



- (51) International Patent Classification:
G06T 19/20 (2011.01)
- (21) International Application Number:
PCT/SG2011/000440
- (22) International Filing Date:
16 December 2011 (16.12.2011)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
61/424,302 17 December 2010 (17.12.2010) US
- (71) Applicant (for all designated States except US):
**AGENCY FOR SCIENCE, TECHNOLOGY AND RE-
SEARCH** [SG/SG]; 1 Fusionopolis Way, #20-10, Connex-
is, Singapore 138632 (SG).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **NISWAR, Arthur**
[ID/SG]; c/o PKM, Institute for Infocomm Research, 1 Fu-
sionopolis Way, #21-01, Connexis, South Tower, Singa-

pore 138632 (SG). **ONG, Ee Ping** [SG/SG]; c/o PKM, In-
stitute for Infocomm Research, 1 Fusionopolis Way, #21-
01, Connexis, South Tower, Singapore 138632 (SG).
NGUYEN, Hong Thai [VN/SG]; c/o PKM, Institute for
Infocomm Research, 1 Fusionopolis Way, #21-01 Connex-
is, South Tower, Singapore 138632 (SG). **HUANG, Zhiy-
ong** [SG/SG]; c/o PKM, Institute for Infocomm Research,
1 Fusionopolis Way, #21-01, Connexis, South Tower,
Singapore 138632 (SG).

- (74) Agent: **ELLA CHEONG SPRUSON & FERGUSON (S)
PTE LTD**; PO 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, QA, RO, RS, RU, RW, SC, SD,

[Continued on next page]

- (54) Title: POSE-INDEPENDENT 3D FACE RECONSTRUCTION FROM A SAMPLE 2D FACE IMAGE

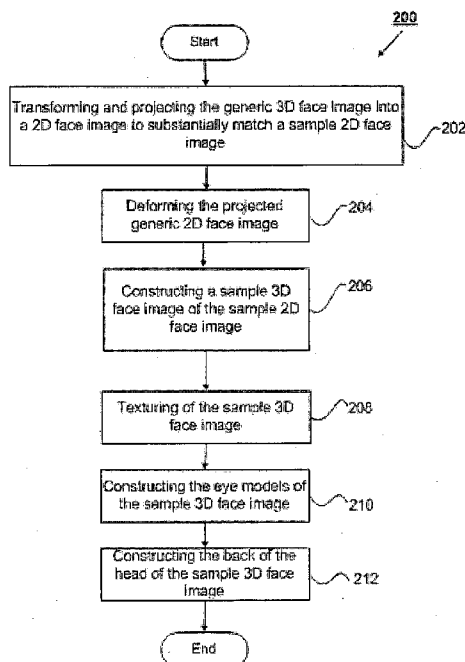


Fig. 2

- (57) Abstract: A method to reconstruct a 3D face image from a sample 2D face image is provided. A method for reconstructing a three-dimensional (3D) face image from a sample two-dimensional (2D) face image transforms a generic 3D face image to substantially match the pose of the 2D face image; projecting the transformed generic 3D face image into a projected 2D face image, deforms the projected 2D face image, and constructs a sample 3D face image from the deformed 2D face image based on the sample 2D face image.



SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR,
TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS,
SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM,
GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT,

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

Pose-Independent 3D Face Reconstruction from a Sample 2D Face Image

This application claims priority from U.S. Provisional Patent Application No. 61/424,302, the content of which is incorporated herein in its entirety by reference.

5

Technical Field

The present application relates to reconstructing a three-dimensional face image from a two-dimensional face image.

10

Background

Three-dimensional (3D) face image reconstruction from one or more two-dimensional (2D) face images is one of the fundamental problems in computer vision. Reconstructing a 3D face image from one or more 2D face image(s) consists in modifying an existing 3D face image, typically a generic 3D face image, based on a set of feature points located on the 2D face image(s). The 2D face image is at least one of one frontal face image, one face image from any pose (e.g., profile view, etc), and two images (e.g., two orthogonal images, two non-orthogonal images).

There are existing techniques to modify the generic 3D face image, such as a linear deformation (e.g., morphable face model, statistical face model, Principal Component Analysis (PCA), etc) or a non-linear deformation (e.g., Radial Basis Function (RBF), Barycentric coordinates image deformation).

A human can usually estimate the 3D shape of a face/head when presented with a sample 2D face image, regardless of the pose of the sample 2D face image. On the other hand, this is a complicated task for computers. The reconstruction of a 3D human face image from a sample 2D face image is a mathematically ill-posed problem, of which several methods have been proposed to solve. However, these methods have their limitations.

The morphable face model or the statistical face model are examples of conventional methods that require a database of 3D face images to construct a generic 3D face image. The generic 3D face image is then adjusted to the sample 2D face image using parameters extracted from the database of 3D face images, which is expensive and time-consuming due to the number of 3D face images that need to be collected.

Simple pose estimation and linear deformation are other examples of conventional methods used to match a generic 3D face image to the 2D face image. Due to the simplicity of these methods, the reconstructed 3D face image is often distorted when viewed at a non-frontal angle.

5 Thus, a need exists to provide a method for reconstructing a 3D face image from at least one 2D face image without the need for a database of 3D face images and still provide an accurate reconstruction of the 3D face image.

Summary

10 In accordance with an aspect of the present disclosure, there is provided a method for reconstructing a three-dimensional (3D) face image from a sample two-dimensional (2D) face image, said method comprising the steps of: transforming a generic 3D face image to substantially match the pose of the 2D face image; projecting said transformed generic 3D face image into a projected 2D face image; 15 deforming said projected 2D face image; and constructing a sample 3D face image from said deformed 2D face image based on said sample 2D face image.

In a further aspect of the present disclosure, the step of constructing a sample 3D face image of the above method comprises the steps of: texturing of said sample 3D face image based on texture of said sample 2D face image; constructing eye 20 models of said sample 3D face image based on eye models of said sample 2D face image; and constructing the back of a head of said sample 3D face image, said back of the head is constructed from said generic 3D face image.

In a further aspect of the present disclosure, the steps of transforming and projecting of the above method comprising the steps of: determining feature points on 25 said generic 3D face image and on said sample 2D face image; transforming said feature points of said generic 3D face image based on initial scaling, rotation, and translation values; projecting said transformed feature points of said generic 3D face image into 2D space; optimizing said scaling, rotation, and translation values to minimize an error value, wherein said error value is a difference between said 30 projected and transformed feature points of said generic 3D face image and said feature points of said sample 2D face image; computing an optimal projection matrix based on said optimized scaling, rotation, and translation values; and projecting said

generic 3D face image into a generic 2D face image using said optimal projection matrix.

5 In a further aspect of the present disclosure, the step of constructing a sample 3D face image from said deformed 2D face image based on said sample 2D face image of the above method comprises the steps of: determining depth information of said sample 2D face image; inserting said determined depth information into said deformed generic 2D face image; and obtaining symmetry of said deformed 2D face image.

10 In a further aspect of the present disclosure, the step of deforming said projected 2D face image further comprises the steps of: dividing said projected 2D face image into regions; and performing a Dirichlet Free-Form Deformation method on each region separately.

15 In another aspect of the present disclosure, there is provided a computer program product wherein instructions to perform a method for reconstructing a three-dimensional (3D) face image from a sample two-dimensional (2D) face image is stored in a computer readable medium, wherein said method comprising the steps of: transforming a generic 3D face image to substantially match the pose of the 2D face image; projecting said transformed generic 3D face into a projected 2D face image; deforming said projected 2D face image; and constructing a sample 3D face image
20 from said deformed and projected 2D face image based on said sample 2D face image.

Other aspects of the invention are also disclosed.

Brief Description of Drawings

25 At least one embodiment of the present invention will now be described with reference to the drawings, in which:

Figs. 1A and 1B form a schematic block diagram of a general purpose computer system upon which embodiments that will be described can be practiced;

Fig. 2 is a flow chart for a method for reconstructing a 3D face image from a sample 2D face image;

30 Fig. 3 is an example of a generic 3D face image;

Fig. 4 is an example of a generic 3D face image with added feature and contour points;

Fig. 5 shows two sample 2D face images with added feature points;

Fig. 6 is a flow chart for a method for transforming and projecting a generic 3D face image into 2D space;

Fig. 7 shows a comparison between the sample 2D face images with the transformed and projected generic 2D face image;

5 Fig. 8 shows a projected generic 2D face image with added contour points;

Fig. 9 shows an example of Voronoi diagrams;

Fig. 10 shows deformed generic 2D face images;

Fig. 11 is a flow chart of a method for reconstructing a 3D face image from the deformed generic 2D face images;

10 Fig. 12 shows a texture image of the generic 3D face image superimposed with a generic mesh.

Figs. 13A to 13C show the pixels of the sample 2D face image for texturing purposes;

Figs. 14A to 14B show the textured face images;

15 Figs. 15A to 15C depict the process of texturing the eyes of the reconstructed 3D face image;

Figs. 16A and 16B show the result of the eye texturing process; and

Figs. 17A and 17B depict the process of creating the back of the head of the reconstructed 3D face image.

20

Detailed Description

Where reference is made in any one or more of the accompanying drawings to steps and/or features, which have the same reference numerals, those steps and/or features have for the purposes of this description the same function(s) or operation(s),
25 unless the contrary intention appears.

Figs. 1A and 1B depict a general-purpose computer system 100, upon which the various arrangements described can be practiced.

