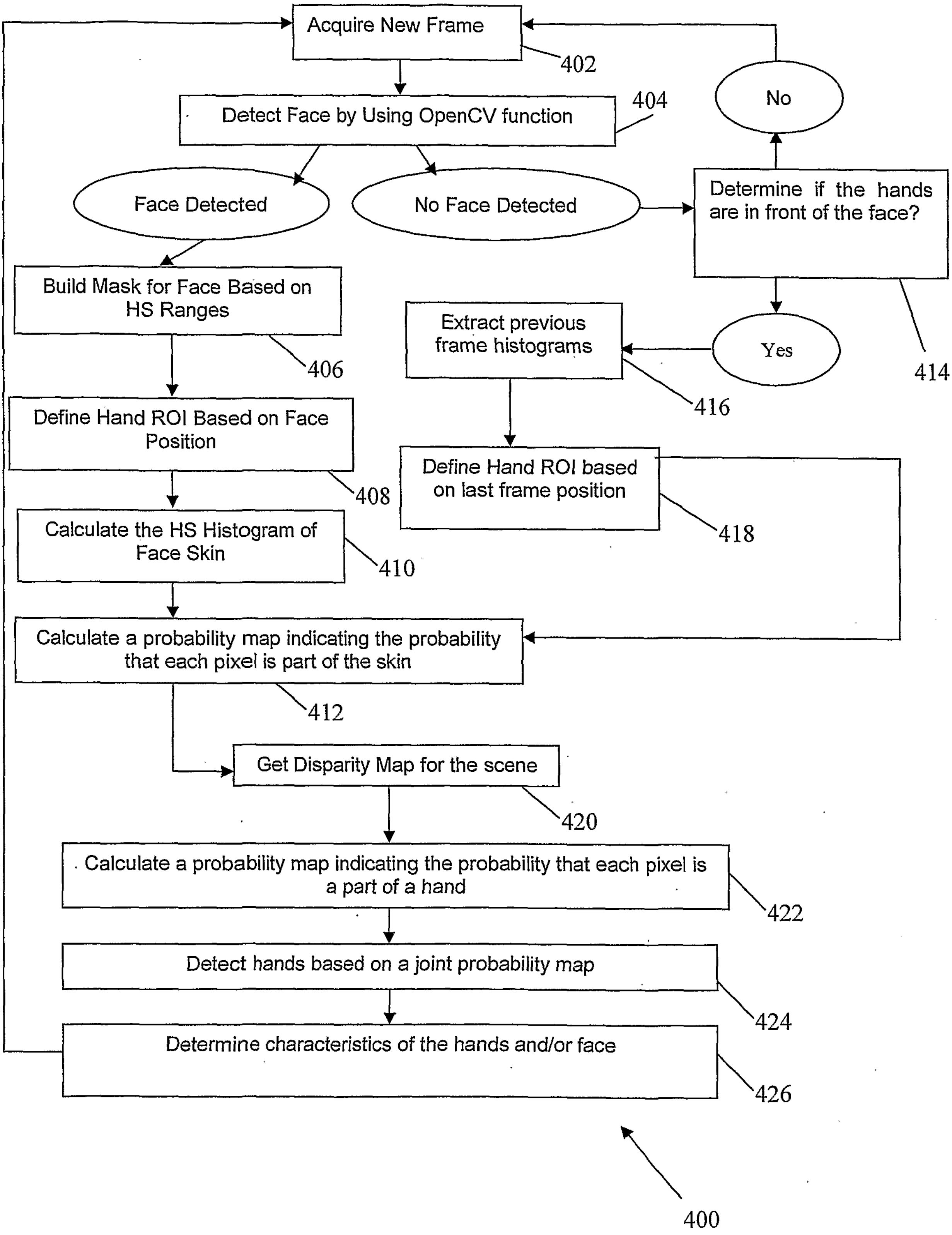
