



- (51) **International Patent Classification:**
G06T 17/00 (2006.01)
- (21) **International Application Number:**
PCT/US2013/075152
- (22) **International Filing Date:**
13 December 2013 (13.12.2013)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
13/712,998 13 December 2012 (13.12.2012) US
- (71) **Applicant:** MICROSOFT CORPORATION [US/US];
One Microsoft Way, Redmond, WA 98052-6399 (US).
- (72) **Inventors:** ZHANG, Zhengyou; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, WA 98052-6399 (US). CAI, Qin; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, WA 98052-6399 (US). CHOU, Philip, A.; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, WA 98052-6399 (US).
- (81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM,

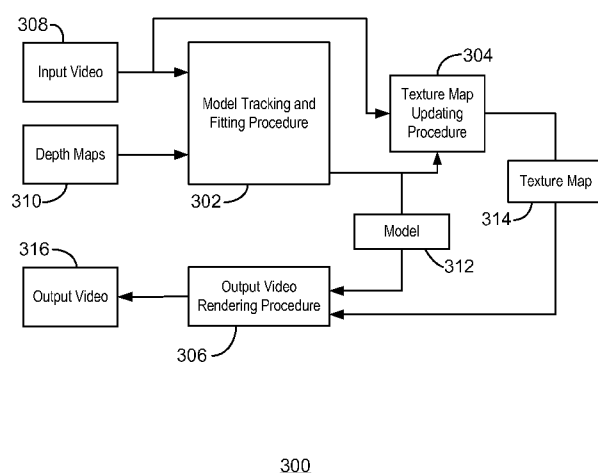
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM,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

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

Declarations under Rule 4.17:

- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

[Continued on next page]

(54) Title: MODEL BASED VIDEO PROJECTION

300

FIG. 3

(57) **Abstract:** A method, system, and computer-readable storage media for model based video projection are provided herein. The method includes tracking an object within a video based on a three-dimensional parametric model via a computing device and projecting the video onto the three-dimensional parametric model. The method also includes updating a texture map corresponding to the object within the video and rendering a three-dimensional video of the object from any of a number of viewpoints by loosely coupling the three-dimensional parametric model and the updated texture map.



Published:

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

MODEL BASED VIDEO PROJECTION

BACKGROUND

[0001] Videos are useful for many applications, including communication
5 applications, gaming applications, and the like. According to current techniques, a video can only be viewed from the viewpoint from which it was captured. However, for some applications, it may be desirable to view a video from a viewpoint other than the one from which it was captured.

SUMMARY

10 [0002] The following presents a simplified summary of the innovation in order to provide a basic understanding of some aspects described herein. This summary is not an extensive overview of the claimed subject matter. It is intended neither to identify key nor critical elements of the claimed subject matter nor to delineate the scope of the subject innovation. Its sole purpose is to present some concepts of the claimed subject matter in a
15 simplified form as a prelude to the more detailed description that is presented later.

[0003] An embodiment provides a method for model based video projection. The method includes tracking an object within a video based on a three-dimensional parametric model via a computing device and projecting the video onto the three-dimensional parametric model. The method also includes updating a texture map
20 corresponding to the object within the video and rendering a three-dimensional video of the object from any of a number of viewpoints by loosely coupling the three-dimensional parametric model and the updated texture map.

[0004] Another embodiment provides a system for model based video projection. The system includes a processor that is configured to execute stored instructions and a system
25 memory. The system memory includes code configured to track an object within a video by deforming a three-dimensional parametric model to fit the video and project the video onto the three-dimensional parametric model. The code is also configured to update a texture map corresponding to the object within the video by updating regions of the texture map that are observed from the video and render a three-dimensional video of the
30 object from any of a number of viewpoints by loosely coupling the three-dimensional parametric model and the updated texture map.

[0005] Another embodiment provides one or more computer-readable storage media including a number of instructions that, when executed by a processor, cause the processor to track an object within a video based on a three-dimensional parametric model, project

the video onto the three-dimensional parametric model, and update a texture map corresponding to the object within the video. The instructions also cause the processor to render a three-dimensional video of the object from any of a number of viewpoints by loosely coupling the three-dimensional parametric model and the updated texture map.

5 [0006] This Summary is provided to introduce a selection of concepts in a simplified form; these concepts are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

10 **BRIEF DESCRIPTION OF THE DRAWINGS**

[0007] Fig. 1 is a block diagram of a networking environment that may be used to implement a method and system for model based video projection;

[0008] Fig. 2 is a block diagram of a computing environment that may be used to implement a method and system for model based video projection;

15 [0009] Fig. 3 is a process flow diagram illustrating a model based video projection technique; and

[0010] Fig. 4 is a process flow diagram showing a method for model based video projection.

[0011] The same numbers are used throughout the disclosure and figures to reference
20 like components and features. Numbers in the 100 series refer to features originally found in Fig. 1, numbers in the 200 series refer to features originally found in Fig. 2, numbers in the 300 series refer to features originally found in Fig. 3, and so on.

DETAILED DESCRIPTION

[0012] As discussed above, a video can typically only be viewed from the viewpoint
25 from which it was captured. However, it may be desirable to view a video from a viewpoint other than the one from which it was captured. Thus, embodiments described herein set forth model based video projection techniques that allow a video or, more specifically, an object of interest in a video to be viewed from multiple different viewpoints. This may be accomplished by estimating the three-
30 dimensional structure of a remote scene and projecting a live video onto the three-dimensional structure such that the live video can be viewed from multiple viewpoints. The three-dimensional structure of the remote scene may be estimated using a parametric model.

[0013] In various embodiments, the model based video projection techniques described herein are used to view a face of a person from multiple viewpoints. According to such embodiments, the parametric model may be a generic face model. The ability to view a face from multiple viewpoints may be useful for many applications, including video conferencing applications and gaming applications, for example.

[0014] The model based video projection techniques described herein may allow for loose coupling between the three-dimensional parametric model and the video including the object of interest. In various embodiments, a complete three-dimensional video of the object of interest may be rendered even if the input video only includes partial information for the object of interest. In addition, the model based video projection techniques described herein provide for temporal consistency in geometry, as well as post-processing such as noise removal and hole filling. For example, temporal consistency in geometry may be maintained by mapping the object of interest within the video to the three-dimensional parametric model and the texture map over time. Noise removal may be accomplished by identifying the object of interest within the input video and discarding all data within the input video that does not correspond to the object of interest. Furthermore, hole filling may be accomplished by using the three-dimensional parametric model and the texture map to fill in or estimate regions of the object of interest that are not observed from the video.

[0015] As a preliminary matter, some of the figures describe concepts in the context of one or more structural components, variously referred to as functionality, modules, features, elements, etc. The various components shown in the figures can be implemented in any manner, for example, by software, hardware (e.g., discrete logic components, etc.), firmware, and so on, or any combination of these implementations. In one embodiment, the various components may reflect the use of corresponding components in an actual implementation. In other embodiments, any single component illustrated in the figures may be implemented by a number of actual components. The depiction of any two or more separate components in the figures may reflect different functions performed by a single actual component. Fig. 1, discussed below, provides details regarding one system that may be used to implement the functions shown in the figures.

[0016] Other figures describe the concepts in flowchart form. In this form, certain operations are described as constituting distinct blocks performed in a certain order. Such implementations are exemplary and non-limiting. Certain blocks described herein can be grouped together and performed in a single operation, certain blocks can be broken apart

into plural component blocks, and certain blocks can be performed in an order that differs from that which is illustrated herein, including a parallel manner of performing the blocks. The blocks shown in the flowcharts can be implemented by software, hardware, firmware, manual processing, and the like, or any combination of these implementations. As used
5 herein, hardware may include computer systems, discrete logic components, such as application specific integrated circuits (ASICs), and the like, as well as any combinations thereof.

[0017] As to terminology, the phrase “configured to” encompasses any way that any kind of functionality can be constructed to perform an identified operation. The
10 functionality can be configured to perform an operation using, for instance, software, hardware, firmware and the like, or any combinations thereof.

[0018] The term “logic” encompasses any functionality for performing a task. For instance, each operation illustrated in the flowcharts corresponds to logic for performing that operation. An operation can be performed using, for instance, software, hardware,
15 firmware, etc., or any combinations thereof.

[0019] As utilized herein, terms “component,” “system,” “client” and the like are intended to refer to a computer-related entity, either hardware, software (e.g., in execution), and/or firmware, or a combination thereof. For example, a component can be a process running on a processor, an object, an executable, a program, a function, a library,
20 a subroutine, and/or a computer or a combination of software and hardware.

[0020] By way of illustration, both an application running on a server and the server can be a component. One or more components can reside within a process and a component can be localized on one computer and/or distributed between two or more computers. The term “processor” is generally understood to refer to a hardware
25 component, such as a processing unit of a computer system.

[0021] Furthermore, the claimed subject matter may be implemented as a method, apparatus, or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof to control a computer to implement the disclosed subject matter. The term “article of manufacture”
30 as used herein is intended to encompass a computer program accessible from any tangible computer-readable storage device, or media.

