



(43) International Publication Date
3 August 2017 (03.08.2017)

- (51) **International Patent Classification:**
G06T 7/00 (2006.01) **G06T 17/00** (2006.01)
- (21) **International Application Number:**
PCT/US2016/015181
- (22) **International Filing Date:**
27 January 2016 (27.01.2016)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP [US/US]; 11445 Compaq Center Drive West, Houston, Texas 77070 (US).
- (72) **Inventors:** RAUDIES, Florian; 1501 Page Mill Road, Palo Alto, California 94304-1100 (US). SATKHOZHINA, Aziza; 1501 Page Mill Road, Palo Alto, California 94304-1100 (US).
- (74) **Agents:** PAGAR, Preetam B. et al.; Hewlett Packard Enterprise, Mail Stop 79, 3404 E. Harmony Road, Fort Collins, Colorado 80528 (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, 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, ST, 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).

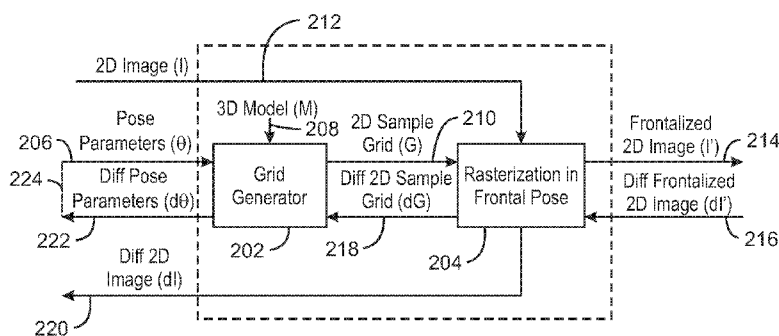
Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

Published:

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

(54) Title: GENERATING POSE FRONTALIZED IMAGES OF OBJECTS

200
FIG. 2

(S7) Abstract: An example embodiment of the present techniques receives an image of an object and a three-dimensional (3D) model of the object. A pose of the object in the image can be estimated based on estimation of a plurality of parameters. The plurality of parameters may describe a 3D rotation and 3D translation and can be learned via error minimization over a plurality of training samples. A frontalized image of the object can be generated based on the estimated pose and the received 3D model of the object.

GENERATING POSE FRONTALIZED IMAGES OF OBJECTS

BACKGROUND

[0001] Many situations exist in which object depictions in images are detected, identified, or verified. For example, faces of people in images can be identified or verified using various techniques.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Certain example embodiments are described in the following detailed description and in reference to the drawings, in which:

[0003] Fig. 1 is a block diagram of an example deep convolutional network that includes a frontalization layer;

[0004] Fig. 2 is a block diagram showing an example frontalization layer;

[0005] Fig. 3 is a block diagram showing an example system that can frontalize objects detected in an image;

[0006] Fig. 4 is a process flow diagram showing an example method of generating pose frontalized images of objects; and

[0007] Fig. 5 is a block diagram showing an example non-transitory, tangible computer-readable medium that stores code for frontalization of faces in images.

DETAILED DESCRIPTION

[0008] As described above, many situations exist in which features of objects in images can be detected, identified, or verified. For example, faces are objects of special importance and face frontalization is an important step for the automated detection, identification, and verification of faces. Detection, as used herein, refers to detecting and locating a face in an image. Identification, as used herein, refers to identifying a person in an image. Verification, as used herein, refers to deciding whether two images depict the same person or not. For any of the three tasks, faces may appear in different poses in an image. For example, a face may point to the lower-right in one image and to the far-left in

another image. Frontalization can be used to reconstruct the frontal view of such a face.

[0009] Accordingly, some examples described herein provide a method for the frontalization of faces through 3D rotations and 3D translations that are learned by a deep convolutional network. Such a frontalization can be used to simplify the tasks of face detection, face identification, and face verification. Moreover, face detection, identification, and verification is used in a variety of security tasks. Thus, the techniques herein can be used for user authentication as it may be less likely that such a system would be fooled by holding a picture of the same person in front of a camera. Conversely, using the present techniques, individuals cannot hide from security cameras by presenting only a side view of their face because the system can transform the side views and generate a frontal pose.

[0010] In addition, the present techniques for face frontalization can be used in a host of security tasks including: border control; a national identity system; access control to devices, homes, cars, secured areas; or face-based analysis of age, sex, ethnicity, or facial expression. Overall, such systems can be used to control access to devices on a small scale as well as access to buildings or countries on a larger scale.

[0011] Further, in some implementations, the present techniques include the use of a separate frontalization layer for performing frontalization tasks in a deep convolutional network. Thus, frontalization and identity tasks are learned through representations in separate layers of the deep convolutional network, rather than being learned by a representation spanning several layers. The break-down into two subsequent tasks reduces the number of samples required for training because frontalization is largely independent of the individual face in the image, and improves the accuracy for the identity task because it can leverage frontal poses, which is an easier task than recognizing identity in non-frontal poses. In this case, working with fewer training samples also reduces the training time for deep convolutional networks. Moreover, the present techniques do not use landmarks for purposes of detection. Thus, the techniques described herein may use fewer training samples resulting in a

reduced training time. Moreover, the techniques enable identification with higher accuracy because identification is performed using frontal poses.

