Pattern Recognition 94 (2019) 196–206

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

A deep Coarse-to-Fine network for head pose estimation from synthetic data

Yujia Wang[a], Wei Liang[a,*], Jianbing Shen[a], Yunde Jia[a], Lap-Fai Yu[b]

[a] Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, PR China
[b] Department of Computer Science, George Mason University, Vienna, VA, USA

ARTICLE INFO

Article history:
Received 11 December 2018
Revised 5 May 2019
Accepted 14 May 2019
Available online 21 May 2019

Keywords:
Head pose estimation
Coarse-to-Fine
Joint learning

ABSTRACT

Various applications of human-computer interaction are based on the estimation of head pose, which is challenging due to different facial appearance, inhomogeneous illumination, partial occlusion, etc. In this paper, we propose a deep neural network following the *Coarse-to-Fine* strategy to estimate head poses. The scheme includes two branches: *Coarse classification* phase classifying the input image into four categories, and *Fine Regression* phase estimating the accurate pose parameters. The two sub-networks are trained jointly. To tackle the problem of insufficient annotated data in training process, we design a rendering pipeline to synthesize realistic head images and generate an annotated dataset with a collection of 310*k* head poses. The results on benchmark datasets and synthetic dataset validate the effectiveness of our approach, as well as the results on images with diverse illumination, occlusion, and motion blur. Moreover, our method can be easily extended to estimate head poses on depth images.

1. Introduction

Head pose estimation has been applied to broad applications in human-computer interactions, *e.g.* gaze detection, driving assistance, disabled assistance, and entertainment. Head pose is also used to understand human attention, behavior or intention, which has been studied and examined extensively in cognitive psychology and neurophysiology community [1].

Over the past several years, head pose estimation remains an attractive research topic, since it is still challenging due to the diversity of the head appearance caused by the head motion and various head pose changes, such as facial texture, inhomogeneous illumination, partially occlusion, etc. A number of algorithms have been proposed to address the head pose estimation problem, and a good survey can be referred to [2].

In general, the existing approaches can be divided into two streams: *classification* and *regression*. Classification approaches aim to classify the head pose into a discrete space, that is, estimating the head pose by assigning a discrete pose category label to each input. These approaches are relatively robust to large head pose variation but *with sparse solution space, e.g.* 15° intervals for each category. On the other hand, regression approaches usually estimate head pose by fitting a regression model on training data to output continuous angles. The results of these regression approaches can reflect small changes of head pose. However, employing individual regression model for accurate continuous head pose estimation *increases model complexity*.

To address the challenge, we approach our method in a coarse-to-fine manner, which leverages the robustness of the classification method and the sensitivity of the regression method to small changes of head pose. In particular, we first determine the specific category of the input image, thus narrowing down the solution space. Then a regression network is selected on the basis of the output category to estimate accurate pose parameters. The coarse-to-fine cascade enables our approach to increase the robustness of head pose estimation without increasing the model complexity and the difficulty in training process of the traditional regression networks. The two sub-networks are achieved via a deep learning model and share the same full-image convolutional feature map and perform joint learning, thus achieving computation efficiently.

Although, CNN techniques have shown good performance on a number of tasks, a major challenge on their application for head pose estimation is obtaining sufficient annotated head pose data, especially the data with variations of head appearance (*e.g.* expression, race, age, and gender), and environmental factors (*e.g.* occlusion, noise, and illumination). Previously released head pose datasets, such as Biwi Kinect Head Pose Dataset [3] and Pointing'04 dataset [4], consist of around 15*k* and 3*k* images, respectively. The limited amount of annotated head images in these datasets makes it hard to apply CNN techniques. We devise an approach for

* Corresponding author.
E-mail address: liangwei@bit.edu.cn (W. Liang).

https://doi.org/10.1016/j.patcog.2019.05.026
0031-3203/

synthesizing realistic head pose images with annotations to overcome the obstacle of insufficient annotated data in head pose estimation.

Our main contributions are summarized below:

(1) *A new deep learning framework following the coarse-to-fine strategy for estimating head pose.* We model the estimation with a cascade *Coarse-to-Fine* process, where the accurate pose parameter estimation is followed by rough pose classification. Both tasks are achieved via deep learning models, which share the same full-image convolutional feature map and perform joint learning. The proposed method spread the network complexity and training burden of traditional regression networks.
(2) *Synthetic head pose image generation.* A synthetic head pose image rendering pipeline is introduced in our approach for breaking the limitation of head pose image quantity, increasing the diversity of the head pose images with high resolution (*e.g.*, different illumination conditions, motion blur, and occlusion), and helping to improve the performance of estimation results. In addition, we conducted an experiment to verify the compatibility between synthetic data and real data. We generate a dataset with a collection of 310*k* head pose images.
(3) *Easily expandability and promising head pose estimation results.* The trained head pose estimation network yields very promising results on both synthetic dataset and real dataset. Moreover, the proposed deep learning framework and image rendering pipeline can be easily extended to handle the task of depth head pose image estimation.

2. Related work

Head pose estimation is an important topic in human-computer interaction, computer vision, and virtual reality that has attracted much research attention. Since the head pose data determines the human visual field, it can be efficient and intuitive applied to control 3D avatars, devices, or products. Head pose can also serve as an important pre-processing step for further analysis of gaze estimation [5,6], human-object interaction understanding [7–9], and human-robot interactions [10]. We review some representative work closely relevant to our problem.

2.1. Head pose estimation

Spurred by the growing demand for human-computer interaction, this field made large progress in the last years. There are two basic approaches currently being adopted for head pose estimation: classification-based approaches and regression-based approaches. The survey of [2] well-summarized head pose estimation techniques.

The classification methods learned a mapping between images and a discretized space of poses. Given a new image, the classifiers assign it to a discrete class [11]. Since the majority of such methods have discretized outputs, only allowing coarse head pose estimation, it is difficult to derive a reliable continuous estimation from the results.

Different from classification methods, regression methods estimate head pose by learning a functional mapping from the image space to one or more pose directions [12,13]. The allure of these approaches is that with a set of labeled training data, a model can be built to provide a precise pose estimation for any new data samples. Due to the breakthrough results achieved by deep learning technologies in many research field, Zavan et al. [14] proposed an automatic pipeline based on convolutional neural networks for detecting different facial regions, processing them, and combining the results generated from each, resulting in a robust head pose estimation and gender recognition. And some recent work can estimate head pose with high accuracy and perform in real time [15,16].

With the development of sensing technologies, such as Time-of-Flight sensors and the Microsoft Kinect, excellent capability and flexibility in capturing RGB-D images are provided. New methods involving both RGB and depth images are emerging. Hong et al. [17] formulated and solved head pose estimation as a multi-task learning problem, which was based on combining different types of features (i.e., features extracted from RGB images and depth image, respectively) with manifold learning method and multi-modal relationship mapping, thus improving the estimation performance. Such multi-modal learning formulation can also be used to solve clic prediction for web image reranking [18], and the manifold learning methods can be applied to human pose recovery [19,20], 3D object recognition [21], and scene recognition [22].

In comparison with these previous estimation methods, this paper focuses on combining classification and regression methods to address the head pose estimation task. More concretely, we use classification network to narrow down the solution space and use regression network with low computational complexity to estimate accurate parameters. We also prove that our method can be easily extended to solve the depth head pose estimation.

2.2. Coarse-to-Fine strategy

Lately, the *Coarse-to-Fine* strategy is proposed to improve the effectiveness and performance of many tasks, i.e., aiming to estimate the parameters roughly in the coarse stage and obtains precise parameters in the fine stage.