[0022] Computer-readable storage media include storage devices (e.g., hard disk, floppy disk, and magnetic strips, among others), optical disks (e.g., compact disk (CD), and digital versatile disk (DVD), among others), smart cards, and flash memory devices

(e.g., card, stick, and key drive, among others). In contrast, computer-readable media (i.e., not storage media) may additionally include communication media such as transmission media for communication signals and the like.

[0023] Moreover, the word “exemplary” is used herein to mean serving as an example, instance, or illustration. Any aspect or design described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other aspects or designs.

[0024] In order to provide context for implementing various aspects of the claimed subject matter, Figs. 1-2 and the following discussion are intended to provide a brief, general description of a computing environment in which the various aspects of the subject innovation may be implemented. For example, a method and system for model based video projection can be implemented in such a computing environment. While the claimed subject matter has been described above in the general context of computer-executable instructions of a computer program that runs on a local computer or remote computer, those of skill in the art will recognize that the subject innovation also may be implemented in combination with other program modules. Generally, program modules include routines, programs, components, data structures, etc., that perform particular tasks or implement particular abstract data types.

[0025] Moreover, those of skill in the art will appreciate that the subject innovation may be practiced with other computer system configurations, including single-processor or multi-processor computer systems, minicomputers, mainframe computers, as well as personal computers, hand-held computing devices, microprocessor-based or programmable consumer electronics, and the like, each of which may operatively communicate with one or more associated devices. The illustrated aspects of the claimed subject matter may also be practiced in distributed computing environments wherein certain tasks are performed by remote processing devices that are linked through a communications network. However, some, if not all, aspects of the subject innovation may be practiced on stand-alone computers. In a distributed computing environment, program modules may be located in local or remote memory storage devices.

[0026] Fig. 1 is a block diagram of a networking environment 100 that may be used to implement a method and system for model based video projection. The networking environment 100 includes one or more client(s) 102. The client(s) 102 can be hardware and/or software (e.g., threads, processes, or computing devices). The networking environment 100 also includes one or more server(s) 104. The server(s) 104 can be hardware and/or software (e.g., threads, processes, or computing devices). The servers

104 can house threads to perform search operations by employing the subject innovation, for example.

[0027] One possible communication between a client 102 and a server 104 can be in the form of a data packet adapted to be transmitted between two or more computer
5 processes. The networking environment 100 includes a communication framework 108 that can be employed to facilitate communications between the client(s) 102 and the server(s) 104. The client(s) 102 are operably connected to one or more client data store(s) 110 that can be employed to store information local to the client(s) 102. The client data store(s) 110 may be stored in the client(s) 102, or may be located remotely, such as in a
10 cloud server. Similarly, the server(s) 104 are operably connected to one or more server data store(s) 106 that can be employed to store information local to the servers 104.

[0028] Fig. 2 is a block diagram of a computing environment that may be used to implement a method and system for model based video projection. The computing environment 200 includes a computer 202. The computer 202 includes a processing unit
15 204, a system memory 206, and a system bus 208. The system bus 208 couples system components including, but not limited to, the system memory 206 to the processing unit 204. The processing unit 204 can be any of various available processors. Dual microprocessors and other multiprocessor architectures also can be employed as the processing unit 204.

[0029] The system bus 208 can be any of several types of bus structures, including the
20 memory bus or memory controller, a peripheral bus or external bus, or a local bus using any variety of available bus architectures known to those of ordinary skill in the art. The system memory 206 is computer-readable storage media that includes volatile memory 210 and non-volatile memory 212. The basic input/output system (BIOS), containing the
25 basic routines to transfer information between elements within the computer 202, such as during start-up, is stored in non-volatile memory 212. By way of illustration, and not limitation, non-volatile memory 212 can include read-only memory (ROM), programmable ROM (PROM), electrically-programmable ROM (EPROM), electrically-erasable programmable ROM (EEPROM), or flash memory.

[0030] Volatile memory 210 includes random access memory (RAM), which acts as
30 external cache memory. By way of illustration and not limitation, RAM is available in many forms, such as static RAM (SRAM), dynamic RAM (DRAM), synchronous DRAM (SDRAM), double data rate SDRAM (DDR SDRAM), enhanced SDRAM (ESDRAM),

SynchLink™ DRAM (SLDRAM), Rambus® direct RAM (RDRAM), direct Rambus® dynamic RAM (DRDRAM), and Rambus® dynamic RAM (RDRAM).

[0031] The computer 202 also includes other computer-readable storage media, such as removable/non-removable, volatile/non-volatile computer storage media. Fig. 2 shows, for example, a disk storage 214. Disk storage 214 includes, but is not limited to, devices like a magnetic disk drive, floppy disk drive, tape drive, Jaz drive, Zip drive, LS-100 drive, flash memory card, or memory stick.

[0032] In addition, disk storage 214 can include storage media separately or in combination with other storage media including, but not limited to, an optical disk drive such as a compact disk ROM device (CD-ROM), CD recordable drive (CD-R Drive), CD rewritable drive (CD-RW Drive) or a digital versatile disk ROM drive (DVD-ROM). To facilitate connection of the disk storage 214 to the system bus 208, a removable or non-removable interface is typically used, such as interface 216.

[0033] It is to be appreciated that Fig. 2 describes software that acts as an intermediary between users and the basic computer resources described in the computing environment 200. Such software includes an operating system 218. The operating system 218, which can be stored on disk storage 214, acts to control and allocate resources of the computer 202.

[0034] System applications 220 take advantage of the management of resources by the operating system 218 through program modules 222 and program data 224 stored either in system memory 206 or on disk storage 214. It is to be appreciated that the claimed subject matter can be implemented with various operating systems or combinations of operating systems.

[0035] A user enters commands or information into the computer 202 through input devices 226. Input devices 226 include, but are not limited to, a pointing device (such as a mouse, trackball, stylus, or the like), a keyboard, a microphone, a gesture or touch input device, a voice input device, a joystick, a satellite dish, a scanner, a TV tuner card, a digital camera, a digital video camera, a web camera, or the like. The input devices 226 connect to the processing unit 204 through the system bus 208 via interface port(s) 228.

Interface port(s) 228 include, for example, a serial port, a parallel port, a game port, and a universal serial bus (USB). Output device(s) 230 may also use the same types of ports as input device(s) 226. Thus, for example, a USB port may be used to provide input to the computer 202 and to output information from the computer 202 to an output device 230.

[0036] An output adapter 232 is provided to illustrate that there are some output devices 230 like monitors, speakers, and printers, among other output devices 230, which are accessible via the output adapters 232. The output adapters 232 include, by way of illustration and not limitation, video and sound cards that provide a means of connection
5 between the output device 230 and the system bus 208. It can be noted that other devices and/or systems of devices provide both input and output capabilities, such as remote computer(s) 234.

[0037] The computer 202 can be a server hosting an event forecasting system in a networking environment, such as the networking environment 100, using logical
10 connections to one or more remote computers, such as remote computer(s) 234. The remote computer(s) 234 may be client systems configured with web browsers, PC applications, mobile phone applications, and the like. The remote computer(s) 234 can be a personal computer, a server, a router, a network PC, a workstation, a microprocessor based appliance, a mobile phone, a peer device or other common network node and the
15 like, and typically includes many or all of the elements described relative to the computer 202. For purposes of brevity, the remote computer(s) 234 is illustrated with a memory storage device 236. Remote computer(s) 234 is logically connected to the computer 202 through a network interface 238 and then physically connected via a communication connection 240.

[0038] Network interface 238 encompasses wire and/or wireless communication
20 networks such as local-area networks (LAN) and wide-area networks (WAN). LAN technologies include Fiber Distributed Data Interface (FDDI), Copper Distributed Data Interface (CDDI), Ethernet, Token Ring and the like. WAN technologies include, but are not limited to, point-to-point links, circuit switching networks like Integrated Services
25 Digital Networks (ISDN) and variations thereon, packet switching networks, and Digital Subscriber Lines (DSL).

[0039] Communication connection(s) 240 refers to the hardware/software employed to connect the network interface 238 to the system bus 208. While communication
connection 240 is shown for illustrative clarity inside computer 202, it can also be external
30 to the computer 202. The hardware/software for connection to the network interface 238 may include, for example, internal and external technologies such as mobile phone switches, modems including regular telephone grade modems, cable modems and DSL modems, ISDN adapters, and Ethernet cards.

[0040] Fig. 3 is a process flow diagram illustrating a model based video projection technique 300. In various embodiments, the model based video projection technique 300 is executed by a computing device. For example, the model based video projection technique 300 may be implemented within the networking environment 100 and/or the computing environment 200 discussed above with respect to Figs. 1 and 2, respectively. The model based video projection technique 300 may include a model tracking and fitting procedure 302, a texture map updating procedure 304, and an output video rendering procedure 306, as discussed further below.