[0012] Fig. 1 is a block diagram of an example deep convolutional network that includes a frontalization layer. The example deep convolutional network is generally referred to by the reference number 100 and can be implemented using the example computing device 302 of Fig. 3 below.

[0013] The example deep convolutional network 100 includes a plurality of functional layers. For example, the layers may include an image detection layer 102, a frontalization layer 104, a convolution layer 106, a bias layer 108, a non-linearity layer 110, a pooling layer 112, a loss calculation layer 114, and a label layer 116. In some examples, each layer of the deep convolutional network 100 may have hundreds to thousands of parameters that can be initialized and then adjusted through training.

[0014] The example deep convolutional network 100 generally provides for mapping of a non-frontal object pose to a frontal object pose from left to right in Fig. 1, and a back projection of errors for training to adjust parameters using transforms that define the frontalization layer 104 from right to left. For example, the image detection layer 102 can detect an image. The image may include one or more objects with non-frontal poses such as faces. In some examples, the image detection layer 102 can send the image to the frontalization layer 104 as indicated by an arrow 118 for frontalization of the faces. No arrow is shown from the frontalization layer 104 to the image detection layer 102, as no errors are back-projected to the input image.

[0015] The frontalization layer 104 receives a two-dimensional image from the image detection layer 102 as indicated by an arrow 118 and outputs a frontalized two-dimensional image as indicated by an arrow 120. For example, a face frontalization can be performed through a 3D rotation and 3D translation as described herein. The 3D rotation can be parameterized through quaternions to avoid ambiguities within the rotation space that Euler angles have. A quaternion is a complex number of the form $w + xi + yj + zk$, where w , x , y , z are real numbers and i , j , k are imaginary units that satisfy certain conditions. This quaternion representation introduces four 3D rotation

parameters, which can be learned. Additional parameters may be used for 3D translation of the face. For example, three additional parameters can be used to represent translation in the x, y, and z axes of a coordinate frame.

[0016] In some examples, seven parameters may thus be used for face frontalization. These seven parameters transform a face from any given pose into the frontal pose. In some examples, the frontalization layer can receive an image of a face in any pose as input and estimate the pose of this input face through the estimation of the seven parameters. These parameters describe a transform from the face in the image, which can appear in any pose, to a 3D face model that appears in a frontal pose. Furthermore, bilinear interpolation from the 3D face model into the 2D image space can be used to rasterize an image of the face in frontal pose. This rasterized 2D image is the output from the frontalization layer 104 as indicated by an arrow 120. In some examples, the frontalization layer 104 can also receive projected output loss feedback as indicated by an arrow 122. The operation of an example frontalization layer is discussed at greater length with respect to the example frontalization layer of Fig. 2 below.

[0017] The convolutional layer 106 reuses the same parameters within a spatial neighborhood. Reuse, as used herein, is defined as a convolutional operation between the parameters and input values. The parameters may be a set of variables that define the convolutional kernel. In the example of Fig. 1, the input values for the convolutional layer 106 can come from the frontalized image that is received from the frontalization layer 104. The convolutional layer 106 can also receive values such as back-projected errors from the bias layer 108.

[0018] The bias layer 108 of the example deep convolutional network 100 computes an additive bias to each value. In some examples, the bias can be a free parameter adjusted through training.

[0019] The non-linearity layer 110 computes the hyperbolic tangents of all received values and has neither a spatial interaction nor any parameters. The hyperbolic tangent provides a non-linearity important to provide learning and generalization properties of a deep convolutional network. Instead of using the

hyperbolic tangent rectified linear units (setting negative values to zero) can be used as well. However; it is important to have a non-linearity.

[0020] The pooling layer 112 combines values within a spatial neighborhood to form a single output. For example, the pooling layer 112 can use max-pooling, which computes the maximum of four spatially neighboring locations in an image as an output. Such max pooling can reduce the width and height of an image by a factor of two in width and height. The pooling layer 112 has no trainable parameters. However, the range of max-pooling and stride length are fixed parameters.

[0021] The loss calculation layer 114 computes an output loss to be projected back to the other layers. For example, the loss calculation layer 114 can compute a soft-max function between a ground-truth label received from the label layer 116 as indicated by an arrow 142 and a predicted label received from, e.g. the pooling layer 112 as indicated by an arrow 136, or, more generally, any other layer preceding the loss layer 112 as indicated by arrows 120, 124, 128, and 132, to generate the output loss. As used herein, a soft-max function refers to a generalization of the logistic function that "squashes" a K-dimensional vector $\mathbf{z}$ of arbitrary real values to a K-dimensional vector $\sigma(\mathbf{z})$ of real values in the range (0, 1) that add up to 1. As used herein, a label refers to a class identifier (CID) for a given data point. A class identifier, as used herein, refers to an identifier for a particular class of objects. A predicted label refers to the output of the deep convolutional network 100 given a particular input. A ground-truth label refers to a label supplied by a user. This output loss can then be projected back through layers 104-112 as indicated by arrows 122, 126, 130, 134, and 138. In particular, as discussed in detail with regard to Fig. 2 below, the frontalization layer 104 may receive the output loss and adjust its parameters accordingly. In some examples, one or more other layers may also receive the output loss and either process the output loss or pass the output loss onto the next layer in the chain.