For image classification, Zhang et al. [23] learned a Visual-Semantic Tree to organize image categories hierarchically in a coarse-to-fine manner. In face detection tasks, some researchers proposed to combines classifiers in a cascade structure [24], which allowed background regions of the image to be quickly discarded while more computation was spent on promising face-like regions. Pavlakos et al. [25] employed a step-wise approach to predict human pose, which consists of a convolutional network for 2D joint localization and a subsequent optimization step to recover 3D pose.

To estimate head pose, Wu et al. [26] proposed a two-stage scheme based on the rationale that visual cues of head pose had unique multi-resolution spatial frequency characterization and structural signature. In the first stage, they projected the head image to a Gabor wavelet transform subspace and determined whether the true pose located in the subset. In the second stage, they used a structural landmark analysis in the transformer domain to refine the estimation.

Inspired by previous works, we propose a cascade CNNs in a *Coarse-to-Fine* framework to estimate head pose. In particular, we use the classification method in the coarse phase to narrow down the solution space and output a rough range in which the true pose is located. In the fine phase, we use the regression method to estimate the true pose from the small range of head pose constrained by the first phase output. Therefore, our proposed method spread the network complexity and training burden of the traditional regression networks.

2.3. Synthetic dataset

In the past decades, researchers have made impressive progress on 3D object modeling and synthesis. Synthesized data has been applied for deep network training in many computer graphics and vision tasks, *e.g.*, autonomous driving [27], license plate recognition

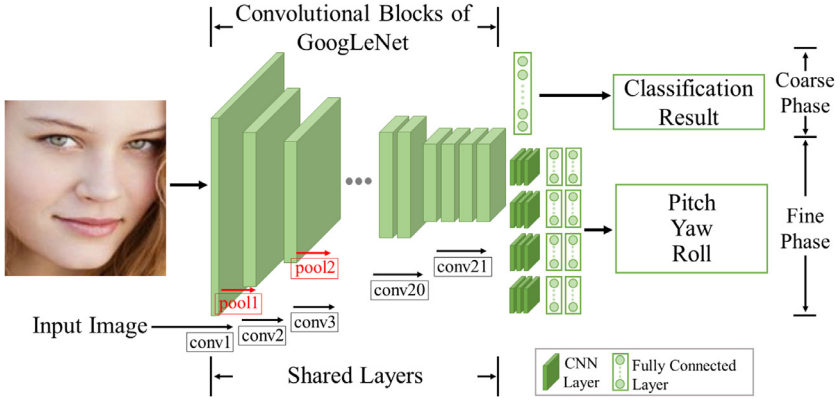

Fig. 1. Architecture of our deep estimation model. It consists of two phases: *Coarse Classification* phase and *Fine Regression* phase, which share several convolutional layers at the bottom. The two networks share several blocks borrowed from the first 21 convolutional blocks of GoogLeNet [34].

[28], 3D reconstruction [29], scene understanding [30], human detection and pose estimation [31], and medical image segmentation [32] and detection [33], which have proved that synthetic data can help to achieve good performance.

In the follow-up phase of the study, we find that the head pose datasets (*e.g.*,. Biwi Kinect Head Pose Dataset [3] and Pointing'04 Dataset [4]) are not sufficient for training a deep network due to the limited variation, the number of available samples, and partially incomplete annotation. Therefore, we devise a rendering pipeline to synthesize head poses with precise angles considering different gender, race, and age. During the generating process, we consider both head structure and face texture, which needs a more complex and detailed geometric structure than the generative models used in the previous works.

3. Approach

In this paper, a head pose is parameterised as three angles of head orientation $\Theta = (\theta_p, \theta_y, \theta_r)$, $\theta_p \in [-75^\circ, 75^\circ]$, $\theta_y \in [-50^\circ, 50^\circ]$, $\theta_r \in [-20^\circ, 20^\circ]$ [35]. θ_p, θ_y and θ_r represent the pitch, yaw, and roll, respectively.

We model head pose estimation in a coarse-to-fine framework, which first classifies the input image into one of the four categories of head poses, and then a regression network is applied to estimate the final parameters of head pose. The estimation algorithm is decomposed into two cascaded deep learning stages, namely, *Coarse Classification* phase and *Fine Regression* phase. As demonstrated in Fig. 1, these networks share several convolutional blocks in the bottom and perform joint learning for head pose estimation, which will be detailed in the following sections.

3.1. Coarse classification

In this section, we introduce our model for roughly estimating head pose (the Coarse Phase in Fig. 1). The bottom of the classification network is a stack of convolutional layers, which are borrowed from the first 21 convolutional blocks of GoogLeNet [34]. We define four categories to narrow down estimation range. Thus this network takes a head image as input and outputs a specific classification result. After that, the final estimation result is obtained within the certain category based on a regression network.

Category. According to the sign of three angles of orientation, we classify head poses into four categories. As shown in Table 1, the labels are defined as $\mathcal{L} = \{L_1, L_2, L_3, L_4\}$. The reason that we ignore the roll angle is that the change of head appearance along the roll axis is not obvious. Thus splitting the space of roll angle will increase classification error, which will be propagated to the regression phase and will result in performance degradation. Therefore, we just take pitch and yaw angles into account. We conduct evaluations of our classification strategy in the section of the experiment.

Table 1
Head pose category definition. The angle range of roll (θ_r) is fixed as $[-20^\circ, 20^\circ]$ throughout all categories.

Label	Range of pitch (θ_p)	Range of yaw (θ_y)
L_1	$[0, 75^\circ]$	$[0, 50^\circ]$
L_2	$[-75^\circ, 0)$	$[0, 50^\circ]$
L_3	$[-75^\circ, 0)$	$[-50^\circ, 0)$
L_4	$[0, 75^\circ]$	$[-50^\circ, 0)$

Loss function. We build a fully-connected layer and design a new linear Softmax classifier for the sub-spaces classification. The new Softmax classifier is trained on data with category annotations.

The goal of training a CNN is to maximize the probability of the correct class, which is achieved by minimizing the softmax loss. Let $C = \{(x_i, y_i)|i \in [1, N]\}$ be the training set. Each image x_i is associated with a label $y_i \in [1, K]$, where $K = 4$. The softmax function is a normalized exponential and is defined as:

$$y_j = \sigma_j(z) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \qquad \text{for } j = 1, \ldots, K \tag{1}$$

where z is the vector that is cast to the softmax layer. The denominator $\sum_{k=1}^{K} e^{z_k}$ acts as a regularizer to make sure that $\sum_{k=1}^{K} y_k = 1$. As the output layer of a neural network, the softmax function can be represented graphically as a layer with K neurons.

3.2. Fine regression

With the obtained category by the classification network, a regression network is applied to estimate the final head pose parameters. We achieve this via a regression network, which takes the 1024-D feature vectors extracted by the share convolutional layers as input, and outputs the head pose parameters, i.e., pitch, yaw, and roll (the Fine Phase in Fig. 1). The bottom of the regression network is the first 21 convolutional blocks of GoogLeNet. We train four regression networks simultaneously, that is, each sub-space in *Coarse Classification* has a corresponding regression network.

Regression Architecture. The bottom of the regression network is the first 21 convolutional blocks of GoogLeNet. On the top of the last convolutional layer, we adopt a three-layer regression network to estimate the final outputs. This regression network contains 3 convolutional layers with a kernel of 5×5 pixels, and 3 max-pooling layers with a kernel of 2×2 pixels. The stride of convolutional layers is set to be 1 and 2 in the pooling layers.