[0041] The model tracking and fitting procedure 302 may include deforming a three-dimensional parametric model based on an input video 308 and, optionally, one or more depth maps 310 corresponding to the input video 308. Specifically, the three-dimensional parametric model may be aligned with an object of interest within the input video 308. The three-dimensional parametric model may then be used to track the object within the input video 308 by fitting the three-dimensional parametric model to the object within the input video 308 and, optionally, the one or more depth maps 310. The updated three-dimensional parametric model 312 may then be used for the output video rendering procedure 306.

[0042] According to the texture map updating procedure 304, the input video 308 and the output of the model tracking and fitting procedure 302 may be used to update a texture map corresponding to the object of interest within the input video 308. Specifically, the object of interest within the video may be mapped to the texture map, and regions of the texture map corresponding to the object that are observed from the video may be updated. In other words, if a texture region is observed in the video frame, the value of the texture region is updated. Otherwise, the value of the texture region remains unchanged.

[0043] In various embodiments, the texture map is updated over time such that every viewpoint of the object of interest that is observed from the video is reflected within the updated texture map 314. The updated texture map 314 may then be saved within the computing device, and may be used for the output video rendering procedure 306 at any point in time.

[0044] The output video rendering procedure 306 may generate an output video 316 based on the updated three-dimensional parametric model 312 and the updated texture map 314. The output video 316 may be a three-dimensional video of the object of interest within the input video 308, rendered from any desired viewpoint. For example, the output video 316 may be rendered from a viewpoint specified by a user of the computing device.

[0045] The process flow diagram of Fig. 3 is not intended to indicate that the model based video projection technique 300 is to include all of the steps shown in Fig. 3, or that all of the steps are to be executed in any particular order. Further, any number of additional steps not shown in Fig. 3 may be included within the model based video projection technique 300, depending on the details of the specific implementation.

[0046] The model based video projection technique 300 of Fig. 3 may be used for any of a variety of applications. The model based video projection technique 300 may be particularly useful for rendering a three-dimensional video of any non-rigid object for which only partial information can be obtained from an input video including the non-rigid object. For example, the model based video projection technique 300 may be used to render a three-dimensional video of a face or entire body of a person for video conferencing or gaming applications. As another example, the model based video projection technique 300 may be used to render a three-dimensional video of a particular object of interest, such as a person or animal, for surveillance or monitoring applications.

[0047] In various embodiments, a regularized maximum likelihood deformable model fitting (DMF) algorithm may be used for the model tracking and fitting procedure 302 described with respect to the model based video projection technique 300 of Fig. 3. Specifically, the regularized maximum likelihood DMF algorithm may be used in conjunction with a commodity depth camera to track an object of interest within a video and fit a model to the object of interest. For ease of discussion, the object of interest may be described herein as being a human face. However, it is to be understood that the object of interest can be any object within a video that is of interest to a user.

[0048] A linear deformable model may be used to represent the possible variations of a human face. The linear deformable model may be constructed by an artist, or may be constructed semi-automatically by a computing device. The linear deformable model may be constructed as a set of K vertices P and a set of facets F . Each vertex $\mathbf{p}_k \in P$ is a point in $\mathbb{R}^3$, and each facet $f \in F$ is a set of three or more vertices from the set P .

Within the linear deformable model, all facets have exactly three vertices. In addition, the linear deformable model is augmented with two artist-defined deformation matrices, including a static deformation matrix $\mathbf{B}$ and an action deformation matrix $\mathbf{A}$. According to weighting vectors $\mathbf{s}$ and $\mathbf{r}$, the two matrices transform the mesh linearly into a target model Q as shown below in Eq. (1).

$$\begin{bmatrix} \mathbf{q}_1 \\ \vdots \\ \mathbf{q}_K \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_K \end{bmatrix} + \mathbf{A} \begin{bmatrix} r_1 \\ \vdots \\ r_N \end{bmatrix} + \mathbf{B} \begin{bmatrix} s_1 \\ \vdots \\ s_M \end{bmatrix} \quad (1)$$

In Eq. (1), M and N are the number of deformations in $\mathbf{B}$ and $\mathbf{A}$, and $\alpha_m \leq s_m \leq \beta_m, m = 1, \dots, M$ and $\theta_n \leq r_n \leq \phi_n, n = 1, \dots, N$ are ranges specified by the artist.

- 5 The static deformations in $\mathbf{B}$ are characteristic to a particular face, such as enlarging the distance between eyes or extending the chin, for example. The action deformations include opening the mouth or raising the eyebrows, for example.

[0049] Let P represent the vertices of the model, and let G represent the three-dimensional points acquired from the depth camera. The rotation $\mathbf{R}$ and translation $\mathbf{t}$ between the model and the depth camera may be computed, as well as the deformation parameters $\mathbf{r}$ and $\mathbf{s}$. The problem may be formulated as discussed below.

[0050] It is assumed that, in a certain iteration, a set of point correspondences between the model and the depth image is available. For each correspondence $(\mathbf{p}_k, \mathbf{g}_k), \mathbf{g}_k \in G$, Eq. (2) is obtained as shown below.

15

$$\mathbf{R}(\mathbf{p}_k + \mathbf{A}_k \mathbf{r} + \mathbf{B}_k \mathbf{s}) + \mathbf{t} = \mathbf{g}_k + \mathbf{x}_k \quad (2)$$

According to Eq. (2), $\mathbf{A}_k$ and $\mathbf{B}_k$ represent the three rows of $\mathbf{A}$ and $\mathbf{B}$ that correspond to vertex k , and $\mathbf{x}_k$ is the depth sensor noise, which can be assumed to follow a zero mean

- 20 Gaussian distribution $\mathcal{N}(0, \Sigma_{\mathbf{x}_k})$. The maximum likelihood solution of the unknowns $\mathbf{R}$, $\mathbf{t}$, $\mathbf{r}$, and $\mathbf{s}$ can be derived by minimizing Eq. (3).

$$J_2(\mathbf{R}, \mathbf{t}, \mathbf{r}, \mathbf{s}) = \frac{1}{K} \sum_{k=1}^K \mathbf{x}_k^T \Sigma_{\mathbf{x}_k}^{-1} \mathbf{x}_k \quad (3)$$

- 25 In Eq. (3), $\mathbf{x}_k = \mathbf{R}(\mathbf{p}_k + \mathbf{A}_k \mathbf{r} + \mathbf{B}_k \mathbf{s}) + \mathbf{t} - \mathbf{g}_k$. Further, $\mathbf{r}$ and $\mathbf{s}$ are subject to inequality constraints, namely, $\alpha_m \leq s_m \leq \beta_m, m = 1, \dots, M$ and $\theta_n \leq r_n \leq \phi_n, n = 1, \dots, N$. In some embodiments, additional regularization terms may be added to the above optimization problem.

[0051] One possible variation is to substitute the point-to-point distance with point-to-plane distance. The point-to-plane distance allows the model to slide tangentially to the surface, which speeds up convergence and makes it less likely to get stuck in local minima. Distance to the plane can be computed using the surface normal, which can be computed from the model based on the current iteration's head pose. Let the surface normal of point $\mathbf{p}_k$ in the model coordinate be $\mathbf{n}_k$. The point-to-plane distance can be computed as shown below in Eq. (4).

$$y_k = (\mathbf{R}\mathbf{n}_k)^T \mathbf{x}_k \quad (4)$$

The maximum likelihood solution is then obtained by minimizing Eq. (5).

$$J_2(\mathbf{R}, \mathbf{t}, \mathbf{r}, \mathbf{s}) = \frac{1}{K} \sum_{k=1}^K \frac{y_k^2}{\sigma_{y_k}^2} \quad (5)$$

In Eq. (5), $\sigma_{y_k}^2 = (\mathbf{R}\mathbf{n}_k)^T \boldsymbol{\Sigma}_{\mathbf{x}_k} (\mathbf{R}\mathbf{n}_k)$, and $\alpha_m \leq s_m \leq \beta_m, m = 1, \dots, M$ and $\theta_n \leq r_n \leq \phi_n, n = 1, \dots, N$.

[0052] Given the correspondence pairs $(\mathbf{p}_k, \mathbf{g}_k)$, since both the point-to-point and the point-to-plane distances are nonlinear, a solution that solves for $\mathbf{r}, \mathbf{s}$ and $\mathbf{R}, \mathbf{t}$ in an iterative fashion may be used.