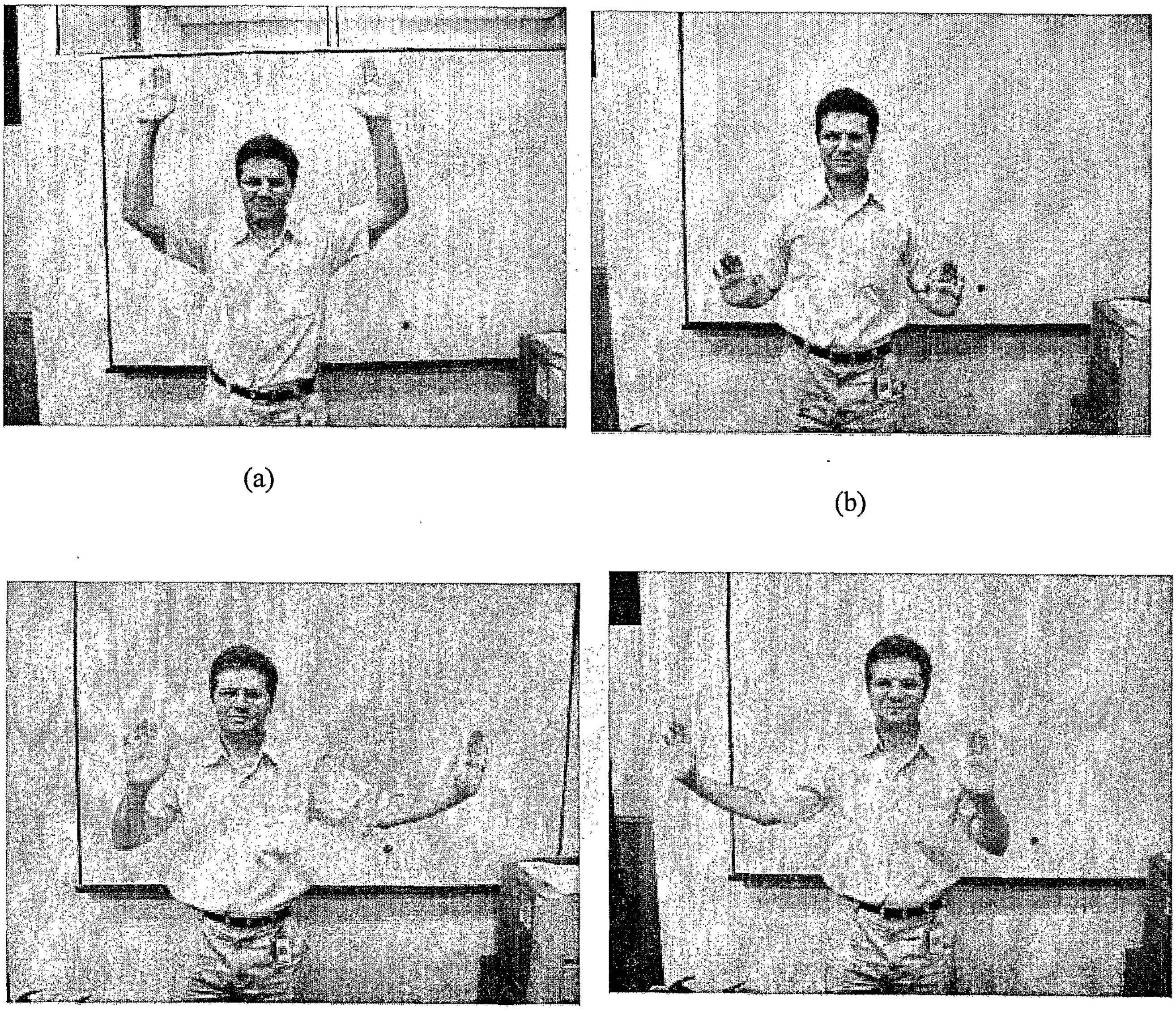