After each convolution layer, we use ReLU to accelerate convergence. Subsequent to the convolutional layers, there are 2 fully-connected layers with 300 hidden units. Moreover, we initialize all weights in each convolutional layer by a zero-mean Gaussian distribution with a standard deviation of 0.01 and use XavierFilter to initialize the weights in fully-connected layers. To alleviate overfitting and enhance the robustness of the CNN model, we employ "dropout" method and set different probabilities in each hidden layers. In the output layer, there is a linear activation function to predict the parameters of head poses Θ. We train the regression network using stochastic gradient descent.

Loss function. We use Euclidean loss (the sum of the individual losses) to measure the distance between ground truth and the predicted value. To avoid the risk of scale unbalance in the network, we add sigmoid operation before the Euclidean loss. Otherwise, the network optimization will crash due to a gradient overflow problem. Let $R = \{(x_i, \Theta_i)|i \in [1, N]\}$ be the set of training data, where x_i denotes an RGB image and Θ_i represents the corresponding ground truth ($\Theta = (\theta_p, \theta_y, \theta_r)$, $\theta_p \in [-75^\circ, 75^\circ]$, $\theta_y \in [-50^\circ, 50^\circ]$, $\theta_r \in [-20^\circ, 20^\circ]$). We define the following loss function and cast the problem of training as a minimization problem:

$$w^* = \arg\min_w L(w) = \arg\min_w \frac{1}{N} f(x_i; w) + \lambda r(w), \quad (2)$$

where λ is the parameter of weight decay, and $r(w)$ is the regularization term (L2-norm) that penalizes large weights to improve generalization. This is achieved by minimizing the loss term between the prediction angles and the corresponding ground truth. $f(x_i; w)$ is a term of the loss defined as:

$$f(x_i; w) = \frac{1}{2}||\psi(x_i; w) - \Theta_i||_2^2, \quad (3)$$

where term $\psi(x_i; w)$ is the predicted value of x_i with the network weight w. Note that λ determines the trade-off between Euclidean loss and large weights.

3.3. Synthetic head pose generation

In this section, we discuss how to obtain a large number of head pose images with precise annotations by rendering techniques. The synthetic data overcomes the barrier of limited data in the training process and enables CNN techniques to achieve good performance.

3D Head. The 3D head models used in our approach show varied textures and geometrical structures. We use a commercial software *FaceGen* to generate such models. This software reconstructs 3D models by computer vision techniques. In total, we generate 300 3D head models which involve different genders, races, and ages.

Rendering. We use Unity 3D engine to render RGB images for each head model automatically. The head model is imported into Unity 3D and placed on a virtual ground. In order to simulate the process of obtaining real data, we place the camera at a fixed position and set the relative distance between the head model center and camera center within a certain range which is generated randomly and limited from 800 to 1300 mm. We render from a perspective view to get realistic head images. In this virtual scene, the head model is rotated with sampled angles that follow a uniform distribution. The angles are used as the pose ground truth of synthesized head images. The pipeline is illustrated in Fig. 2.

It is still challenging to mimic real data. The synthetic data has no noise, whereas noise is very common in real data, which leads to a subtle appearance difference between the synthetic data and real data. To mimic the effect of noise, we add Gaussian noise to each generated image to simulate the sensor noise in the imaging process, similarly as in [36].

3.4. Implementation details

Training. Two datasets: our synthetic data and Biwi Kinect Head Pose Dataset, are used for training our model. The images in these two datasets are annotated with different categories and specific parameters (pitch, yaw, and roll). There are 260*k* synthetic images and 15*k* real images in total. Before feeding the input images and ground-truth to the network, the images are preprocessed by cropping to keep the face region only [37]; resizing to 256×256 pixels; converting to grayscale. For all training process, we use the same split strategy of data that includes both synthetic images and real images, 80% for training and 20% for testing.

Our two sub-networks are trained simultaneously. In each training iteration, we use a min-batch of 128 images, 64 of which are annotated with the groundtruth category labels, and the rest with pitch, yaw, and roll groundtruth. The whole training scheme of our model is presented in Fig. 3(a). The *conv1* to *conv21* blocks are shared between both tasks of classification and regression simultaneously using all the images in the batch. For the layers specialized for each sub-network, they are trained using only those images in the batch with the corresponding ground-truth. Concisely, in Fig. 3(a), we just illustrate one regression network as a representative of the regression phase.

Both classification and regression networks are initialized from the weights of GoogLeNet, which is pre-trained on the large-scale image classification dataset, ImageNet, with 1M images. The learning rate is set to be 0.01, weight decay is 0.0005, and momentum is 0.9. The network was trained over 40*k* iterations. Moreover, we use a GPU-based engine and optimize the network parameters using asynchronous stochastic gradient descent.

Testing. While our two sub-networks are trained in parallel, they work in a cascaded way (see Fig. 3(b)) during testing. Given an input image for estimation, we first classify the head pose into one of the four defined categories. Then one regression network is selected according to the category to estimate the final parameters of the head pose. Since the two initial convolutional blocks are shared between the classification and regression networks, we directly feed the convolutional feature into the selected regression network.

4. Experiments

We implemented our approach and conducted experiments on a Linux machine equipped with an Intel i7-5930K CPU and an Nvidia GeForce GTX 1080 GPU.

4.1. Dataset details

We evaluate our approach on both synthetic dataset and real dataset: synthetic images from our generated dataset, real images from Biwi Kinect Head Pose Dataset [3], Pointing'04 dataset [4] and Boston dataset [38]. To further verify the potential of the synthetic images when facing the dataset bias problem, we conduct analysis between the synthetic data and real data. Please refer to *supplementary material* for exemplary images and detailed analysis of these datasets.

Synthetic dataset: There are 310*k* RGB images totally, which are generated from 300 3D head models (200 males and 100 females). The RGB images have a resolution of 640×480 pixels. After cropping, the face image has 90×110 pixels. As shown in Fig. 4, the head pose range covers $\pm 75^\circ$ of pitch, $\pm 50^\circ$ of yaw, and $\pm 20^\circ$ of roll.

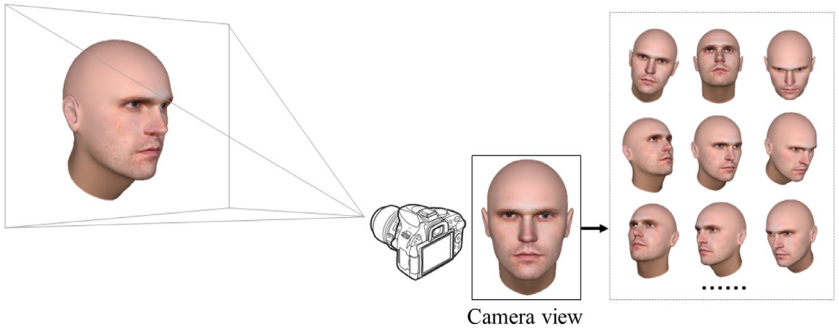

Fig. 2. The pipeline for generating synthetic head pose images. The 3D head is rotated along the pitch, yaw and roll axes based on the randomly generated ground truth.