[0053] In order to generate an iterative solution for the identity noise covariance matrix, it may first be assumed that the depth sensor noise covariance matrix is a scaled identity matrix, *i.e.*, $\boldsymbol{\Sigma}_{\mathbf{x}_k} = \sigma^2 \mathbf{I}_3$, where $\mathbf{I}_3$ is a 3x3 identity matrix. Let $\tilde{\mathbf{R}} = \mathbf{R}^{-1}$, $\tilde{\mathbf{t}} = \tilde{\mathbf{R}}\mathbf{t}$. Further, let $\mathbf{y}_k$ be as shown below in Eq. (6).

$$\mathbf{y}_k = \tilde{\mathbf{R}}\mathbf{x}_k = \mathbf{p}_k + \mathbf{A}_k\mathbf{r} + \mathbf{B}_k\mathbf{s} + \tilde{\mathbf{t}} - \tilde{\mathbf{R}}\mathbf{g}_k \quad (6)$$

Since $\mathbf{x}_k^T \mathbf{x}_k = (\mathbf{R}\mathbf{y}_k)^T (\mathbf{R}\mathbf{y}_k) = \mathbf{y}_k^T \mathbf{y}_k$, the likelihood function can be written as shown below in Eq. (7).

$$J_1(\mathbf{R}, \mathbf{t}, \mathbf{r}, \mathbf{s}) = \frac{1}{K\sigma^2} \sum_{k=1}^K \mathbf{x}_k^T \mathbf{x}_k = \frac{1}{K\sigma^2} \sum_{k=1}^K \mathbf{y}_k^T \mathbf{y}_k \quad (7)$$

[0054] Similarly, for point-to-plane distance, since $y_k = (\mathbf{R}\mathbf{n}_k)^T \mathbf{x}_k = \mathbf{n}_k^T \mathbf{R}^T \mathbf{R} \mathbf{y}_k = \mathbf{n}_k^T \mathbf{y}_k$, and $\sigma_{y_k}^2 = (\mathbf{R}\mathbf{n}_k)^T \boldsymbol{\Sigma}_{\mathbf{x}_k} (\mathbf{R}\mathbf{n}_k) = \sigma^2$, Eq. (8) is obtained as shown below.

5

$$J_2(\mathbf{R}, \mathbf{t}, \mathbf{r}, \mathbf{s}) = \frac{1}{K\sigma^2} \sum_{k=1}^K \mathbf{y}_k^T \mathbf{N}_k \mathbf{y}_k \quad (8)$$

In Eq. (8), $\mathbf{N}_k = \mathbf{n}_k \mathbf{n}_k^T$.

[0055] The rotation matrix $\tilde{\mathbf{R}}$ may be decomposed into an initial rotation matrix $\tilde{\mathbf{R}}_0$ and an incremental rotation matrix $\Delta\tilde{\mathbf{R}}$, where the initial rotation matrix can be the rotation matrix of the head in the previous frame, or an estimation of $\tilde{\mathbf{R}}$ obtained in another algorithm. In other words, let $\tilde{\mathbf{R}} = \Delta\tilde{\mathbf{R}}\tilde{\mathbf{R}}_0$. Since the rotation angle of the incremental rotation matrix is small, the rotation angle may be linearized as shown below in Eq. (9).

15

$$\Delta\tilde{\mathbf{R}} \approx \begin{bmatrix} 1 & -\omega_3 & \omega_2 \\ \omega_3 & 1 & -\omega_1 \\ -\omega_2 & \omega_1 & 1 \end{bmatrix} \quad (9)$$

In Eq. (9), $\omega = [\omega_1, \omega_2, \omega_3]^T$ is the corresponding small rotation vector. Further, let

$\mathbf{q}_k = \tilde{\mathbf{R}}_0 \mathbf{g}_k = [q_{k1}, q_{k2}, q_{k3}]^T$. The variable $\mathbf{y}_k$ can be written in the form of unknowns $\mathbf{r}, \mathbf{s}, \tilde{\mathbf{t}}$, and ω as shown below in Eq. (10).

20

$$\mathbf{y}_k = \mathbf{p}_k + \mathbf{A}_k \mathbf{r} + \mathbf{B}_k \mathbf{s} + \tilde{\mathbf{t}} - \Delta\tilde{\mathbf{R}} \mathbf{q}_k \approx (\mathbf{p}_k - \mathbf{q}_k) + [\mathbf{A}_k, \mathbf{B}_k, \mathbf{I}_3, [\mathbf{q}_k]_{\times}] \begin{bmatrix} \mathbf{r} \\ \mathbf{s} \\ \tilde{\mathbf{t}} \\ \omega \end{bmatrix} \quad (10)$$

In Eq. (10), $[\mathbf{q}_k]_{\times}$ is the skew-symmetric matrix of $\mathbf{q}_k$, as shown below in Eq. (11).

25

$$[\mathbf{q}_k]_{\times} = \begin{bmatrix} 0 & -q_{k3} & q_{k2} \\ q_{k3} & 0 & -q_{k1} \\ -q_{k2} & q_{k1} & 0 \end{bmatrix} \quad (11)$$

[0056] Let $\mathbf{H}_k = [\mathbf{A}_k, \mathbf{B}_k, \mathbf{I}_3, [\mathbf{q}_k]_x]$, $\mathbf{u}_k = \mathbf{p}_k - \mathbf{q}_k$, and let $\mathbf{z} = [\mathbf{r}^T, \mathbf{s}^T, \tilde{\mathbf{t}}^T, \omega^T]^T$. Eq. (12) may then be obtained as shown below.

$$\mathbf{y}_k = \mathbf{u}_k + \mathbf{H}_k \mathbf{z} \quad (12)$$

5

Therefore, Eqs. (13) and (14) can be obtained as shown below.

$$J_1 = \frac{1}{K\sigma^2} \sum_{k=1}^K \mathbf{y}_k^T \mathbf{y}_k = \frac{1}{K\sigma^2} \sum_{k=1}^K (\mathbf{u}_k + \mathbf{H}_k \mathbf{z})^T (\mathbf{u}_k + \mathbf{H}_k \mathbf{z}) \quad (13)$$

$$J_2 = \frac{1}{K\sigma^2} \sum_{k=1}^K \mathbf{y}_k^T \mathbf{N}_k \mathbf{y}_k = \frac{1}{K\sigma^2} \sum_{k=1}^K (\mathbf{u}_k + \mathbf{H}_k \mathbf{z})^T \mathbf{N}_k (\mathbf{u}_k + \mathbf{H}_k \mathbf{z}) \quad (14)$$

10

Both likelihood functions are quadratic with respect to $\mathbf{z}$. Since there are linear constraints on the range of values for $\mathbf{r}$ and $\mathbf{s}$, the minimization problem can be solved with quadratic programming.

15 [0057] The rotation vector ω is an approximation of the actual incremental rotation matrix. One can simply insert $\Delta \tilde{\mathbf{R}} \tilde{\mathbf{R}}_0$ to the position of $\tilde{\mathbf{R}}_0$ and repeat the above optimization process until it converges.

[0058] A solution for the arbitrary noise covariance matrix may also be generated. When the sensor noise covariance matrix is arbitrary, an iterative solution may be
20 obtained. Since $\mathbf{y}_k = \tilde{\mathbf{R}} \mathbf{x}_k$, $\Sigma_{\mathbf{y}_k} = \tilde{\mathbf{R}} \Sigma_{\mathbf{x}_k} \tilde{\mathbf{R}}^T$. A feasible solution can be obtained if $\tilde{\mathbf{R}}$ is replaced with its estimation $\tilde{\mathbf{R}}_0$ as shown below in Eq. (15).

$$\Sigma_{\mathbf{y}_k} \approx \tilde{\mathbf{R}}_0 \Sigma_{\mathbf{x}_k} \tilde{\mathbf{R}}_0^T \quad (15)$$

25 The solution to Eq. (16) is known for the current iteration. Subsequently, Eqs. (16) and (17) may be obtained.

$$J_1 = \frac{1}{K} \sum_{k=1}^K \mathbf{y}_k^T \Sigma_{\mathbf{y}_k}^{-1} \mathbf{y}_k = \frac{1}{K} \sum_{k=1}^K (\mathbf{u}_k + \mathbf{H}_k \mathbf{z})^T \Sigma_{\mathbf{y}_k}^{-1} (\mathbf{u}_k + \mathbf{H}_k \mathbf{z}) \quad (16)$$

$$J_2 = \frac{1}{K} \sum_{k=1}^K \frac{\mathbf{y}_k^T \mathbf{N}_k \mathbf{y}_k}{\mathbf{n}_k^T \boldsymbol{\Sigma}_{\mathbf{y}_k} \mathbf{n}_k} = \frac{1}{K} \sum_{k=1}^K \frac{(\mathbf{u}_k + \mathbf{H}_k \mathbf{z})^T \mathbf{N}_k (\mathbf{u}_k + \mathbf{H}_k \mathbf{z})}{\mathbf{n}_k^T \boldsymbol{\Sigma}_{\mathbf{y}_k} \mathbf{n}_k} \quad (17)$$

[0059] The quadratic functions with respect to $\mathbf{z}$ can be solved via quadratic programming. Again, the minimization may be repeated until convergence by inserting $\Delta \tilde{\mathbf{R}} \tilde{\mathbf{R}}_0$ to the position of $\tilde{\mathbf{R}}_0$ in each iteration.

[0060] For the model tracking and fitting procedure 302 described herein, the above maximum likelihood DMF framework is applied differently in two stages. During the initialization stage, the goal is to fit the generic deformable model to an arbitrary person. It may be assumed that a set of L ($L \leq 10$ in the current implementation) neutral face

frames is available. The action deformation vector $\mathbf{r}$ is assumed to be zero. The static deformation vector $\mathbf{s}$ and the face rotations and translations are jointly solved as follows.