[0022] Thus, the label layer 116 may receive ground-truth labels from one or more users. For example, to classify bananas and apples from images, images of bananas and apples may be received and bananas may be assigned a CID

of 0 and apples may be assigned a CID of 1. For example, a user can label images as bananas or apples. A predicted label is the output of the deep convolutional network 100 given an image of a banana or apple as an input. For example, the output can be a CID of 0 or 1.

[0023] The diagram of Fig. 1 is not intended to indicate that the example deep convolutional network 100 is to include all of the components shown in Fig. 1. Rather, the example deep convolutional network 100 can include fewer or additional components not illustrated in Fig. 1 (e.g., additional layers, etc.) as indicated by dashed lines 140. Moreover, the diagram of Fig. 1 is not intended to indicate that the components of the example deep convolutional network 100 are to be arranged in any particular order. For example, the frontalization layer 104 is shown between the image detection layer 102 and the convolutional layer 106, but can be alternatively plugged into the deep convolutional network 100 between any two layers.

[0024] Fig. 2 is a block diagram showing an example frontalization layer. The frontalization layer is generally referred to by the reference number 200 and can again be implemented using the example computing device 302 of Fig. 3 below.

[0025] The frontalization layer 200 includes grid generator component 202 and a rasterization component 204. The grid generator component 202 can receive pose parameters Θ as indicated by an arrow 206 and a supplied 3D model M as indicated by an arrow 208 and generate a two-dimensional (2D) sample grid G . For example, the 3D model may be retrieved from a database and represent a mean face given a plurality of 3D point cloud reconstructions of thousands of faces. The 2D sample grid G can define the mapping of the 2D non-frontal pose into a 2D frontal pose of the object in the image. The 2D sample grid G can then be sent to the rasterization component 204 as indicated by arrow 210. The rasterization component 204 can receive the non-frontalized object view in an 2D input image I as indicated by an arrow 212 and rasterize a frontalized object view in the 2D image I' as indicated by an arrow 214. The above steps thus summarize the computations of the forward direction corresponding to left to right in Fig. 1 above.

[0026] The backward direction of Fig. 2 indicated by arrows pointing from right to left can start with the rasterization component 204 receiving a differential image dl' as indicated by arrow 216. The differential image dl' may contain errors or differentials for the given frontalized object view and object classification. For example, the differential image dl' can include a mismatch between the predicted label and ground-truth label. Thus, differential refers to the partial derivative of the output loss with regards to input values. In the example of Fig. 2, $d(\text{loss})/d(l')$ feeds into the frontalization layer and $d(\text{loss})/d(l)$ is output in reverse direction to the layer that feeds this frontalization layer. In some examples, through the use of partial derivatives, this differential image dl' can then be linked back to the used sampling grid G resulting in the differential sampling grid dG . In turn, the differential 2D sampling grid dG can be related back as indicated by an arrow 218 to differentials in the pose parameters Θ , indicated by the differentials $d\Theta$ 222. The differential frontalized 2D image dl' 216 can likewise be related back to a differential 2D image dl as indicated by an arrow 220. In some examples, the pose parameters Θ may only exist in the grid generator G , which is indicated in Fig. 2 by the connection of the inward 206 and outward 222 going arrows for pose parameters with a dashed line 224. For example, the pose parameters are internally used in the frontalization layer of the example deep convolution layer of Fig. 1 above.

[0027] Accordingly, a backward pointing arrow for the dl of the frontalization is absent in Fig. 1. This is because the input image has no free parameters that need to be adjusted. In some examples, however, the frontalization layer could alternatively be at other positions in the chain of layers of a deep convolutional network. In those cases, a backward pointing arrow may be used to indicate the relation back of parameters.

[0028] In some examples, pose parameters can be learned through error minimization over training samples. For example, a given data set may include 10 individuals, with 25 poses for each individual. For example, the 25 poses may include 5 elevation angles each at 5 azimuth angles. Each individual may also be present 100 times in each pose. Thus, the training set may have a total of $10 \times 25 \times 100 = 25,000$ samples. The frontalization layer can be initialized

using the identity transform, assuming only frontal poses. Any non-frontal face that is presented to the network produces a large error for a given loss function on the output when compared to the frontal view. For example, the non-frontal face may have an azimuth angle of 10 degrees and an elevation angle 5 degrees. This error can be calculated at another layer and back-propagated through the deep convolutional network to correct the seven parameters in the frontalization layer. For example, the error can be calculated at the loss layer of Fig. 1 above. Thus, the parameters of the frontalization layer can be trained using any suitable training set. Moreover, the learning of individualized parameters for the transform can compensate for any inaccuracies of the mean 3D face model.

[0029] Fig. 3 is a block diagram of a system that can frontalize objects detected within an image. The system is generally referred to by the reference number 300.

[0030] The system 300 may include a computing device 302, and one or more client computers 304, in communication over a network 306. As used herein, a computing device 302 may include a server, a personal computer, a tablet computer, and the like. As illustrated in Fig. 3, the computing device 302 may include one or more processors 308, which may be connected through a bus 310 to a display 312, a keyboard 314, one or more input devices 316, and an output device, such as a printer 318. The input devices 316 may include devices such as a mouse or touch screen. The processors 308 may include a single core, multiples cores, or a cluster of cores in a cloud computing architecture. In some examples, the processors 308 may include a graphics processing unit (GPU). The computing device 302 may also be connected through the bus 310 to a network interface card (NIC) 320. The NIC 320 may connect the computing device 302 to the network 306.