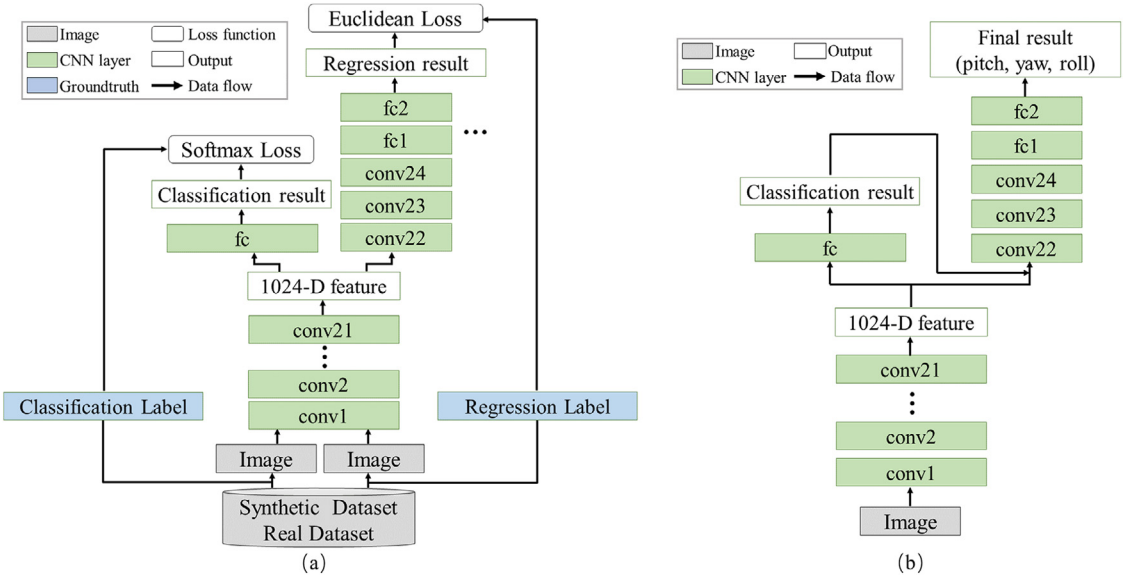

Fig. 3. Schematic diagram of our model in training (a) and testing (b). (The real dataset in (a) includes Biwi Dataset, Pointing'04 Dataset, and Boston Dataset.)

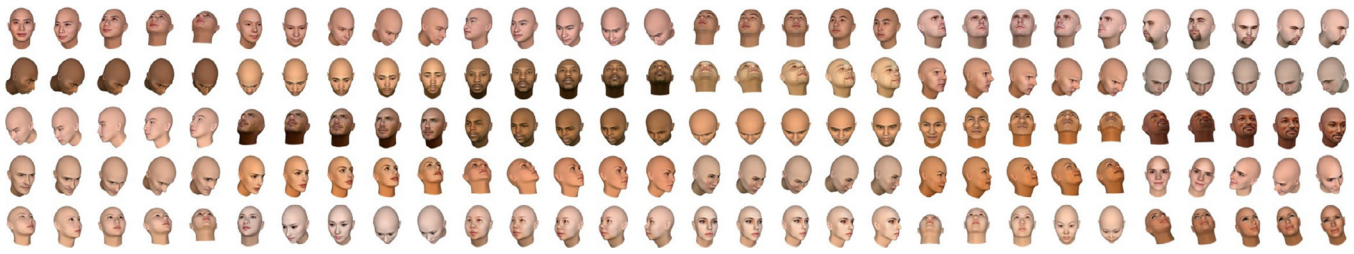

Fig. 4. Examplar synthetic head poses.

Biwi dataset: This dataset contains 15.7k images taken from 20 people. Each image has a resolution of 640×480 pixels, and a face typically consists of 90×110 pixels. The head pose range covers $\pm 50^\circ$ of pitch, $\pm 75^\circ$ of yaw and $\pm 20^\circ$ of roll. Additionally, to the best of our knowledge, this is the only released dataset that contains both RGB and depth images, which are shown in the first line of Fig. 5.

Pointing'04 dataset: This dataset contains 2.7k images taken from 14 people, who wear glasses or not and have varied skin colors. The head pose is represented by two angles: pitch and yaw. Each angle is a discrete value which varies from -90° to $+90^\circ$. The yaw has an interval of 15°, and the pitch has an interval of 15° and 30°. The dataset is demonstrated in the second line of Fig. 5.

Boston dataset: This dataset contains 14.4k images taken from 5 male. The ground-truth of each image changes continuously on pitch, yaw, roll. The range of these three angles covers $\pm 20^\circ$. Some examples are shown in the last line of Fig. 5.

Analysis: The presence of a bias in data collection has recently attracted a lot of attention in the computer vision community showing the limits in the generalization of any learning method trained on a specific dataset. To further verify the potential of the synthetic images when facing the dataset bias problem, we conduct an analysis on the existing dataset with continuous annotations (Biwi Dataset). We randomly extracted a sequence from the synthetic dataset and Biwi Dataset respectively, and visualized the distributions of each angle (i.e., pitch, yaw, and roll), which are presented in Fig. 6. The distribution of our synthetic data is uniform across each angle value of pitch, yaw, and roll respectively. On the contrary, the distribution of real data is mainly concentrated on certain angle values.

Fig. 5. Exemplar images from different datasets. We evaluate *Coarse Classification Phase* on three kinds of real datasets: Biwi Kinect Head Pose Dataset, Pointing'04 Dataset and Boston Dataset.

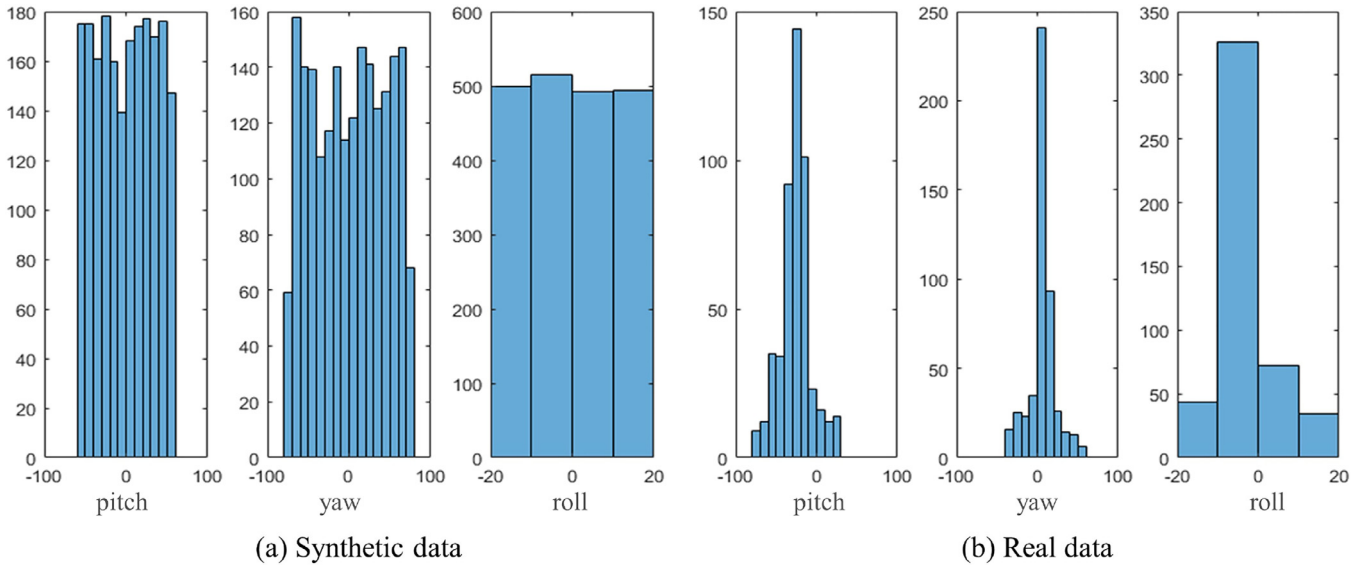

Fig. 6. The distribution of synthetic data and real data across each angle value, respectively, i.e., pitch, yaw, and roll. Obviously, the distribution of our synthetic data (vertical axis) is more uniform over the range of values for each angle (horizontal axis).

Table 2
Comparison of different classification strategy on both synthetic dataset and real datasets. Each row demonstrates the accuracy of one classification strategy, when taking different angles into account.