[0061] The correspondences are denoted as $(\mathbf{p}_{lk}, \mathbf{g}_{lk})$, where $l = 1, \dots, L$ represents the frame index. Assume in the previous iteration that $\tilde{\mathbf{R}}_{l0}$ is the rotation matrix for frame l . Let $\mathbf{q}_{lk} = \tilde{\mathbf{R}}_{l0} \mathbf{g}_{lk}$ and $\mathbf{H}_{lk} = [\mathbf{B}_k, \mathbf{0}, \mathbf{0}, \dots, \mathbf{I}_3, [\mathbf{q}_{lk}]_x, \dots, \mathbf{0}, \mathbf{0}]$, where $\mathbf{0}$ represents a 3x3 zero matrix. Let $\mathbf{u}_{lk} = \mathbf{p}_{lk} - \mathbf{q}_{lk}$, and the unknown vector $\mathbf{z} = [\mathbf{s}^T, \tilde{\mathbf{t}}_1^T, \omega_1^T, \dots, \tilde{\mathbf{t}}_L^T, \omega_L^T]^T$. Following Eqs. (16) and (17), the overall likelihood function may be rewritten as shown below in Eqs. (18) and (19).

$$J_{\text{init1}} = \frac{1}{KL} \sum_{l=1}^L \sum_{k=1}^K (\mathbf{u}_{lk} + \mathbf{H}_{lk} \mathbf{z})^T \boldsymbol{\Sigma}_{\mathbf{y}_{lk}}^{-1} (\mathbf{u}_{lk} + \mathbf{H}_{lk} \mathbf{z}) \quad (18)$$

$$J_{\text{init2}} = \frac{1}{KL} \sum_{l=1}^L \sum_{k=1}^K \frac{(\mathbf{u}_{lk} + \mathbf{H}_{lk} \mathbf{z})^T \mathbf{N}_{lk} (\mathbf{u}_{lk} + \mathbf{H}_{lk} \mathbf{z})}{\mathbf{n}_{lk}^T \boldsymbol{\Sigma}_{\mathbf{y}_{lk}} \mathbf{n}_{lk}} \quad (19)$$

According to Eqs. (18) and (19), $\mathbf{n}_{lk}$ is the surface normal vector for point $\mathbf{p}_{lk}$, $\mathbf{N}_{lk} = \mathbf{n}_{lk} \mathbf{n}_{lk}^T$, and $\boldsymbol{\Sigma}_{\mathbf{y}_{lk}} \approx \tilde{\mathbf{R}}_{l0} \boldsymbol{\Sigma}_{\mathbf{x}_{lk}} \tilde{\mathbf{R}}_{l0}^T$. In addition, $\mathbf{x}_{lk}$ is the sensor noise for depth input $\mathbf{g}_{lk}$.

[0062] The point-to-point and point-to-plane likelihood functions are used jointly in the current implementation. A selected set of point correspondences is used for J_{init1} , and

another selected set of point correspondences is used for $J_{\text{init}2}$. The overall target function is the linear combination shown below in Eq. (20).

$$J_{\text{init}} = \lambda_1 J_{\text{init}1} + \lambda_2 J_{\text{init}2} \quad (20)$$

5

In Eq. (20), λ_1 and λ_2 are the weights between the two functions. The optimization is conducted through quadratic programming.

[0063] After the static deformation vector $\mathbf{S}$ has been initialized, the face is tracked frame by frame. The action deformation vector $\mathbf{r}$, face rotation $\mathbf{R}$, and translation $\mathbf{t}$ may be estimated, while keeping $\mathbf{s}$ fixed. In some embodiments, additional regularization terms may also be added in the target function to further improve the results.

10

[0064] A natural assumption is that the expression change between the current frame and the previous frame is small. According to embodiments described herein, if the previous frame's face action vector is $\mathbf{r}^{t-1}$, the ℓ_2 regularization term may be added according to Eq. (21).

15

$$J_{\text{track}} = \lambda_1 J_1 + \lambda_2 J_2 + \lambda_3 \|\mathbf{r} - \mathbf{r}^{t-1}\|_2^2 \quad (21)$$

In Eq. (21), J_1 and J_2 follow Eqs. (16) and (17). Similar to the initialization process, J_1 and J_2 use different sets of feature points. The term $\|\mathbf{r} - \mathbf{r}^{t-1}\|_2^2 = (\mathbf{r} - \mathbf{r}^{t-1})^T (\mathbf{r} - \mathbf{r}^{t-1})$ is the squared ℓ_2 norm of the difference between the two vectors.

20

[0065] The $\mathbf{r}$ vector represents a particular action a face can perform. Since it is difficult for a face to perform all actions simultaneously, the $\mathbf{r}$ vector may be sparse in general. Thus, an additional ℓ_1 regularization term may be imposed, as shown below in Eq. (22).

25

$$J_{\text{track}} = \lambda_1 J_1 + \lambda_2 J_2 + \lambda_3 \|\mathbf{r} - \mathbf{r}^{t-1}\|_2^2 + \lambda_4 \|\mathbf{r}\|_1 \quad (22)$$

In Eq. (22), $\|\mathbf{r}\|_1 = \sum_{n=1}^N |r_n|$ is the ℓ_1 norm. This regularized target function is now in the form of an ℓ_1 -regularized least squares problem, which can be reformulated as a

30

convex quadratic program with linear inequality constraints. This can be solved with quadratic programming methods.

[0066] Multiple neutral face frames may be used for model initialization. The likelihood function J_{init} contains both point-to-point and point-to-plane terms, as shown in Eq. (20). For the point-to-plane term $J_{\text{init}2}$, the corresponding point pairs are derived by the standard procedure of finding the closest point on the depth map from the vertices on the deformable model. However, the point-to-plane term alone may not be sufficient, since the depth maps may be noisy and the vertices of the deformable model can drift tangentially, leading to unnatural faces.

[0067] For each initialization frame, face detection and alignment may first be performed on the texture image. The alignment algorithm may provide a number of landmark points of the face, which are assumed to be consistent across all the frames. These landmark points are separated into four categories. The first category includes landmark points representing eye corners, mouth corners, and the like. Such landmark points have clear correspondences $\mathbf{p}_{lk}$ in the linear deformable face model. Given the calibration information between the depth camera and the texture camera, the landmark points can simply be projected to the depth image to find the corresponding three-dimensional world coordinate $\mathbf{g}_{lk}$.

[0068] The second category includes landmark points on the eyebrows and upper and lower lips. The deformable face model has a few vertices that define eyebrows and lips, but the vertices do not all correspond to the two-dimensional feature points provided by the alignment algorithm. In order to define correspondences, the following procedure may be performed. First, the previous iteration's head rotation $\mathbf{R}_0$ and translation $\mathbf{t}_0$ may be used to project the face model vertices $\mathbf{p}_{lk}$ of the eyebrows and upper and lower lips to the texture image $\mathbf{v}_{lk}$. Second, the closest point on the curve defined by the alignment results to $\mathbf{v}_{lk}$ may be found and may be defined as $\mathbf{v}'_{lk}$. Third, $\mathbf{v}'_{lk}$ may be back projected to the depth image to find its three-dimensional world coordinate $\mathbf{g}_{lk}$.

[0069] The third category includes landmark points surrounding the face, which may be referred to as silhouette points. The deformable model also has vertices that define these boundary points, but there is no correspondence between them and the alignment results. Moreover, when back projecting the silhouette points to the three-dimensional world coordinate, the silhouette points may easily hit a background pixel in the depth

image. For these points, a procedure that is similar to the procedure that is performed for the second category of landmark points may be performed. However, the depth axis may be ignored when computing the distance between $\mathbf{p}_{lk}$ and $\mathbf{g}_{lk}$. Furthermore, the fourth category of landmark points includes all of the white points, which are not used in the current implementation.

[0070] During tracking, both the point-to-point and point-to-plane likelihood terms may be used, with additional regularization as shown in Eq. (22). The point-to-plane term is computed similarly as that during model initialization. Feature points detected and tracked from the texture images may be relied on to define the point correspondences.