As seen in Fig. 1A, the computer system 100 includes: a computer module 101; input devices such as a keyboard 102, a mouse pointer device 103, a scanner 126, a camera 127, and a microphone 180; and output devices including a
30 printer 115, a display device 114 and loudspeakers 117. An external Modulator-Demodulator (Modem) transceiver device 116 may be used by the computer module 101 for communicating to and from a communications network 120 via a

connection 121. The communications network 120 may be a wide-area network (WAN), such as the Internet, a cellular telecommunications network, or a private WAN. Where the connection 121 is a telephone line, the modem 116 may be a traditional “dial-up” modem. Alternatively, where the connection 121 is a high capacity (e.g., cable) connection, the modem 116 may be a broadband modem. A wireless modem may also be used for wireless connection to the communications network 120.

The computer module 101 typically includes at least one processor unit 105, and a memory unit 106. For example, the memory unit 106 may have semiconductor random access memory (RAM) and semiconductor read only memory (ROM). The computer module 101 also includes an number of input/output (I/O) interfaces including: an audio-video interface 107 that couples to the video display 114, loudspeakers 117 and microphone 180; an I/O interface 113 that couples to the keyboard 102, mouse 103, scanner 126, camera 127 and optionally a joystick or other human interface device (not illustrated); and an interface 108 for the external modem 116 and printer 115. In some implementations, the modem 116 may be incorporated within the computer module 101, for example within the interface 108. The computer module 101 also has a local network interface 111, which permits coupling of the computer system 100 via a connection 123 to a local-area communications network 122, known as a Local Area Network (LAN). As illustrated in Fig. 1A, the local communications network 122 may also couple to the wide network 120 via a connection 124, which would typically include a so-called “firewall” device or device of similar functionality. The local network interface 111 may comprise an EthernetTM circuit card, a BluetoothTM wireless arrangement or an IEEE 802.11 wireless arrangement; however, numerous other types of interfaces may be practiced for the interface 111.

The I/O interfaces 108 and 113 may afford either or both of serial and parallel connectivity, the former typically being implemented according to the Universal Serial Bus (USB) standards and having corresponding USB connectors (not illustrated). Storage devices 109 are provided and typically include a hard disk drive (HDD) 110. Other storage devices such as a floppy disk drive and a magnetic tape drive (not illustrated) may also be used. An optical disk drive 112 is typically provided to act as a non-volatile source of data. Portable memory devices, such

optical disks (e.g., CD-ROM, DVD, Blu-ray Disc), USB-RAM, portable, external hard drives, and floppy disks, for example, may be used as appropriate sources of data to the system 100.

The components 105 to 113 of the computer module 101 typically
5 communicate via an interconnected bus 104 and in a manner that results in a conventional mode of operation of the computer system 100 known to those in the relevant art. For example, the processor 105 is coupled to the system bus 104 using a connection 118. Likewise, the memory 106 and optical disk drive 112 are coupled to the system bus 104 by connections 119. Examples of computers on which the
10 described arrangements can be practised include PC and compatibles, Sun Sparcstations, Apple Mac or a like computer systems.

The method of reconstructing a 3D face image from a sample 2D face image may be implemented using the computer system 100 wherein the processes of Figs. 2 to 17, to be described, may be implemented as one or more software application
15 programs 133 executable within the computer system 100. In particular, all the steps of the method of reconstructing a 3D face image from a sample 2D face image are effected by instructions 131 (see Fig. 1B) in the software 133 that are carried out within the computer system 100. The software instructions 131 may be formed as one or more code modules, each for performing one or more particular tasks. The
20 software may also be divided into two separate parts, in which a first part and the corresponding code modules performs the reconstruction of the 3D face image method and a second part and the corresponding code modules manage a user interface between the first part and the user.

The software may be stored in a computer readable medium, including the
25 storage devices described below, for example. The software is loaded into the computer system 100 from the computer readable medium, and then executed by the computer system 100. A computer readable medium having such software or computer program recorded on the computer readable medium is a computer program product. The use of the computer program product in the computer system 100
30 preferably effects an advantageous apparatus for reconstructing a 3D face image from a sample 2D face image.

The software 133 is typically stored in the HDD 110 or the memory 106. The software is loaded into the computer system 100 from a computer readable medium,

and executed by the computer system 100. Thus, for example, the software 133 may be stored on an optically readable disk storage medium (e.g., CD-ROM) 125 that is read by the optical disk drive 112. A computer readable medium having such software or computer program recorded on it is a computer program product. The use
5 of the computer program product in the computer system 100 preferably effects an apparatus for reconstructing a 3D face image from a single 2D face image.

In some instances, the application programs 133 may be supplied to the user encoded on one or more CD-ROMs 125 and read via the corresponding drive 112, or alternatively may be read by the user from the networks 120 or 122. Still further, the
10 software can also be loaded into the computer system 100 from other computer readable media. Computer readable storage media refers to any non-transitory tangible storage medium that provides recorded instructions and/or data to the computer system 100 for execution and/or processing. Examples of such storage media include floppy disks, magnetic tape, CD-ROM, DVD, Blu-ray Disc, a hard disk
15 drive, a ROM or integrated circuit, USB memory, a magneto-optical disk, or a computer readable card such as a PCMCIA card and the like, whether or not such devices are internal or external of the computer module 101. Examples of transitory or non-tangible computer readable transmission media that may also participate in the provision of software, application programs, instructions and/or data to the computer
20 module 101 include radio or infra-red transmission channels as well as a network connection to another computer or networked device, and the Internet or Intranets including e-mail transmissions and information recorded on Websites and the like.

The second part of the application programs 133 and the corresponding code modules mentioned above may be executed to implement one or more graphical user
25 interfaces (GUIs) to be rendered or otherwise represented upon the display 114. Through manipulation of typically the keyboard 102 and the mouse 103, a user of the computer system 100 and the application may manipulate the interface in a functionally adaptable manner to provide controlling commands and/or input to the applications associated with the GUI(s). Other forms of functionally adaptable user
30 interfaces may also be implemented, such as an audio interface utilizing speech prompts output via the loudspeakers 117 and user voice commands input via the microphone 180.

Fig. 1B is a detailed schematic block diagram of the processor 105 and a “memory” 134. The memory 134 represents a logical aggregation of all the memory modules (including the HDD 109 and semiconductor memory 106) that can be accessed by the computer module 101 in Fig. 1A.

5 When the computer module 101 is initially powered up, a power-on self-test (POST) program 150 executes. The POST program 150 is typically stored in a ROM 149 of the semiconductor memory 106 of Fig. 1A. A hardware device such as the ROM 149 storing software is sometimes referred to as firmware. The POST program 150 examines hardware within the computer module 101 to ensure proper
10 functioning and typically checks the processor 105, the memory 134 (109, 106), and a basic input-output systems software (BIOS) module 151, also typically stored in the ROM 149, for correct operation. Once the POST program 150 has run successfully, the BIOS 151 activates the hard disk drive 110 of Fig. 1A. Activation of the hard disk drive 110 causes a bootstrap loader program 152 that is resident on the hard disk
15 drive 110 to execute via the processor 105. This loads an operating system 153 into the RAM memory 106, upon which the operating system 153 commences operation. The operating system 153 is a system level application, executable by the processor 105, to fulfil various high level functions, including processor management, memory management, device management, storage management, software application
20 interface, and generic user interface.

 The operating system 153 manages the memory 134 (109, 106) to ensure that each process or application running on the computer module 101 has sufficient memory in which to execute without colliding with memory allocated to another process. Furthermore, the different types of memory available in the system 100 of
25 Fig. 1A must be used properly so that each process can run effectively. Accordingly, the aggregated memory 134 is not intended to illustrate how particular segments of memory are allocated (unless otherwise stated), but rather to provide a general view of the memory accessible by the computer system 100 and how such is used.

 As shown in Fig. 1B, the processor 105 includes a number of functional
30 modules including a control unit 139, an arithmetic logic unit (ALU) 140, and a local or internal memory 148, sometimes called a cache memory. The cache memory 148 typically includes a number of storage registers 144 - 146 in a register section. One or more internal busses 141 functionally interconnect these functional modules. The

processor 105 typically also has one or more interfaces 142 for communicating with external devices via the system bus 104, using a connection 118. The memory 134 is coupled to the bus 104 using a connection 119.

5 The application program 133 includes a sequence of instructions 131 that may include conditional branch and loop instructions. The program 133 may also include data 132 which is used in execution of the program 133. The instructions 131 and the data 132 are stored in memory locations 128, 129, 130 and 135, 136, 137, respectively. Depending upon the relative size of the instructions 131 and the memory locations 128-130, a particular instruction may be stored in a single memory
10 location as depicted by the instruction shown in the memory location 130. Alternately, an instruction may be segmented into a number of parts each of which is stored in a separate memory location, as depicted by the instruction segments shown in the memory locations 128 and 129.

In general, the processor 105 is given a set of instructions which are executed
15 therein. The processor 1105 waits for a subsequent input, to which the processor 105 reacts to by executing another set of instructions. Each input may be provided from one or more of a number of sources, including data generated by one or more of the input devices 102, 103, data received from an external source across one of the networks 120, 102, data retrieved from one of the storage devices 106, 109 or data
20 retrieved from a storage medium 125 inserted into the corresponding reader 112, all depicted in Fig. 1A. The execution of a set of the instructions may in some cases result in output of data. Execution may also involve storing data or variables to the memory 134.

The disclosed reconstruction of a 3D face image arrangements use input
25 variables 154, which are stored in the memory 134 in corresponding memory locations 155, 156, 157. The reconstruction of a 3D face image arrangements produce output variables 161, which are stored in the memory 134 in corresponding memory locations 162, 163, 164. Intermediate variables 158 may be stored in memory locations 159, 160, 166 and 167.

30 Referring to the processor 105 of Fig. 1B, the registers 144, 145, 146, the arithmetic logic unit (ALU) 140, and the control unit 139 work together to perform sequences of micro-operations needed to perform "fetch, decode, and execute" cycles

for every instruction in the instruction set making up the program 133. Each fetch, decode, and execute cycle comprises:

(a) a fetch operation, which fetches or reads an instruction 131 from a memory location 128, 129, 130;

5 (b) a decode operation in which the control unit 139 determines which instruction has been fetched; and

(c) an execute operation in which the control unit 139 and/or the ALU 140 execute the instruction.