Acquire New Frame
402
Detect Face by Using OpenCV function
404
Face Detected
No Face Detected
Determine if the hands are in front of the face?
No
Yes
414
Build Mask for Face Based on HS Ranges
406
Extract previous frame histograms
416
Define Hand ROI Based on Face Position
408
Define Hand ROI based on last frame position
418
Calculate the HS Histogram of Face Skin
410
Calculate a probability map indicating the probability that each pixel is part of the skin
412
Get Disparity Map for the scene
420
Calculate a probability map indicating the probability that each pixel is a part of a hand
422
Detect hands based on a joint probability map
424
Determine characteristics of the hands and/or face
426
400

Figure 4



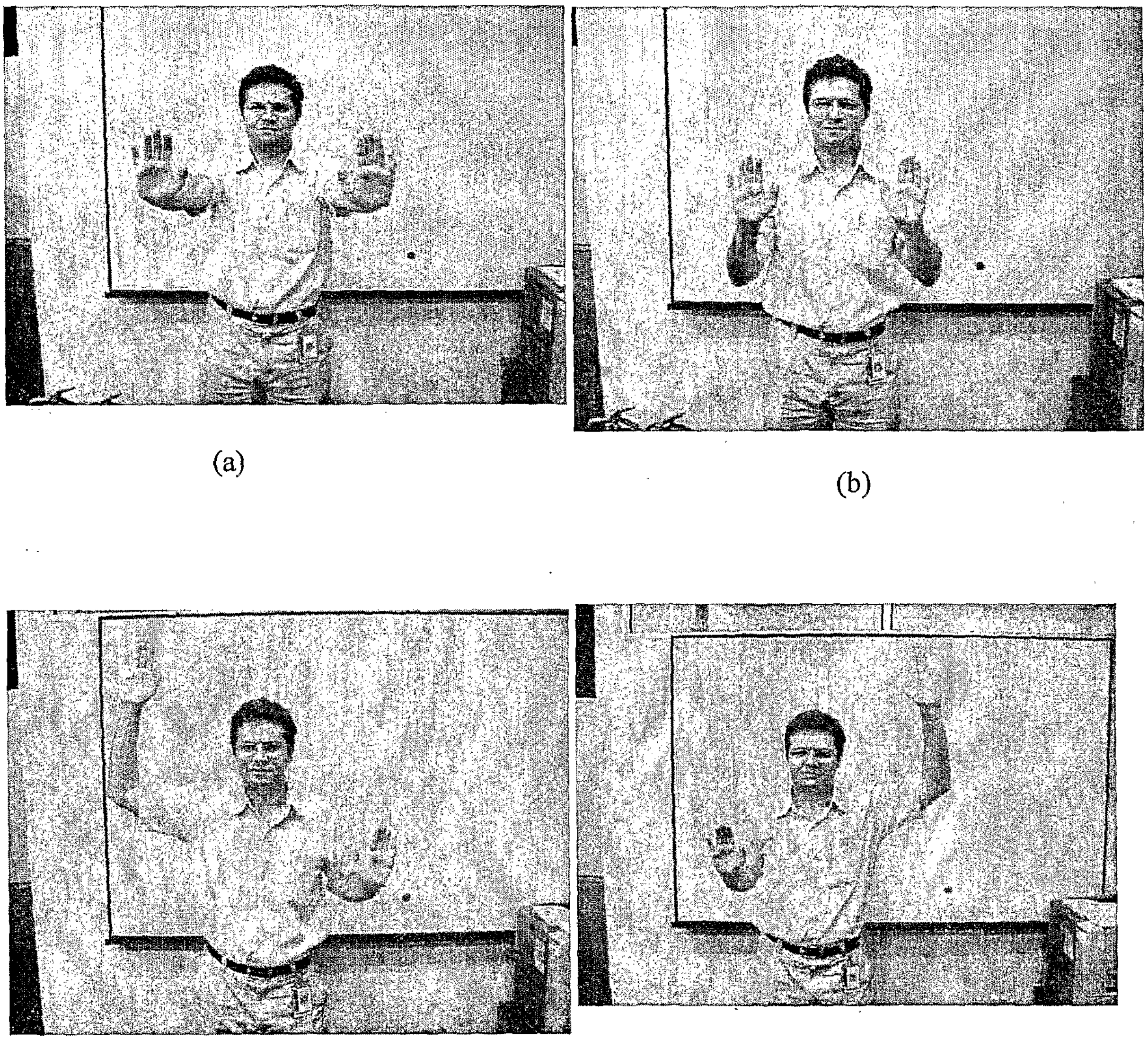
(a)

(b)

(c)

(d)

Figure 5



(a)

(b)

(c)

(d)