[0031] The network 306 may be a local area network (LAN), a wide area network (WAN), or another network configuration. The network 306 may include routers, switches, modems, or any other kind of interface device used for interconnection. The network 306 may connect to several client computers 304. Through the network 306, several client computers 304 may connect to

the computing device 302. Further, the computing device 302 may access images across network 306. The client computers 304 may be similarly structured as the computing device 302.

[0032] The computing device 302 may have other units operatively coupled to the processor 308 through the bus 310. These units may include non-transitory, tangible, machine-readable storage media, such as storage 322. The storage 322 may include any combinations of hard drives, read-only memory (ROM), random access memory (RAM), RAM drives, flash drives, optical drives, cache memory, and the like. The storage 322 may include a store 324, which can include any images captured or generated in accordance with an embodiment of the present techniques. Although the store 324 is shown to reside on computing device 302, a person of ordinary skill in the art would appreciate that the store 324 may reside on the computing device 302 or any of the client computers 304.

[0033] The storage 322 may include a plurality of modules 326. For example, the modules 326 may be a set of instructions stored on the storage device 322, as shown in Fig. 3. The instructions, when executed by the processor 308, may direct the computing device 302 to perform operations. In some examples, the instructions can be executed by a graphics processing unit (GPU). In some examples, the grid generator 328, rasterizer 330, and/or task performer 332 may be implemented as logic circuits or computer-readable instructions stored on an integrated circuit such as an Application Specific Integrated Circuit (ASIC), a Field Programmable Gate Array (FPGA), or other type of processor. The grid generator 328 can receive an image of an object and a three-dimensional (3D) model of the object. For example, the 3D model can represent a mean object based on 3D point cloud representations of a plurality of objects. In some examples, the objects may be faces. The grid generator 328 can also estimate a pose of the object in the image based on estimation of a plurality of parameters. For example, the parameters can include four parameters describing a quaternion to represent a 3D rotation and three components of a vector representing a 3D translation. In some examples, the plurality of parameters describe a 3D rotation and 3D translation. For

example, the plurality of parameters can be learned via error minimization over a plurality of training samples. In some examples, the grid generator 328 can generate a two-dimensional sample grid based on the estimated pose parameters. The rasterizer 330 can generate a frontalized image of the object based on the estimated pose and the 3D model of the object.

[0034] The task performer 332 can detect an object in another image based on a comparison with the frontalized image. In some examples, the task performer 332 can identify a person in another image based on the frontalized image. In some examples, the task performer 332 can verify that a person appears in another image based on the frontalized image. In some examples, the task performer can detect a frontalized face within an image. The client computers 304 may include storage similar to storage 322. For example, the storage may be the non-transitory, tangible computer-readable medium of Fig. 5 below.

[0035] Fig. 4 is a process flow diagram showing a method of generating frontalized images of objects. The example method is generally referred to by the reference number 400 and can be implemented using the processor 308 of the example system 300 of Fig. 3 above.

[0036] At block 402, the processor receives an image of an object and a three-dimensional (3D) model of the object. For example, the object can be the face of a person. The 3D model can be a face model. For example, the face model may represent a mean face based on 3D point cloud representations of a plurality of faces.

[0037] At block 404, the processor estimates a pose of the object in the image based on estimation of a plurality of parameters. For example, the plurality of parameters describe a 3D rotation and 3D translation and are to be learned via error minimization over a plurality of training samples. In some examples, the processor can receive a differential frontalized image and project back a differential sample grid to generate a plurality of differential pose parameters to be used to generate a set of new pose parameters. In some examples, the processor can receive a differential frontalized image and projecting back a differential image. For example, the differential frontalized

image may be received from a layer of a deep convolutional network and the differential image sent to another layer of the deep convolutional network.

[0038] At block 406, the processor generates a frontalized image of the object based on the estimated pose and the 3D model of the object. For example, the processor can rasterize the frontalized image through a bilinear interpolation of the model of the object into a two-dimensional image space.

[0039] At block 408, the processor detects, identifies, or verifies an object based on the frontalized image. For example, the processor can detect a face in an image based on the frontalized image. In some examples, the processor can identify a person in an image based on the frontalized image. In some examples, the processor can verify that a person appears in an image based on the frontalized image.

[0040] This process flow diagram is not intended to indicate that the blocks of the example method 400 are to be executed in any particular order, or that all of the blocks are to be included in every case. Further, any number of additional blocks not shown may be included within the example method 400, depending on the details of the specific implementation.

[0041] Fig. 5 is a block diagram showing a non-transitory, tangible computer-readable medium that stores code for frontalization. The non-transitory, tangible computer-readable medium is generally referred to by the reference number 500.

[0042] The non-transitory, tangible computer-readable medium 500 may correspond to any storage device that stores computer-implemented instructions, such as programming code or the like. For example, the non-transitory, tangible computer-readable medium 500 may include one or more of a non-volatile memory, a volatile memory, and/or one or more storage devices.