Thereafter, a further fetch, decode, and execute cycle for the next instruction
10 may be executed. Similarly, a store cycle may be performed by which the control unit 139 stores or writes a value to a memory location 132.

Each step or sub-process in the processes of Figs. 2 to 17 is associated with one or more segments of the program 133 and is performed by the register section 144, 145, 147, the ALU 140, and the control unit 139 in the processor 105
15 working together to perform the fetch, decode, and execute cycles for every instruction in the instruction set for the noted segments of the program 133.

The method of reconstructing a 3D face image from a sample 2D face image may alternatively be implemented in dedicated hardware such as one or more integrated circuits performing the functions or sub functions of the 3D face image
20 reconstruction method. Such dedicated hardware may include graphic processors, digital signal processors, or one or more microprocessors and associated memories.

Fig. 2 is a flow chart of a method 200 for reconstructing a 3D face image from a sample 2D face image. The sample 2D face image may be in any pose (e.g., frontal
25 face image, profile face image, etc).

Transforming and Projecting a Generic 3D Face Image into a Generic 2D Face Image (Step 202)

The method 200 commences with step 202 whereby a generic 3D face image
30 is transformed (i.e., rotated, translated, and scaled) and projected into a generic 2D face image to match the pose of the sample 2D face image.

The Generic 3D Face Image

Typically, a generic 3D face image may be obtained from a commercially available third party face modeling software, such as FaceGen of Toronto, Canada (www.facegen.com) or the like. An example of a frontal part of a generic 3D face image is shown in Fig. 3. The 3D generic face image 300 consists of separate meshes for the eyes 302 and the skin 306. In addition, the 3D generic face image 300 also includes different texture for the face and the back of the head. In this particular example, the number of vertices 310 for each eye is 171, and the number of vertices 310 for the whole skin (i.e., the face and the back of the head) is 5905. The number of vertices 310 for the face area only is 3905. In other embodiments, the number of vertices 310 for the eyes and the whole skin may be increased or reduced depending on desired level of reconstruction accuracy, speed of computing time, etc.

The transformation of the generic 3D face image 300 involves the face area, on which feature points have been predetermined. As shown in Fig. 4, the feature points consists of face feature points 410, 415, 417 and face contour points 420, 425.

The face feature points 410 (i.e., all the light-coloured points) are used in the matching of the generic 3D face image 300 to the sample 2D face image. Conversely, the face feature points 415, 417 are not used in the matching. Feature points 415 will be used in the next step where the generic 3D face image 300 is deformed, whilst feature points 417 will be used for determining the gaze of the generic 3D face image 300.

The generic 3D face image 300 also includes possible face contour points 420. The face contour points are determined from the possible face contour points 420 after the generic 3D face image 300 has been projected into 2D space, which is after step 610 of method 600, which will be discussed hereinafter. This applies to both frontal and non-frontal poses. Depending on the pose of the sample 2D face image, different number of face contour points 420 is determined on the temple, upper, and lower cheeks. In the example of a frontal face pose, shown in Fig. 5 as item 510, one face contour point 420 in each of the six locations 421 are determined: the left-most point 422 (smallest x -coordinate value) for the locations on the left side of the face and the right-most point 423 (greatest x -coordinate value) for the locations on the right side of the face. In total, six face contour points 420 and five face contour points 425 (i.e.,

two points at the lower part of the ears and three points on the chin) will be determined for the entire face of the frontal face pose 510.

In the example of a non-frontal face pose, shown in Fig. 5 as item 520, one face contour point 420 is determined in each of the three locations 421: the left-most point 422 or the right-most point 423, depending on the pose. For example, if the face is facing left (as per face 520), the left-most point 422 is chosen in each of the location 421 resulting in a total of three face contour points 420 for the whole face.

The face contour points 425 are fixed points that will be used in the next step where the generic 3D face image 300 is deformed.

The Sample 2D Face Image

A sample 2D face image may be obtained via the scanner 126, the camera 127, the HDD 110, disk storage medium 125, or the communications network 120, 122. Upon obtaining the sample 2D face image, feature points 540 are located in the sample 2D face image, as shown in Fig. 5. Currently, these feature points are located manually.

In the example of Fig. 5, there are forty nine feature points 540 for the frontal face image 510 and forty four feature points 540 for the non-frontal face image 520. From these feature points 540, the 2 feature points of the eyes 530 are used to modify the generic 3D face image's eyes 417.

Estimating the Head Pose of the sample 2D Face Image and Projecting the Generic 3D Face Image into a 2D Face Image.

The head pose of the sample 2D face image 510, 520 is estimated by using the method described, for example, in I. Shimizu, Z. Zhang, S. Akamatsu, K. Deguchi, "Head Pose Determination from One Image Using a Generic Model," Proc. of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition (FG), pp. 100-105, 1998, the content of which is incorporated herein in its entirety by reference.

Fig. 6 is a flow chart of the method 600 for transforming and projecting the generic 3D face image 300 to match the pose of the sample 2D face image 510, 520. Fig. 6 commences with step 602 whereby the feature points 410 of the generic 3D face image 300 is transformed (i.e., scaled (**K**), rotated (**R**), and translated (**t**)) to obtain transformed feature points. The equation for this transformation is:

$$\mathbf{p}_F^t = \mathbf{K} (\mathbf{R} \mathbf{p}_F + \mathbf{t}) \quad (\text{eqn. 1})$$

where:

$\mathbf{p}_F^t$ = transformed feature points;

$\mathbf{p}_F$ = feature points (410);

5

$\mathbf{K}$ = scaling value;

$\mathbf{R}$ = rotational value;

$\mathbf{t}$ = translational value;

The initial values of $\mathbf{K}$ and $\mathbf{R}$ are equal to the identity matrix ($\mathbf{I}$) and $\mathbf{t}$ is $[0 \ 0 \ 0]^T$.

10

Step 602 then proceeds to step 604.

At step 604, the transformed points ($\mathbf{p}_F^t$) are projected into 2D space ($\mathbf{w}_F^t = [u_F^t \ v_F^t]^T$) with an orthographic projection matrix $\mathbf{G}$:

$$u_F^t = \frac{\mathbf{g}_1^T \mathbf{p}_F^t + \mathbf{g}_{14}}{\mathbf{g}_3^T \mathbf{p}_F^t + \mathbf{g}_{34}} \quad (\text{eqn. 2})$$

$$v_F^t = \frac{\mathbf{g}_2^T \mathbf{p}_F^t + \mathbf{g}_{24}}{\mathbf{g}_3^T \mathbf{p}_F^t + \mathbf{g}_{34}} \quad (\text{eqn. 3})$$

15

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1^T & \mathbf{g}_{14} \\ \mathbf{g}_2^T & \mathbf{g}_{24} \\ \mathbf{g}_3^T & \mathbf{g}_{34} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{eqn. 4})$$

where:

20

$\mathbf{w}_F^t$ is the projected points of the generic 3D face image 300 in 2D space;

u_F^t and v_F^t are the 2D coordinates of the transformed and 2D projected feature points 410;

$$\mathbf{g}_1^T = [1 \ 0 \ 0];$$

$$\mathbf{g}_2^T = [0 \ -1 \ 0]; \text{ and}$$

25

$$\mathbf{g}_3^T = [0 \ 0 \ 0].$$

The second element of $\mathbf{g}_2^T$ is -1 since the direction of the y-axis in the image plane is reversed (from top to bottom). For simple orthographic projection to the xy-

plane where the direction of the y -axis is upwards (bottom to top), the second element of $\mathbf{g}_2^T$ is 1 (small y -value in xyz -plane is projected to small y -value in the xy -plane). However, since the direction of the y -axis in the image plane is downwards (top to bottom), this element is -1 (small y -value in xyz -plane is projected to large y -value in the image xy -plane).

Upon completion of the orthographic projection of the generic 3D face image 300, the method 600 proceeds to step 606.

At step 606, the values of $\mathbf{K}$, $\mathbf{R}$, and $\mathbf{t}$ are adjusted to minimize the error value E between the projected points $\mathbf{w}_F^I$ and corresponding points of the sample 2D face image ($\mathbf{w}_F^I$). The equation for the error value E is:

$$E = \sum |\mathbf{w}_F^I - \mathbf{w}_F^t| \quad (\text{eqn. 5})$$

The error minimizing process is a non-linear optimization, which comprises of 9 parameters: 3 parameters for $\mathbf{K}$ (scaling on the x , y , and z axes), 3 for $\mathbf{R}$ (rotation around the x , y , and z axes) and 3 for $\mathbf{t}$ (translation on the x , y , and z axes). An example of an optimization algorithm is a subspace trust region method, which is based on the interior-reflective Newton method described in T. F. Coleman, Y. Li, "On the Convergence of Reflective Newton Methods for Large-Scale Nonlinear Minimization Subject to Bounds," Mathematical Programming, vol. 67, no. 2, pp. 189-224, 1994; and T. F. Coleman, Y. Li, "An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds," SIAM Journal on Optimization, vol. 6, pp. 418-445, 1996, the contents of which are incorporated herein in their entirety by reference.

Upon obtaining optimal values of $\mathbf{K}_o$, $\mathbf{R}_o$ and $\mathbf{t}_o$, step 606 proceeds to step 608.

At step 608, the optimal values are $\mathbf{K}_o$, $\mathbf{R}_o$ and $\mathbf{t}_o$ are used to compute the optimal projection matrix $\mathbf{H}$:

$$\mathbf{H} = \mathbf{G} \begin{bmatrix} \mathbf{K}_o \mathbf{R}_o & \mathbf{K}_o \mathbf{t}_o \\ 0^T & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{h}_1^T & h_{14} \\ \mathbf{h}_2^T & h_{24} \\ \mathbf{h}_3^T & h_{34} \end{bmatrix} \quad (\text{eqn. 6})$$

Step 608 then proceeds to step 610.