[0071] The feature points are detected in the texture image of the previous frame using the Harris corner detector. The feature points are then tracked to the current frame by matching patches surrounding the points using cross correlation. In some cases, however, the feature points may not correspond to any vertices in the deformable face model. Given the previous frame's tracking results, the feature points are first represented with their barycentric coordinates. Specifically, for two-dimensional feature point pair $\mathbf{v}_k^{t-1}$ and $\mathbf{v}_k^t$, the parameters n_1 , n_2 , and n_3 are obtained such that Eq. (23) holds.

$$\mathbf{v}_k^{t-1} = n_1 \hat{\mathbf{p}}_{k_1}^{t-1} + n_2 \hat{\mathbf{p}}_{k_2}^{t-1} + n_3 \hat{\mathbf{p}}_{k_3}^{t-1} \quad (23)$$

In Eq. (23), $n_1 + n_2 + n_3 = 1$, and $\hat{\mathbf{p}}_{k_1}^{t-1}$, $\hat{\mathbf{p}}_{k_2}^{t-1}$, and $\hat{\mathbf{p}}_{k_3}^{t-1}$ are the two-dimensional projections of the deformable model vertices $\mathbf{p}_{k_1}$, $\mathbf{p}_{k_2}$, and $\mathbf{p}_{k_3}$ onto the previous frame. Similarly the Eq. (2), Eq. (24) may be obtained as shown below.

$$\mathbf{R} \sum_{i=1}^3 n_i (\mathbf{p}_k + \mathbf{A}_k \mathbf{r} + \mathbf{B}_k \mathbf{s}) + \mathbf{t} = \mathbf{g}_k + \mathbf{x}_k \quad (24)$$

In Eq. (24), $\mathbf{g}_k$ is the back projected three-dimensional world coordinate of the two-dimensional feature point $\mathbf{v}_k^t$. Let $\bar{\mathbf{p}}_k = \sum_{i=1}^3 n_i \mathbf{p}_{k_i}$, $\bar{\mathbf{A}}_k = \sum_{i=1}^3 n_i \mathbf{A}_{k_i}$, and $\bar{\mathbf{B}}_k = \sum_{i=1}^3 n_i \mathbf{B}_{k_i}$. Eq. (24) will be identical form as Eq. (2). Therefore, tracking is still solved using Eq. (22).

[0072] Due to the potential of strong noise in the depth sensor, it may be desirable to model the actual sensor noise with the correct $\Sigma_{\mathbf{x}_k}$ instead of using an identity matrix for approximation. The uncertainty of the three-dimensional point $\mathbf{g}_k$ has at least two sources, including the uncertainty in the depth image intensity, which translates to uncertainty along the depth axis, and the uncertainty in feature point detection and matching in the texture image, which translates to uncertainty along the imaging plane.

[0073] Assuming a pinhole, no-skew projection model for the depth camera, Eq. (25) may be obtained.

$$z_k \begin{bmatrix} u_k \\ v_k \\ 1 \end{bmatrix} = \mathbf{K} \mathbf{g}_k = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} \quad (25)$$

According to eq. (25), $\mathbf{v}_k = [u_k, v_k]^T$ is the two-dimensional image coordinate of the feature point k in the depth image, and $\mathbf{g}_k = [x_k, y_k, z_k]^T$ is the three-dimensional world coordinate of the feature point. In addition, $\mathbf{K}$ is the intrinsic matrix, where f_x and f_y are the focal lengths, and u_0 and v_0 are the center biases.

[0074] For the depth camera, the uncertainty of u_k and v_k is generally caused by feature point uncertainties in the texture image, and the uncertainty in z_k is due to the depth derivation scheme. These two uncertainties can be considered as independent of each other. Let $\mathbf{c}_k = [u_k, v_k, z_k]^T$. Eq. (26) may then be obtained as shown below.

$$\Sigma_{\mathbf{c}_k} = \begin{bmatrix} \Sigma_{\mathbf{v}_k} & 0 \\ 0^T & \sigma_{z_k}^2 \end{bmatrix} \quad (26)$$

From Eq. (26), Eq. (27) may be obtained.

$$\mathbf{G}_k \triangleq \frac{\partial \mathbf{g}_k}{\partial \mathbf{c}_k} = \begin{bmatrix} \frac{z_k}{f_x} & 0 & \frac{u_k - u_0}{f_x} \\ 0 & \frac{z_k}{f_y} & \frac{v_k - v_0}{f_y} \\ 0 & 0 & 1 \end{bmatrix} \quad (27)$$

Hence, as an approximation, the sensor's noise covariance matrix may be defined according to Eq. (28).

$$\mathbf{\Sigma}_{\mathbf{x}_k} \approx \mathbf{G}_k \mathbf{\Sigma}_{\mathbf{c}_k} \mathbf{G}_k^T \quad (28)$$

5

[0075] In the current implementation, to compute $\mathbf{\Sigma}_{\mathbf{c}_k}$ from Eq. (26), it may be assumed that $\mathbf{\Sigma}_{\mathbf{v}_k}$ is diagonal, *i.e.*, $\mathbf{\Sigma}_{\mathbf{v}_k} = \sigma^2 \mathbf{I}_2$, where $\mathbf{I}_2$ is the 2x2 identity matrix, and $\sigma = 1.0$ pixels. Knowing that the depth sensor derives depth based on triangulation, following Eq. (24), the depth image noise covariance $\sigma_{z_k}^2$ may be modeled as shown below in Eq. (29).

10

$$\sigma_{z_k}^2 = \frac{\sigma_0^2 z_k^4}{f_d^2 B^2} \quad (29)$$

In Eq. (29), $f_d = \frac{f_x + f_y}{2}$ is the depth camera's average focal length; $\sigma_0 = 0.059$ pixels;

15

and $B = 52.3875$ millimeters based on calibration. Since σ_{z_k} depends on z_k , its value depends on each pixel's depth value and cannot be predetermined.

[0076] It is to be understood that the model tracking and fitting procedure 302 of Fig. 3 may be performed using any variation of the techniques described above. For example, the conditions and equations described above with respect to the model tracking and fitting procedure 302 may be modified based on the details of the specific implementation of the model based video projection technique 300.

20

[0077] Fig. 4 is a process flow diagram showing a method 400 for model based video projection. In various embodiments, the method 400 is executed by a computing device. For example, the method 400 may be implemented within the networking environment 100 and/or the computing environment 200 discussed above with respect to Figs. 1 and 2, respectively.

25

[0078] The method begins at block 402, at which an object within a video is tracked based on a three-dimensional parametric model. The video may be obtained from a physical camera. For example, the video may be obtained from a camera that is coupled to the computing device that is executing the method 400, or may be obtained from a

30

remote camera via a network. The three-dimensional parametric model may be generated based on data relating to various objects of interest. For example, the parametric model may be a generic face model that is generated based on data relating to a human face.

[0079] The object may be any object within the video that has been designated as

5 being of interest to a user of the computing device, for example. In various embodiments, the user may specify the type of object that is to be tracked, and an appropriate three-dimensional parametric model may be selected accordingly. In other embodiments, the three-dimensional parametric model automatically determines and adapts to the object within the video.

10 [0080] In various embodiments, the object within the video is tracked by aligning the three-dimensional parametric model with the object within the video. The three-dimensional parametric model may then be deformed to fit the video. In some

embodiments, if one or more depth maps (or three-dimensional points clouds) corresponding to the video are available, the three-dimensional parametric model is

15 deformed to fit the video and the one or more depth maps. The one or more depth maps may include images that contain information relating to the distance from the viewpoint of the camera that captured the video to the surface of the object within the scene. In addition, tracking the object within the video may include determining parameters for the three-dimensional parametric model based on data corresponding to the object within the
20 video.

[0081] At block 404, the video is projected onto the three-dimensional parametric model. At block 406, a texture map corresponding to the object within the video is updated. The texture map may be updated by mapping the object within the video to the texture map. This may be accomplished by updating regions of the texture map

25 corresponding to the object that are observed from the video. Thus, the texture map may be updated such that the object within the video is closely represented by the texture map.

[0082] At block 408, a three-dimensional video of the object is rendered from any of a number of viewpoints by loosely coupling the three-dimensional parametric model and the updated texture map. For example, the three-dimensional video may be rendered from a
30 viewpoint specified by the user of the computing device. The three-dimensional video may then be used for any of a variety of applications, such as video conferencing applications or gaming applications.

[0083] In various embodiments, loosely coupling the three-dimensional parametric model and the updated texture map includes allowing the three-dimensional parametric

model to not fully conform to the texture of the texture map. For example, if the object is a human face, the mouth region may be flat and not follow the texture of the lips and teeth within the texture map very closely. This may result in a higher quality visual representation of the object than is achieved when a more complex model is inferred from the video. Moreover, the three-dimensional parametric model may be simple, e.g., may not include very many parameters. Thus, strict coupling between the three-dimensional parametric model and the texture map may not be achievable. The degree of coupling that is achieved between the three-dimensional parametric model and the texture map may vary depending on the details of the specific implementation. For example, the degree of coupling may vary based on the complexity of the three-dimensional parametric model and the complexity of the object being tracked.

[0084] The process flow diagram of Fig. 4 is not intended to indicate that the method 400 is to include all of the steps shown in Fig. 3, or that all of the steps are to be executed in any particular order. Further, any number of additional steps not shown in Fig. 4 may be included within the method 400, depending on the details of the specific implementation. For example, the texture map may be updated based on the object within the video over a specified period of time, and the updated texture map may be used to render the three-dimensional video of the object from any specified viewpoint at any point in time. In addition, texture information relating to the updated texture map may be stored as historical texture information and used to render the object or a related object at some later point in time.

[0085] Further, if the tracked object is an individual's face, blending between the three-dimensional parametric model corresponding to the object and the remaining real-time captured video information corresponding to the rest of the body may be performed. In various embodiments, blending between the three-dimensional parametric model corresponding to the object and the video information corresponding to the rest of the body may allow for rendering of the entire body of the individual, with an emphasis on the face of the individual. In this manner, the individual's face may be viewed in context within the three-dimensional video, rather than as a disconnected object.