Figure 6

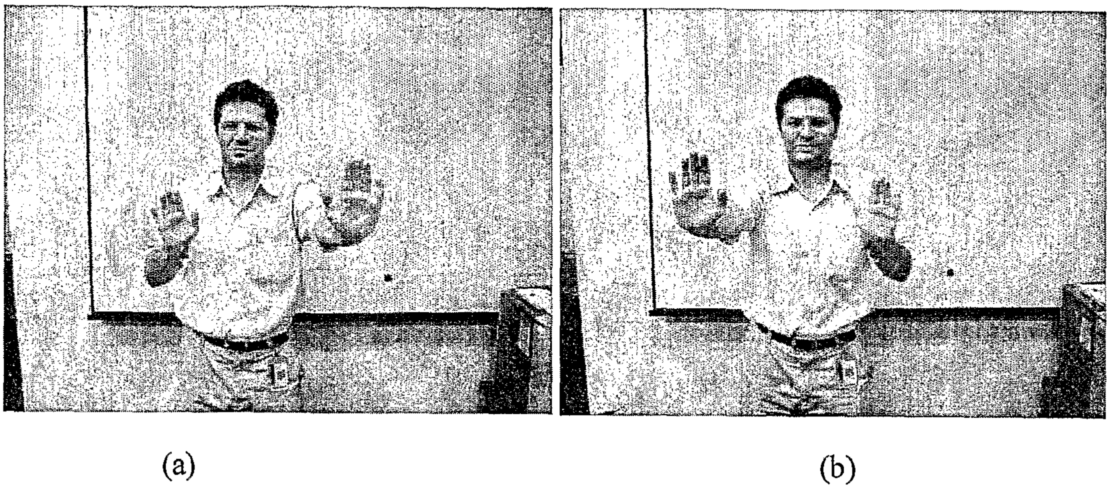


Figure 7