At step 610, the generic 3D face image mesh 302, 306 is projected into the 2D space ($\mathbf{w} = [u \ v]^T$) using $\mathbf{H}$:

$$u = \frac{\mathbf{h}_1^T \mathbf{p} + h_{14}}{\mathbf{h}_3^T \mathbf{p} + h_{34}} \quad (\text{eqn. 7})$$

$$v = \frac{\mathbf{h}_2^T \mathbf{p} + h_{24}}{\mathbf{h}_3^T \mathbf{p} + h_{34}} \quad (\text{eqn. 8})$$

The resulting projected generic 2D face image mesh 702 is shown in Fig. 7.

10 The generic 3D face image projected points 710 (of the face feature points 410) are close to the sample 2D face image corresponding points 540. Method 600 concludes at step 610, which also concludes step 202 of method 200.

Deforming the Projected Generic 2D Face Image (Step 204)

15 Method 200 then proceeds to step 204 for deforming the projected 2D face image mesh 702, with reference to Figs 7 and 8 considered together. (NB: In Fig. 7, only the projected points 710 are shown.) Before deforming the projected 2D face image mesh 702, the face contour points (not shown) of the projected 2D face image 702 are determined from the 2D projection of the possible face contour points 420.

20 The determined contour points 425 and the face feature points 415, together with the projected feature points 710, form a set of points required for the deformation.

The deformation of the projected generic 2D face image mesh 702 consists in computing the coordinates of non-feature points based on the displacement between the feature points 710 of the projected generic 2D face image mesh 702 and the

25 corresponding feature points 540 of the sample 2D face image 510, 520. The non-feature points are the rest of the mesh points. These non-feature points are located at the intersection between the lines of the mesh (not marked explicitly like the feature points).

The displacement between projected feature points 710 (and the projected

30 versions of 415 and 425, and the selected points of 420) and sample feature points 540 is determined using a Dirichlet Free-Form Deformation (DFFD) method, which is

a non-linear deformation method as described in L. Moccozet, N. M.-Thalmann, “*Dirichlet Free-Form Deformation and Their Application to Hand Simulation*,” Proc. of Computer Animation 1997, pp. 93-102, 1997, the content of which is incorporated herein in its entirety by reference. Based on the DFFD method, the displacement of a point 710 is a weighted sum of the displacement of the point’s neighbours. The weights are determined by the natural coordinates of the point 710 relative to its natural neighbors.

The DFFD Method

First, the projected generic 2D face image mesh 702 is divided into several regions. In this particular example, the projected generic 2D face image mesh 702 is divided into 13 regions: around the eyes (2 regions) (regions not shown), nose 820, upper lip - visible and invisible part (2 regions) (regions not shown), lower lip - visible and invisible part (2 regions) (regions not shown), between the nose and the upper lip (regions not shown), between the lower lip and the chin (regions not shown), left cheek 810, right cheek (region not shown), forehead (region not shown), and neck (region not shown). The DFFD method is then performed in each region separately.

For example, for a set of 2D points A, B, C, D, E, X (shown in Fig. 9), the corresponding displacements are Δw_A to Δw_E . The DFFD method computes the displacement of X (Δw_X) by first determining the Voronoi diagram of the set of points with and without the point X . In Fig. 9, the solid line is the Voronoi diagram with point X and the dashed line is the Voronoi diagram without point X . The Voronoi diagram with point X is bounded by nodes P, Q, R, S , and T , whilst the Voronoi diagram without point X is bounded by nodes K, L , and M .

After determining the Voronoi diagrams, the DFFD method determines the natural neighbours of point X . In this example, points A to E are the natural neighbours.

Upon determination of point X ’s natural neighbours, the DFFD method computes the natural coordinates of point X corresponding to each neighbor. For example, the natural coordinate of point X corresponding to point A is the ratio of 2 areas:

$$n_A = \frac{PKQ}{PQRST} \quad (\text{eqn. 9})$$

where:

n_A = natural coordinate of point X in relation to point A ;

PKQ = the area bounded by nodes P , K , and Q ; and

5 $PQRST$ = the area bounded by nodes P , Q , R , S , and T .

The total sum of the natural coordinates of X corresponding to the natural neighbours (n_A to n_E) is 1. The natural coordinate of X is a barycentric coordinate, in accordance with R. Sibson, "*A Vector Identity for the Dirichlet Tessellation*,"
 10 Mathematical Proceedings of the Cambridge Philosophical Society, vol. 87, no. 1, pp. 151-155, 1980, the content of which is incorporated herein in its entirety by reference.

$$X = n_A A + n_B B + n_C C + n_D D + n_E E \quad (\text{eqn. 10})$$

Upon obtaining the natural coordinate of point X , the displacement of point X
 15 $(\Delta \mathbf{w}_X)$ can be obtained using:

$$\Delta \mathbf{w}_X = \sum_j n_j \Delta \mathbf{w}_j \quad (\text{eqn. 11})$$

where j is the natural neighbours of point X .

Deformed generic 2D face images 1010 are produced by applying the DFFD
 20 method to the projected generic 2D face image mesh 702. The deformed generic 2D face images 1010 are shown in Fig. 10.

Step 204 concludes and method 200 proceeds to step 206.

Constructing a 3D Face Image of the Sample 2D Face Image (Step 206)

25 At step 206, a sample 3D face image of the sample 2D face image 510, 520 is reconstructed. The process of constructing the sample 3D face image is outlined by method 1100. Method 1100 is depicted in Fig. 11 which shows a flow chart for constructing a sample 3D face image from the sample 2D face image. Method 1100 commences with step 1102 whereby depth information is obtained by adjusting the
 30 optimal scaling parameter $\mathbf{K}_0$, and the obtained depth information is inserted into the deformed generic 2D face image 1010.

Inserting Depth Information into the Deformed Generic 2D Face Image

The deformed generic 2D face images 1010 are without any depth information. Thus, the deformed generic 2D face images 1010 still retain the profile of the generic 3D face image 300 (e.g., nose length of the generic 3D face image, cheek protrusion of the generic 3D face image, etc).

In order to obtain the depth information of the sample 2D face image 510, 520, the optimal scaling parameter $\mathbf{K}_o$ is adjusted to include an estimated depth information of the sample 2D face image 510, 520.

The optimal scaling parameter $\mathbf{K}_o$, which is obtained at step 606, only contains the scaling values for the x and y -axes:

$$\mathbf{K}_o = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{eqn.12})$$

Thus, there is no scaling performed in the z -axis when $\mathbf{K}_o$ is applied to the generic 3D face image 300.

In order to adjust the depth of the transformed 3D face image 300, $\mathbf{K}_o$ is adjusted as follows:

$$\mathbf{K}_o = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix}; \quad k_z = \frac{k_x + k_y}{2} \quad (\text{eqn. 13})$$

Using the new $\mathbf{K}_o$, the generic 3D face image 300 is transformed using eqn. 1. The equation, which has been reproduced here for ease of reference, is as follows:

$$\mathbf{p}_1 = \mathbf{K}_o (\mathbf{R}_o \mathbf{p}_F + \mathbf{t}_o) \quad (\text{eqn. 14})$$

where $\mathbf{p}_1$ = transformed feature points with depth information;

$\mathbf{K}_o$ = optimal scaling parameter with z -axis scaling value;

$\mathbf{R}_o$ = optimal rotational parameter;

$\mathbf{p}_F$ = feature points (410); and

$\mathbf{t}_o$ = optimal translational parameter.

The resulting transformed generic 3D face image has the same pose as the sample 2D face image 510, 520. The z-coordinates of $\mathbf{p}_1$ are then taken as the depth of the deformed generic 2D face image 1010, resulting in $\mathbf{m}_1$ which is the coordinates of the mesh of the transformed 3D face mesh. The parameter $\mathbf{m}_1$ is then rotated with $\mathbf{R}_o^{-1}$ to normalize the pose back to the frontal pose of the generic 3D model.

Step 1102 concludes and method 1100 proceeds to step 1104.

Obtaining Symmetry for a reconstructed sample 3D Face Image

At step 1104, symmetry of the reconstructed sample 3D face image of the sample 2D face image 510, 520 is obtained. The step begins by flipping (i.e., mirrored in the y-axis) the deformed generic 2D face image 1010 and merging the flipped deformed generic 2D face image with the z-coordinates of $\mathbf{p}_2$. The parameter $\mathbf{p}_2$ is obtained from:

$$\mathbf{p}_2 = \mathbf{K}_o (\mathbf{R}'_o \mathbf{p}_F + \mathbf{t}_o) \quad (\text{eqn. 15})$$

where $\mathbf{p}_2$ = transformed feature points of the generic 3D face image;

$\mathbf{K}_o$ = optimal scaling parameter;

$\mathbf{R}'_o$ = optimal rotational parameter for the flipped deformed generic 2D face image 1010 that is obtained from the components of $\mathbf{R}_o$:

$$\mathbf{R}'_o = \mathbf{R}_{o,x} \mathbf{R}_{o,y}^{-1} \mathbf{R}_{o,z}^{-1}$$

where $\mathbf{R}_{o,i}$ is the optimal rotation matrix around the i -axis;

$\mathbf{p}_F$ = feature points (410); and

$\mathbf{t}_o$ = optimal translational parameter.