[0086] Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.

CLAIMS

1. A method for model based video projection, comprising:
tracking an object within a video based on a three-dimensional parametric model
via a computing device;
projecting the video onto the three-dimensional parametric model;
updating a texture map corresponding to the object within the video; and
rendering a three-dimensional video of the object from any of a plurality of
viewpoints by loosely coupling the three-dimensional parametric model
and the updated texture map.
2. The method of claim 1, wherein tracking the object within the video
comprises aligning the three-dimensional parametric model with the object within the
video.
3. The method of any combination of claims 1-2, wherein tracking the object
within the video comprises deforming the three-dimensional parametric model to fit the
video.
4. The method of any combination of claims 1-3, wherein tracking the object
within the video comprises:
deforming the three-dimensional parametric model to fit the video and a depth map
corresponding to the video;
determining parameters for the three-dimensional parametric model based on data
corresponding to the object within the video; or
any combination thereof.
5. The method of any combination of claims 1-4, wherein updating the texture
map comprises mapping the object within the video to the texture map by updating regions
of the texture map corresponding to the object that are observed from the video.
6. The method of any combination of claims 1-5, wherein rendering the three-
dimensional video from any of the plurality of viewpoints comprises rendering the three-
dimensional video from a viewpoint specified by a user of the computing device.

7. The method of any combination of claims 1-6, comprising:
updating the texture map based on the object within the video over a specified period of time; and
using the updated texture map to render the three-dimensional video of the object from any specified viewpoint at any point in time.
8. The method of any combination of claims 1-7, comprising using the three-dimensional video for a video conferencing application.
9. The method of any combination of claims 1-8, comprising using the three-dimensional video for a gaming application.
10. A system for model based video projection, comprising:
a processor that is configured to execute stored instructions; and
a system memory, wherein the system memory comprises code configured to:
track an object within a video by deforming a three-dimensional parametric model to fit the video;
project the video onto the three-dimensional parametric model;
update a texture map corresponding to the object within the video by updating regions of the texture map that are observed from the video; and
render a three-dimensional video of the object from any of a plurality of viewpoints by loosely coupling the three-dimensional parametric model and the updated texture map.
11. The system of claim 10, wherein the system memory comprises code configured to track the object within the video by deforming the three-dimensional parametric model to fit the video and a depth map corresponding to the video.
12. The system of any combination of claims 10-11, wherein the system memory comprises code configured to render the three-dimensional video from a viewpoint specified by a user of the system.

13. The system of any combination of claims 10-12, wherein the processor is configured to send the three-dimensional video of the object to:

a video conferencing application.

a gaming application;

or any combination thereof.

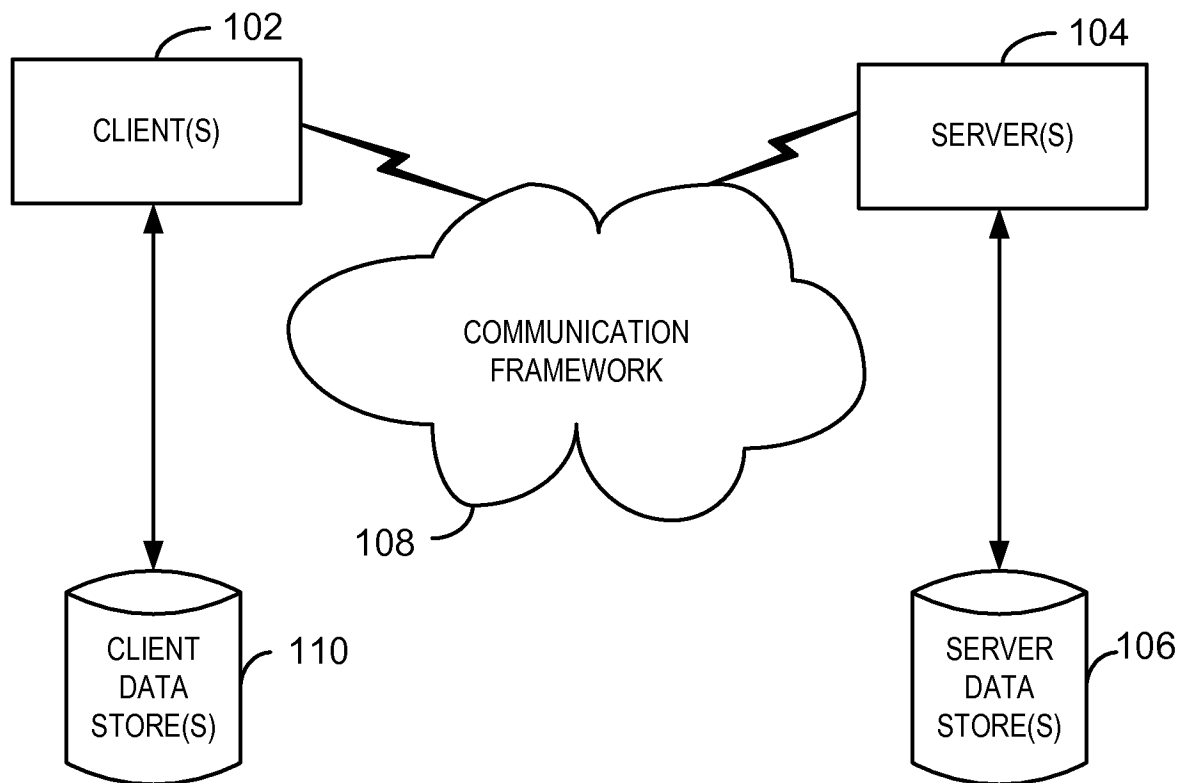
14. The system of any combination of claims 10-13, wherein the processor is configured to:

store the updated texture map within the system memory; and

use the updated texture map to render the three-dimensional video of the object from any specified viewpoint at any point in time.

15. One or more computer-readable storage media comprising a plurality of instructions that, when executed by a processor, cause the processor to carry out a method or realize a system of any one of the preceding claims.