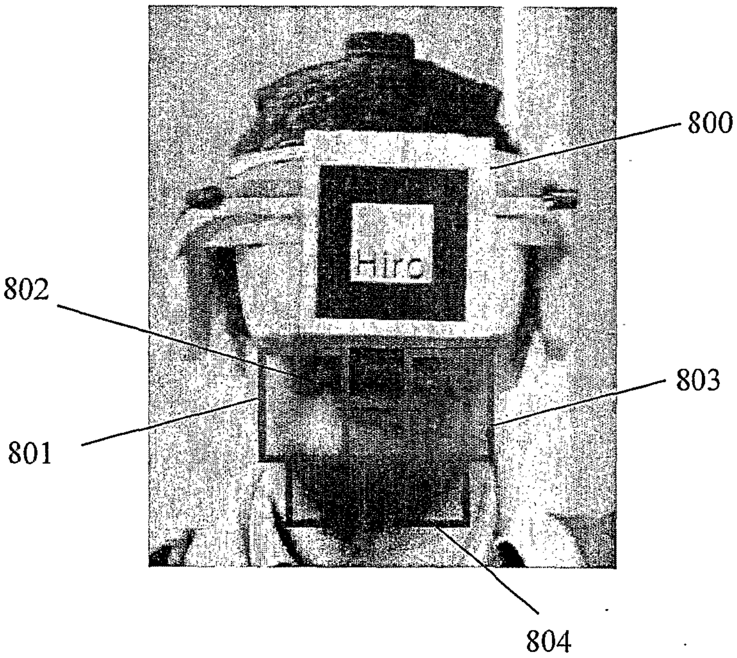
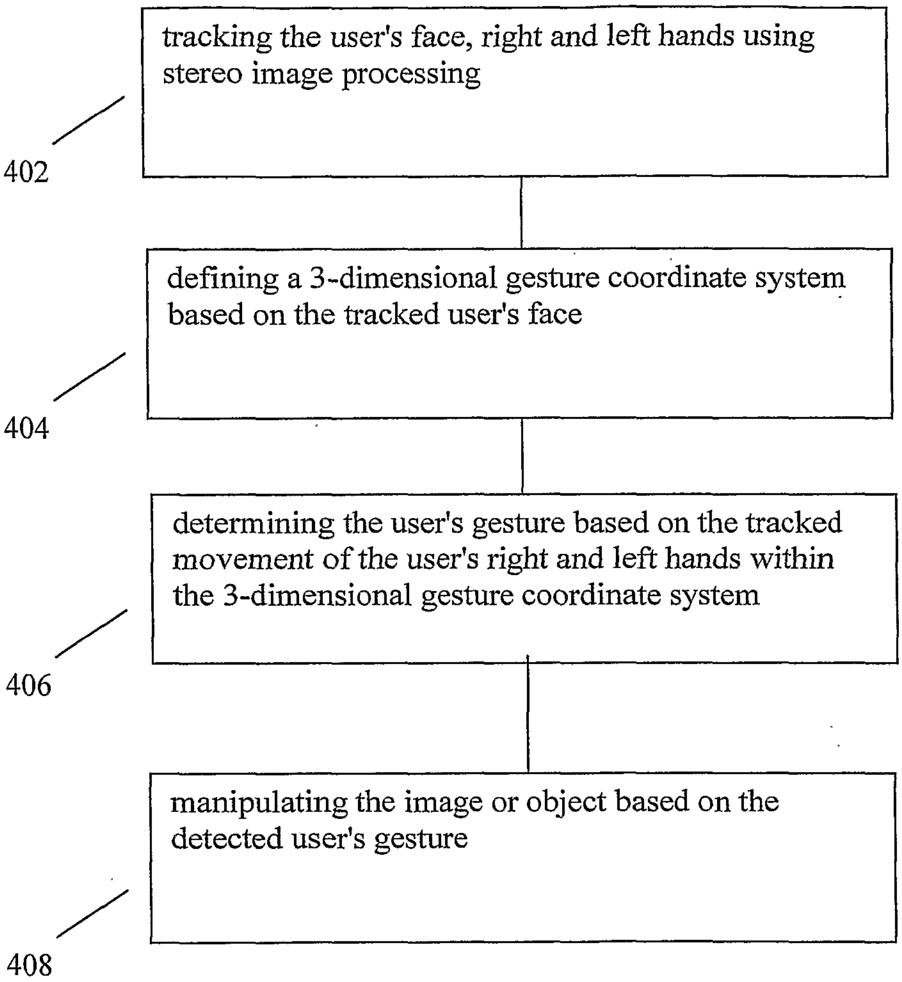