In this particular example, it is assumed that flipping the deformed generic 2D face image 1010 does not involve any rotation around the x-axis (i.e., the head is not tilted backward or forward). The result of eqn. 15 is $\mathbf{m}_2$, which is in the pose of the sample 2D face image 510, 520. The parameter $\mathbf{m}_2$ is the coordinates of the mesh of the transformed 3D face image. The parameter $\mathbf{m}_2$ is then rotated with $\mathbf{R}'_o^{-1}$ to normalize the pose back to the frontal pose of the generic 3D model.

Step 1104 concludes, and method 1100 proceeds to step 1106.

Constructing a 3D Face Image of the Sample 2D Face Image

At step 1106, the sample 3D face image of the sample 2D face image 510, 520 is reconstructed. The sample 3D face image ($\mathbf{m}_0$) (wherein $\mathbf{m}_0$ is the coordinates of the mesh of the transformed 3D face mesh) is constructed from $\mathbf{m}_1$ and $\mathbf{m}_2$ as follows:

5

$$\mathbf{m}_0 = \frac{3}{k_x + k_y + k_z} \left(\frac{\mathbf{R}_o^{-1} \mathbf{m}_1 + \mathbf{R}_o'^{-1} \mathbf{m}_2}{2} - \mathbf{t}_o \right) \quad (\text{eqn. 16})$$

Step 1106 concludes, which also concludes method 1100 and step 206. Method 200 continues to step 208.

10

Texturing of the Sample 3D Face Image (Step 208)

At step 208, the sample 3D face image is textured according to the texture of the sample 2D face image 510, 520.

Currently, the texture of the sample 3D face image is the same as the texture of the generic 3D face image 300, as shown in Fig. 3.

15

Step 208 commences by obtaining a texture image of the generic 3D face image superimposed with a generic mesh. The superimposed face image 1202 is typically provided by the commercially available third party face modeling software. In this example, the image 1202 is provided by the FaceGen software.

20

The pixels in each rectangle of the face image 1202 are then replaced by corresponding pixels in the sample 2D face image 510, 520.

In the case where the sample 2D face image is a frontal image 510, the pixels are taken from the area 1302 inside the 2D mesh, as shown in Fig. 13A.

25

In the case where the sample 2D face image is a non-frontal image 520, pixels from half of the face 1304 of the image 520 are taken, as shown in Fig. 13B. The image 520 is then flipped, and pixels from the other half of the face 1306 are taken, as shown in Fig. 13C.

The resulting image of the pixel substitution is shown in Fig. 14A for the frontal image 510 and in Fig. 14B for the non-frontal image 520.

Method 200 then proceeds to step 210.

30

Constructing the Eye Models of the Sample 3D Face Image (Step 210)

At step 210, the eyes of the sample 3D face image are constructed. Step 210 commences by creating a new 3D face image mesh 1502, as shown in Fig. 15A, and generating meshes for both eyes 1504. The new 3D eye mesh 1504 is constructed by
5 deforming the eye meshes of the generic 3D face image with DFFD, in 3D space, using 8 control points for each eye. The DFFD algorithm in 3D space is the same as in 2D space. However, in 3D space, the natural coordinate is a ratio between 2 volumes instead of 2 areas.

To generate texture for the eye meshes 1504, the generic 3D eyes are projected
10 into 2D space to fit the eye regions and to match the eye gaze of the sample 2D face image 510, 520. This includes computations of the optimal scale and eye movement parameters of the generic 3D face image 300 so that the projection of the generic 3D eyes, using the optimal projection matrix H , fits the eye regions of the sample 2D face image 510, 520.

15 The computation of the optimal scale is the same as the computation of estimation of the head pose, which includes searching the optimal value to minimize the error between the projected control points of the generic 3D eyes (1508) and the control points of the new skin mesh around the eyes (1510).

The optimal eye movement parameters is computed by searching for the
20 optimal value to minimize the error between the projected control points at the center of the eyes 1520 and the feature points 1530, in Fig. 15B, on the eyes of the sample 2D face image 510, 520.

Fig. 15C shows the result of the optimal projection. The texture images for both eyes are generated by replacing the pixels of the generic eye meshes 1504 with
25 the pixels taken from the eye meshes 1540 of the sample 2D face image 510, 520.

After the construction of the new 3D face image mesh 1504 and the textures for the face skin and eyes, the eyes texture is mapped by associating each rectangle of the texture mesh 1540 to corresponding rectangle of the new 3D face image mesh 1504. The resulting sample 3D face images 1610 are shown in Fig. 16A and 16B.

30 Method 200 proceeds to step 212.

**Constructing the Back of the Head of the Sample 3D Face Image
(Step 212)**

At step 212, the back of the head is fitted to the sample 3D face image. The process of fitting the back of the head consists in searching for optimal scaling, rotational, and translational parameters for matching the border vertices 1710 of the sample 3D face image 1610 and the border vertices of the back of the head 1720, as shown in Fig. 17A.

After the optimal parameters have been found, the back of the head is transformed using these optimal parameters, and the border points of the face 1710 and the back of the head 1720 are adjusted by setting their values to their average. The texture for the back of the head is taken from the generic 3D face image 300. The complete sample 3D face image 1750 is shown in Fig. 17B.

The arrangements described are applicable to the computer and data processing industries and particularly for a photo-editing tool to change the face pose in a picture as required. The present invention may also be useful to the police to investigate the picture of a suspect where only one picture is available and it is necessary to have the picture of different poses. Another possible application is entertainment, for example 3D face images can be created for avatars of players in 3D games.

The foregoing describes only some embodiments of the present invention, and modifications and/or changes can be made thereto without departing from the scope and spirit of the invention, the embodiments being illustrative and not restrictive.

CLAIMS:

1. A method for reconstructing a three-dimensional (3D) face image from a sample two-dimensional (2D) face image, said method comprising:
 - 5 transforming a generic 3D face image to substantially match the pose of the 2D face image;
projecting said transformed generic 3D face image into a projected 2D face image;
deforming said projected 2D face image; and
 - 10 constructing a sample 3D face image from said deformed 2D face image based on said sample 2D face image.
2. The method of claim 1, wherein said step of constructing a sample 3D face image comprises:
 - 15 texturing of said sample 3D face image based on texture of said sample 2D face image;
constructing eye models of said sample 3D face image based on eye models of said sample 2D face image; and
constructing the back of a head of said sample 3D face image, said back of the
 - 20 head is constructed from said generic 3D face image.
3. The method of any one of the preceding claims, wherein the transforming and projecting comprise:
 - 25 determining feature points on said generic 3D face image and on said sample 2D face image;
transforming said feature points of said generic 3D face image based on initial scaling, rotation, and translation values;
projecting said transformed feature points of said generic 3D face image into 2D space;
 - 30 optimizing said scaling, rotation, and translation values to minimize an error value, wherein said error value is a difference between said projected and transformed feature points of said generic 3D face image and said feature points of said sample 2D face image;

computing an optimal projection matrix based on said optimized scaling, rotation, and translation values; and

projecting said generic 3D face image into a generic 2D face image using said optimal projection matrix.

5

4. The method of any one of the preceding claims, wherein constructing a sample 3D face image from said deformed 2D face image based on said sample 2D face image comprises:

determining depth information of said sample 2D face image;

10 inserting said determined depth information into said deformed generic 2D face image; and

obtaining symmetry of said deformed 2D face image.

5. The method of any one of the preceding claims, wherein deforming said projected 2D face image further comprises:

15

dividing said projected 2D face image into regions; and

performing a Dirichlet Free-Form Deformation method on each region separately.

20 6. A computer program product having instructions to perform a method for reconstructing a three-dimensional (3D) face image from a sample two-dimensional (2D) face image is stored in a computer readable medium, wherein said method comprises:

25

transforming a generic 3D face image to substantially match the pose of the 2D face image;

projecting said transformed generic 3D face into a projected 2D face image;

deforming said projected 2D face image; and

constructing a sample 3D face image from said deformed and projected 2D face image based on said sample 2D face image.

30

7. The computer program product of claim 6, wherein said constructing a sample 3D face image comprises:

texturing of said sample 3D face image based on texture of said sample 2D face image;

constructing eye models of said sample 3D face image based on eye models of said sample 2D face image; and

5 constructing the back of a head of said sample 3D face image, said back of the head is constructed from said generic 3D face image.

8. The computer program product of either one of claims 6 or 7, wherein transforming and projecting comprises:

10 determining feature points on said generic 3D face image and on said sample 2D face image;

transforming said feature points of said generic 3D face image based on initial scaling, rotation, and translation values;

15 projecting said transformed feature points of said generic 3D face image into 2D space;

optimizing said scaling, rotation, and translation values to minimize an error value, wherein said error value is a difference between said projected and transformed feature points of said generic 3D face image and said feature points of said sample 2D face image;

20 computing an optimal projection matrix based on said optimized scaling, rotation, and translation values; and

projecting said generic 3D face image into a generic 2D face image using said optimal projection matrix.

25 9. The computer program product of any one of claims 6 to 8, wherein constructing a sample 3D face image from said deformed 2D face image based on said sample 2D face image comprises:

determining depth information of said sample 2D face image;

30 inserting said determined depth information into said deformed generic 2D face image; and

obtaining symmetry of said deformed 2D face image.

10. The computer program product of any one of claims 6 to 9, wherein deforming said projected 2D face image further comprises:

dividing said projected 2D face image into regions; and

performing a Dirichlet Free-Form Deformation method on each region

5 separately.