[0043] Examples of non-volatile memory include, but are not limited to, electrically erasable programmable read only memory (EEPROM) and read only memory (ROM). Examples of volatile memory include, but are not limited to, static random access memory (SRAM), and dynamic random access memory (DRAM). Examples of storage devices include, but are not limited to, hard

disks, compact disc drives, digital versatile disc drives, and flash memory devices.

[0044] A processor 502 generally retrieves and executes the computer-implemented instructions stored in the non-transitory, tangible computer-readable medium 500 for frontalization of faces in images. A grid generator module 504 can receive an image of a face and a three-dimensional (3D) face model. In some examples, the module 504 can estimate a pose of the face in the image based on estimation of a plurality of parameters. The plurality of parameters can be learned via error minimization over a plurality of training samples. For example, the plurality of parameters may describe a 3D rotation and 3D translation. In some examples, the 3D rotation parameters may be four quaternions.

[0045] A rasterizer module 506 can generate a frontalized image of the face based on the estimated pose and the 3D face model. In some examples, the rasterizer module 506 can receive a differential frontalized image and project back a differential sample grid to generate a plurality of differential pose parameters to be used to generate a set of new pose parameters. For example, the new pose parameters can be used to generate an updated frontalized image. In some examples, the rasterizer module 506 can rasterize the frontalized image via a bilinear interpolation of the 3D face model into a two-dimensional image space. In some examples, the rasterizer module 508 can receive a differential frontalized image and project back a differential image.

[0046] A task module 508 can detect a face in another image based on the frontalized image. For example, the face can be detected in a particular portion of the other image. In some examples, the task module 508 can also identify a person in another image based on the frontalized image. For example, given a particular person's face stored in a database of frontalized images, the same person's face can be identified in additional images based on the frontalized image. In some examples, the task module 508 can verify that a person appears in another image based on the frontalized image. For example, the task module 508 can be used to compare faces having different poses in two images.

[0047] Although shown as contiguous blocks, the software components can be stored in any order or configuration. For example, if the computer-readable medium 500 is a hard drive, the software components can be stored in non-contiguous, or even overlapping, sectors.

[0048] The present techniques are not restricted to the particular details listed herein. Indeed, those skilled in the art having the benefit of this disclosure will appreciate that many other variations from the foregoing description and drawings may be made within the scope of the present techniques. Accordingly, it is the following claims including any amendments thereto that define the scope of the present techniques.

CLAIMS

What is claimed is:

1. A method for generating pose frontalized images of objects, comprising:
receiving an image of an object and a three-dimensional (3D) model of the object;
estimating, via a processor, a pose of the object in the image based on estimation of a plurality of parameters, wherein the plurality of parameters describe a 3D rotation and 3D translation and are to be learned via error minimization over a plurality of training samples; and
generating, via the processor, a frontalized image of the object based on the estimated pose and the 3D model of the object.
2. The method of claim 1, wherein learning the plurality of parameters further comprises receiving a differential frontalized image and projecting back a differential sample grid to generate a plurality of differential pose parameters to be used to generate a set of new pose parameters.
3. The method of claim 1, wherein generating the frontalized image further comprises rasterizing the frontalized image via a bilinear interpolation of the model of the object into a two-dimensional image space.
4. The method of claim 1, further comprising receiving a differential frontalized image and projecting back a differential image.
5. The method of claim 1, further comprising detecting a face in another image based on the frontalized image, identifying a person in another image based on the frontalized image, verifying that a person appears in another image based on the frontalized image, or any combination thereof.

-15-

6. A system for generating pose frontalized images of objects, comprising:
a grid generator to receive an image of an object and a three-dimensional (3D) model of the object and estimate a pose of the object in the image based on estimation of a plurality of parameters, wherein the plurality of parameters describe a 3D rotation and 3D translation and are to be learned via error minimization over a plurality of training samples; and
a rasterizer to generate a frontalized image of the object based on the estimated pose and the 3D model of the object.
7. The system of claim 6, wherein the grid generator is to further generate a two-dimensional sample grid based on the estimated pose parameters.
8. The system of claim 6, further comprising a task performer to detect an object in another image based on a comparison with the frontalized image, identify a person in another image based on the frontalized image, verify that the person appears in another image based on the frontalized image, or any combination thereof.
9. The system of claim 6, wherein the 3D model represents a mean object based on 3D point cloud representations of a plurality of objects.
10. The system of claim 6, wherein the parameters comprise four quaternions representing the 3D rotation and three components of a vector representing the 3D translation.
11. A non-transitory, tangible computer-readable medium, comprising code to direct a processor to:
receive an image of a face and a three-dimensional (3D) face model;
estimate a pose of the face in the image based on estimation of a plurality of parameters, wherein the plurality of parameters