#	Category (θ_p, θ_y, θ_r)	Accuracy (Synthetic images)	Accuracy (Real images)
1	($\surd$, $\surd$, $\surd$)	76.95%	37.14%
2	($\surd$, $\surd$, $-$)	**99.92%**	**92.35%**
3	($\surd$, $-$, $\surd$)	65.25%	59.33%
4	($-$, $\surd$, $\surd$)	68.79%	62.16%
5	($\surd$, $-$, $-$)	98.11%	92.03%
6	($-$, $\surd$, $-$)	98.50%	91.46%
7	($-$, $-$, $\surd$)	77.47%	61.38%

4.2. Evaluation on coarse classification

Classification strategy. To further verify the efficiency of the classification strategy, we design six baselines, each of which corresponds to different category definition. As shown in Table 2, each row shows the accuracy of one classification strategy when taking different angles into account. The considered angle is marked by a tick on the corresponding column. The sign of each considered angle determines how to define categories. On the first row, there are 8 categories, where the solution space is split into eight parts according to the sign of pitch, yaw, and roll. For the 2nd to 4th rows, there are four categories on each row. For the 5th to 7th rows, there are two categories on each row.

For the network of each baseline, we train the network with synthetic images and real images with different labels. For testing, we use the remaining 10k synthetic images and 20k images from Biwi Dataset (10k), Pointing'04 Dataset (2k) and Boston Dataset (8k). All input RGB images are resized to a fixed size of 256×256 pixels. Table 2 illustrates the comparison results, which shows that the second strategy achieves the best performance with an average accuracy of 99.92% and 92.35% across the synthetic and real dataset respectively. It is clear that the performance of strategies considering the roll angle does not achieve satisfactory results. This is mainly due to the fact that the range of roll angle is relatively narrow and the head appearance along the roll axis is not obvious.

Head pose classifier. We test three classification models including AlexNet, VGGNet, and ResNet-50 on 10k synthetic images and around 13.8k real images (5.7k from Biwi Dataset, 0.7k from Pointing'04 Dataset, and 6.4k from Boston Dataset). As illustrated in

Table 3
Performance of head pose classification in *Coarse Classification* phase on different networks.

Method	Synthetic Dataset	Biwi Dataset	Pointing'04 dataset	Boston Dataset
AlexNet	99.89%	80.91%	76.71.%	71.75%
VGGNet	92.30%	84.02%	69.29%	80.05%
ResNet-50	98.88%	89.63%	89.00%	90.45%
Our method	**99.92%**	**90.16%**	**89.86%**	**92.86%**

Table 4
The mean error (ME) and standard deviation (STD) of head pose estimation on the real dataset (Biwi Dataset), expressed in degrees. The result in each cell of runtime represents computational estimation time for each head pose image measured on the same machine equipment. Note that, we only list the computation time of the methods that estimate all three head pose parameters (i.e., pitch, yaw, and roll).

Method	Pitch error (°)		Yaw error (°)		Roll error (°)		Runtime (ms)
	ME	STD	ME	STD	ME	STD	
Drouard et al.	8.85	9.97	8.7	9.0	–	–	–
Ricci et al.	10.5	–	9.1	–	–	–	–
Liu et al.	11.35	8.29	9.65	8.04	10.42	5.81	0.49
Ahn et al.	11.89	14.35	7.12	11.71	12.78	9.66	0.68
Our method	**5.48**	**3.23**	**4.76**	**4.33**	**4.29**	**3.30**	0.56

Table 5
Performance of different training dataset in both *Coarse Classification* phase and *Fine Regression* phase on real dataset (Biwi Dataset). The regression results are visualized by mean error (ME) and standard deviation (STD), expressed in degrees.

Phase	Method	Testing on Biwi Dataset					
Classification Phase	Baseline I	64.71%					
	Baseline II	59.47%					
	Our method	**90.16%**					
		Pitch error (°)		Yaw error (°)		Roll error (°)	
		ME	STD	ME	STD	ME	STD
Regression Phase	Baseline III	16.47	10.14	6.14	8.32	21.11	4.75
	Baseline IV	29.51	12.19	5.51	14.56	22.47	3.01
	Our method	**5.48**	**3.23**	**4.76**	**4.33**	**4.29**	**3.30**

Table 3, the quantitative results of our method with the shared layers from GoogLeNet are more accurate than other models.

Additionally, comparing the performance of our network with different training data: synthetic data only and real data only, the improvement of testing on the real dataset is significant: 64.71% (synthetic data only), 59.47% (real data only) $\rightarrow$ 92.35%, which shown in Table 5. The improvement verifies the effectiveness of our synthetic data, and also demonstrate the compatibility between synthetic data and real data on classification task.

4.3. Evaluation on fine regression

We adopt the testing set of Biwi Dataset for evaluating the performance of our regression network.

We compare our method to four state-of-the-art methods [15,16,39,40]. We opt for the mean error and the standard deviation of error metric, which is the commonly used evaluation criterion in head pose estimation. The results are illustrated in Table 4. In each cell, the first value refers to the mean error of pitch, yaw, and roll angle, respectively. The second value represents the standard deviation of error. It is clear that our regression network achieves impressive results on Biwi Dataset even with a relatively simple network architecture: the mean error is under 5.5° for all of the three angles, with a small standard deviation under 4.5°.

Moreover, we also compare the performance of our regression network with only real data during the training process. The testing results on Biwi Dataset exceed the acceptable range, i.e., 29.51 ± 12.19 shown in Table 5, which is due to the insufficient real datasets for training a deep network. Images from these real datasets suffer from the limited variation and number of available samples. Not surprisingly, the reported results demonstrate significant improvement when training with both synthetic and real data, thus verifying the compatibility between these two types of data consequently.

Overall, our two sub-networks estimate promising results on par with the state-of-the-art approaches. Moreover, the computation time for each testing head pose image in Biwi Dataset achieves 0.56ms. Considering the shared convolutional layers at the bottom of these two networks, our model achieves a good tradeoff between performance and computation efficiency.

4.4. Discussions on various conditions

Inhomogeneous illumination, partial occlusions and motion blur are very common in real images or videos. To further verify that our *Coarse-to-Fine* cascade CNNs are robust to such situations, we show that our networks can also be tested on a variety of images that involve different illumination conditions, occlusions and blurs. Since the annotations of this kind of real images are scarce, we further generate another 50*k* synthetic RGB head pose images for testing. Please refer to our *supplementary materials* for the exemplary generated images.

Inhomogeneous illumination. The image quality obtained by cameras in the wild is always affected by varying illumination conditions. To better understand the influence of illumination conditions, we test our approach on 30*k* synthetic images that are rendered under different illumination conditions, i.e., different illumination angles, intensities, and colors.

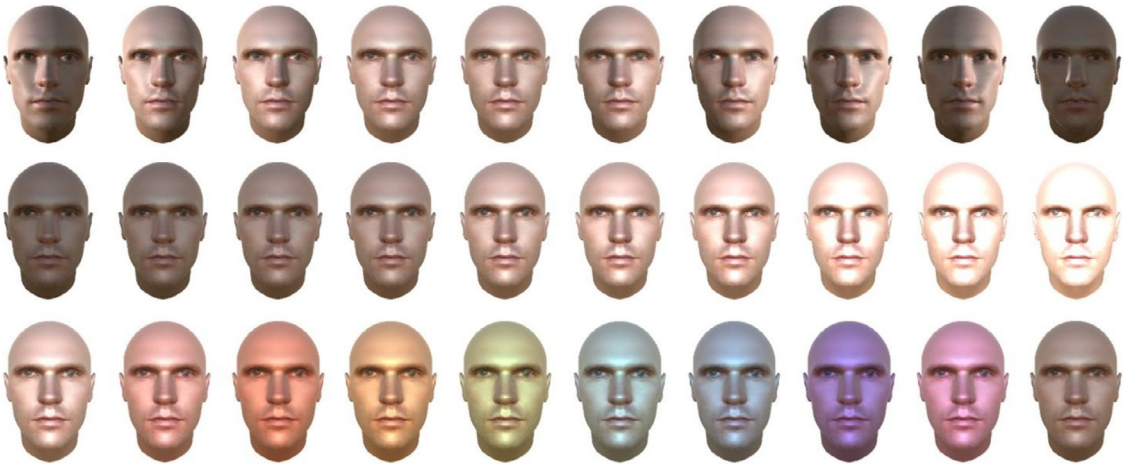

Fig. 7. Exemplar images of different illumination conditions. please in color print for better visualization.