1/22

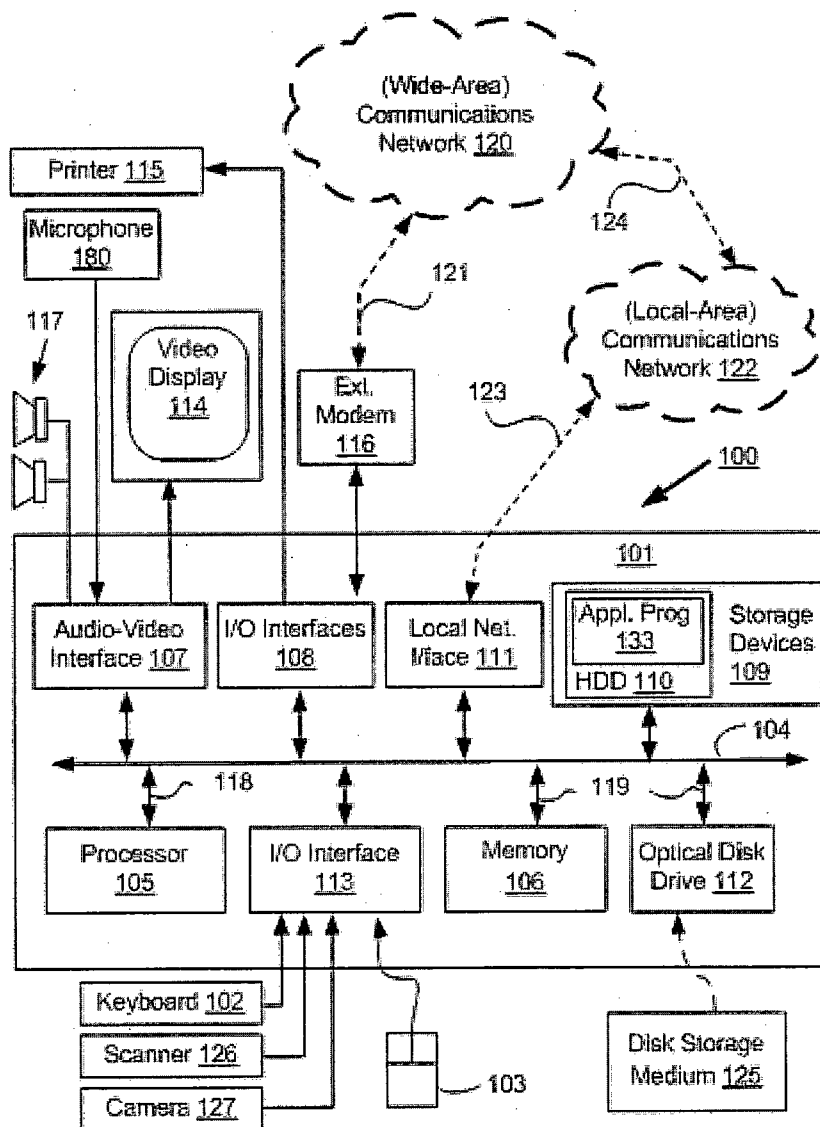


Fig. 1A

2/22

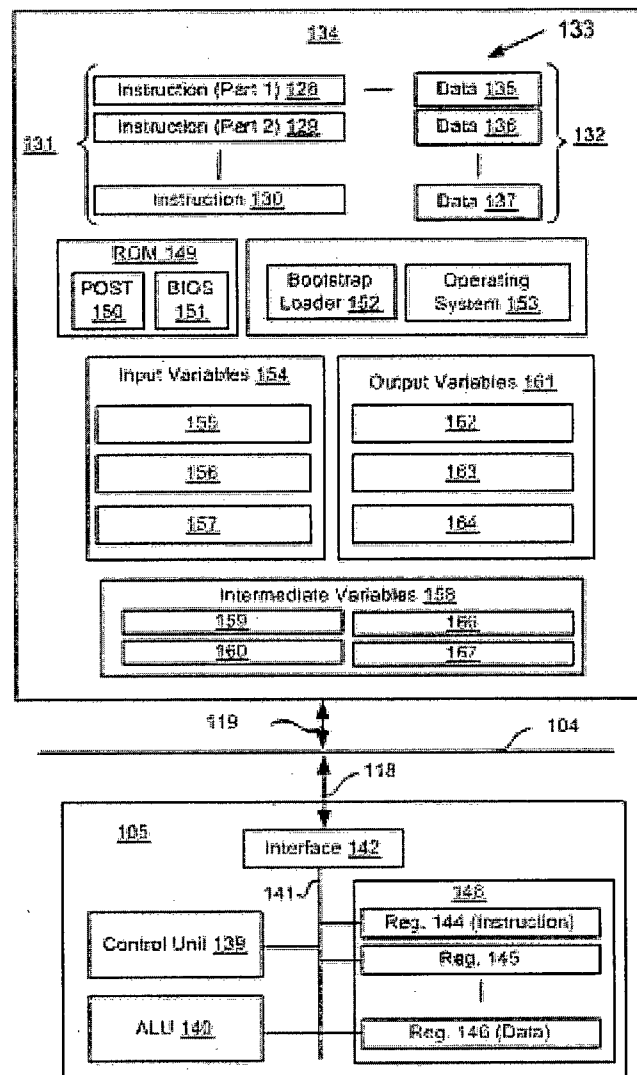


Fig. 1B

3/22

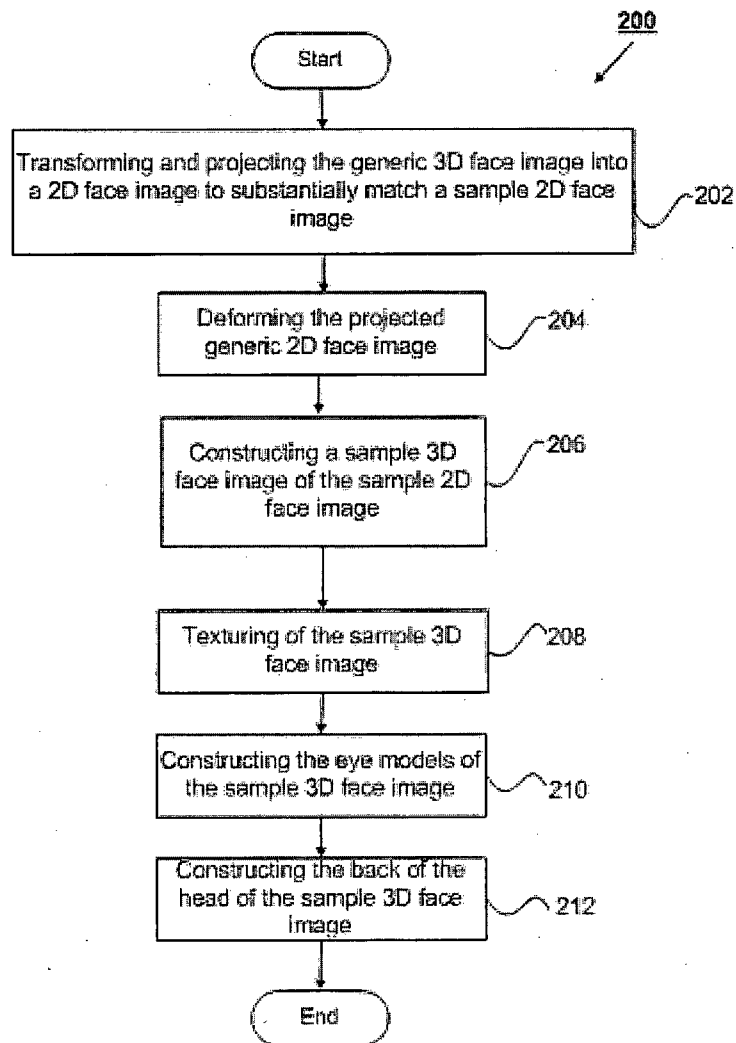


Fig. 2

4/22

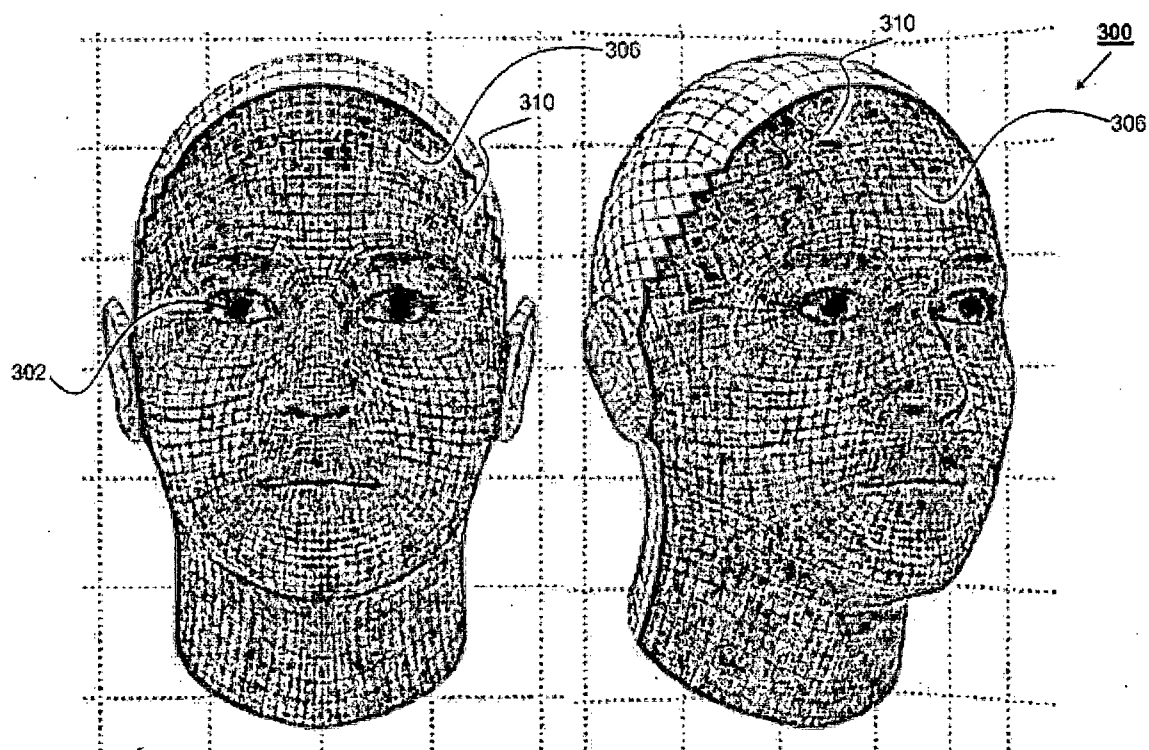


Fig. 3

5/22

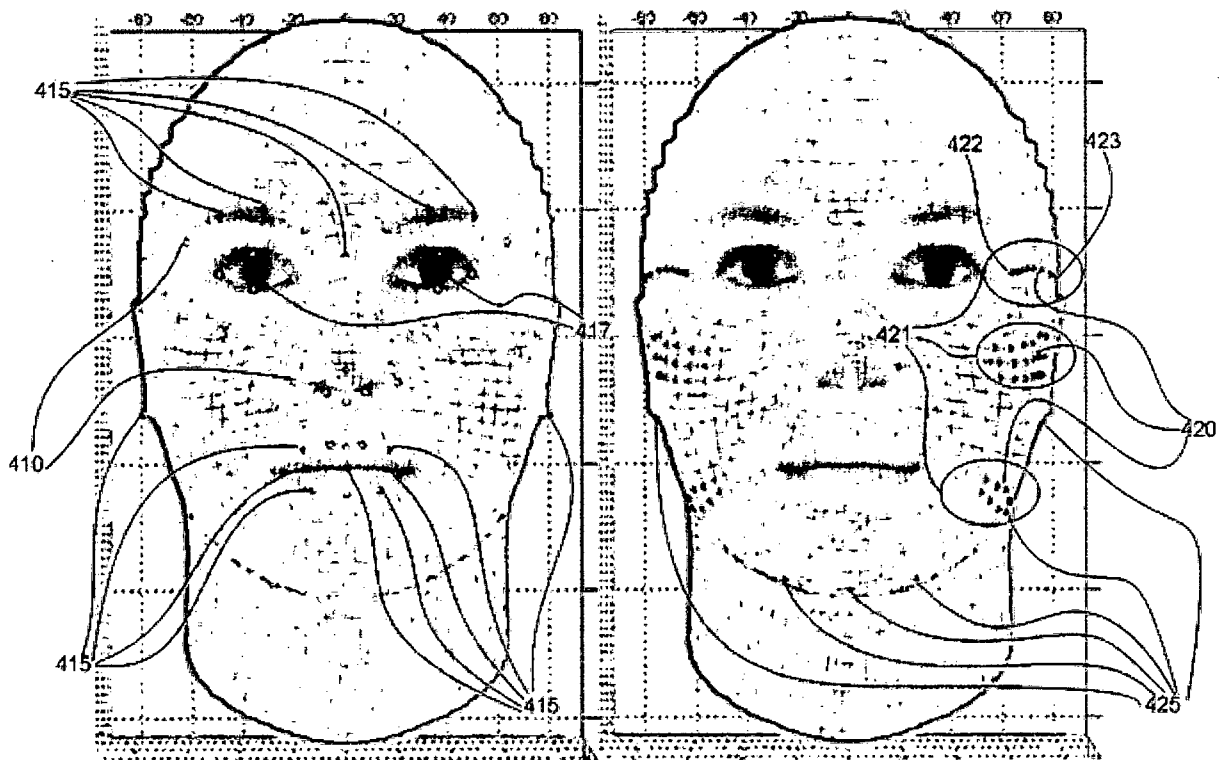


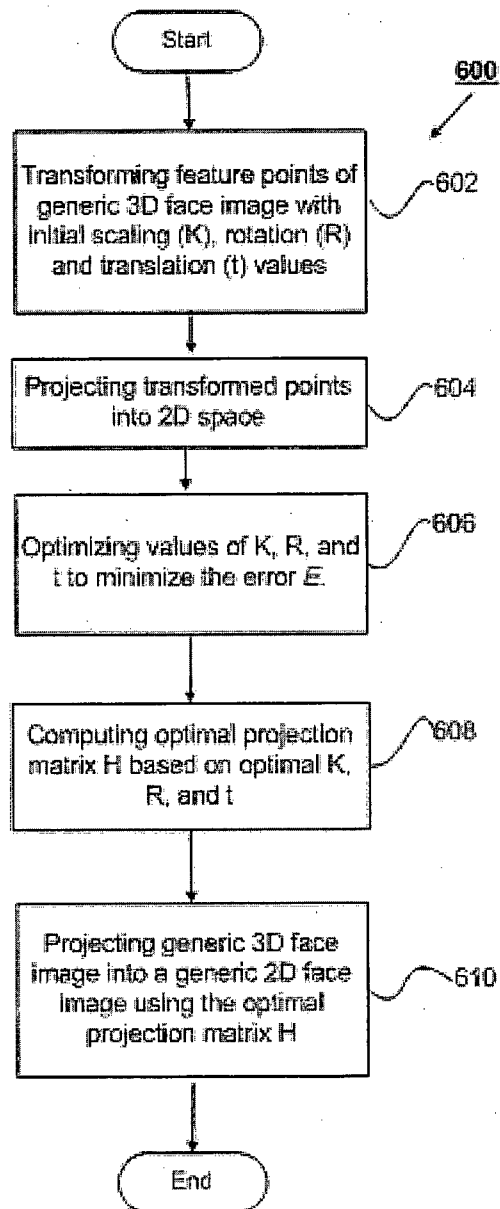