-16-

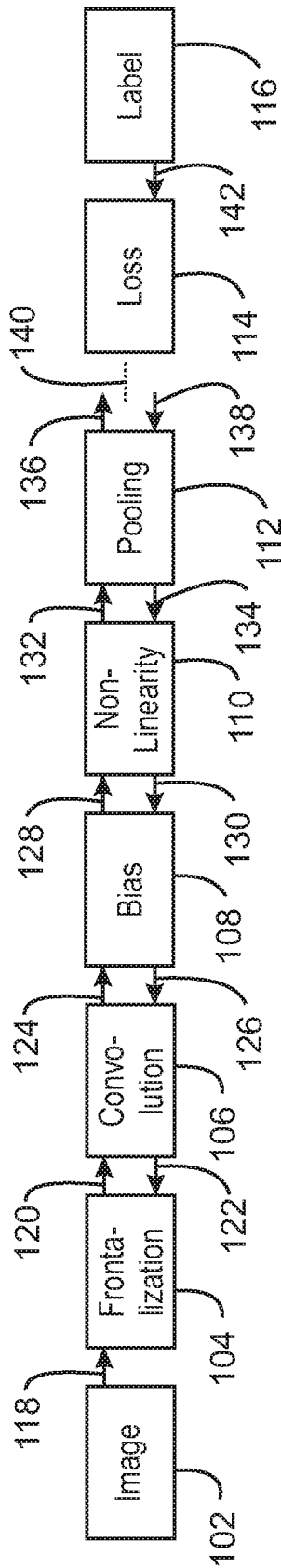
describe a 3D rotation and 3D translation and are to be learned via error minimization over a plurality of training samples; and generate a frontalized image of the face based on the estimated pose and the 3D face model.

12. The non-transitory, tangible computer-readable medium of claim 11, further comprising code to direct the processor to receive a differential frontalized image and project back a differential sample grid to generate a plurality of differential pose parameters to be used to generate a set of new pose parameters.

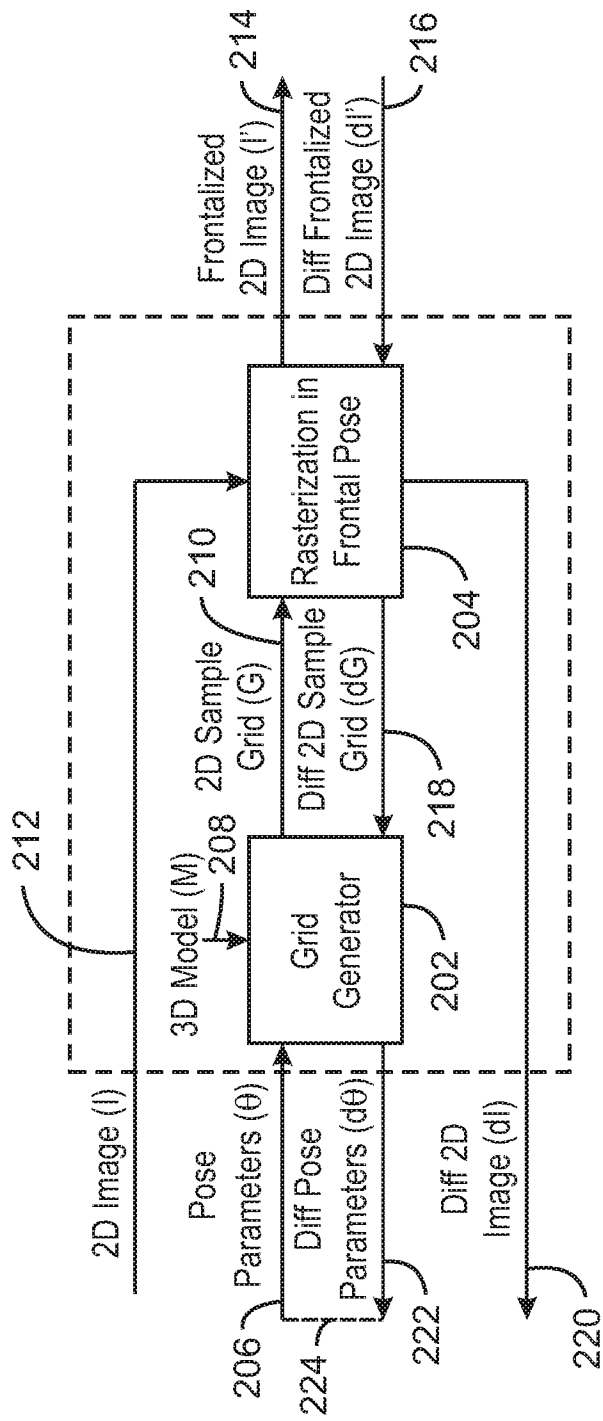
13. The non-transitory, tangible computer-readable medium of claim 11, further comprising code to direct the processor to rasterize the frontalized image via a bilinear interpolation of the 3D face model into a two-dimensional image space.

14. The non-transitory, tangible computer-readable medium of claim 11, further comprising code to direct the processor to receive a differential frontalized image and project back a differential image.

15. The non-transitory, tangible computer-readable medium of claim 11, further comprising code to direct the processor to detect the face in another image based on the frontalized image, identify a person in another image based on the frontalized image, verify that a person appears in another image based on the frontalized image, or any combination thereof.