1/4

100**FIG. 1**

2/4

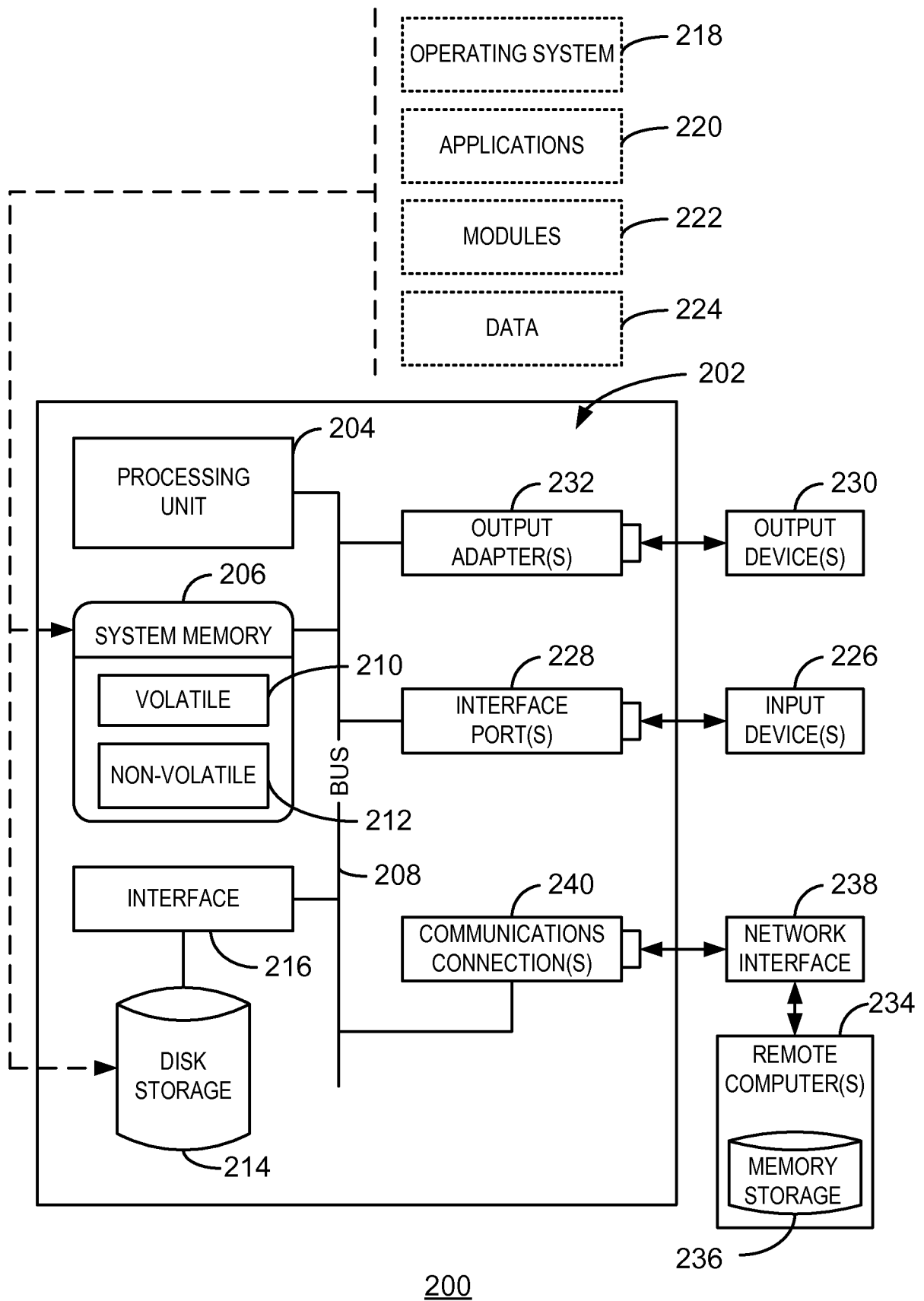
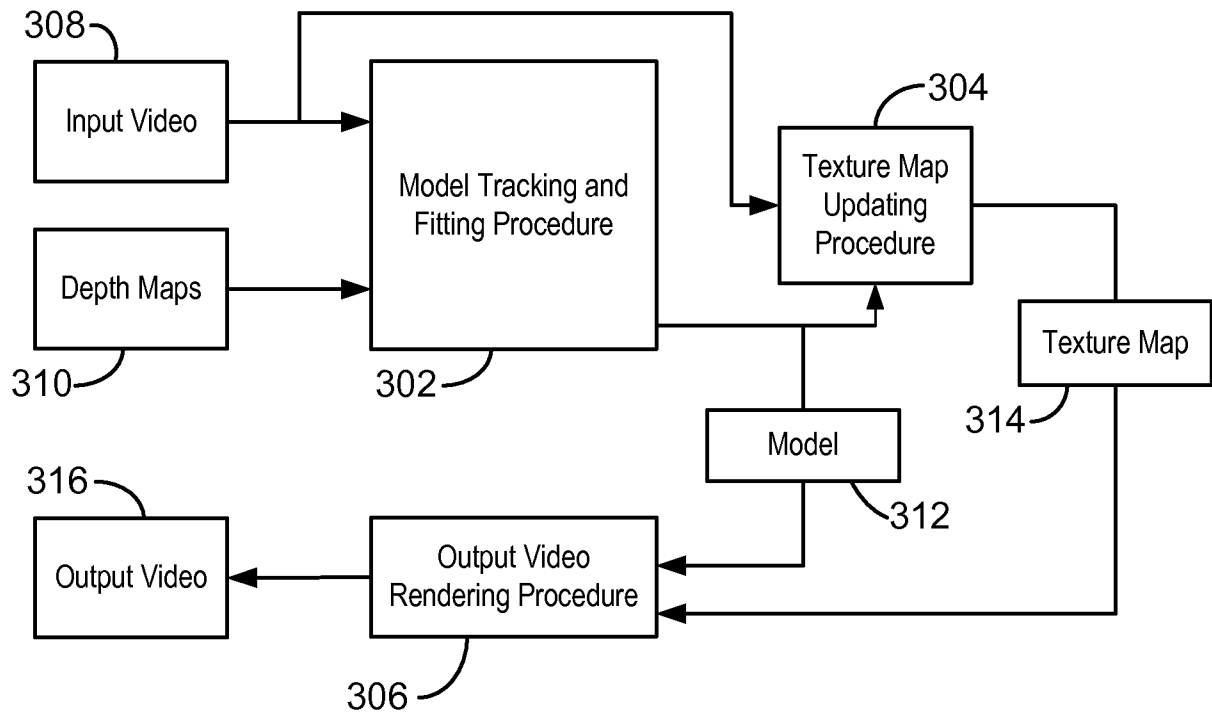
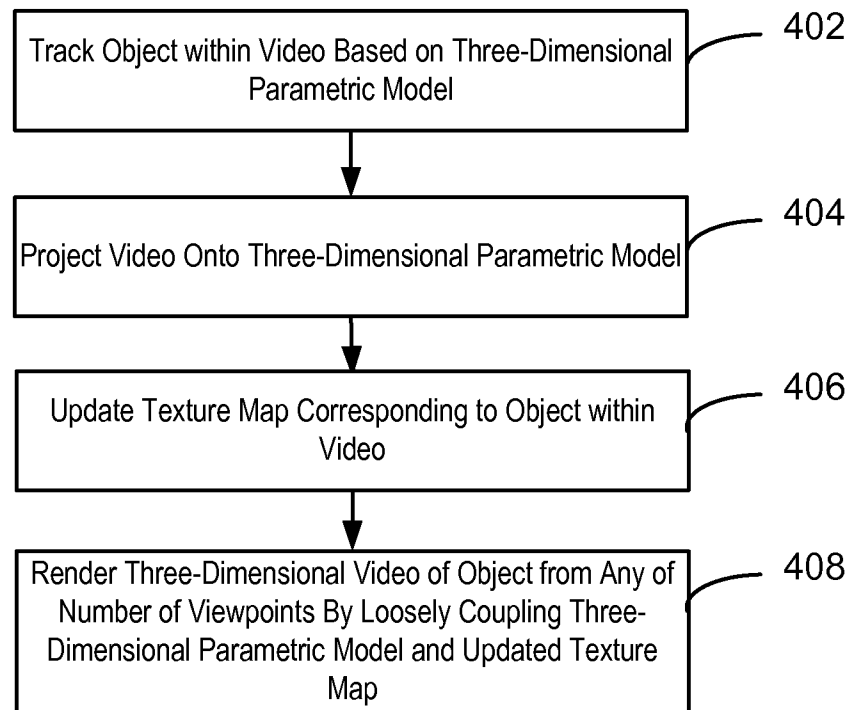


FIG. 2

3/4

300**FIG. 3**

4/4

400**FIG. 4**

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2013/075152

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06T17/00
ADD.

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)
G06T

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)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ZOLLHÖFER, M., MARTINEK, M., GREINER G., STAMMINGER, M. AND SÜSSMUTH, J.: "Automatic reconstruction of personalized avatars from 3D face scans", COMPUTER ANIMATION AND VIRTUAL WORLDS, [Online] vol. 22, 12 April 2011 (2011-04-12), pages 195-202, XP002720391, DOI: 10.1002/cav.405 Retrieved from the Internet: URL: http://onlinelibrary.wiley.com/doi/10.1002/cav.405/pdf [retrieved on 2014-02-14] abstract figures 4,5,6 sections 2.4, 3, 3.4 and 4 ----- -/-	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

17 February 2014

Date of mailing of the international search report

24/03/2014

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

dos Santos, Luís

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2013/075152

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	SATO, H., TERASHIMA, N. AND TOMINAGA, H.: "Human Face Image Creation for Virtual Space Teleconferencing using Camera Input Images", IAPR WORKSHOP ON MACHINE VISION APPLICATIONS, [Online] 28 November 2000 (2000-11-28), - 30 November 2000 (2000-11-30), pages 375-378, XP002720392, Retrieved from the Internet: URL:http://b2.cvl.iis.u-tokyo.ac.jp/mva/pr oceedings/CommemorativeDVD/2000/papers/200 0375.pdf> [retrieved on 2014-02-17] abstract figure 1	1-15
X	----- GARCIA, E., DUGELAY, J.-L.: "Low cost 3d face acquisition and modeling", INFORMATION TECHNOLOGY: CODING AND COMPUTING, 2001. PROCEEDINGS. INTERNATIONAL CONFERENCE ON, [Online] April 2001 (2001-04), pages 657-661, XP002720393, DOI: 10.1109/ITCC.2001.918872 Retrieved from the Internet: URL:http://ieeexplore.ieee.org/xpls/abs_a1 1.jsp?arnumber=918872&tag=1> [retrieved on 2014-02-17] abstract sections 7, 8; figure 1	1-15
X	----- WO 2004/042662 A1 (UNIV SOUTHERN CALIFORNIA [US]; YOU SUYA [US]; NEUMANN ULRICH [US]) 21 May 2004 (2004-05-21) paragraphs [0022] - [0026], [0029], [0031]	1-15
A	----- WASCHBUSCH ET AL: "Point-sampled 3D video of real-world scenes", SIGNAL PROCESSING. IMAGE COMMUNICATION, ELSEVIER SCIENCE PUBLISHERS, AMSTERDAM, NL, vol. 22, no. 2, 16 March 2007 (2007-03-16) , pages 203-216, XP005938669, ISSN: 0923-5965, DOI: 10.1016/J.IMAGE.2006.11.009 abstract figure 10 -----	1-15

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2013/075152

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2004042662 A1	21-05-2004	AU 2003277240 A1	07-06-2004
		EP 1567988 A1	31-08-2005
		JP 4185052 B2	19-11-2008
		JP 2006503379 A	26-01-2006
		US 2004105573 A1	03-06-2004
		WO 2004042662 A1	21-05-2004