Fig. 4

6/22



Fig. 5

7/22

**Fig. 6**

8/22



Fig. 7

9/22



Fig. 8

10/22

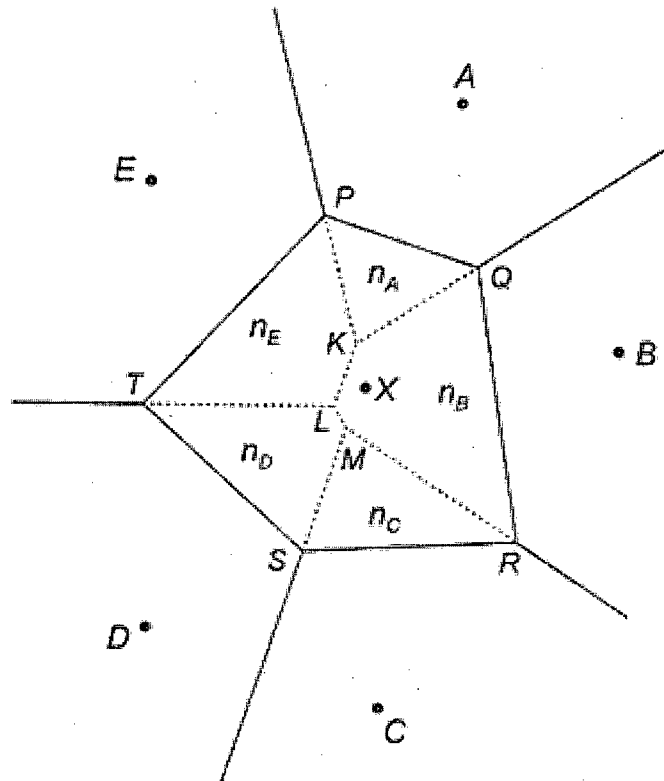


Fig. 9

11/22



Fig. 10

12/22

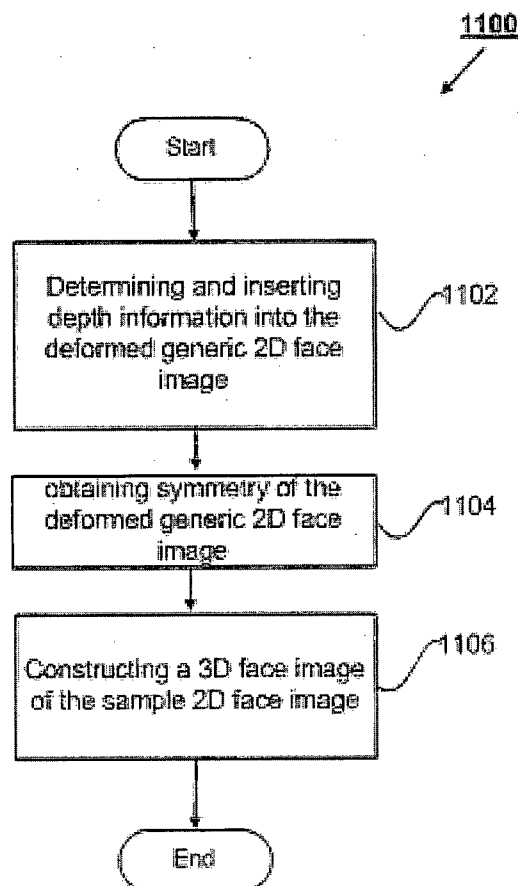


Fig. 11

13/22

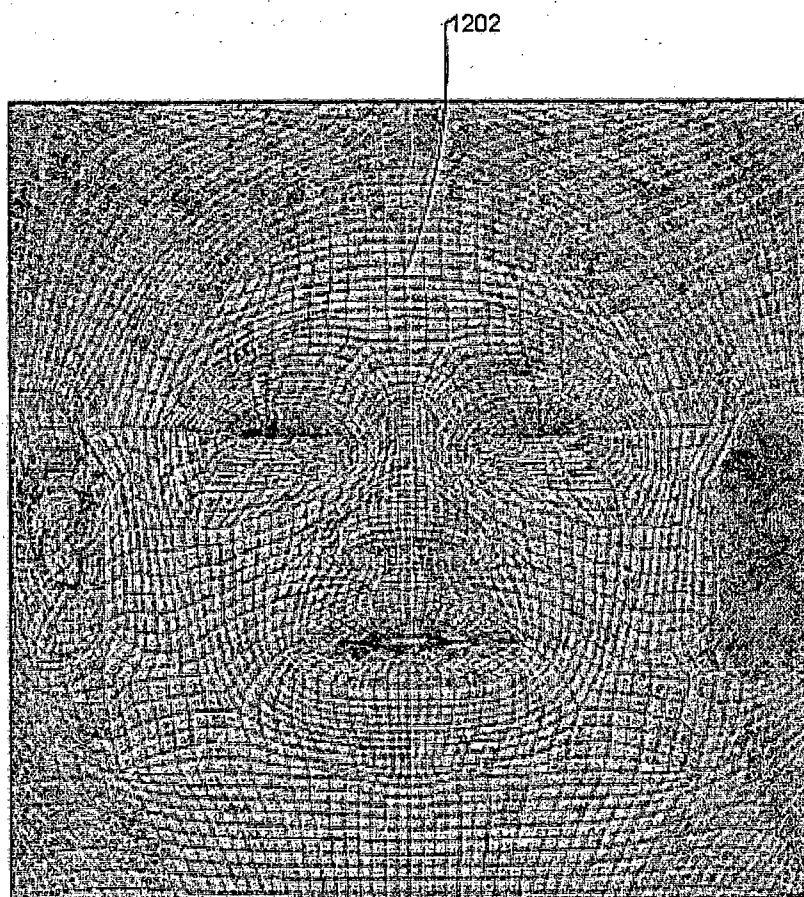


Fig. 12

14/22



Fig. 13A

15/22



Fig. 13B

16/22