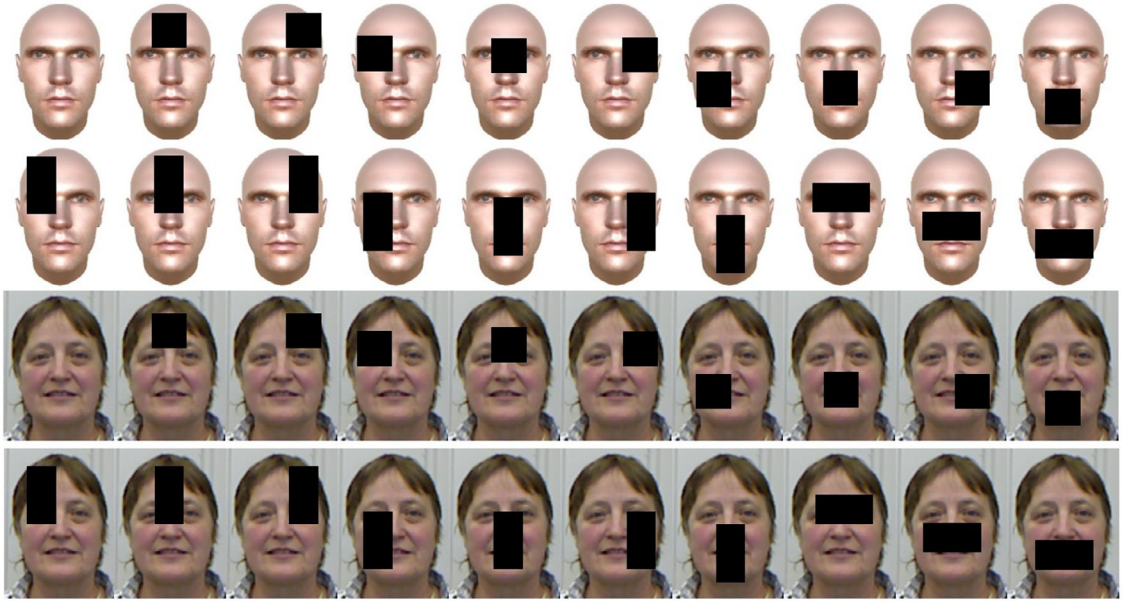

Fig. 8. Exemplar images of different occlusions.

Table 6
The mean error (ME) and standard deviation (STD) of head pose estimation under different illumination conditions, expressed in degrees.

Illumination conditions	Pitch error (°)		Yaw error (°)		Roll error (°)		Runtime (ms)
	ME	STD	ME	STD	ME	STD	
Angles	6.23	5.13	7.20	5.02	5.44	4.21	0.57
Intensities	6.09	4.94	8.25	5.30	5.33	5.47	0.62
Colors	6.92	4.42	9.57	5.62	4.99	6.75	0.53

We apply the following settings for each rendered frame: (1) for different angles of illumination, we only change the horizontal direction of the illumination angle. The illumination angles range from -100° to 100° with an interval of 20°. (2) For illumination intensity, we set the range from 0.3 to 1.2 with an interval of 0.1. (3) For different colors, we choose ten typical colors for testing, including red, orange, yellow, green, cyan, blue, purple, pink, gray and white. As shown in Fig. 7, the newly rendered images under different conditions are significantly different from the training images.

In Table 6, we use the same head pose images rendered under different illumination conditions for testing and we only consider performance on synthetic data. Our approach achieves 93.55%, 96.36%, and 90.20% in classification accuracy for different conditions, *i.e.*, different angles, intensities, and colors. The mean and standard deviation of the final head pose estimation errors tend to be consistent. Although the mean errors of each parameter have increased slightly, it still implies that the cascade network and synthetic dataset can effectively solve the problem. In Table 6, we include the corresponding computation time. We can observe that our model only takes about 0.57ms to process a head pose image under different illumination conditions.

Motion blur. It is common for the subject's head to move during image capturing, which causes motion blur. Handling the

Table 7
The mead error (ME) and standard deviation (STD) of head pose estimation on blurred images and images with occlusions, expressed in degrees.

Test	Pitch error (°)		Yaw error (°)		Roll error (°)		Runtime (ms)
images	ME	STD	ME	STD	ME	STD	
Blurred	6.39	7.11	8.07	7.75	6.30	5.97	0.60
Occlusion	7.41	8.71	8.48	8.09	7.88	4.13	0.67

Fig. 9. Exemplar images of blurred images. (a) The original synthetic image and the corresponding blurred image. (b) The real image and the corresponding blurred image.

Table 8
The mead error (ME) and standard deviation (STD) of different similarity measures of head pose estimation on real depth images from Biwi Dataset, expressed in degrees. The result in each cell of runtime represents computational estimation time for each head pose image measured on the same computer configuration.

Method	Pitch error (°)		Yaw error (°)		Roll error (°)		Runtime (ms)
	ME	STD	ME	STD	ME	STD	
Fanelli et al.	5.2	7.7	6.6	12.6	6.0	7.1	2.79
Wang et al.	8.8	14.3	8.5	11.1	7.4	10.8	40
Borghi et al.	4.5	4.7	5.3	5.2	7.5	7.3	29
Our method	**4.23**	**5.13**	**5.16**	**5.32**	**5.39**	**2.61**	**1.89**

effect of motion blur is an important problem in many visual analysis types of research. Therefore, we test our approach on its ability to deal with such a situation. We blurred the head pose images with a Gaussian function and obtain a collection of 10*k* images. Fig. 9 shows some examples of both synthetic images and real images.

The classification accuracy achieves 89.05% under such blurring conditions. The estimation results of blurred real images are shown in the first row of Table 7. Compared to the results in Table 4, it is clear that the results decrease obviously in pitch and yaw. The reason is that our generated training images are all in high resolution with no blur.

Occlusion. Real images of head pose often have occlusions, which may be caused by the camera's view or some minor self-occlusions. To further test the strength of our approach in handling occlusions of images, we generate 10*k* RGB images covered with black masks (shown in Fig. 8). We use a fixed size of a black square with 60×60 pixels and a black rectangle with 60×120 pixels to randomly cover both synthetic images and real images.

The classification accuracy achieves 71.29%, and the estimation results are reported in the second row of Table 7, which indicate that the cascade CNNs can effectively estimate the occluded images. From the table, we can see that the accuracy has declined slightly. In order to enhance the robustness of our proposed method, we will improve the approach of synthesizing images to mimic more realistic images and add them to the training data.

4.5. Leveraging depth images

With the development of sensing technologies, such as Time-of-Flight sensors and the Microsoft Kinect, capturing RGB-D images becomes easy. New methods that work on depth images are emerging, such as human pose detection in depth frames [41]. To give a deeper insight of our proposed coarse-to-fine method, we explore whether our approach can be extended to the depth images to estimate head poses.