Figure 8


tracking the user's face, right and left hands using stereo image processing
402
defining a 3-dimensional gesture coordinate system based on the tracked user's face
404
determining the user's gesture based on the tracked movement of the user's right and left hands within the 3-dimensional gesture coordinate system
406
manipulating the image or object based on the detected user's gesture
408

900

Figure 9

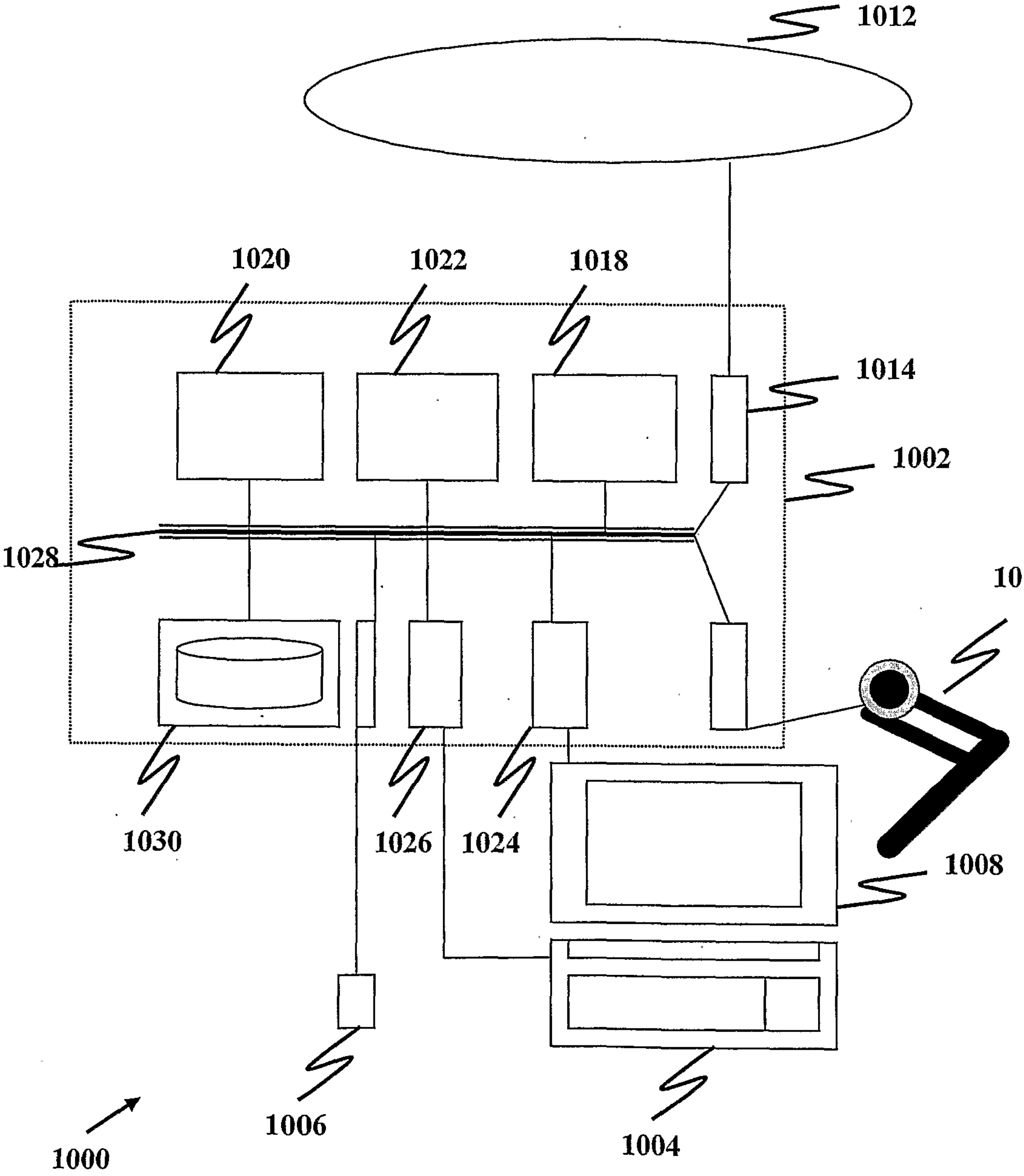

1012
1020
1022
1018
1014
1002
1028
1010
1030
1026
1024
1008
1006
1004
1000

Figure 10

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SG2009/000203

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl.

G06F 3/01 (2006.01) ***G06K 9/00*** (2006.01)

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)

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
WPI, Google Patent Search, Esp@cenet: keywords: (3D, gesture, manipulation, stereo) and similar terms

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2009/0077504 A1 (BELL et al.) 19 March 2009 para. [0008, 0026, 0044, 0055, 0058,0071, 0077-0085, 0088, 0095]; Fig. 4 items 440-450, 470, 490	1 and 16-19
Y	para. [0093-0094]	2-15
Y	US 6215890 B1 (MATSUO et al.) 10 April 2001 column 11 line 14 – column 12 line 2	2-15
O	MANDERS et al. 'Interacting with 3D objects in a virtual environment using an intuitive gesture system.' In: Proceeding of the 7th AVM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry (VRCAI '08), December, 2008, pages 1-5 Whole Document	1-19

□ 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	"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
"E"	earlier application or patent but published on or after the international filing date	"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
"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)	"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
"O"	document referring to an oral disclosure, use, exhibition or other means	"&"	document member of the same patent family
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 06 August 2009	Date of mailing of the international search report 17 AUG 2009
Name and mailing address of the ISA/AU AUSTRALIAN PATENT OFFICE PO BOX 200, WODEN ACT 2606, AUSTRALIA E-mail address: pct@ipaustralia.gov.au Facsimile No. +61 2 6283 7999	Authorized officer **Surya Prakash** AUSTRALIAN PATENT OFFICE (ISO 9001 Quality Certified Service) Telephone No : +61 2 6283 2101

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/SG2009/000203

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report		Patent Family Member					
US	2009077504	WO	2009035705				
US	6215890	CN	1218936	EP	0905644	JP	11174948

Due to data integration issues this family listing may not include 10 digit Australian applications filed since May 2001.

END OF ANNEX