Fig. 13C

17/22



Fig. 14A



Fig. 14B

18/22

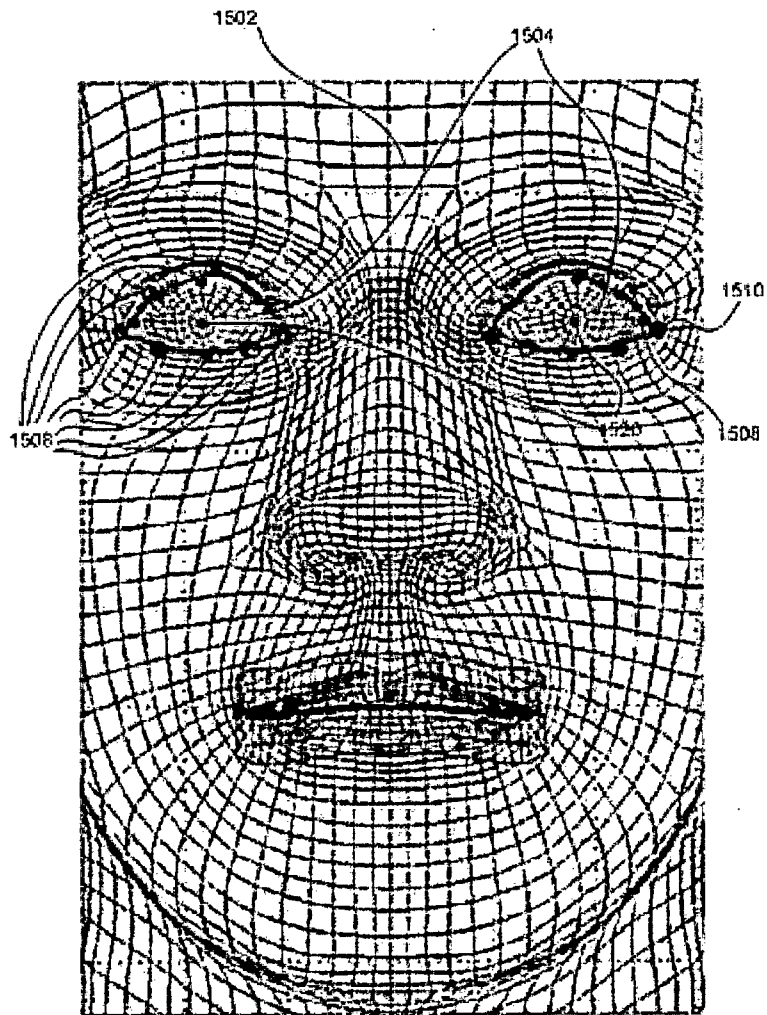


Fig. 15A

19/22



Fig. 15B

20/22



Fig. 15C

21/22

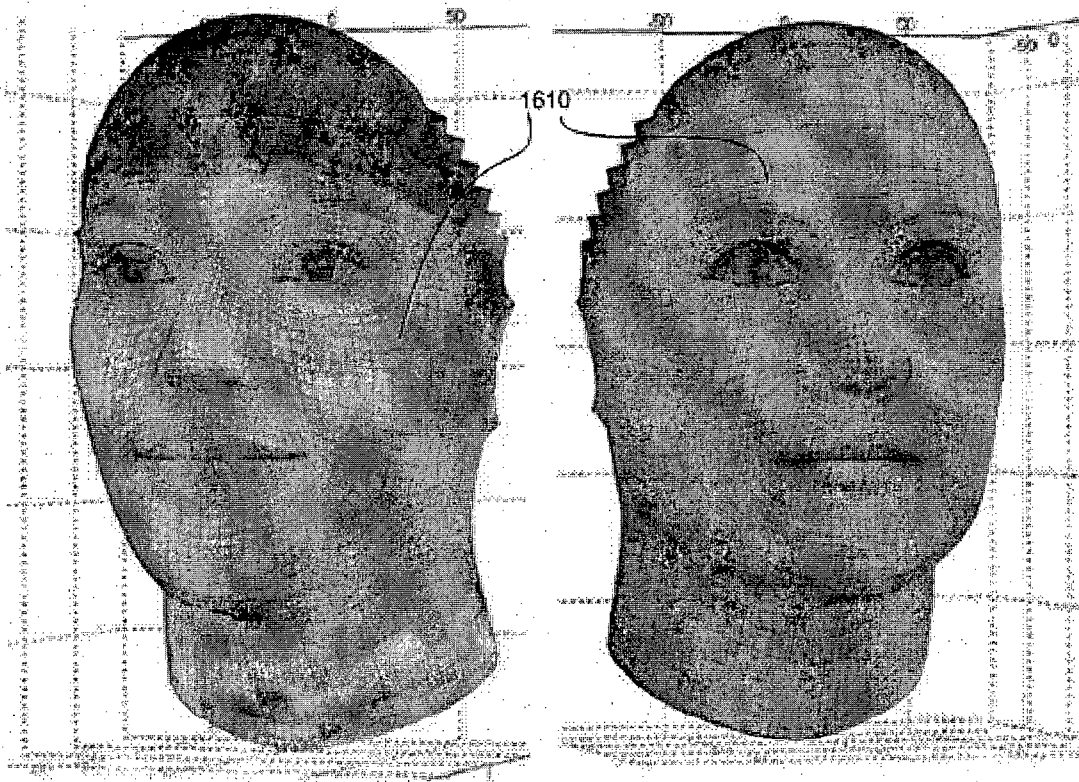


Fig. 16A

Fig. 16B

22/22

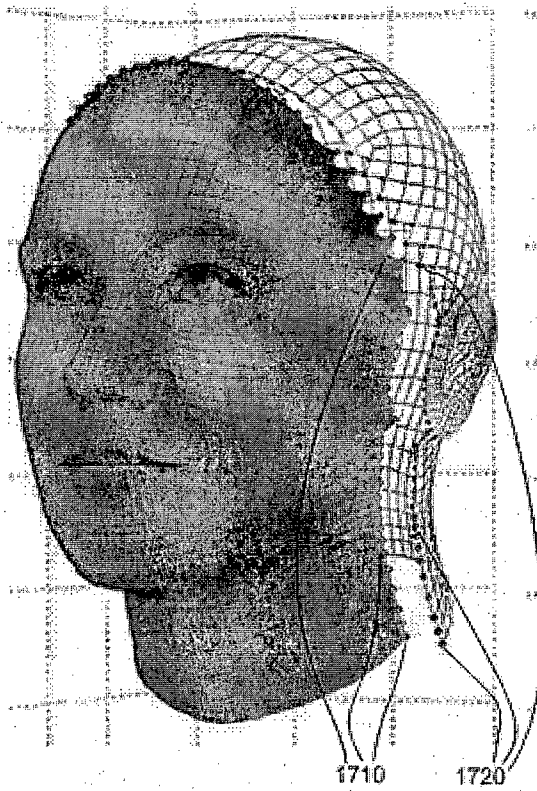


Fig. 17A

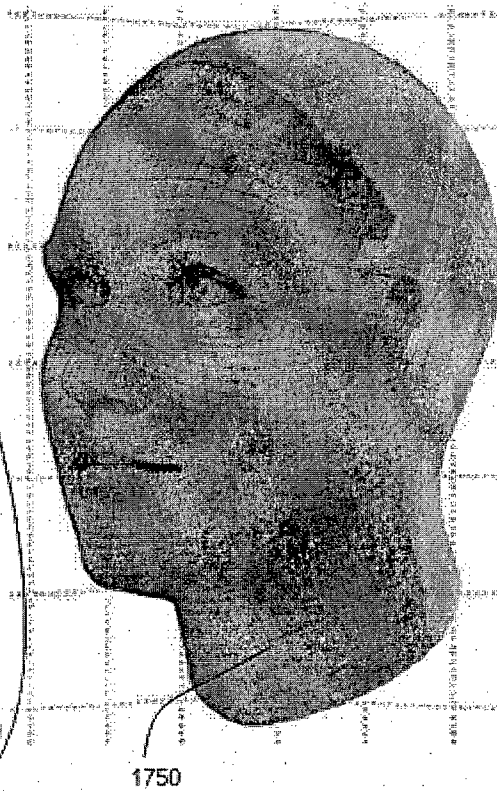


Fig. 17B