We generate a depth head pose dataset using a similar approach of generating the synthetic RGB images, which includes 60*k* depth images. The depth images are obtained by calculating the distance between the camera and each pixel of the head model using Physics Raycast in Unity 3D. We compute the distance between the camera and the intersection point as the depth value. Some generated depth images are shown in Fig. 10. Similar to the process of the generated RGB images, the generated depth images are preprocessed by cropping to keep the face region only; resizing to 256×256 pixels; normalizing the pixel values of depth images to the range of [0,1].

We compare our method with three proposed methods on depth images: *random forest-based estimation [3]*, *SIFT-HOG based estimation [42]*, and *CNN based estimation [43]*. The mean and standard deviation of the head pose estimation error are reported in Table 8. Note that all these approaches are tested on Biwi Dataset, where the images are captured by Kinect. Although the standard deviations of pitch and yaw estimation of our approach are slightly higher than that of other methods, it still demonstrates that our cascade networks can achieve reasonable performance on depth images: the mean error is under 5.4° for all of the three angles. For all the methods, we include their computation time. All timings were measured on the same computer configuration. We can observe that our method only takes about 1.89ms to estimate a head pose depth image, and is the fastest one. Moreover, for the non-deep learning method [3], it is the fastest method (7.7ms per image on CPU).

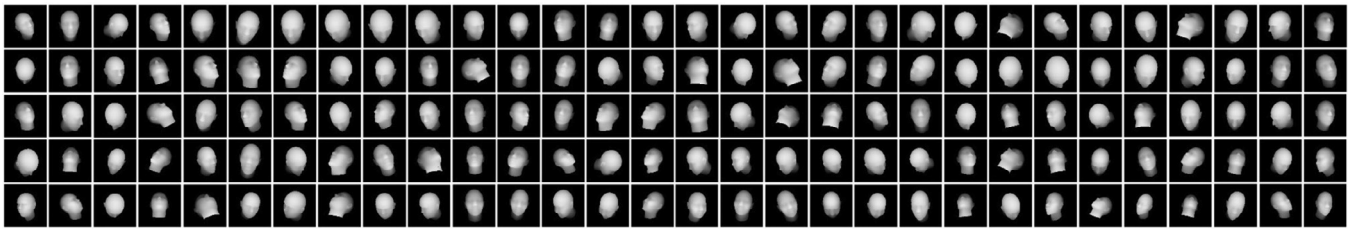

Fig. 10. Examplar images of depth head poses. (For visualization of the depth images, we rescale the range of pixel value.)

5. Conclusion

This work proposed a novel head pose estimation model, which explores to leverage synthetic data to train a *Coarse-to-Fine* deep network in a joint leaning manner. Extensive experiments have been conducted on several publicly available benchmarks and detailed analysis are reported on issues such as the estimation accuracy on images with different illumination conditions, motion blur, and occlusions.

Currently, we have synthesized head pose images with different gender, age, and race, but our synthesis framework is flexible to accommodate additional criteria such as different emotion expressions, hairstyles, and accessories. Moreover, the proposed image synthesis pipeline can be easily modified so as to synthesize diverse images for solving different tasks, *e.g.*, hand pose estimation, human pose estimation, and gait recognition, avoiding the process of manually collecting and annotating data.

The proposed approach can be applied to support different applications. For example, in driving assistance, the proposed pose estimation approach can help to detect human attention and fatigue driving, which will improve the driving safety. In entertainment, our approach can collaborate with current AVATAR technique [44,45] to animate a virtual character, which can have the same pose with the user in the real world. In addition, our approach may assist the disabled people to interact with computer or robot in a natural way.

Limitation and Future Work. The main practical limitation is that our head pose is estimated in terms of the single input image. This enables the generalization of our approach to any applications but ignoring contextual information when applied to video. Moving forward, one interesting direction for future work is to considering contextual head pose constraints to limit it to a more subtle range, thus improving the final parameter estimation accuracy.

By learning with depth head pose images, we are able to learn a model that generalizes to any illumination conditions. An interesting future direction would be to train a neural network on depth images with motion blur and occlusions.

Acknowledgments

This research is supported by the Natural Science Foundation of China (NSFC) under award number 61876020.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2019.05.026.