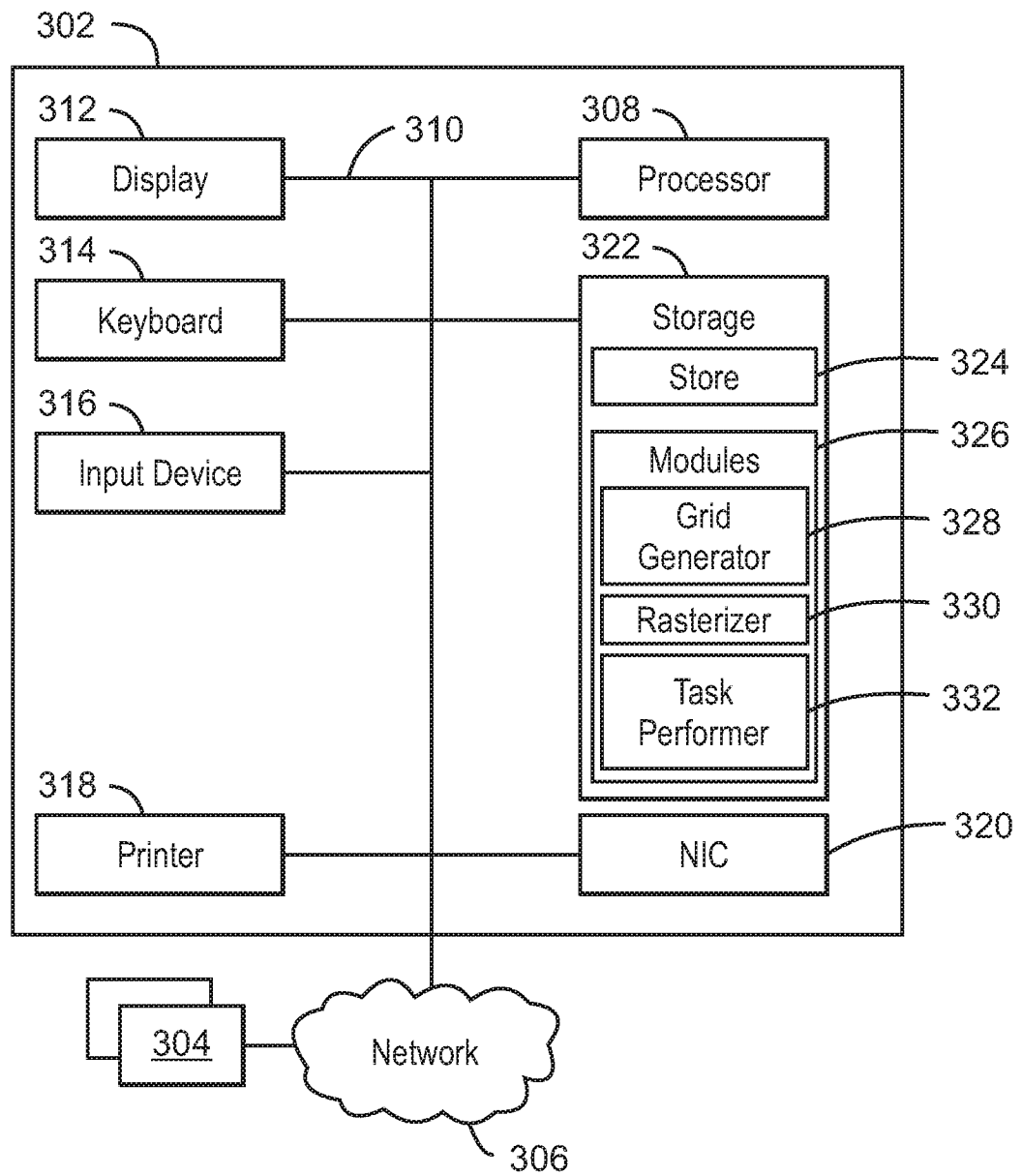


100
FIG. 1



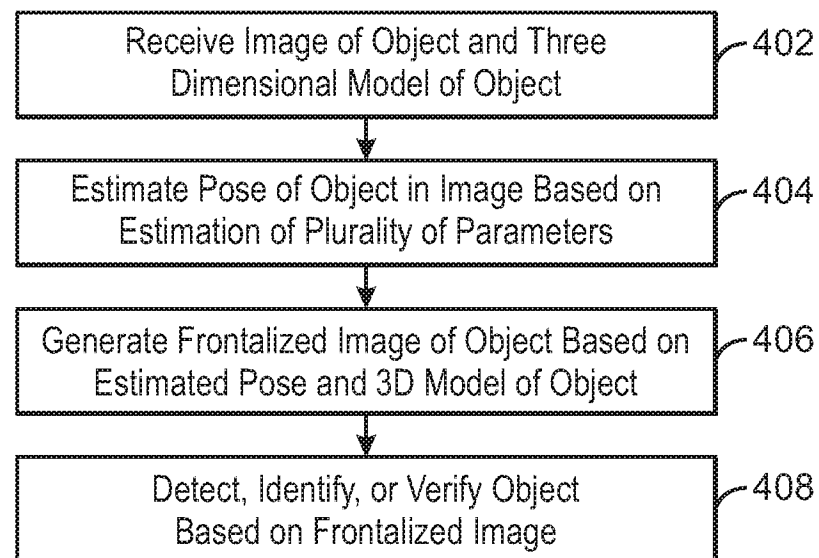
200
FIG. 2

3/5



300
FIG. 3

4/5



400
FIG. 4

5/5

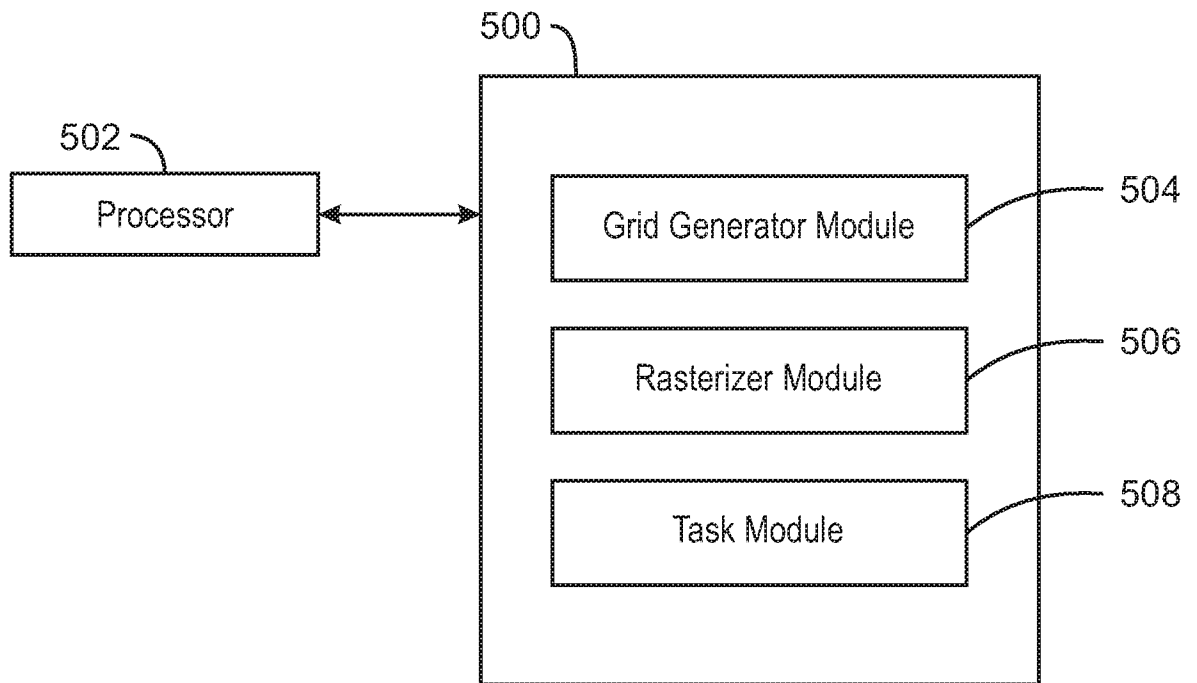


FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2016/015181**A. CLASSIFICATION OF SUBJECT MATTER****G06T 7/00(2006.01)i, G06T 17/00(2006.01)i**

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

B. FIELDS SEARCHEDMinimum documentation searched (classification system followed by classification symbols)
G06T 7/00; G06T 15/04; G06K 9/52; H04N 7/15; G06T 15/20; G06K 9/00; G06T 17/00Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility modelsElectronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: face, frontalization, pose, 3D, and similar terms.**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	TAL HASSNER et al., `Effective Face Frontalization in Unconstrained Images`, arXiv:1411.7964v1 [cs.CV], 28 November 2014, pp. 1-10 (http://arxiv.org/pdf/1411.7964.pdf) See pages 1-4, 7.	1-4, 6-7, 9-14
A		5, 8, 15
A	KR 10-2011-0088361 A (KOREA ELECTRONICS TELECOMM et al.) 03 August 2011 See paragraphs [0019], [0043], [0078]; and figure 5.	1-15
A	US 2009-0309878 A1 (TAKAHIDE OTANI et al.) 17 December 2009 See paragraphs [0046]-[0050]; and figure 1.	1-15