References

[1] A. Doshi, M.M. Trivedi, Head and eye gaze dynamics during visual attention shifts in complex environments, J. Vis. 12 (2) (2012). 9–9
[2] E. Murphy-Chutorian, M.M. Trivedi, Head pose estimation in computer vision: a survey, in: Proceedings of the TPAMI, 31, 2009, pp. 607–626.
[3] G. Fanelli, M. Dantone, J. Gall, A. Fossati, L. Van Gool, Random forests for real time 3d face analysis, in: Proceedings of the IJCV, 101, 2013, pp. 437–458.
[4] N. Gourier, D. Hall, J.L. Crowley, Estimating face orientation from robust detection of salient facial structures, in: Proceedings of the FG Net Workshop on Visual Observation of Deictic Gestures, 6, 2004.
[5] S.S. Mukherjee, N.M. Robertson, Deep head pose: gaze-direction estimation in multimodal video, in: Proceedings of the TMM, 17, 2015, pp. 2094–2107.
[6] Y. Lang, W. Liang, F. Xu, Y. Zhao, L.-F. Yu, Synthesizing personalized training programs for improving driving habits via virtual reality, in: Proceedings of the IEEE Conference on Virtual Reality, 2018.
[7] S. Qi, W. Wang, B. Jia, J. Shen, S.-C. Zhu, Learning human-object interactions by graph parsing neural networks, in: Proceedings of the ECCV, 2018, pp. 401–417.
[8] C. Li, W. Liang, C. Quigley, Y. Zhao, L.-F. Yu, Earthquake safety training through virtual drills, in: Proceedings of the TVCG, 23(4), 2017, pp. 1275–1284.
[9] W. Liang, J. Liu, y. Lang, B. Ning, L.-F. Yu, Functional workspace optimization via learning personal preferences from virtual experiences, in: Proceedings of the TVCG, 25(5), 2019, pp. 1836–1845.
[10] S. Sheikhi, J.-M. Odobez, Combining dynamic head pose–gaze mapping with the robot conversational state for attention recognition in human–robot interactions, Pattern Recognit. Lett. 66 (2015) 81–90.
[11] B. Han, S. Lee, H.S. Yang, Head pose estimation using image abstraction and local directional quaternary patterns for multiclass classification, Pattern Recognit. Lett. 45 (2014) 145–153.
[12] V. Drouard, R. Horaud, A. Deleforge, S.O. Ba, G.D. Evangelidis, Robust head-pose estimation based on partially-latent mixture of linear regressions, in: Proceedings of the TIP, 26, 2017, pp. 1428–1440.
[13] S. Malassiotis, M.G. Strintzis, Robust real-time 3d head pose estimation from range data, Pattern Recognit. 38 (8) (2005) 1153–1165.
[14] F.H.d.B. Zavan, O.R. Bellon, L. Silva, G.G. Medioni, Benchmarking parts based face processing in-the-wild for gender recognition and head pose estimation, Pattern Recognit. Lett. 123 (2019) 104–110.
[15] X. Liu, W. Liang, Y. Wang, S. Li, M. Pei, 3d head pose estimation with convolutional neural network trained on synthetic images, in: Proceedings of the ICIP, 2016, pp. 1289–1293.
[16] B. Ahn, J. Park, I.S. Kweon, Real-time head orientation from a monocular camera using deep neural network, in: Proceedings of the ACCV, 2014, pp. 82–96.
[17] C. Hong, J. Yu, J. Zhang, X. Jin, K.-H. Lee, Multi-modal face pose estimation with multi-task manifold deep learning, IEEE Trans. Ind. Inf. (2018). 1–1
[18] J. Yu, Y. Rui, D. Tao, Click prediction for web image reranking using multimodal sparse coding, in: Proceedings of the TIP, 23, 2014, pp. 2019–2032.
[19] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep autoencoder for human pose recovery, in: Proceedings of the TIP, 24, 2015, pp. 5659–5670.
[20] C. Hong, J. Yu, D. Tao, M. Wang, Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval, IEEE Trans. Ind. Electron. 62 (6) (2015) 3742–3751.
[21] C. Hong, J. Yu, J. You, X. Chen, D. Tao, Multi-view ensemble manifold regularization for 3d object recognition, Inf. Sci. 320 (2015) 395–405.
[22] W. Liu, X. Ma, Y. Zhou, D. Tao, J. Cheng, P-laplacian regularization for scene recognition, IEEE Trans. Cybern. (99) (2018) 1–14.
[23] J. Zhang, K. Mei, Y. Zheng, J. Fan, Learning multi-layer coarse-to-fine representations for large-scale image classification, Pattern Recognit. 91 (2019) 175–189.
[24] H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, A convolutional neural network cascade for face detection, in: Proceedings of the CVPR, 2015, pp. 5325–5334.
[25] G. Pavlakos, X. Zhou, K.G. Derpanis, K. Daniilidis, Coarse-to-fine volumetric prediction for single-image 3d human pose, in: Proceedings of the CVPR, 2017, pp. 1263–1272.
[26] J. Wu, M.M. Trivedi, A two-stage head pose estimation framework and evaluation, Pattern Recognit. 41 (3) (2008) 1138–1158.
[27] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, A.M. Lopez, The synthia dataset: a large collection of synthetic images for semantic segmentation of urban scenes, in: CVPR, 2016, pp. 3234–3243.
[28] T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini, E. Magli, Robust license plate recognition using neural networks trained on synthetic images, Pattern Recognit. 93 (2019) 134–146.
[29] H. Wang, J. Yang, W. Liang, X. Tong, Deep single-view 3d object reconstruction with visual hull embedding, in: Proceedings of the AAAI, 2019.
[30] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, R. Cipolla, Understanding real world indoor scenes with synthetic data, in: Proceedings of the CVPR, 2016, pp. 4077–4085.
[31] L. Pishchulin, A. Jain, M. Andriluka, T. Thormählen, B. Schiele, Articulated people detection and pose estimation: Reshaping the future, in: Proceedings of the CVPR, 2012, pp. 3178–3185.
[32] I.K. Kallel, S. Almouahed, B. Solaiman, É. Bossé, An iterative possibilistic knowledge diffusion approach for blind medical image segmentation, Pattern Recognit. 78 (2018) 182–197.
[33] C. Gou, H. Zhang, K. Wang, F.-Y. Wang, Q. Ji, Cascade learning from adversarial synthetic images for accurate pupil detection, Pattern Recognit. 88 (2019) 584–594.
[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the CVPR, 2015, pp. 1–9.
[35] V.F. Ferrario, C. Sforza, G. Serrao, G. Grassi, E. Mossi, Active range of motion of the head and cervical spine: a three-dimensional investigation in healthy young adults, J. Orthop. Res. 20 (1) (2002) 122–129.
[36] J.T. Barron, J. Malik, Intrinsic scene properties from a single RGB-D image, in: Proceedings of the TPAMI, 38, 2016, pp. 690–703.
[37] S. Wu, M. Kan, Z. He, S. Shan, X. Chen, Funnel-structured cascade for multi-view face detection with alignment-awareness, Neurocomputing 221 (2017) 138–145.
[38] M. La Cascia, S. Sclaroff, V. Athitsos, Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3d models 22 (4) (2000) 322–336.
[39] V. Drouard, S. Ba, G. Evangelidis, A. Deleforge, R. Horaud, Head pose estimation via probabilistic high-dimensional regression, in: Proceedings of the ICIP, 2015, pp. 4624–4628.
[40] E. Ricci, J.-M. Odobez, Learning large margin likelihoods for realtime head pose tracking, in: Proceedings of the ICIP, 2009, pp. 2593–2596.
[41] L. Kong, X. Yuan, A.M. Maharjan, A hybrid framework for automatic joint detection of human poses in depth frames, Pattern Recognit. 77 (2018) 216–225.
[42] B. Wang, W. Liang, Y. Wang, Y. Liang, Head pose estimation with combined 2d sift and 3d hog features, in: Proceedings of the ICIG, IEEE, 2013, pp. 650–655.

[43] G. Borghi, M. Venturelli, R. Vezzani, R. Cucchiara, Poseidon: face-from-depth for driver pose estimation, in: Proceedings of the CVPR, 2017, pp. 5494–5503.
[44] Y. Lang, W. Liang, L.-F. Yu, Virtual agent positioning driven by scene semantics in mixed reality, in: Proceedings of the IEEE Virtual Reality, 2019.
[45] Y. Lang, W. Liang, Y. Wang, L.-F. Yu, 3d face synthesis driven by personality impression, in: Proceedings of the AAAI, 2019.

Yujia Wang recieved the B.S. in the school of computer science in 2014 from Beijing Institute of Technology (BIT), Beijing, China. She is currently a Ph.D. candidate in computer science at BIT. Her research interests lie in the field of Computer Vision and Computer Graphics.

Wei Liang is an associate professor with the School of Computer Science and Technology, Beijing Institute of Technology (BIT), Beijing, China. She received her Ph.D. in computer science from BIT in 2005. She worked as a visiting professor at Vision Cognition Learning and Autonomous (VCLA), UCLA, USA in 2014 and 2015. Her research has been published in top Vision, AI, and VR venues, including *ICCV, AAAI, IJCAI*, and *IEEE VR*. Her research interests lie in the field of Computer Vision, HCI and Cognitive Science.

Jianbing Shen (M'11-SM'12) is a Full Professor with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. He has obtained many flagship honors including the Fok Ying Tung Education Foundation from Ministry of Education, the Program for Beijing Excellent Youth Talents from Beijing Municipal Education Commission, and the Program for New Century Excellent Talents from Ministry of Education. He worked as a visiting professor in the Department of Information Technology and Electrical Engineering, ETH Zurich, Switzerland, and the Department of Computer Science, University of California, Los Angeles, USA. His research interests include computer vision and pattern recognition. He serves as an associate editor on the editorial boards of *Neurocomputing*.

Yunde Jia is a professor with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. He was a visiting scientist at Carnegie Mellon University (CMU) between 1995 and 1997, and visiting fellow at the Australian National University (ANU) in 2011. He received the B.S., M.S. and Ph.D. degrees in Mechatronics from the BIT in 1983, 1986 and 2000, respectively. He has published more than 200 academic papers and accessed to 12 national invention patents. His research interests include computer vision, media computing and intelligent systems.

Lap-Fai Yu is a professor at George Mason University. He obtained his Ph.D. degree in Computer Science from UCLA in 2013, where he received the Outstanding Recognition in Research in Computer Science. He is the recipient of the Cisco Outstanding Graduate Research Award, the Award of Excellence from Microsoft Research, the UCLA Dissertation Year Fellowship and the Sir Edward Youde Memorial Fellowship. His research has been published in top graphics and vision venues, including *ACM Transactions on Graphics (SIGGRRAPH), IEEE CVPR*, and *IEEE ICCV*; was awarded the Best Paper Honorable Mention in 3DV 2016; and has been featured in New Scientist, the UCLA Headlines, the UCLA Mathematics Newsletter and newspapers internationally.