A	US 2015-0161435 A1 (ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE) 11 June 2015 See paragraphs [0052]-[0069]; and figure 2.	1-15
A	WO 2013-187551 A1 (CENTER OF HUMAN-CENTERED INTERACTION FOR COEXISTENCE) 19 December 2013 See page 13, line 19 - page 14, line 12; and figure 10.	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

10 October 2016 (10.10.2016)

Date of mailing of the international search report

11 October 2016 (11.10.2016)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

BYUN, Sung Cheal

Telephone No. +82-42-481-8262



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2016/015181

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
KR 10-2011-0088361 A	03/08/2011	None	
US 2009-0309878 A1	17/12/2009	CN 101604387 A CN 101604387 B EP 2133819 A2 EP 2133819 A3 JP 04569670 B2 JP 2009-301170 A KR 10-2009-0129357 A TW 201003570 A	16/12/2009 18/04/2012 16/12/2009 28/11/2012 27/10/2010 24/12/2009 16/12/2009 16/01/2010
US 2015-0161435 A1	11/06/2015	KR 10-2015-0065445 A	15/06/2015
WO 2013-187551 A1	19/12/2013	EP 2860969 A1 JP 05839427 B2 JP 2014-522622 A KR 10-1350996 B1 KR 10-2013-0138533 A US 2014-0218467 A1 US 9077974 B2	15/04/2015 06/01/2016 04/09/2014 13/01/2014 19/12/2013 07/08/2014 07/07/2015